School of Computing
UNIVERSITY OF GEORGIA

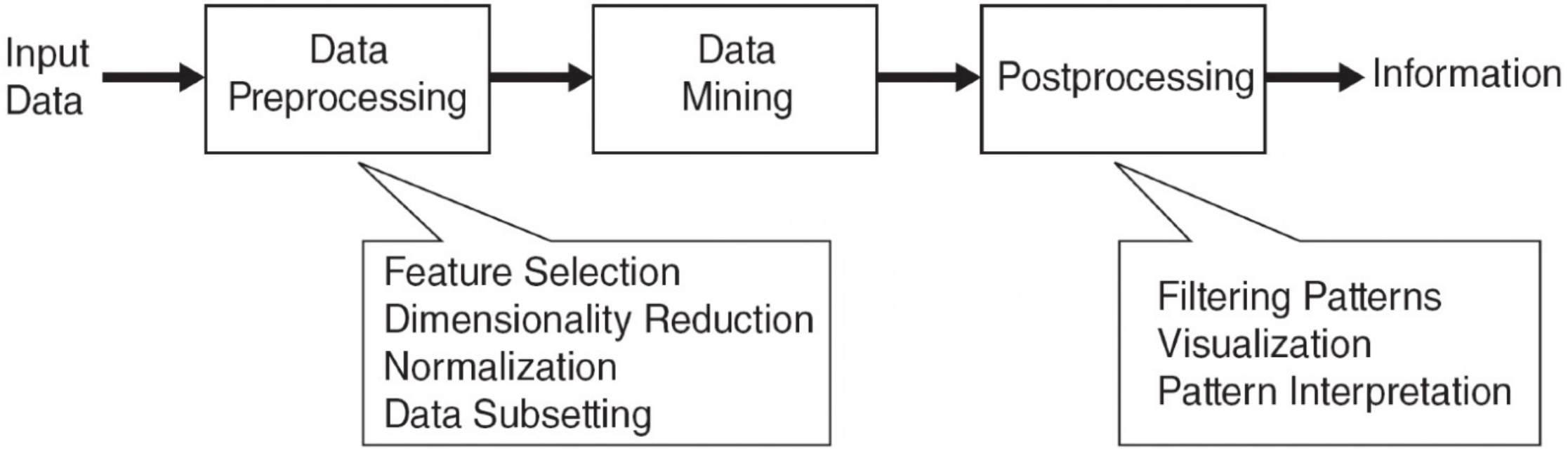
CSCI 4380/6380 DATA MINING

Fei Dou

Assistant Professor
School of Computing
University of Georgia

October 19, 24, 2023

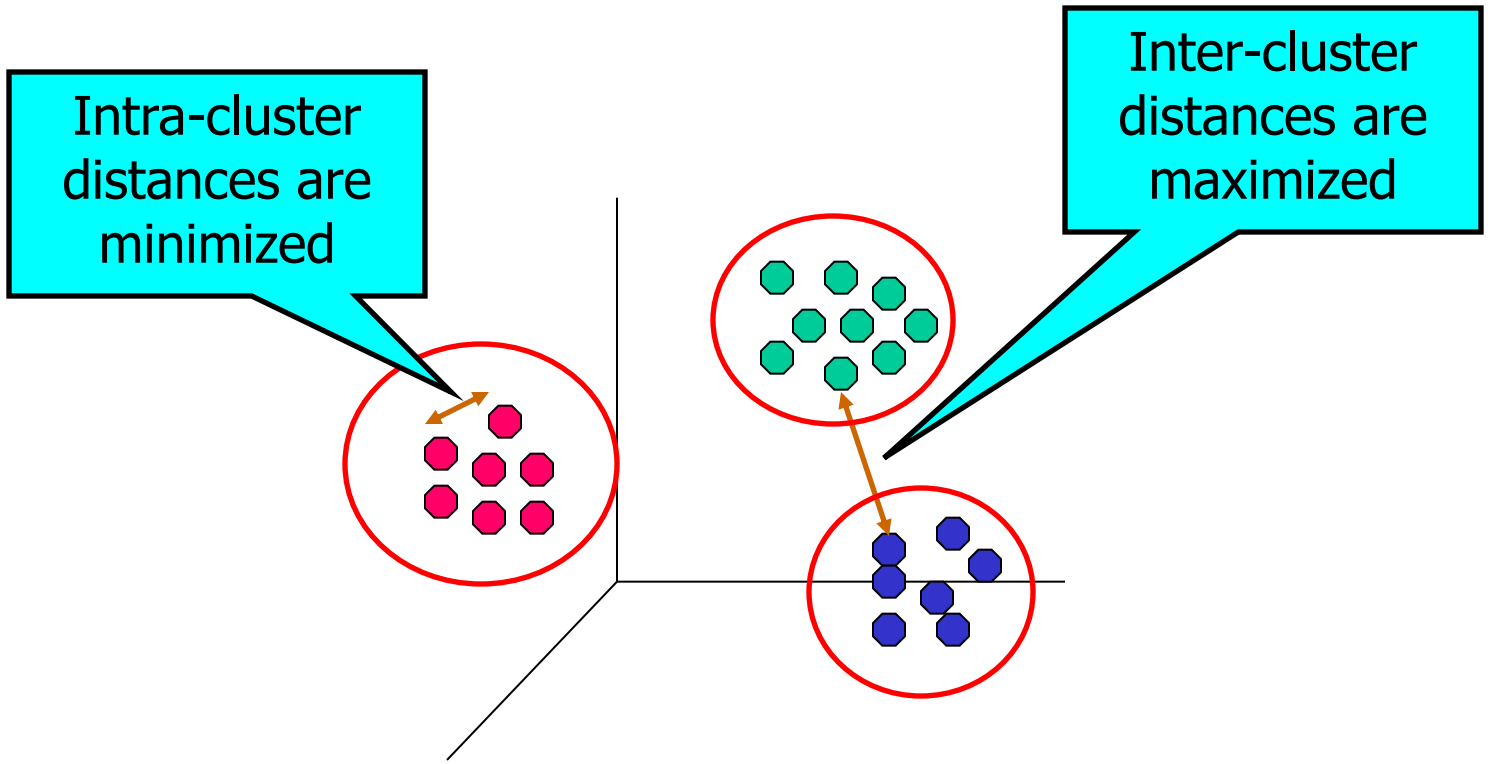
Recap: Data Mining Process



Clustering - Concept

What is Cluster Analysis?

- Given a set of objects, place them in groups such that the objects in a group are similar (or related) to one another and different from (or unrelated to) the objects in other groups



Cluster Analysis - Motivation

- Pattern (motif) discovery
- Explore data distribution organize database
- Preprocessing (summarization) step for more complex applications, rule discovery, indexing, classification, and anomaly detection
- Visualization and interpretation

Clustering - Applications

- **Energy consumption:** group consumers based on their energy consumption patterns.
- **Gene-based clustering** :group genes based on similar expression patterns.
- **Finance**: finding seasonality patterns (retail patterns); discovery patterns from stock time-series.
- **Healthcare**: detecting brain activity using Electroencephalography (EEG) data.
- **Climate**: understanding earth climate, find patterns of atmospheric and ocean.
- **Land use**: Identification of areas of similar land use in an earth observation database.

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

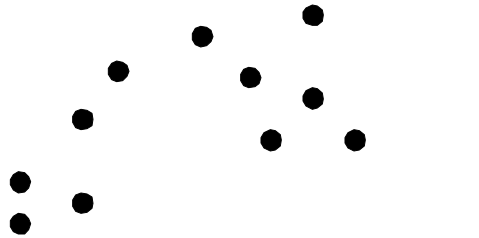
Two Clusters

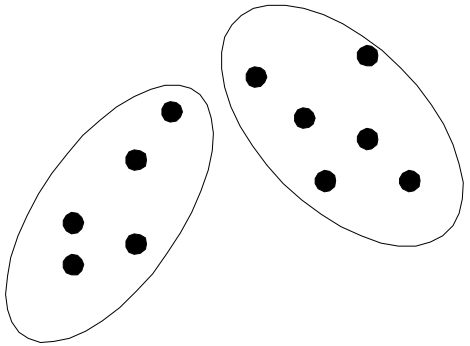
Four Clusters

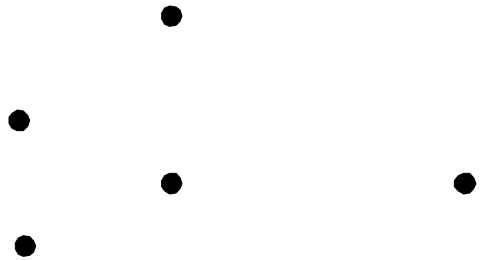
Types of Clusterings

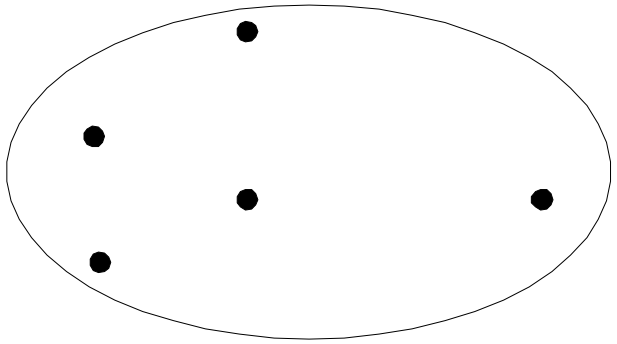
- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
  - Partitional Clustering
    - A division of data objects into non-overlapping subsets (clusters)
  - Hierarchical clustering
    - A set of nested clusters organized as a hierarchical tree

Partitional Clustering





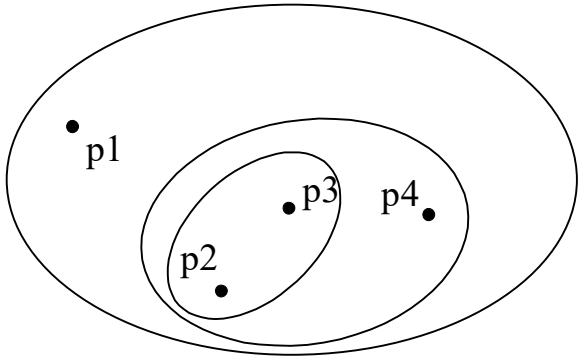




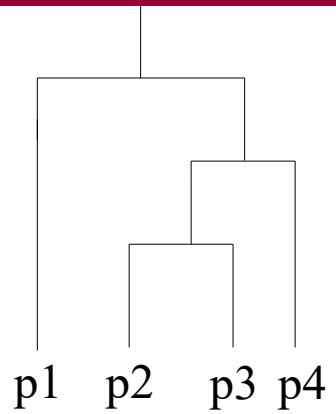
Original Points

A Partitional Clustering

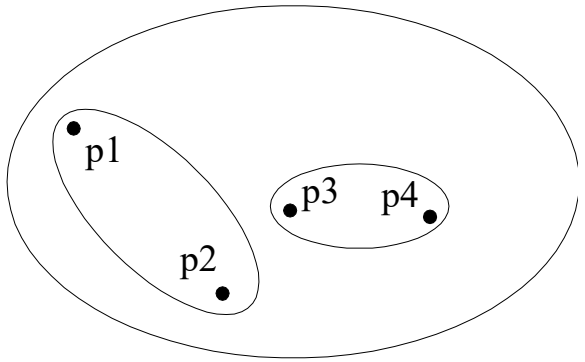
Hierarchical Clustering



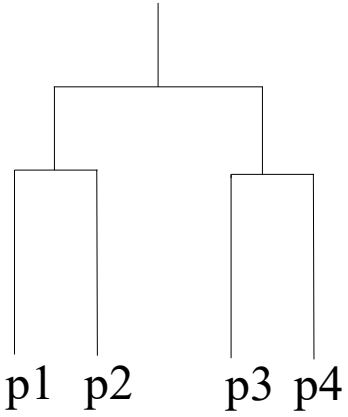
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

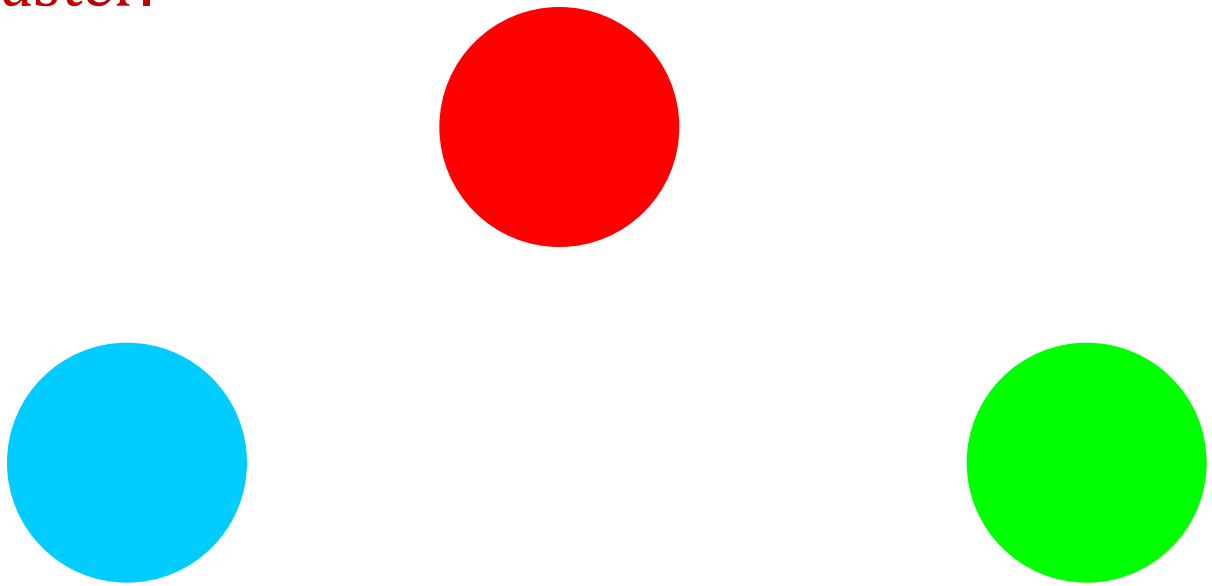
- Exclusive versus non-exclusive
  - In non-exclusive clusterings, points may belong to multiple clusters.
    - Can belong to multiple classes or could be 'border' points
  - Fuzzy clustering (one type of non-exclusive)
    - In fuzzy clustering, a point belongs to every cluster with some weight between 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

Types of Clusters

- Well-separated clusters
- Prototype-based clusters
- Contiguity-based clusters
- Density-based clusters
- Described by an Objective Function

Types of Clusters: Well-Separated

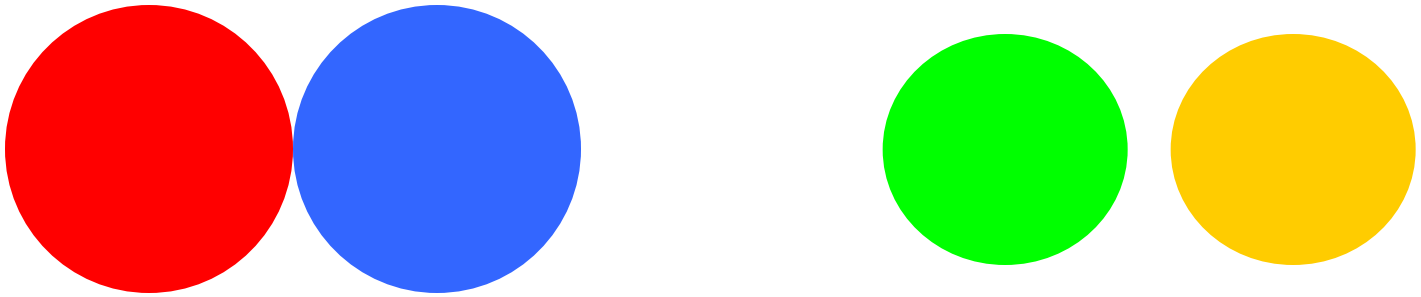
- Well-Separated Clusters:
  - A cluster is a set of points such that any point in a cluster is closer (or more similar) to every other point in the cluster than to any point not in the cluster.



3 well-separated clusters

Types of Clusters: Prototype-Based

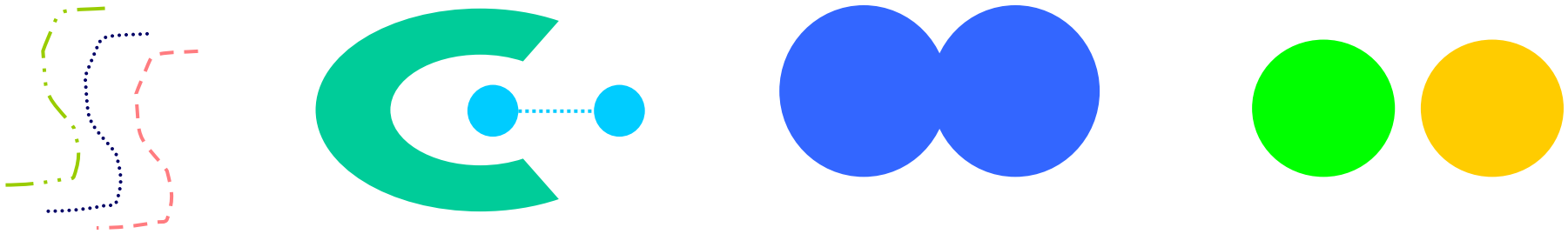
- Prototype-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the prototype or "center" of a cluster, than to the center of any other cluster
  - The center of a cluster is often a centroid, the average of all the points in the cluster, or a medoid,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



4 center-based clusters

Types of Clusters: Contiguity-Based

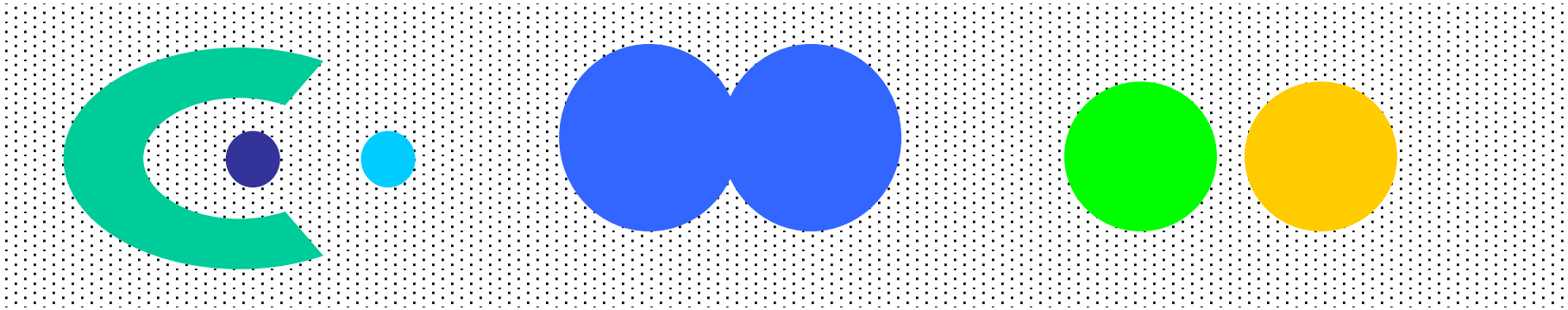
- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



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: 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.

Clustering - Two Important Aspects

- **Properties of input data**
  - Define the similarity or dissimilarity between points.

- **Requirement of clustering**
  - Objective of clustering Clustering evaluation

Clustering - Requirements

- Scalability
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Discovery of clusters with arbitrary shape
- High dimensionality
- Interpretability and usability

Clustering - Methods

- Partitional
- Hierarchical
- Density-based
- Mixture model
- Spectral methods

K-Means Clustering

Clustering - K-means

K-means clustering algorithm

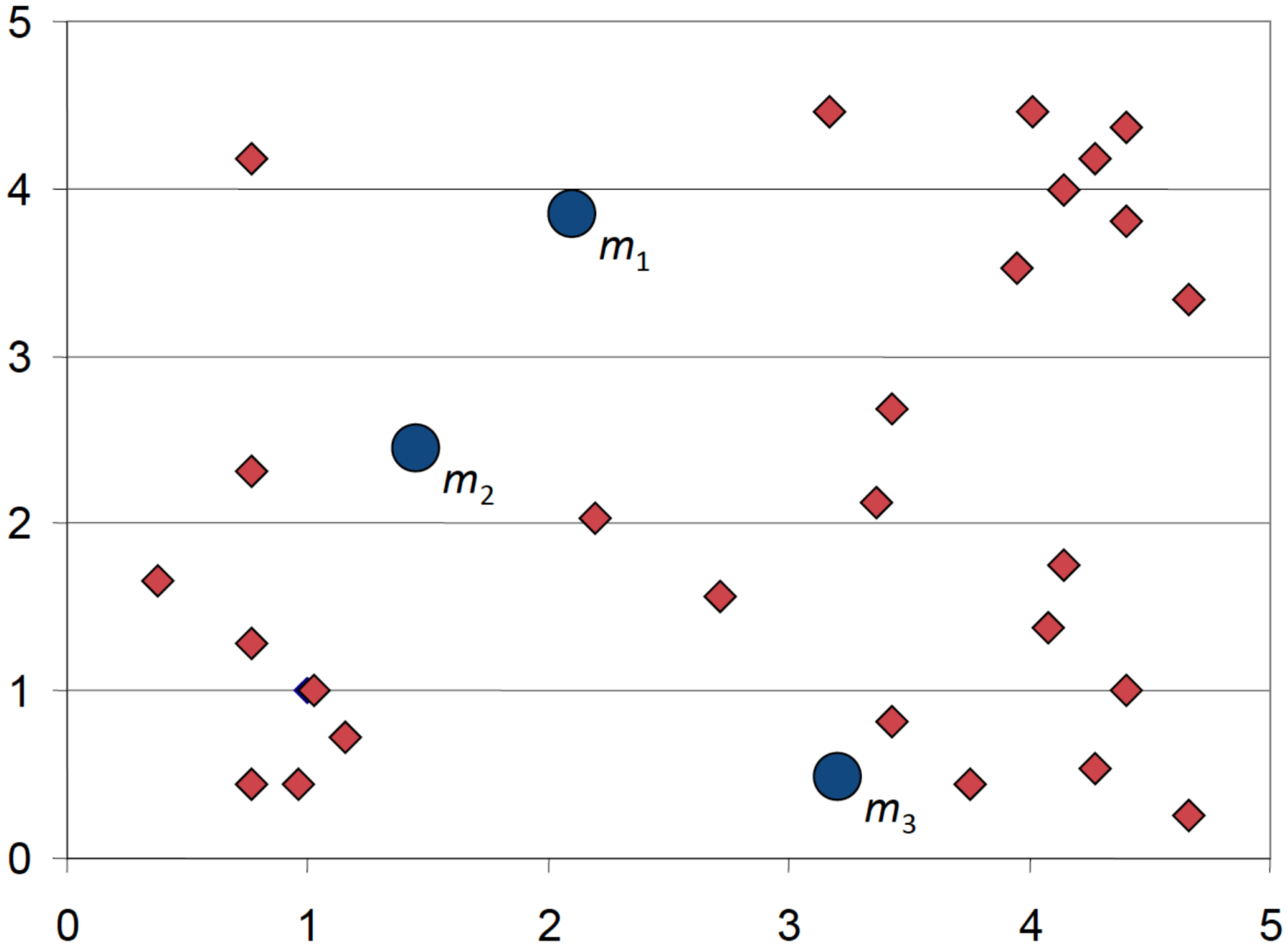
- **Objective**: Partition samples $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ into K clusters.
- **Initialization**: specify the K initial cluster centers (centroids)
  Iteration until no change (centroids)
  - For each $\mathbf{x}i$,
    - Calculate the distances between $\mathbf{x}i$ and K centroids;
    - (Re)assign $\mathbf{x}i$ to the cluster whose centroids is the closest to $\mathbf{x}i$
  - Update the cluster centroids based on current assignment

K-means Clustering

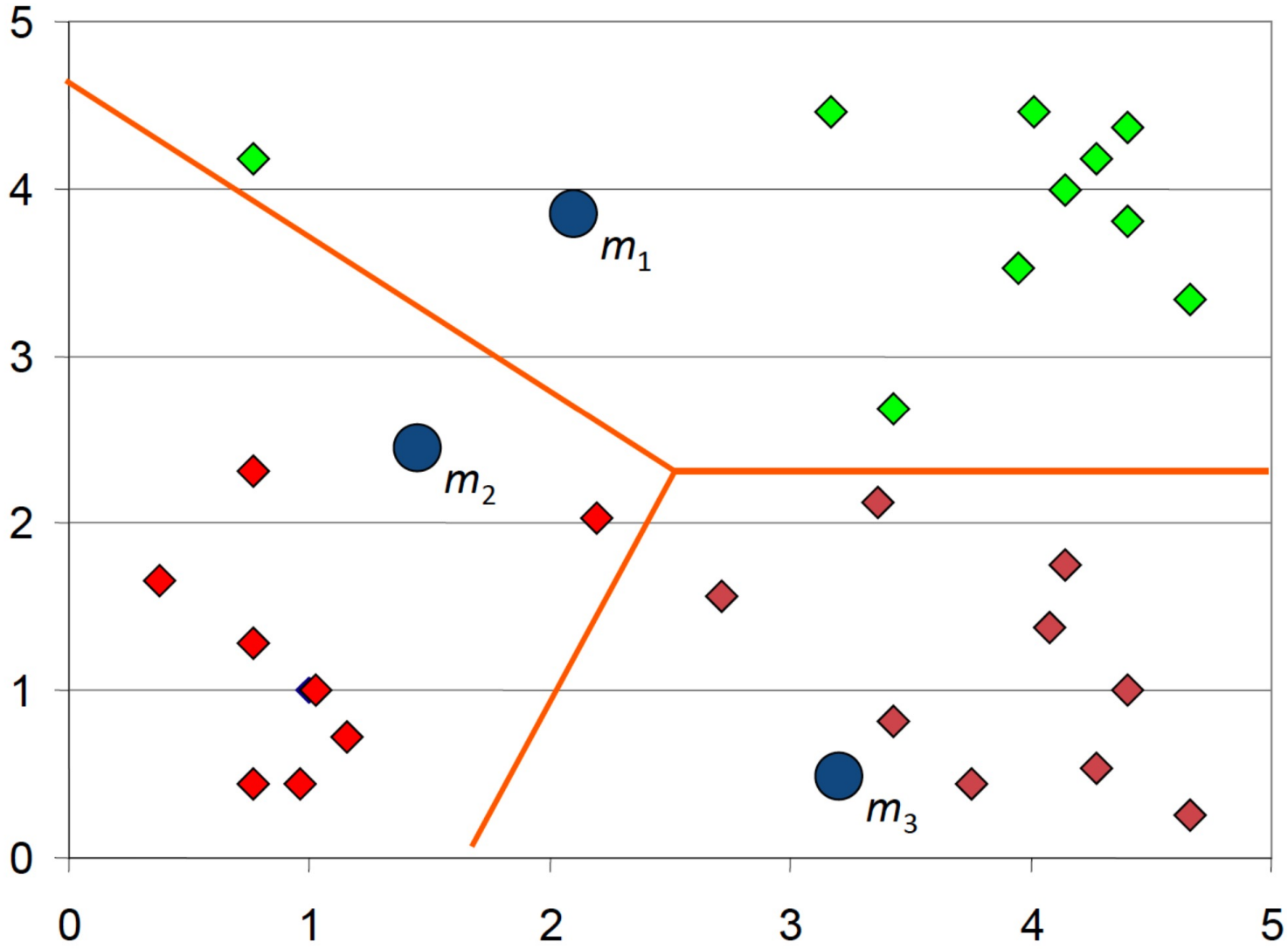
- Partitional clustering approach
- Number of clusters, K, must be specified
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- The basic algorithm is very simple

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

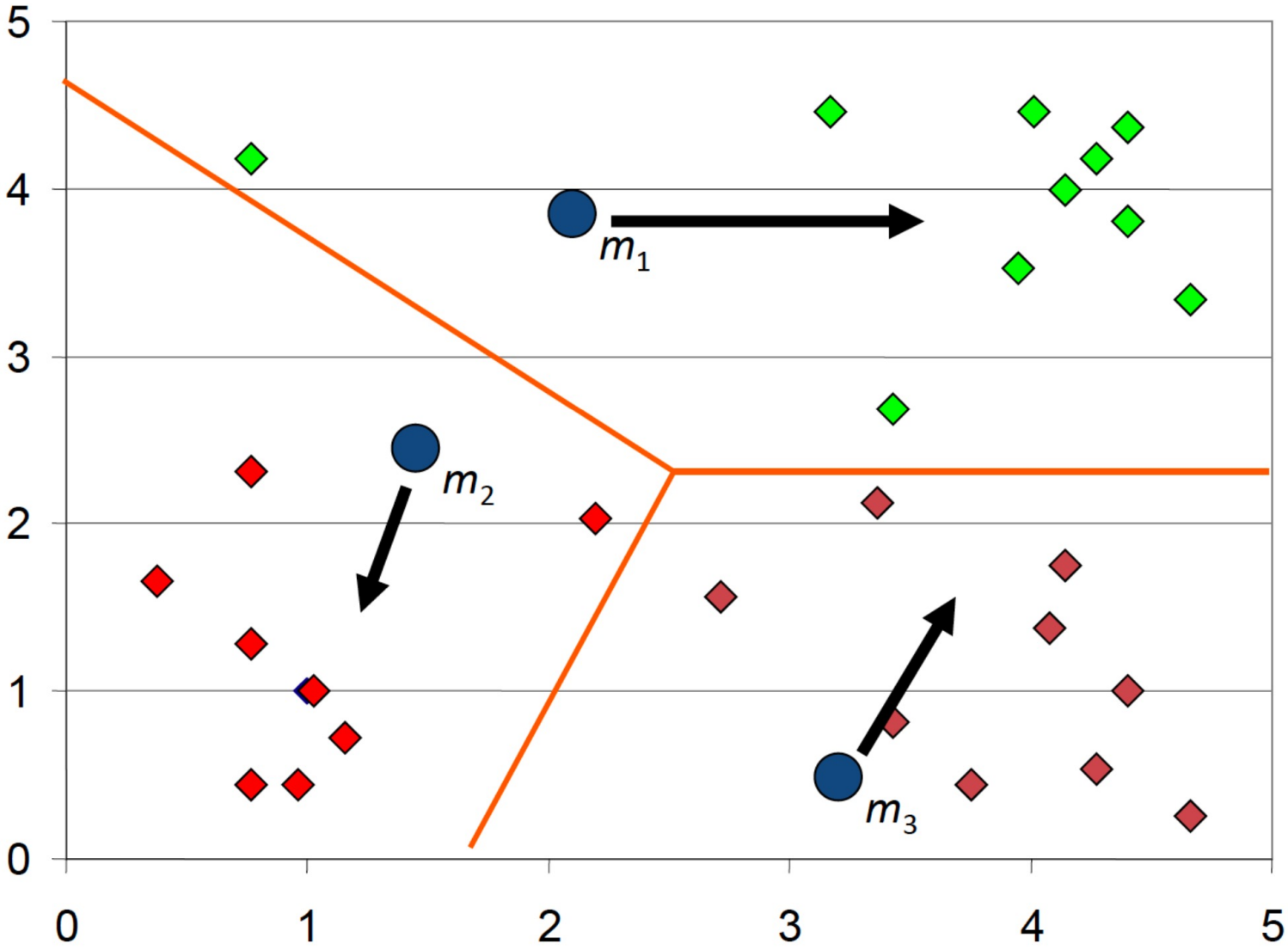
Clustering - K-means



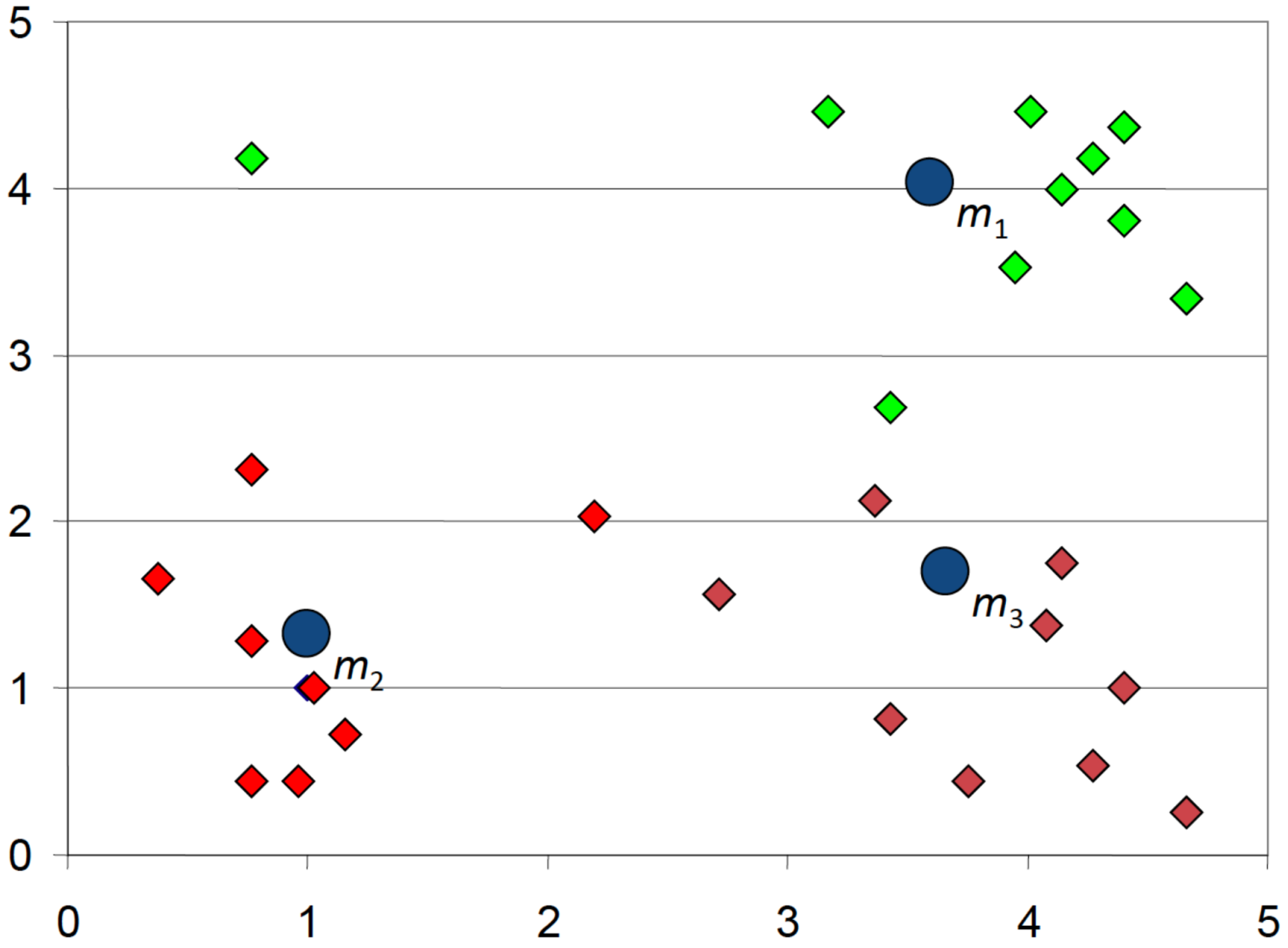
Clustering - K-means



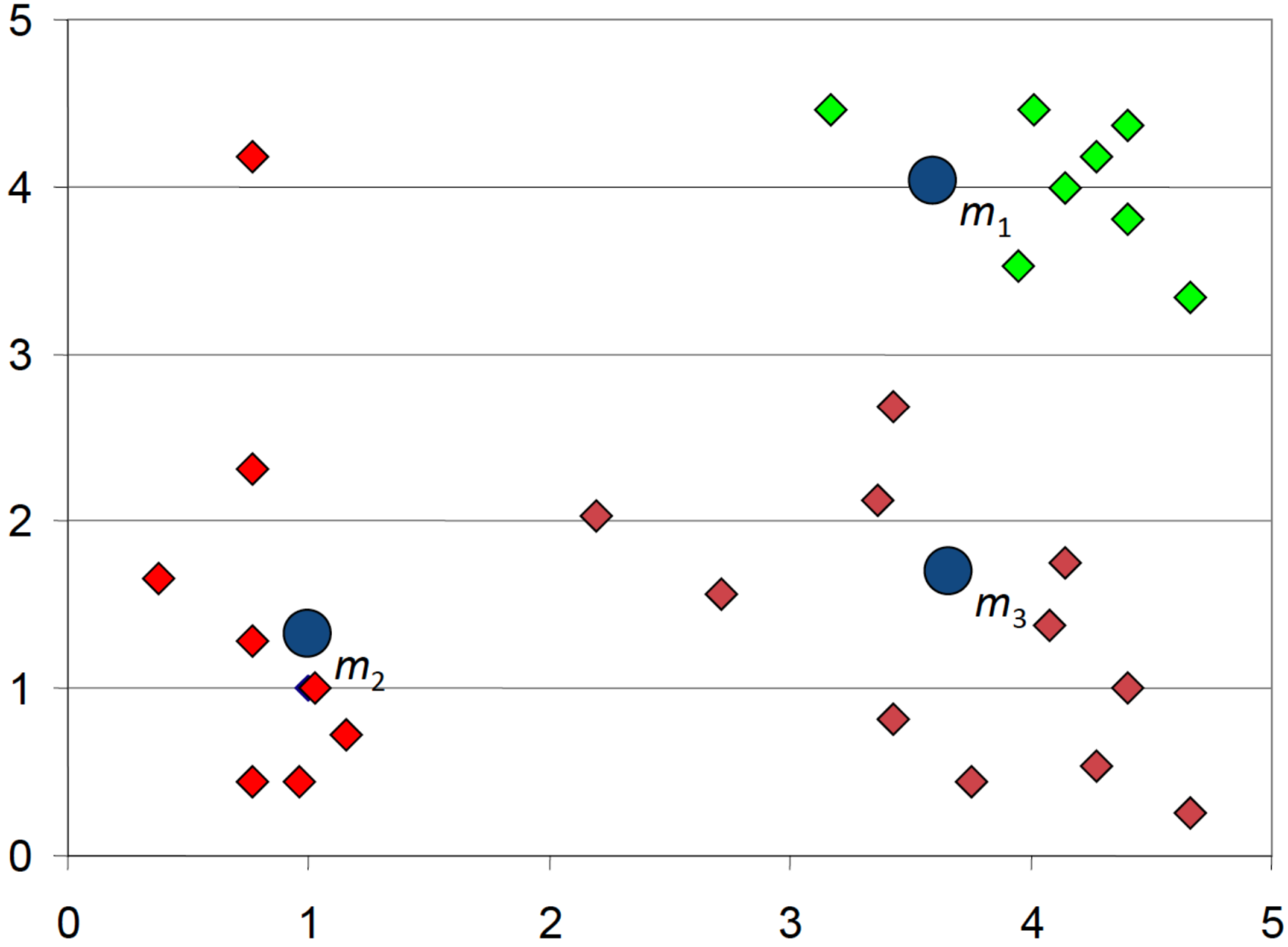
Clustering - K-means



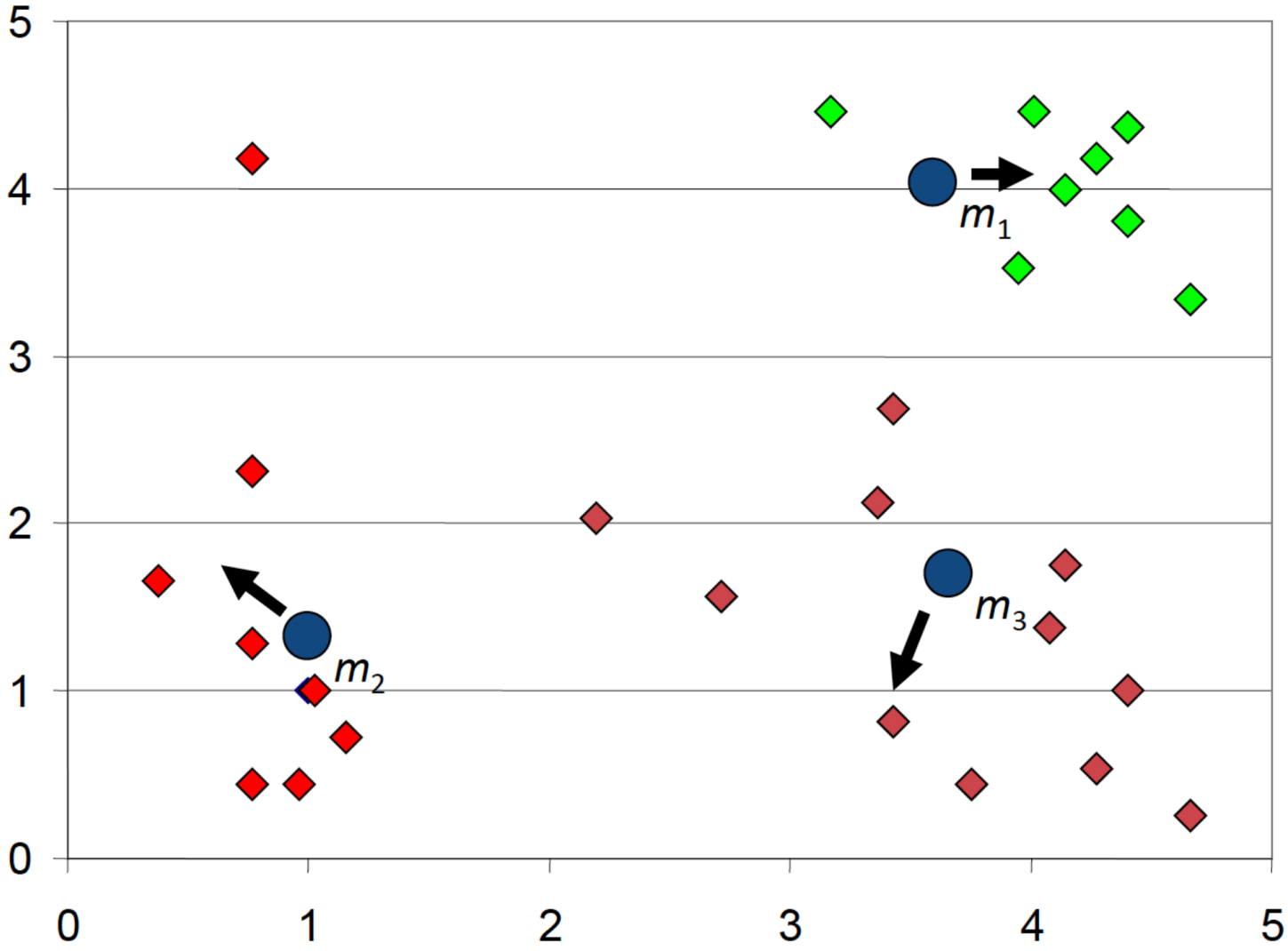
Clustering - K-means



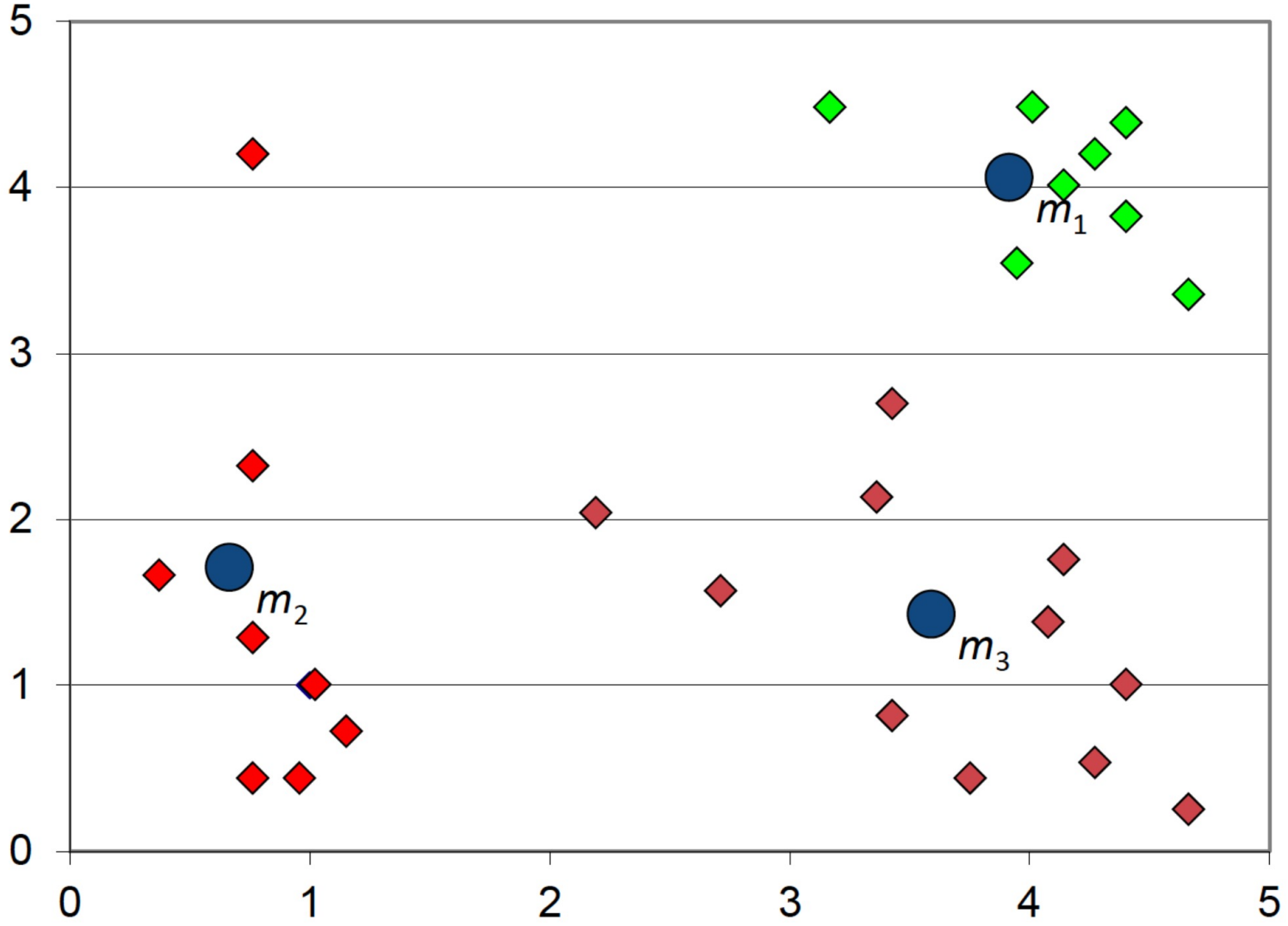
Clustering - K-means



Clustering - K-means



Clustering - K-means



K-means Objective Function

- A common objective function (used with Euclidean distance measure) is Sum of Squared Error (SSE)
  - Suppose the centriod of cluster C_j is m_j
  - For each sample, the error is the distance to the nearest cluster center
  - 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 centroid (mean) for cluster C_i
  - SSE improves in each iteration of K-means until it reaches a local or global minima.

How to minimize SSE

$$min \sum_{j} \sum_{x \in C_j} \|x - m_j\|^2$$

- **Two sets of variables to minimize**
  - Each sample **x** belongs to which cluster, i.e., $\mathbf{x} \in C_j$
  - Cluster centroid $\mathbf{m}_j$
- **Block coordinate descent**
  - Fix the cluster centroid—find cluster assignment that minimizes the current error
  - Fix the cluster assignment—compute the cluster centroids that minimize the current error

Cluster Assignment

$$min \sum_{j} \sum_{x \in C_j} \|x - m_j\|^2$$

- Cluster centroid $\mathbf{m}_j$ is known
- For each sample
  - Choose C_j among all clusters for $\mathbf{x}$ such that the distance between $\mathbf{x}$ and $\mathbf{m}_j$ is minimum.
  - Choose other clusters will result in bigger error
- Minimize error on each sample will minimize the SSE

Cluster Centroid Update

$$min \sum_{j} \sum_{x \in C_j} \|x - m_j\|^2$$

- For each cluster
  - Choose cluster centriod $\mathbf{m}_j$ as the center of the points

$$m_j = \frac{\sum_{x \in C_j} x}{|C_j|}$$

- Minimize error on each cluster will minimize the SSE

Clustering - K-means

- **Strength**
  - Complexity: O($mKnd$), where n is # of samples, K is # clusters, m is # of iterations, d is the dimension. Normally, K , $m << n$
  - Easy to implement
- **Issues**
  - Need to specify k, the number of clusters
  - Local minimum– Initialization matters
  - Empty clusters may appear
  - Sensitive to outliers

Clustering - K-means: Initialization

- Multiple runs
  - Average the results or choose the one that has the smallest SSE
- 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 (K-means++)
- Mean splitting

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

Clustering - K-means: Empty clusters

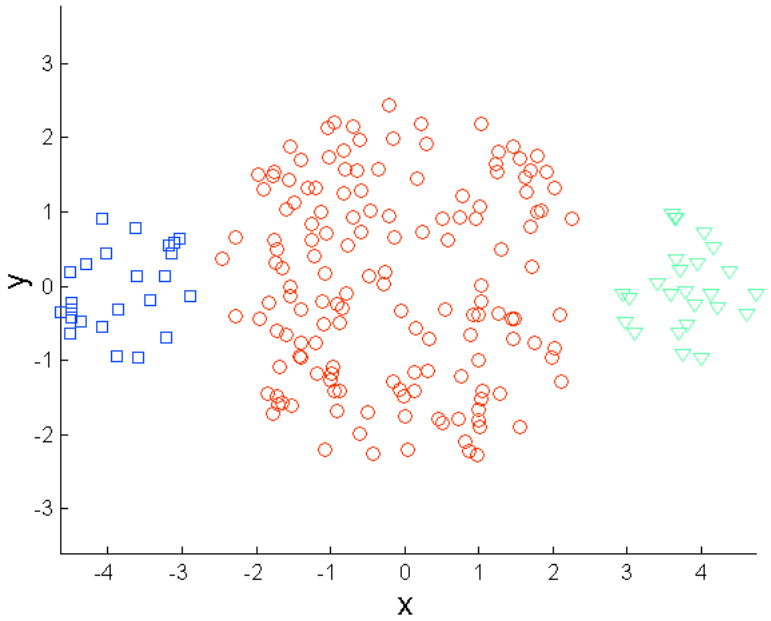
- Basic K-means algorithm can yield empty clusters
- At the end of each iteration of K-means
  - check the number of elements in each cluster
  - if too low, throw the cluster away
  - reinitialize the mean with a perturbed version of the populated cluster
- Other strategies
  - Choose the point that 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

Limitations of K-means

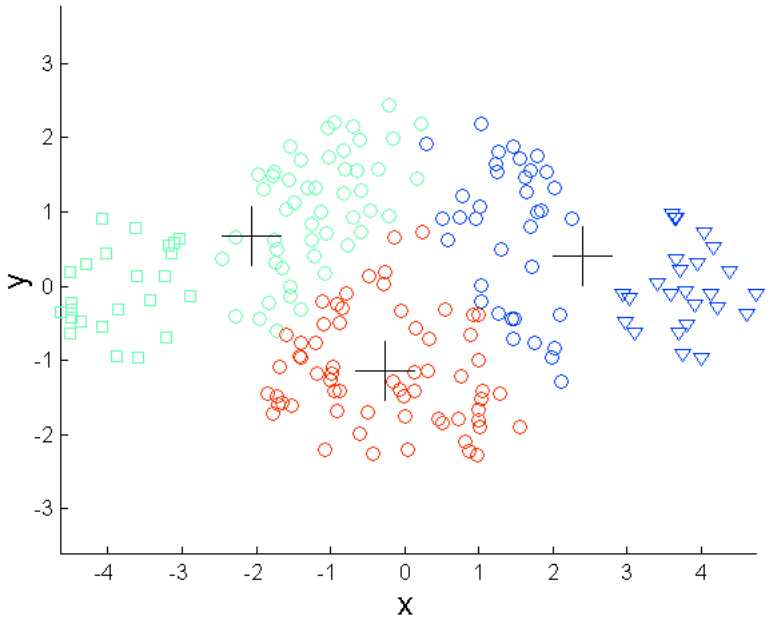
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes

- K-means has problems when the data contains outliers.
  - One possible solution is to remove outliers before clustering

Limitations of K-means: Differing Sizes

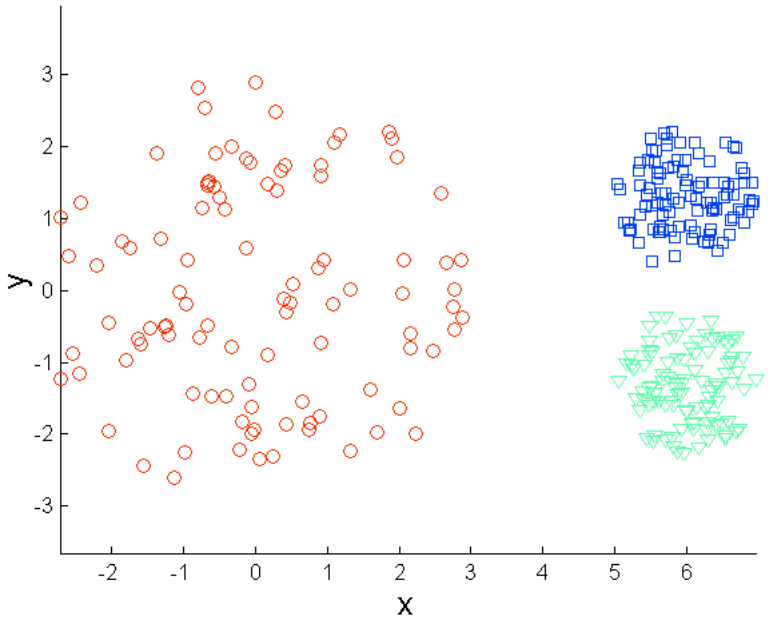


Original Points

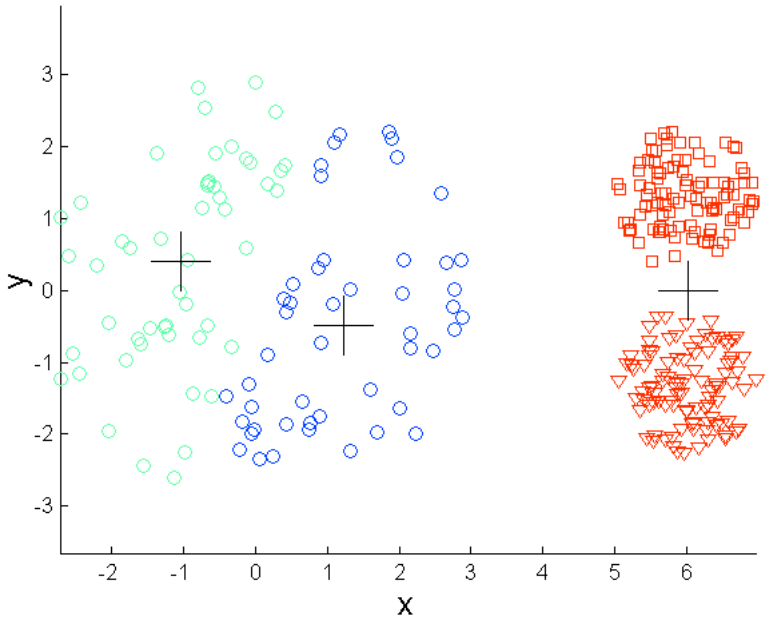


K-means (3 Clusters)

