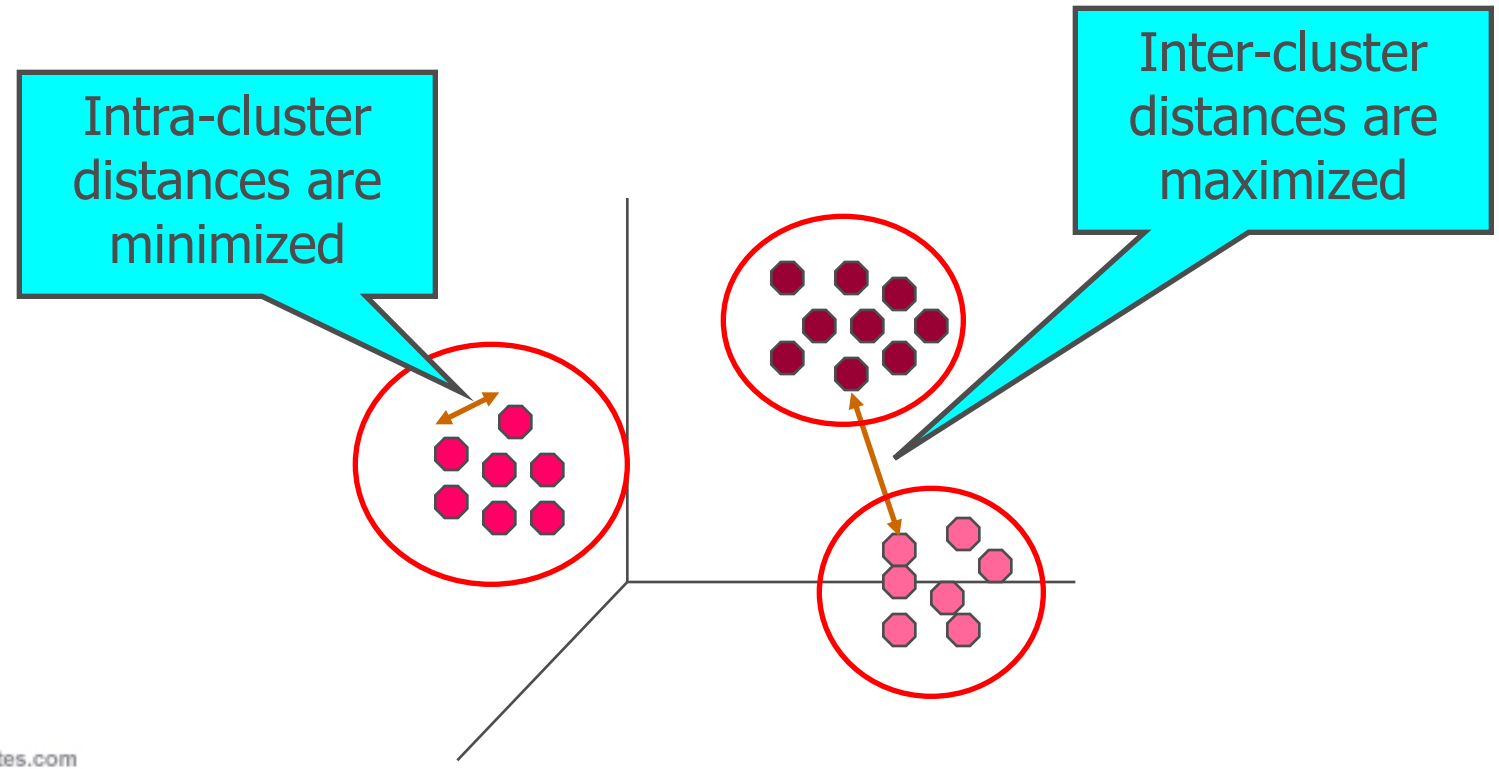
Part 10

Basic Cluster Analysis

NGURAH AGUS SANJAYA ER, S.KOM, M.KOM, PH.D
E-mail: agus.sanjaya@cs.unud.ac.id

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

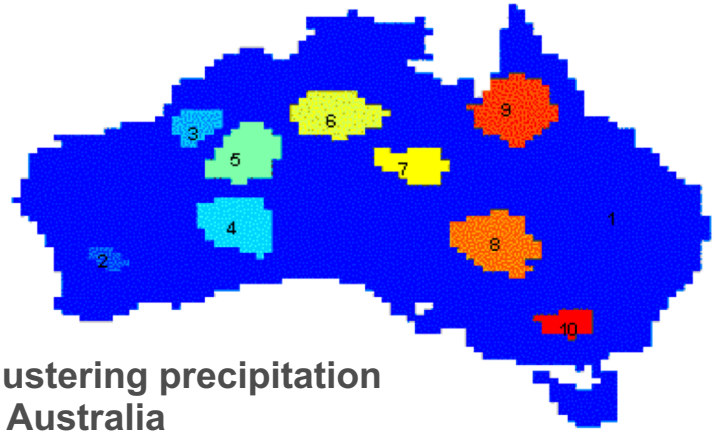


Applications of Cluster Analysis

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

	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?

- 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
- Graph partitioning
  - Some mutual relevance and synergy, but areas are not identical

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Two Clusters

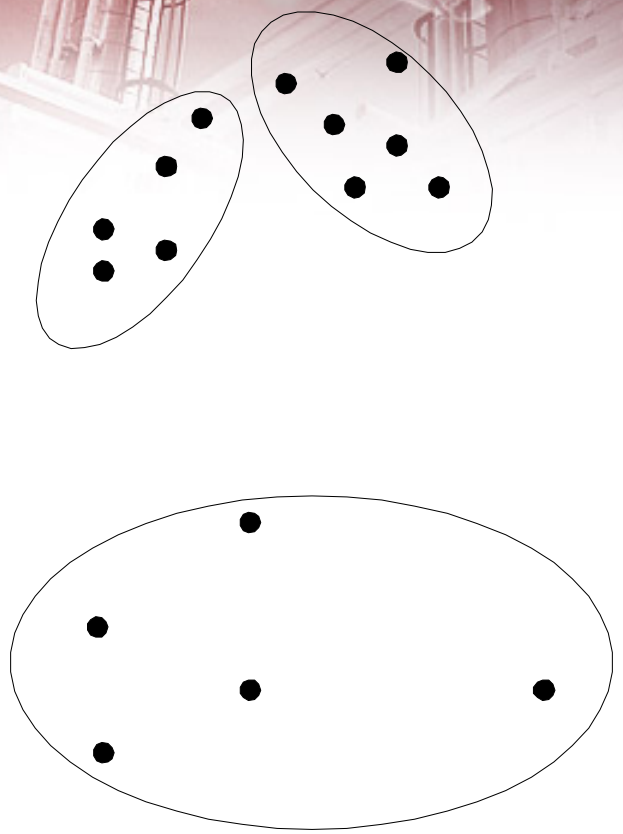
Four Clusters

Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division 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

