Clustering
Lecture 2: Partitional Methods

Jing Gao
SUNY Buffalo

Outline

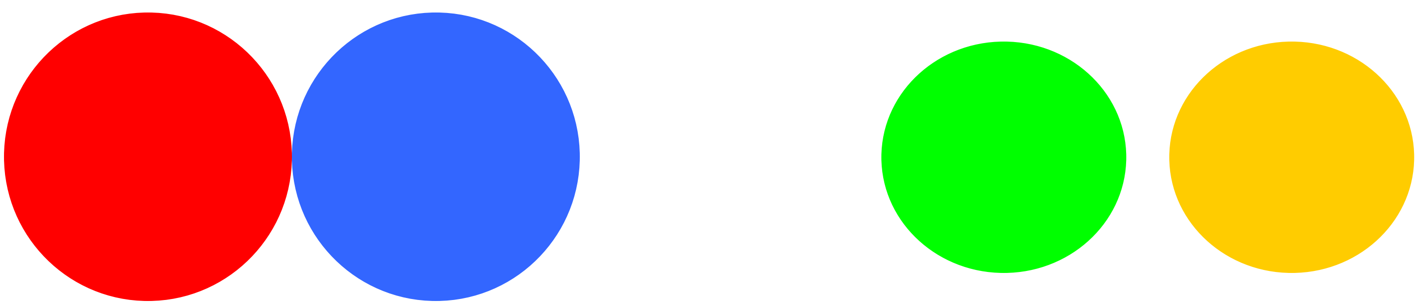
- **Basics**
  - Motivation, definition, evaluation
- **Methods**
  - Partitional
  - Hierarchical
  - Density-based
  - Mixture model
  - Spectral methods
- **Advanced topics**
  - Clustering ensemble
  - Clustering in MapReduce
  - Semi-supervised clustering, subspace clustering, co-clustering, etc.

Partitional Methods

- K-means algorithms
- Optimization of SSE
- Improvement on K-Means
- K-means variants
- Limitation of K-means

Partitional Methods

- **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 called centroid
  - Each point is assigned to the cluster with the closest centroid
  - The number of clusters usually should be specified



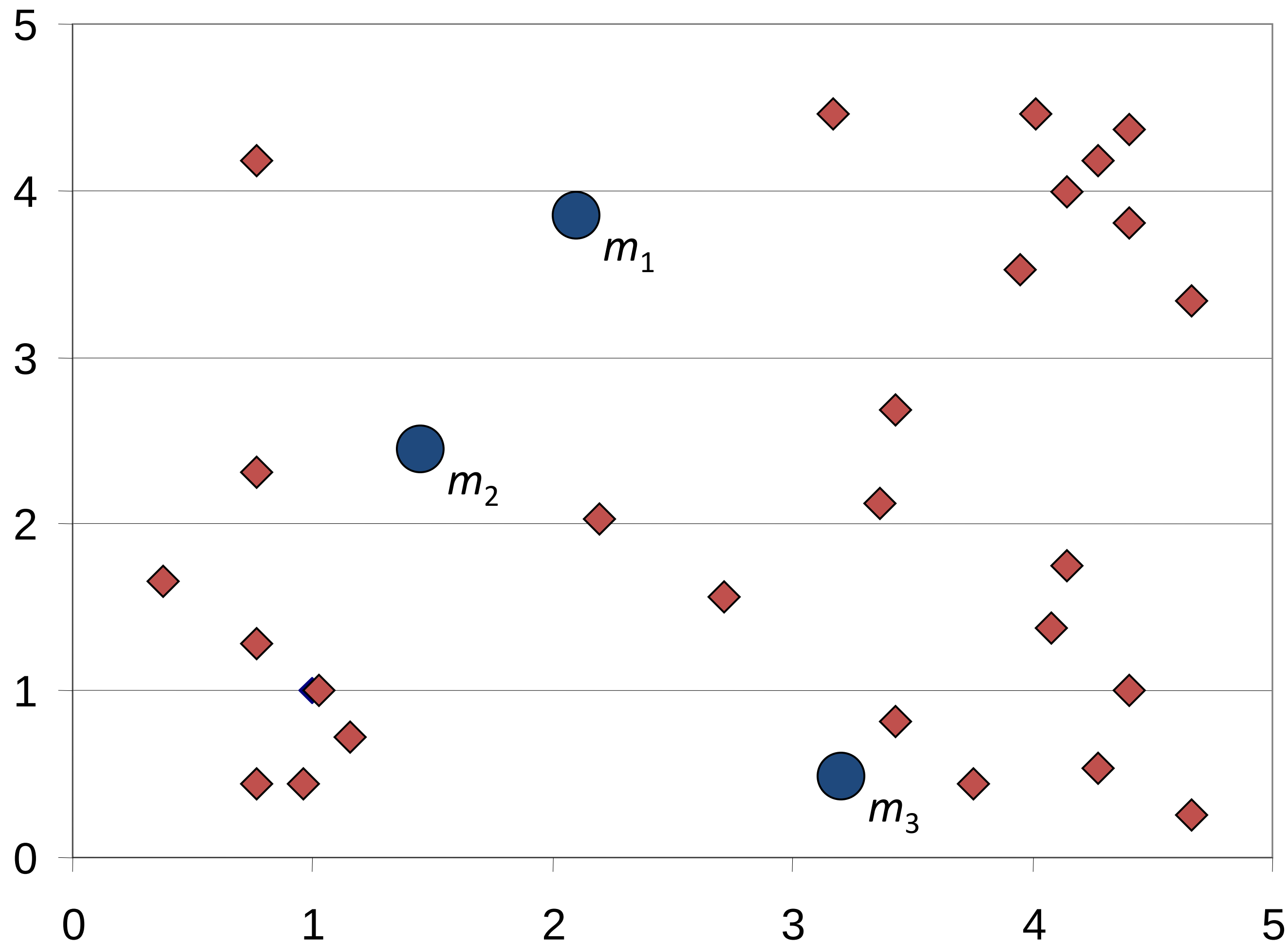
4 center-based clusters

K-means

- **Partition $\{x_1,\ldots,x_n\}$ into K clusters**
  - K is predefined
- **Initialization**
  - Specify the initial cluster centers (centroids)
- **Iteration until no change**
  - For each object x_i
    - Calculate the distances between x_i and the K centroids
    - (Re)assign x_i to the cluster whose centroid is the closest to x_i
  - Update the cluster centroids based on current assignment

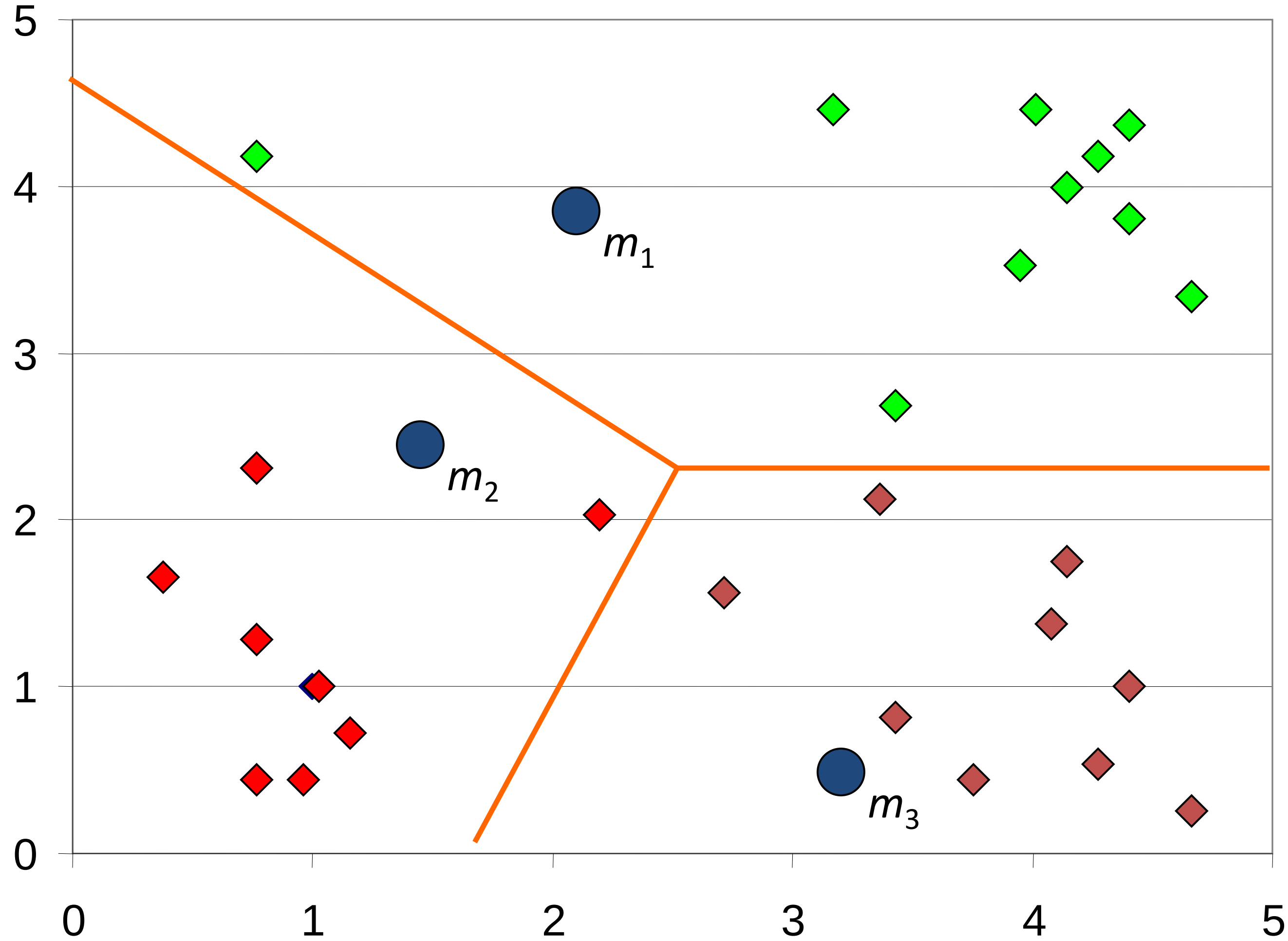
K-means: Initialization

Initialization: Determine the three cluster centers



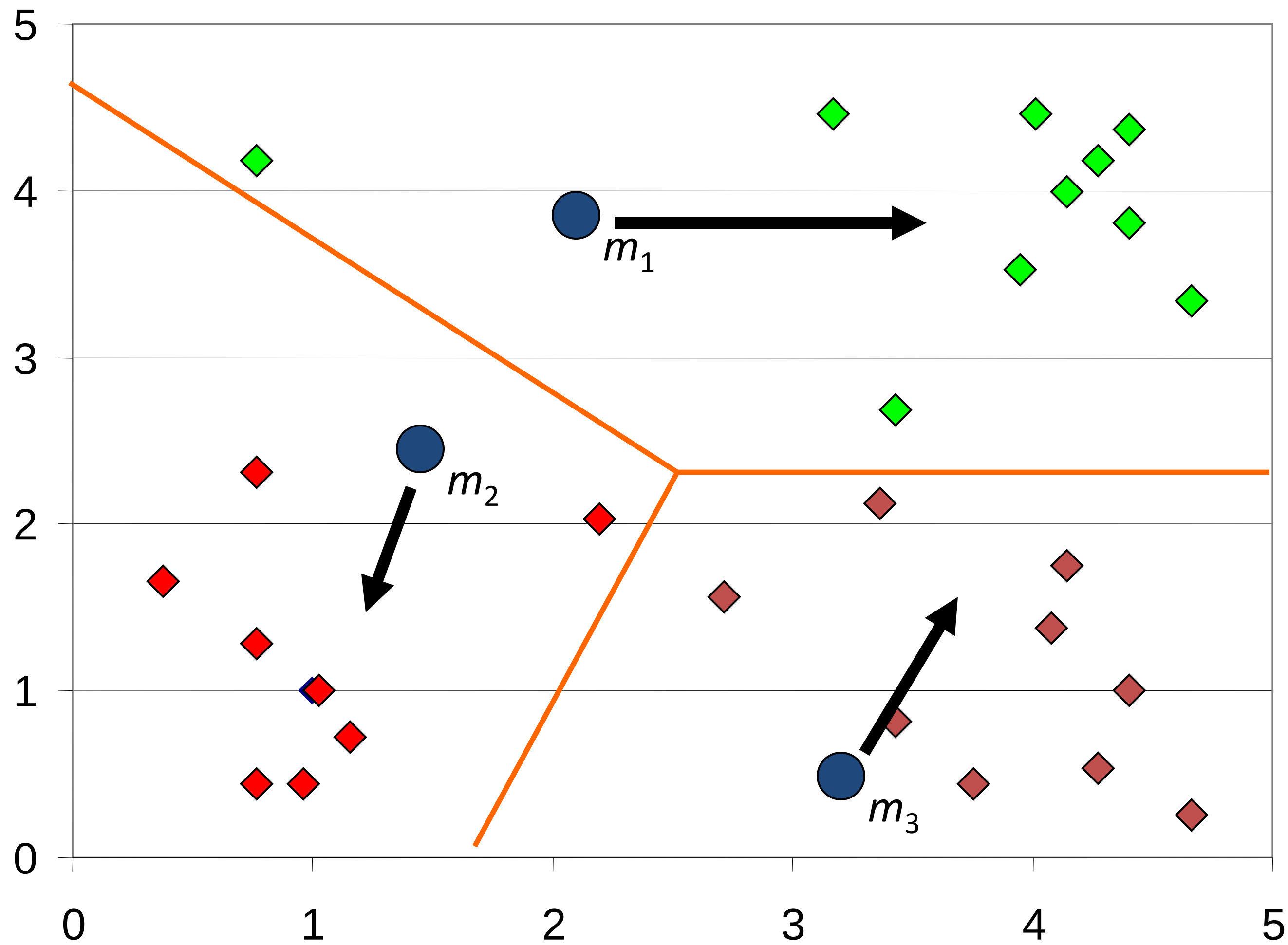
K-means Clustering: Cluster Assignment

Assign each object to the cluster which has the closet distance from the centroid to the object



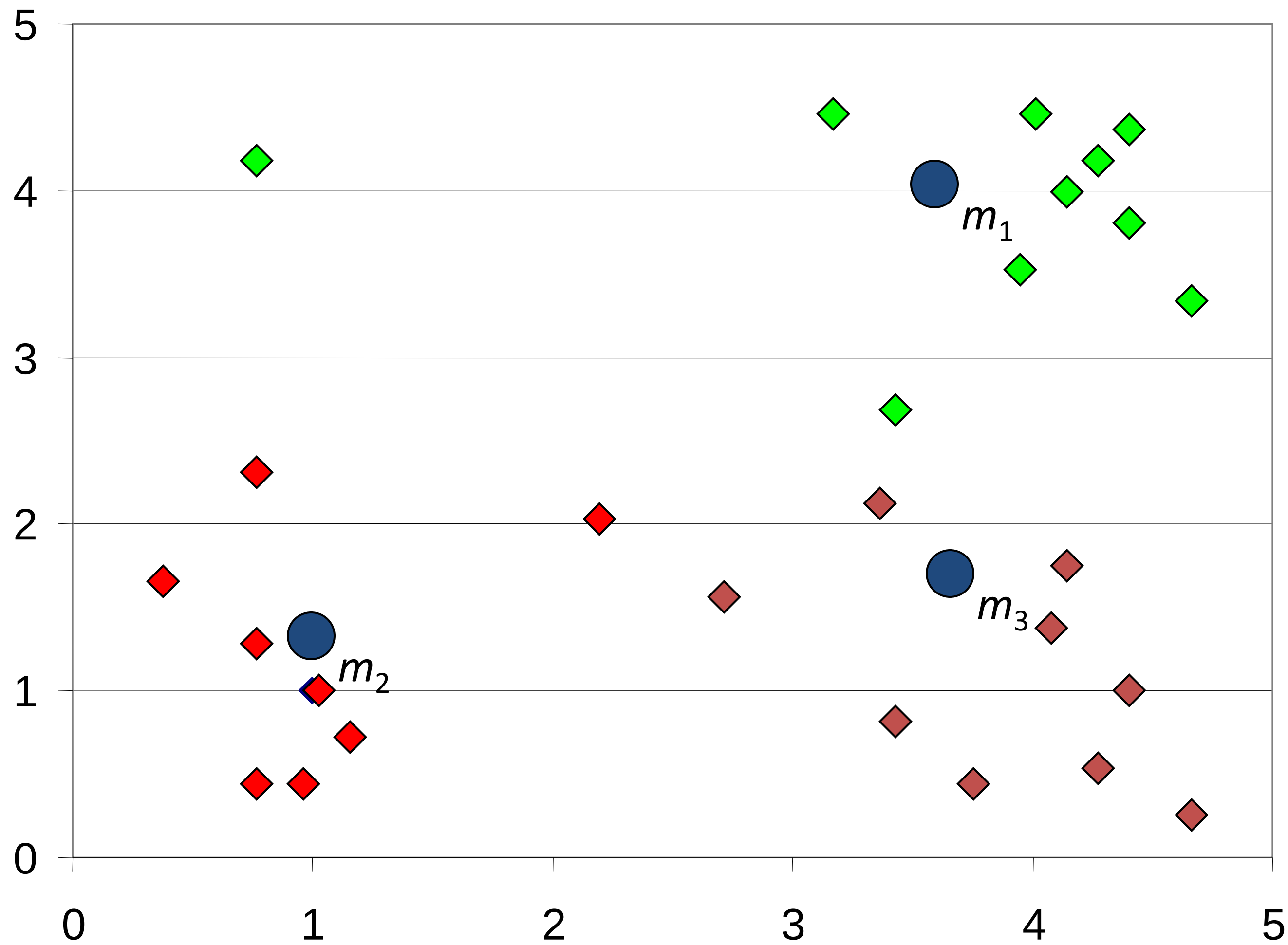
K-means Clustering: Update Cluster Centroid

Compute cluster centroid as the center of the points in the cluster



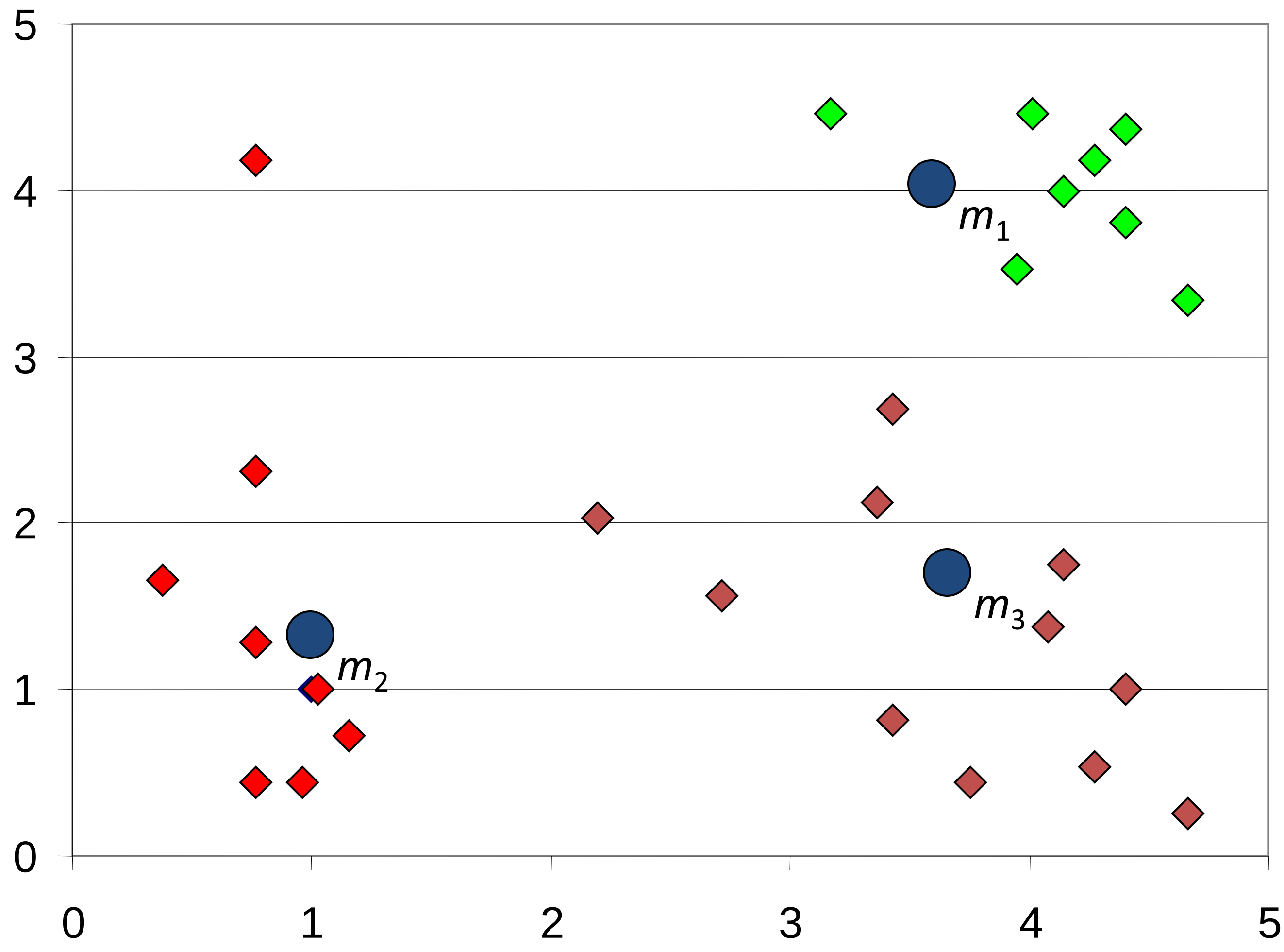
K-means Clustering: Update Cluster Centroid

Compute cluster centroid as the center of the points in the cluster



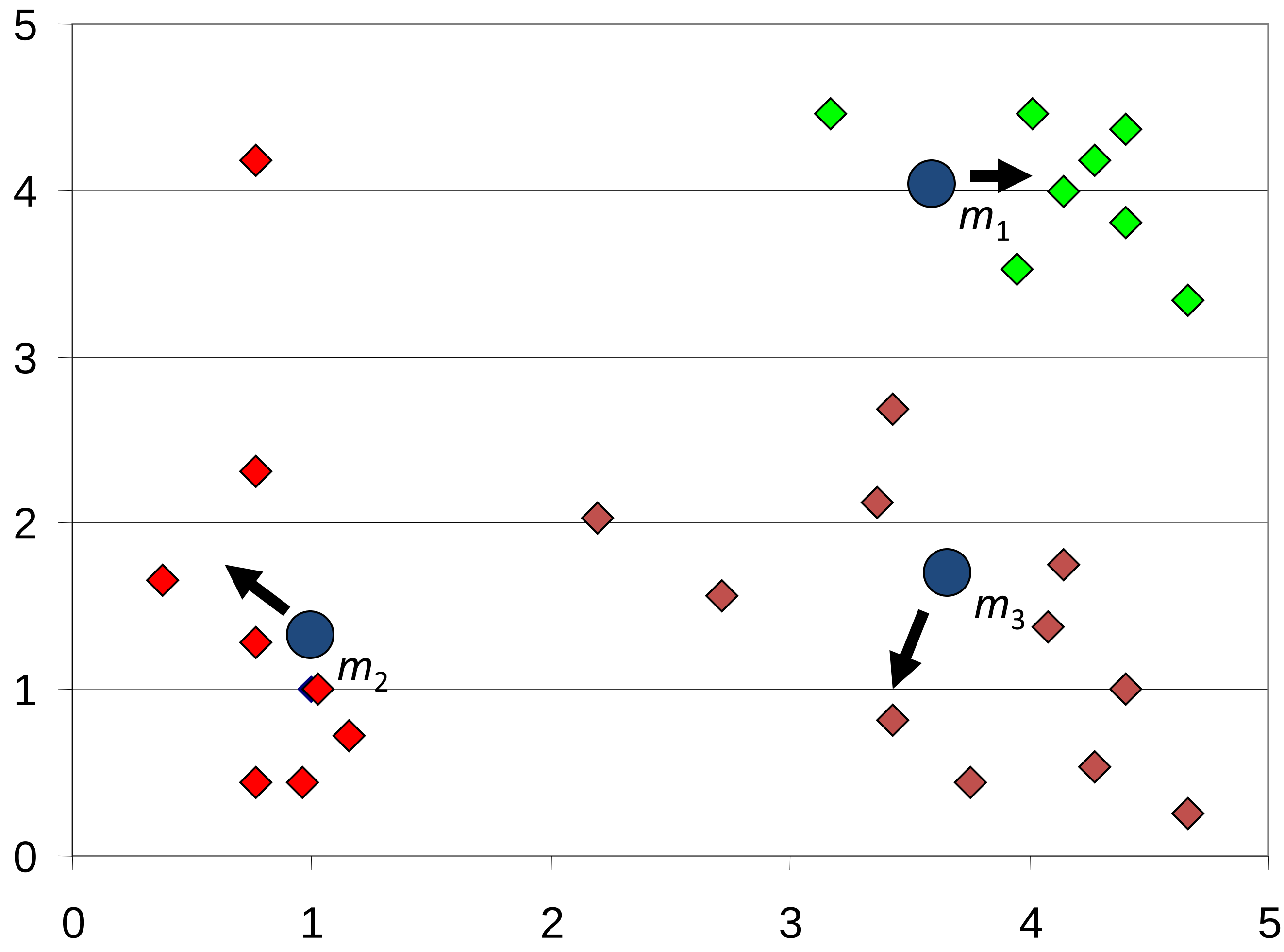
K-means Clustering: Cluster Assignment

Assign each object to the cluster which has the closet distance from the centroid to the object



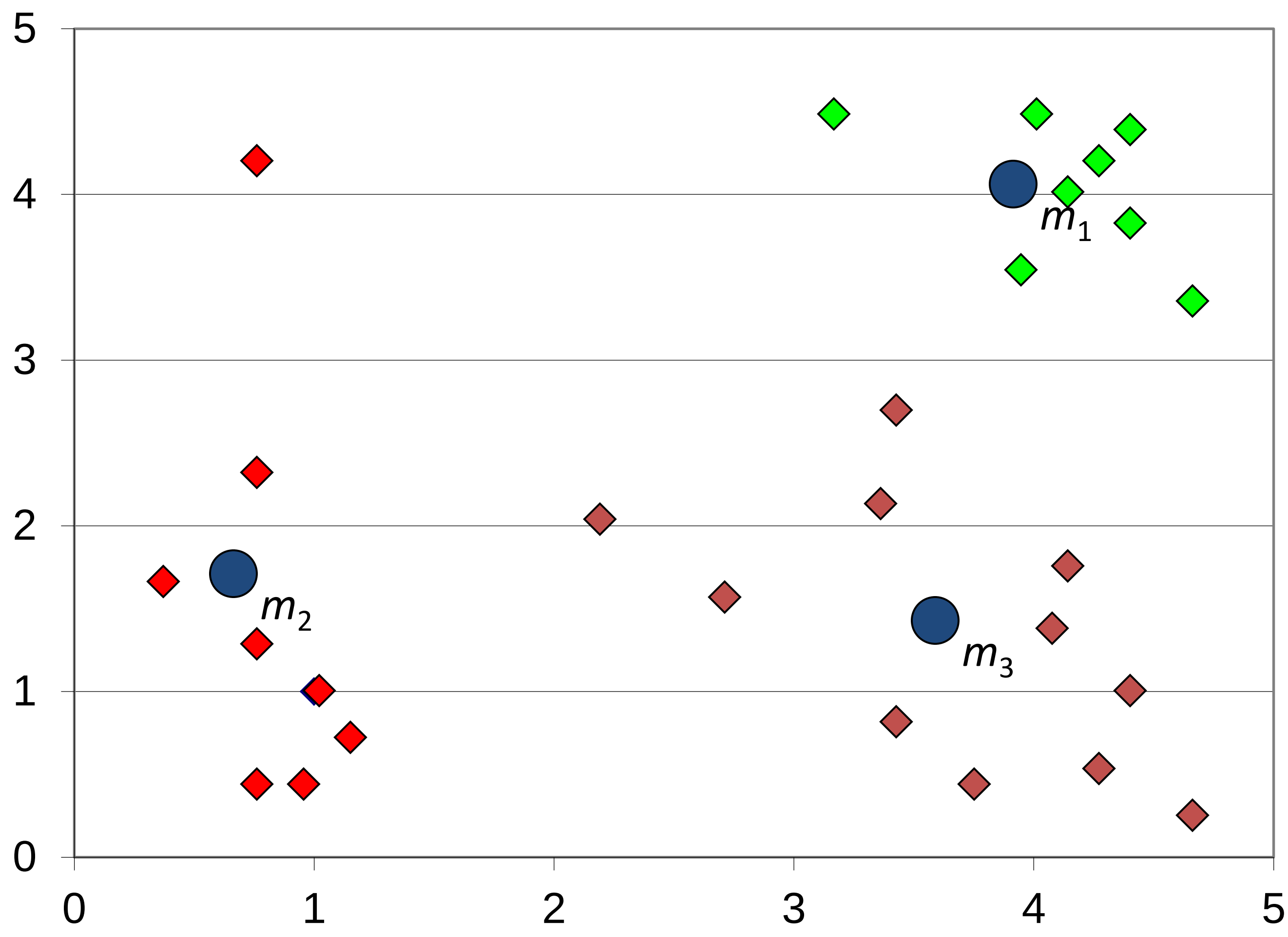
K-means Clustering: Update Cluster Centroid

Compute cluster centroid as the center of the points in the cluster



K-means Clustering: Update Cluster Centroid

Compute cluster centroid as the center of the points in the cluster



Partitional Methods

- K-means algorithms
- Optimization of SSE
- Improvement on K-Means
- K-means variants
- Limitation of K-means

Sum of Squared Error (SSE)

- Suppose the centroid of cluster C_j is m_j
- For each object x in C_j, compute the squared error between x and the centroid m_j
- Sum up the error of all the objects

$$SSE = \sum_{j} \sum_{x \in C_j} (x - m_j)^2$$

$$SSE = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

How to Minimize SSE

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

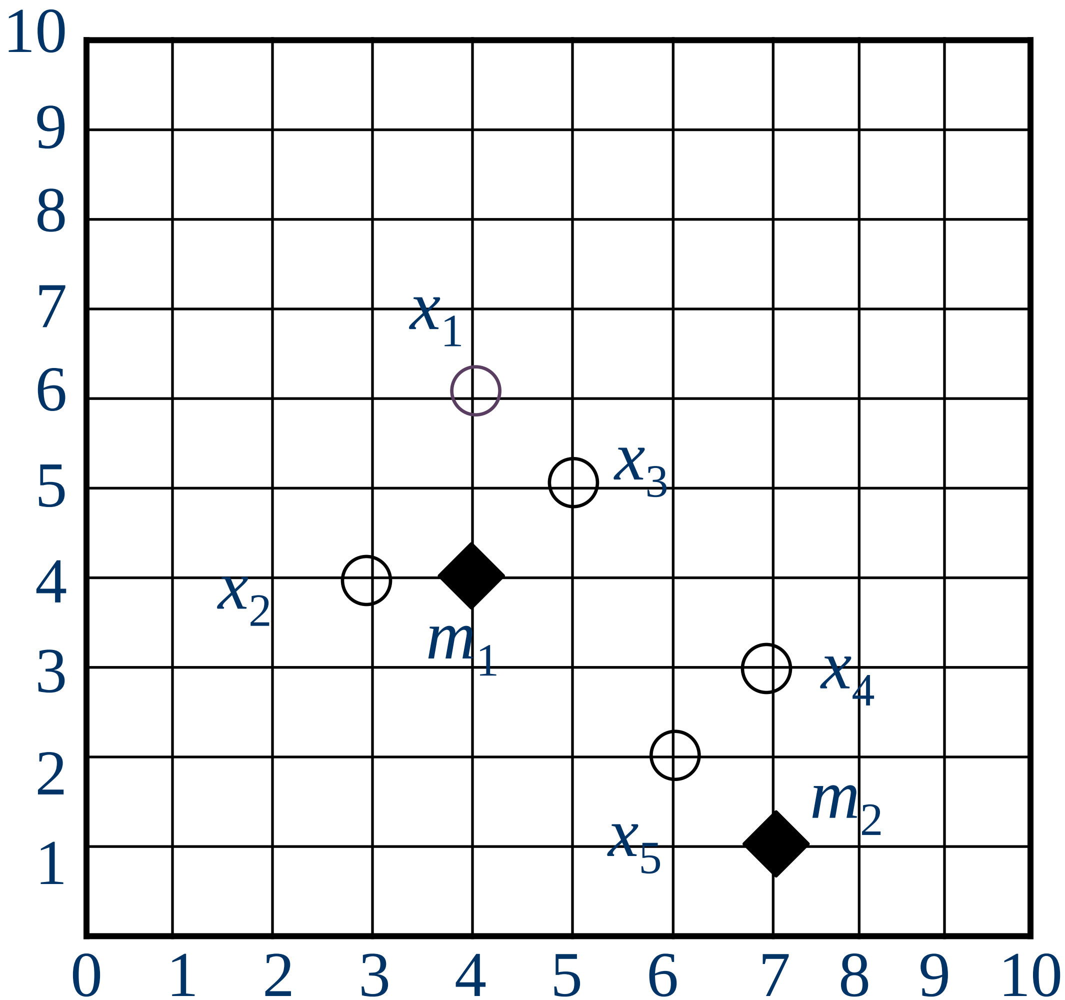
- **Two sets of variables to minimize**
  - Each object *x* belongs to which cluster? $x \in C_j$
  - What's the cluster centroid? m_j
- **Block coordinate descent**
  - Fix the cluster centroid—find cluster assignment that minimizes the current error
  - Fix the cluster assignment—compute the cluster centroids that minimize the current error

Cluster Assignment Step

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

- Cluster centroids (m_j) are known
- For each object
  - Choose C_j among all the clusters for x such that the distance between x and m_j is the minimum
  - Choose another cluster will incur a bigger error
- Minimize error on each object will minimize the SSE

Example—Cluster Assignment



Given m_1, m_2, which cluster each of the five points belongs to?

