Cluster Analysis

MATH2319

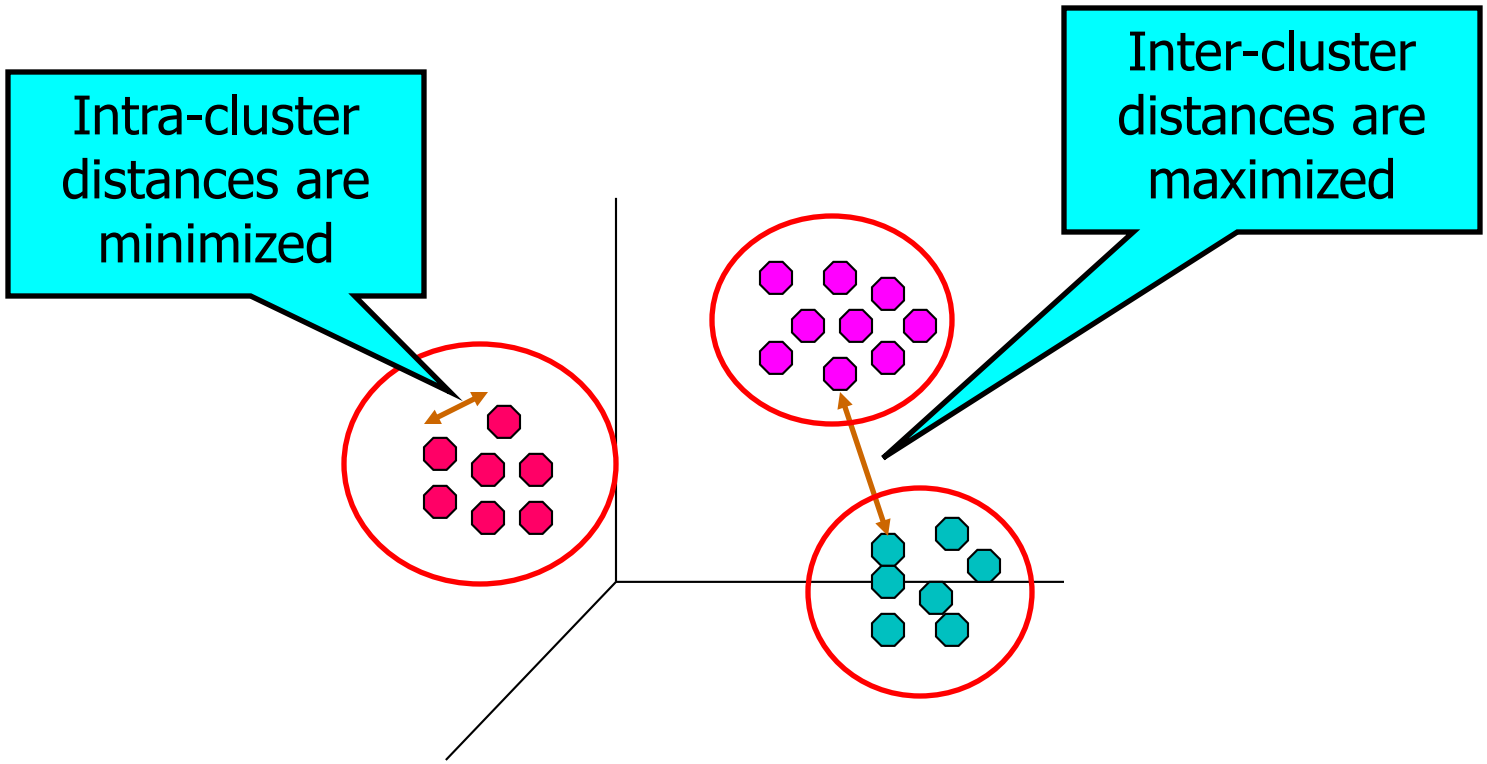
Reference:

“Introduction to Data Mining”

Tan, Steinbach, and Kumar

What is Cluster Analysis?

Finding groups of objects such that the objects in a group will be similar to one another and different from the objects in other groups.



Applications

- Group similar customers for a marketing campaign
- Group genes that have similar functions
- Usually clustering is applied **before** classification: you can cluster similar-looking countries, hotels, products, items, etc. to reduce dimensionality.
  - Suppose you are working on a classification problem involving hotel names as a descriptive feature.
  - Say there are 100K hotels. If you perform one-hot encoding, you will end up with 100K new features, a terrible idea due to curse of dimensionality.
  - In this case, you can first cluster hotels, say into 10 different groups. You can then perform one-hot encoding on these 10 groups and proceed with your project as usual.

