Data Mining
Cluster Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 7

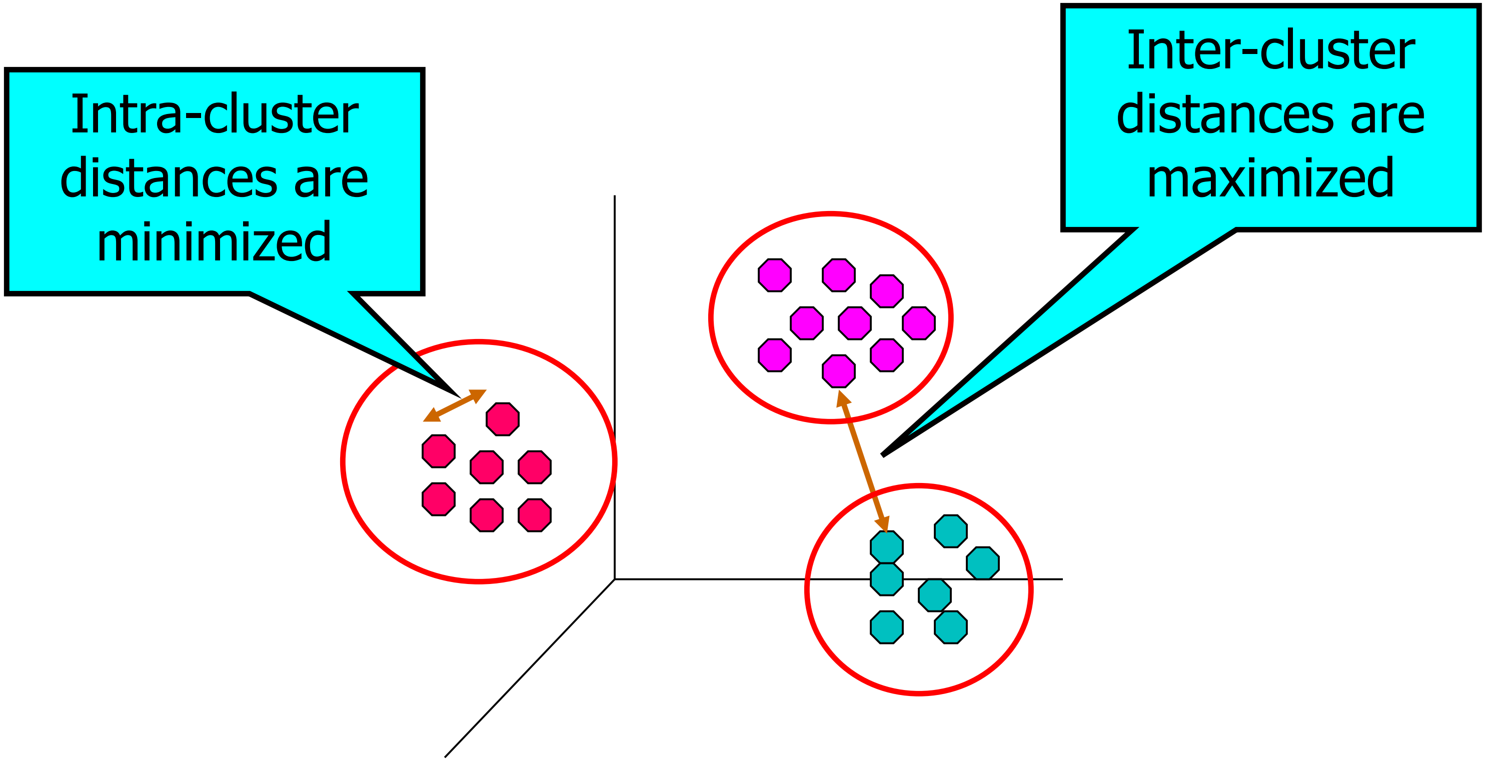
Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

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

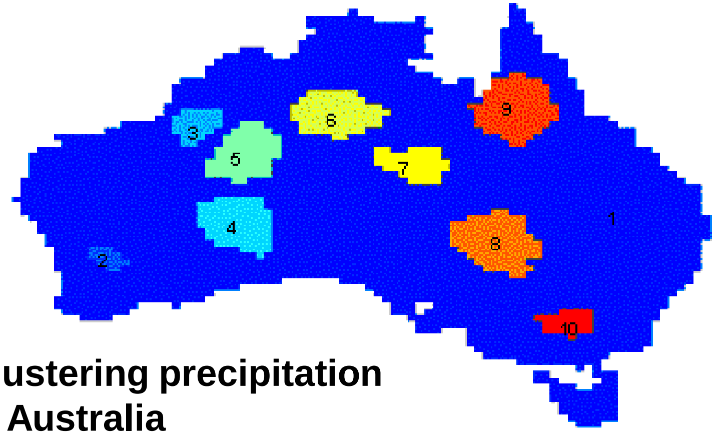
Inter-cluster distances are maximized

Applications of Cluster Analysis

- **Understanding**
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	Discovered Clusters	*Industry Group*
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP

- **Summarization**
  - Reduce the size of large data sets

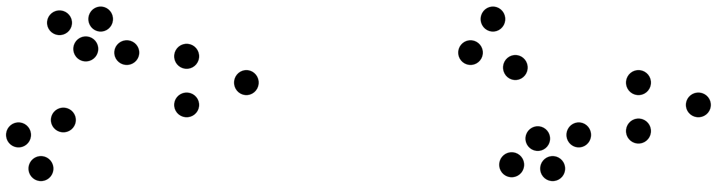


Clustering precipitation in Australia

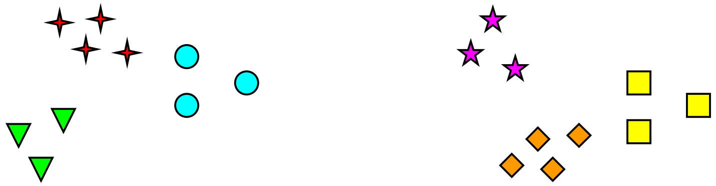
What is not Cluster Analysis?

- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification
  - Clustering is a grouping of objects based on the data
- Supervised classification
  - Have class label information
- Association Analysis
  - Local vs. global connections

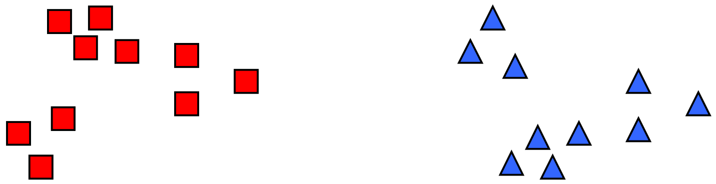
Notion of a Cluster can be Ambiguous



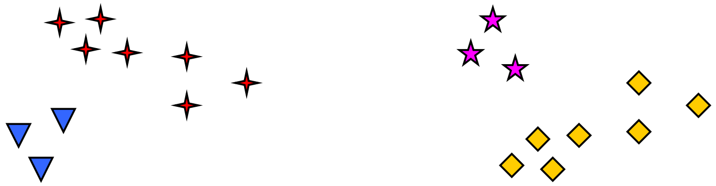
How many clusters?



Six Clusters



Two Clusters



Four Clusters

Types of Clusterings

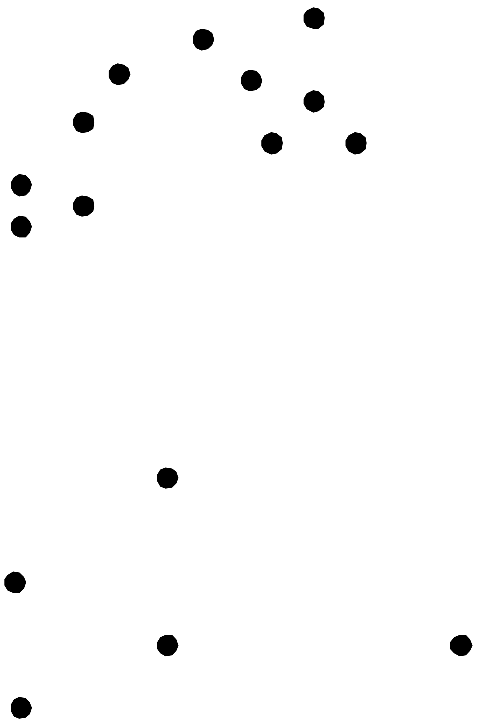
- A clustering is a set of clusters

- Important distinction between hierarchical and partitional sets of clusters

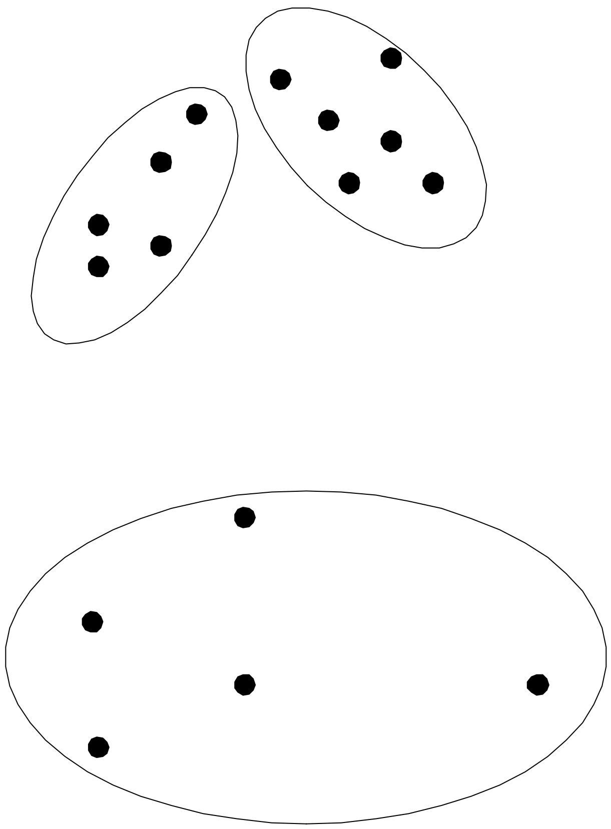
- Partitional Clustering
  - A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering

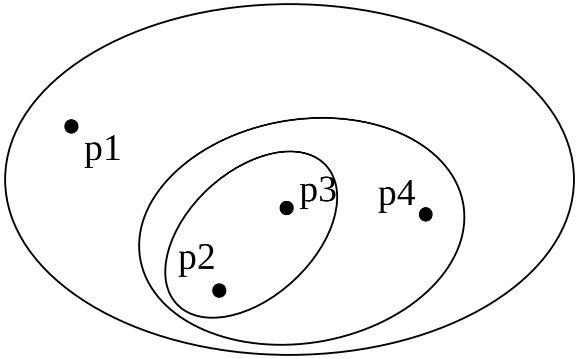


Original Points

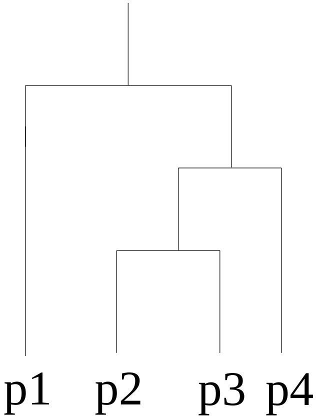


A Partitional Clustering

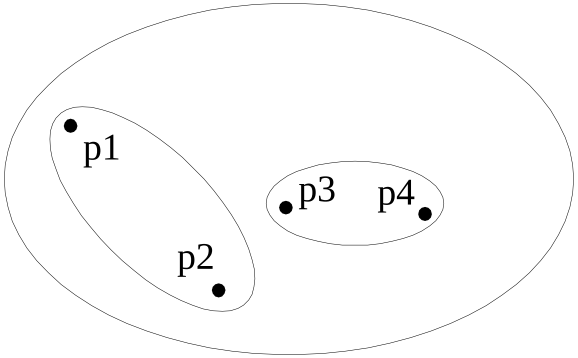
Hierarchical Clustering



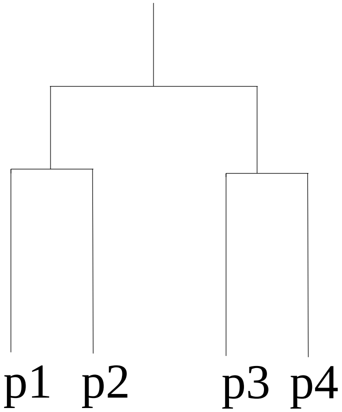
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- 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
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Clusters of widely different sizes, shapes, and densities

Types of Clusters

- Well-separated clusters

- Center-based clusters

- Contiguous clusters

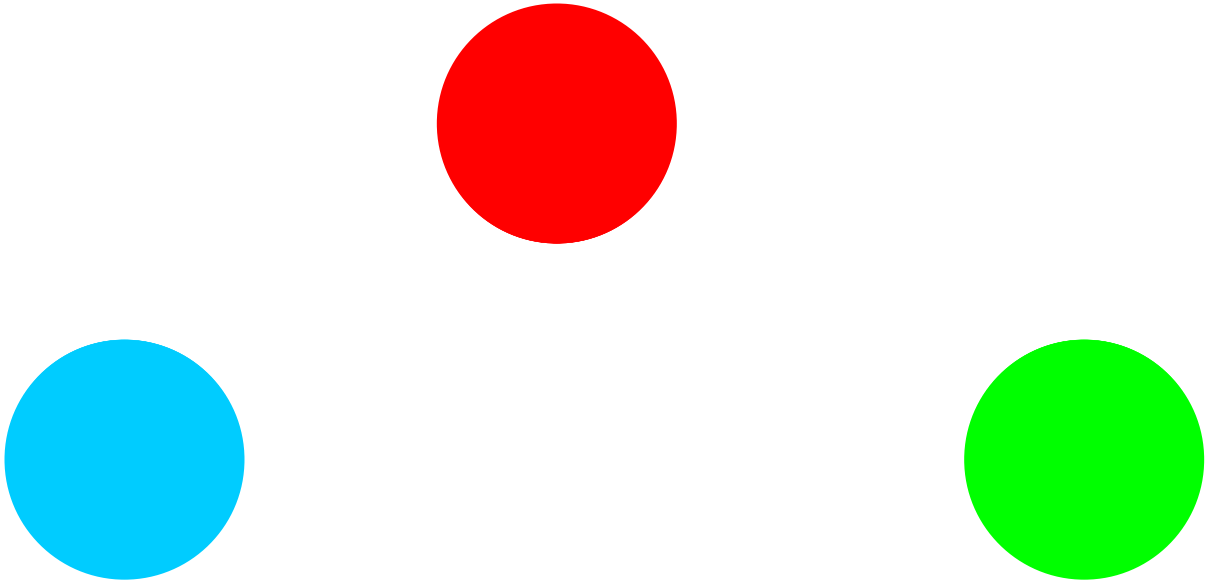
- Density-based clusters

- Property or Conceptual

- Described by an Objective Function

Types of Clusters: Well-Separated

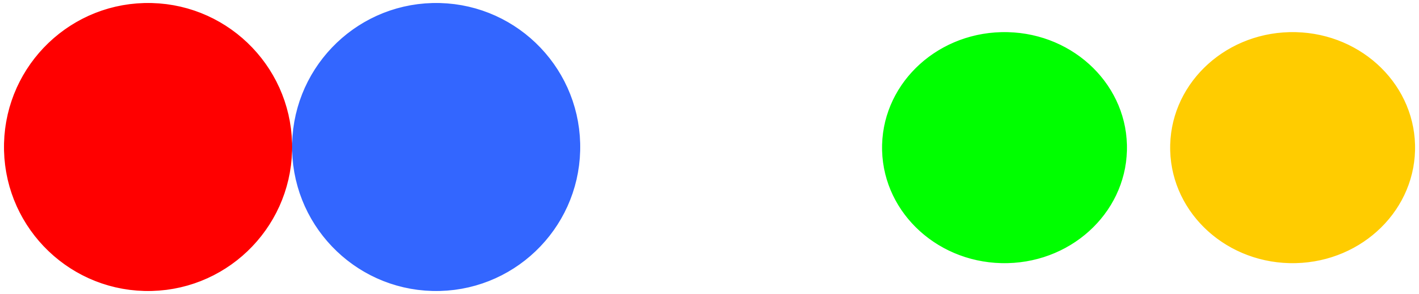
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Center-Based

