

---

# CS 484

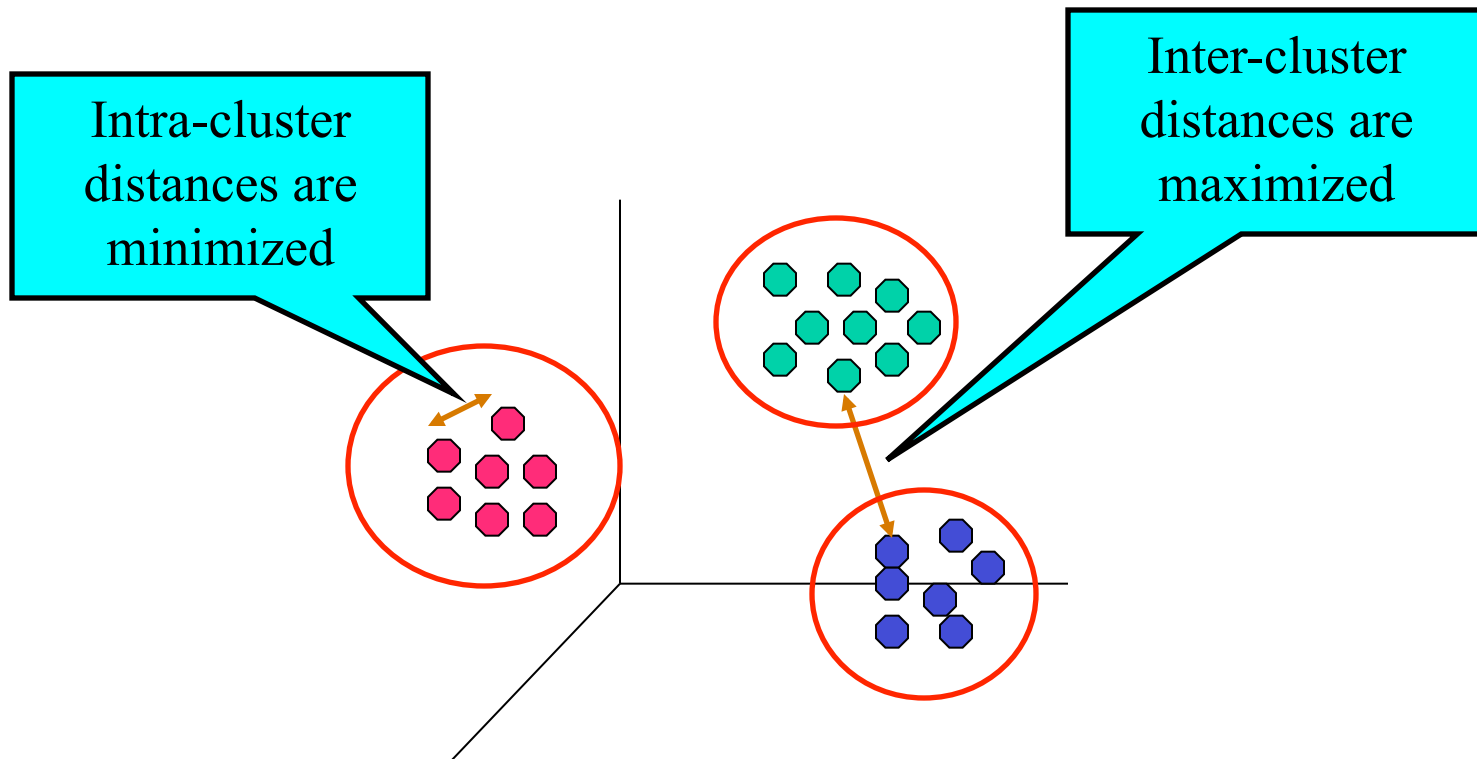
# Data Mining

Clustering 1

---

# 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



# Applications of Cluster Analysis

- **Understanding**

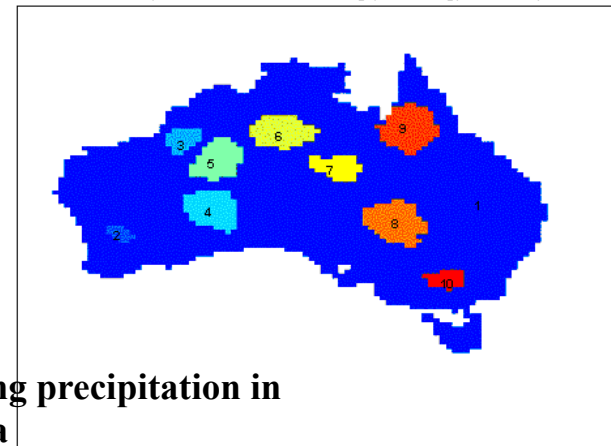
- Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations

	<i>Discovered Clusters</i>	<i>Industry Group</i>
<b>1</b>	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
<b>2</b>	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
<b>3</b>	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
<b>4</b>	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

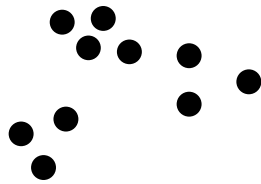
10 Precip Clusters usin SNN Clusterhg (12 mo. avg, NN = 100 )



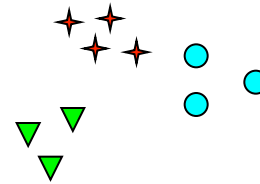
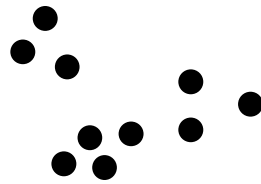
# What is not Cluster Analysis?

- **Supervised classification**
  - Have class label information
- **Simple segmentation**
  - Dividing students into different registration groups alphabetically, by last name
- **Results of a query**
  - Groupings are a result of an external specification

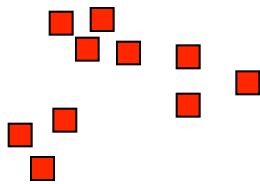
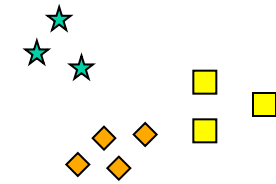
# Notion of a Cluster can be Ambiguous



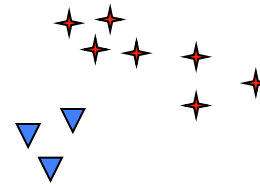
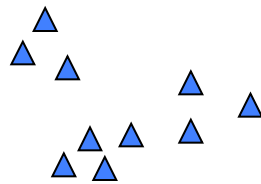
How many clusters?



