



Pattern Recognition

Cluster Analysis : K-means

CSI 415

Mohammad Imam Hossain | Lecturer, Dept. of CSE | UIU

Supervised Learning

Supervised Machine Learning

- ▶ Supervised learning, also known as supervised machine learning, uses **labeled datasets** to train algorithms that to classify data or predict outcomes accurately.
- ▶ The training dataset includes inputs and correct outputs, which allow the model to learn over time.
- ▶ The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.
- ▶ Supervised learning can be separated into two types of problems when data mining:
 - **Classification** : A supervised machine learning where the learned model/function has a **discrete value** as its output.
 - **Regression** : A supervised machine learning where the learned function has a **continuous real number** as its output.

Supervised Learning – Issue

Supervised Machine Learning

- Supervised learning, also known as supervised machine learning, uses labeled datasets to train algorithms that to classify data or predict outcomes accurately.
- The training dataset includes inputs and correct outputs, which allow the model to learn over time.
- The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized.
- The **labelled datasets** allow supervised learning algorithms to **avoid computational complexity** as they don't need a large training set to produce intended outcomes.
- The supervised learning algorithms tend to be **more accurate** than other unsupervised and semi-supervised learning models, they require upfront human intervention to label the data appropriately.
- **Unsupervised and Semi-supervised learning** can be more appealing alternatives as it can be **time-consuming and costly** to rely on domain expertise to label data appropriately for supervised learning.

Unsupervised Learning

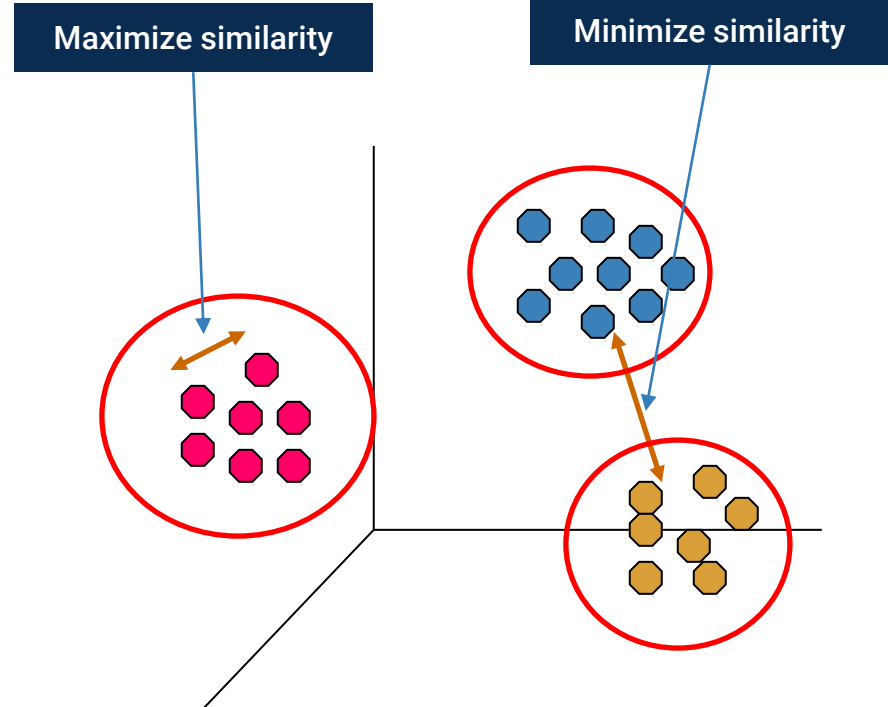
Unsupervised Machine Learning

- ▶ Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster **unlabeled** datasets.
- ▶ These algorithms **discover hidden patterns or data groupings (similarities)** without the need for human intervention (supervision).
- ▶ Unsupervised learning models are utilized for 3 main tasks:
 - Clustering
 - Association rules
 - Dimensionality reduction
- ▶ Challenges
 - Computational complexity due to a high volume of training data.
 - Longer training times.
 - Higher risk of inaccurate results.
 - Human intervention to validate output variables.
 - Lack of transparency into the basis on which data was clustered.

Cluster Analysis

Cluster Analysis

- ▶ Cluster analysis **groups data objects** based only on information found in the data that describes the objects and their relationships.
 - ▶ The goal is that the **objects within a group be similar** (or related) to one another and **different from the objects in other groups**.
 - ▶ The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering.
 - ▶ The terms **Segmentation** and **Partitioning** are sometimes used as synonyms for **Clustering**.
-
- ▶ **Cluster** – group of objects that share some property.
 - ▶ **Clustering** – an entire collection of clusters.



Cluster Analysis

- ▶ The definition of a cluster is imprecise and that the best definition depends on the nature of data and the desired results.



(a) Original points.



(b) Two clusters.



(c) Four clusters.



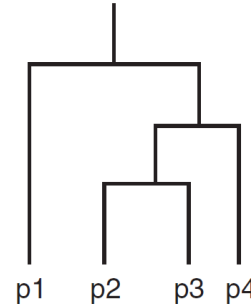
(d) Six clusters.

Different ways of clustering the same set of points.

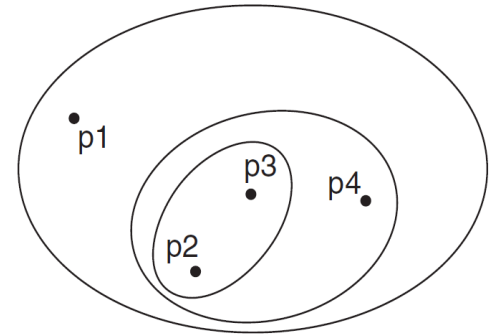
Types of Clustering – Partitional vs Hierarchical

Hierarchical Clustering

- ▶ A set of **nested clusters** that are organized as a hierarchical tree.
- ▶ Each node (cluster) in the tree (except for the leaf nodes) is the union of its children (subclusters), and the root of the tree is the cluster containing all the objects.
- ▶ Often, but not always, the leaves of the tree are singleton clusters of individual data objects.
- ▶ It is often displayed graphically using a tree-like diagram called a **dendrogram**.