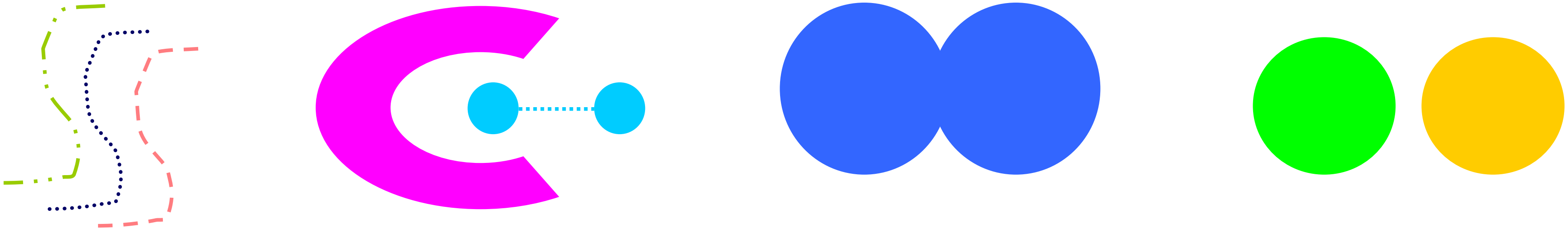
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid, the most "representative" point of a cluster



4 center-based clusters

Types of Clusters: Contiguity-Based

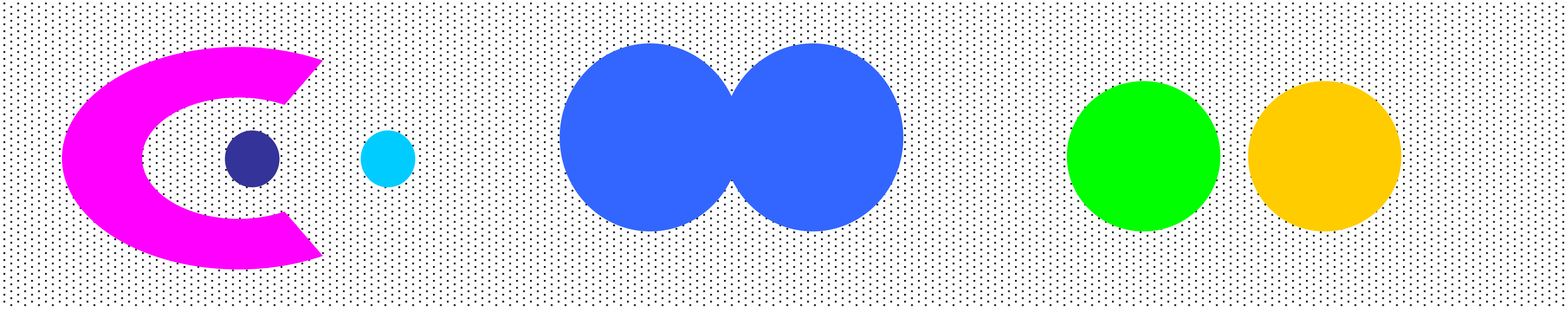
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.

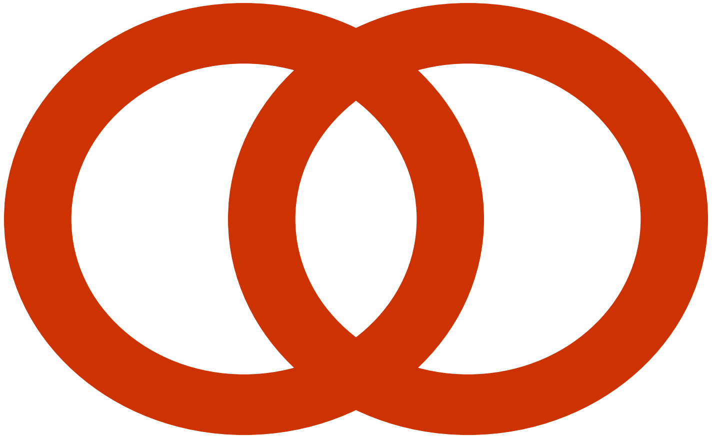


6 density-based clusters

Types of Clusters: Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

.



2 Overlapping Circles

Types of Clusters: Objective Function

- Clusters Defined by an Objective Function
  - Finds clusters that minimize or maximize an objective function.
  - Enumerate all possible ways of dividing the points into clusters and evaluate the `goodness' of each potential set of clusters by using the given objective function. (NP Hard)
  - Can have global or local objectives.
    - Hierarchical clustering algorithms typically have local objectives
    - Partitional algorithms typically have global objectives
  - A variation of the global objective function approach is to fit the data to a parameterized model.
    - Parameters for the model are determined from the data.
    - Mixture models assume that the data is a ‘mixture' of a number of statistical distributions.

Map Clustering Problem to a Different Problem

- Map the clustering problem to a different domain and solve a related problem in that domain
  - Proximity matrix defines a weighted graph, where the nodes are the points being clustered, and the weighted edges represent the proximities between points
  - Clustering is equivalent to breaking the graph into connected components, one for each cluster.
  - Want to minimize the edge weight between clusters and maximize the edge weight within clusters

Characteristics of the Input Data Are Important

- Type of proximity or density measure
  - Central to clustering
  - Depends on data and application

- Data characteristics that affect proximity and/or density are
  - Dimensionality
    - Sparseness
  - Attribute type
  - Special relationships in the data
    - For example, autocorrelation
  - Distribution of the data

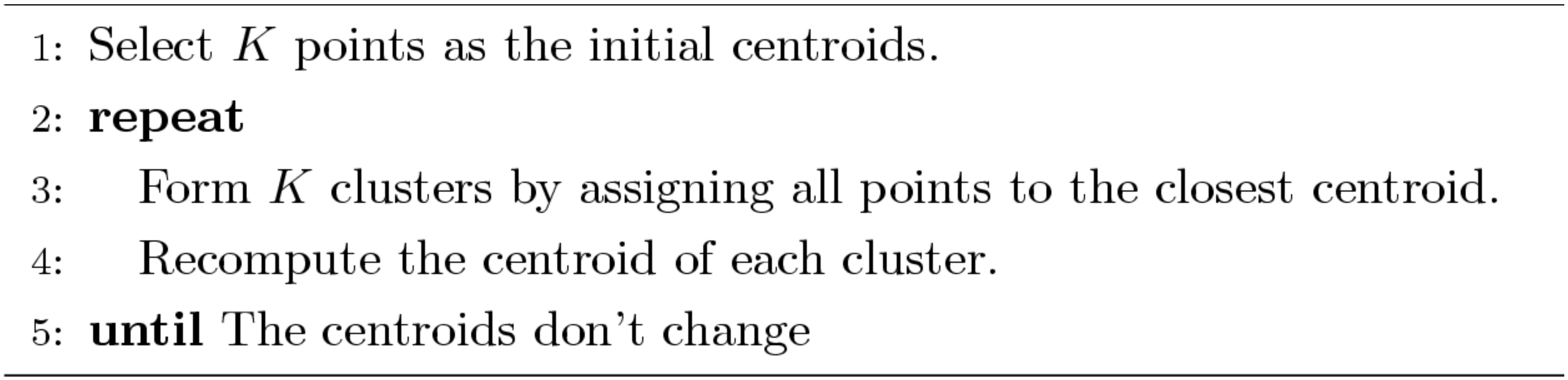
- Noise and Outliers
  - Often interfere with the operation of the clustering algorithm

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- Density-based clustering

K-means Clustering

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



```
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
```

Example

- The following observation values are desired to be clustered with the k-means method.

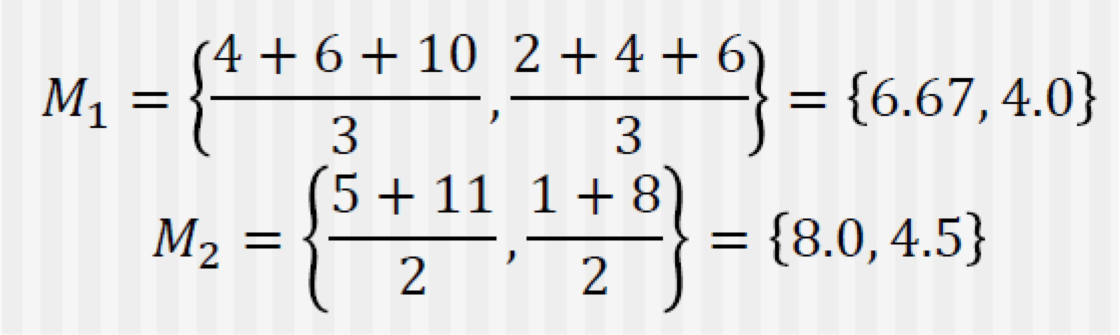
	Attribute 1	Attribute 2
X1	4	2
X2	6	4
X3	5	1
X4	10	6
X5	11	8

- The number of sets is initially considered k = 2. Two sets are randomly determined.

$$C_1 = \{X_1, X_2, X_4\}$$
$$C_2 = \{X_3, X_5\}$$

	Attribute 1	Attribute 2	Cluster
X1	4	2	C1
X2	6	4	C1
X3	5	1	C2
X4	10	6	C1
X5	11	8	C2

- Step 1. The centers of the two specified clusters are calculated as follows.



$$M_1 = \left\{\frac{4+6+10}{3}, \frac{2+4+6}{3}\right\} = \{6.67, 4.0\}$$

$$M_2 = \left\{\frac{5+11}{2}, \frac{1+8}{2}\right\} = \{8.0, 4.5\}$$

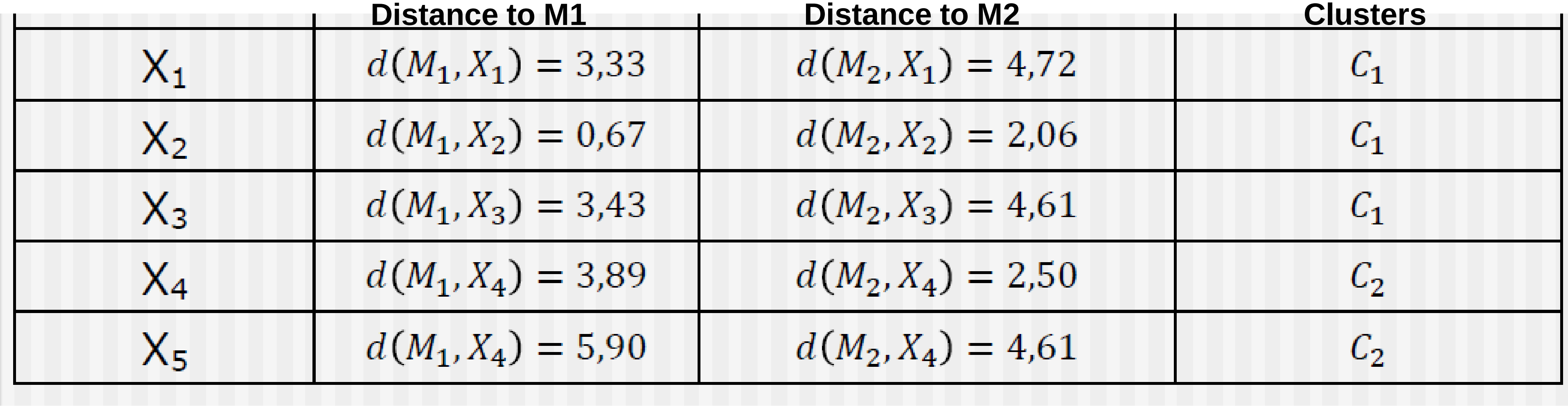
- Since the distances from the M1 and M2 centers are desired to be minimum, the following calculations are made. These distances are calculated using the Euclidean distance formula.
- Distance between X1 and M1

$$d(M_1, X_1) = \sqrt{(6{,}67 - 4)^2 + (4 - 2)^2} = 3{,}33$$

- Distance between X1 and M2

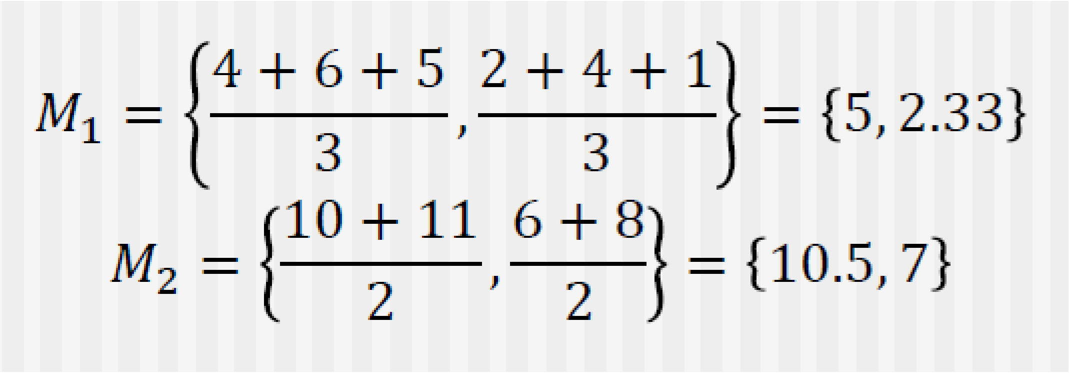
$$d(M_2, X_1) = \sqrt{(8 - 4)^2 + (4{,}5 - 2)^2} = 4{,}72$$

- As a result of these operations, considering the distances of $X1$ to centers $M1$ and $M2$, it is seen that $d(M1, X1) < d(M2, X1)$. In this case, it is understood that the center $M1$ is closer to the observation value $X1$. So it is considered to be $X1 \in C1$. Similarly, a table is created for all observation values.



	Distance to M1	Distance to M2	Clusters
X_1	$d(M_1,X_1) = 3{,}33$	$d(M_2,X_1) = 4{,}72$	C_1
X_2	$d(M_1,X_2) = 0{,}67$	$d(M_2,X_2) = 2{,}06$	C_1
X_3	$d(M_1,X_3) = 3{,}43$	$d(M_2,X_3) = 4{,}61$	C_1
X_4	$d(M_1,X_4) = 3{,}89$	$d(M_2,X_4) = 2{,}50$	C_2
X_5	$d(M_1,X_4) = 5{,}90$	$d(M_2,X_4) = 4{,}61$	C_2

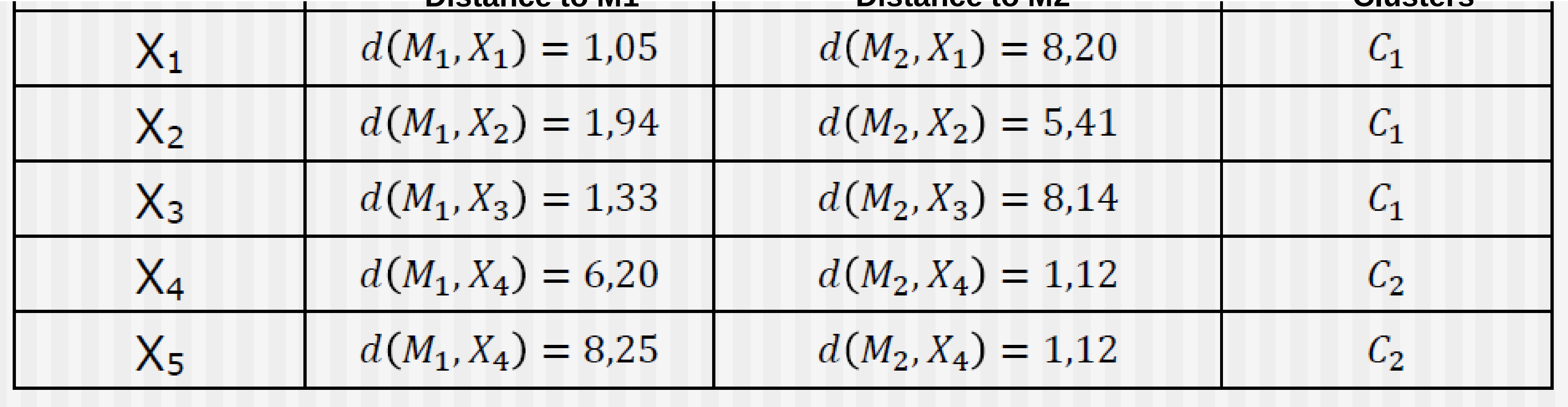
- The centers of the two newly found clusters are calculated as follows.



$$M_1 = \left\{\frac{4+6+5}{3}, \frac{2+4+1}{3}\right\} = \{5, 2.33\}$$

$$M_2 = \left\{\frac{10+11}{2}, \frac{6+8}{2}\right\} = \{10.5, 7\}$$

- The distances of all observations to the new centers are calculated again.