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

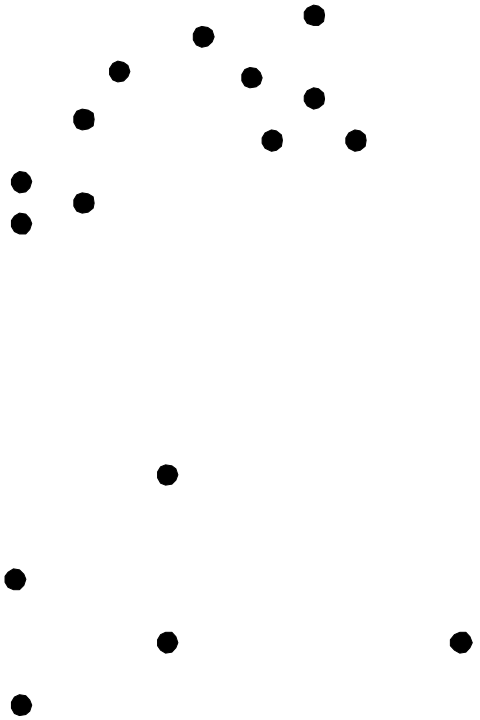
Two Clusters

Four Clusters

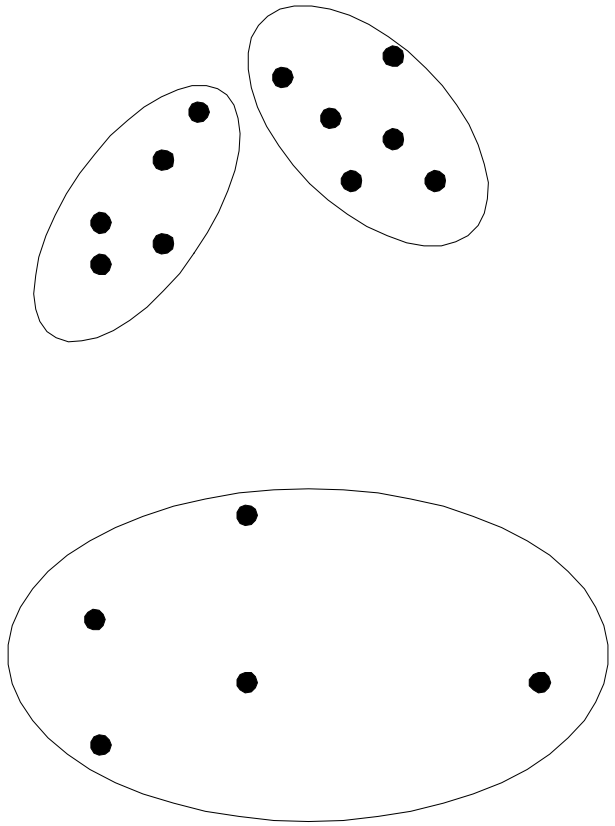
Types of Clusterings

- Two types of clustering: Hierarchical and partitional:
  - 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

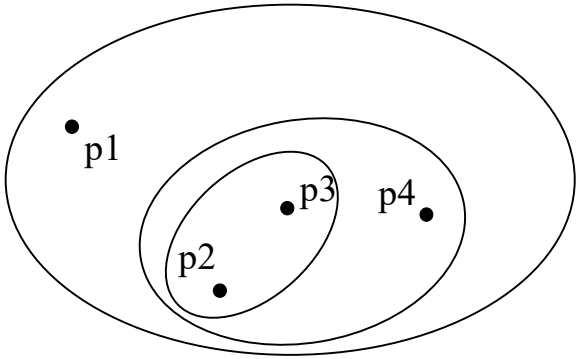


Original Points

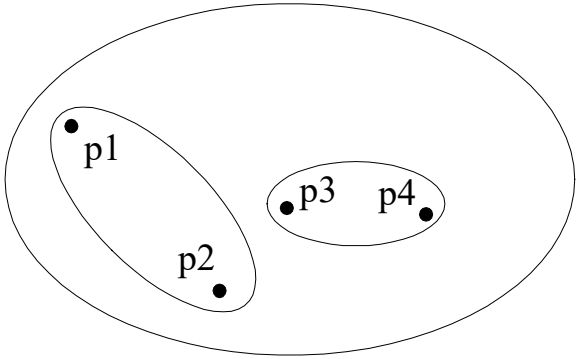


A Partitional Clustering

Hierarchical Clustering



Traditional Hierarchical Clustering



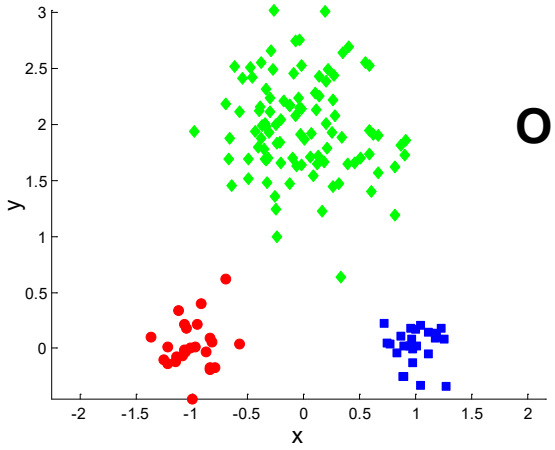
Non-traditional Hierarchical Clustering

An Algorithm for Clustering: K-means

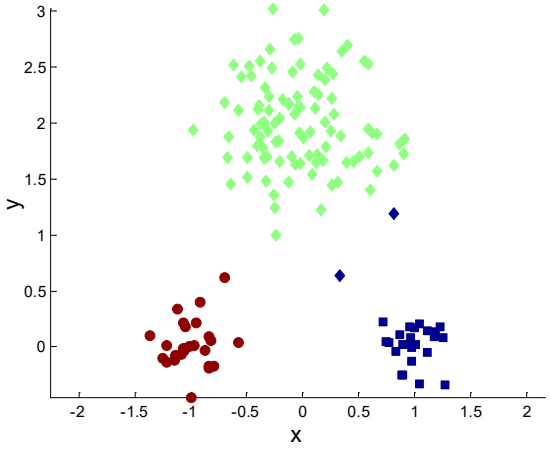
- Partitional clustering approach
- Each cluster is associated with a ***centroid*** (center point – mean of all observations in the cluster)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, *K*, must be specified ahead of time
- The basic algorithm is simple:

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

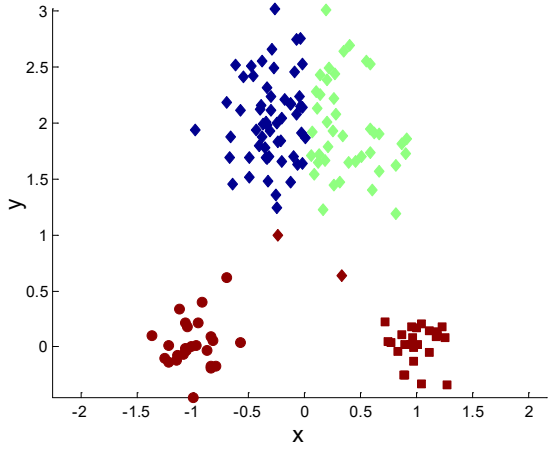
Two different K-means Clusterings



Original Points

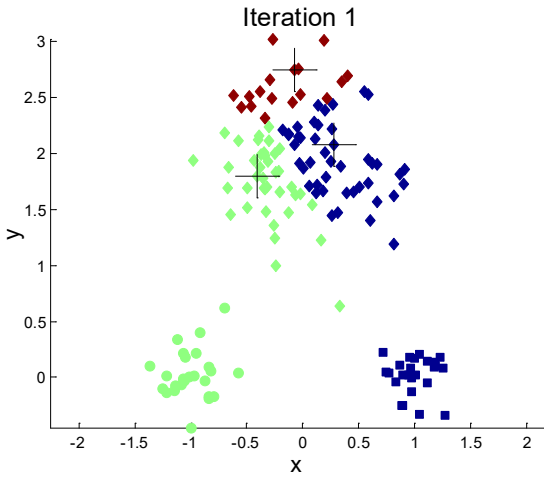


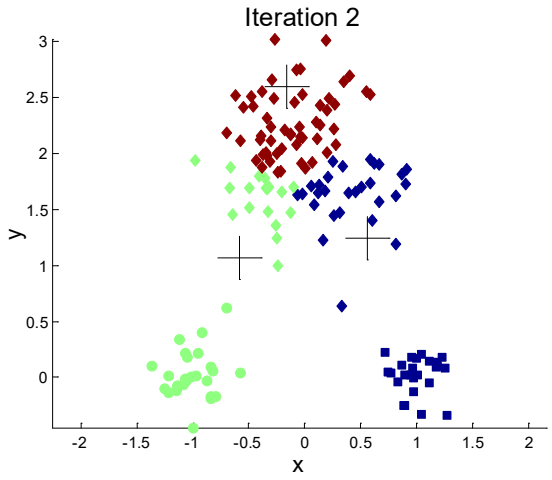
Optimal Clustering

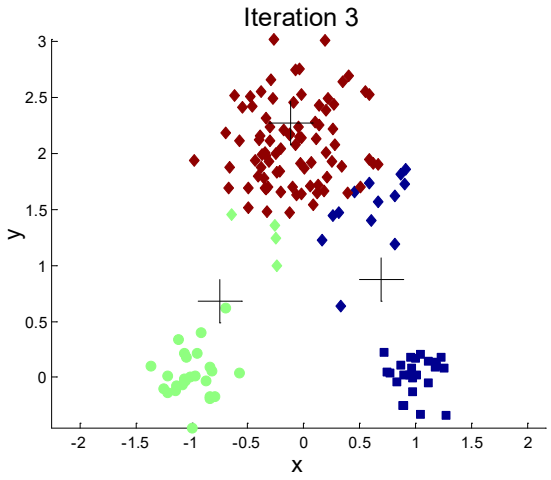


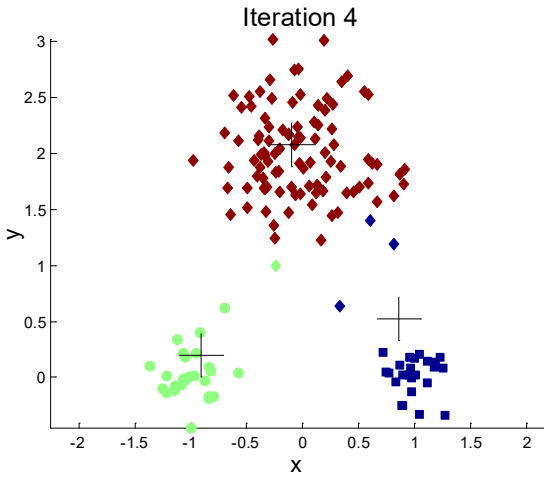
Sub-optimal Clustering

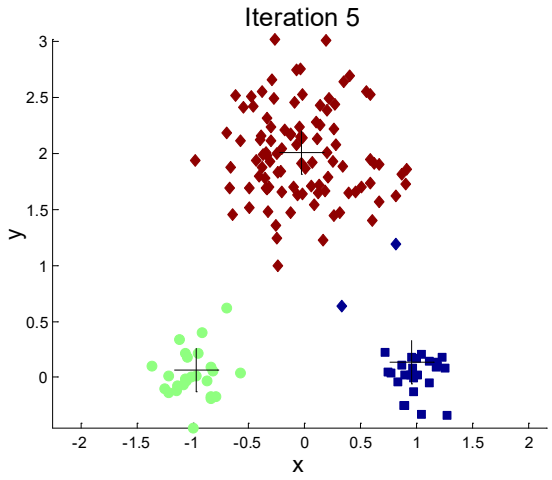
Importance of Choosing Initial Centroids