Dendrogram



Hierarchical Clustering

Types of Clustering – Partitional vs Hierarchical

Hierarchical Clustering

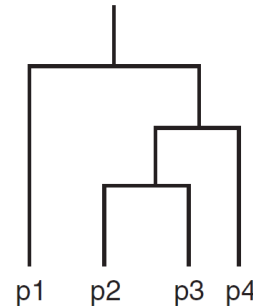
Two basic approaches for generating a hierarchical clustering.

- ▶ **Agglomerative:** Start with the points as individual clusters and, at each step, merge the closest pair of clusters. [Bottom-up approach]

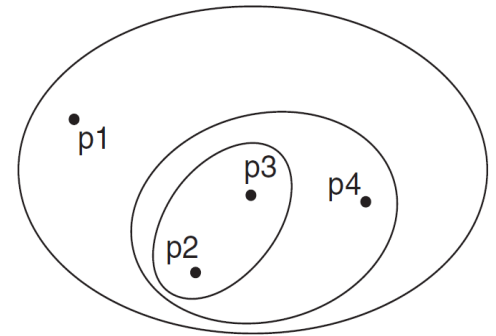
Proximity between clusters – 4 different methods:

- Single Link or MIN
- Complete Link or MAX or CLIQUE
- Group Average
- Ward's Linkage

- ▶ **Divisive:** Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting. [Top-down approach]



Dendrogram

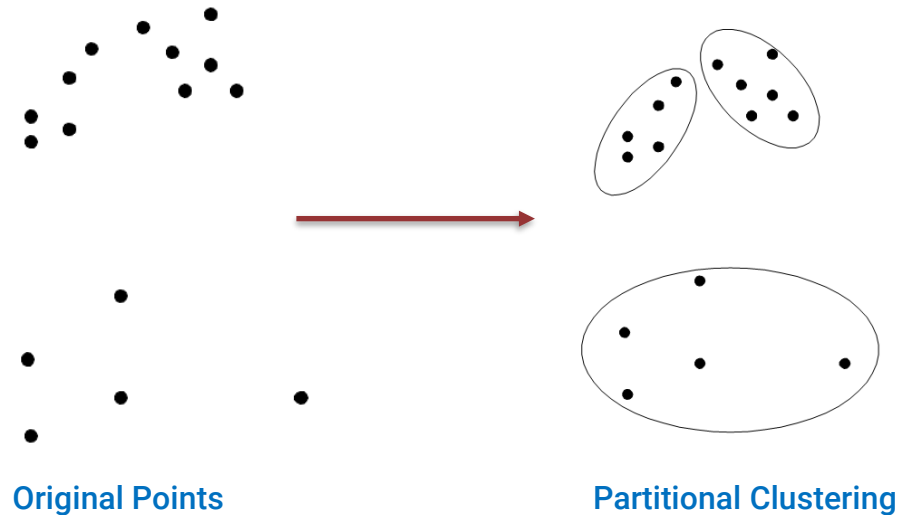


Hierarchical Clustering

Types of Clustering – Partitional vs Hierarchical

Partitional Clustering

A division of the set of data objects into **non-overlapping subsets** (clusters) such that each data object is in exactly one subset.



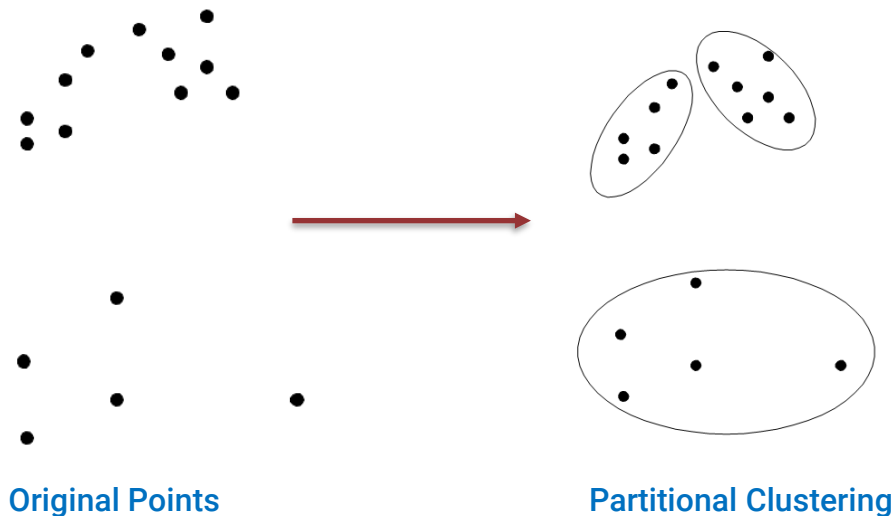
Types of Clustering – Partitional vs Hierarchical

Partitional Clustering

Proximity measurement – quantifies the notion of “closest”

- ▶ Manhattan distance
- ▶ Euclidean distance
- ▶ Cosine similarity
- ▶ Jaccard distance

etc.



Types of Clustering – Exclusive vs Overlapping vs Fuzzy

Exclusive

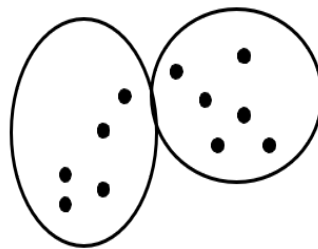
- Assign each object to a single cluster.