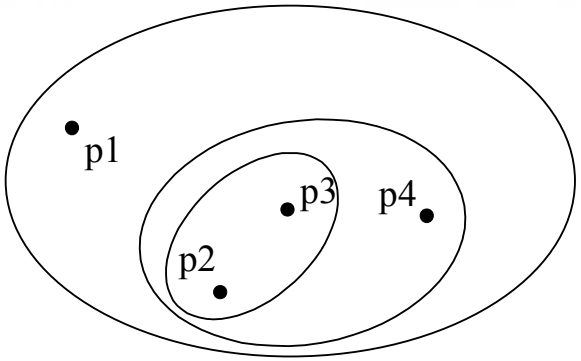
Partitional Clustering

Original Points

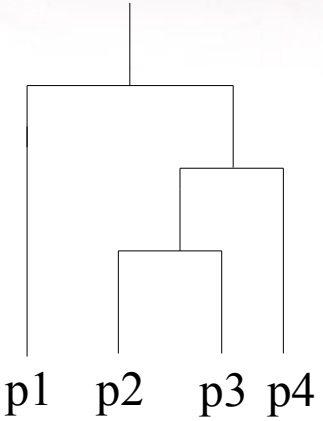


A Partitional Clustering

Hierarchical Clustering



Hierarchical Clustering



Dendrogram

Other Distinctions Between Sets of Clusters

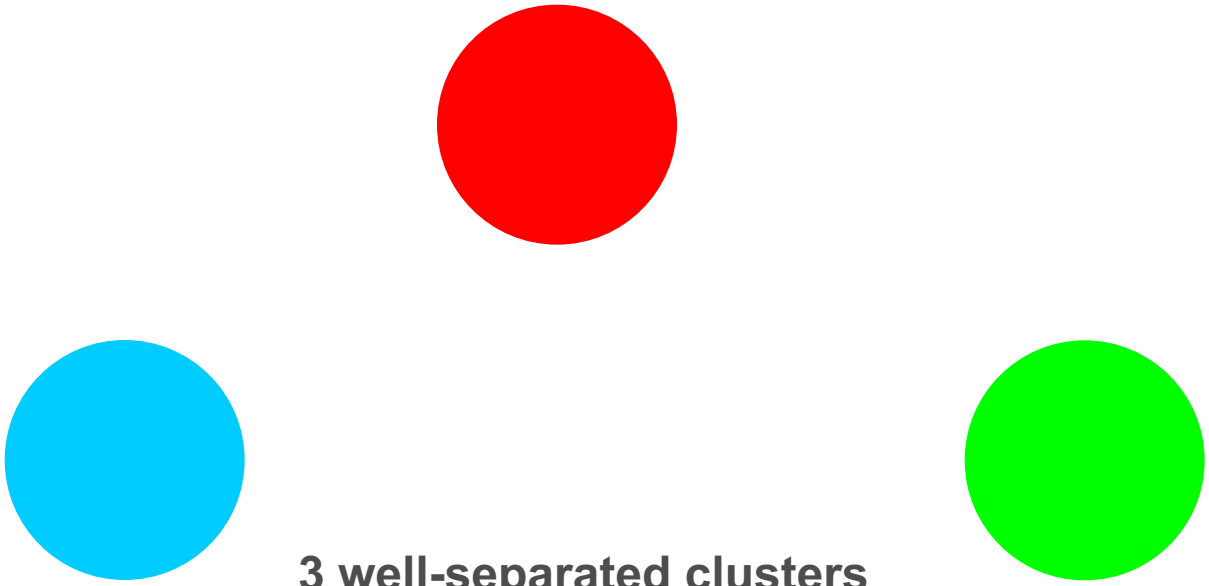
- Exclusive versus non-exclusive
- In non-exclusive clusterings, points may belong to multiple clusters.
  - Can represent multiple classes or 'border' points
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
  - 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

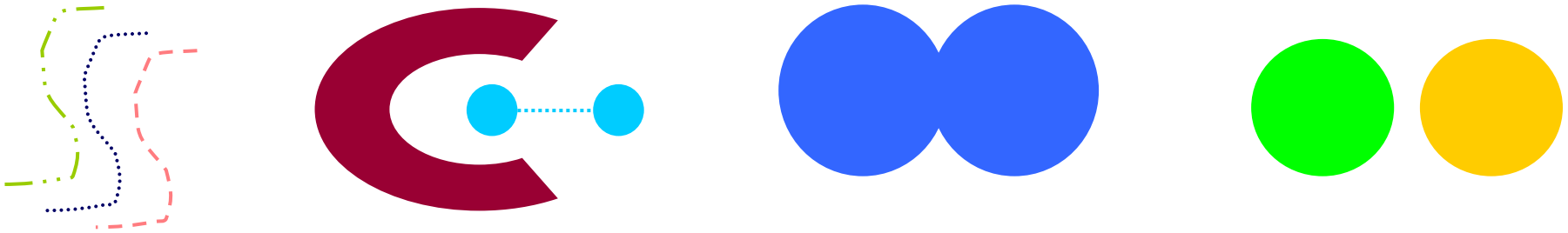
Types of Clusters: Center-Based

- 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

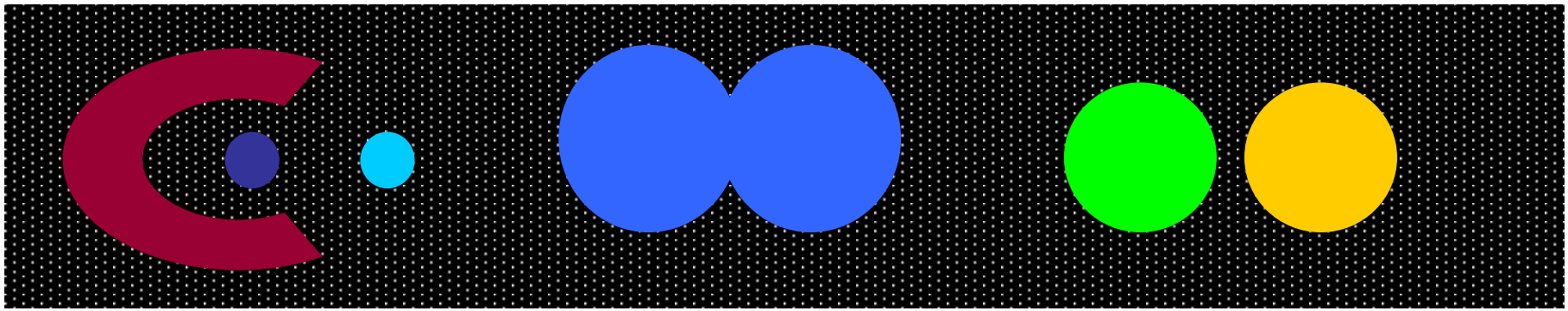
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to some other points in the cluster than to any point not in the cluster.



8 contiguous clusters

Types of Clusters: Density-Based

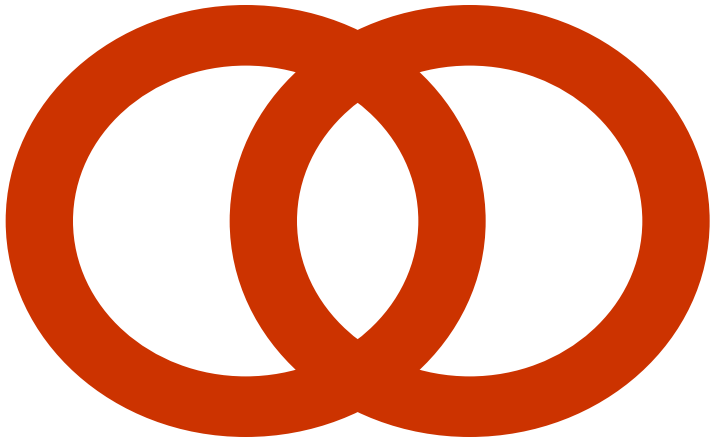
- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



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

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

Algorithm 1 Basic K-means Algorithm.

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 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**

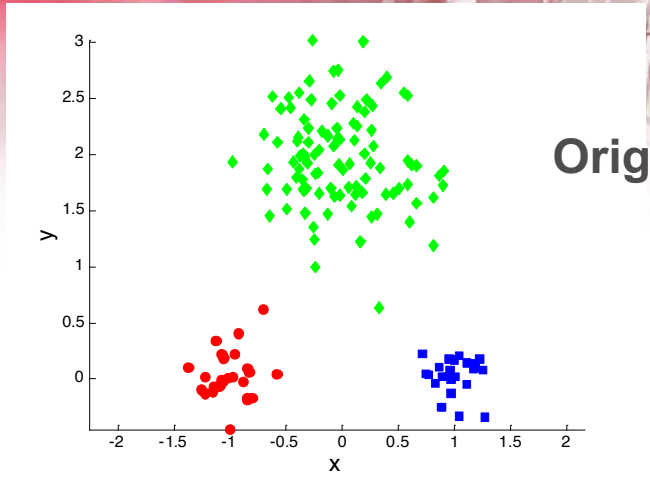
Evaluating K-means Clusters

- 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.

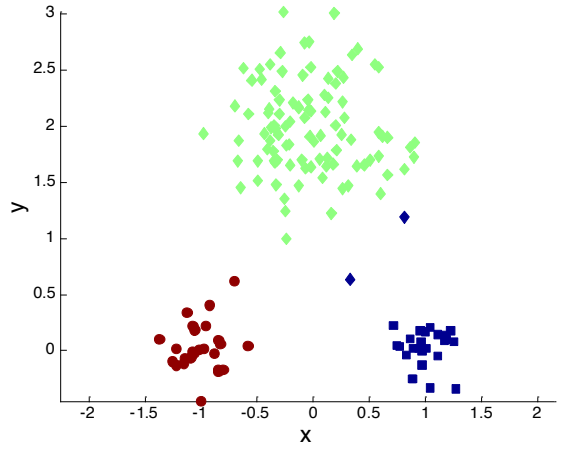
$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(c_i, x) \qquad c_i = \frac{1}{m_i} \sum_{x \in C_i} x$$

  - x is a data point in cluster C_i and c_i is the centroid for cluster C_i while m_i is the number of points in C_i
  - 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