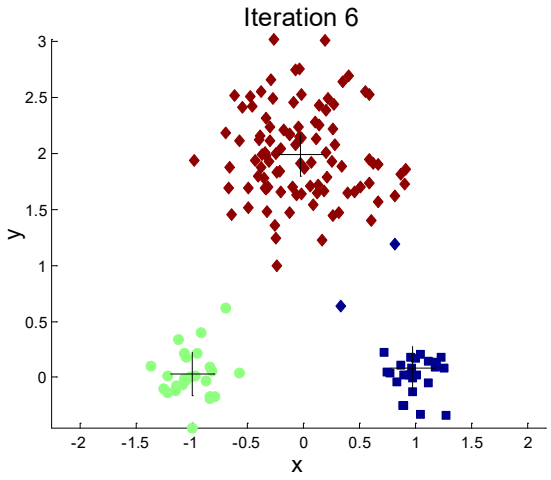




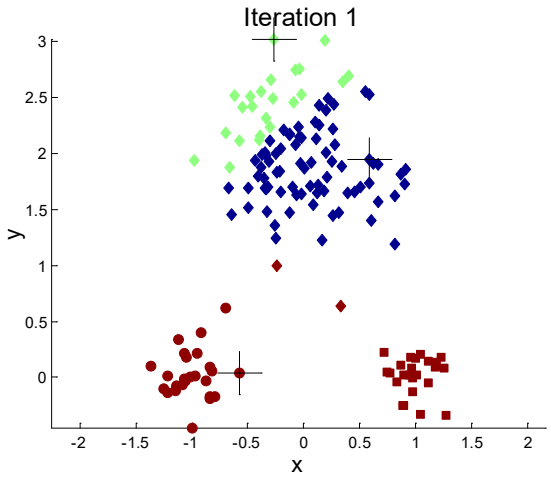


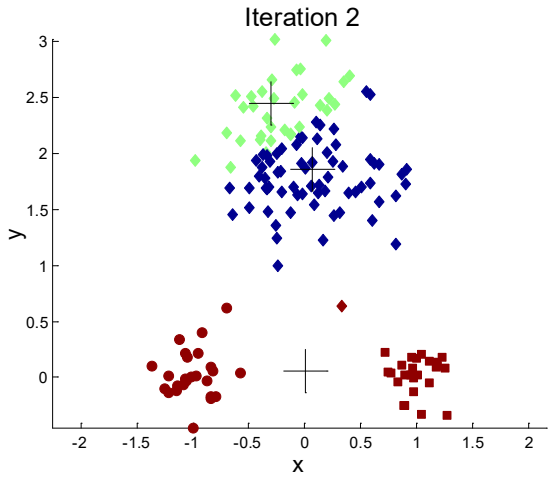


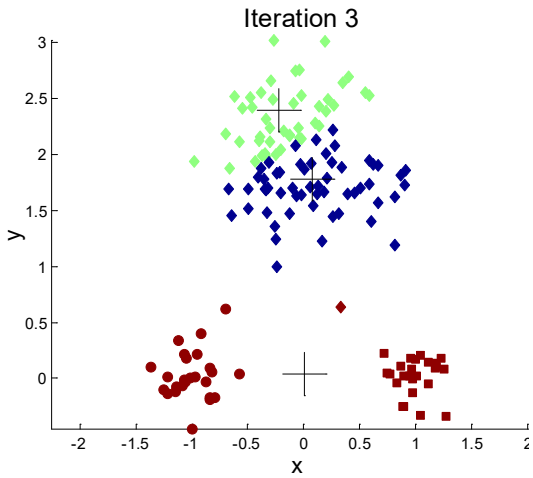


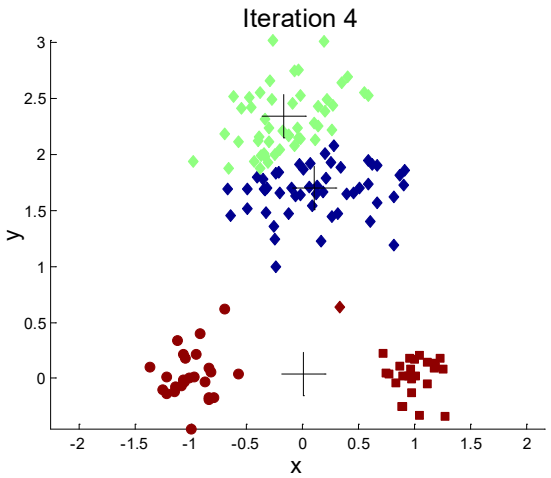


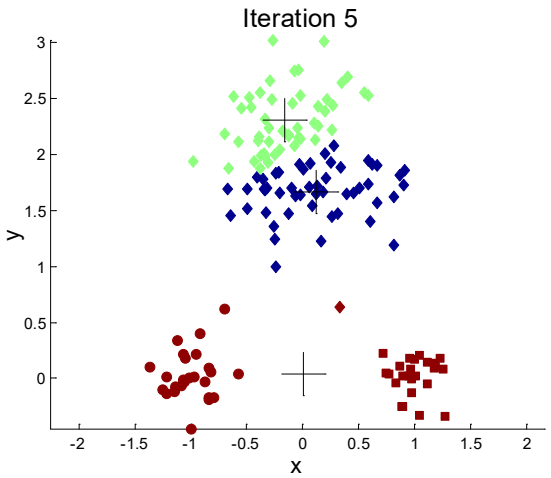
Importance of Choosing Initial Centroids ...











K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- ‘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’
- Classification of a new observation:
  - Save cluster centroids
  - Assign new observation to the cluster it is closest to

