Cluster analysis

- Cluster analysis groups data objects based only on the attributes of the data.
- The main objective is that
  - The objects within a group be similar to one another and
  - They are different from the objects in the other groups.

Cluster analysis

- Cluster analysis is important in the following areas:
  - Biology
  - Medicine
  - Information retrieval
  - Business

Cluster analysis

- Cluster analysis provides an abstraction from individual data objects to the clusters in which those data objects reside.
- Some clustering techniques characterize each cluster in terms of a cluster prototype.
- The prototype is a data object that is representative of the other objects in the cluster.

Different types of clusterings

- We consider the following types of clusterings
  - Partitional versus hierarchical
  - Exclusive versus fuzzy
  - Complete versus partial

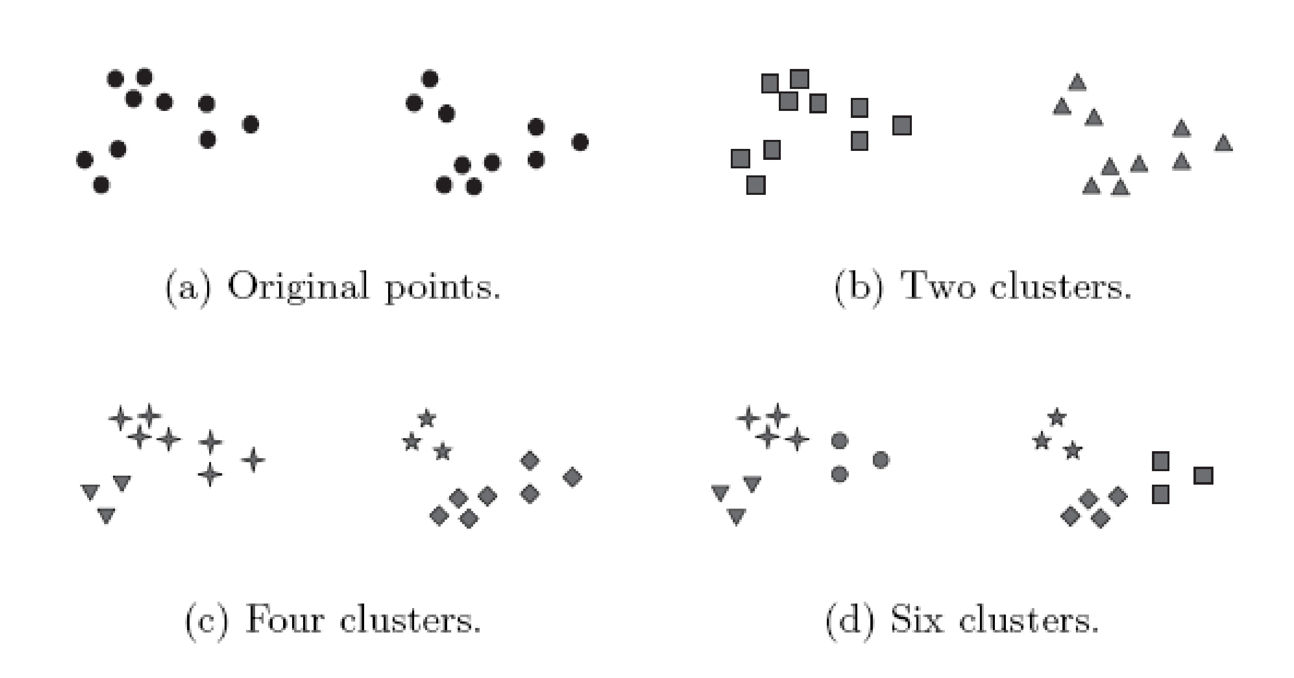
Partitional versus hierarchical

- A partitional clustering is a division of the set of data objects into subsets (clusters).
- A hierarchical clustering is a set of nested clusters that are organized as a tree.
  - Each node (cluster) in the tree (except for the leaf nodes) is the union of its children (sub-clusters).
  - 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.

Partitional versus hierarchical

- The following figures form a hierarchical (nested) clustering with 1, 2, 4 and 6 clusters at each level.
- A hierarchical clustering can be viewed as a sequence of partitional clusterings.
- A partitional clustering can be obtained by taking any member of that sequence, i.e. by cutting the hierarchical tree at a certain level.

Partitional versus hierarchical



Exclusive versus fuzzy

- In an exclusive clustering, each object is assigned to a single cluster.
- However, there are many situations in which a point could reasonably be placed in more than one cluster.

Exclusive versus fuzzy

- In a fuzzy clustering, every object belongs to every cluster with a membership weight that is between
  - 0 (absolutely does not belong) and
  - 1 (absolutely belongs).
- This approach is useful for avoiding the arbitrariness of assigning an object to only one cluster when it is close to several.
- A fuzzy clustering can be converted to an exclusive clustering by assigning each object to the cluster in which its membership value is the highest.

Complete versus partial

- A complete clustering assigns every object to a cluster.
- A partial clustering does not assign every object to a cluster.
- The motivation of partial clustering is that some objects in a data set may not belong to well-defined groups.
- Instead, they may represent noise or outliers.

K-means

- K-means is a prototype-based clustering technique which creates a one-level partitioning of the data objects.
- Specifically, K-means defines a prototype in terms of the centroid of a group of points.
- K-means is typically applied to objects in an n-dimensional space.

K-means

- The basic K-means algorithm is summarized below

1. Select K points as initial centroids
2. Repeat
   a. Form K clusters by assigning each point to its closest centroid.
   b. Re-compute the centroid of each cluster.
3. Until centroids do not change.

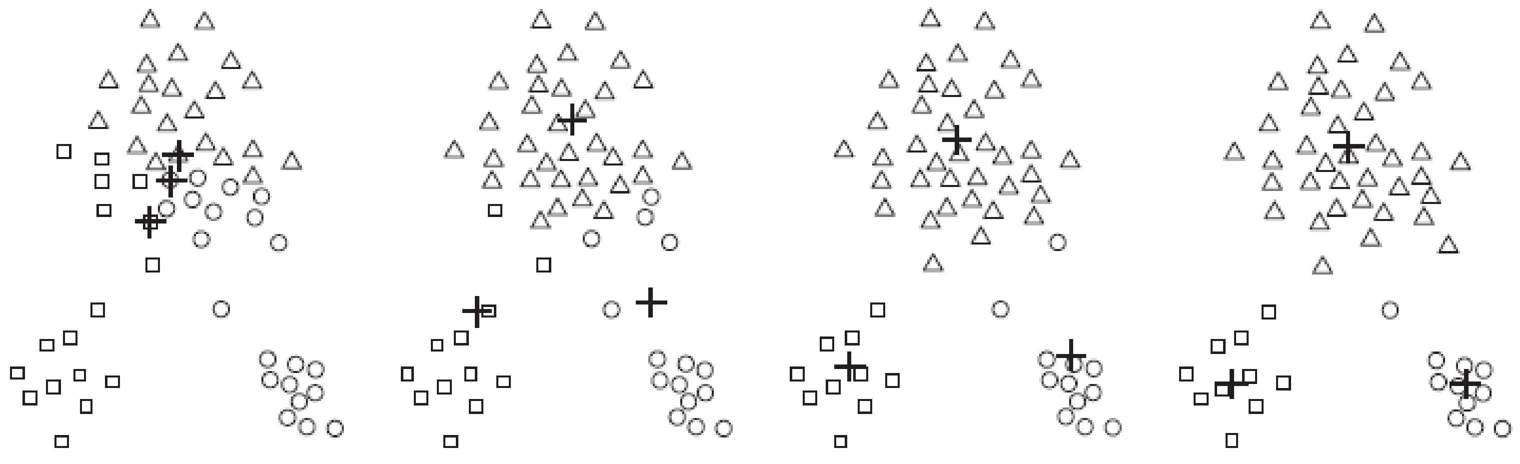
K-means

- We first choose K initial centroids, where K is a user-defined parameter, namely, the number of clusters desired.
- Each point is then assigned to the closest centroid.
- Each collection of points assigned to a centroid is a cluster.
- The centroid of each cluster is then updated based on the points assigned to the cluster.
- We repeat the assignment and update steps until the centroids remain the same.

K-means

- These steps are illustrated in the following figures.
- Starting from three centroids, the final clusters are found in four assignment-update steps.

K-means



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

K-means

- Each sub-figure shows
  - The centroids at the start of the iteration and
  - The assignment of the points to those centroids.
- The centroids are indicated by the “+” symbol.
- All points belonging to the same cluster have the same marker shape.

K-means

- In the first step, points are assigned to the initial centroids, which are all in the largest group of points.
- After points are assigned to a centroid, the centroid is then updated.
- In the second step
  - Points are assigned to the updated centroids and
  - The centroids are updated again.

K-means

- We can observe that two of the centroids move to the two small groups of points at the bottom of the figures.
- When the K-means algorithm terminates, the centroids have identified the natural groupings of points.

Distance measure

- To assign a point to the closest centroid, we need a measure that quantifies the notion of "closest".
- Euclidean (L_2) distance is often used for this purpose.

Distance measure

- The goal of clustering is typically expressed by an objective function.
- We consider the case where Euclidean distance is used.
- For our objective function, which measures the quality of a clustering solution, we can use the sum of the squared error (SSE).

Distance measure

- We calculate the Euclidean distance of each data point to its associated centroid.
- We then compute the total sum of the squared distances, which is also known as the sum of the squared error (SSE).
- A small value of SSE means that the prototypes (centroids) of this clustering are good representations of the points in their clusters.

Distance measure

- The SSE is defined as follows:
  - $$SSE = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{c}_i)^2$$
- In this equation
  - $\mathbf{x}$ is a data object.
  - C_i is the i-th cluster.
  - $\mathbf{c}_i$ is the centroid of cluster C_i.
  - d is the Euclidean (L_2) distance between two objects.

Distance measure

- It can be shown that the mean of the data points in the cluster minimizes the SSE of the cluster.
- The centroid (mean) of the i-th cluster is defined as
  - $$\mathbf{c}_i = \frac{1}{m_i} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$
- In this equation, m_i is the number of objects in the i-th cluster.

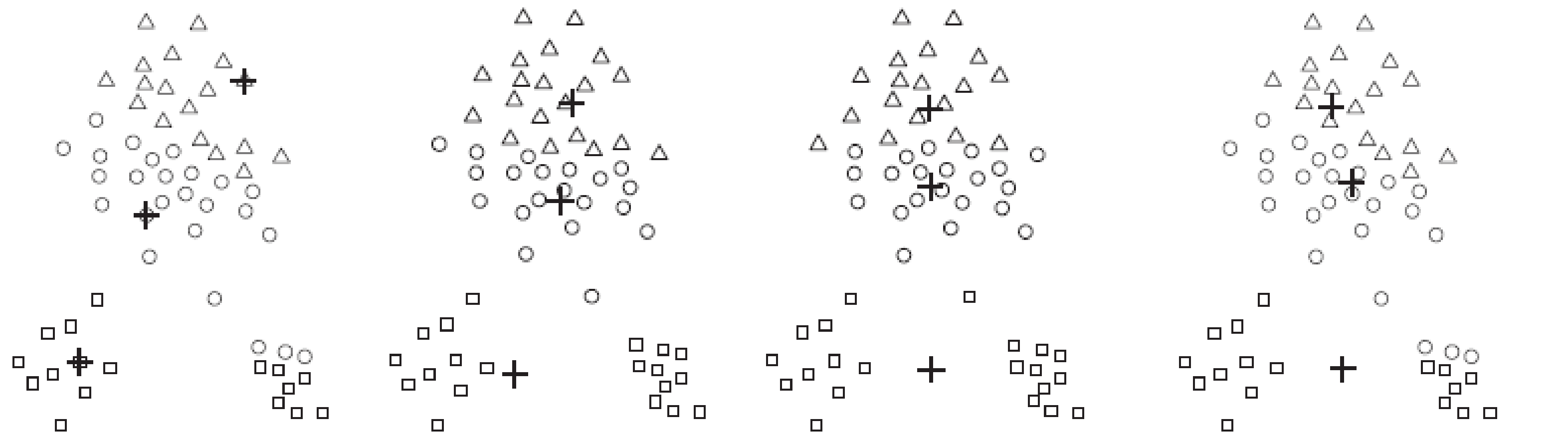
Distance measure

- Steps 2a and 2b of the K-means algorithm attempt to minimize the SSE.
- Step 2a forms clusters by assigning each point to its nearest centroid, which minimizes the SSE for the given set of centroids.
- Step 2b recomputes the centroids so as to further minimize the SSE.

Choosing initial centroids

- Choosing the proper initial centroids is the key step of the basic K-means procedure.
- A common approach is to choose the initial centroids randomly.
- However, randomly selected initial centroids may be poor choices.
- This is illustrated in the following figures.

Choosing initial centroids



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Choosing initial centroids

- One technique that is commonly used to address the problem of choosing initial centroids is to perform multiple runs.
- Each run uses a different set of initial centroids.
- We then choose the set of clusters with the minimum SSE.

Outliers

- Outliers can influence the clusters that are found.
- When outliers are present, the resulting cluster centroids may not be as representative as they otherwise would be.
- Because of this, it is often useful to discover outliers and eliminate them beforehand.

Outliers

- To identify the outliers, we can keep track of the contribution of each point to the SSE.
- We then eliminate those points with unusually high contributions to the SSE.

Post-processing

- Two post-processing strategies that decrease the SSE by increasing the number of clusters are
  - Split a cluster
    - The cluster which, when split into two sub-clusters, results in the largest decrease in SSE after updating the centroids is chosen.
  - Introduce a new cluster centroid
    - Often the point that is farthest from its associated cluster centroid is chosen.
    - We can determine this if we keep track of the contribution of each point to the SSE.

Post-processing

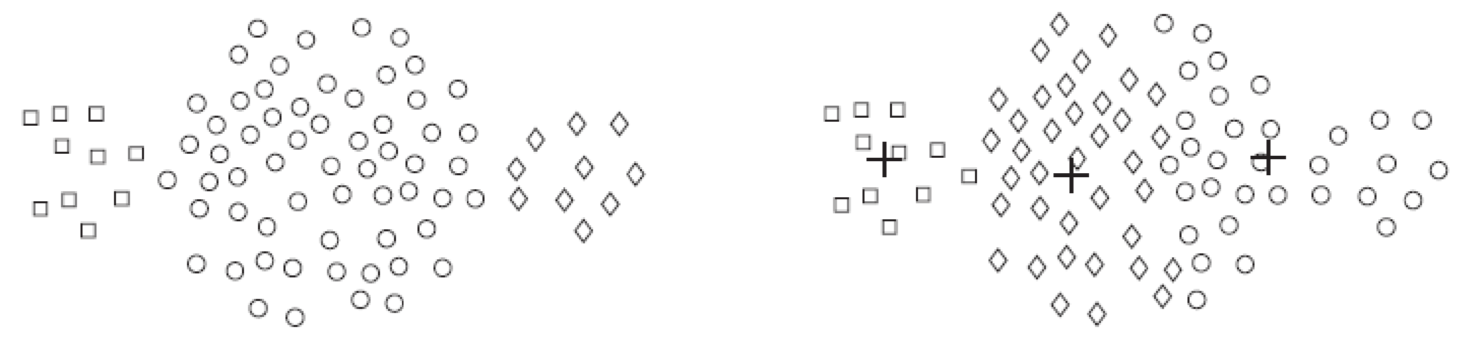
- Two post-processing strategies that decrease the number of clusters, while trying to minimize the increase in SSE, are
  - Disperse a cluster
    - This is accomplished by removing the centroid that corresponds to the cluster.
    - The points in that cluster are then re-assigned to other clusters.
    - The cluster that is dispersed should be the one that increases the SSE the least after updating the centroids.
  - Merge two clusters
    - We can merge the two clusters that result in the smallest increase in SSE after updating the centroids.

Limitations of K-means

- K-means has a number of limitations.
- In particular, the algorithm has difficulty in detecting clusters with non-spherical shapes or widely different sizes or densities.
- This is because K-means is designed to look for globular clusters of similar sizes and densities, or clusters that are well separated.
- This is illustrated by the following examples.

Limitations of K-means

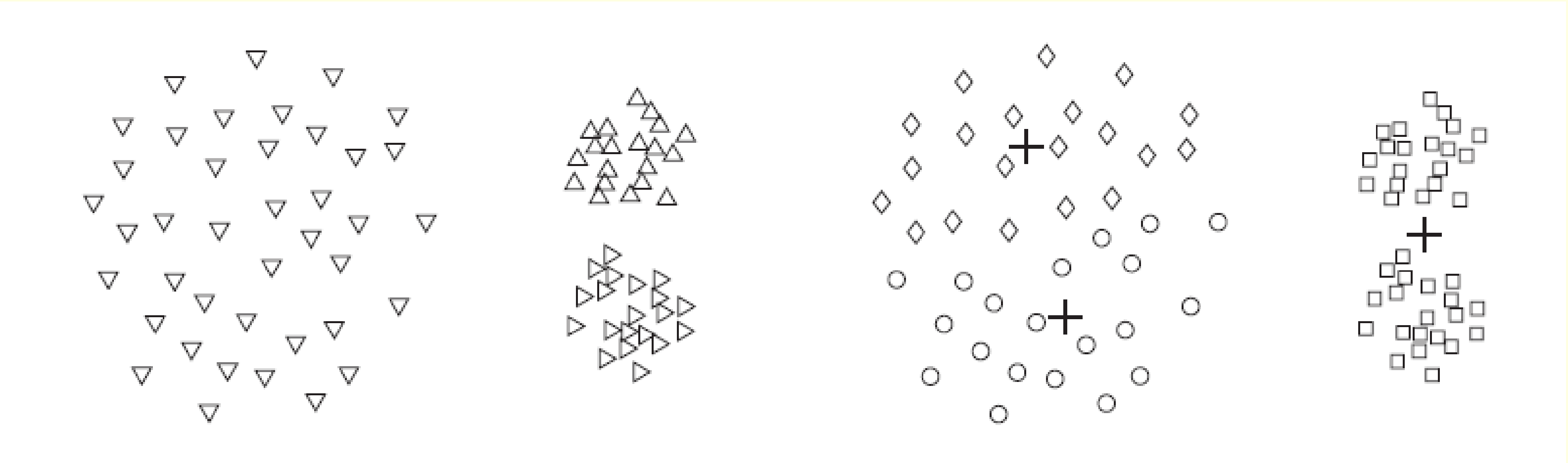
- In this example, K-means cannot find the three natural clusters because one of the clusters is much larger than the other two.
- As a result, the largest cluster is divided into sub-clusters.
- At the same time, one of the smaller clusters is combined with a portion of the largest cluster.



(a) Original points. (b) Three K-means clusters.

Limitations of K-means

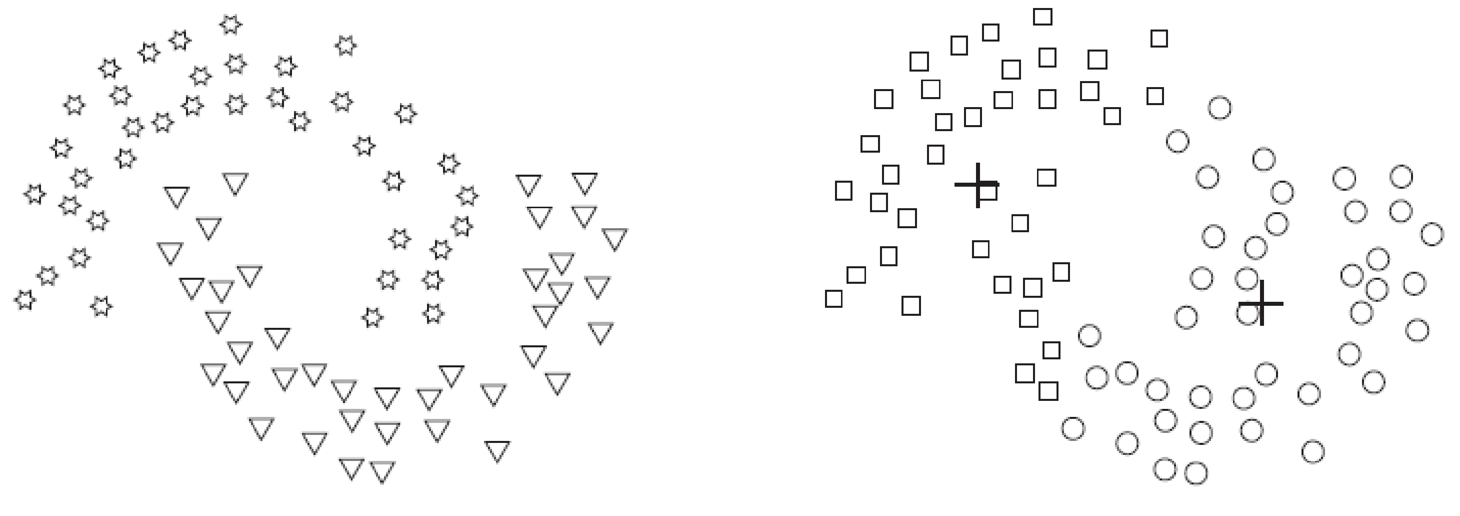
- In this example, K-means fails to find the three natural clusters.
- This is because the two smaller clusters are much denser than the largest cluster.



(a) Original points. (b) Three K-means clusters.

Limitations of K-means

- In this example, K-means finds two clusters that mix portions of the two natural clusters.
- This is because the natural clusters are not globular in shape.



(a) Original points. (b) Two K-means clusters.