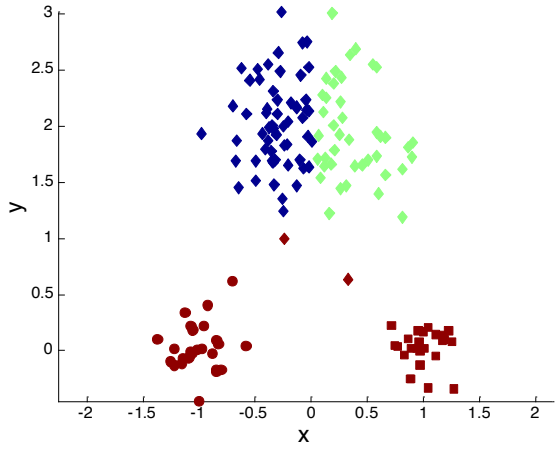
Two different K-means Clusterings



Original Points

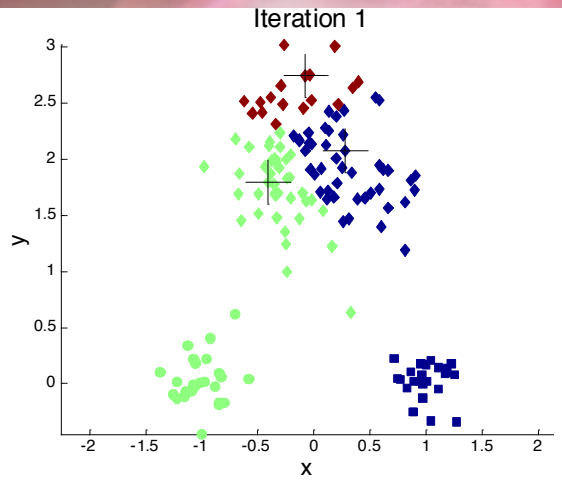


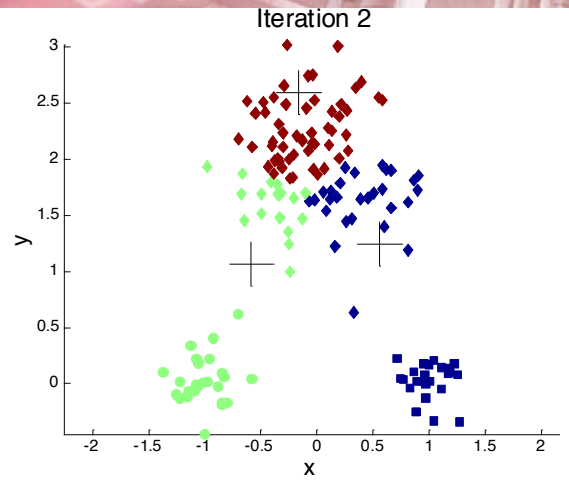
Optimal Clustering

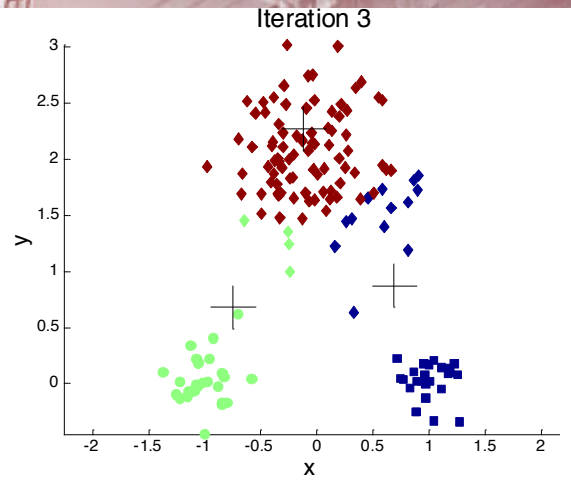


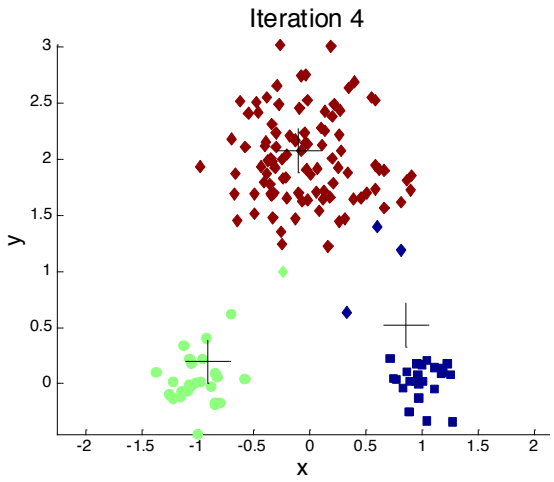
Sub-optimal Clustering

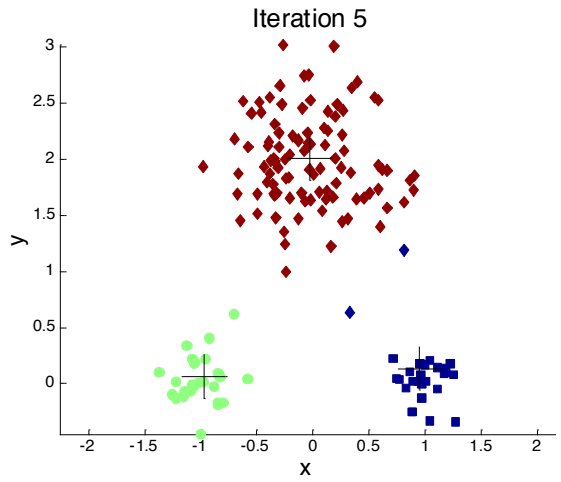
Importance of Choosing Initial Centroids