Assign points to the closet centroid—minimize SSE

$$x_1, x_2, x_3 \in C_1$$

$$x_4, x_5 \in C_2$$

$$SSE = (x_1 - m_1)^2 + (x_2 - m_1)^2 + (x_3 - m_1)^2 + (x_4 - m_2)^2 + (x_5 - m_2)^2$$

Cluster Centroid Computation Step

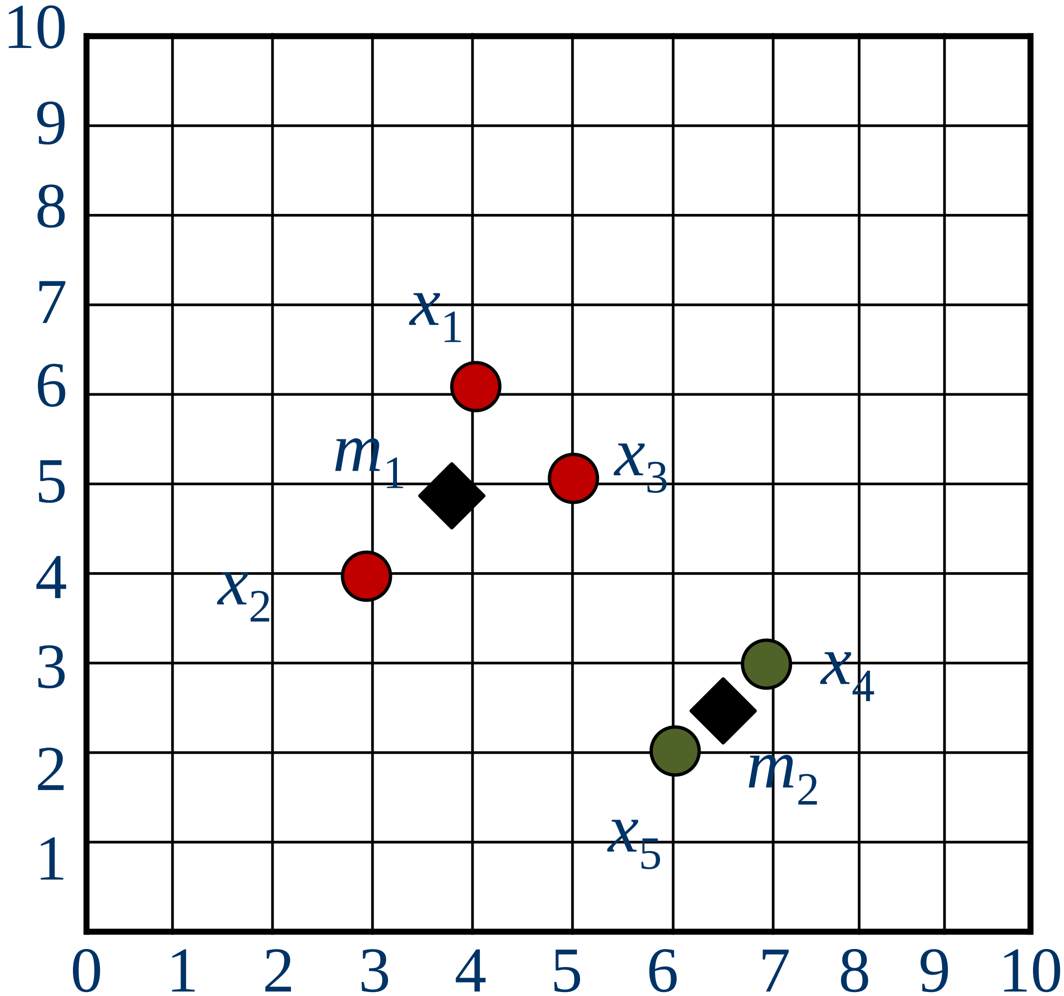
$$\min \sum_{j} \sum_{x \in C_j} (x - m_j)^2$$

- For each cluster
  - Choose cluster centroid 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

Example—Cluster Centroid Computation



Given the cluster assignment, compute the centers of the two clusters

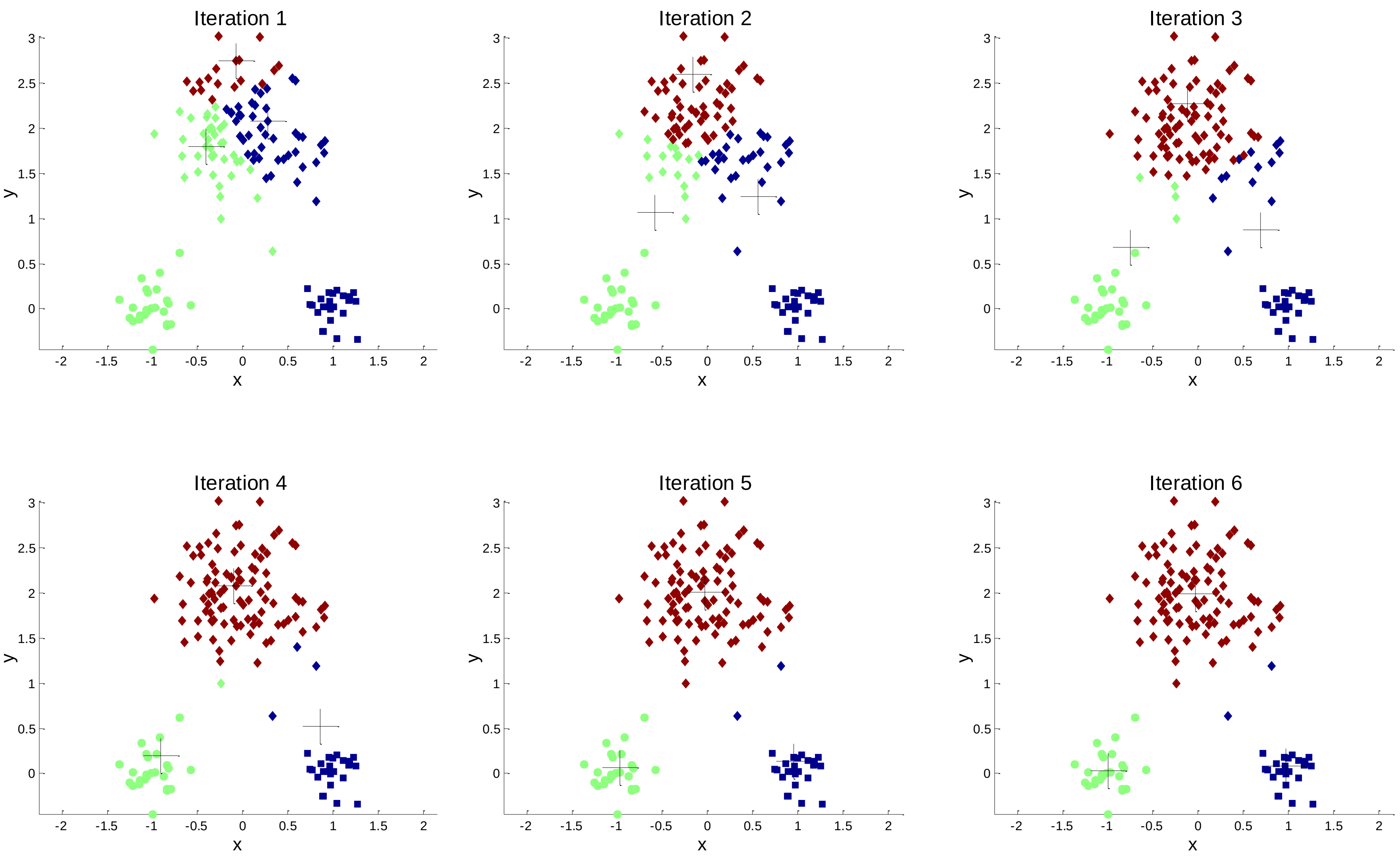
Comments on the K-Means Method

- **Strength**
  - Efficient: O(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*
  - Easy to implement
- **Issues**
  - Need to specify *K*, the number of clusters
  - Local minimum– Initialization matters
  - Empty clusters may appear

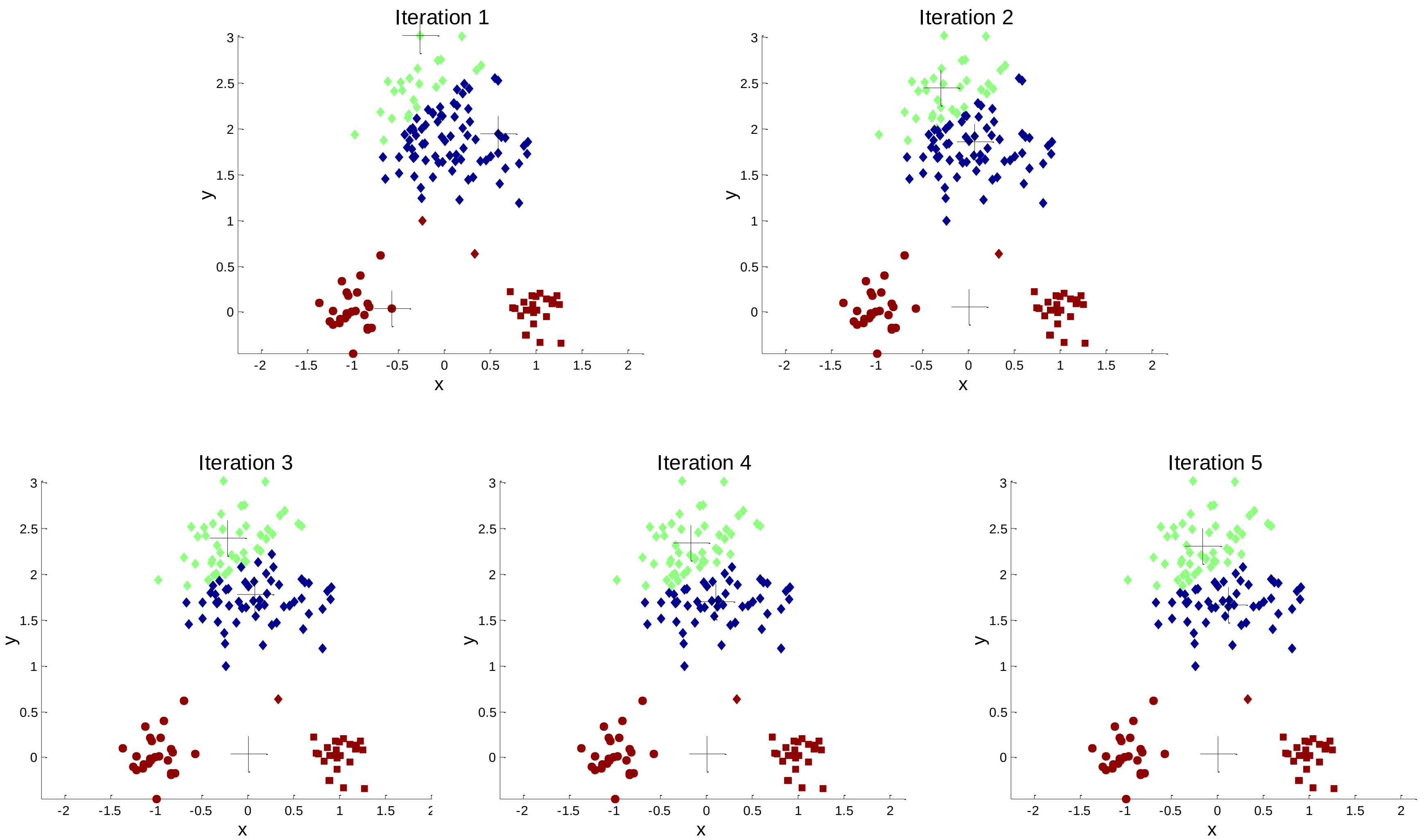
Partitional Methods

- K-means algorithms
- Optimization of SSE
- Improvement on K-Means
- K-means variants
- Limitation of K-means

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



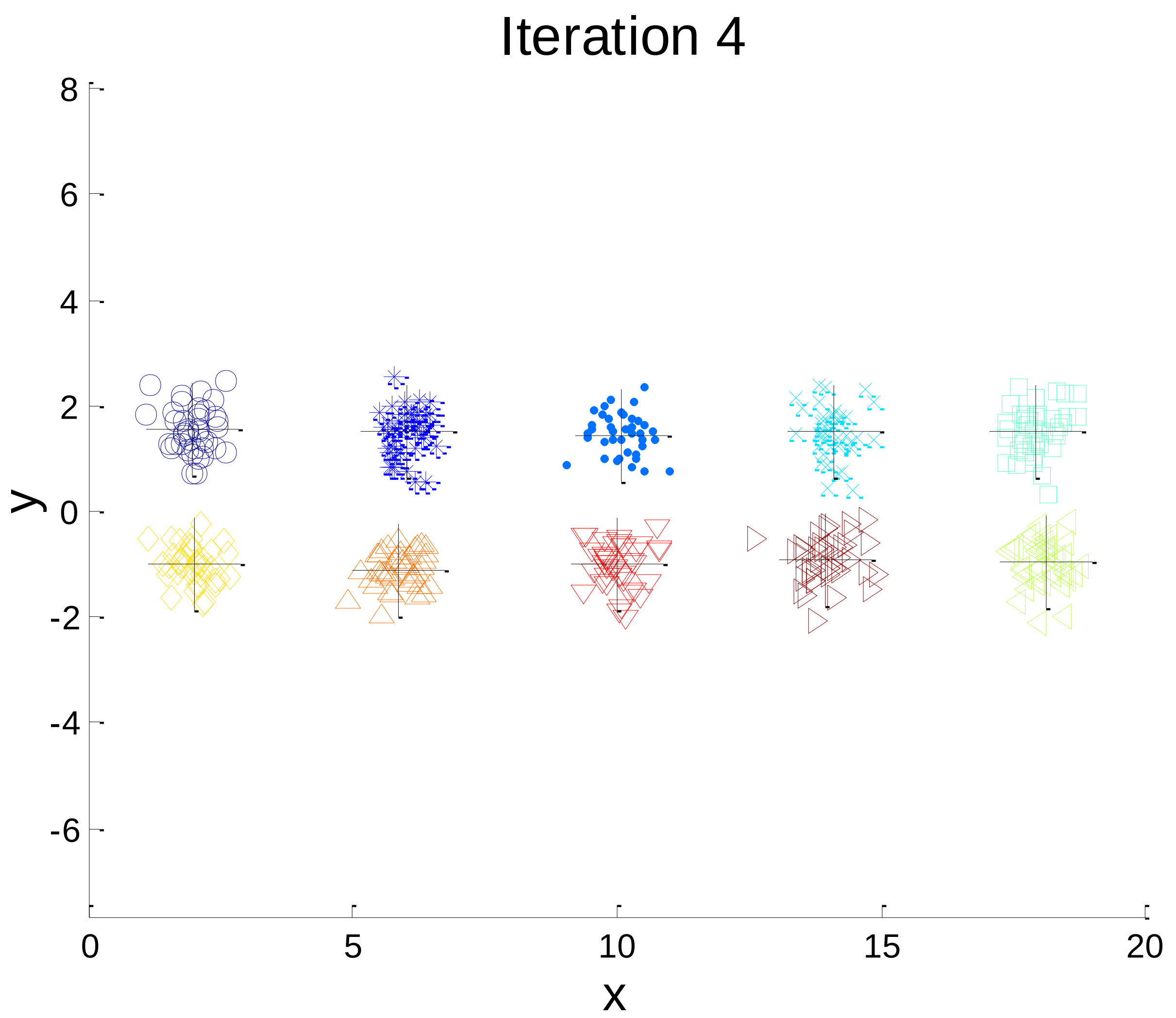
Problems with Selecting Initial Points

- 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

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

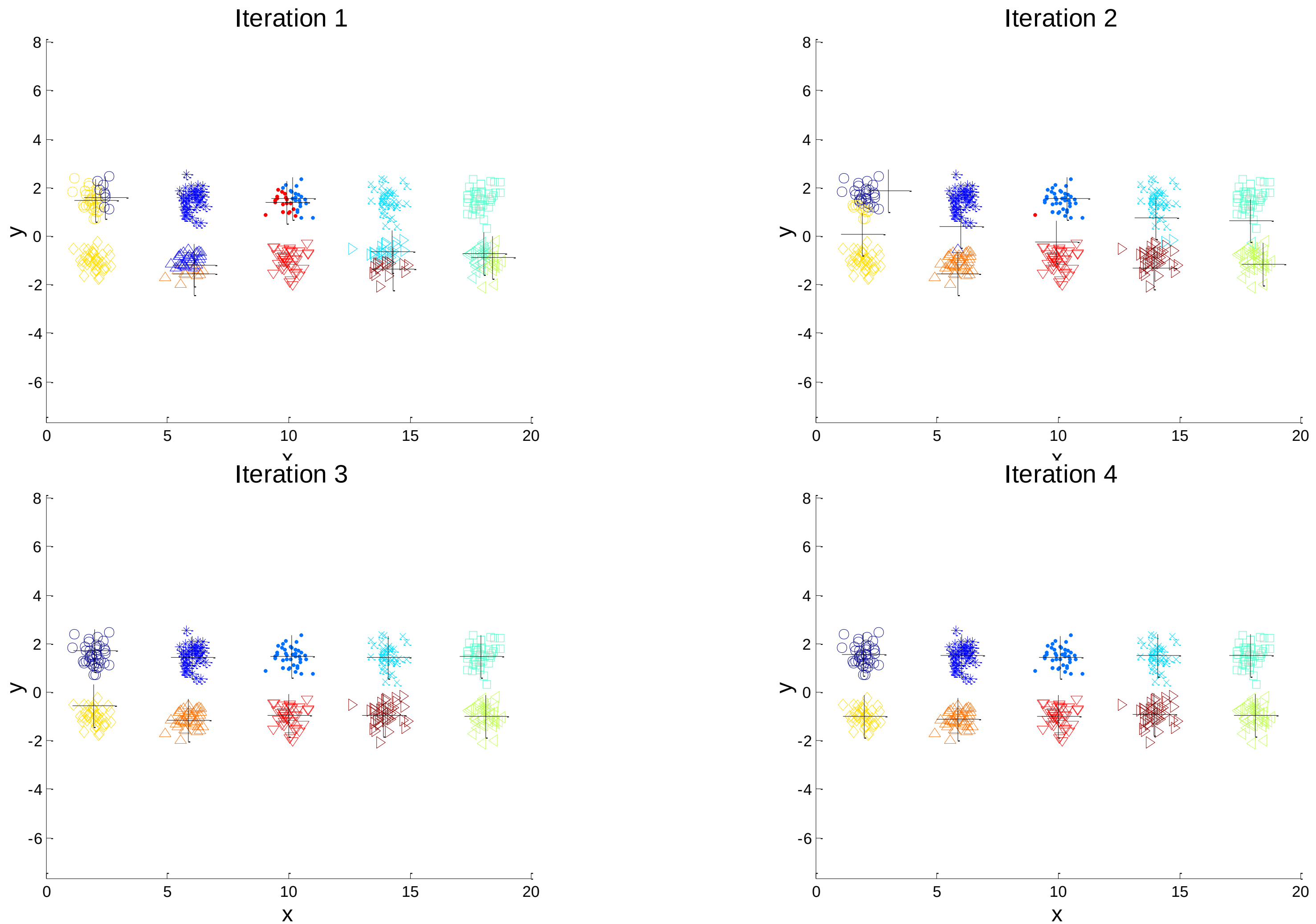
  - For example, if K = 10, then probability = $10!/10^{10}$ = 0.00036
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't

10 Clusters Example



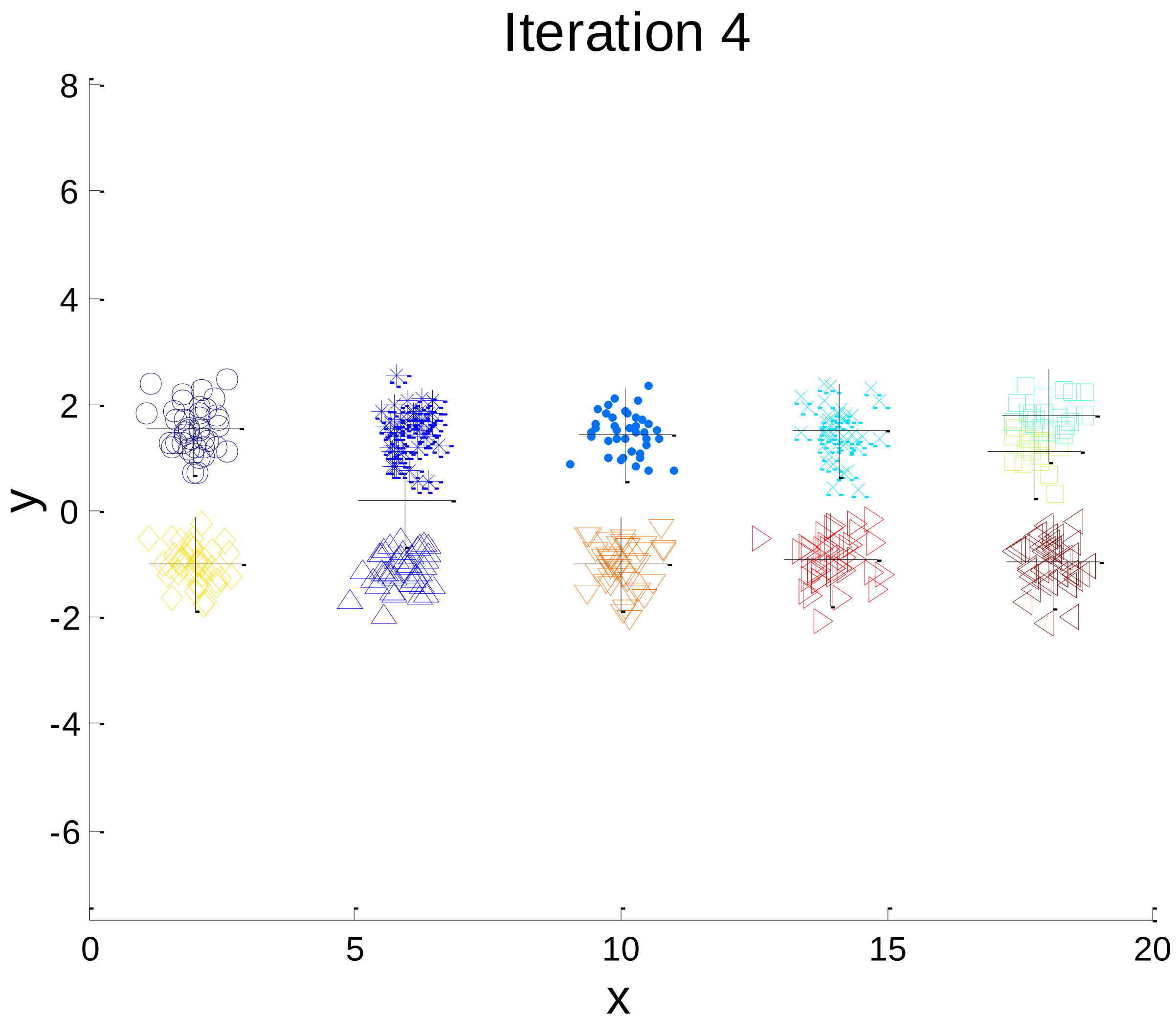
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



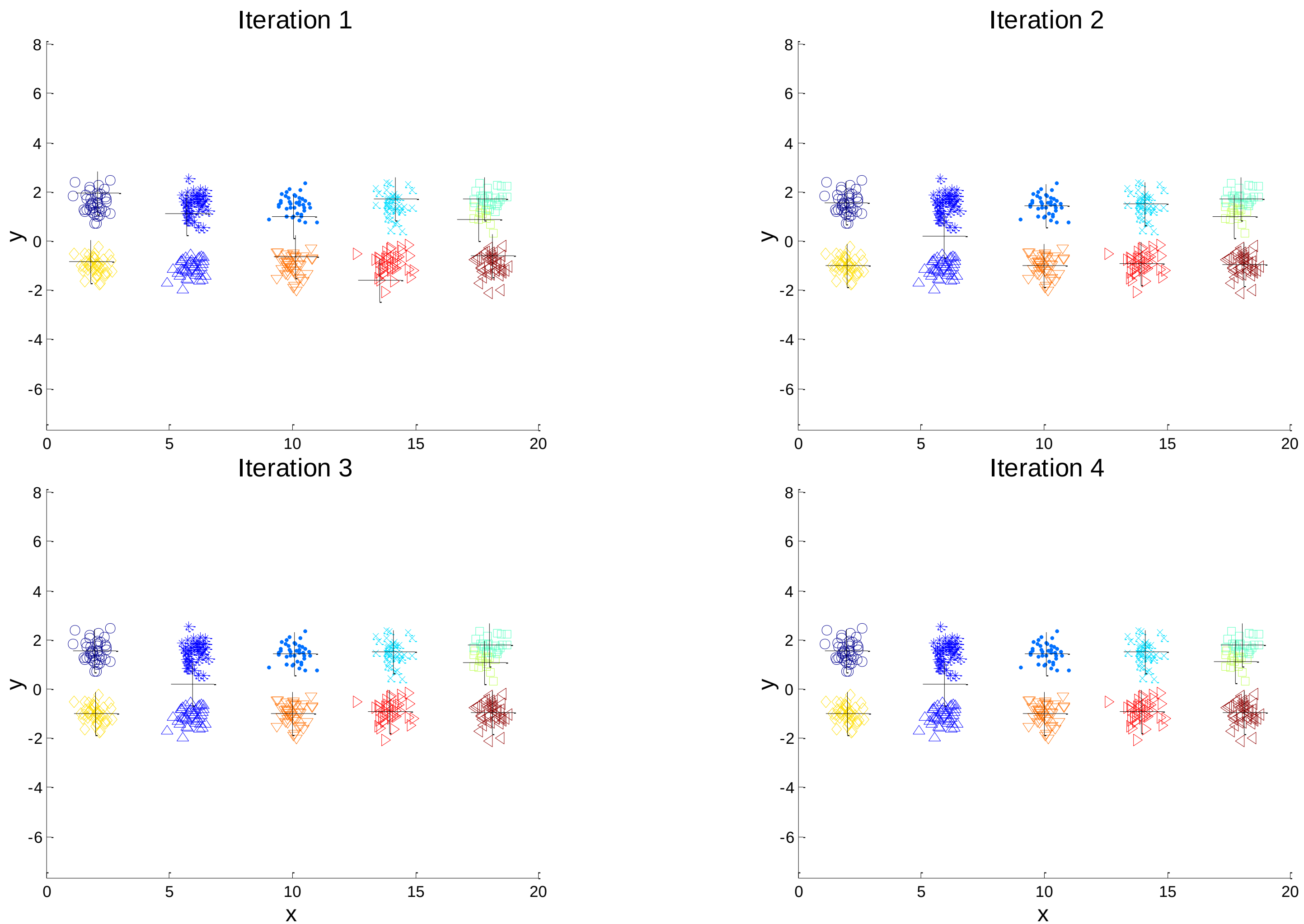
Starting with two initial centroids in one cluster of each pair of clusters

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
  - 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
- Postprocessing—Use K-means' results as other algorithms' initialization
- 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

```
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
```

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

Partitional Methods

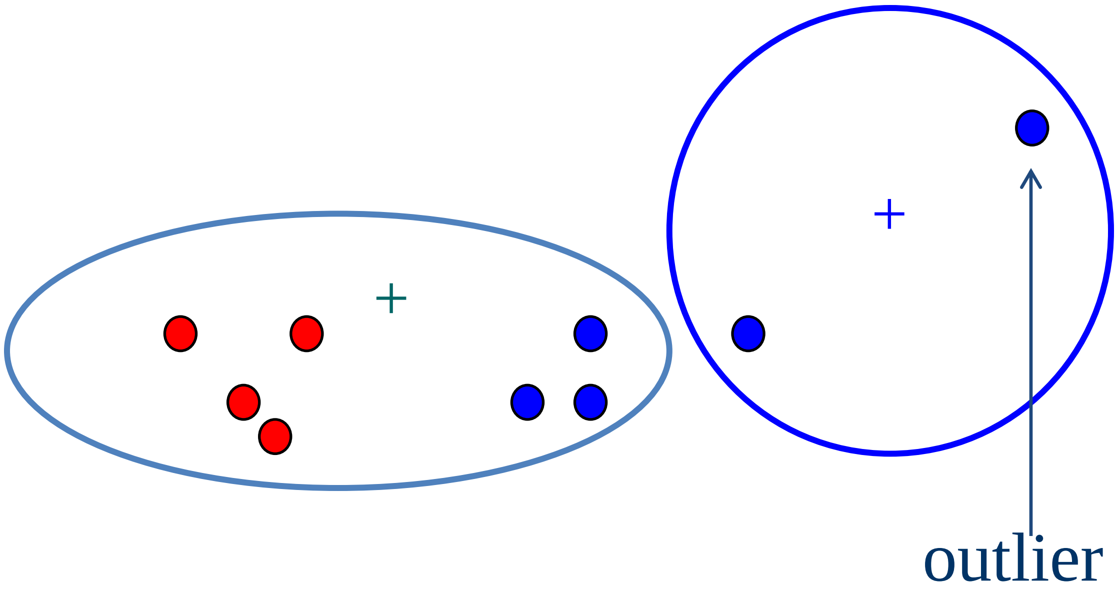
- K-means algorithms
- Optimization of SSE
- Improvement on K-Means
- K-means variants
- Limitation of K-means

Variations of the K-Means Method

- **Most of the variants of the K-means which differ in**
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- **Two important issues of K-means**
  - Sensitive to noisy data and *outliers*
    - K-medoids algorithm
  - Applicable only to objects in a continuous multi-dimensional space
    - Using the K-modes method for categorical data

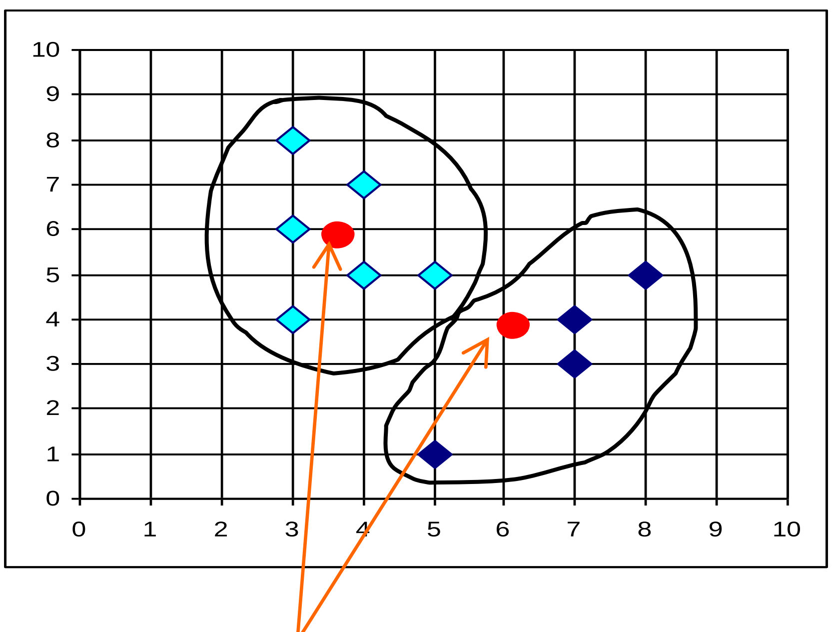
Sensitive to Outliers

- **K-means is sensitive to outliers**
  - Outlier: objects with extremely large (or small) values
    - May substantially distort the distribution of the data

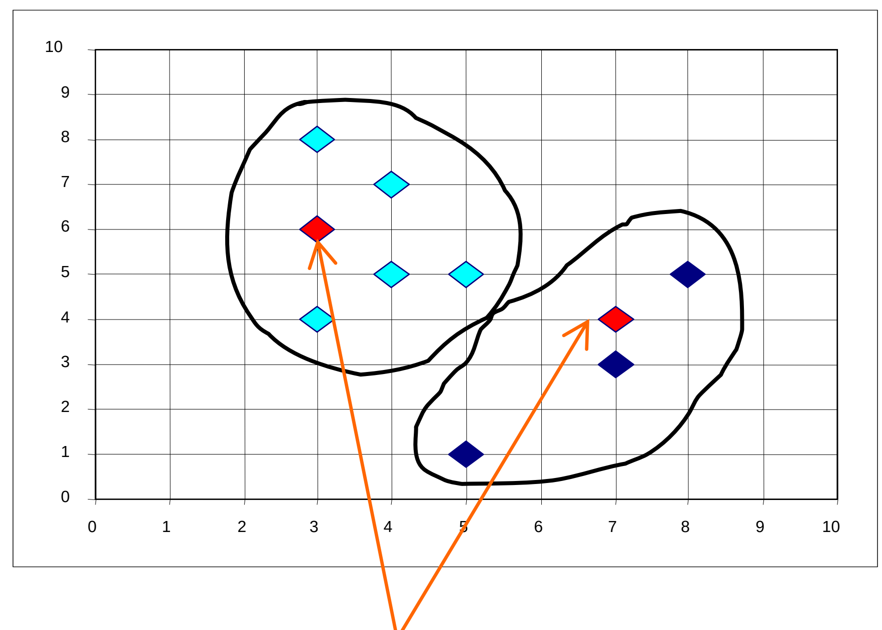


K-Medoids Clustering Method

- Difference between 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-medoids is more robust than K-means in the presence of outliers because a medoid is less influenced by outliers or other extreme values



k-means

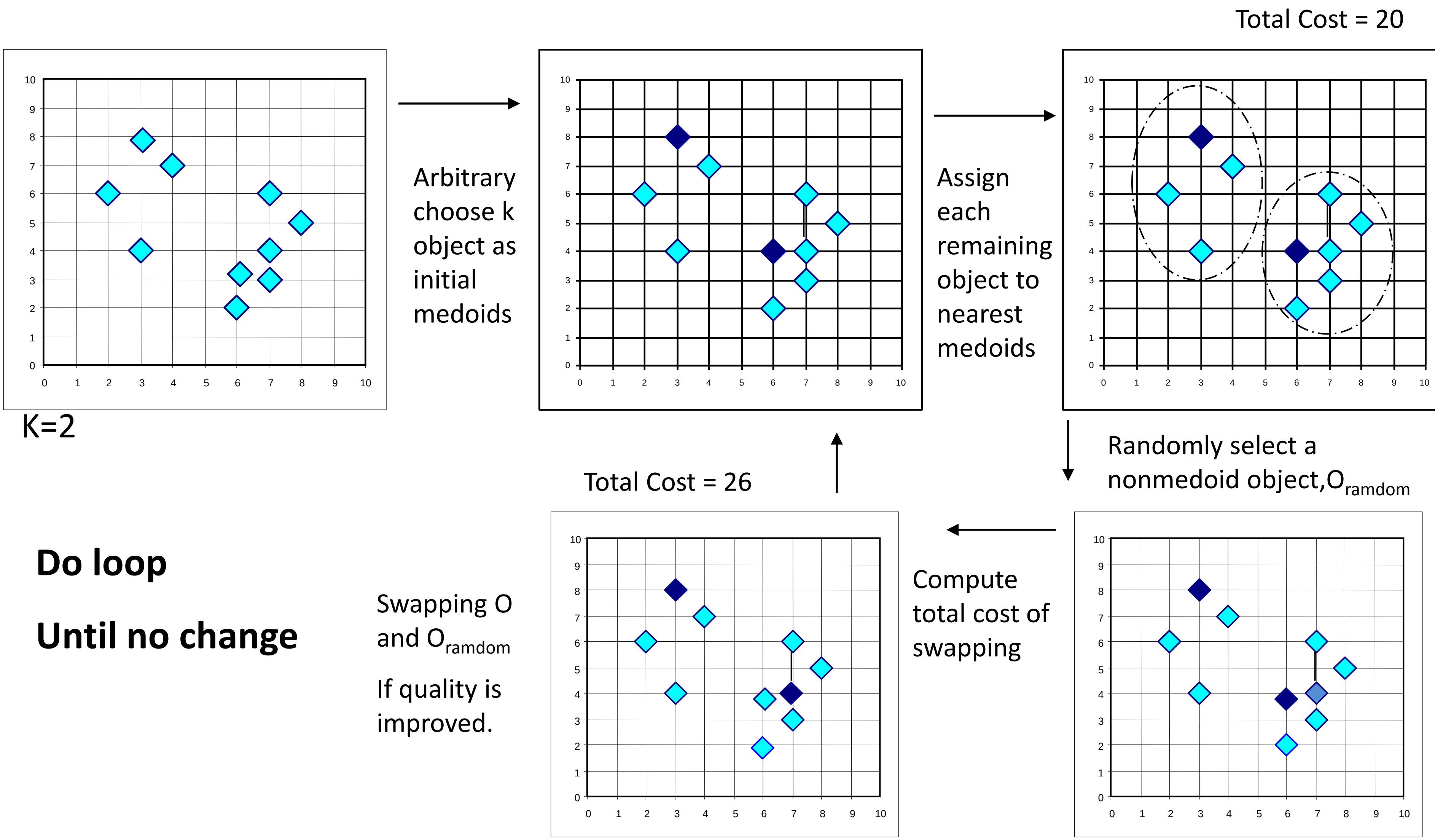


k-medoids

The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets. Time complexity is $O(k(n-k)^2)$ for each iteration where n is # of data objects, k is # of clusters
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

PAM: A Typical K-Medoids Algorithm



K-modes Algorithm

- Handling categorical data: K-modes (Huang'98)
  - Replacing means of clusters with *modes*
    - Given *n* records in cluster, mode is a record made up of the most frequent attribute values
  - Using new dissimilarity measures to deal with categorical objects
- A mixture of categorical and numerical data: K-prototype method

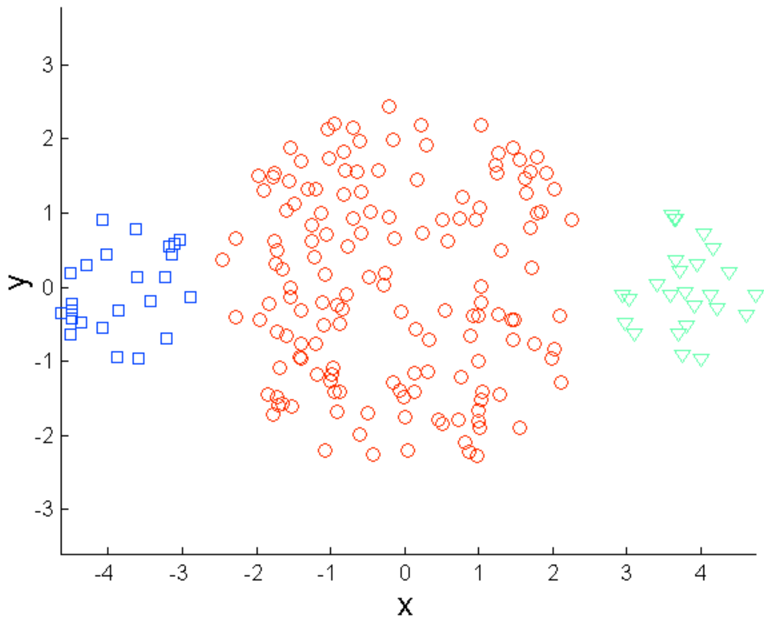
age	income	student	credit_rating
< = 30	high	no	fair
< = 30	high	no	excellent
31…40	high	no	fair
> 40	medium	no	fair
> 40	low	yes	fair
> 40	low	yes	excellent
31…40	low	yes	excellent
< = 30	medium	no	fair
< = 30	low	yes	fair
> 40	medium	yes	fair
< = 30	medium	yes	excellent
31…40	medium	no	excellent
31…40	high	yes	fair

mode = (<=30, medium, yes, fair)

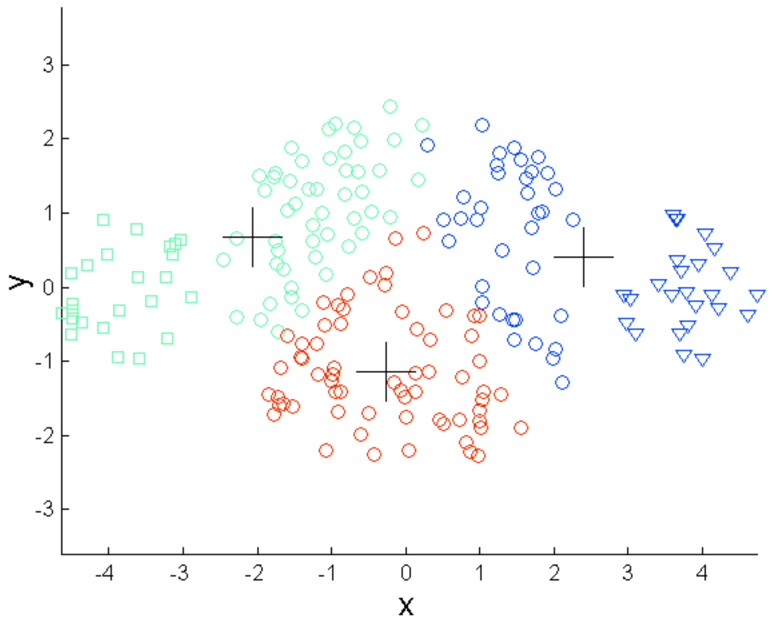
Limitations of K-means

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

Limitations of K-means: Differing Sizes

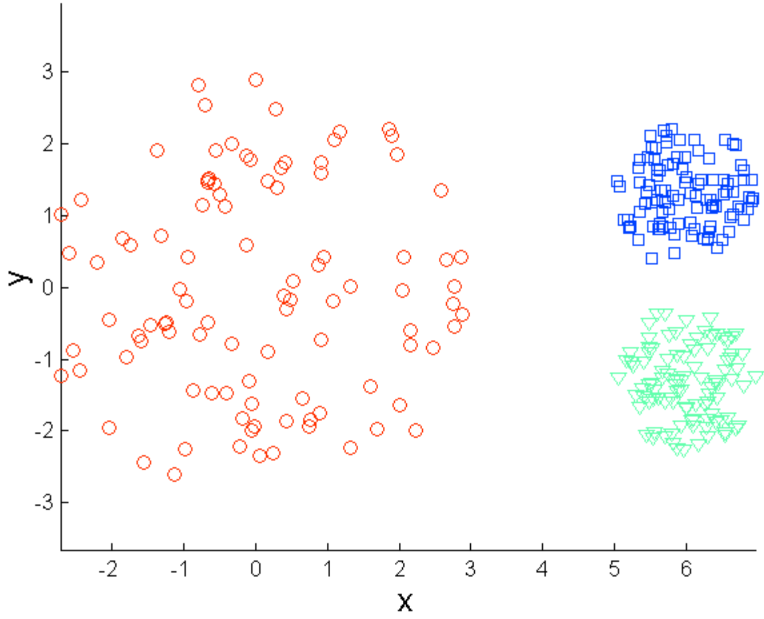


Original Points

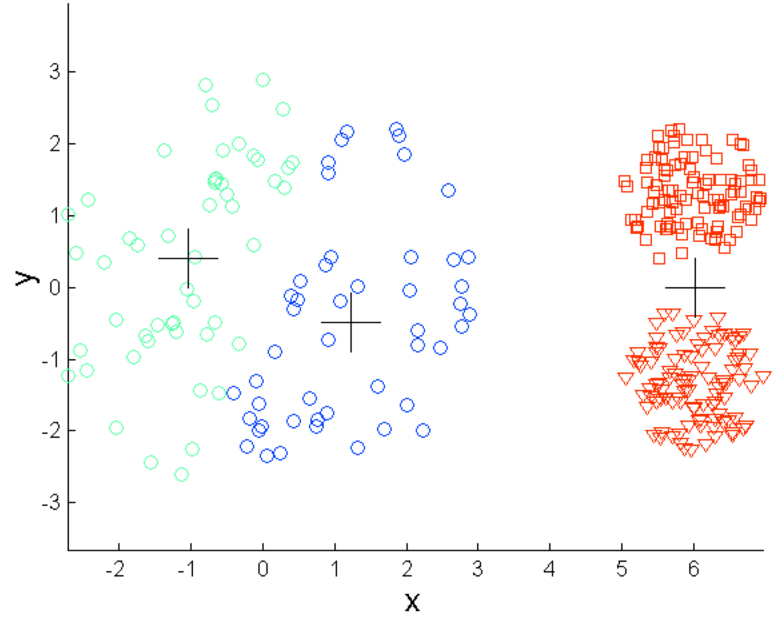


K-means (3 Clusters)

Limitations of K-means: Differing Density

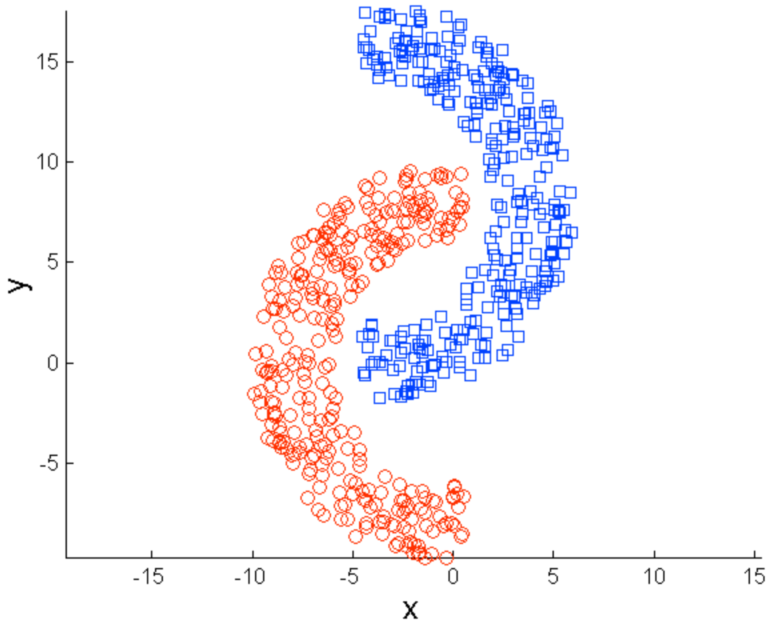


Original Points

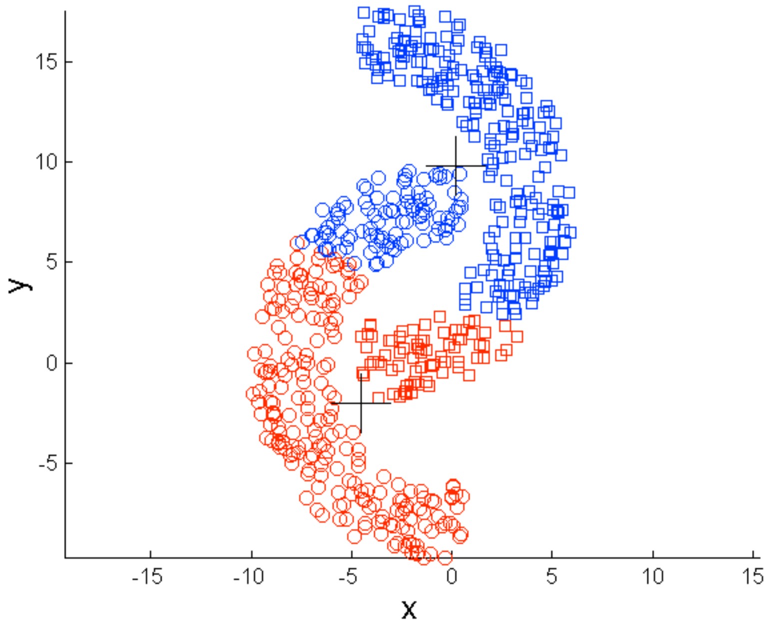


K-means (3 Clusters)

Limitations of K-means: Irregular Shapes

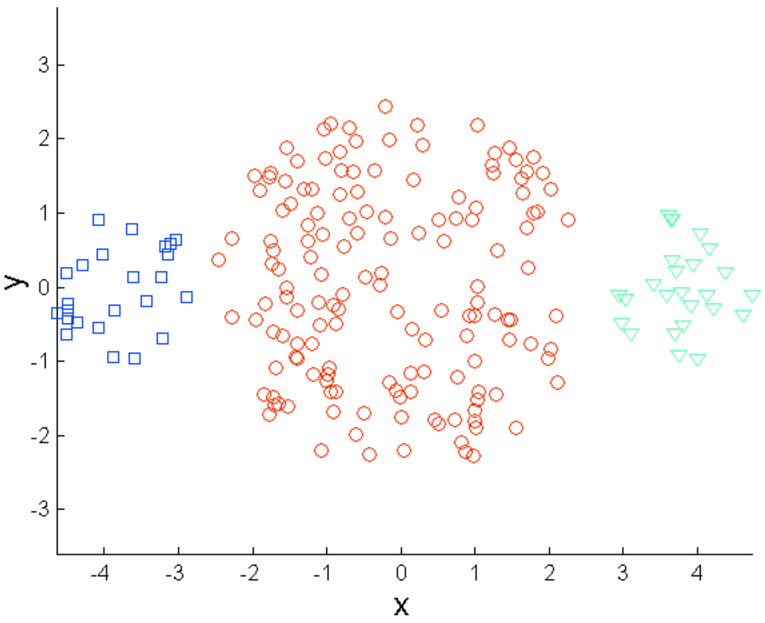


Original Points

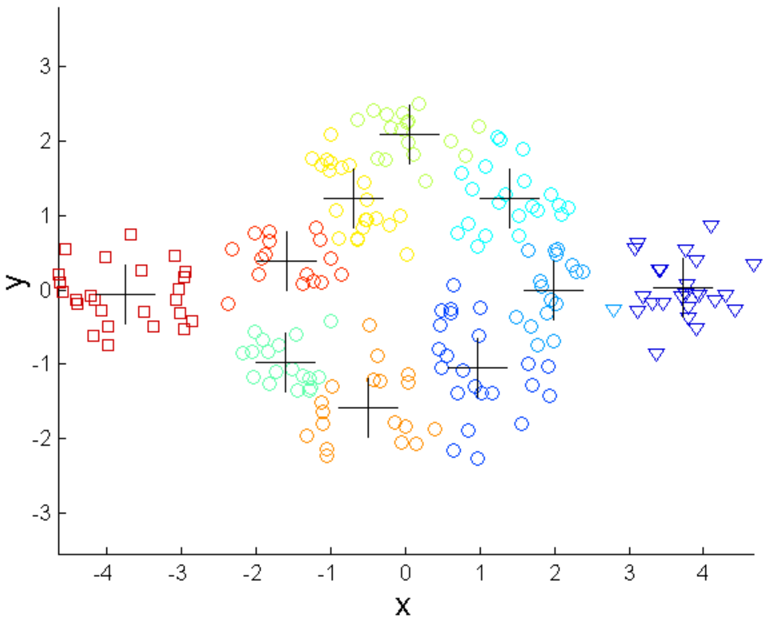


K-means (2 Clusters)

Overcoming K-means Limitations



Original Points

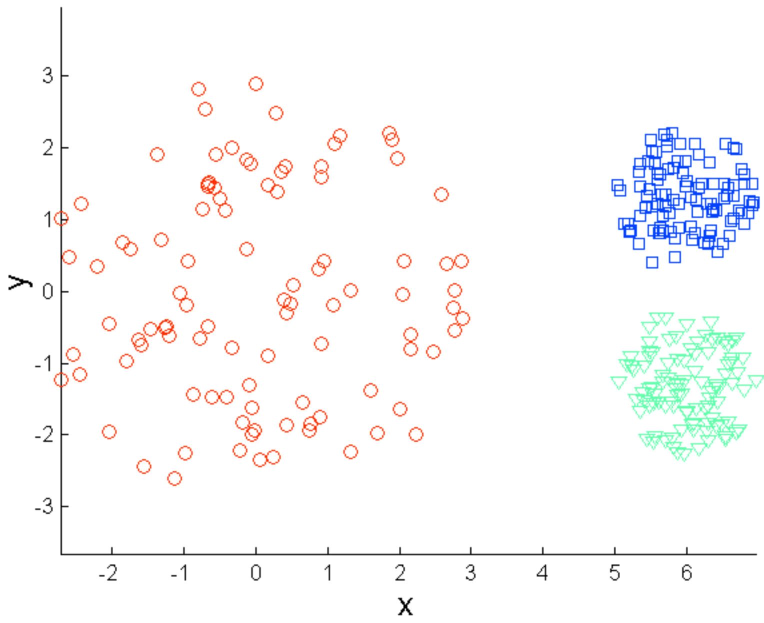


K-means Clusters

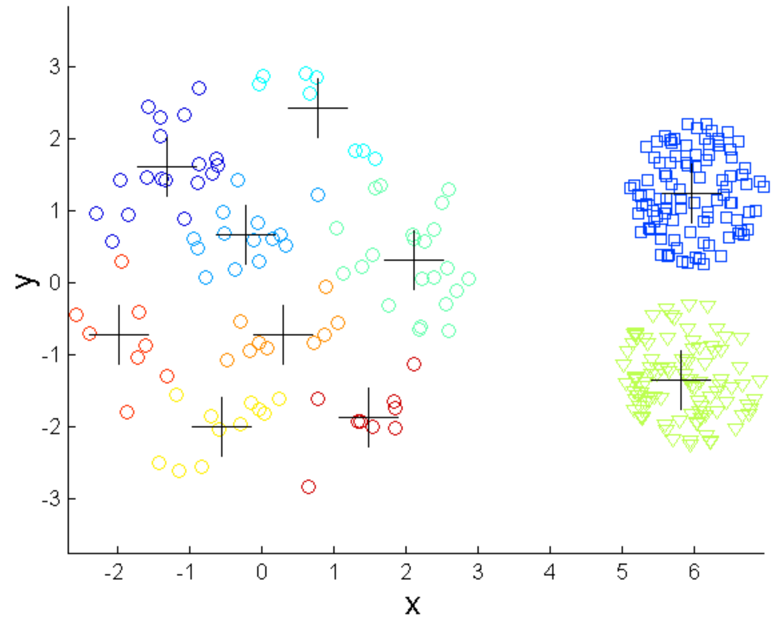
One solution is to use many clusters.

Find parts of clusters, but need to put together.

Overcoming K-means Limitations

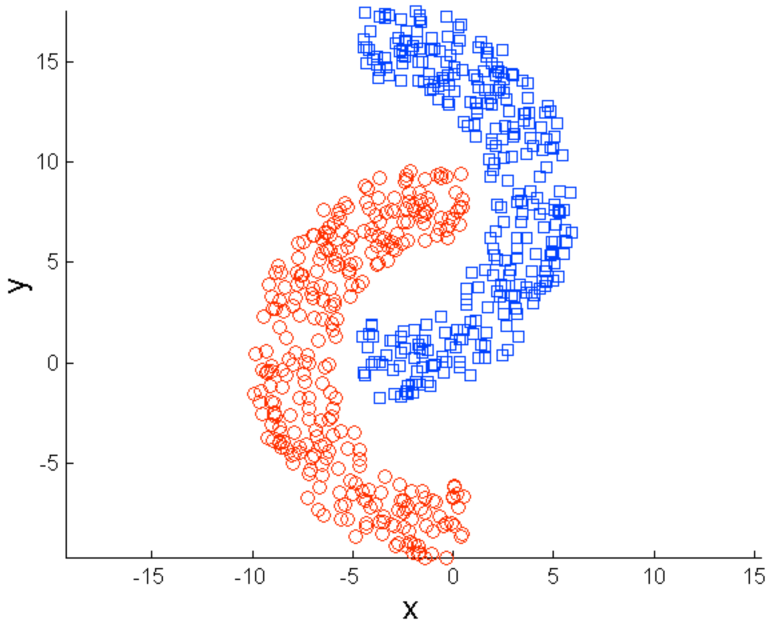


Original Points

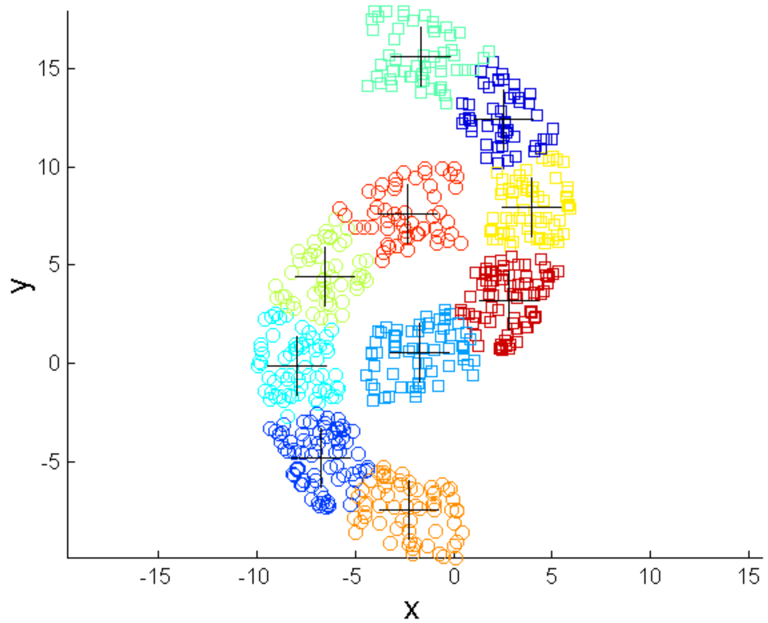


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Take-away Message

- What's partitional clustering?
- How does K-means work?
- How is K-means related to the minimization of SSE?
- What are the strengths and weakness of K-means?
- What are the variants of K-means?