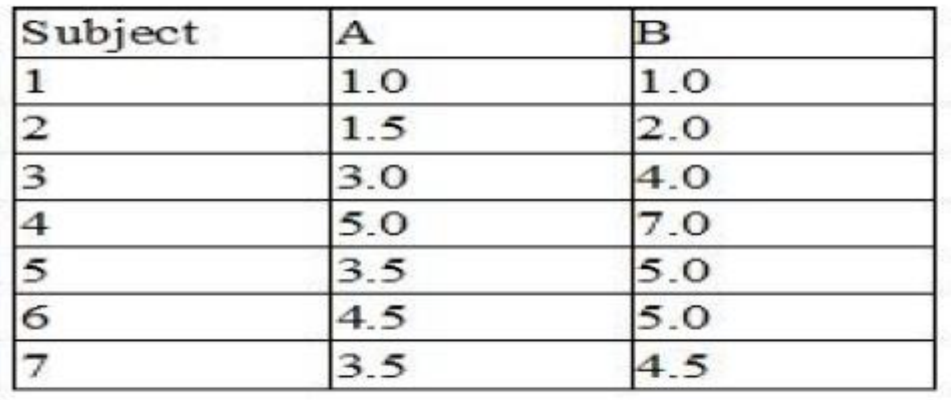


	Distance to M1	Distance to M2	Clusters
X_1	$d(M_1, X_1) = 1{,}05$	$d(M_2, X_1) = 8{,}20$	C_1
X_2	$d(M_1, X_2) = 1{,}94$	$d(M_2, X_2) = 5{,}41$	C_1
X_3	$d(M_1, X_3) = 1{,}33$	$d(M_2, X_3) = 8{,}14$	C_1
X_4	$d(M_1, X_4) = 6{,}20$	$d(M_2, X_4) = 1{,}12$	C_2
X_5	$d(M_1, X_4) = 8{,}25$	$d(M_2, X_4) = 1{,}12$	C_2

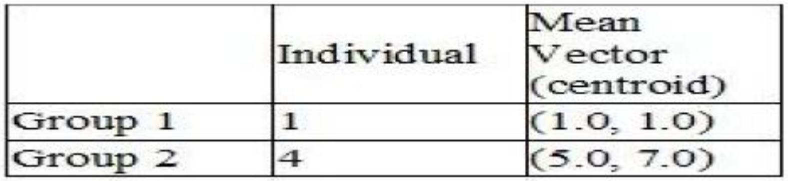
Since there is no change in the clusters compared to the previous step, iteration is ended.

K-Means: Step-By-Step Example

- As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables on each of seven individuals: This data set is to be grouped into two clusters. As a first step in finding a sensible initial partition, let the A & B values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster means, giving:

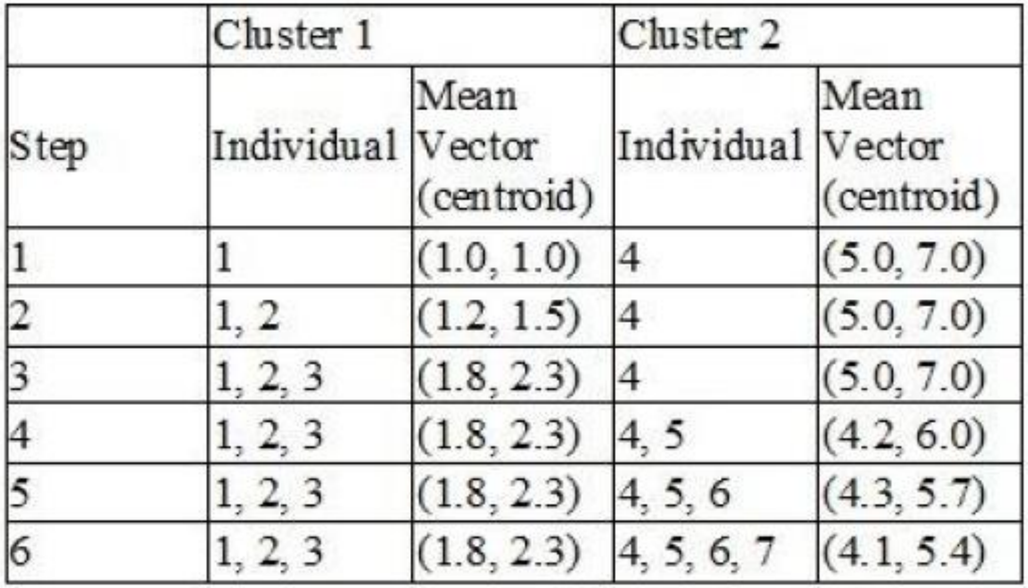


Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5



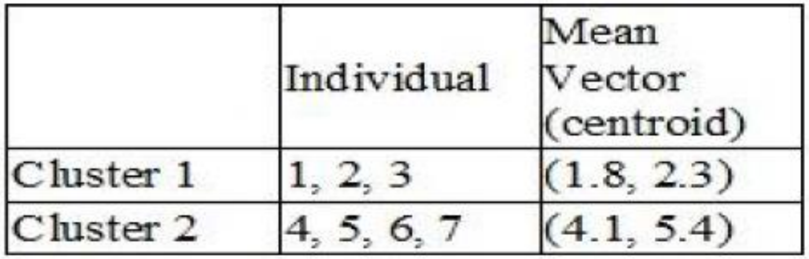
	Individual	Mean Vector (centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

- The remaining individuals are now examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean. The mean vector is recalculated each time a new member is added. This leads to the following series of steps:



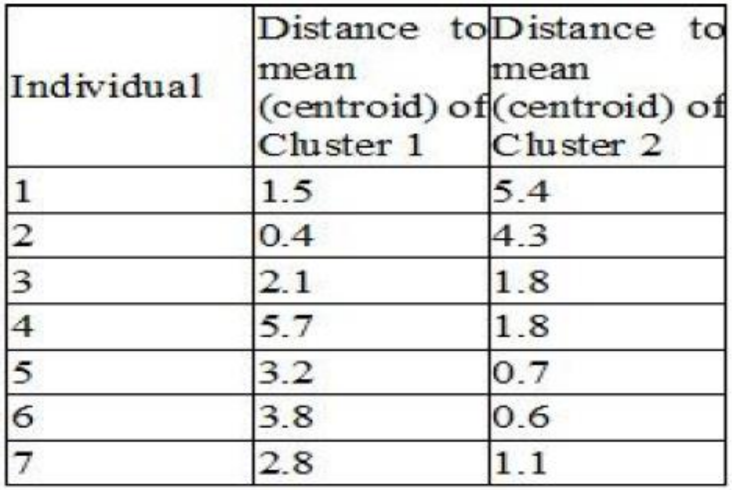
	Cluster 1		Cluster 2	
Step	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

- Now the initial partition has changed, and the two clusters at this stage having the following characteristics:

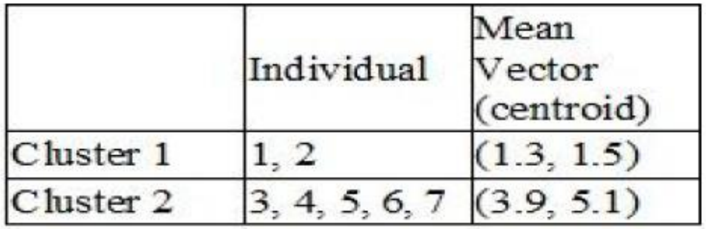


	Individual	Mean Vector (centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)
Cluster 2	4, 5, 6, 7	(4.1, 5.4)

- But we cannot yet be sure that each individual has been assigned to the right cluster. So, we compare each individual’s distance to its own cluster mean and to that of the opposite cluster. And we find:

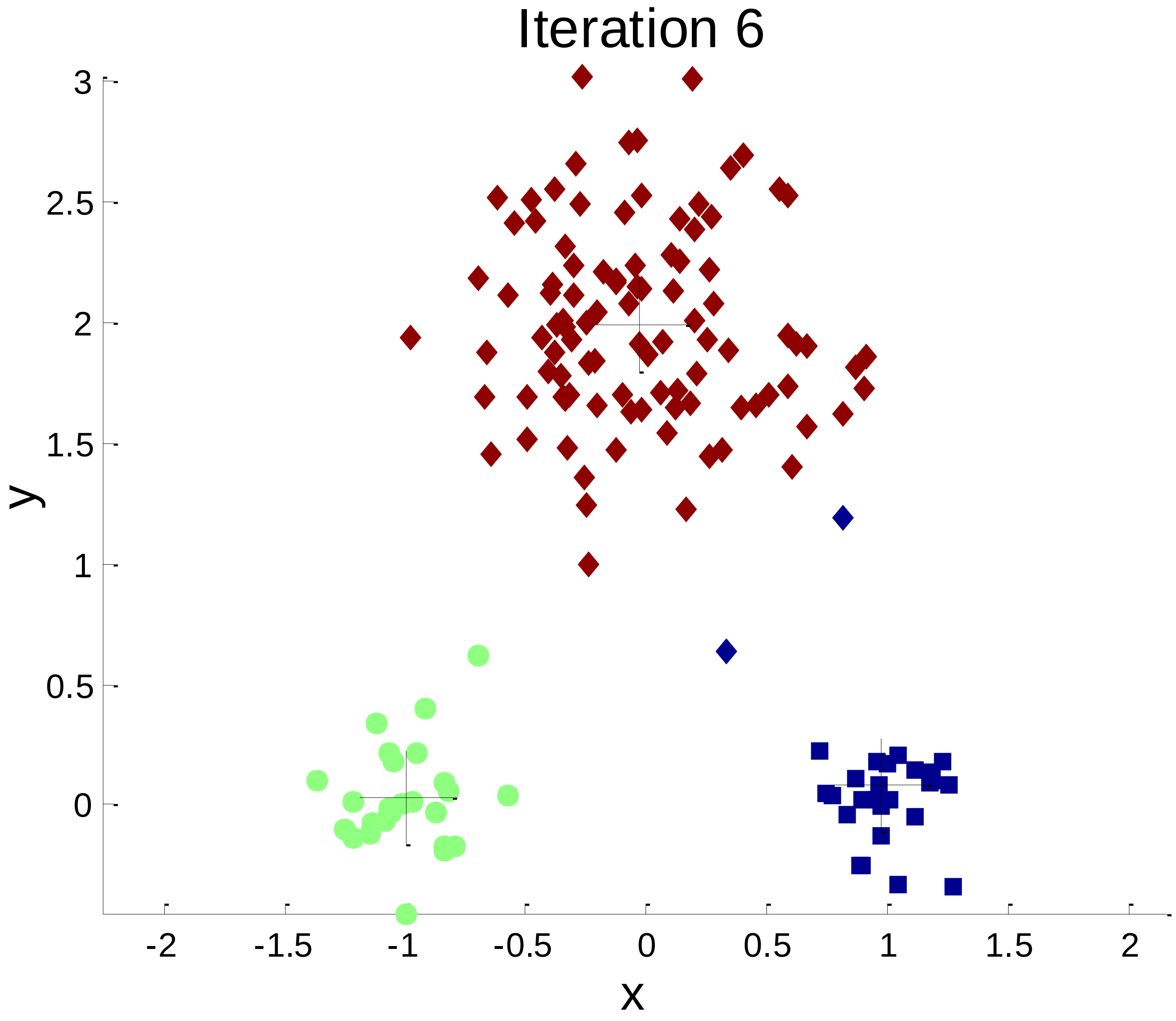


Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

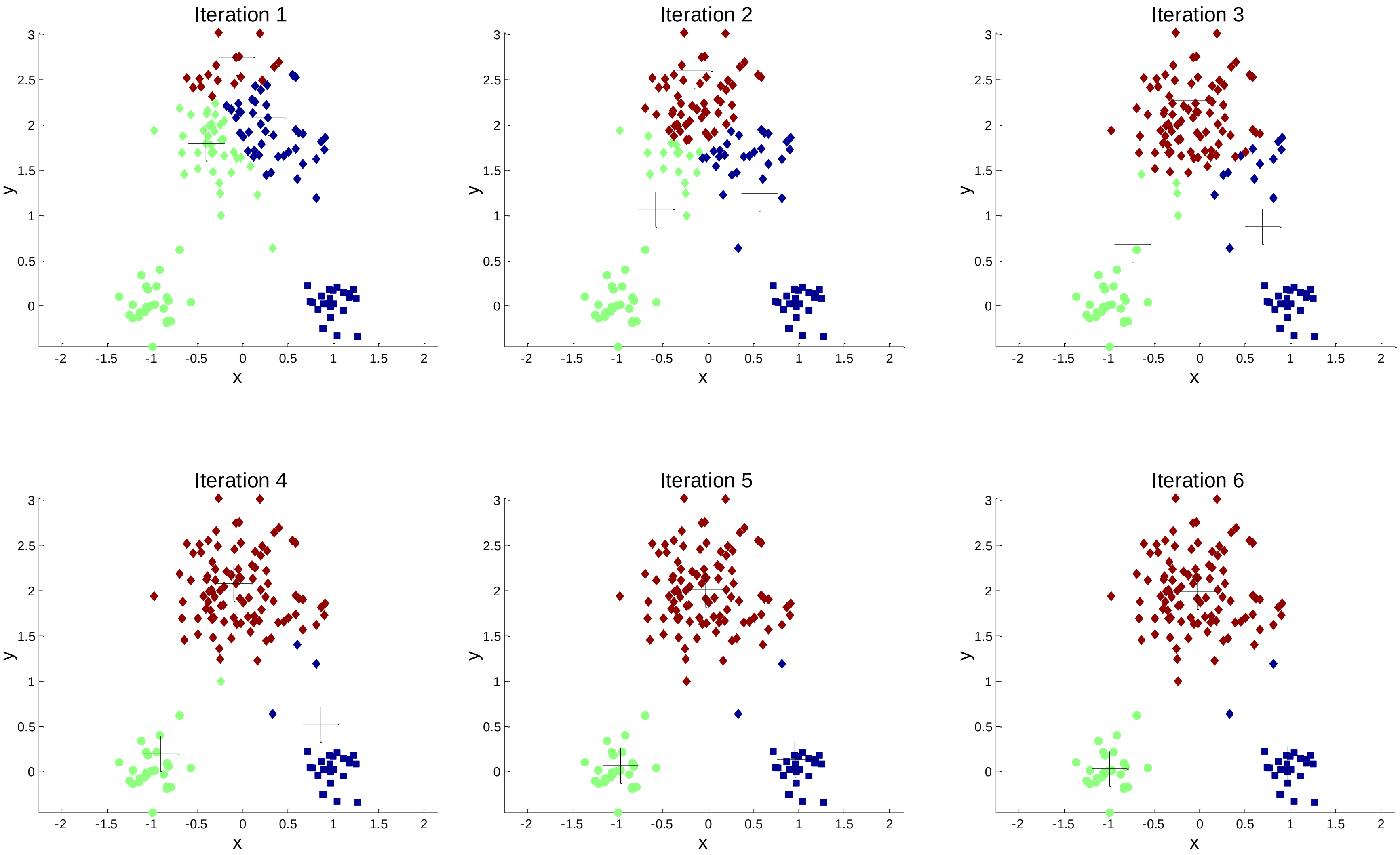


	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

Example of K-means Clustering



Example of K-means Clustering



K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, 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’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
    I = number of iterations, d = number of attributes

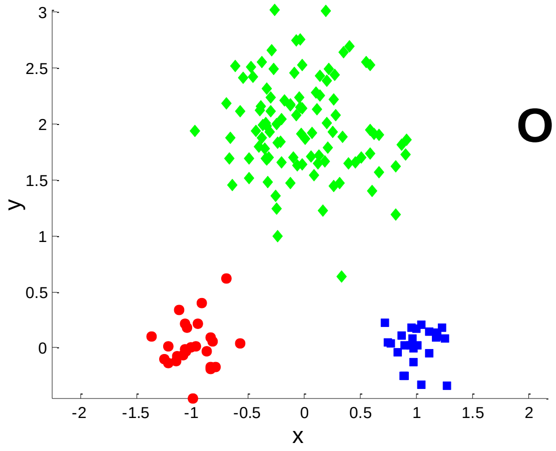
Evaluating K-means Clusters

- 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.