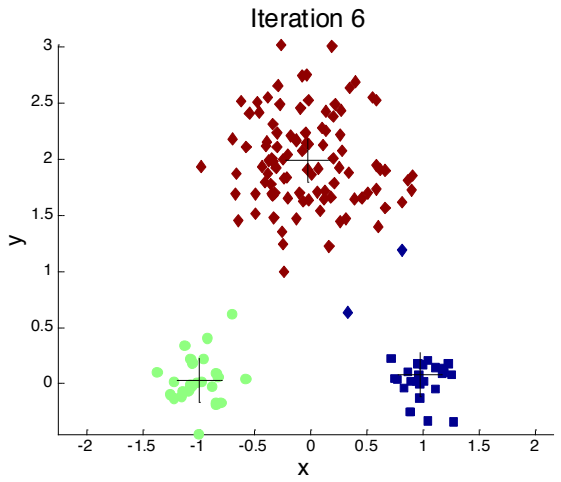




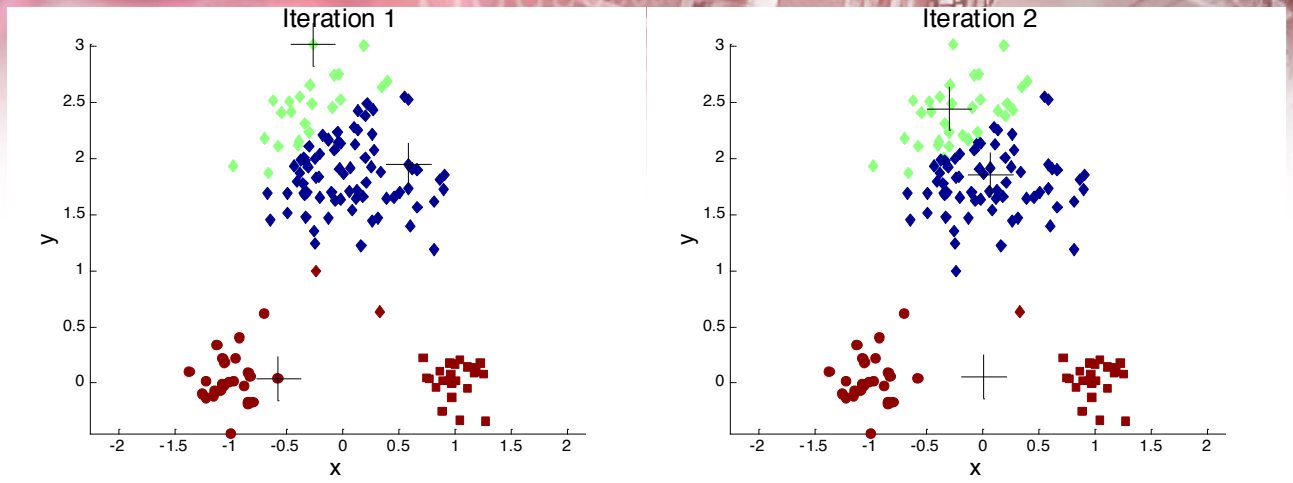


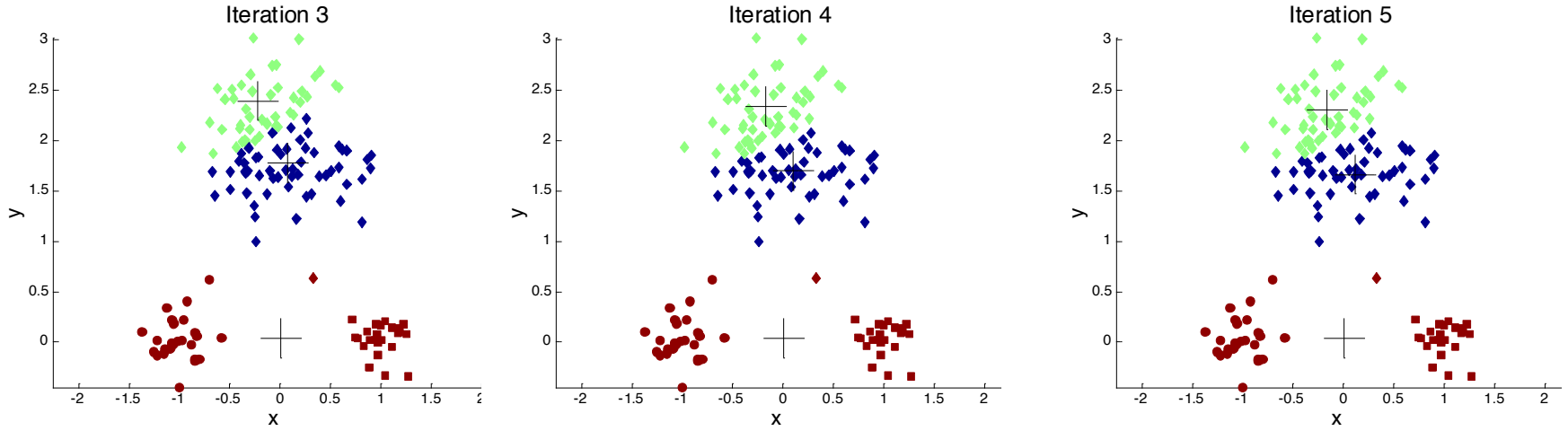






Importance of Choosing Initial Centroids …





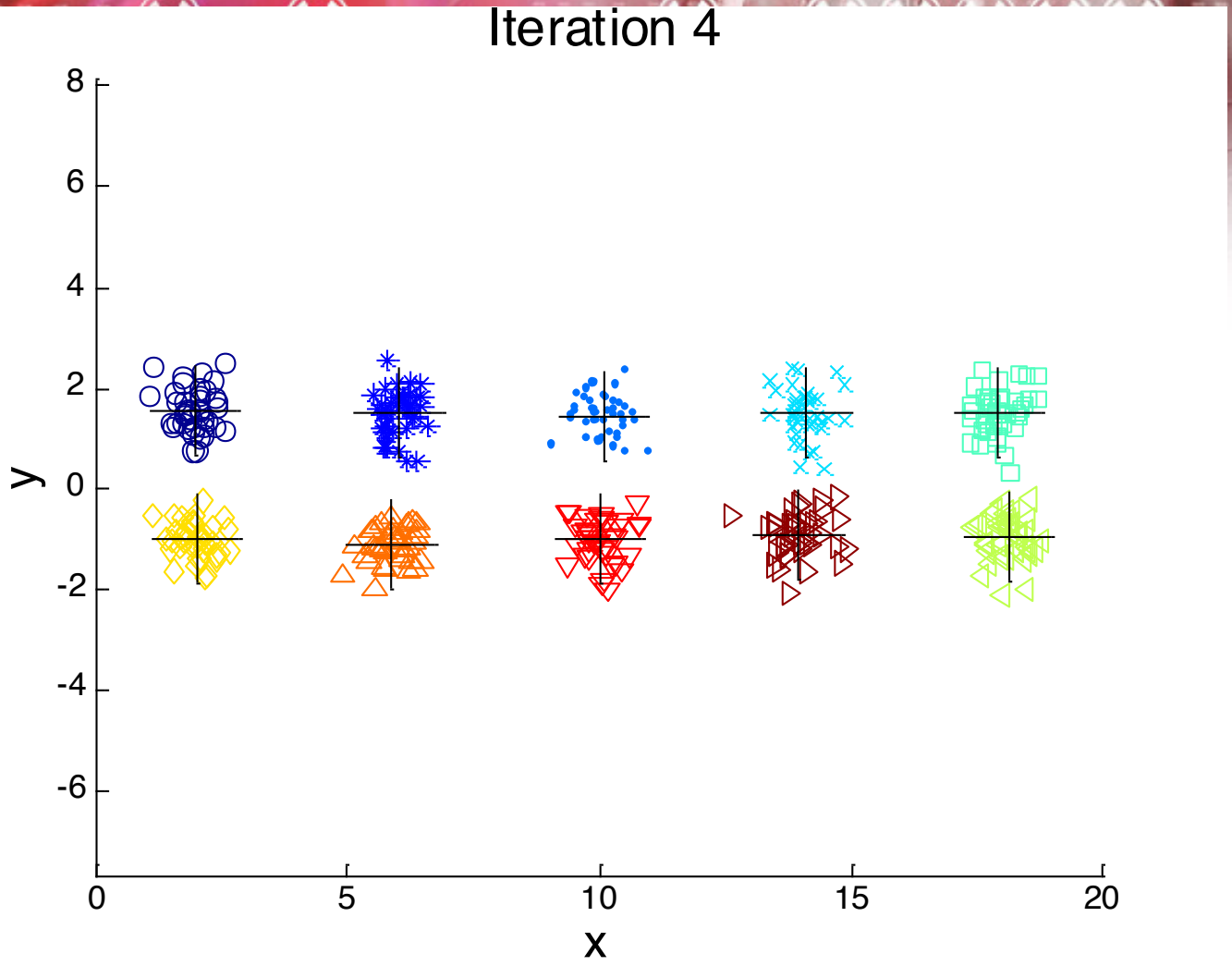
Problems with Selecting Initial Points

- 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

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

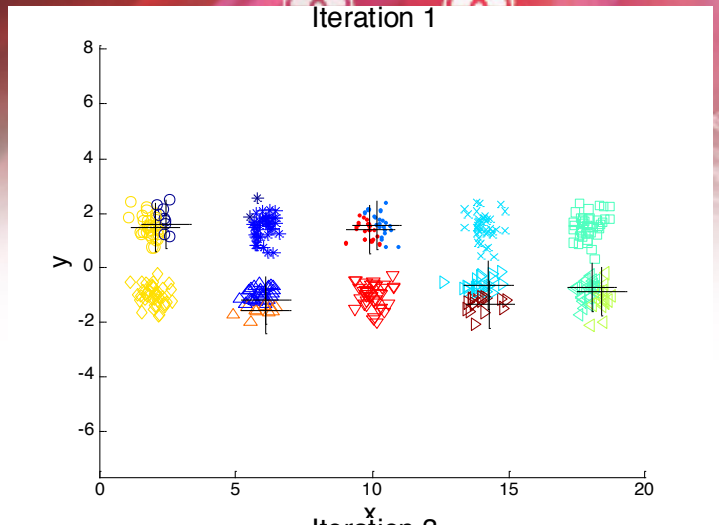
  - For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