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 centroid
  - 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 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
    - **Watch out:** A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

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

Pre-processing and Post-processing

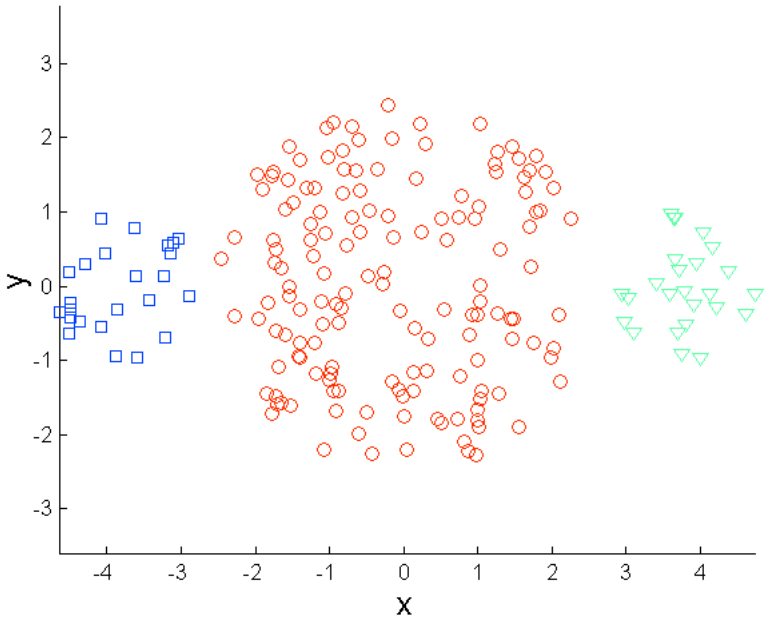
- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

Limitations of K-means

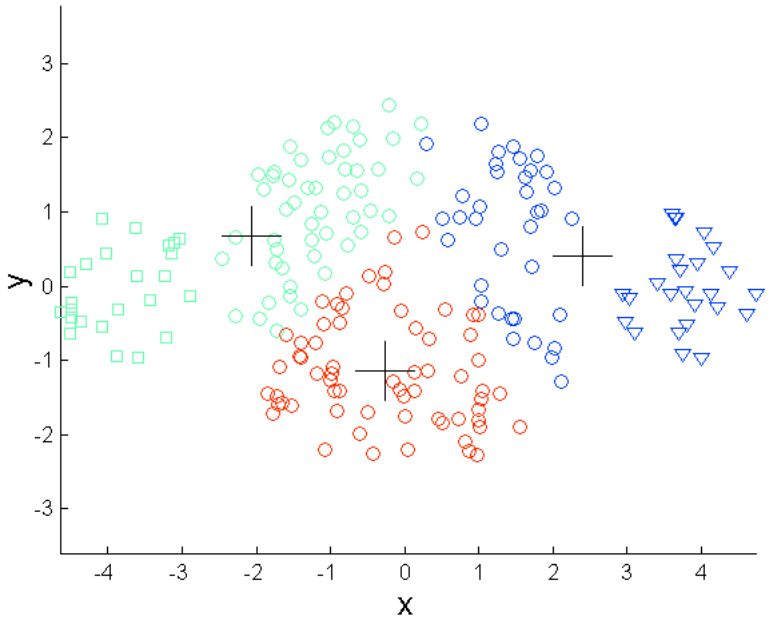
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