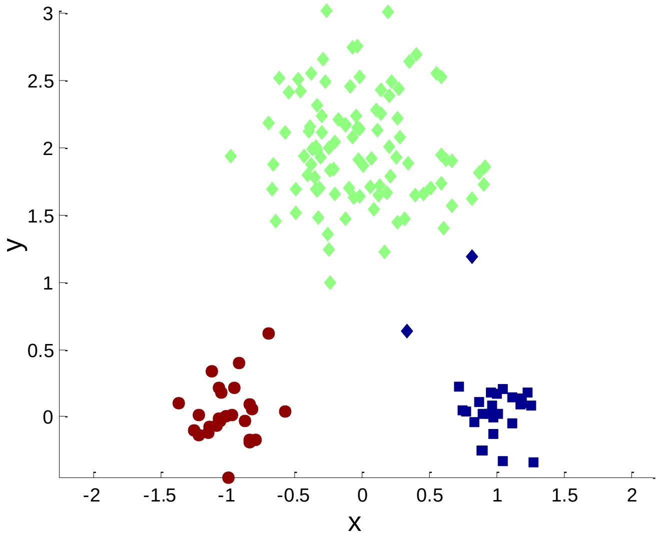
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

  - x is a data point in cluster C_i and m_i is the representative point for cluster C_i
    - can show that m_i corresponds to the center (mean) of the cluster
  - Given two sets of clusters, we prefer the one with the smallest error
  - One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

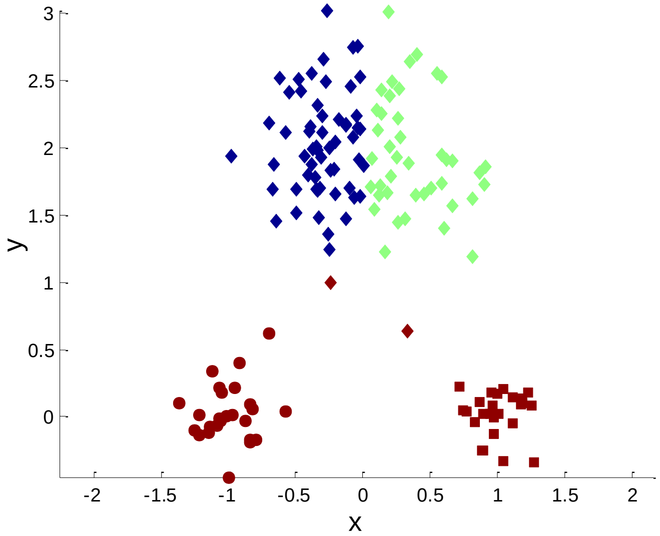
Two different K-means Clusterings



Original Points



Optimal Clustering



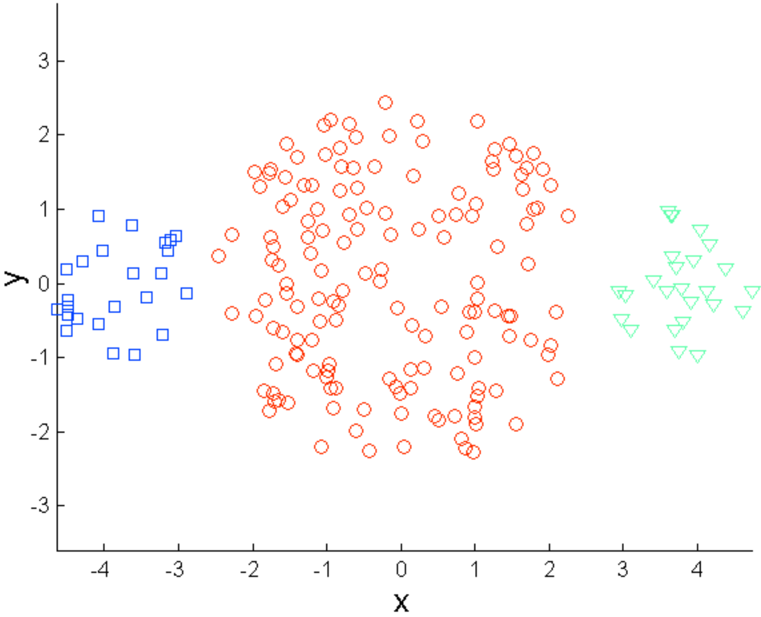
Sub-optimal Clustering

Limitations of K-means

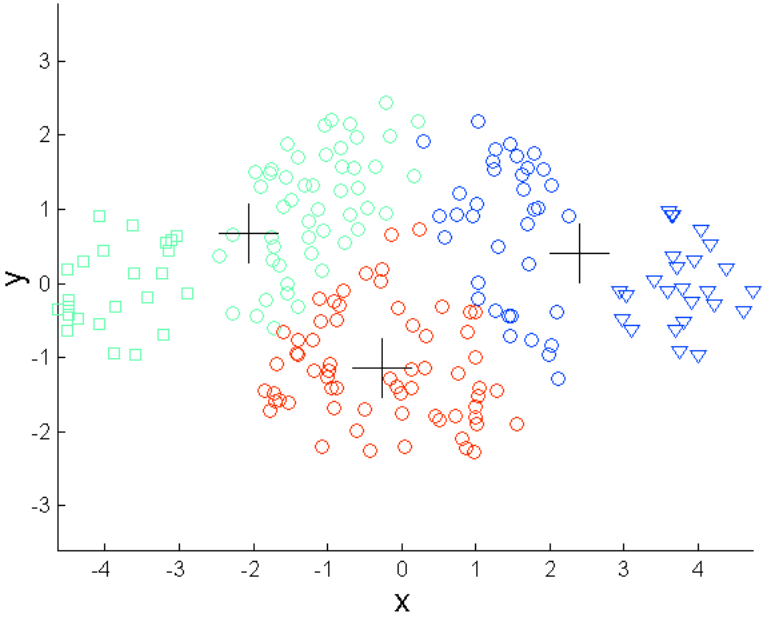
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

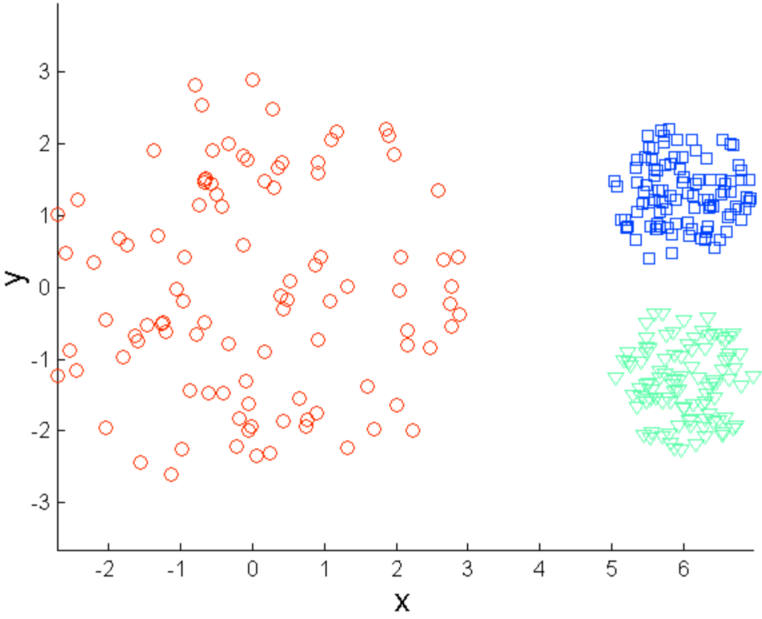


Original Points

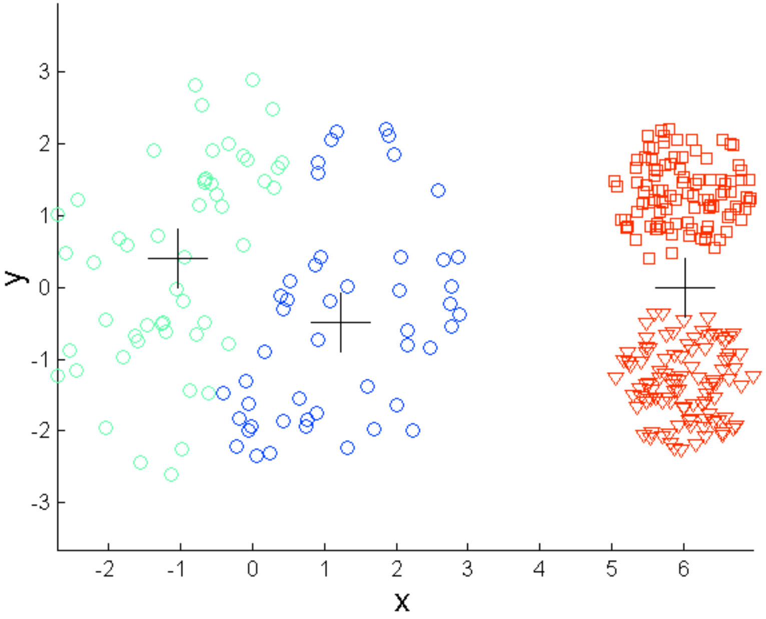


K-means (3 Clusters)

Limitations of K-means: Differing Density

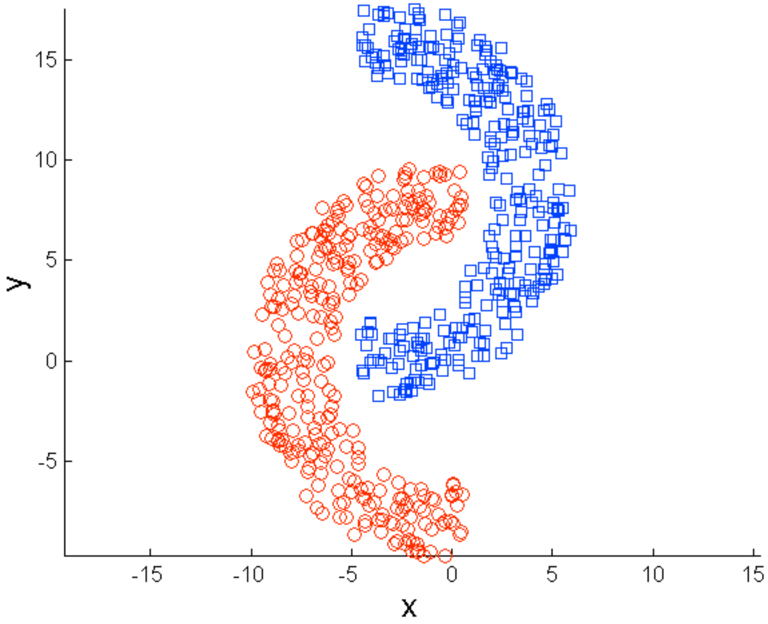


Original Points

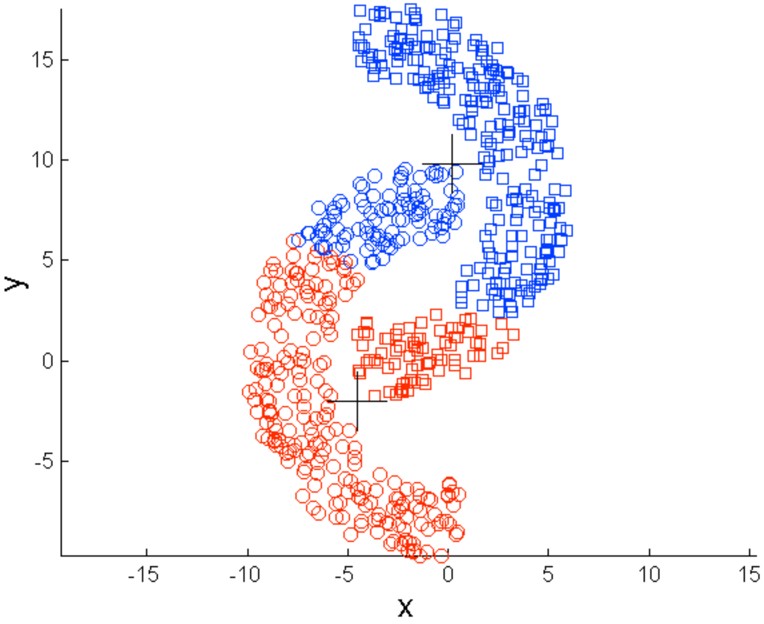


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

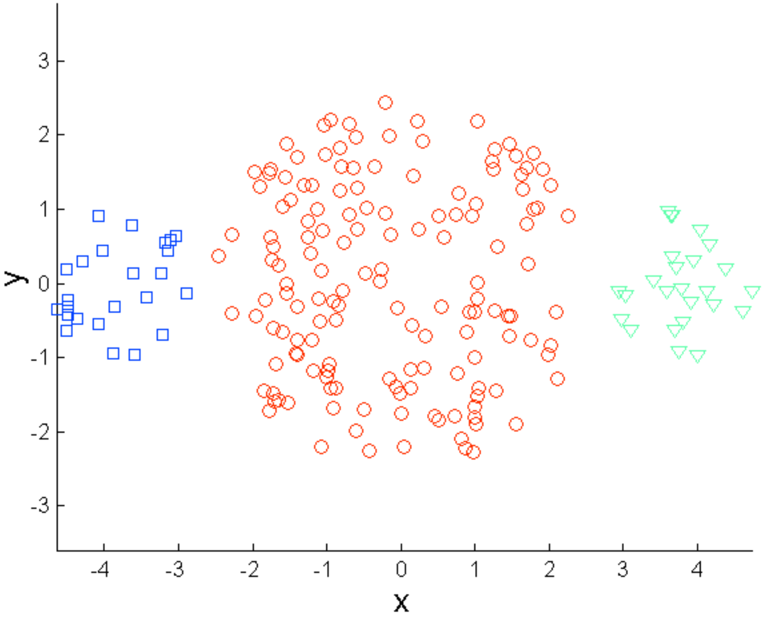


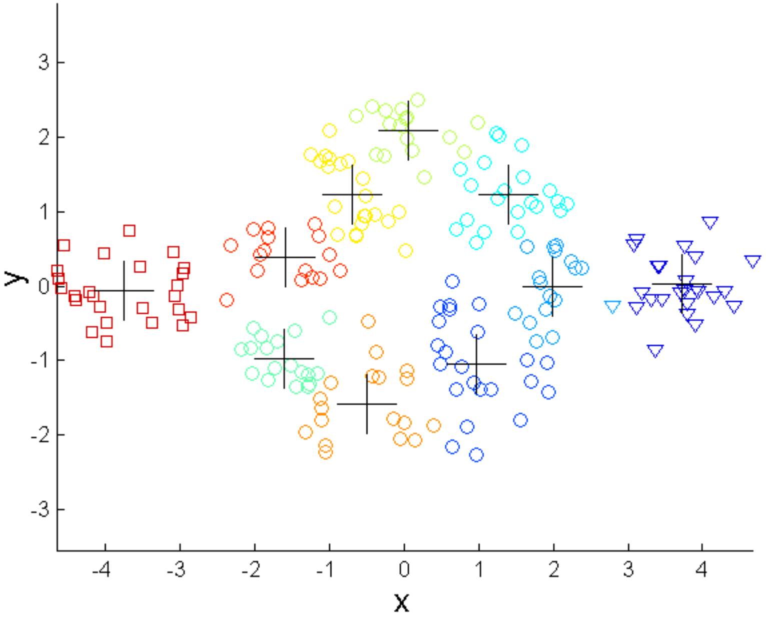
Original Points



K-means (2 Clusters)