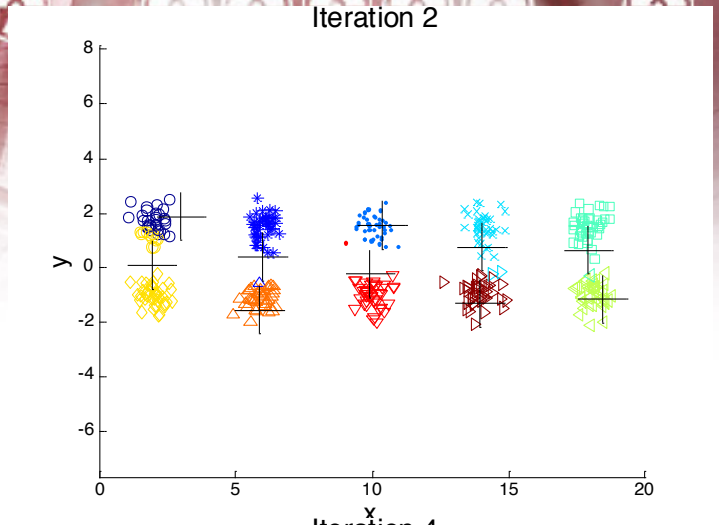
10 Clusters Example

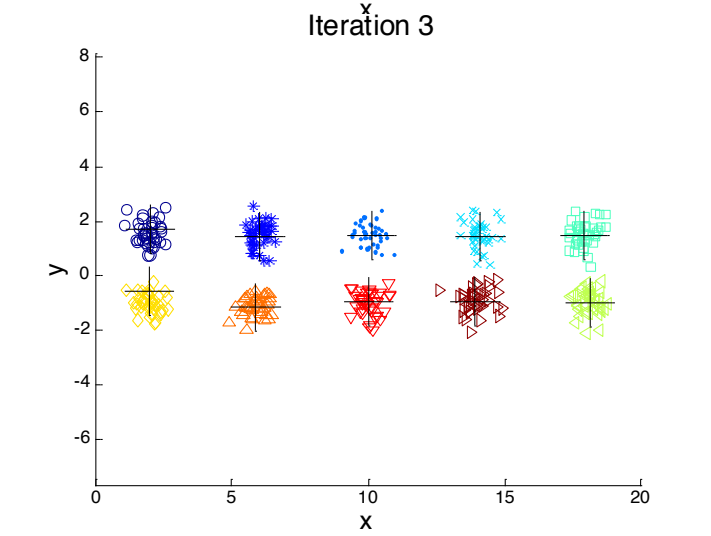


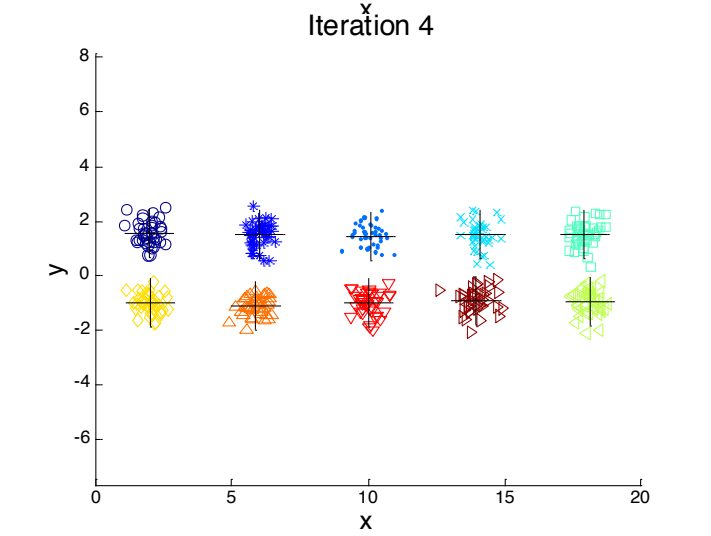
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



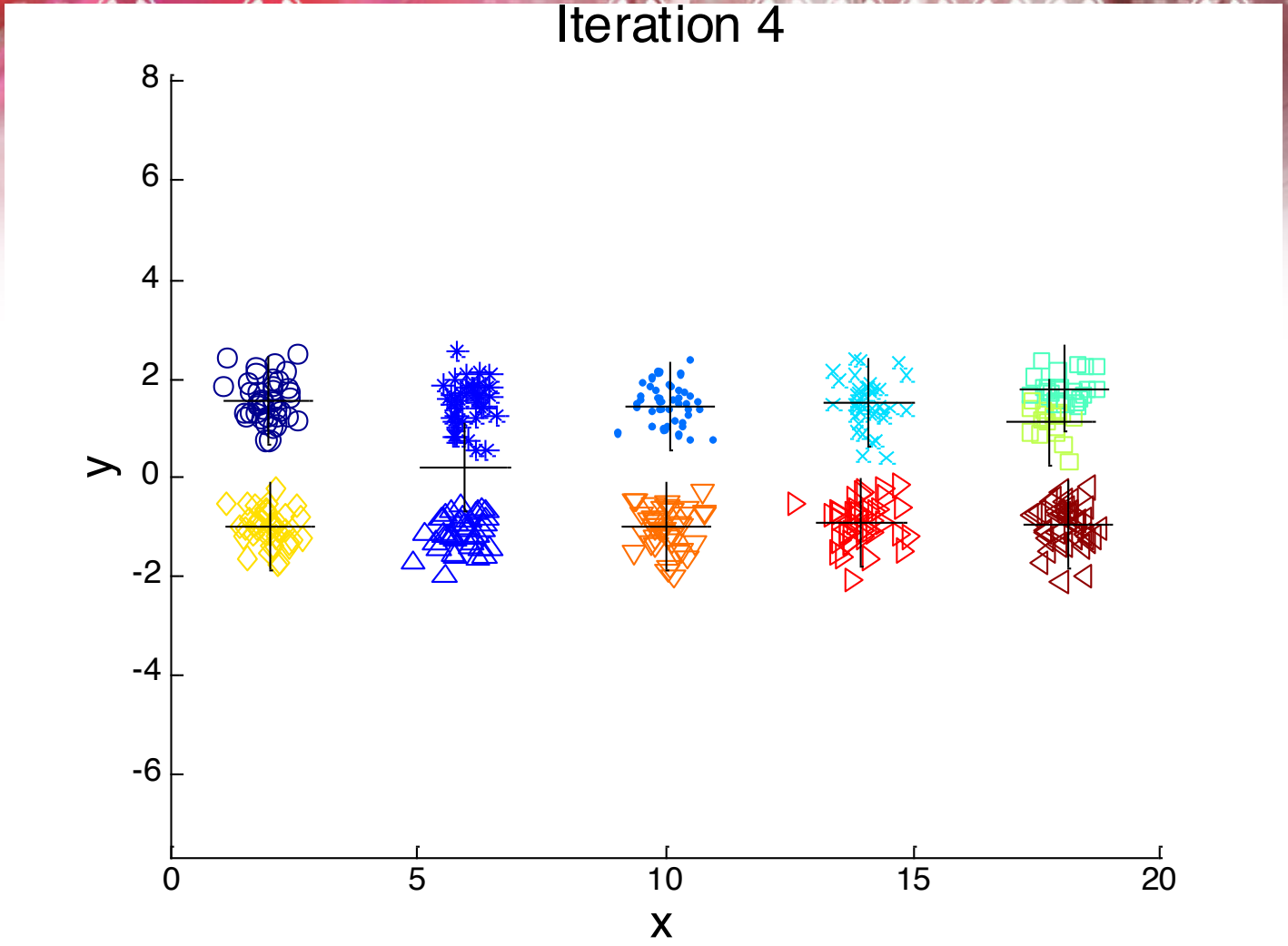






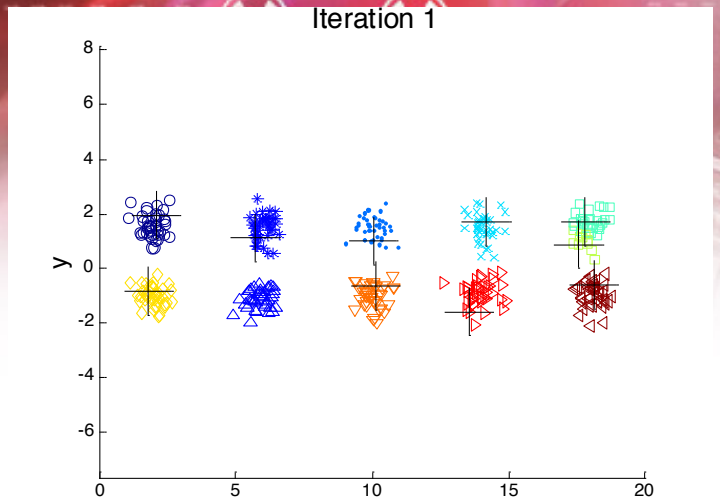
Starting with two initial centroids in one cluster of each pair of clusters

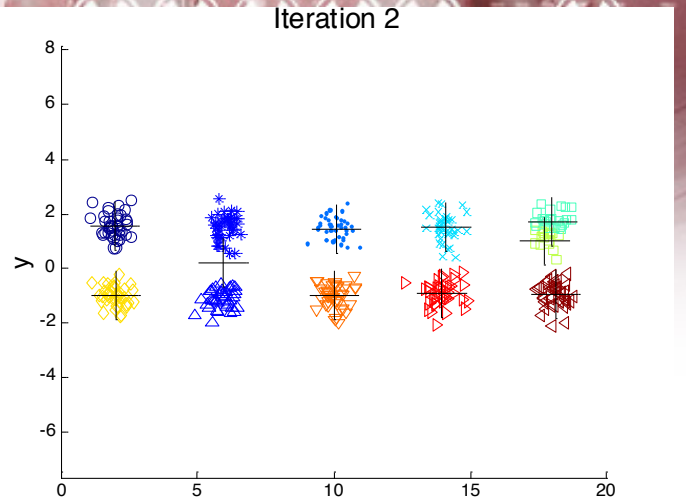
10 Clusters Example

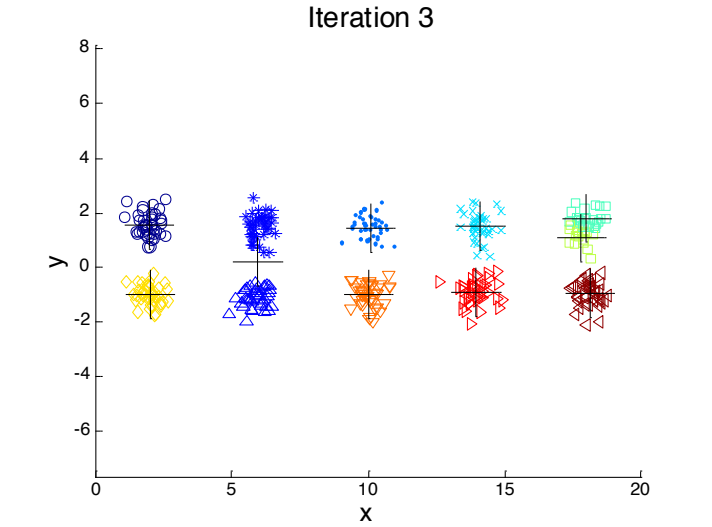


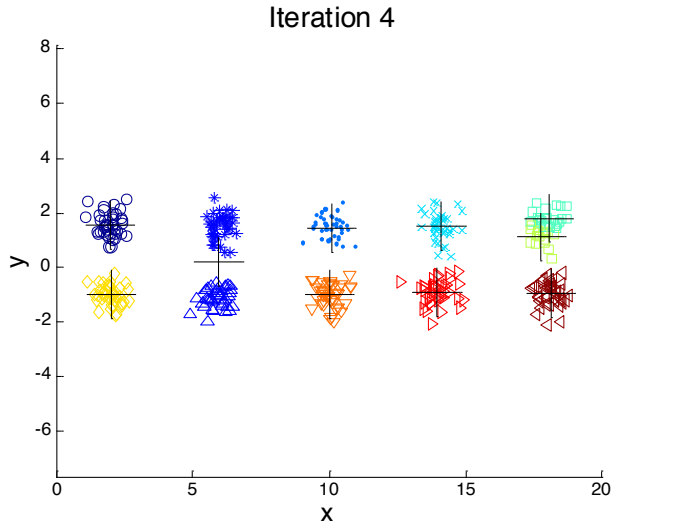
Starting with some pairs of clusters having three initial centroids, while other have only one.

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 points and use hierarchical clustering to extract K clusters then use the centroids as the initial centroids
  - Only works: sample is small, k is relatively small compared to sample
- Select first point as random or take centroid of all points, for each successive initial centroid take the farthest point from any of the initial centroid already selected
  - Can select outliers, expensive to compute the farthest point
- Bisecting K-means
  - Not as susceptible to initialization issues
- Postprocessing

Handling Empty Clusters

- Basic K-means algorithm can yield empty clusters

- Several strategies to choose replacement centroid:
  - Choose the point that is farthest away from any current centroid
  - Choose a point from the cluster with the highest SSE
  - If there are several empty clusters, the above can be repeated several times.

Updating Centroids 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

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

Bisecting K-means

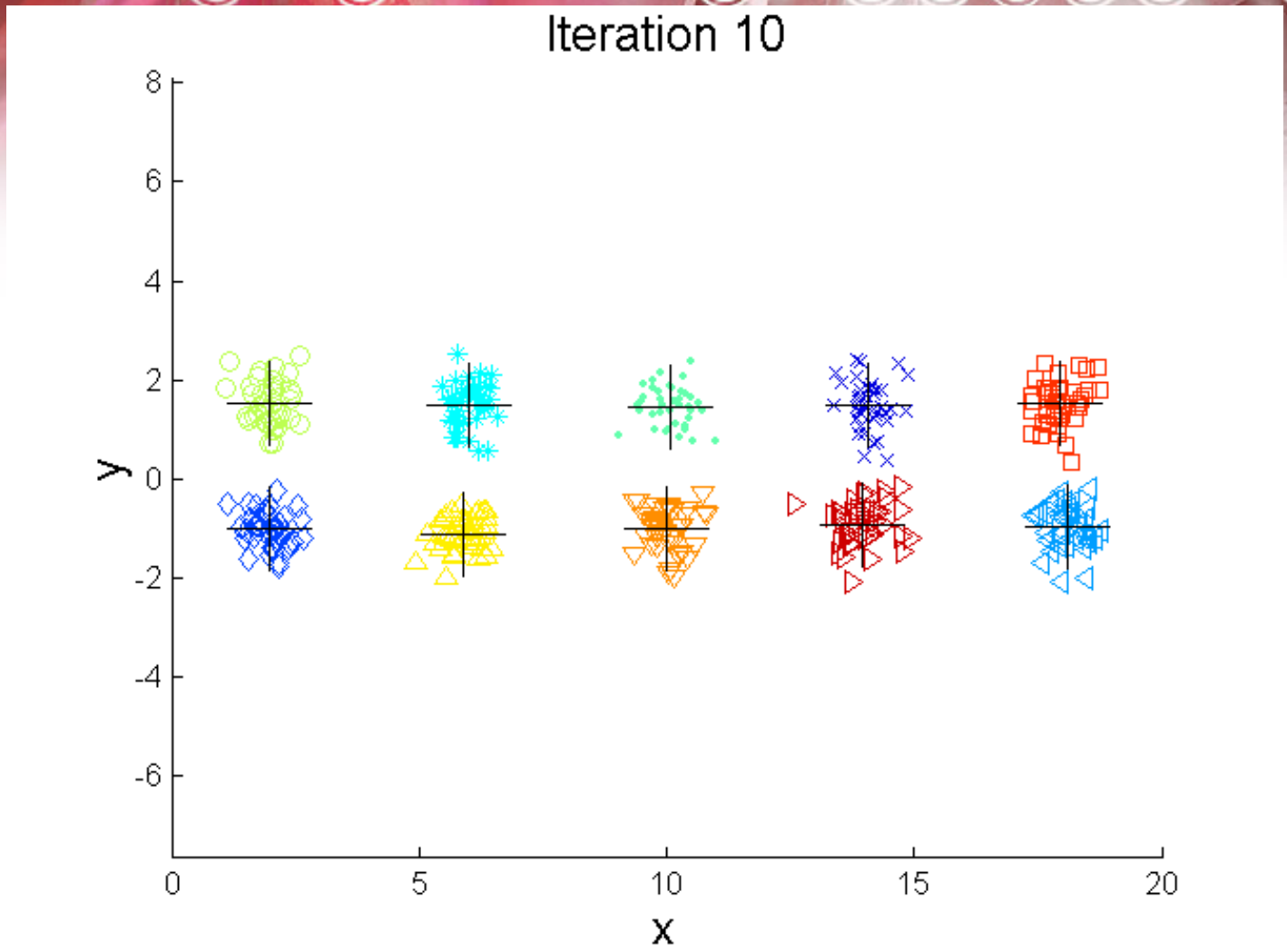
- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitional or a hierarchical clustering
  - To obtain K clusters, split the set of all points into two clusters
  - Select one of those clusters to split until K clusters have been produced
  - There are a number of ways to choose which cluster to split:
    - Split the largest cluster at each step
    - Split the one with the largest SSE
    - Or used a criterion based on both size and SSE

Bisecting K-means

- Bisecting K-means algorithm

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

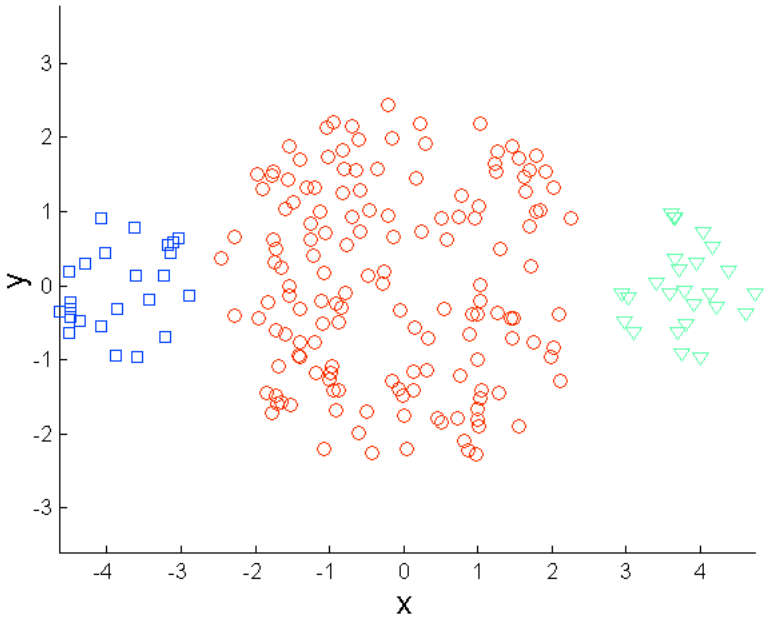


Limitations of K-means

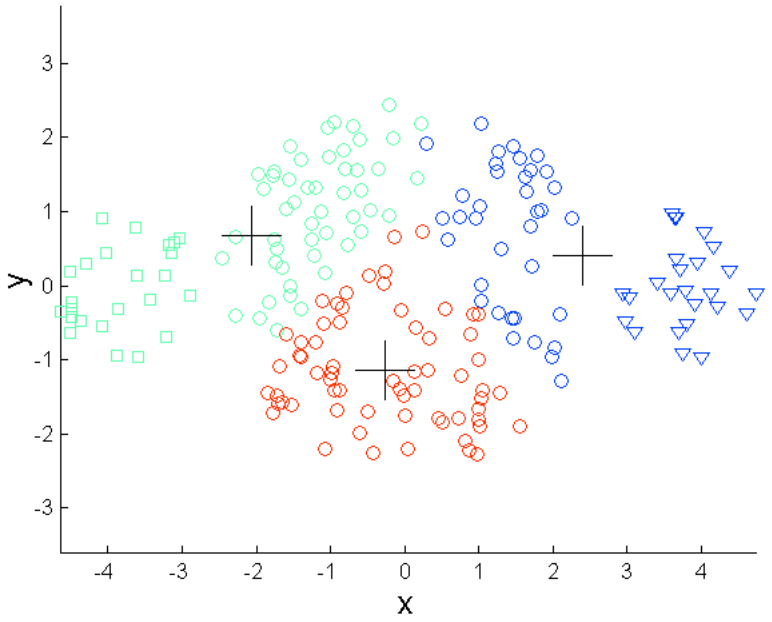
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

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

Limitations of K-means: Differing Sizes

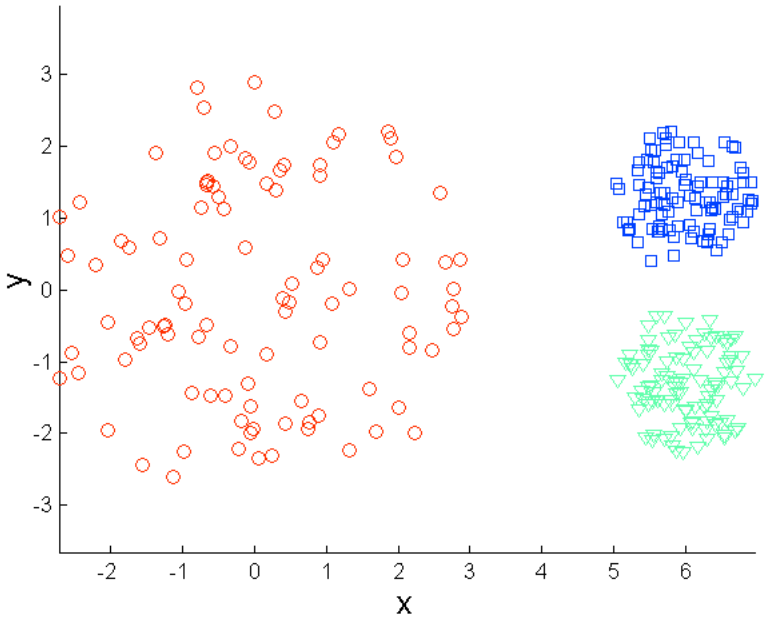


Original Points

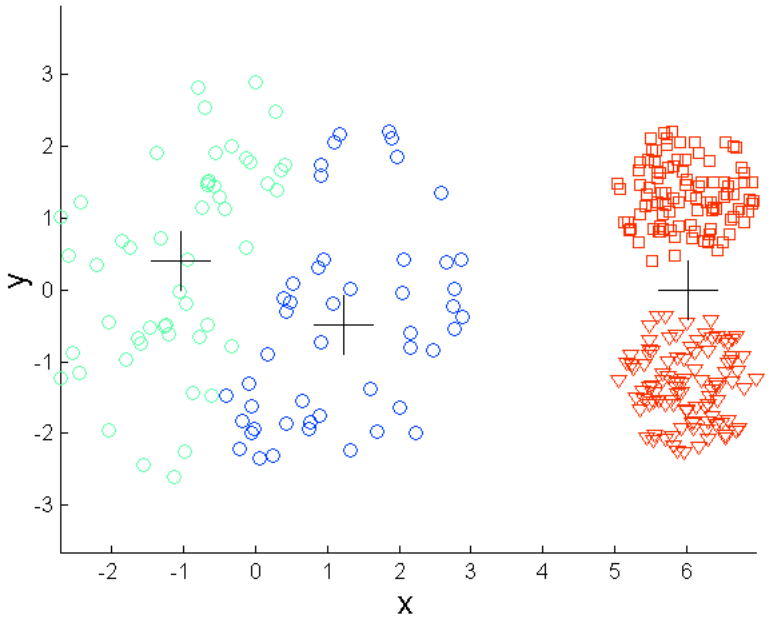


K-means (3 Clusters)

Limitations of K-means: Differing Density

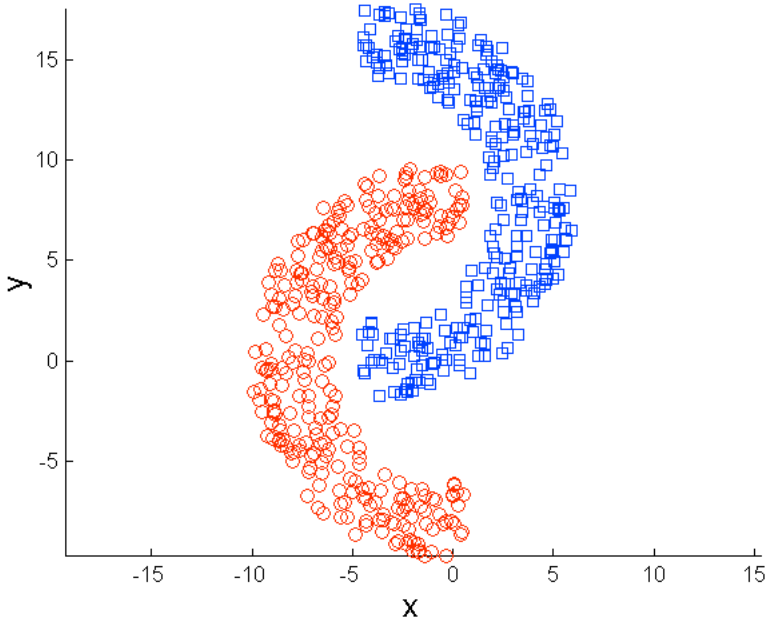


Original Points

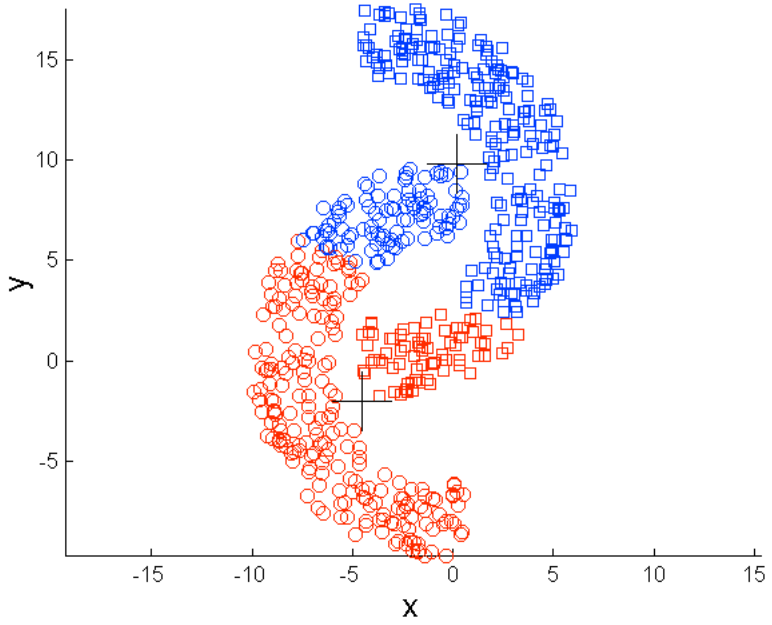


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

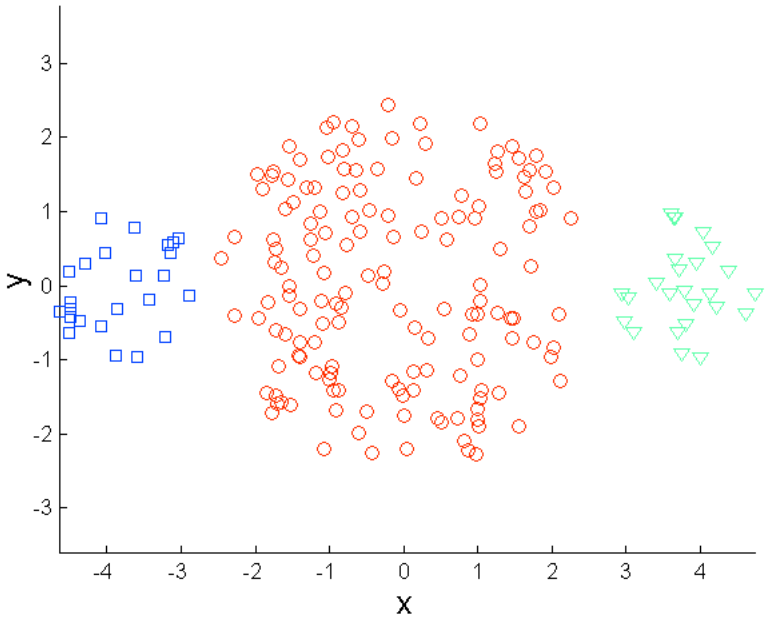


Original Points

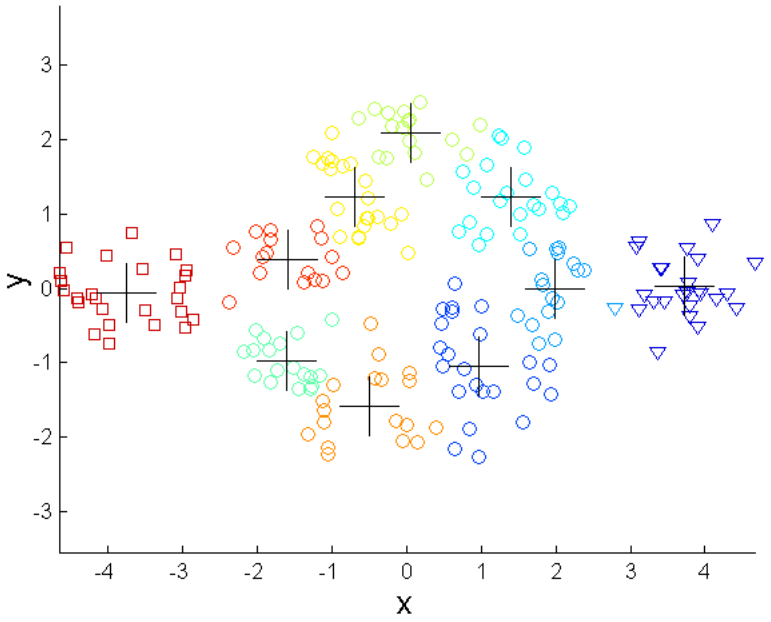


K-means (2 Clusters)

Overcoming K-means Limitations



Original Points

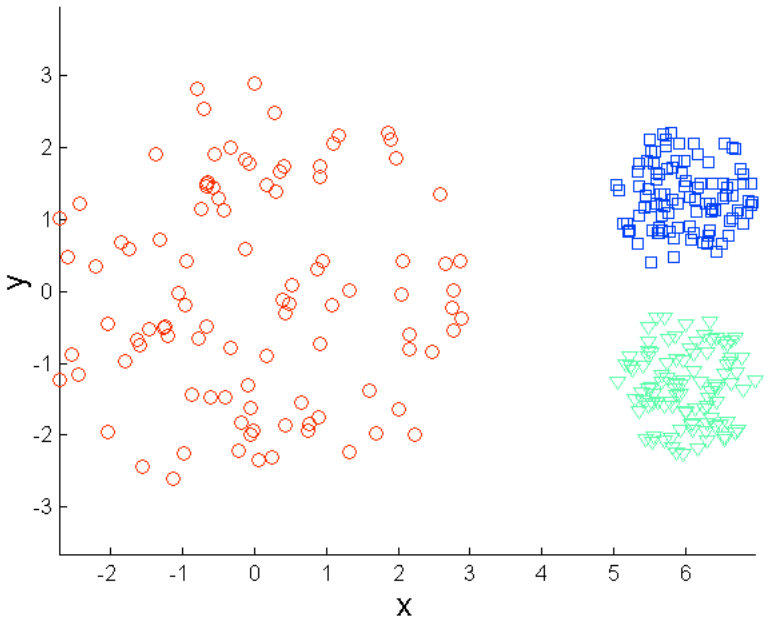


K-means Clusters

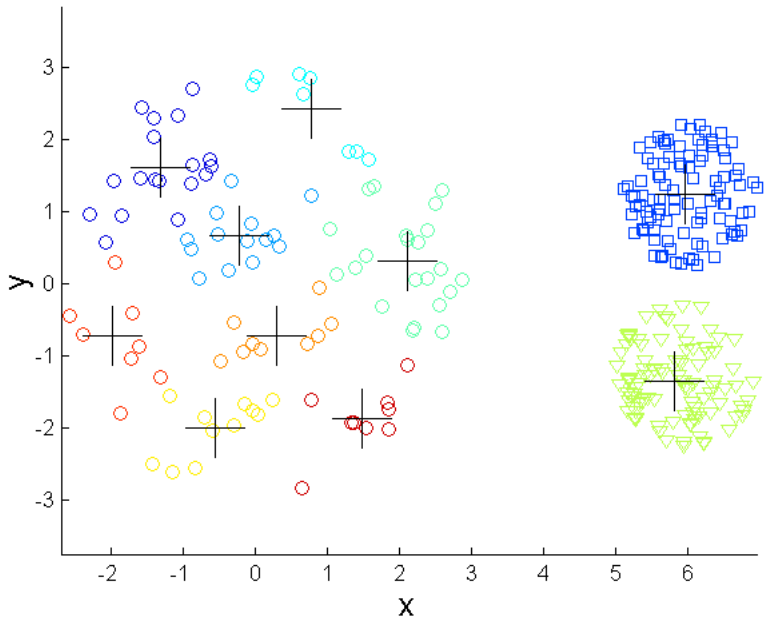
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

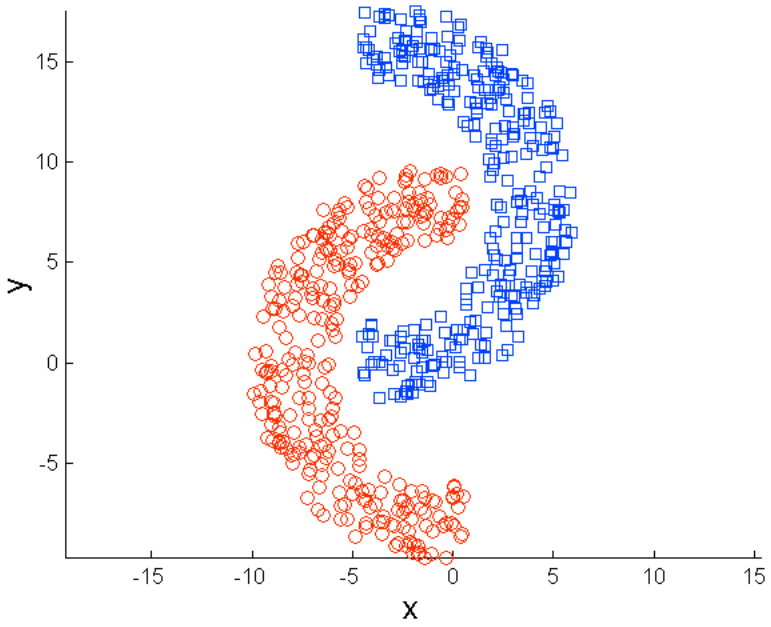


Original Points

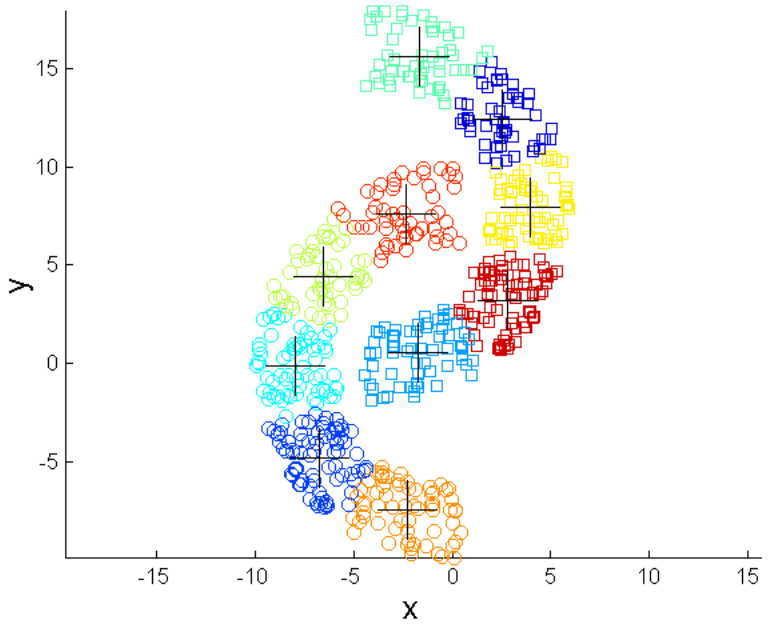


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Hierarchical Clustering

- Hierarchical clustering has two basic approaches:
  - Agglomerative: start with all points as a cluster, then at each iteration merge two clusters which are the closest → need to define cluster proximity
  - Divisive: start with only one cluster (consist of all points), then at each iteration split a cluster until only singleton clusters of individual points remain
- Agglomerative is the most common, often displayed graphically by a dendogram

BasicAgglomerative Hierarchical Clustering Algorithm

1. Compute the proximity matrix, if necessary
2. Repeat
3. merge the closes two clusters
4. update the proximity matrix to reflect the proximity between the new cluster and the original clusters
5. Until only one cluster remains

Proximity Measure in Agglomerative Hierarchical Clustering

- Single Link (MIN) → the minimum of the distance between any two points in the two different clusters
- Complete Link (MAX) → the maximum of the distance between any two points in the two different clusters
- Group Average → the average pairwise proximity among all pairs of points in the different clusters

$$proximity(C_i, C_j) = \frac{\sum_{\substack{x \in C_i \\ y \in C_j}} proximity(x, y)}{m_i * m_j}$$