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

Limitations of K-means: Differing Sizes

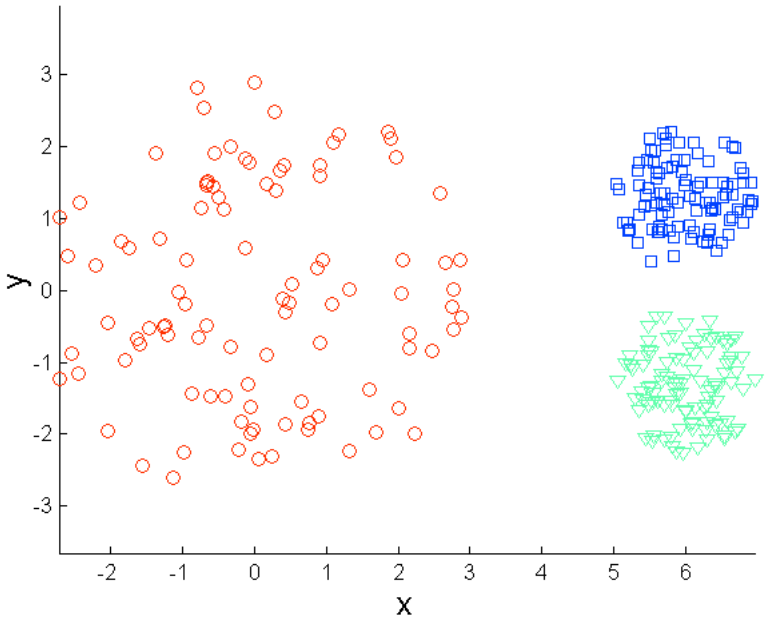


Original Points

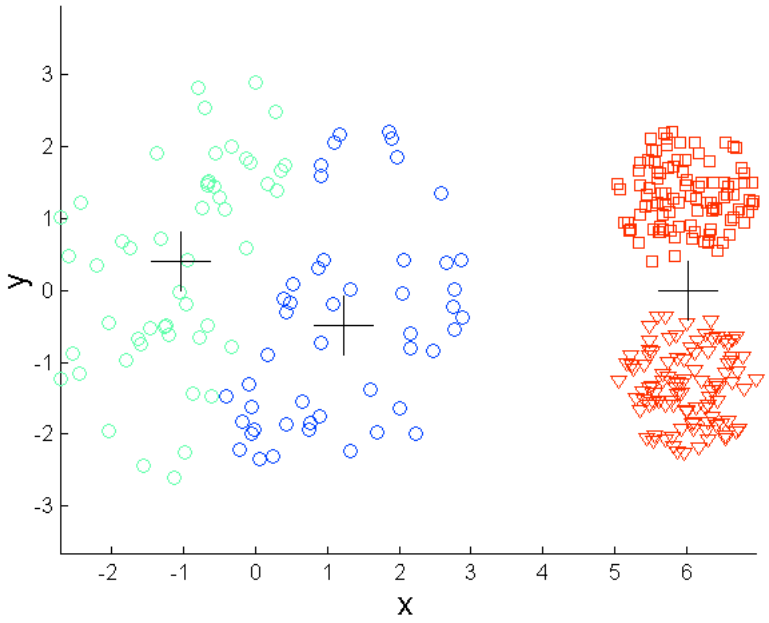


K-means (3 Clusters)