Overlapping or Non-exclusive

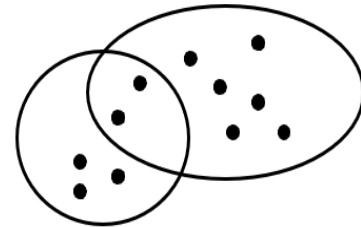
- An object could reasonably be placed in more than one cluster.

Fuzzy (soft, probabilistic)

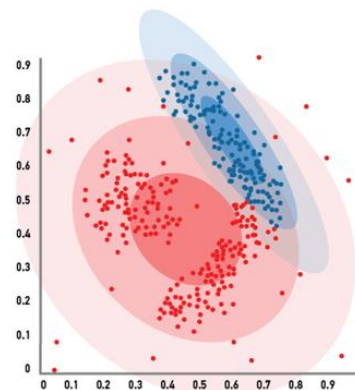
- Every object belongs to every cluster with a membership weight that is between 0 (absolutely doesn't belong) and 1 (absolutely belongs).



Exclusive



Overlapping



Fuzzy

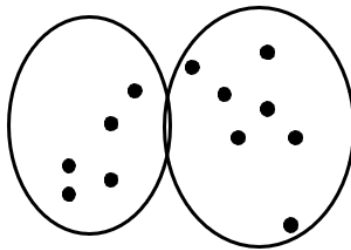
Types of Clustering – Complete vs Partial

Complete

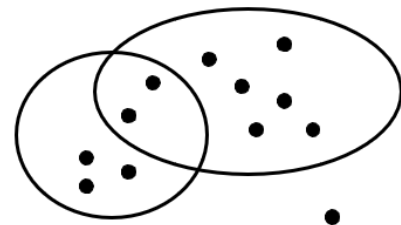
- Assigns each object to a cluster

Partial

- Doesn't assign each object to a cluster.
- The motivation for a partial clustering is that some objects in a data set may not belong to well-defined groups. Many times objects in the data set may represent noise, outliers, or "uninteresting background".



Complete



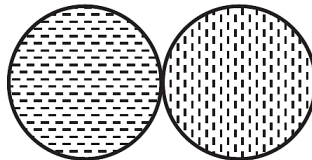
Partial

Types of Clusters

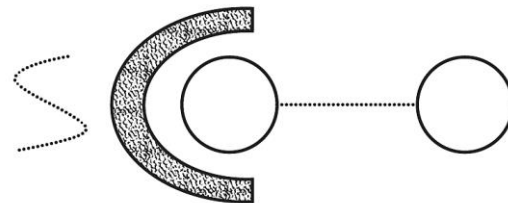


(a) Well-separated clusters. Each point is closer to all of the points in its cluster than to any point in another cluster.

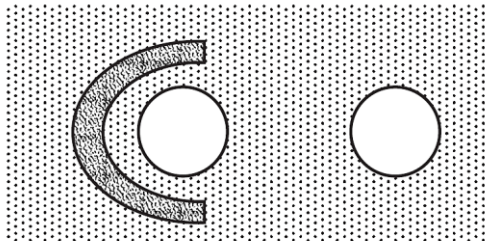
Prototype-based Cluster



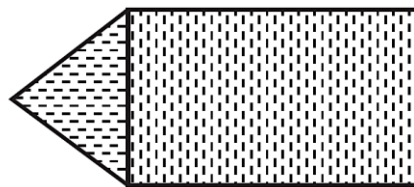
(b) Center-based clusters. Each point is closer to the center of its cluster than to the center of any other cluster.



(c) Contiguity-based clusters. Each point is closer to at least one point in its cluster than to any point in another cluster.



(d) Density-based clusters. Clusters are regions of high density separated by regions of low density.



(e) Conceptual clusters. Points in a cluster share some general property that derives from the entire set of points. (Points in the intersection of the circles belong to both.)

K-means Clustering

- ▶ **Partitional** clustering algorithm.
- ▶ All the clusters are **prototype (center) based**.
- ▶ Each cluster is associated with a **centroid**.
- ▶ Each point is assigned to the cluster with the closest centroid, and each collection of points assigned to a centroid is a cluster.
- ▶ **K**, a user-specified parameter, represents the number of desired clusters.

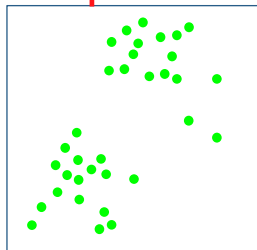
K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

$k=2$

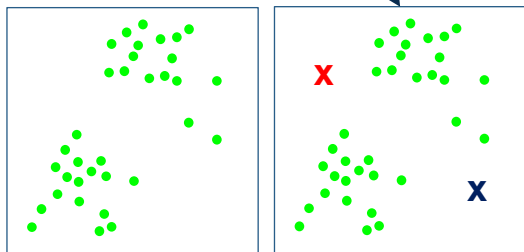
input



K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

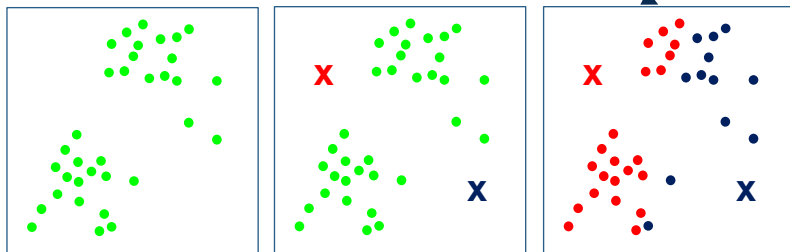
- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

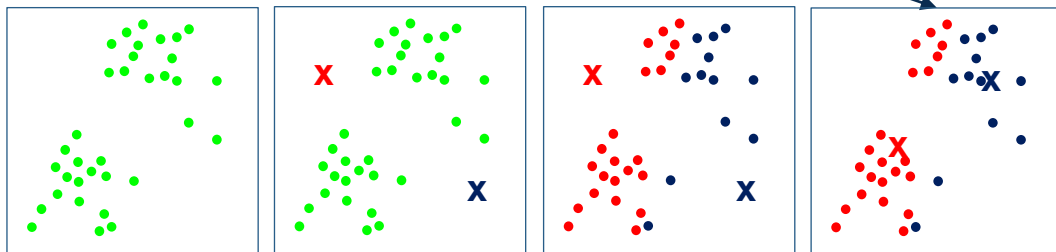
- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Algorithm


Algorithm 8.1 Basic K-means algorithm.