Limitations of K-means: Differing Density

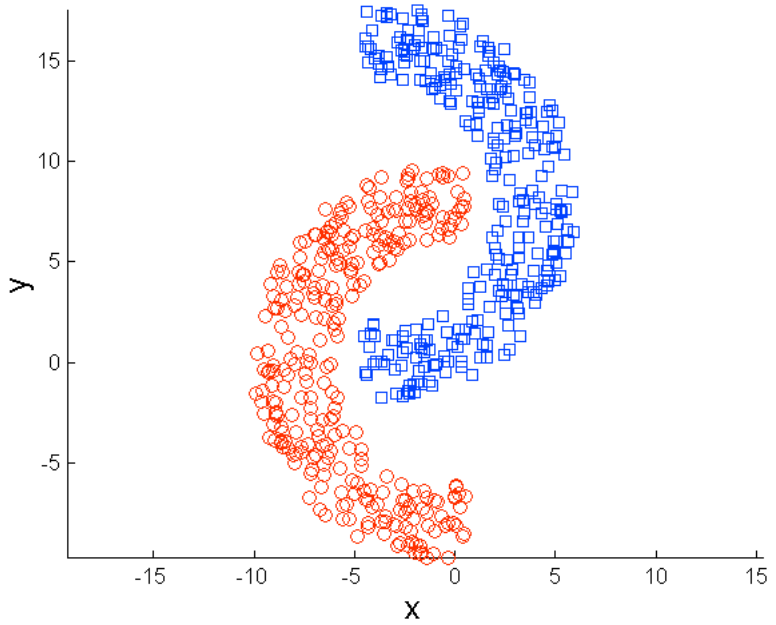


Original Points

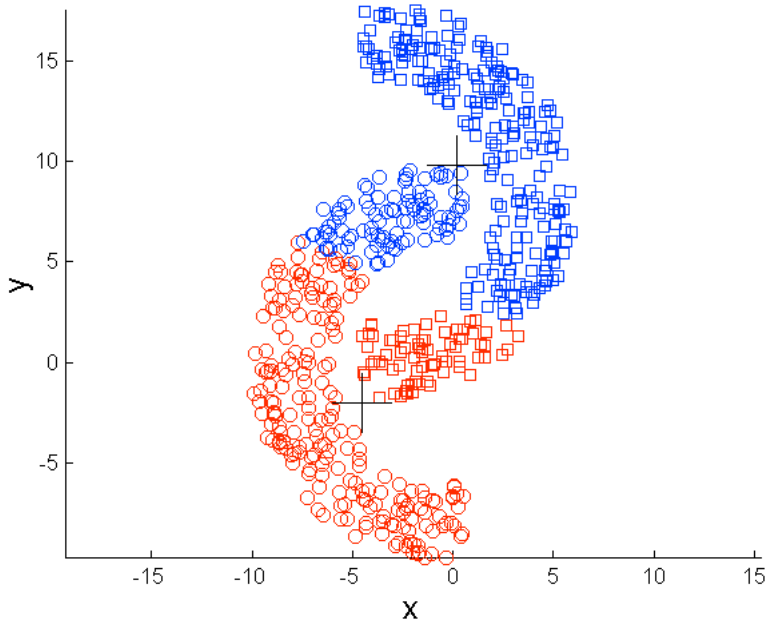


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

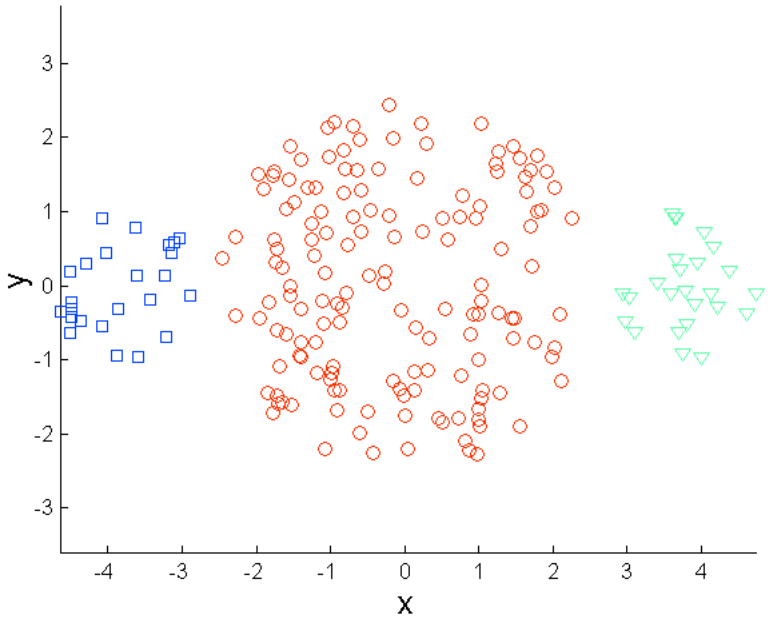


Original Points

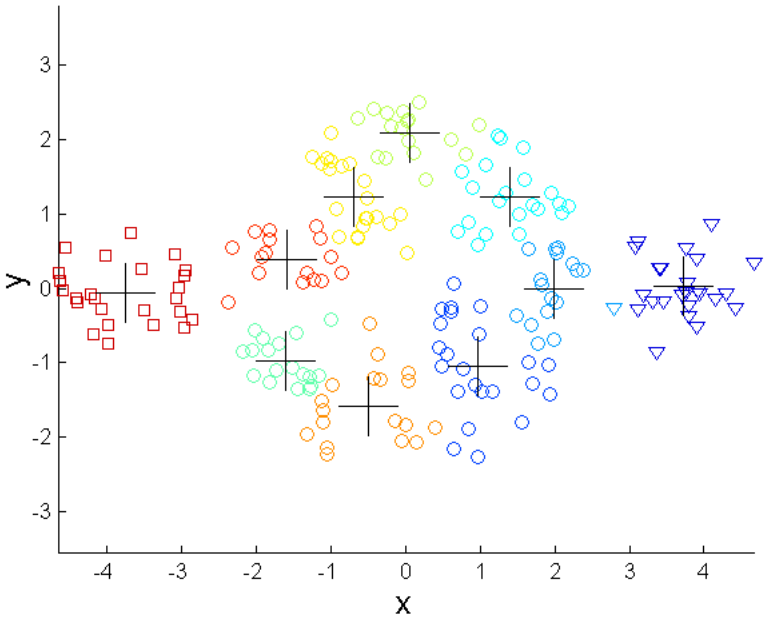


K-means (2 Clusters)

Overcoming K-means Limitations



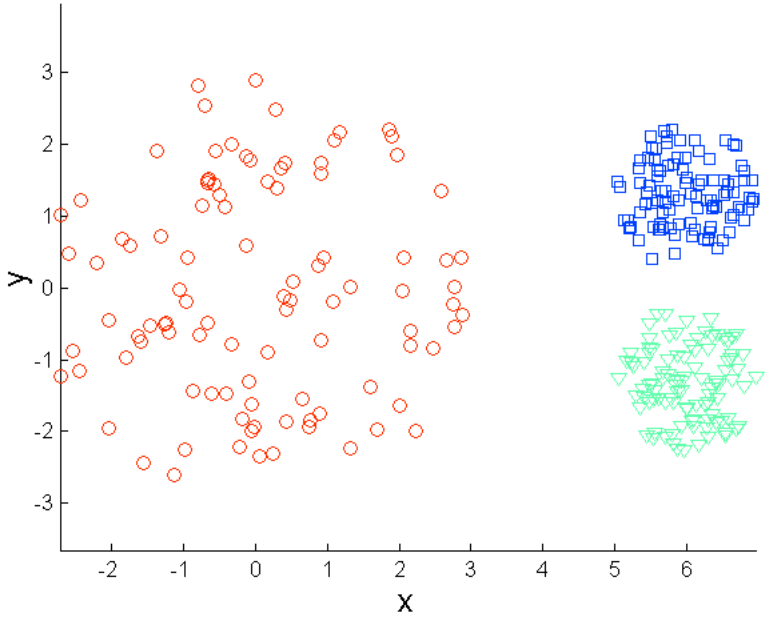
Original Points



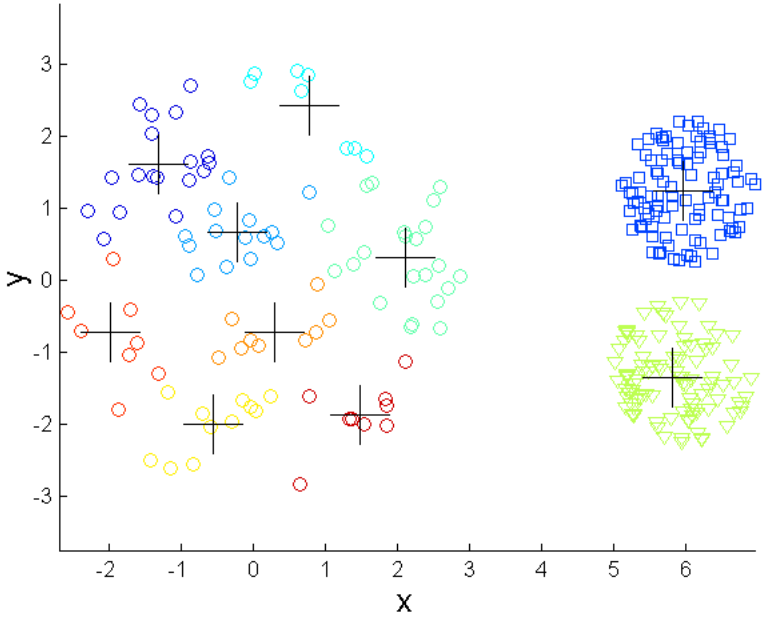
K-means Clusters

One solution is to find a large number of clusters such that each of them represents a part of a natural cluster. But these small clusters need to be put together in a post-processing step.

Overcoming K-means Limitations



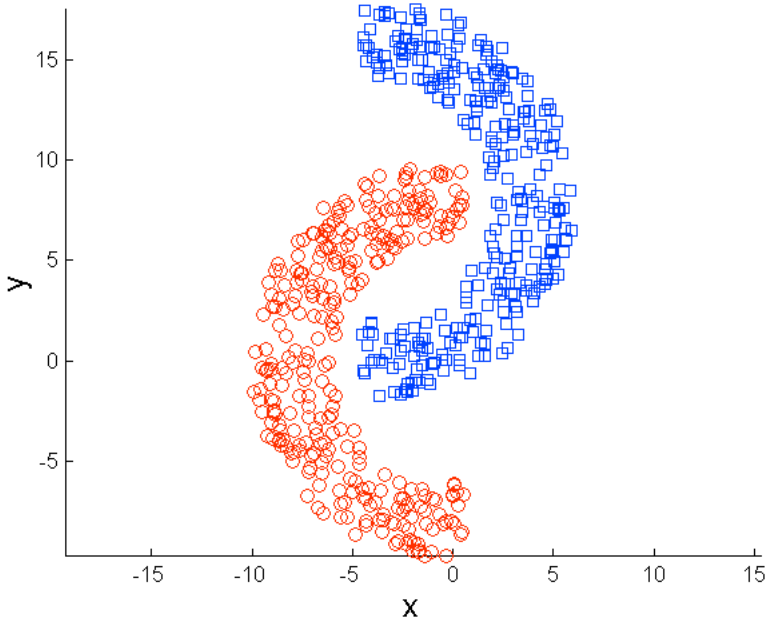
Original Points



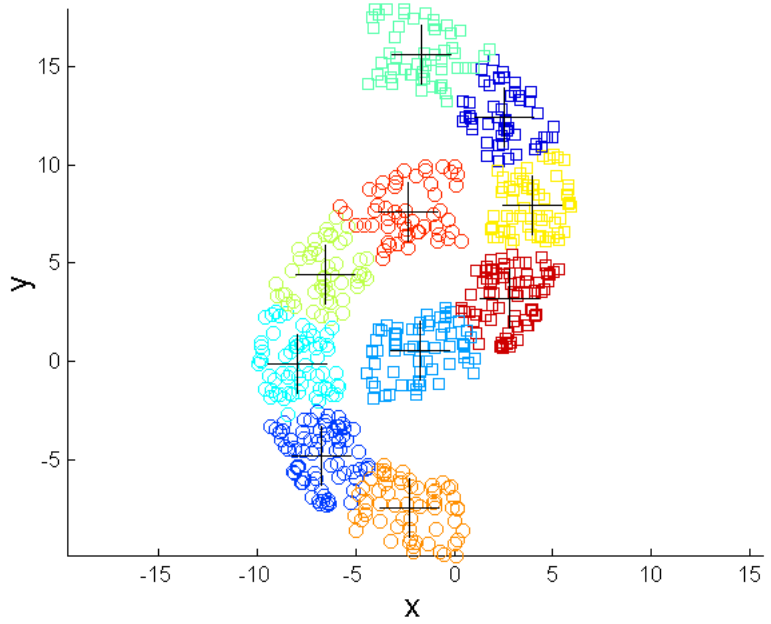
K-means Clusters

One solution is to find a large number of clusters such that each of them represents a part of a natural cluster. But these small clusters need to be put together in a post-processing step.

Overcoming K-means Limitations



Original Points

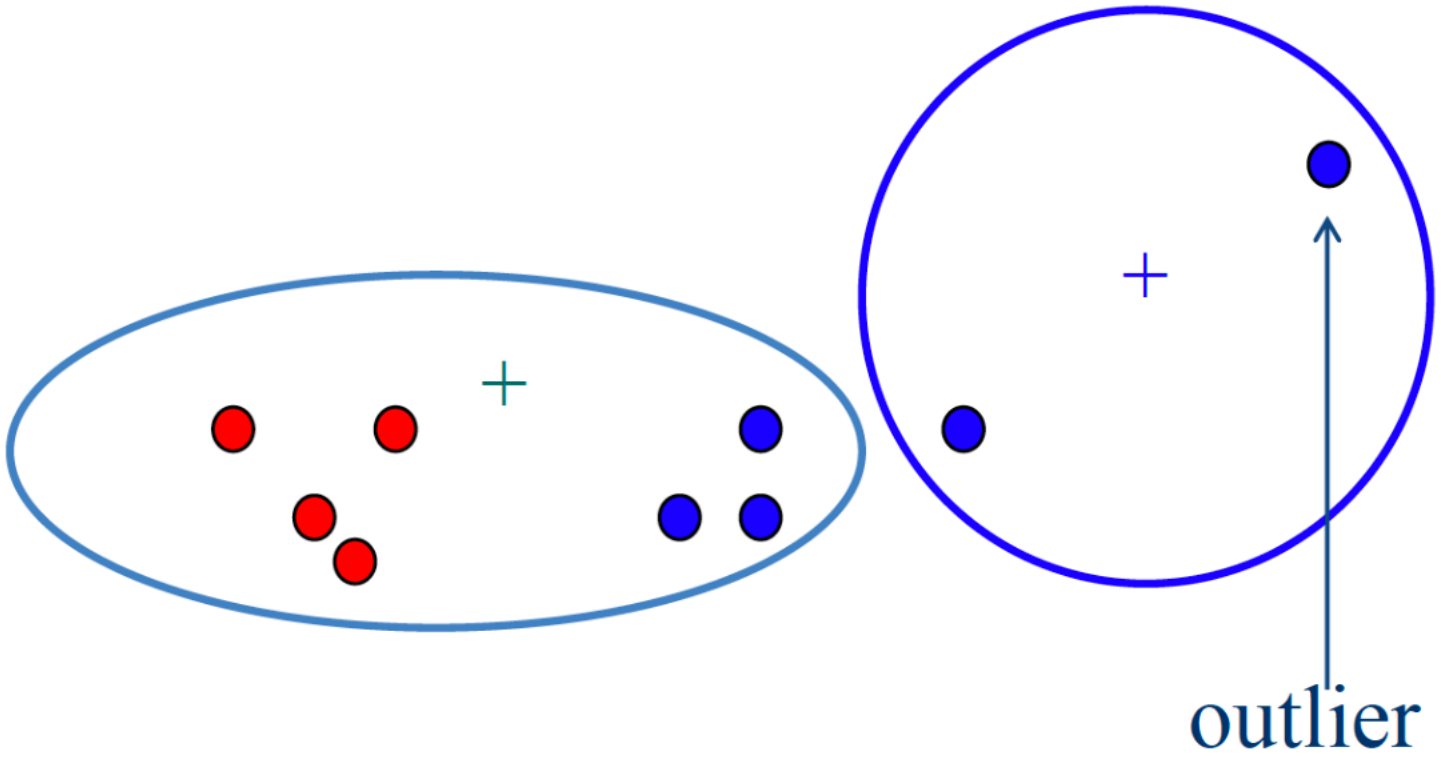


K-means Clusters

One solution is to find a large number of clusters such that each of them represents a part of a natural cluster. But these small clusters need to be put together in a post-processing step.

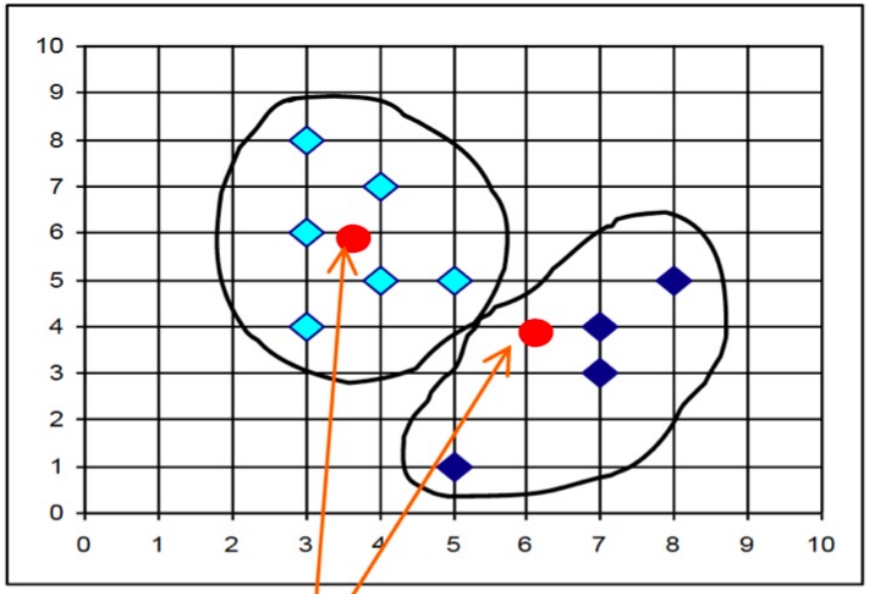
K-means: Outliers

- Outlier: samples with extremely large (or small) values

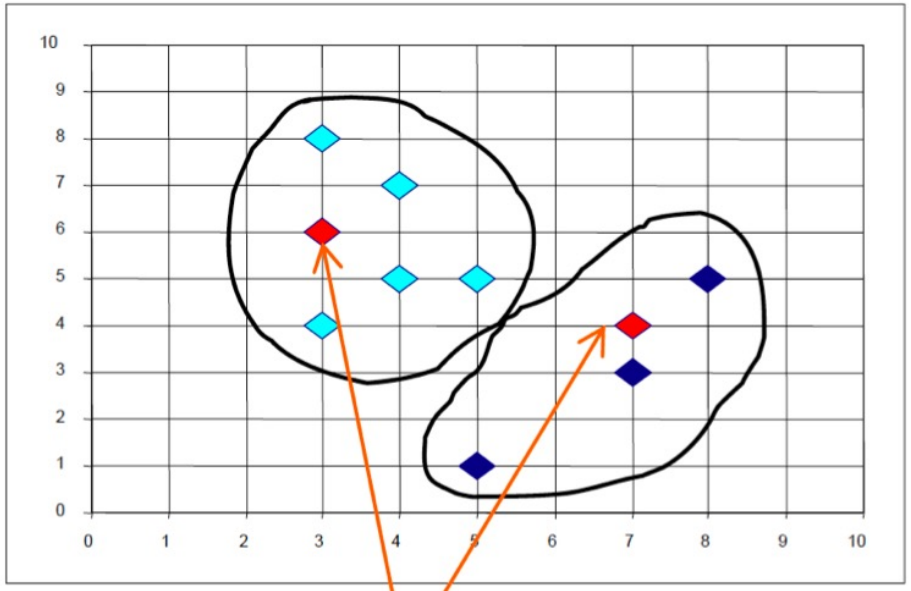


Clustering - K-medoids

- **K-means and K-medoids**
  - K-means: Computer cluster centers (may not be the original data point)
  - K-medoids: Each cluster's centroid is represented by a point in the cluster



k-means



k-medoids