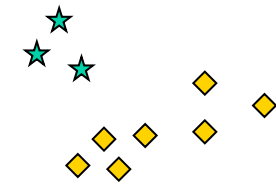
Six Clusters



Two Clusters



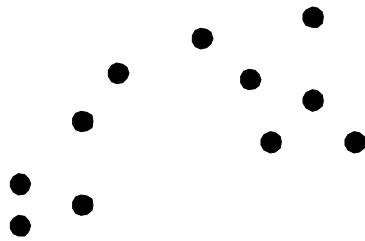
Four Clusters



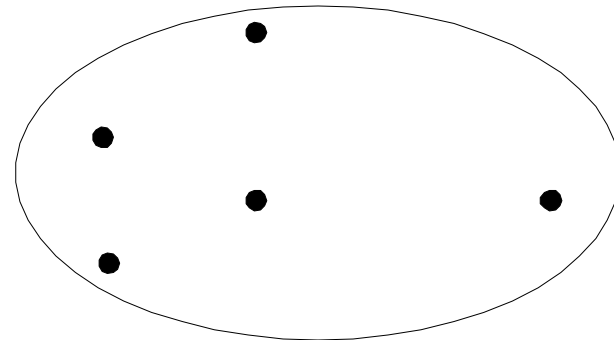
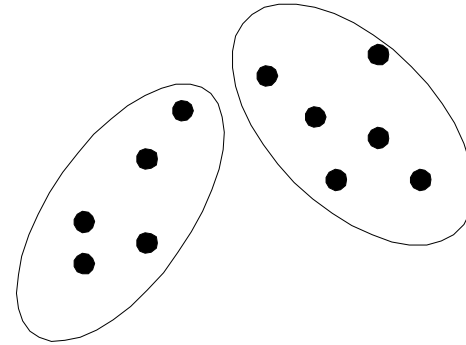
# Popular Types of Clusterings

- 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

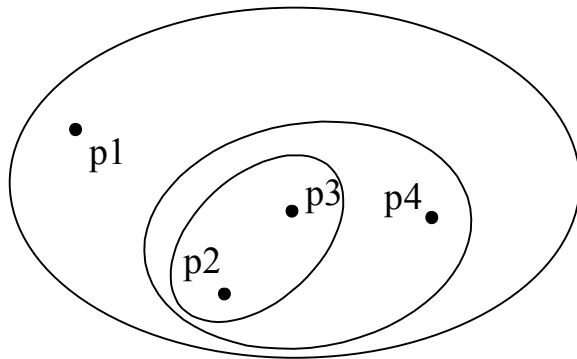


**Original Points**

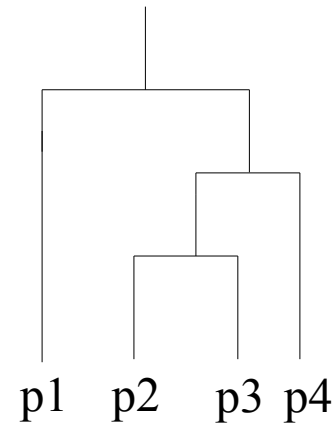


**A Partitional Clustering**

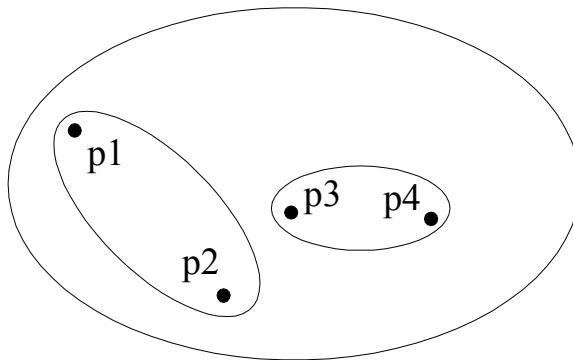
# Hierarchical Clustering



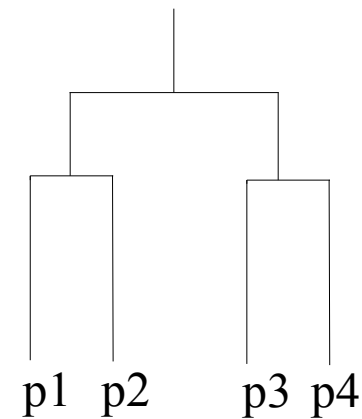
**Traditional Hierarchical Clustering**



**Traditional Dendrogram**



**Non-traditional Hierarchical Clustering**



**Non-traditional Dendrogram**



# Other Distinctions Between Sets of Clusters

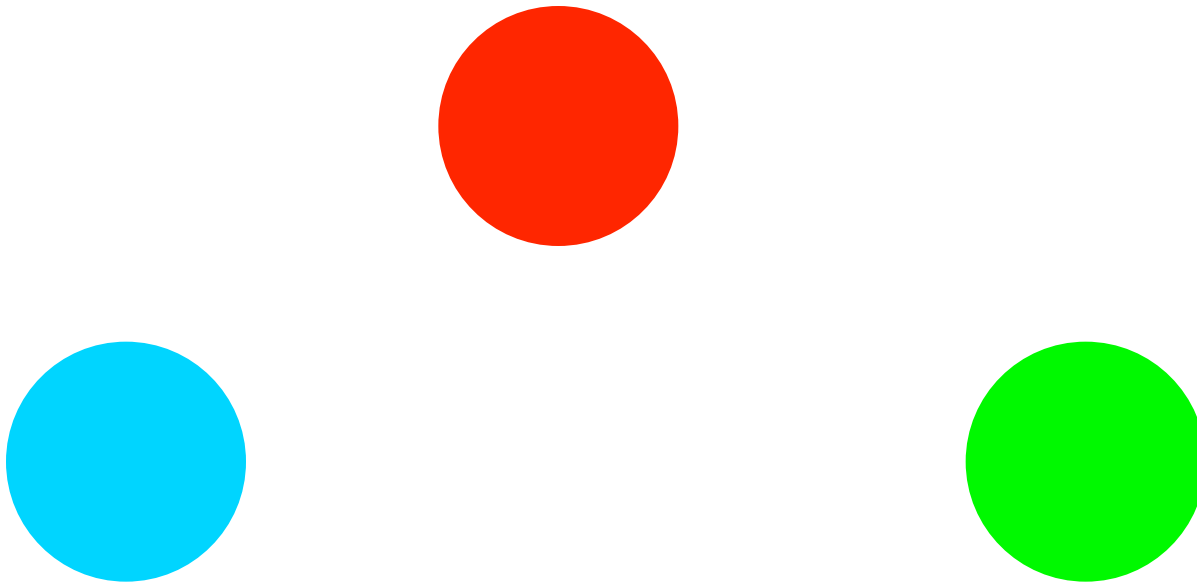
- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - Probabilistic clustering has similar characteristics
- Partial versus complete
  - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
  - Cluster 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

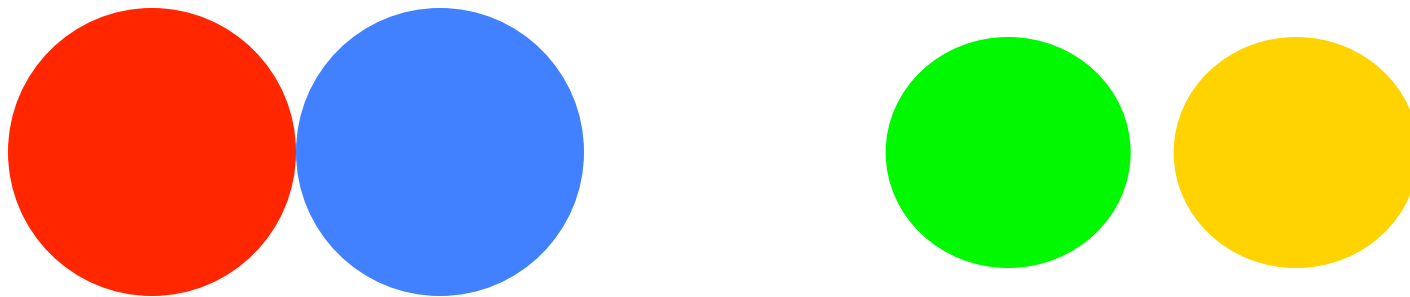
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



**3 well-separated clusters**

# Types of Clusters: Center-Based

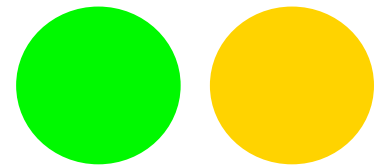
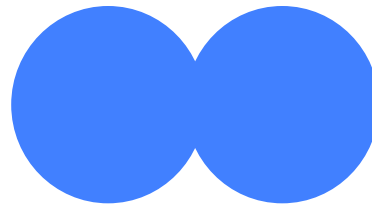
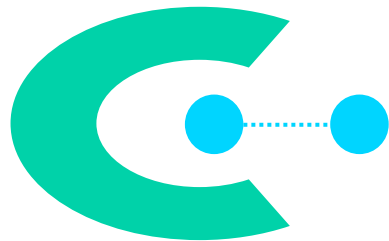
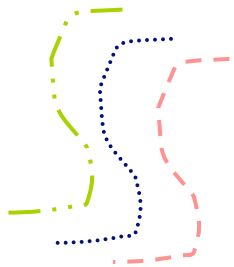
- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
  - The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most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



4 center-based clusters

# Types of Clusters: Contiguity-Based

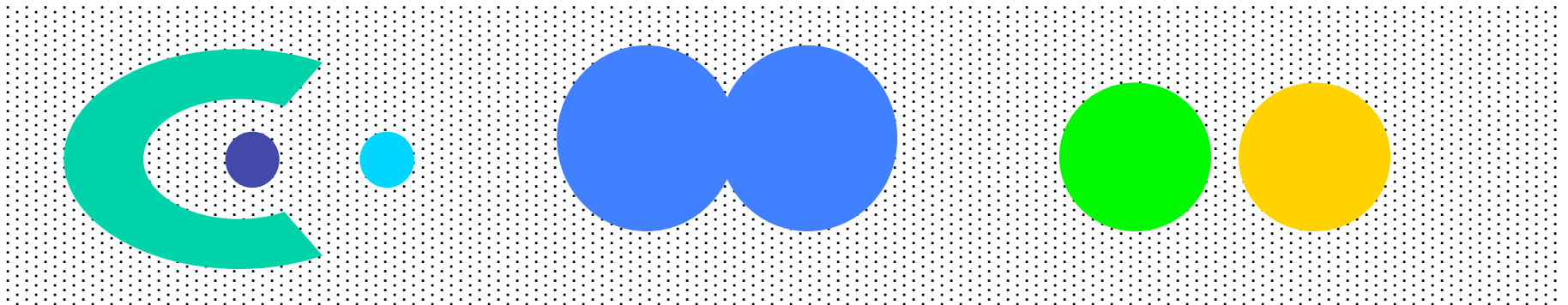
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



**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.

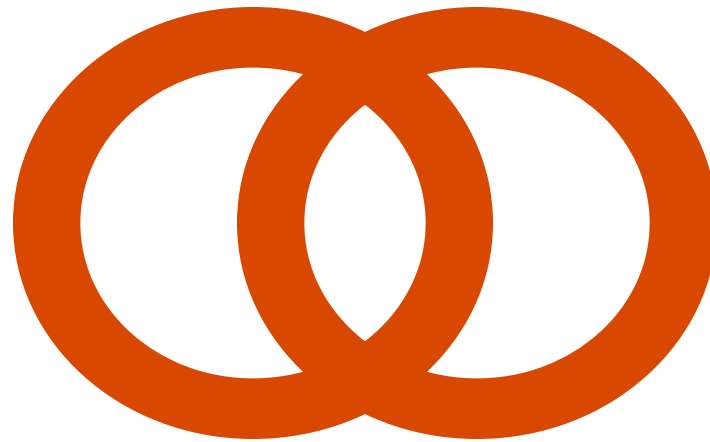


**6 density-based clusters**

---

# 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.




# Types of Clusters: Objective Function

- 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



# Clustering Algorithms

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

# K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters,  $K$ , must be specified
- 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

---

---

# Interactive Demo

- [http://home.dei.polimi.it/matteucc/Clustering/tutorial\\_html/AppletKM.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html)



# 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

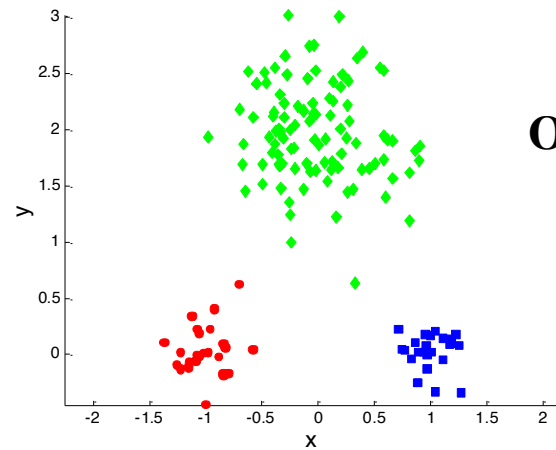
# Evaluating K-means Clusters

- 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.

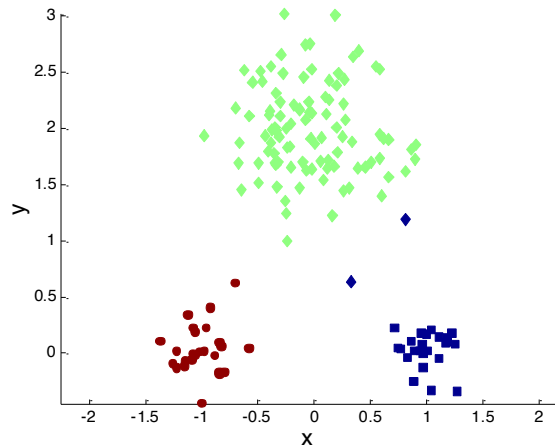
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

- $x$  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 clusters, we can choose 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  $K$

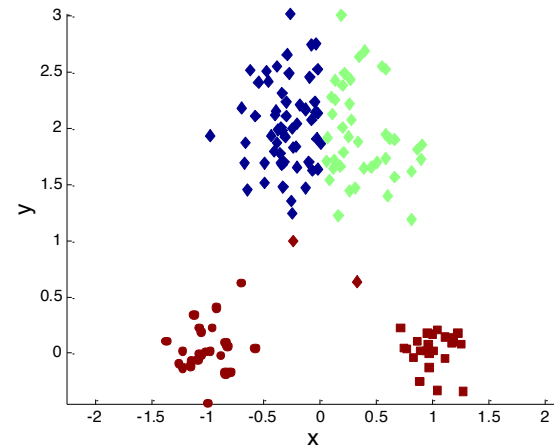
# Two different K-means Clusterings



**Original Points**

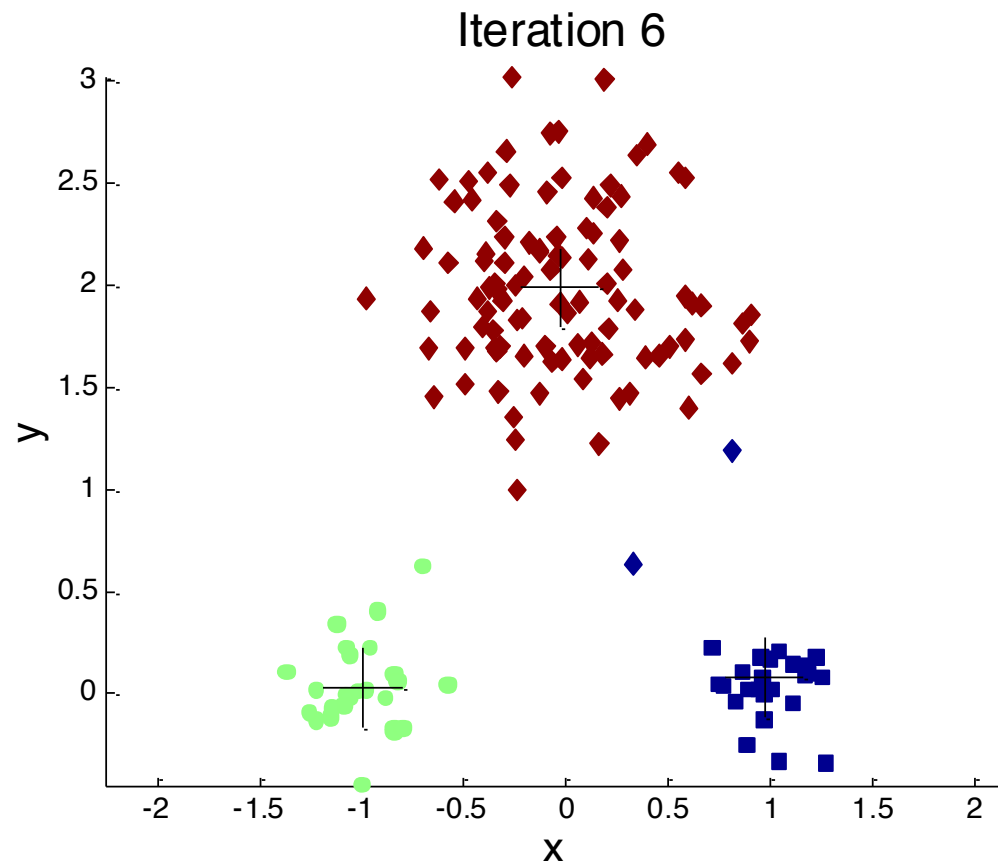


**Optimal Clustering**



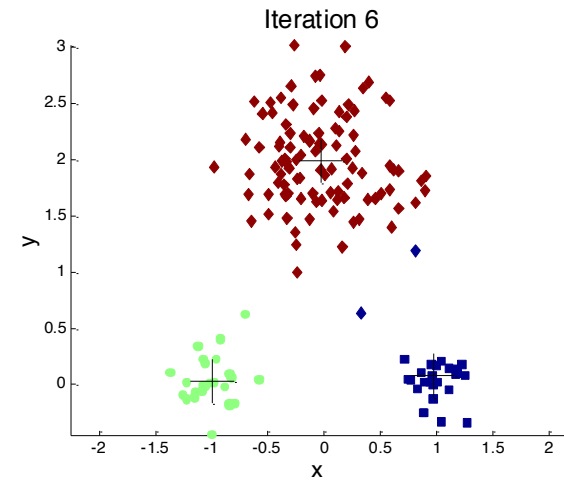
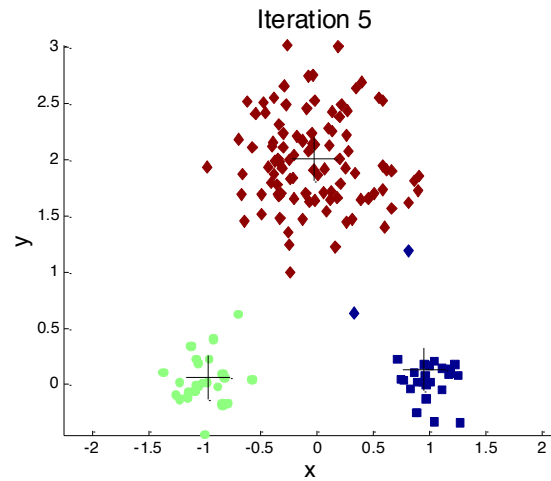
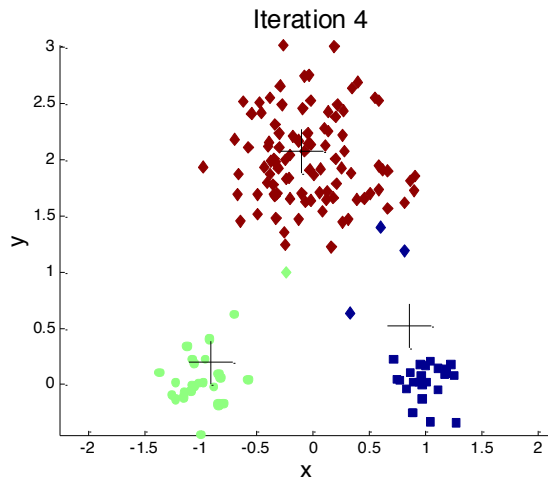
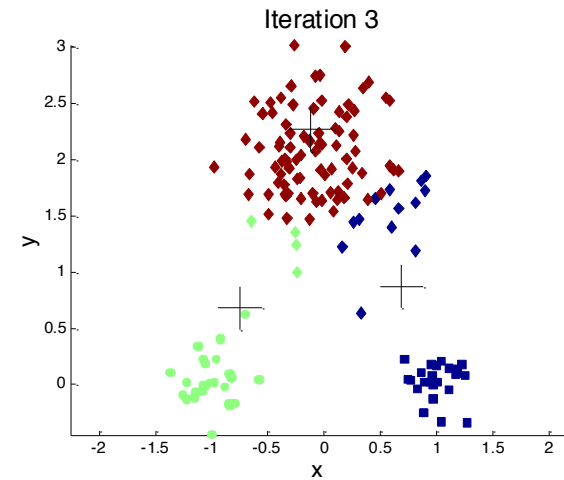
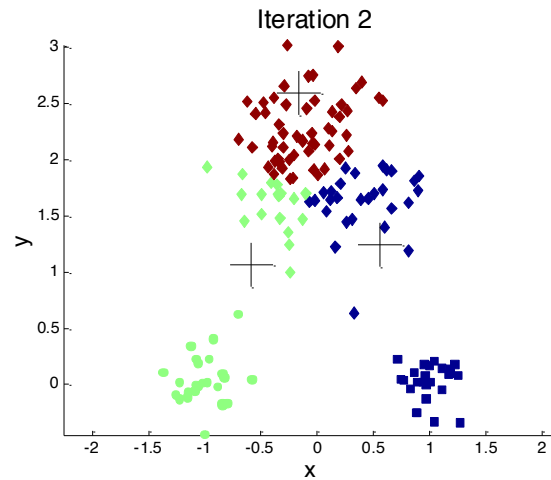
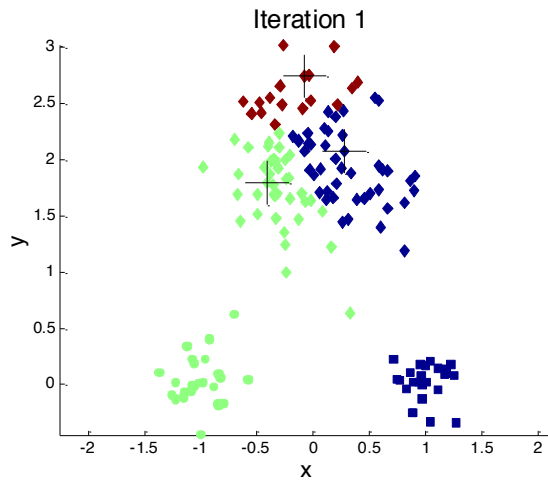
**Sub-optimal Clustering**

# Importance of Choosing Initial Centroids

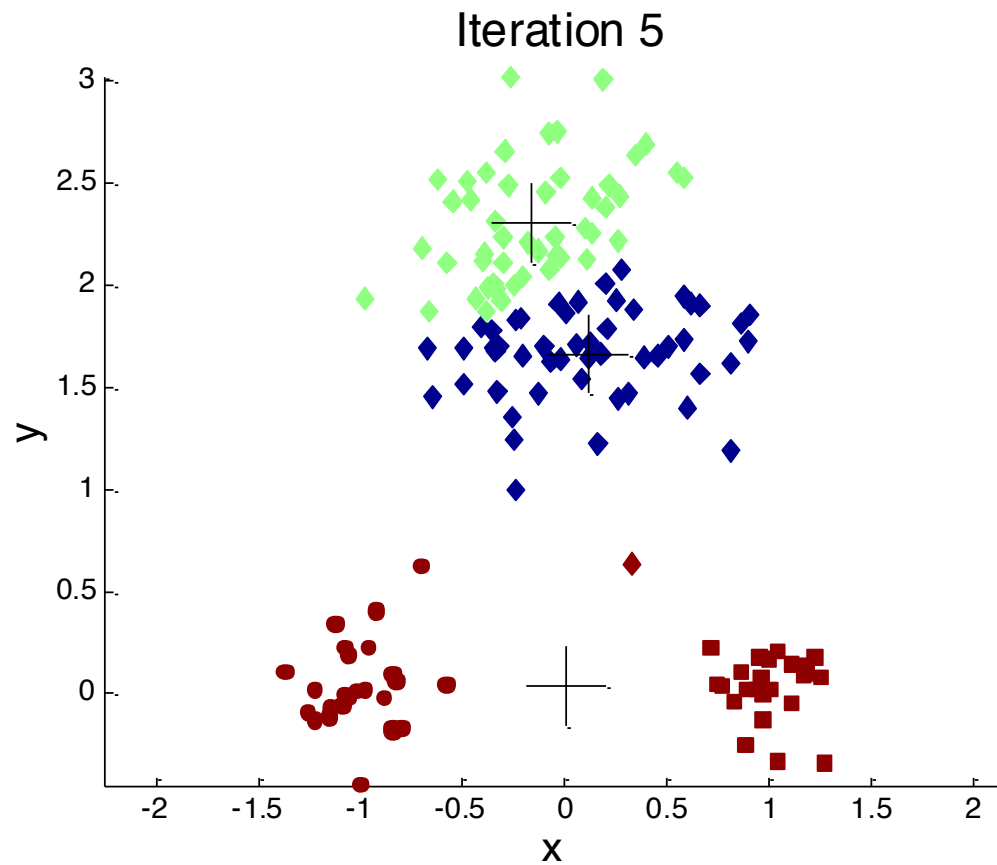




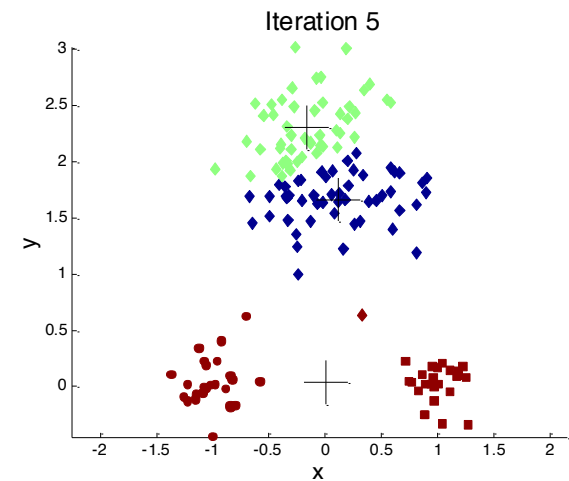
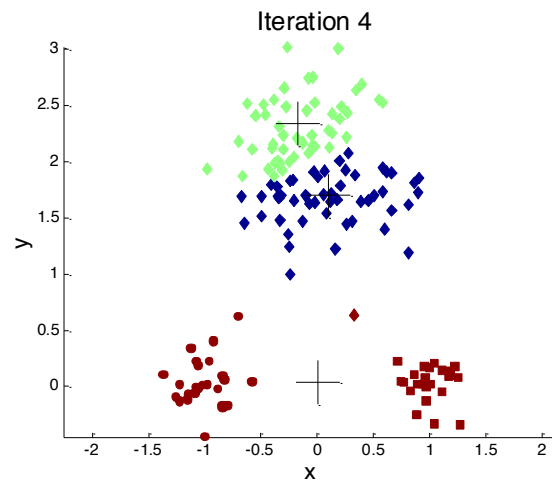
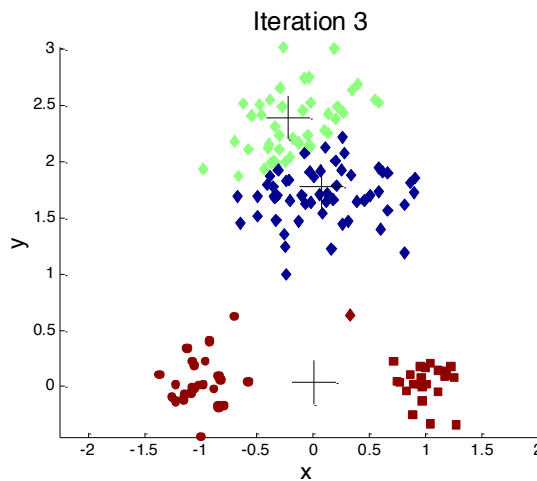
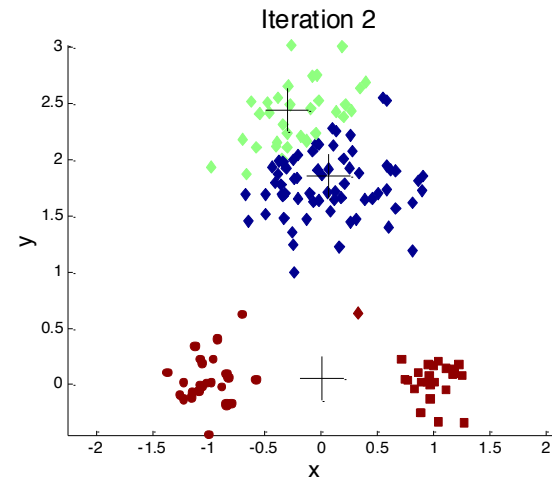
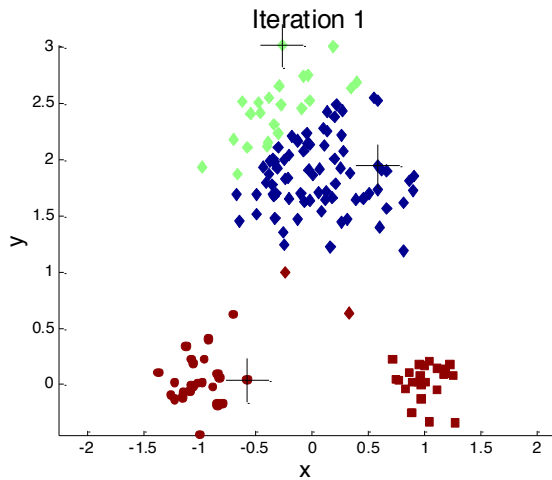
# Importance of Choosing Initial Centroids



# Importance of Choosing Initial Centroids



# Importance of Choosing Initial Centroids ...



# Problems with Selecting Initial Points

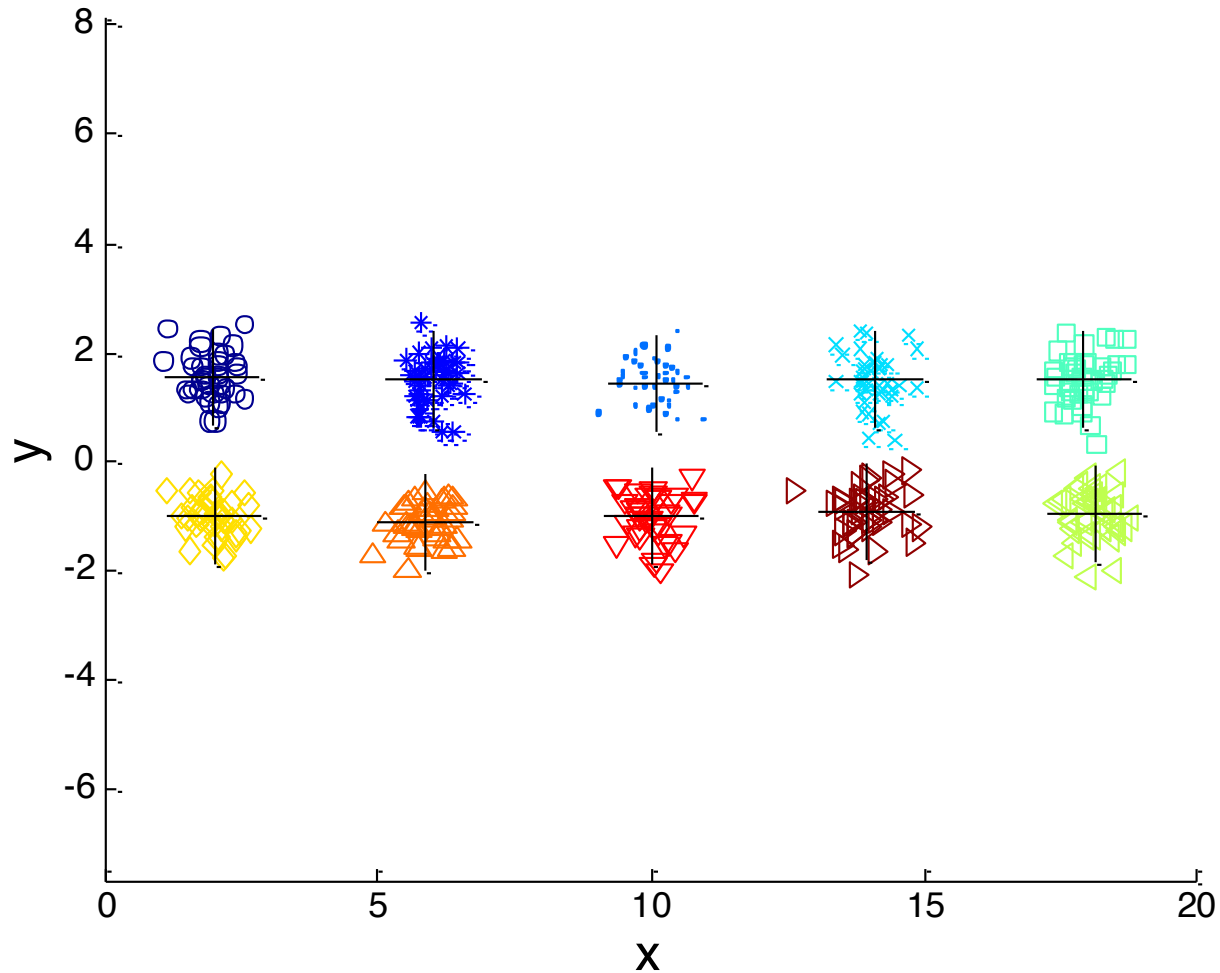
- If there are  $K$  ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when  $K$  is large
  - If clusters are the same size,  $n$ , then

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

- 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

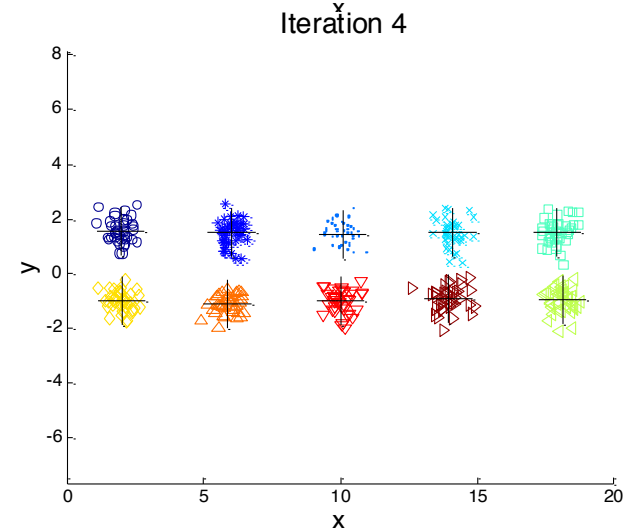
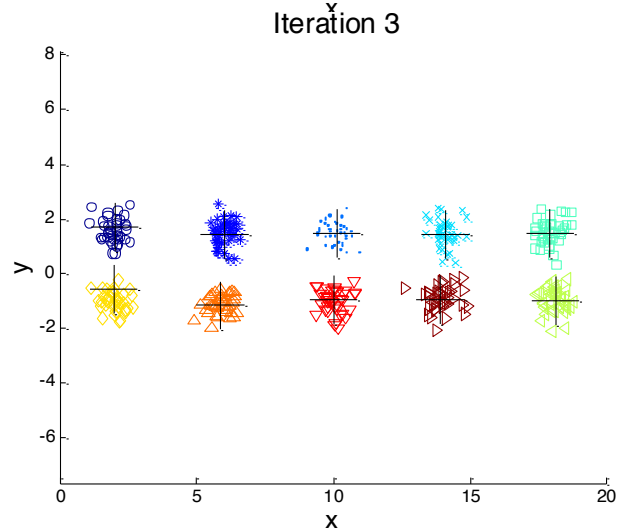
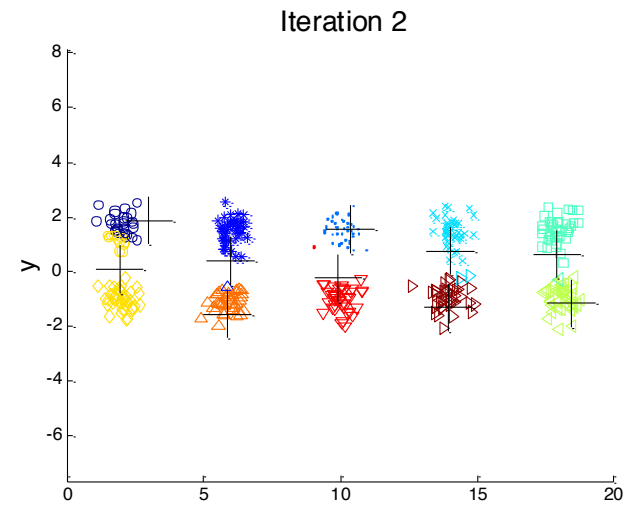
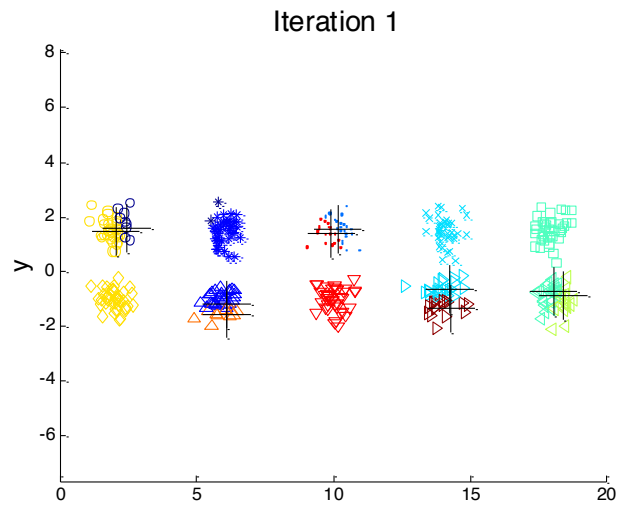
# 10 Clusters Example

Iteration 4



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

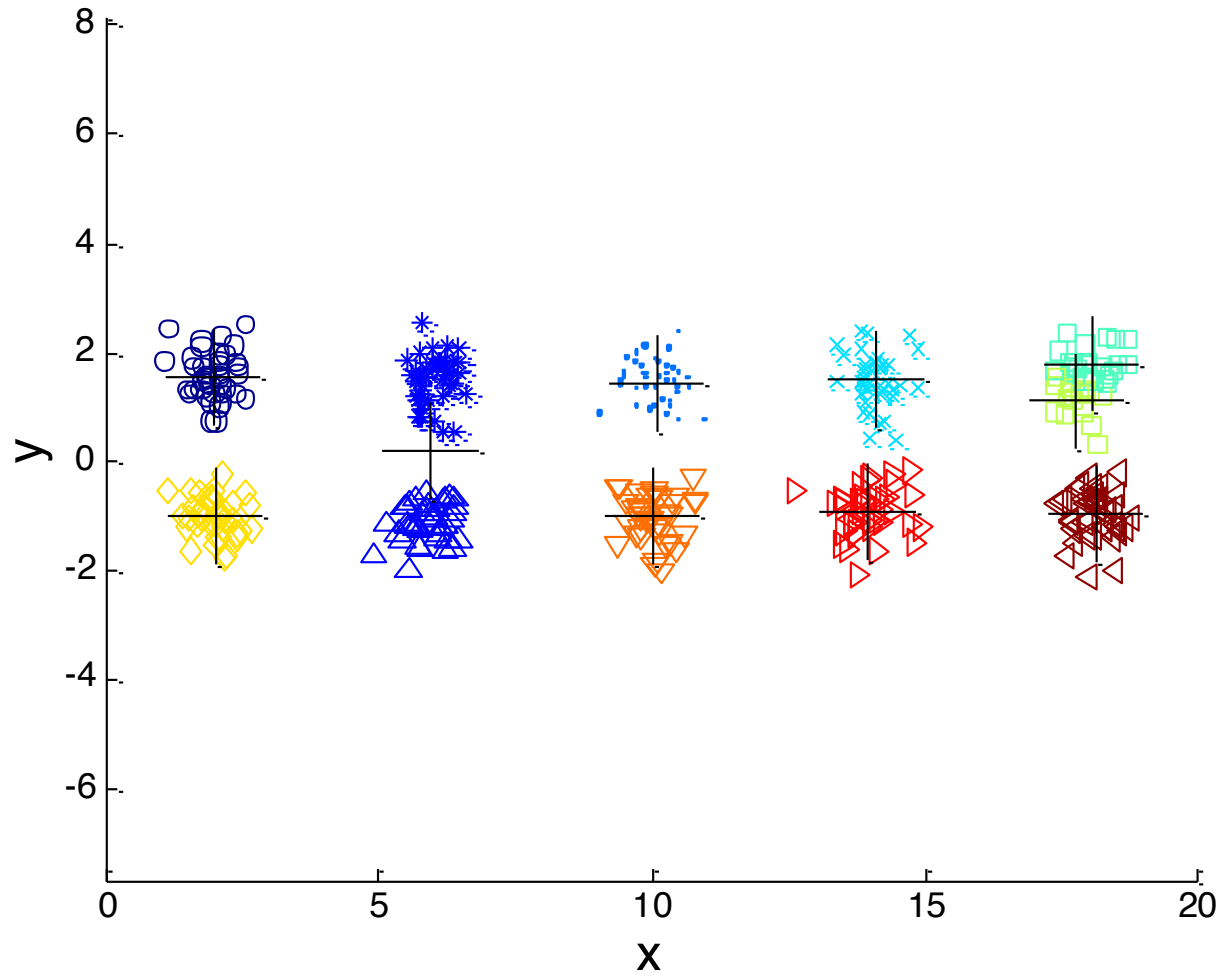
# 10 Clusters Example



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

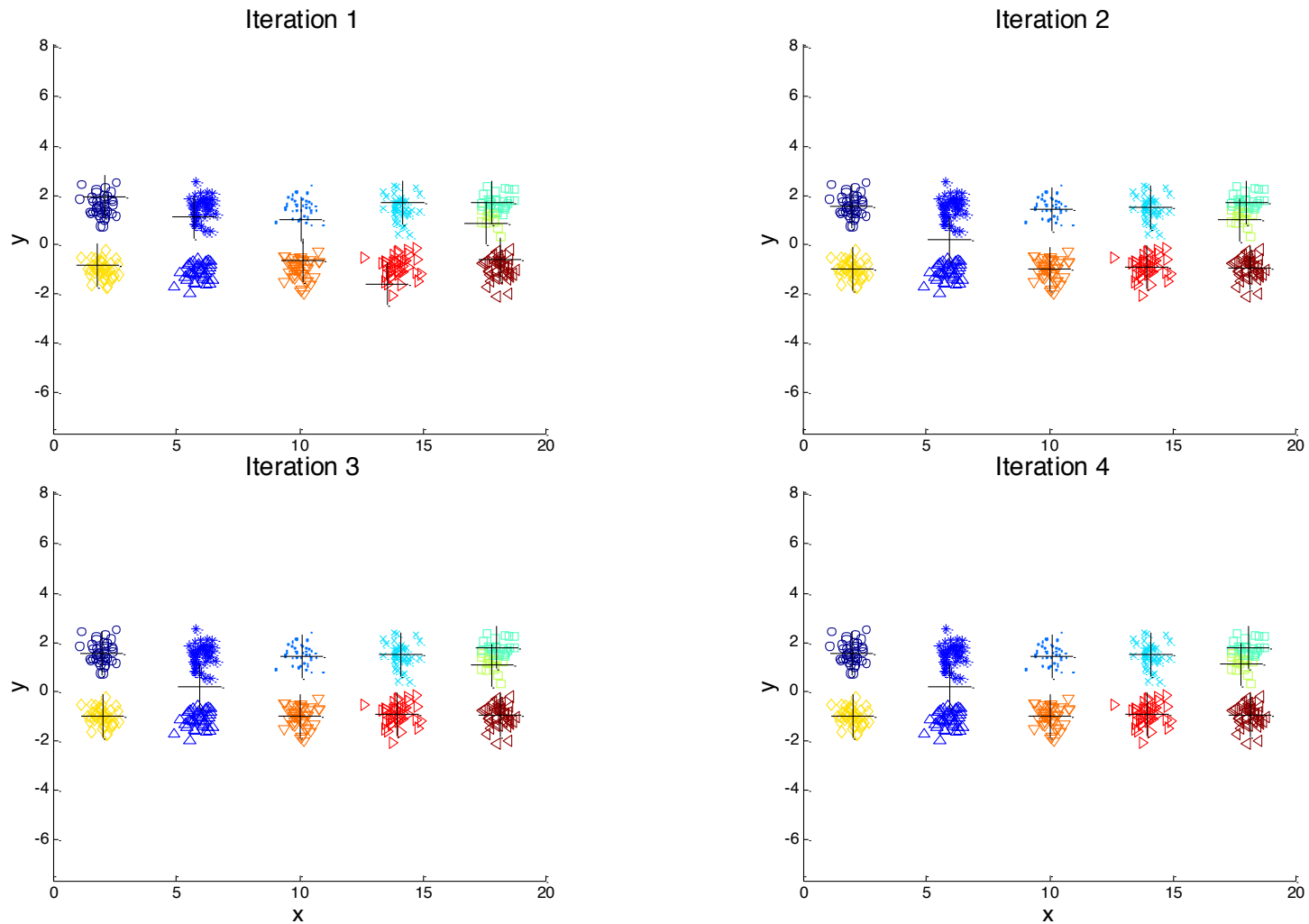
# 10 Clusters Example

Iteration 4



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
- Bisecting K-means
  - Not as susceptible to initialization issues

# Bisecting K-means

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

---

- 1: Initialize the list of clusters to contain the cluster containing all points.
- 2: **repeat**
- 3:   Select a cluster from the list of clusters
- 4:   **for**  $i = 1$  to *number\_of\_iterations* **do**
- 5:     Bisect the selected cluster using basic K-means
- 6:   **end for**
- 7:   Add the two clusters from the bisection with the lowest SSE to the list of clusters.
- 8: **until** Until the list of clusters contains  $K$  clusters

---

# Bisecting K-means Example

Iteration 10