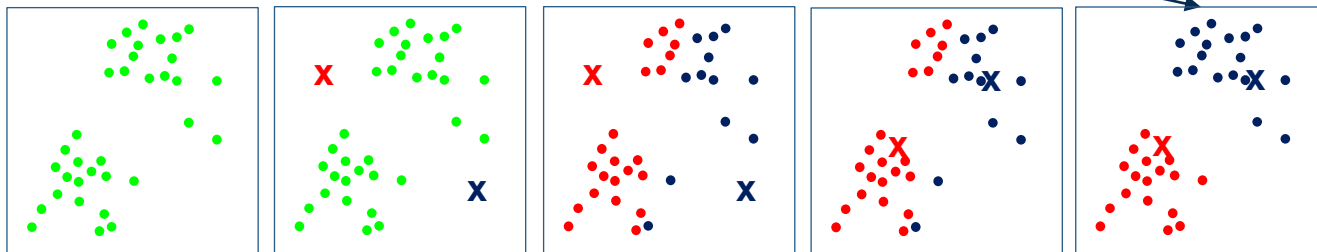
- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

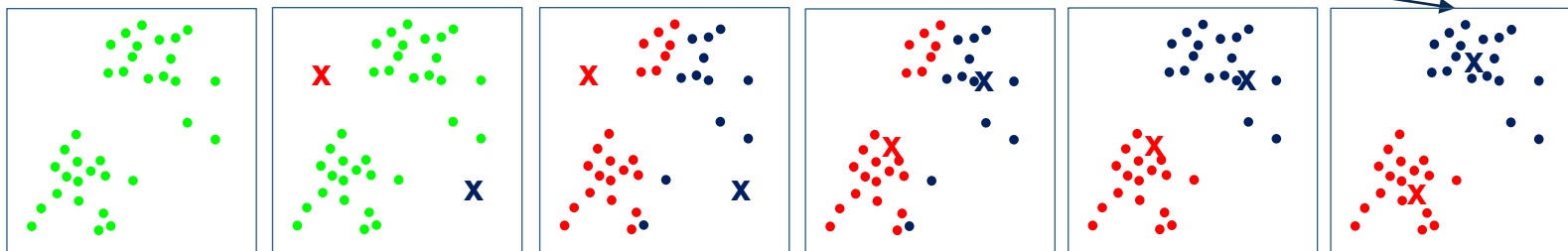
- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid. 
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

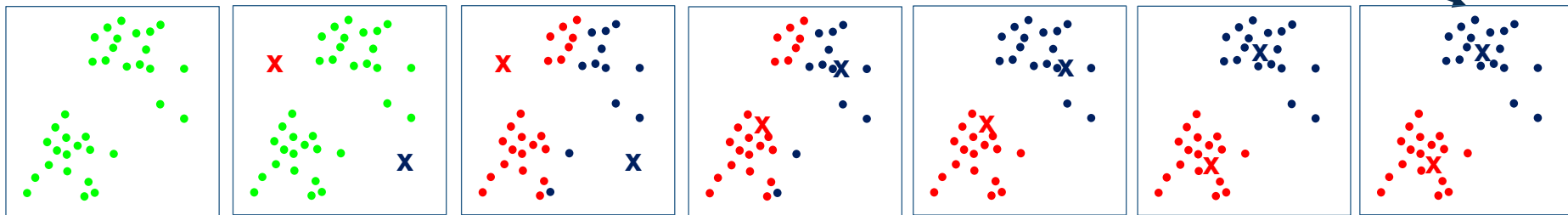
- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

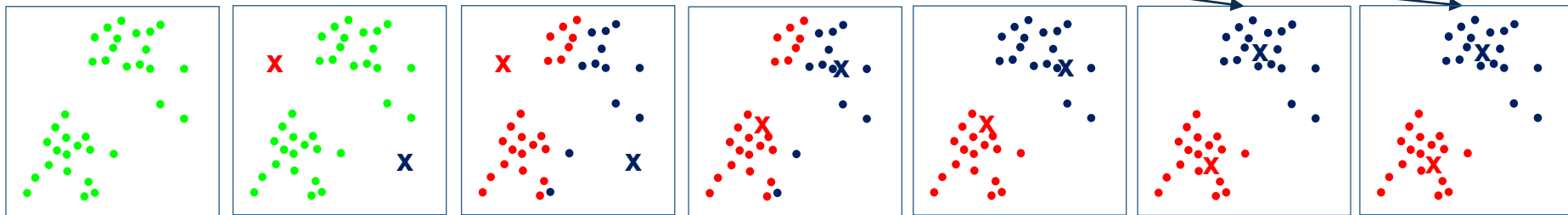
- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Algorithm

Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-



K-means Clustering – Proximity Measurement

Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

Proximity Measurement – quantifies the notion of “closest”

- ▶ Manhattan distance
- ▶ Euclidean distance
- ▶ Cosine similarity
- ▶ Jaccard distance etc.

K-means Clustering – Objective Function

Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

Centroid Measurement

- It can vary depending on the **proximity measure** for the data and the **goal (objective function)** of the clustering.

Assumptions

- Proximity measure:

Squared Euclidean Distance, $dist(q, p)^2 = \sum_{i=1}^d (q_i - p_i)^2$, d = dimension of the data object

- Goal or Objective function (to minimize):

Sum of the Squared Error, $SSE = \sum_{i=1}^K \sum_{x \in C_i} dist(c_i, x)^2$, c_i = centroid of cluster C_i

K-means Clustering – Centroid Measurement

Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

Centroid Measurement

- Based on the assumptions (Euclidean distance and SSE), it can be shown that the centroid that minimizes the SSE of the cluster is the **mean**.

Centroid of cluster C_i , $c_i = \frac{1}{m_i} \sum_{x \in C_i} x$, $m_i = \text{number of objects in the } i^{\text{th}} \text{ cluster } C_i$

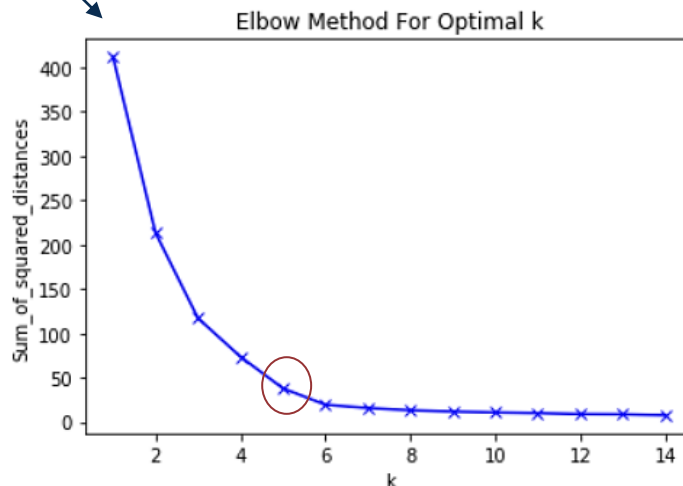
Example

If 3 points of a cluster are (1,1), (2,3), and (6,2) then the centroid of the cluster is $\left(\frac{1+2+6}{3}, \frac{1+3+2}{3}\right) = (3, 2)$

K-means Clustering – Optimal Value of K

Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
- 2: **repeat**
- 3: Form K clusters by assigning each point to its closest centroid.
- 4: Recompute the centroid of each cluster.
- 5: **until** Centroids do not change.



K-means Clustering – Example

Problem

Consider 8 data points: $P_1(2, 10)$, $P_2(2, 5)$, $P_3(8, 4)$, $P_4(5, 8)$, $P_5(7, 5)$, $P_6(6, 4)$, $P_7(1, 2)$, $P_8(4, 9)$

Now cluster these eight points into three clusters.

K-means Clustering – Example

Solution

Let's assume the Initial centroids for the 3 clusters are, $P_1(2, 10)$, $P_4(5, 8)$ and $P_7(1, 2)$

Points	Cluster 1 (2, 10)	Cluster 2 (5, 8)	Cluster 3 (1, 2)	Assigned Cluster
$P_1(2, 10)$	0	13	65	Cluster 1
$P_2(2, 5)$	25	18	10	Cluster 3
$P_3(8, 4)$	72	25	53	Cluster 2
$P_4(5, 8)$	13	0	52	Cluster 2
$P_5(7, 5)$	50	13	45	Cluster 2
$P_6(6, 4)$	52	17	29	Cluster 2
$P_7(1, 2)$	65	52	0	Cluster 3
$P_8(4, 9)$	5	2	58	Cluster 2

Iteration 1

Updated Centroids

Cluster 1:

$P_1(2, 10)$
Centroid = (2, 10)

Cluster 2:

$P_3(8, 4)$, $P_4(5, 8)$, $P_5(7, 5)$, $P_6(6, 4)$,
 $P_8(4, 9)$
Centroid = (6, 6)

Cluster 3:

$P_2(2, 5)$, $P_7(1, 2)$
Centroid = (1.5, 3.5)

K-means Clustering – Example

Solution

Points	Cluster 1 (2, 10)	Cluster 2 (6, 6)	Cluster 3 (1.5, 3.5)	Assigned Cluster
$P_1(2, 10)$	0	32	42.5	Cluster 1
$P_2(2, 5)$	25	17	2.5	Cluster 3
$P_3(8, 4)$	72	8	42.5	Cluster 2
$P_4(5, 8)$	13	5	32.5	Cluster 2
$P_5(7, 5)$	50	2	32.5	Cluster 2
$P_6(6, 4)$	52	4	20.5	Cluster 2
$P_7(1, 2)$	65	41	2.5	Cluster 3
$P_8(4, 9)$	5	13	36.5	Cluster 1

Updated Centroids

Cluster 1:

$P_1(2, 10), P_8(4, 9)$
Centroid = (3, 9.5)

Cluster 2:

$P_3(8, 4), P_4(5, 8), P_5(7, 5), P_6(6, 4)$
Centroid = (6.5, 5.25)

Cluster 3:

$P_2(2, 5), P_7(1, 2)$
Centroid = (1.5, 3.5)

Iteration 2

Similarly repeat the steps until centroids do not change

K-means Clustering – Initial Centroid

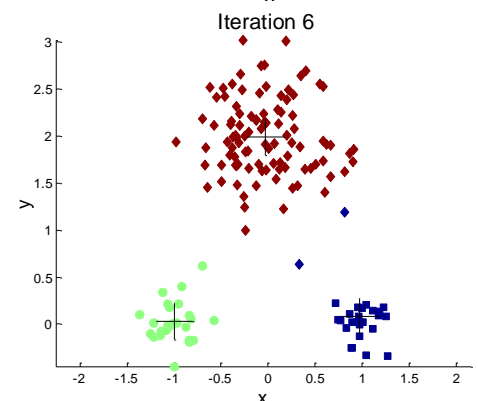
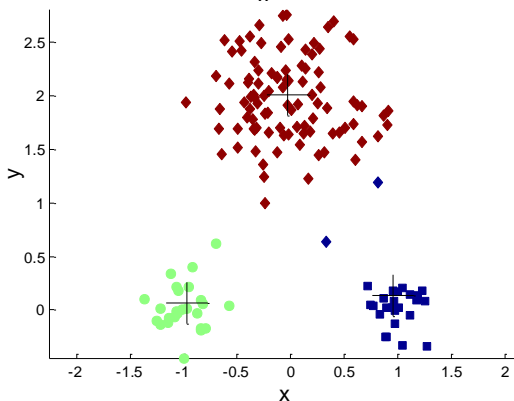
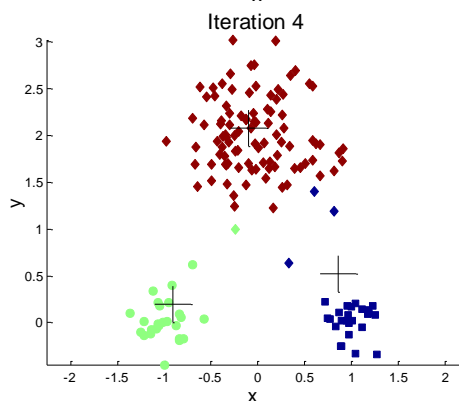
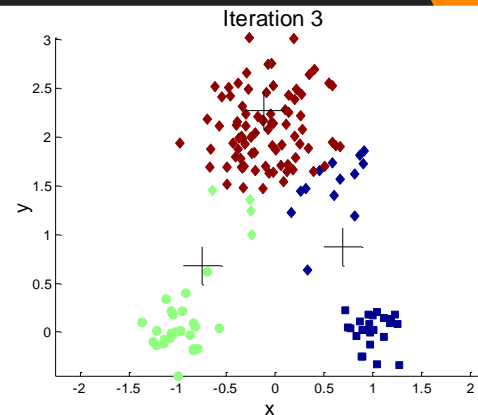
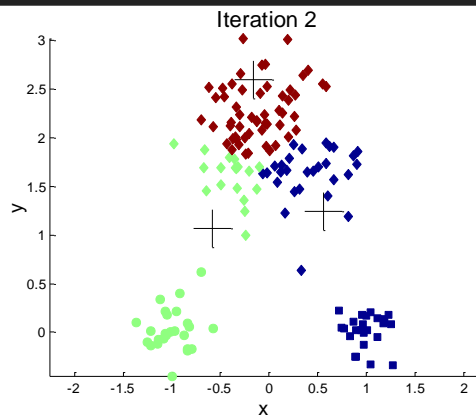
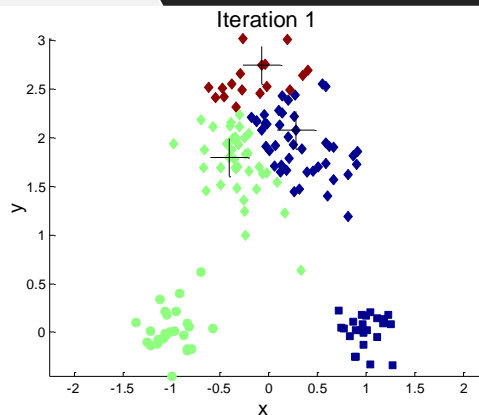
Algorithm 8.1 Basic K-means algorithm.

- 1: Select K points as initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning each point to its closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** Centroids do not change.
-

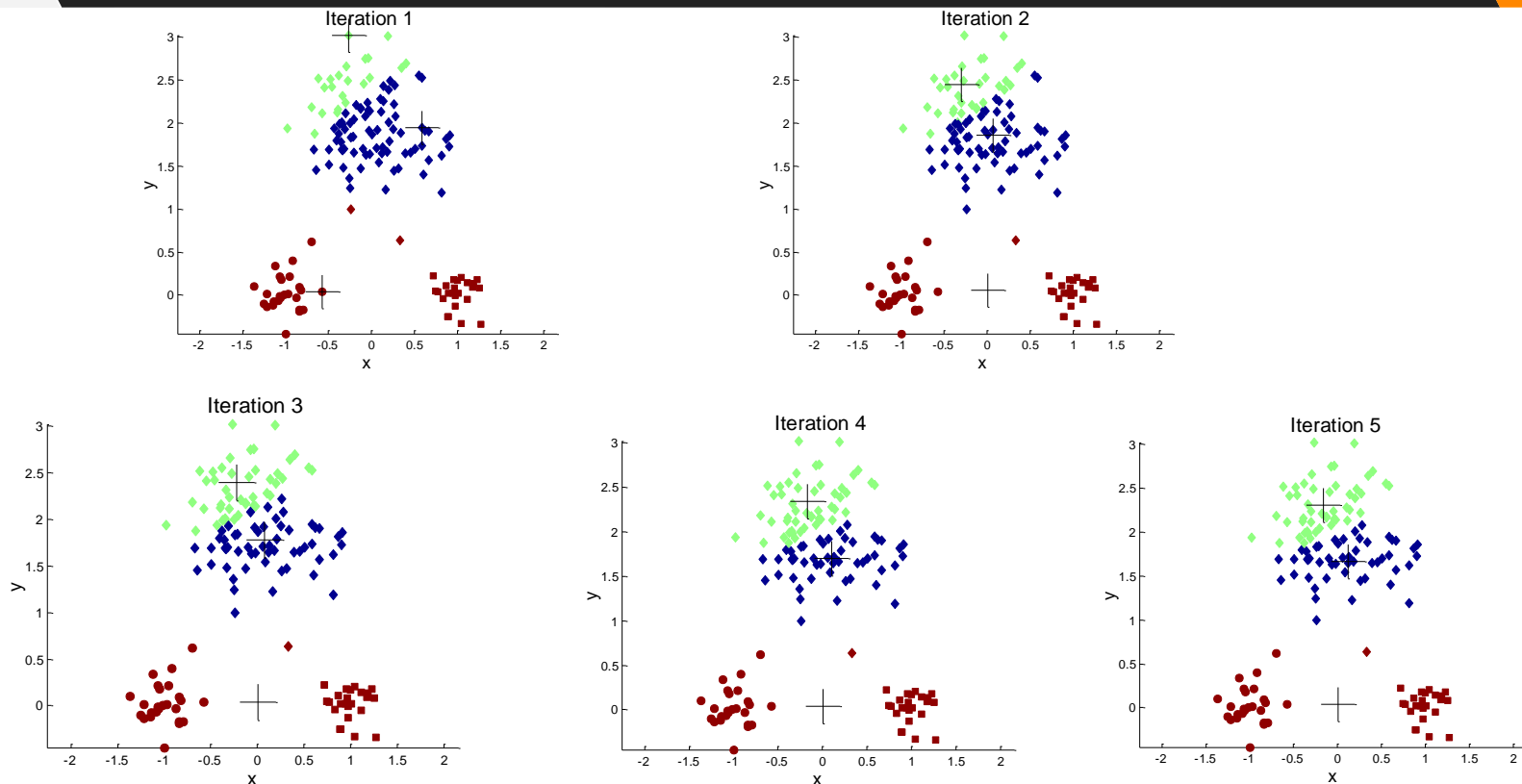
Choosing Initial Centroids

- When random initialization of centroids is used, different runs of K-means typically produce different total SSEs.
- Choosing the proper initial centroids is the key step of the basic K-means procedure.
- Randomly selected initial centroids may be poor.

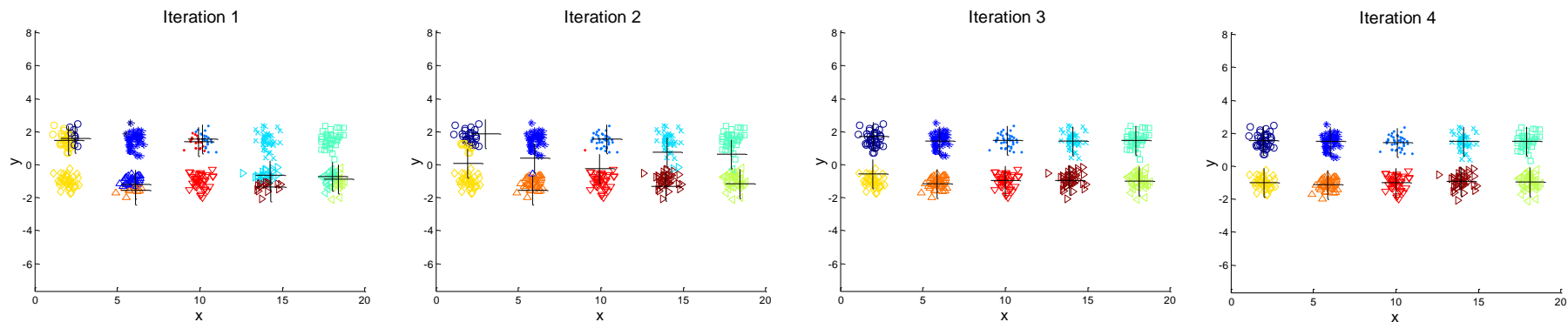
K-means Clustering – Initial Centroid (Optimal SSE)



K-means Clustering – Initial Centroid (Sub-optimal SSE)

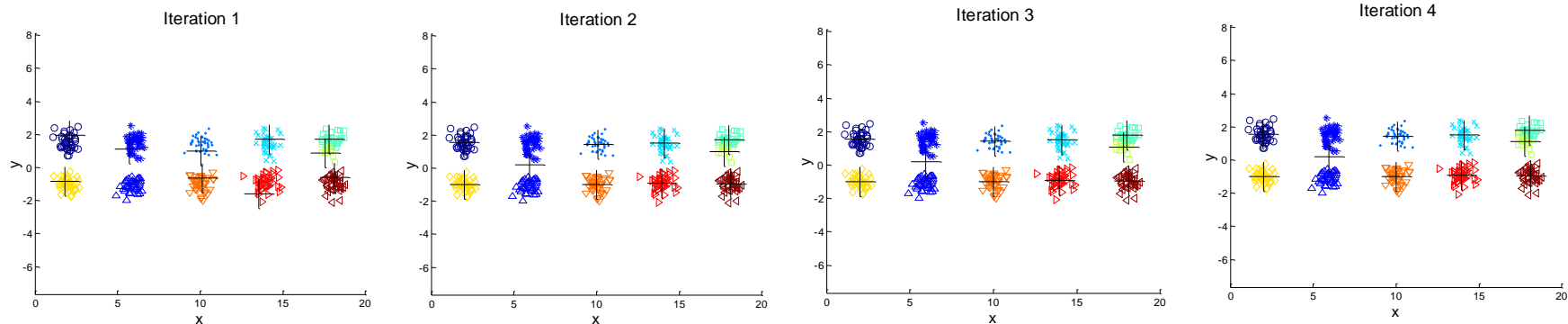


K-means Clustering – Initial Centroid Problem



Optimal clustering will be obtained as long as two initial centroids fall anywhere in a pair of clusters.

K-means Clustering – Initial Centroid Problem



As the number of clusters becomes larger, it is increasingly likely that at least one pair of clusters will have only one initial centroid.