Limitations of K-means: Differing Density

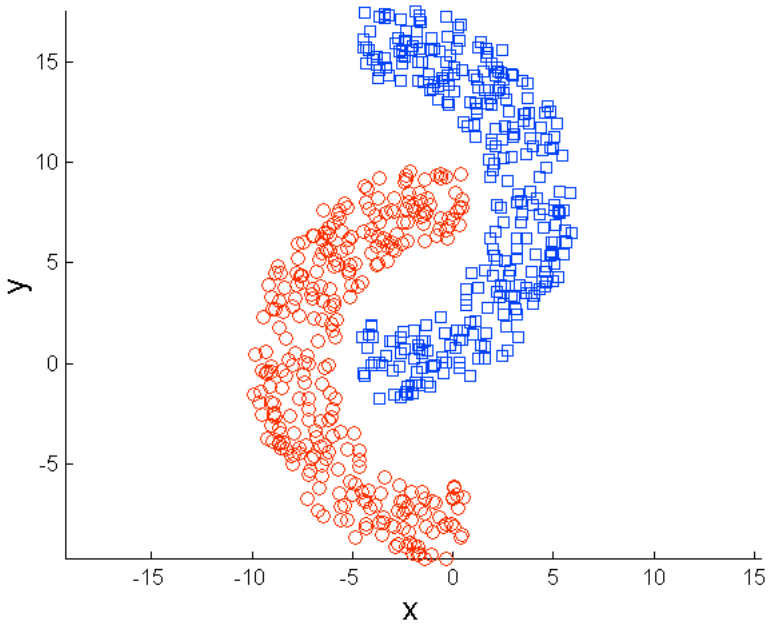


Original Points

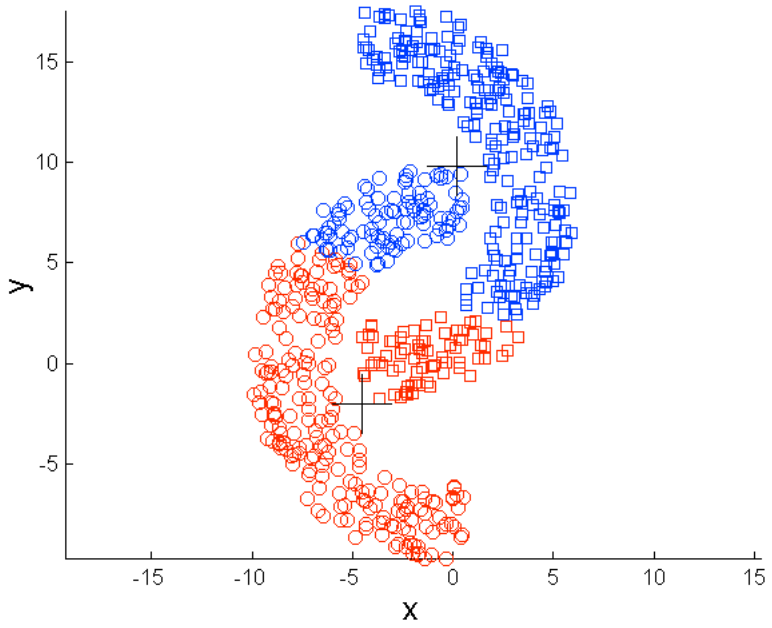


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

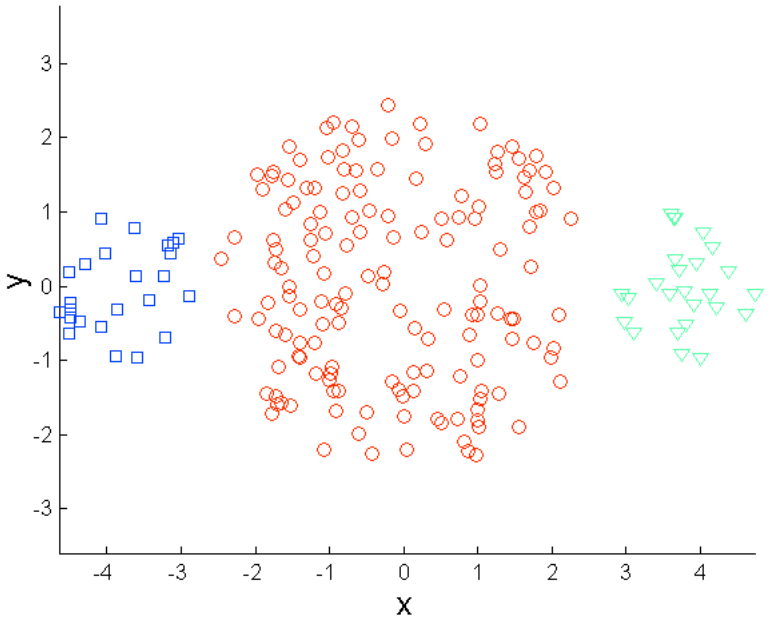


Original Points

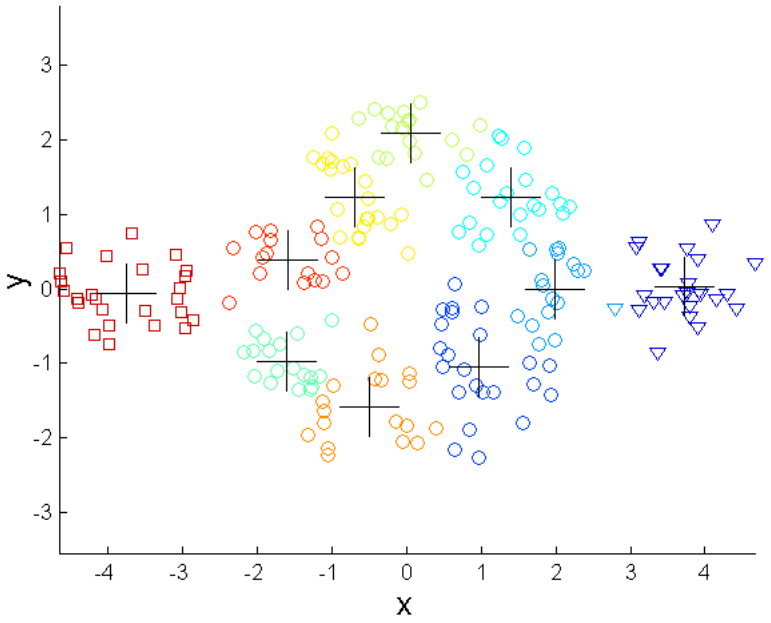


K-means (2 Clusters)