Overcoming K-means Limitations





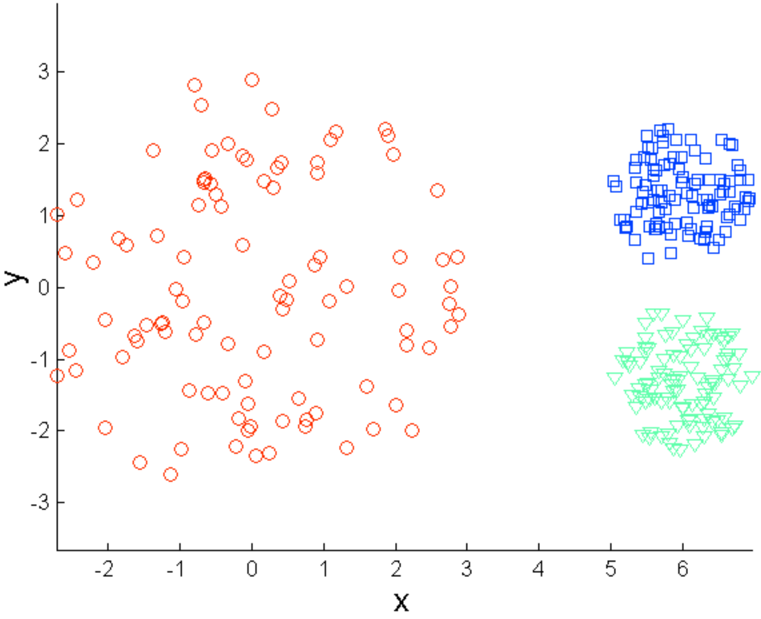
Original Points

K-means Clusters

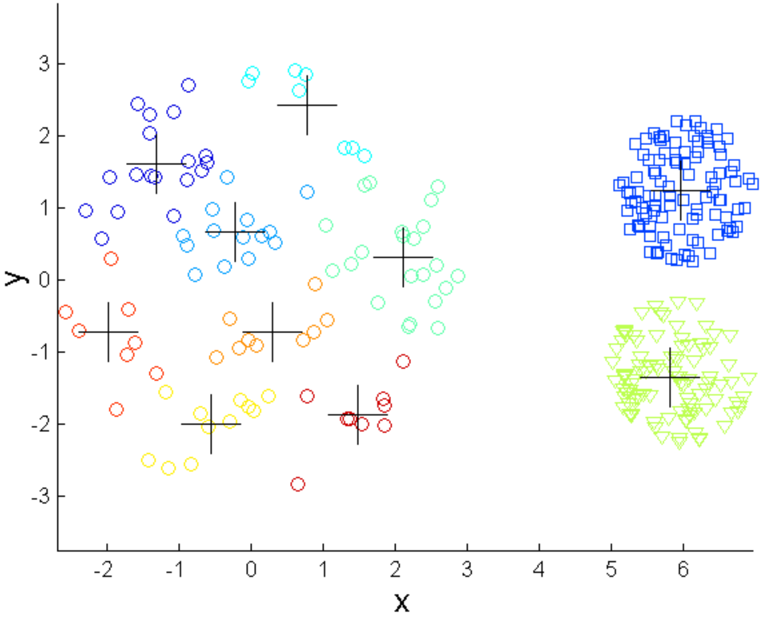
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

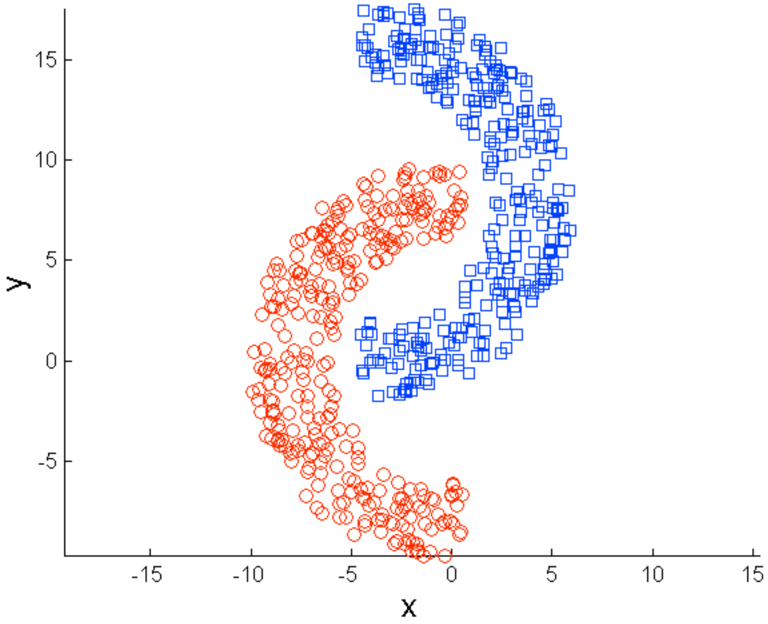


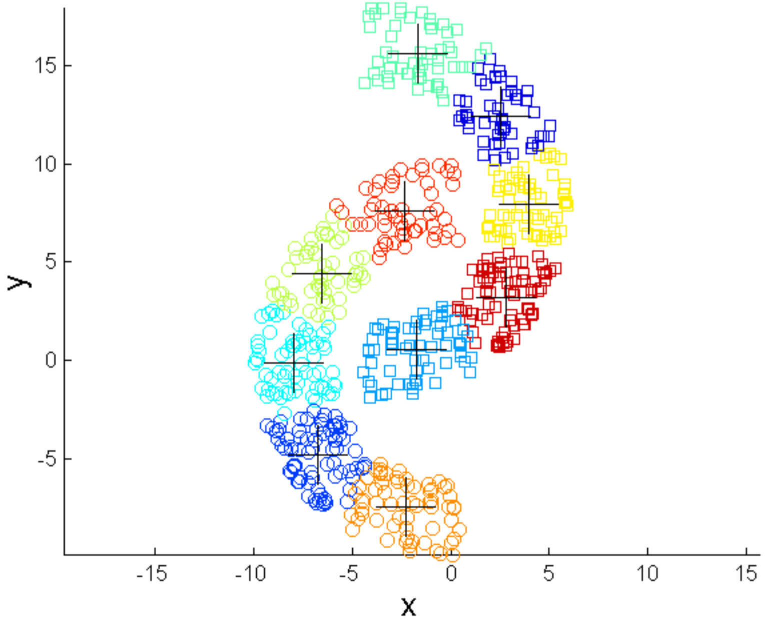
Original Points



K-means Clusters

Overcoming K-means Limitations

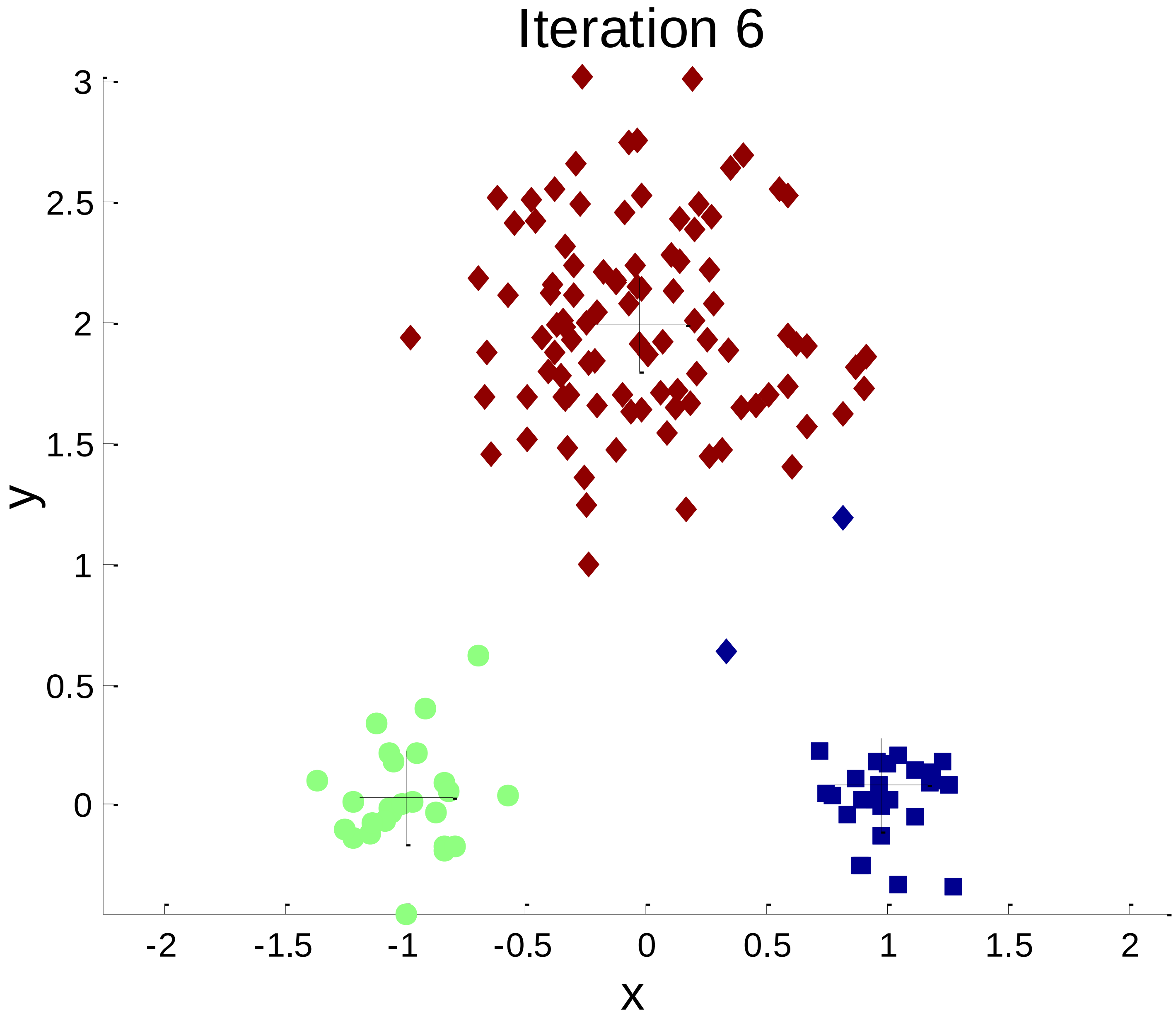




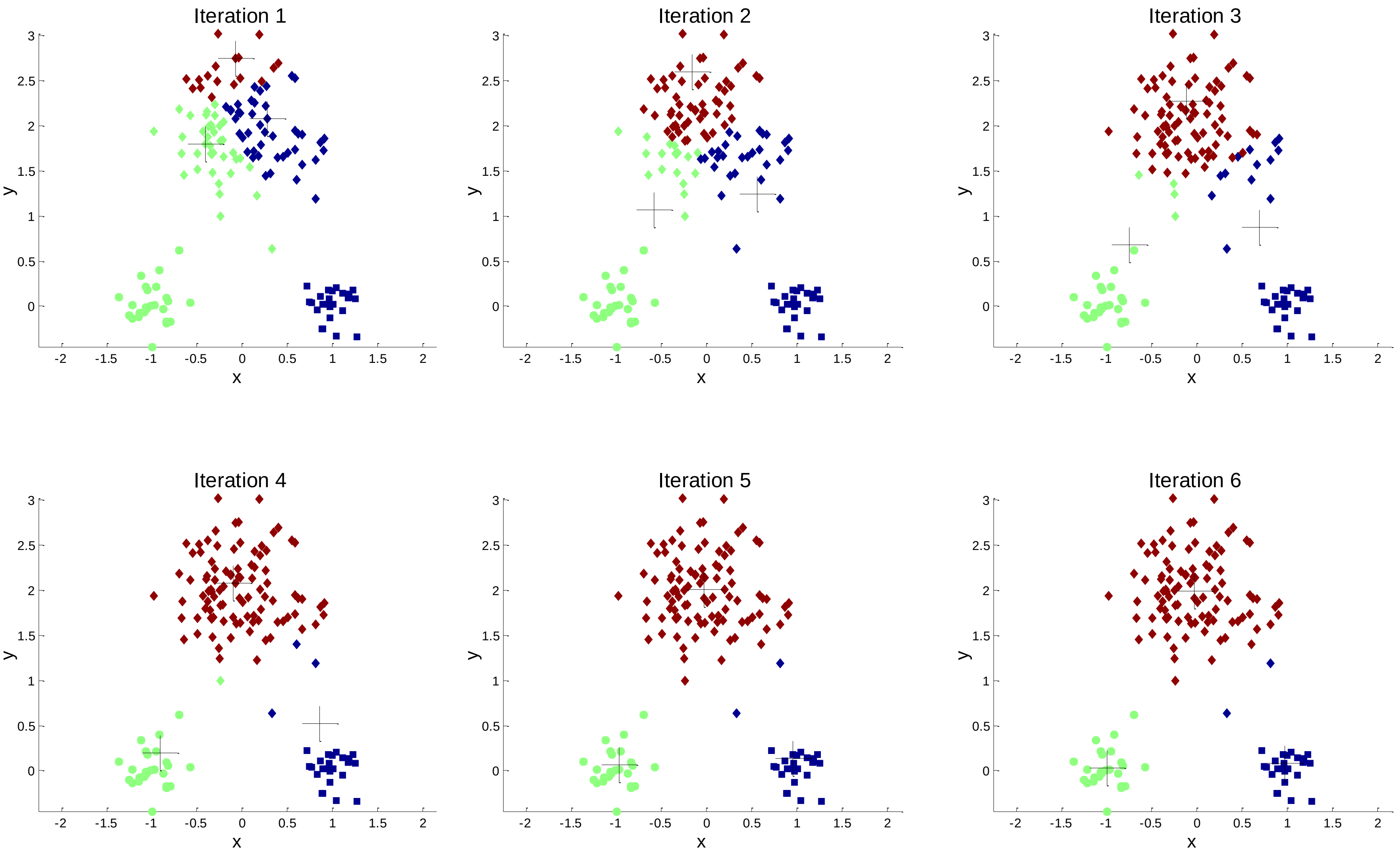
Original Points

K-means Clusters

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



Iteration 1

Iteration 2

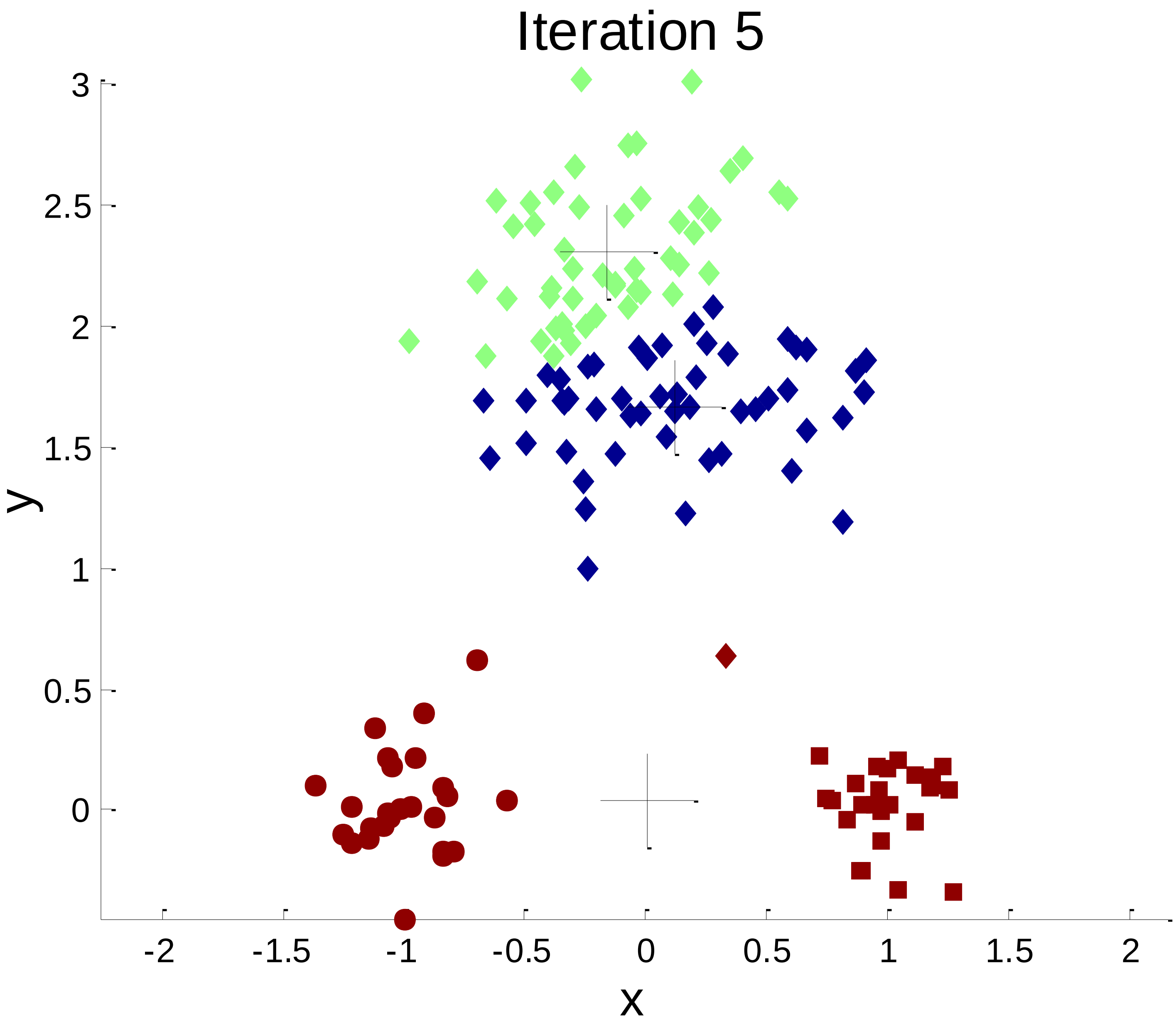
Iteration 3

Iteration 4

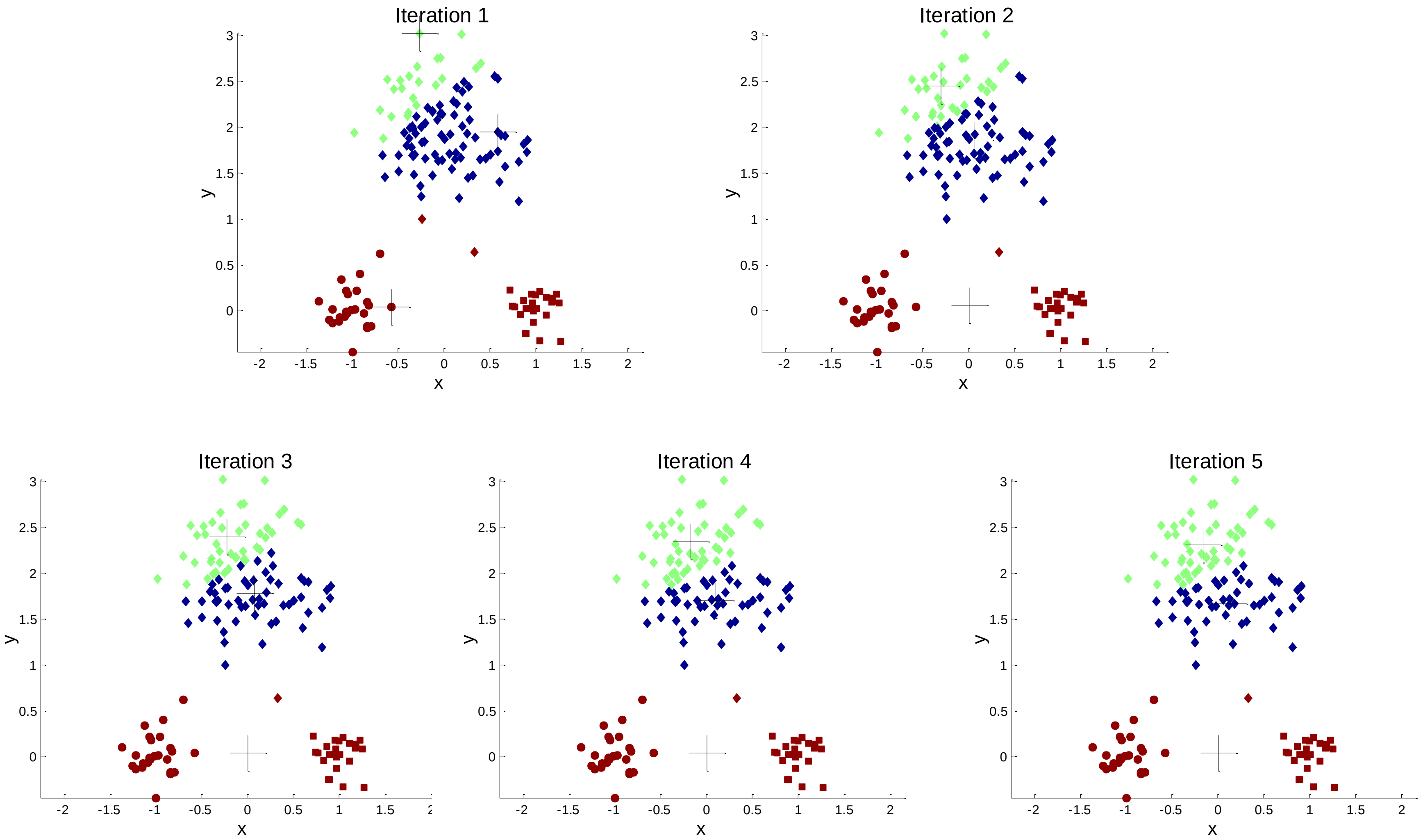
Iteration 5

Iteration 6

Importance of Choosing Initial Centroids …



Importance of Choosing Initial Centroids …



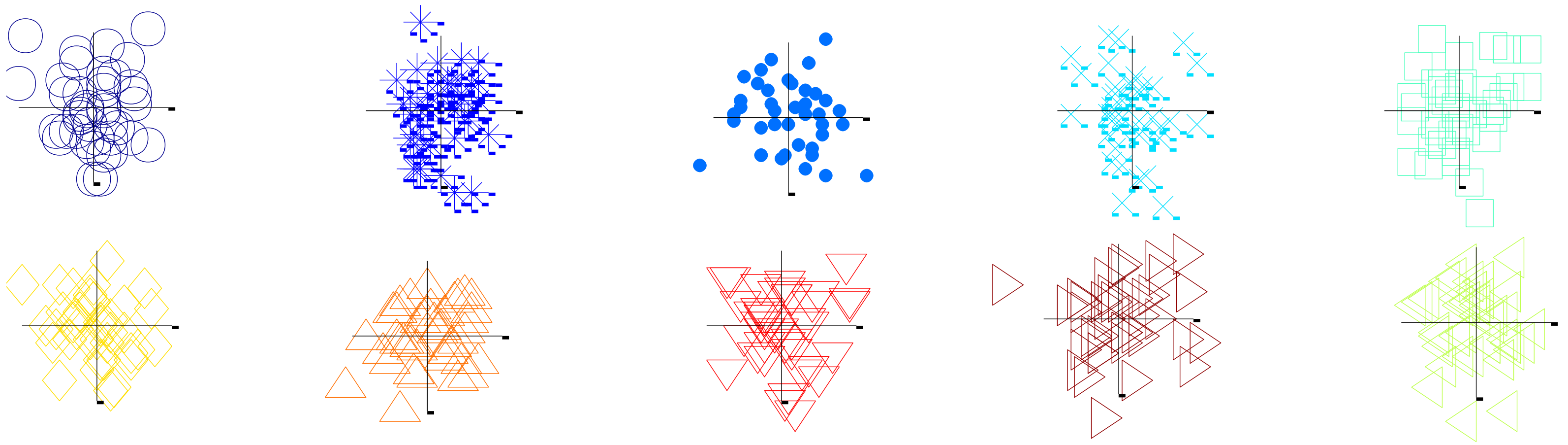
Problems with Selecting Initial Points

- If there are K ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

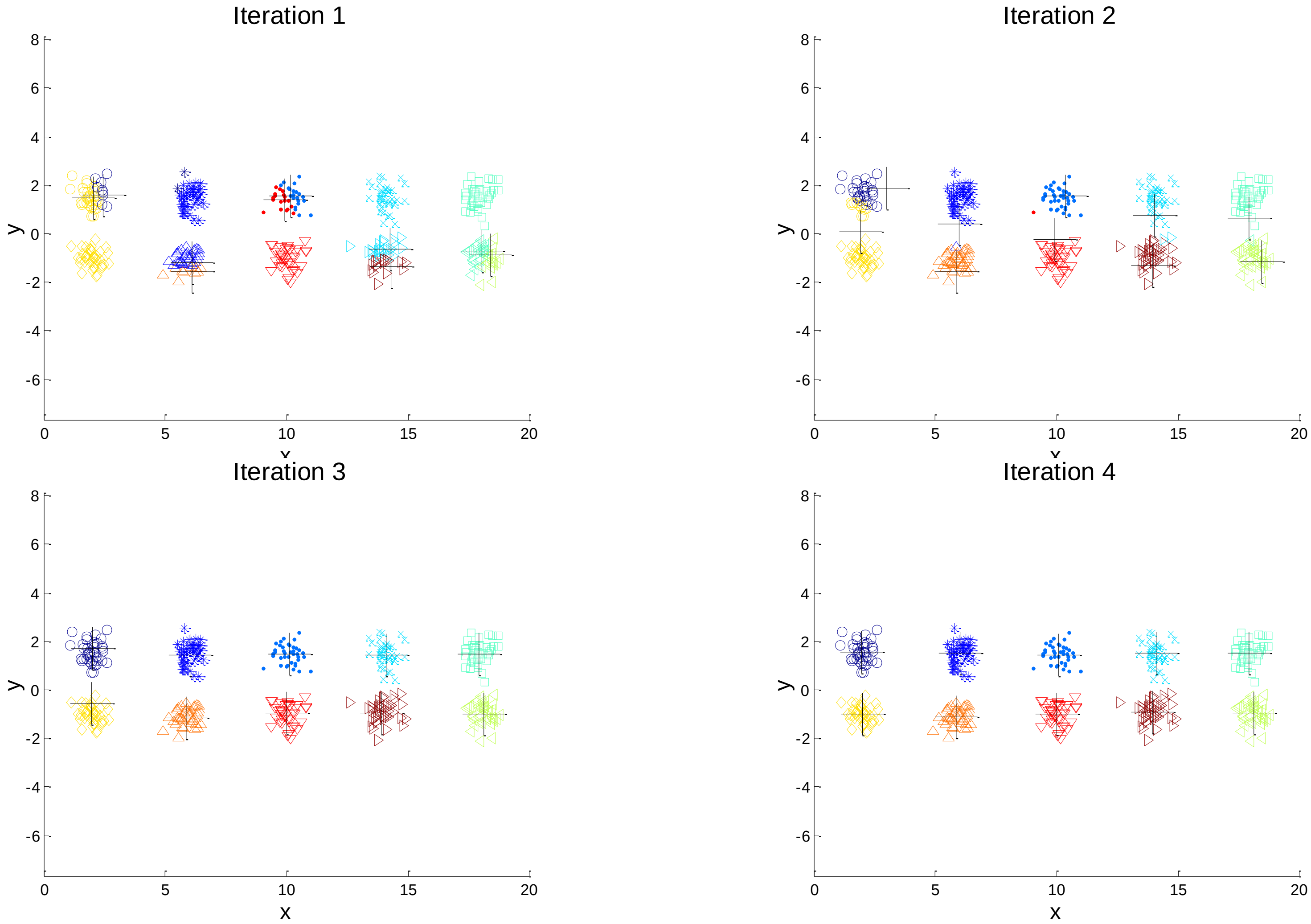
  - For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
  - Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t
  - Consider an example of five pairs of clusters

10 Clusters Example



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

10 Clusters Example


Iteration 1

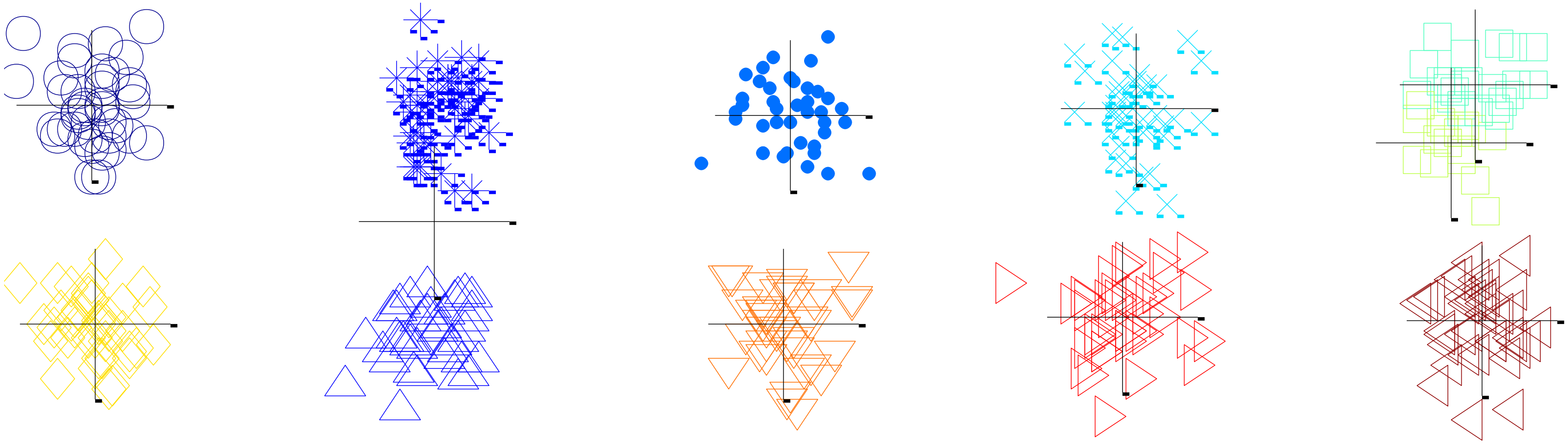
Iteration 2

Iteration 3

Iteration 4

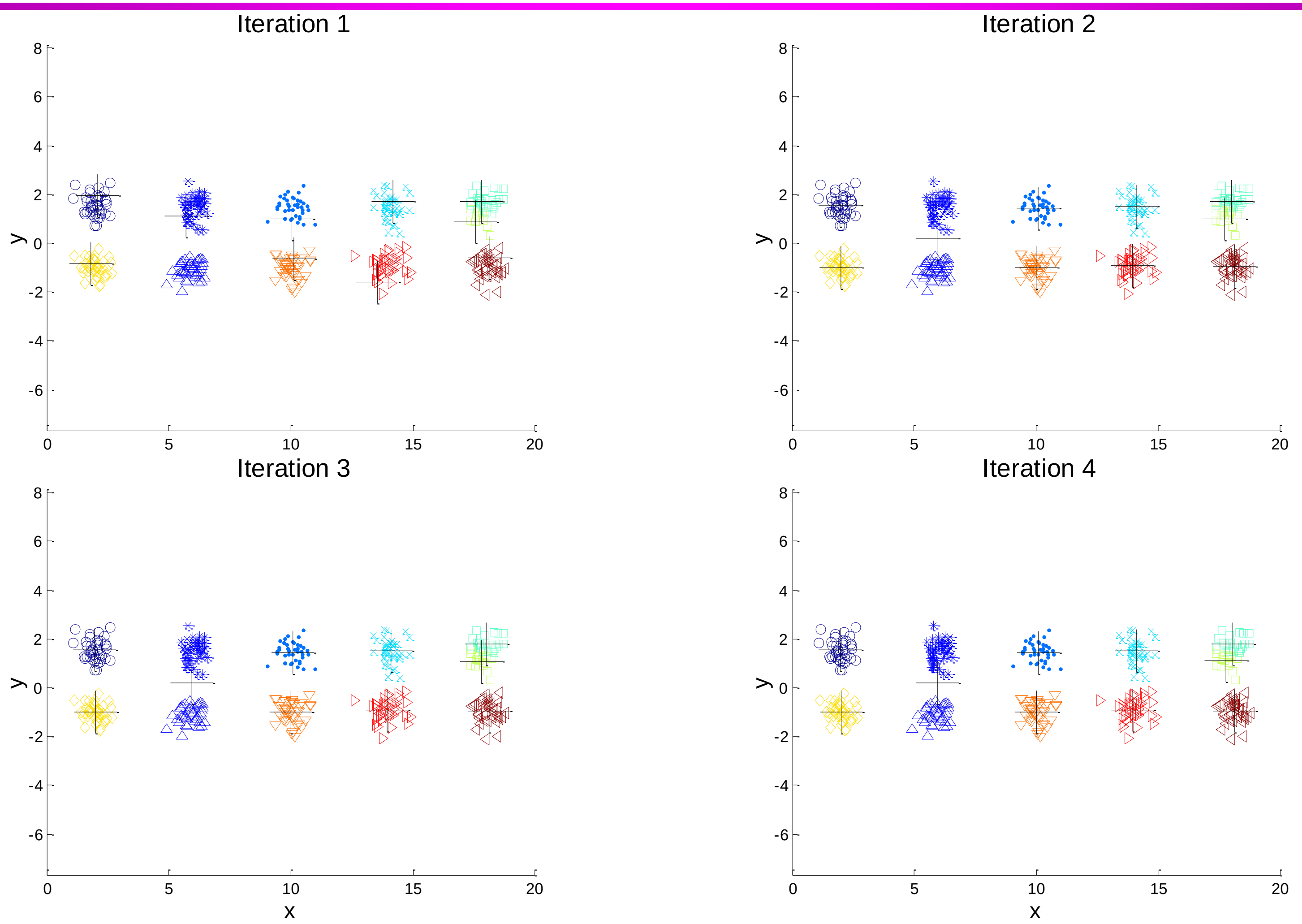
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



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

10 Clusters Example



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

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on your side
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated
- Postprocessing
- Generate a larger number of clusters and then perform a hierarchical clustering
- Bisecting K-means
  - Not as susceptible to initialization issues

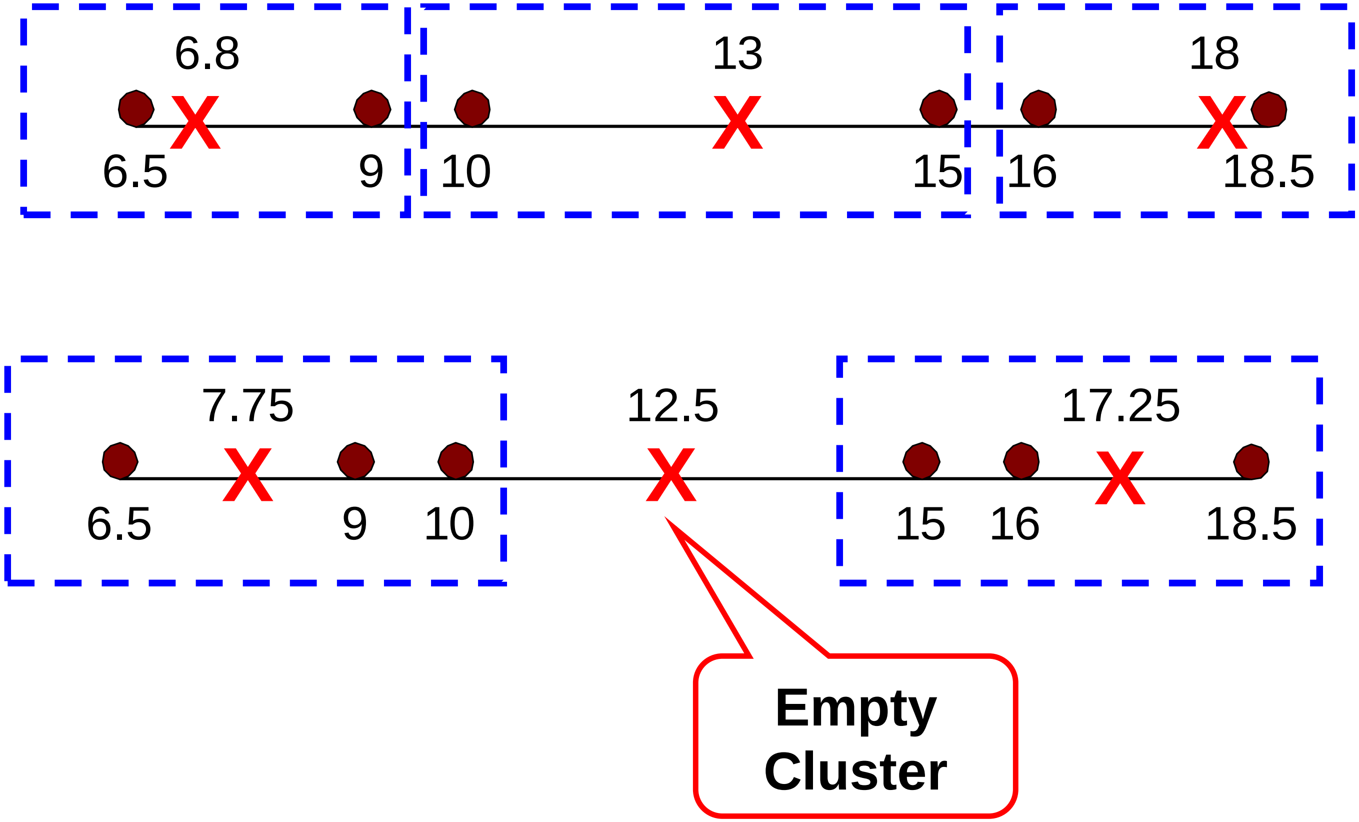
K-means++

- This approach can be slower than random initialization, but very consistently produces better results in terms of SSE
  - The k-means++ algorithm guarantees an approximation ratio O(log k) in expectation, where k is the number of centers
- To select a set of initial centroids, *C*, perform the following

1. Select an initial point at random to be the first centroid
2. For k – 1 steps
3. For each of the N points, x_i, $1 \leq i \leq N$, find the minimum squared distance to the currently selected centroids, $C_1, \ldots, C_j$, $1 \leq j < k$, i.e., $\min_j d^2(C_j, x_i)$
4. Randomly select a new centroid by choosing a point with probability proportional to $\frac{\min_j d^2(C_j, x_i)}{\sum_i \min_j d^2(C_j, x_i)}$ is
5. End For

Empty Clusters

- K-means can yield empty clusters



Handling Empty Clusters

- Basic K-means algorithm can yield empty clusters

- Several strategies
  - Choose the point that contributes most to SSE
  - Choose a point from the cluster with the highest SSE
  - 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 get an empty cluster
  - Can use “weights” to change the impact

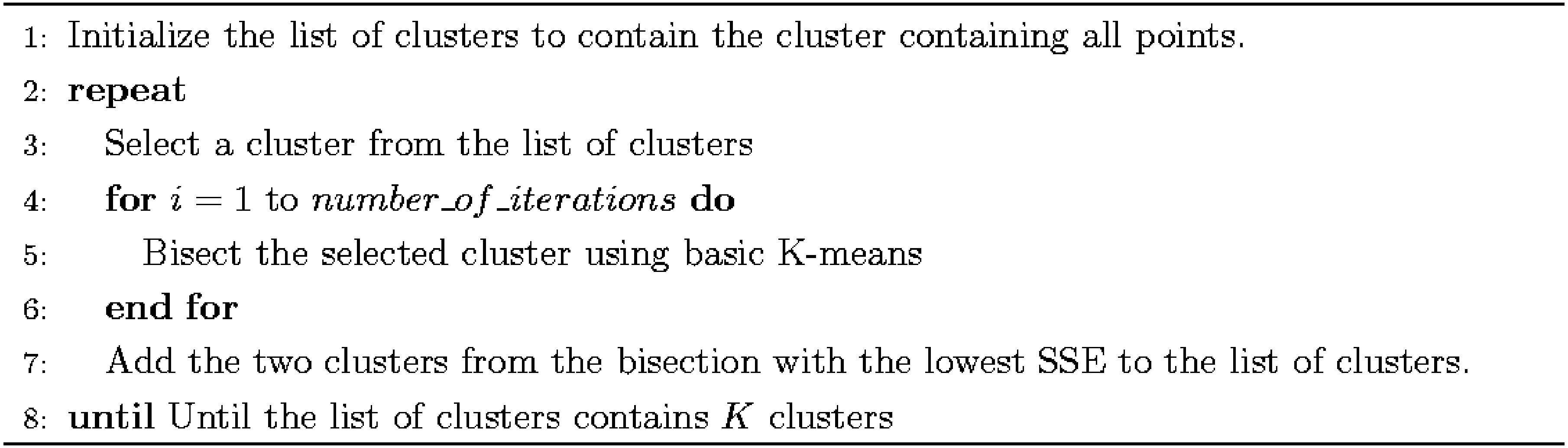
Pre-processing and Post-processing

- 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
    - ISODATA

Bisecting K-means

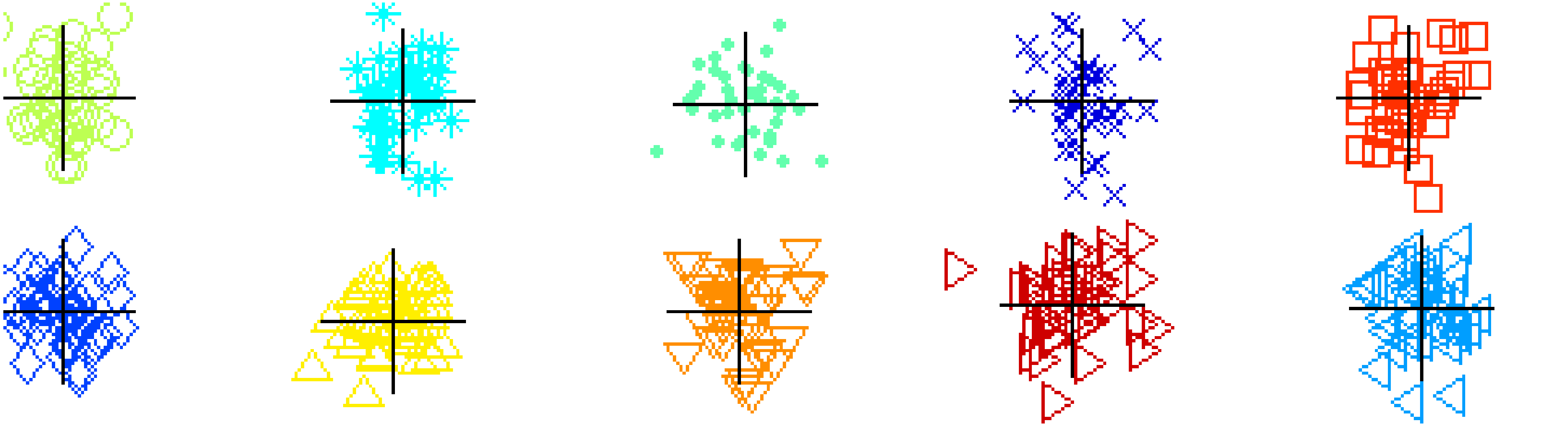
- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitional or a hierarchical clustering



```
Initialize the list of clusters to contain the cluster containing all points.
repeat
    Select a cluster from the list of clusters
    for $i = 1$ to $number\_of\_iterations$ do
        Bisect the selected cluster using basic K-means
    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
```

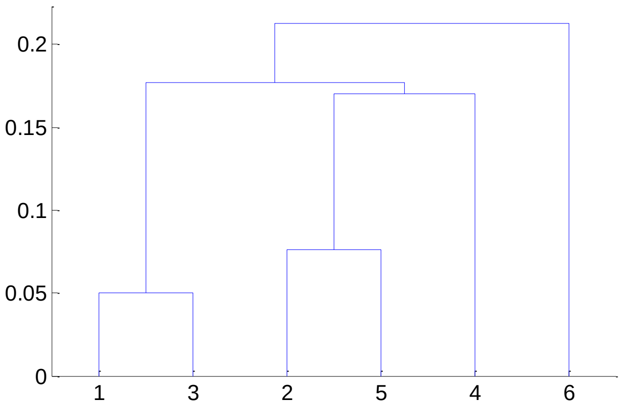
CLUTO: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

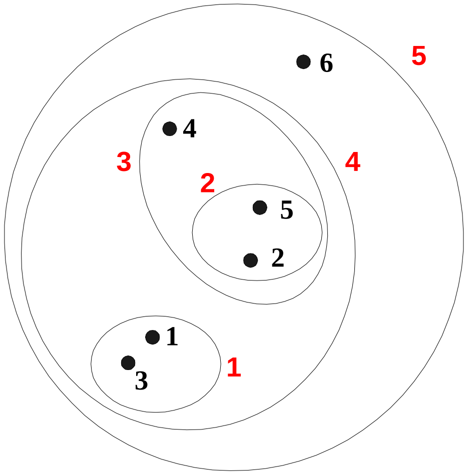
Bisecting K-means Example



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





Hierarchical Clustering

- Hierarchical clustering is set of methods that recursively cluster two items at a time.
- There are basically two different types of algorithms,
  - agglomerative
  - partitioning.
- In partitioning algorithms, the entire set of items starts in a cluster which is partitioned into two more homogeneous clusters. Then the algorithm restarts with each of the new clusters, partitioning each into more homogeneous clusters until each cluster contains only identical items (possibly only 1 item). If there is time towards the end of the course we may discuss partitioning algorithms.

Hierarchical Clustering

- In agglomerative algorithms, each item starts in its own cluster and the two most similar items are then clustered. You continue accumulating the most similiar items or clusters together two at a time until there is one cluster. For both types of algorithms, the clusters at each step can be displayed in a dendrogram.

Agglomerative Process

- Choose a distance function for items d(xi,xj)
- Choose a distance function for clusters D(Ci,Cj)
  - for clusters formed by just one point, *D* should reduce to *d*.
- Start from N clusters, each containing one item. Then, at each iteration:
  - a) using the current matrix of cluster distances, find two closest clusters.
  - b) update the list of clusters by merging the two closest.
  - c) update the matrix of cluster distances accordingly
- Repeat until all items are joined in one cluster.

Distance Measures

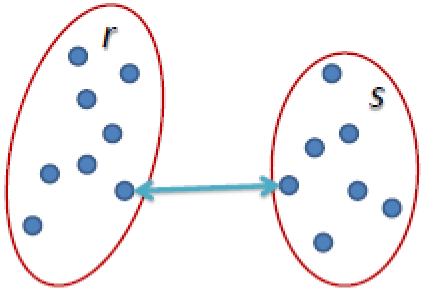
'euclidean':	Usual square distance between the two vectors (2 norm).
'maximum':	Maximum distance between two components of *x* and *y* (supremum norm)
'manhattan':	Absolute distance between the two vectors (1 norm).
'canberra':	$\sum(\lvert xi-yi \rvert / \lvert xi+yi \rvert)\sum(\lvert xi-yi \rvert / \lvert xi+yi \rvert)$. Terms with zero numerator and denominator are omitted from the sum and treated as if the values were missing.
'minkowski':	The *p* norm, the pth root of the sum of the *p*th powers of the differences of the components.
'correlation':	1 - *r* where *r* is the Pearson or Spearman correlation
'absolute correlation':	$1 - \lvert r \rvert$

Defining Cluster Distance: The Linkage Function

- So far we have defined a distance between items. The linkage function tells you to measure the distance between clusters. Again, there are many choices.
- Typically you consider either a new item that summarizes the items in the cluster, or a new distance that summarizes the distance between the items in the cluster and items in other clusters. Here is a list of three methods. In each example, x is in one cluster and y is in the other.

Single

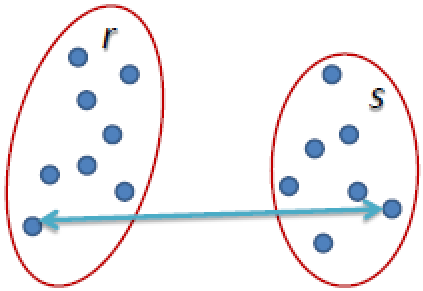
- In single linkage hierarchical clustering, the distance between two clusters is defined as the *shortest* distance between two points in each cluster. For example, the distance between clusters “r” and “s” to the left is equal to the length of the arrow between their two closest points



f=min(d(x,y))

Complete

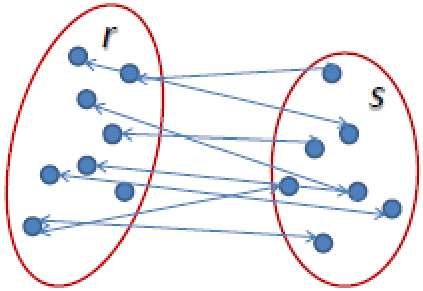
- In complete linkage hierarchical clustering, the distance between two clusters is defined as the *longest* distance between two points in each cluster. For example, the distance between clusters “r” and “s” to the left is equal to the length of the arrow between their two furthest points.



f=max(d(x,y))

Average

- In average linkage hierarchical clustering, the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster. For example, the distance between clusters “r” and “s” to the left is equal to the average length each arrow between connecting the points of one cluster to the other.



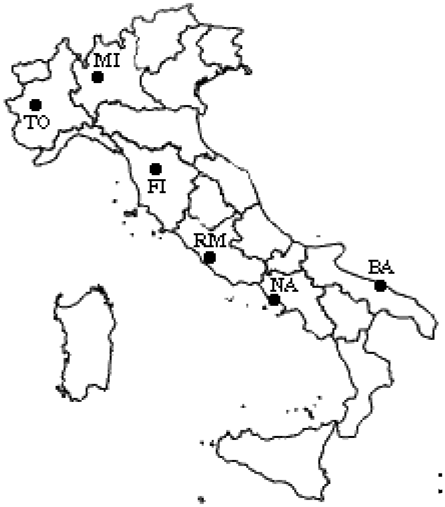
f=average(d(x,y))

An Example

- Let’s now see a simple example: a hierarchical clustering of distances in kilometers between some Italian cities. The method used is single-linkage.

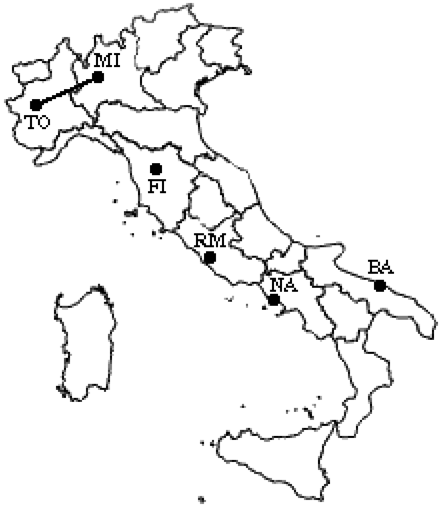
- **Input distance matrix** (L = 0 for all the clusters):

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0



	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0

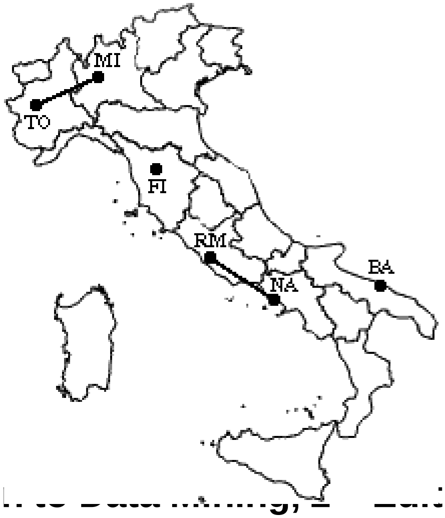
The nearest pair of cities is MI and TO, at distance 138. These are merged into a single cluster called "MI/TO". The level of the new cluster is D(MI/TO) = 138



	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0

min d(i,j) = d(NA,RM) = 219 => merge NA and RM into a new cluster called NA/RM

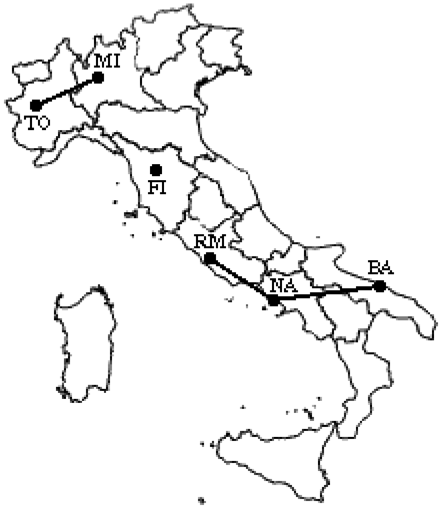
D(NA/RM) = 219



	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0

min d(i,j) = d(BA,NA/RM) = 255 => merge BA and NA/RM into a new cluster called BA/NA/RM

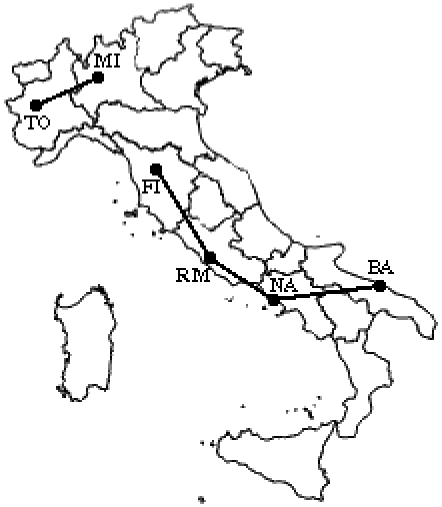
D(BA/NA/RM) = 255



	BA/NA/RM	FI	MI/TO
BA/NA/RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

min d(i,j) = d(BA/NA/RM,FI) = 268 => merge BA/NA/RM and FI into a new cluster called BA/FI/NA/RM

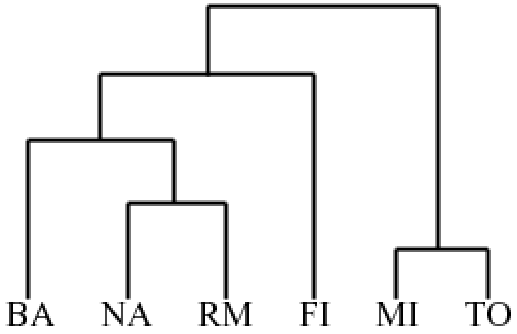
D(BA/FI/NA/RM) = 268



	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0

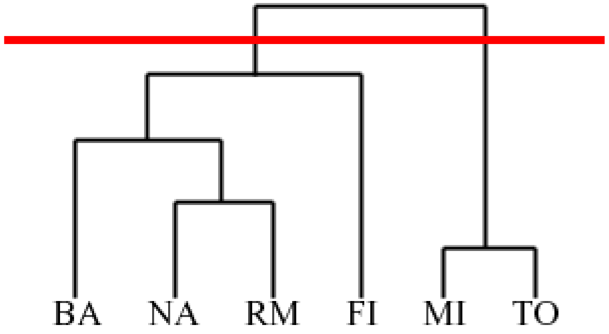
Finally, we merge the last two clusters at level 295.

The process is summarized by the following hierarchical tree:



Final Stage

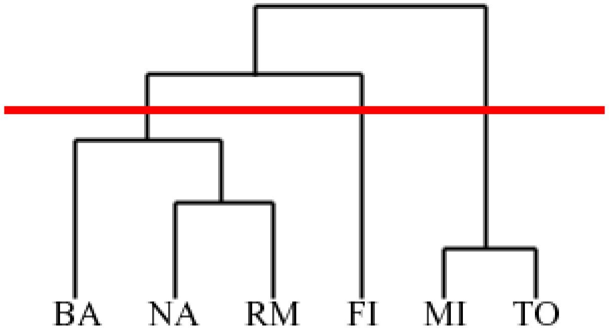
2 cluster



$C_1 = \{BA, NA, RM, FI\}$

$C_2 = \{MI, TO\}$

3 cluster

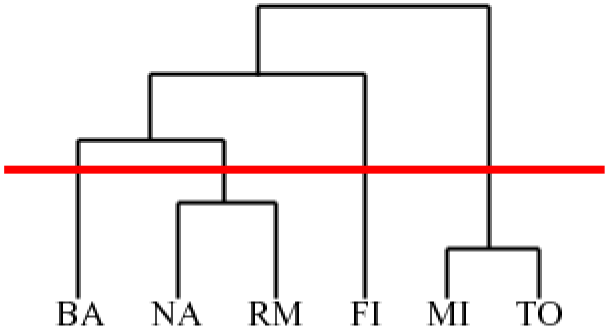


$C_1 = \{BA, NA, RM\}$

$C_2 = \{FI\}$

$C_3 = \{MI, TO\}$

4 cluster



$C_1 = \{BA\}$

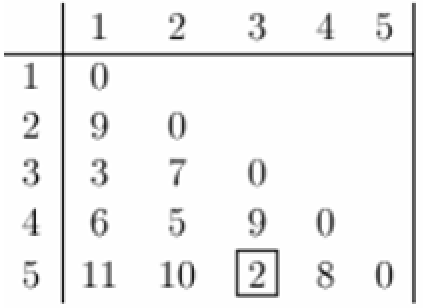
$C_2 = \{NA, RM\}$

$C_3 = \{NA, RM\}$

$C_4 = \{MI, TO\}$

Example 2

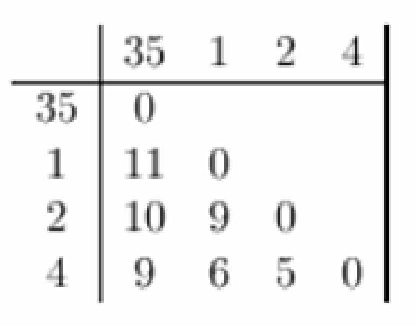
- The table below is an example of a distance matrix. Only the lower triangle is shown, because the upper triangle can be filled in by reflection.



	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	$\boxed{2}$	8	0

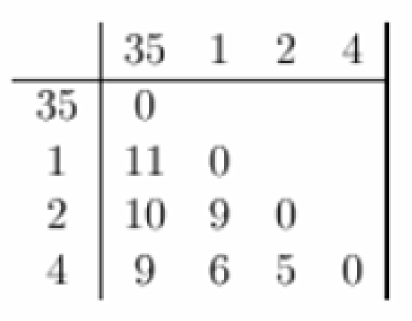
- Now lets start clustering. The smallest distance is between three and five and they get linked up or merged first into a the cluster '35'.

- To obtain the new distance matrix, we need to remove the 3 and 5 entries, and replace it by an entry "35" .
- Since we are using complete linkage clustering, the distance between "35" and every other item is the maximum of the distance between this item and 3 and this item and 5.
- For example, d(1,3)= 3 and d(1,5)=11. So, D(1,"35")=11. This gives us the new distance matrix.



	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

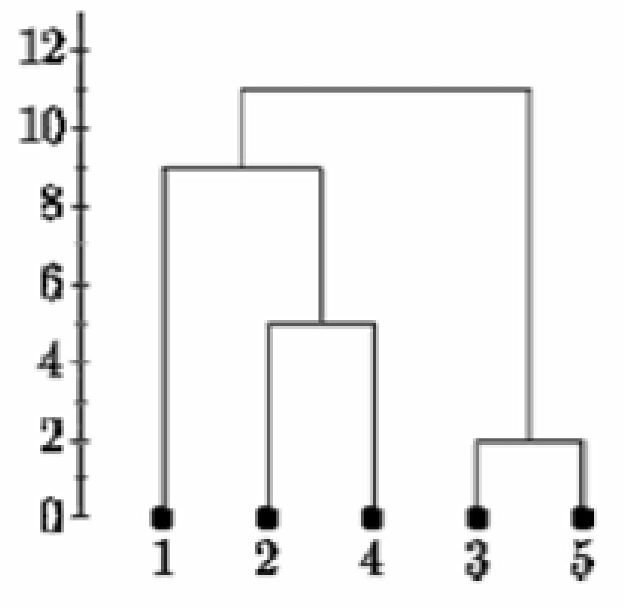
- The items with the smallest distance get clustered next. This will be 2 and 4.



	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

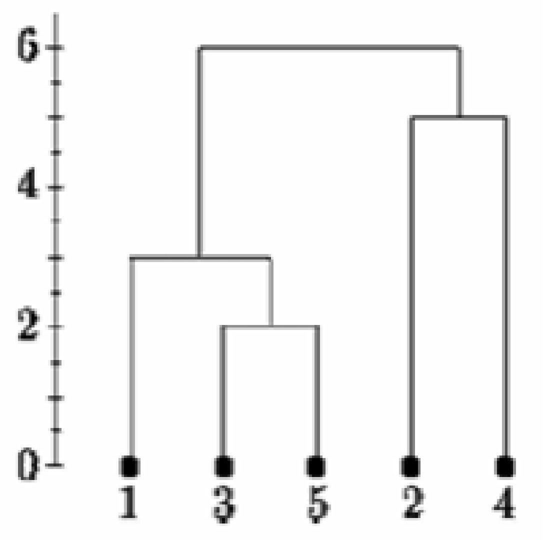
- Continuing in this way, after 6 steps, everything is clustered. This is summarized below. On this plot, the y-axis shows the distance between the objects at the time they were clustered. This is called the cluster height. Different visualizations use different measures of cluster height.

Dendogram



Self Study

- Below is the single linkage dendrogram for the same distance matrix.



- Solve it?
- Also find the dendogram for the average linkage.

Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)