K-means Clustering – Initial Centroid Problem Solution

► Multiple runs

- Perform multiple runs, each with a different set of randomly chosen initial centroids, and then select the set of clusters with the minimum SSE.
 - **May not work very well**, depending on the data set and the number of clusters sought.
- Take a sample of points and cluster them using a **Hierarchical clustering** technique.
- Extract K clusters from the hierarchical clustering and the centroids of those clusters are used as the initial centroids of the basic K-means clustering.
 - **Limitations:** Practical only if the sample is relatively small and K is relatively small compared to the sample size.

More Efficient Approach

- Bisecting K-means (less susceptible to initialization problems)
- Postprocessing

K-means Clustering – Bisecting K-means

Bisecting K-means

- ▶ To obtain K clusters, split the set of all points into two clusters, select one of these clusters to split, and so on, until K clusters have been produced.
- ▶ Finally, use the obtained K centroids from this approach as the initial centroid of global basic K-means clustering algorithm.

Algorithm 8.2 Bisecting K-means algorithm.

- 1: Initialize the list of clusters to contain the cluster consisting of all points.
 - 2: **repeat**
 - 3: Remove a cluster from the list of clusters.
 - 4: {Perform several “trial” bisections of the chosen cluster.}
 - 5: **for** $i = 1$ to *number of trials* **do**
 - 6: Bisect the selected cluster using basic K-means.
 - 7: **end for**
 - 8: Select the two clusters from the bisection with the lowest total SSE.
 - 9: Add these two clusters to the list of clusters.
 - 10: **until** Until the list of clusters contains K clusters.
-

K-means Clustering – Bisecting K-means

Bisecting K-means

- ▶ To obtain K clusters, split the set of all points into two clusters, select one of these clusters to split, and so on, until K clusters have been produced.
- ▶ Finally, use the obtained K centroids from this approach as the initial centroid of global basic K-means clustering algorithm.

Algorithm 8.2 Bisecting K-means algorithm.

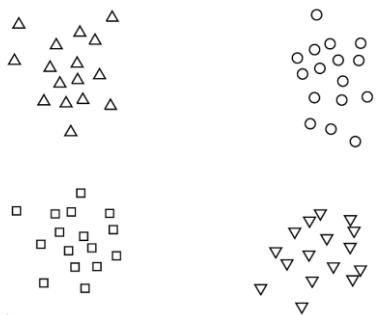
```
1: Initialize the list of clusters to contain the cluster consisting of all points.  
2: repeat  
3:   Remove a cluster from the list of clusters.  
4:   {Perform several “trial” bisections of the chosen cluster.}  
5:   for  $i = 1$  to number of trials do  
6:     Bisect the selected cluster using basic K-means.  
7:   end for  
8:   Select the two clusters from the bisection with the lowest total SSE.  
9:   Add these two clusters to the list of clusters.  
10: until Until the list of clusters contains  $K$  clusters.
```

Which cluster to split !!

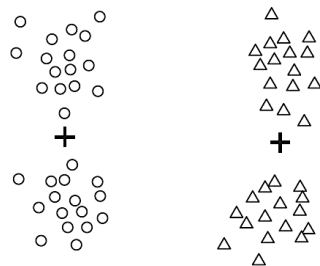
Different approaches

- ▶ Choose the largest cluster at each step.
- ▶ Choose the one with the largest SSE.
- etc.

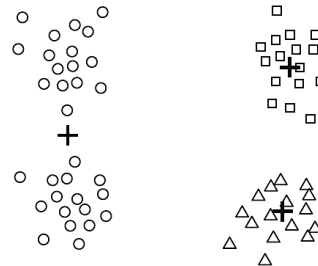
K-means Clustering – Bisecting K-means Example



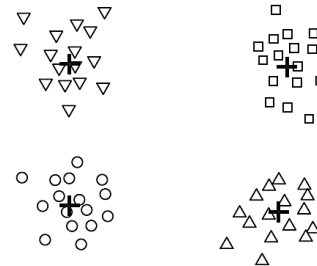
(a) Initial points.



(a) Iteration 1.



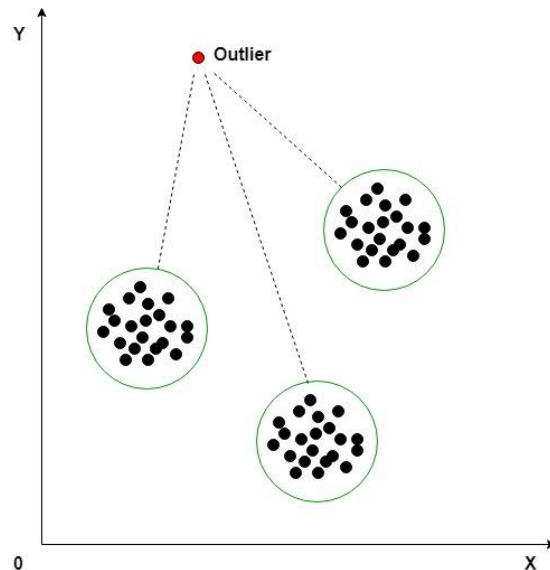
(b) Iteration 2.



(c) Iteration 3.

K-means Clustering – Issues

- ▶ Basic K-means algorithm can yield **empty clusters**.
 - **Solution:** choose a replacement centroid
- ▶ **Outliers Influence**
 - An outlier is a data point that is noticeably different from the rest. They can be caused by measurement or execution error.
 - When outliers are present, the resulting cluster centroids may not be as representative as they otherwise would be and thus, the SSE will be higher as well.
 - **Solution:**
 - **Preprocessing:** discover and eliminate outliers.
 - **Postprocessing:** Track of the SSE contributed by each point and eliminate those points with unusually high contributions. Also eliminate small clusters since they frequently represent groups of outliers.



THANKS!

Any questions?

You can find me at imam@cse.uiu.ac.bd