Overcoming K-means Limitations



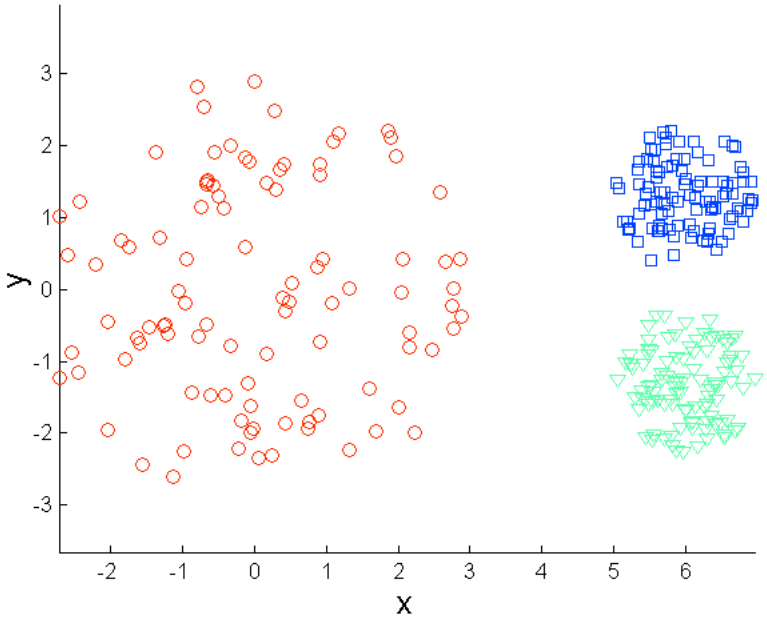
Original Points



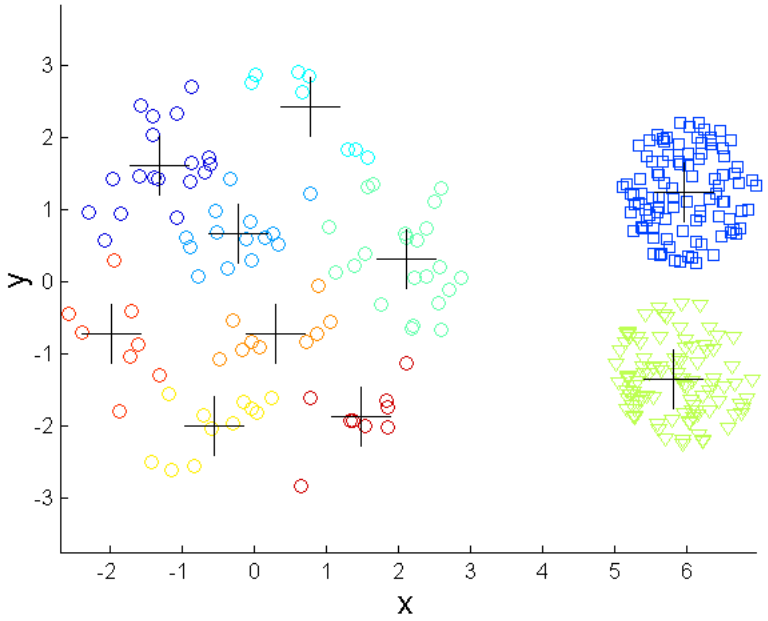
K-means Clusters

One solution is to use many clusters. You can find parts of clusters, but then you need to put them together.

Overcoming K-means Limitations

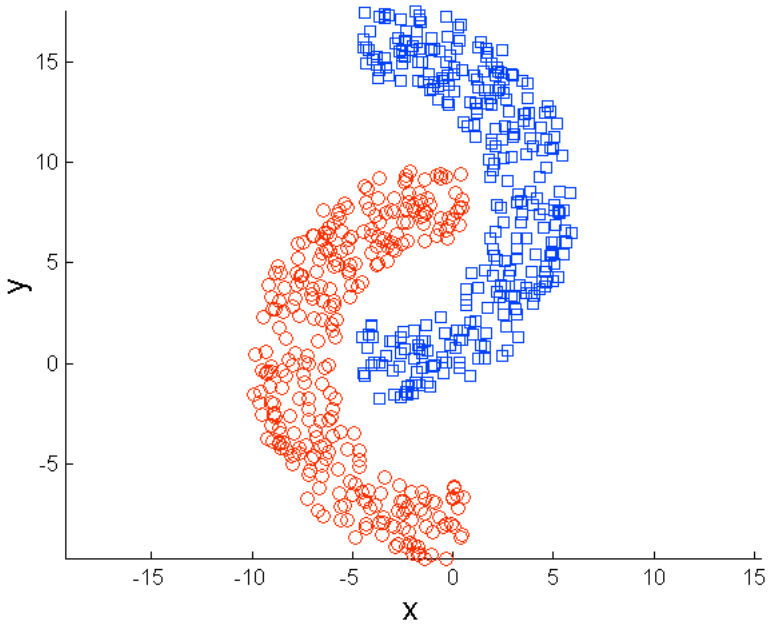


Original Points

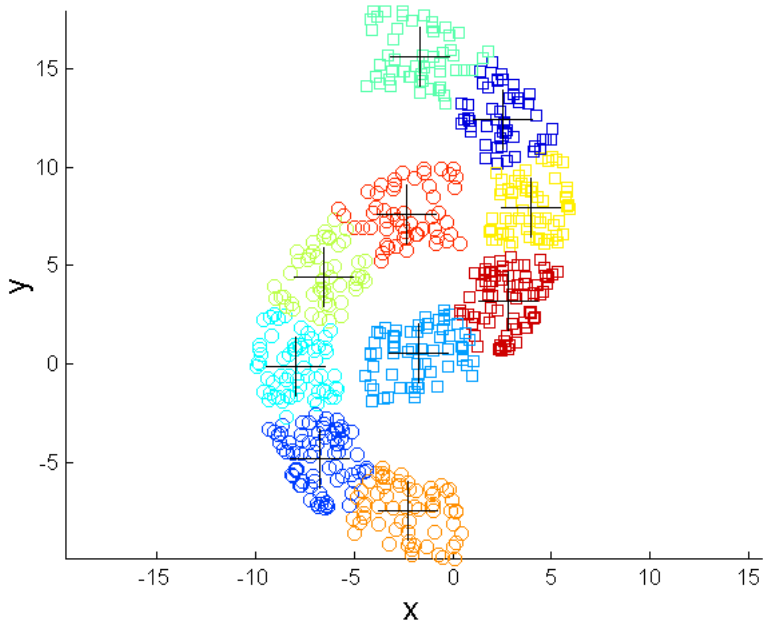


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters