

K-Means Clustering

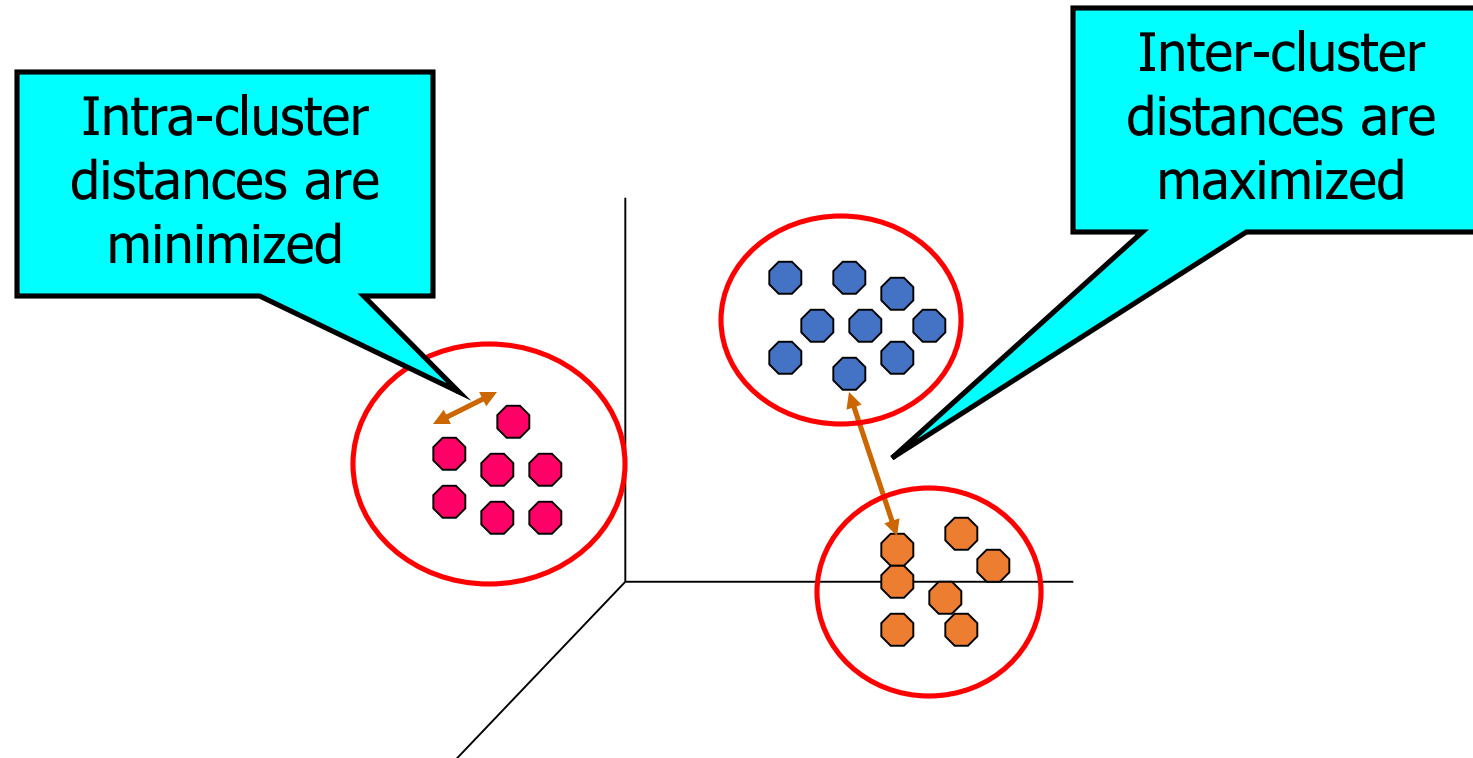
Prof. Navneet Goyal

Department of Computer Science

BITS-Pilani, Pilani Campus

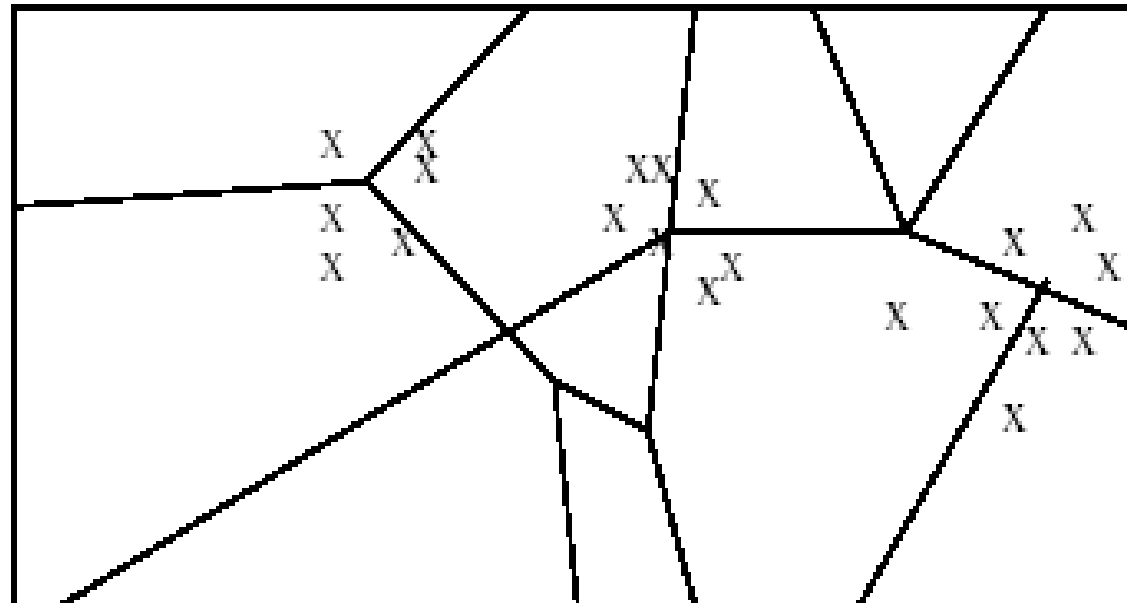
What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

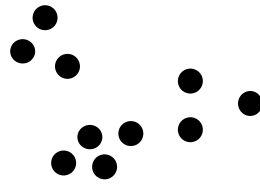
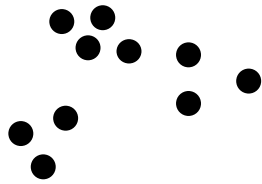


Applications of Cluster Analysis

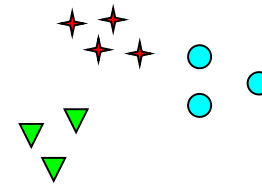
Many years ago, during a cholera outbreak in London, a physician plotted the location of cases on a map, getting a plot that looked like Fig. 14. Properly visualized, the data indicated that cases clustered around certain intersections, where there were polluted wells, not only exposing the cause of cholera, but indicating what to do about the problem. Alas, not all data mining is this easy, often because the clusters are in so many dimensions that visualization is very hard.



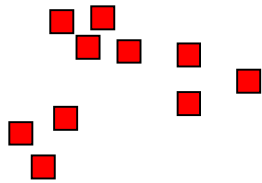
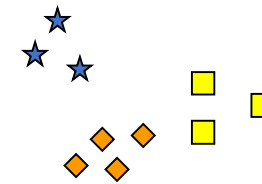
Notion of a Cluster can be Ambiguous



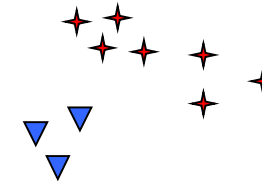
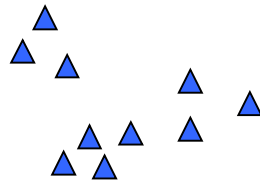
How many clusters?



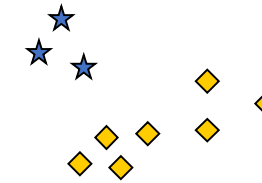
Six Clusters



Two Clusters



Four Clusters



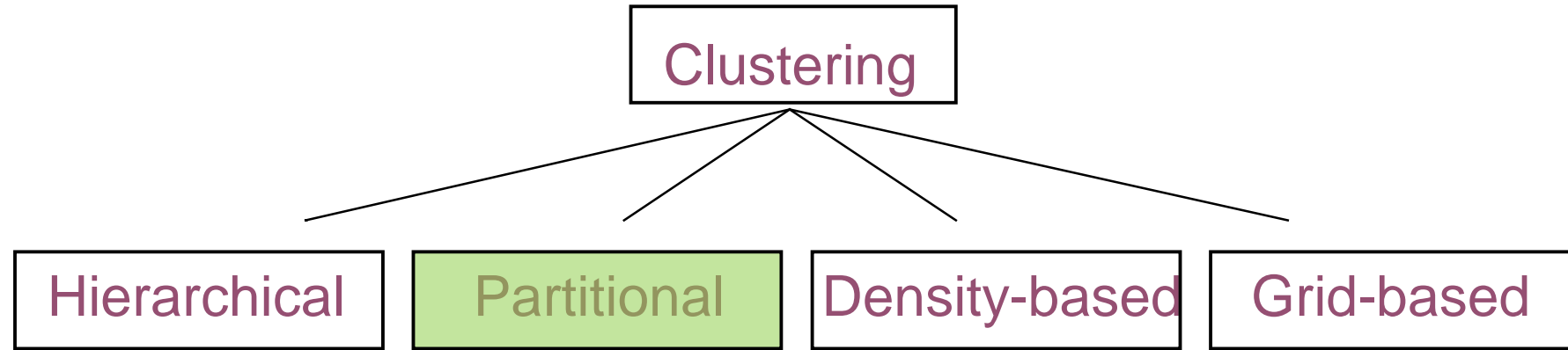
Clustering vs. Classification

- ◆ No prior knowledge
 - ◆ Number of clusters
 - ◆ Meaning/interpretation of clusters
- ◆ Unsupervised learning

Clustering Problem

- ◆ Given a database $D=\{t_1, t_2, \dots, t_n\}$ of tuples and an integer value k , the *Clustering Problem* is to define a mapping $f:D \rightarrow \{1, \dots, k\}$ where each t_i is assigned to one cluster K_j , $1 \leq j \leq k$.
- ◆ A *Cluster*, K_j , contains precisely those tuples mapped to it.
- ◆ Unlike classification problem, clusters are not known a priori.

Clustering Approaches



Partitioning Algorithms:

Basic Concept

- Partitioning method: Construct a partition of a database D of n objects into a set of k clusters
- Given a k , find a partition of k *clusters* that optimizes the chosen partitioning criterion
 - Global optimal: exhaustively enumerate all partitions
 - Heuristic methods: *k-means* and *k-medoids* algorithms
 - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
 - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

K-means


- Works when we know k , the number of clusters we want to find
- Idea:
 - Randomly pick k points as the “centroids” of the k clusters
 - Loop:
 - For each point, put the point in the cluster to whose centroid it is closest
 - Recompute the cluster centroids
 - Repeat loop (until there is no change in clusters between two consecutive iterations.)

Iterative improvement of the objective function:

Sum of the squared distance from each point to the centroid of its cluster

K-means Example

- For simplicity, 1-dimension objects and $k=2$.
 - Numerical difference is used as the distance
- Objects: 1, 2, 5, 6, 7
- K-means:
 - Randomly select 5 and 6 as centroids;
 - \Rightarrow Two clusters $\{1, 2, 5\}$ and $\{6, 7\}$; $\text{meanC1}=8/3$, $\text{meanC2}=6.5$
 - $\Rightarrow \{1, 2\}, \{5, 6, 7\}$; $\text{meanC1}=1.5$, $\text{meanC2}=6$
 - \Rightarrow no change.
 - Aggregate dissimilarity
 - (sum of squares of distance of each point of each cluster from *its* cluster center--(intra-cluster distance))
 $= 0.5^2 + 0.5^2 + 1^2 + 0^2 + 1^2 = 2.5$


$$|1-1.5|^2$$

K-Means Example

- Given: {2,4,10,12,3,20,30,11,25}, $k=2$
- Randomly choose seeds: $m_1=3, m_2=4$
- $K_1=\{2,3\}, K_2=\{4,10,12,20,30,11,25\},$
 $m_1=2.5, m_2=16$
- $K_1=\{2,3,4\}, K_2=\{10,12,20,30,11,25\},$
 $m_1=3, m_2=18$
- $K_1=\{2,3,4,10\}, K_2=\{12,20,30,11,25\},$
 $m_1=4.75, m_2=19.6$
- $K_1=\{2,3,4,10,11,12\}, K_2=\{20,30,25\},$
 $m_1=7, m_2=25$
- Stop as the clusters with these means are the same

Pros & Cons of *K*-means

- Relatively efficient: $O(tkn)$

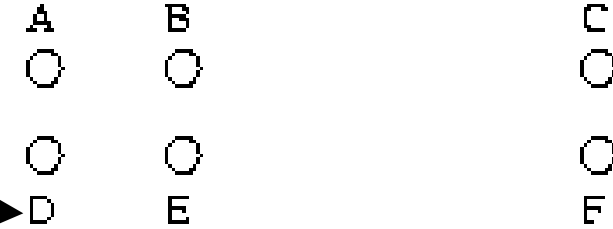
n : # objects, k : # clusters, t : # iterations; $k, t \ll n$.

- Applicable only when mean is defined
- What about categorical data?
- Need to specify the number of clusters
- Unable to handle noisy data and outliers

Problems with K-means

- Need to know k in advance
 - Could try out several k ?
 - Unfortunately, cluster tightness increases with increasing K . The best intra-cluster tightness occurs when $k=n$ (every point in its own cluster)
- Tends to go to local minima that are sensitive to the starting centroids
 - Try out multiple starting points
- Disjoint and exhaustive
 - Doesn't have a notion of "outliers"
 - Outlier problem can be handled by K-medoid or neighborhood-based algorithms
- Assumes clusters are spherical in vector space
 - Sensitive to coordinate changes, weighting etc.

Example showing sensitivity to seeds



In the above, if you start with B and E as centroids you converge to $\{A, B, C\}$ and $\{D, E, F\}$

If you start with D and F you converge to $\{A, B, D, E\}$ $\{C, F\}$

Good Initial Centroids

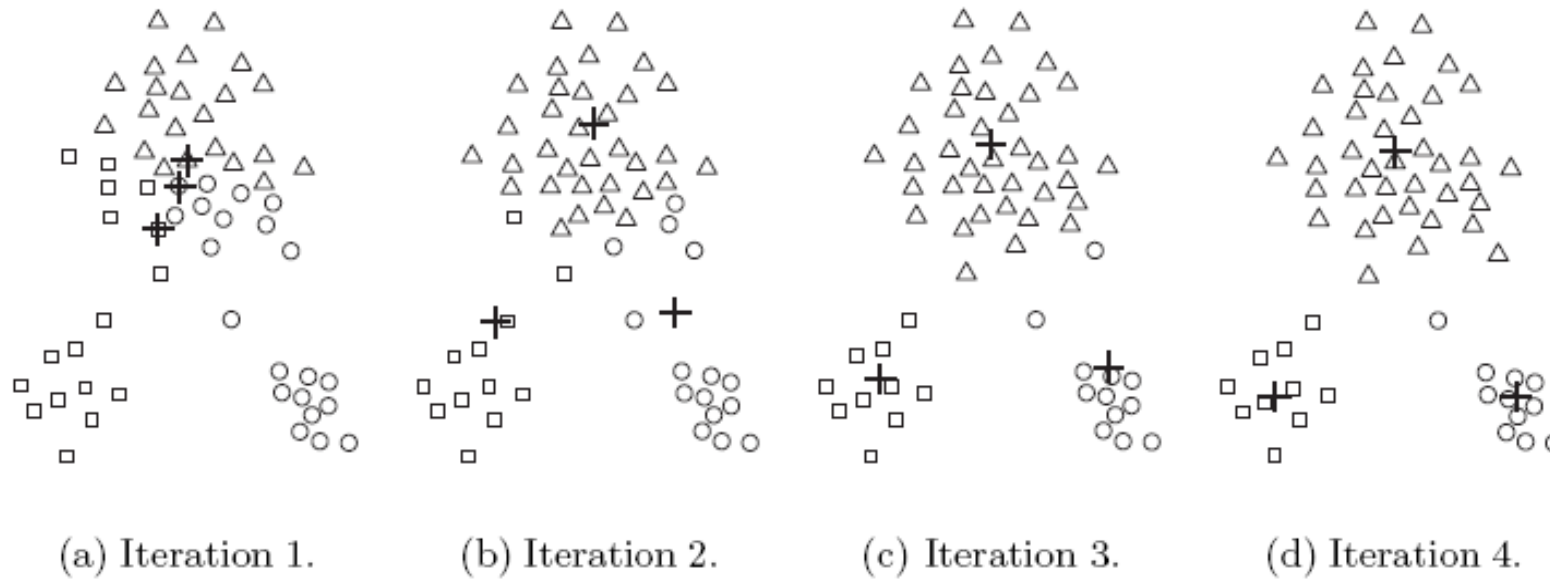


Figure 8.3. Using the K-means algorithm to find three clusters in sample data.

Poor Initial Centroids

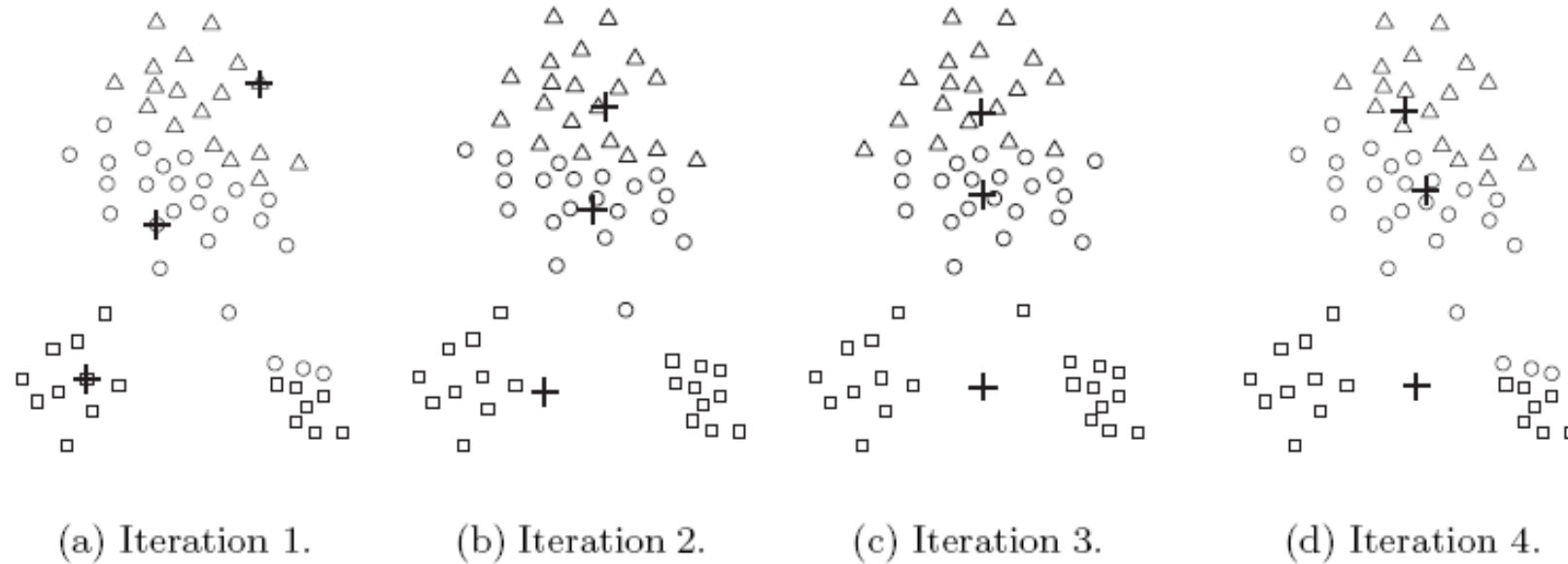


Figure 8.5. Poor starting centroids for K-means.

Good Initial Centroids

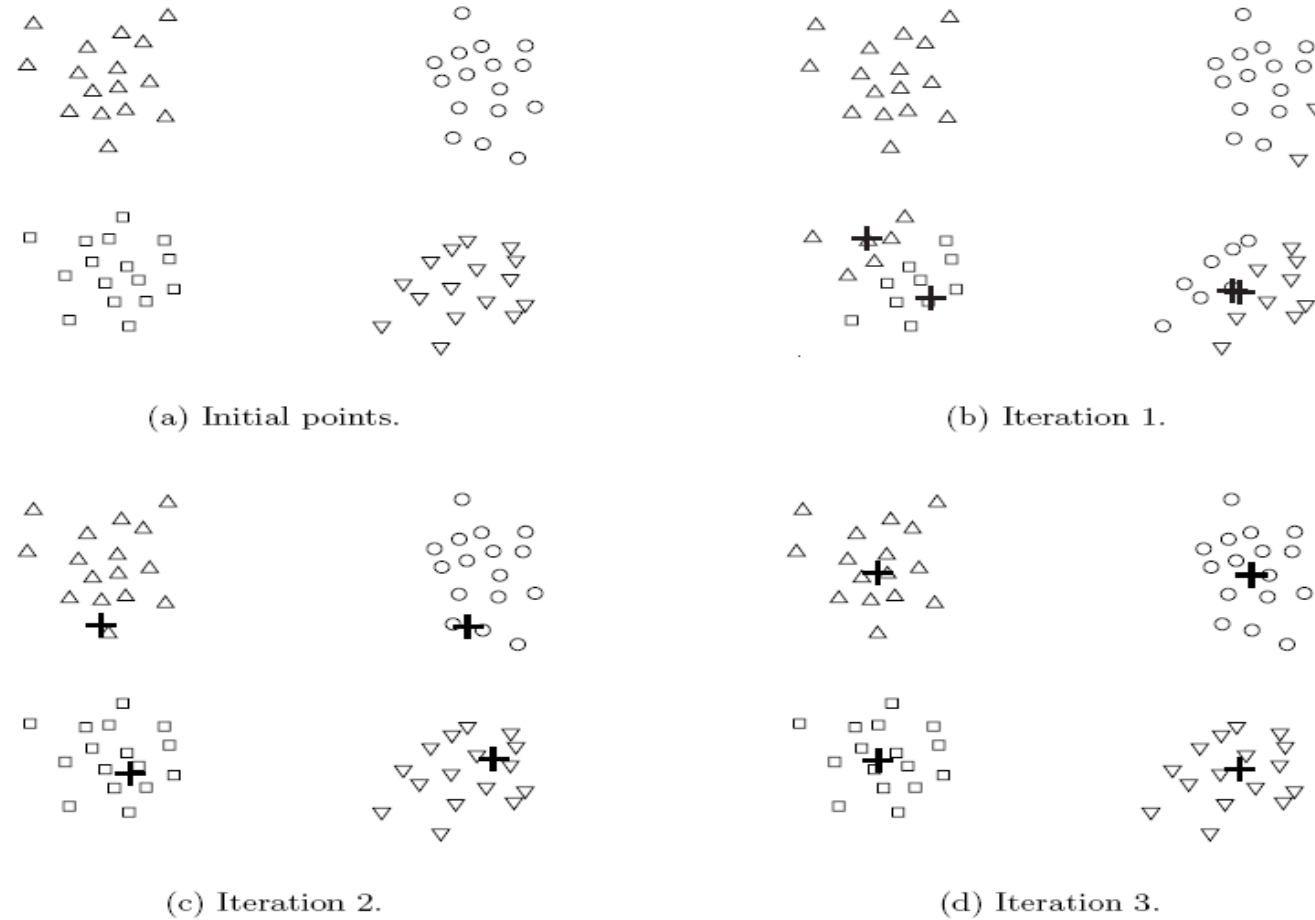


Figure 8.6. Two pairs of clusters with a pair of initial centroids within each pair of clusters.

Source of figure: Introduction to Data Mining by Tan et. al.

Poor Initial Centroids

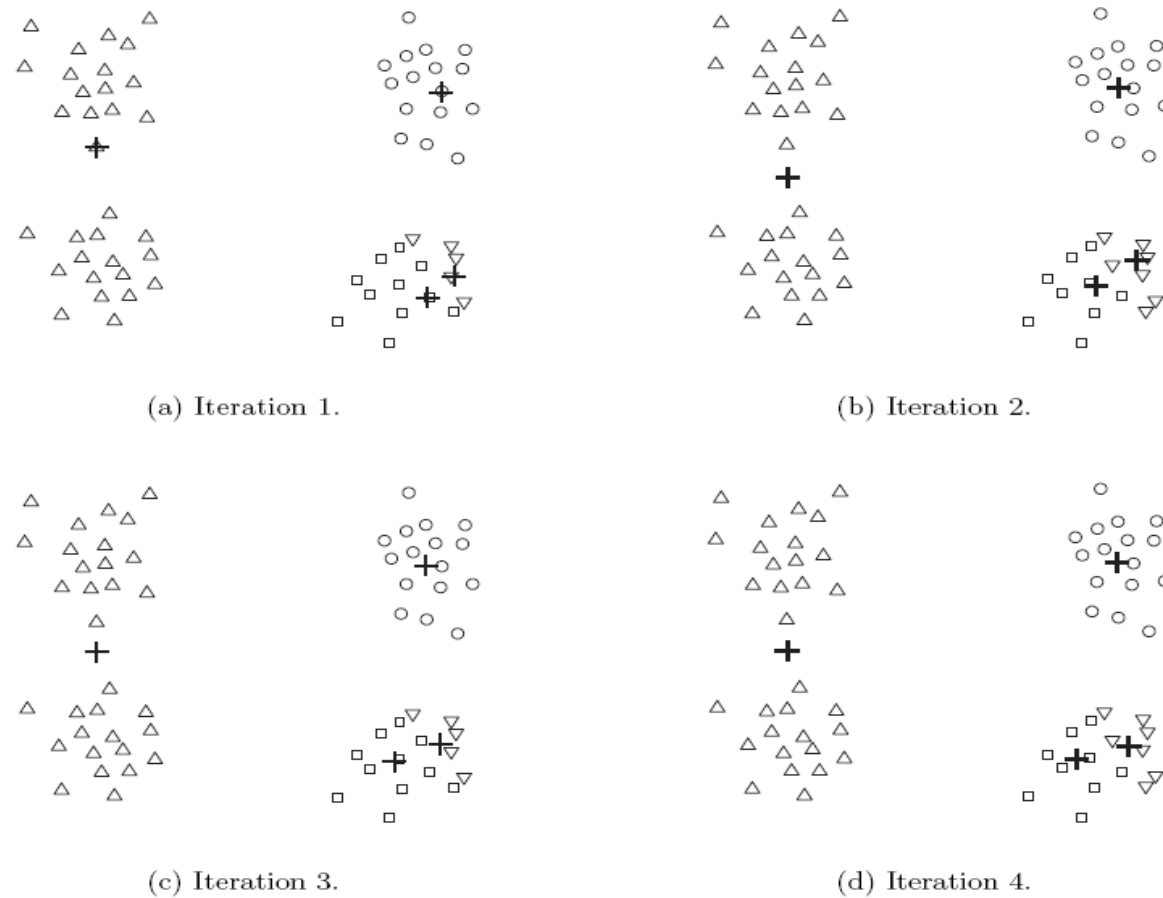


Figure 8.7. Two pairs of clusters with more or fewer than two initial centroids within a pair of clusters.

Source of figure: Introduction to Data Mining by Tan et. al.

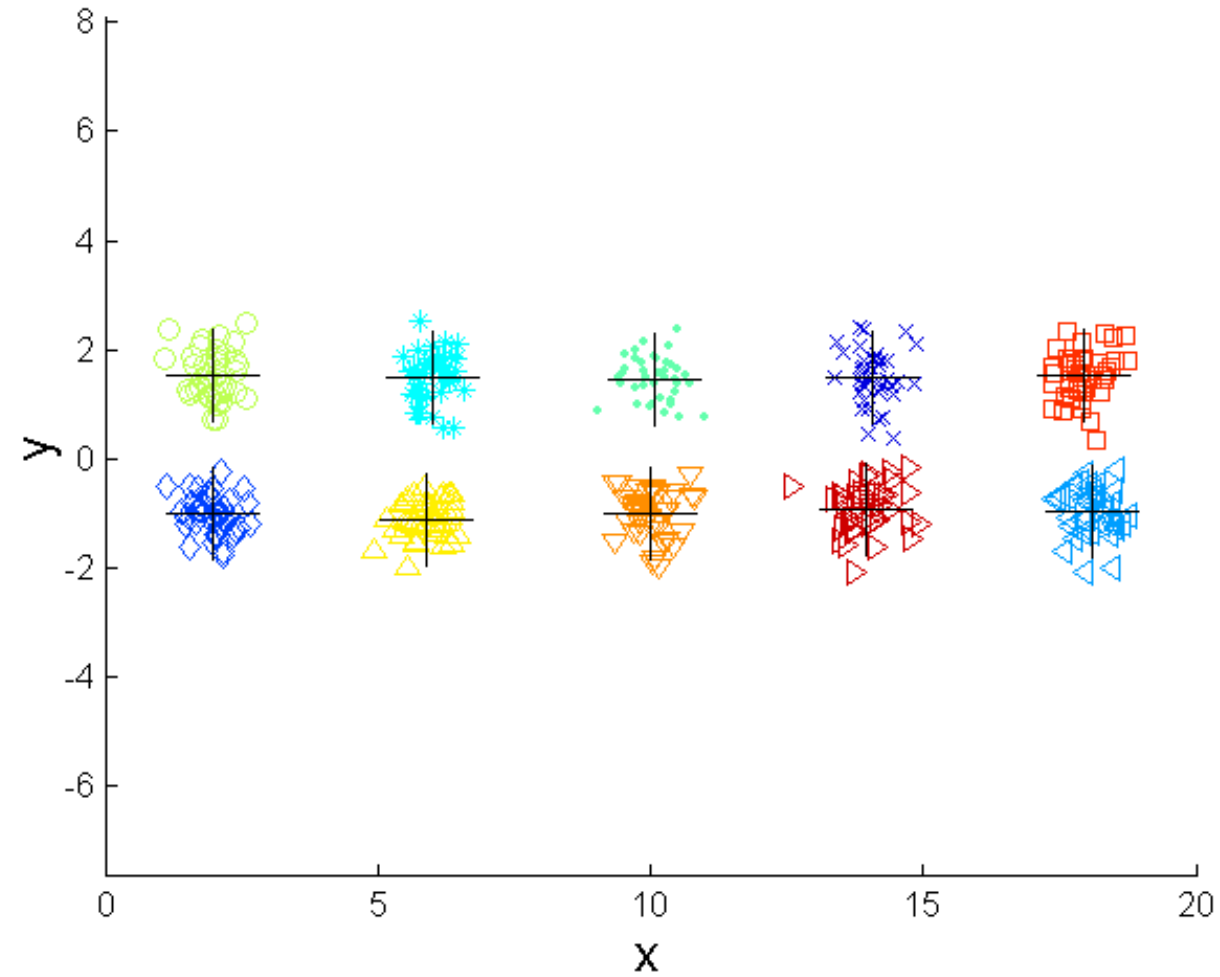
Bisecting K-means

- Bisecting K-means algorithm
 - Variant of K-means that can produce a partitional or a hierarchical clustering

```
1: Initialize the list of clusters to contain the cluster containing all points.
2: repeat
3:   Select a cluster from the list of clusters
4:   for  $i = 1$  to number_of_iterations do
5:     Bisect the selected cluster using basic K-means
6:   end for
7:   Add the two clusters from the bisection with the lowest SSE to the list of clusters.
8: until Until the list of clusters contains  $K$  clusters
```

Bisecting K-means Example

Iteration 10



Source of figure: Introduction to Data Mining by Tan et. al.

K-Means

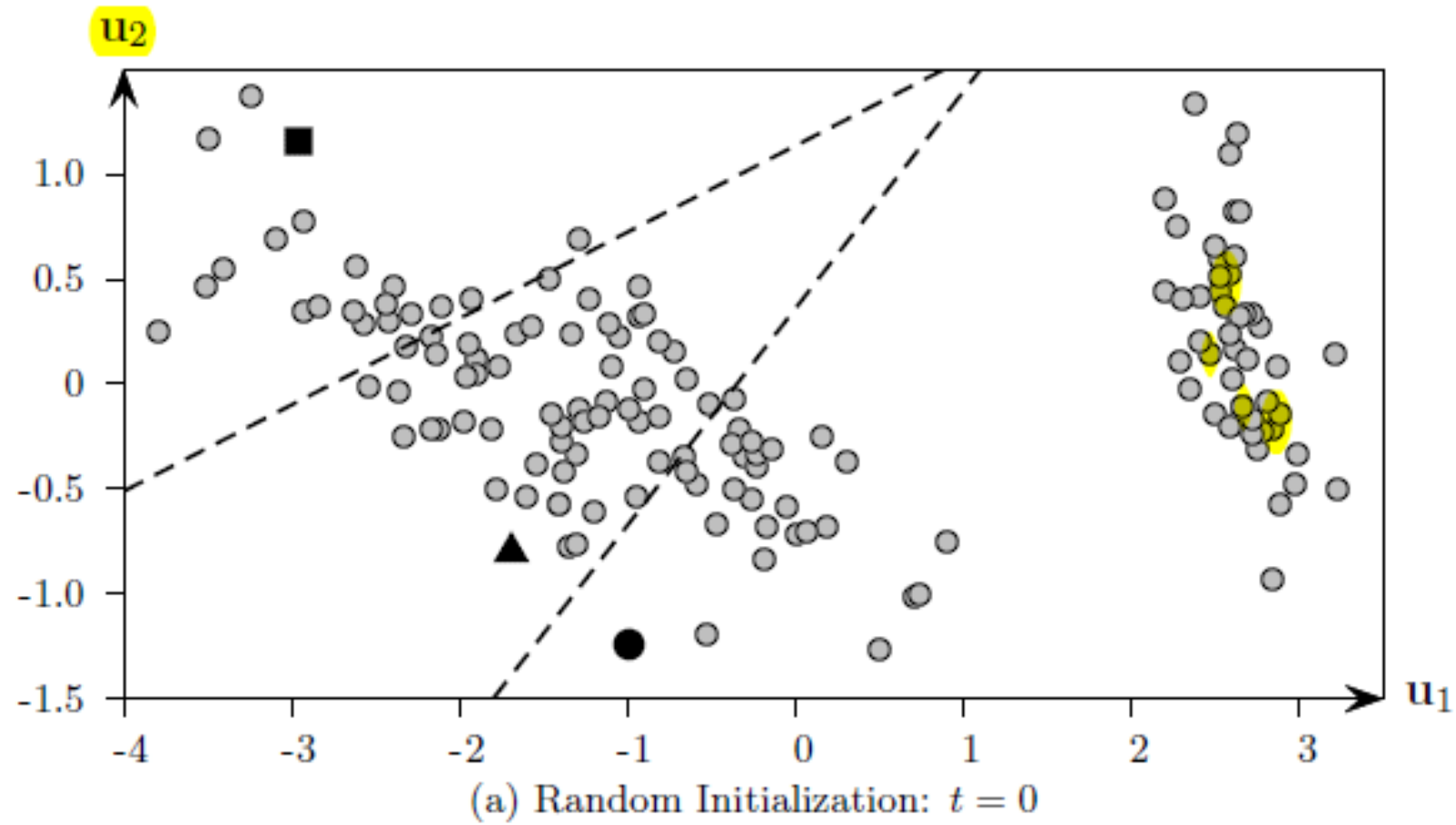


Figure taken from book by M J Zaki on Data Mining Analysis

K-Means

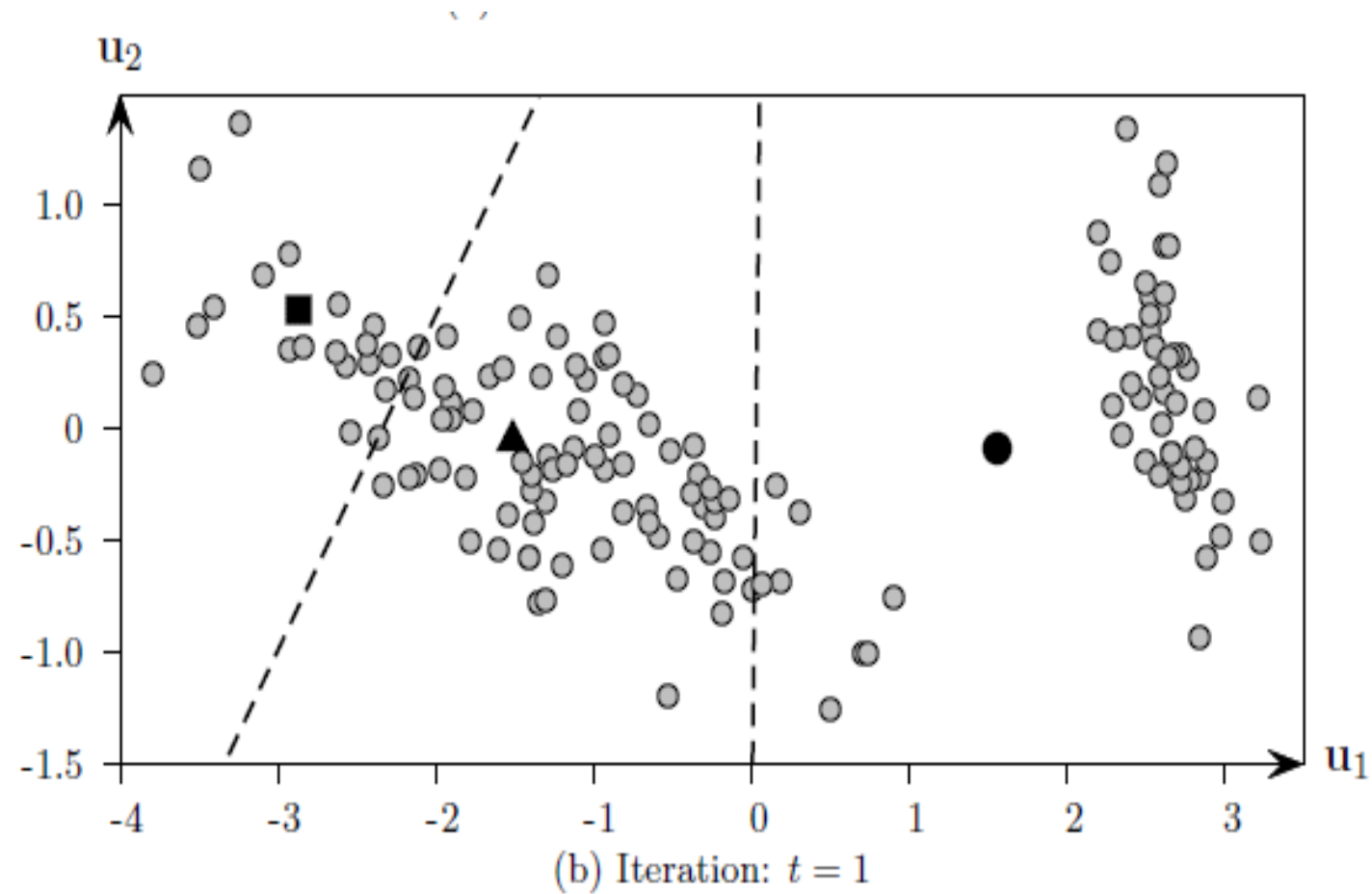


Figure taken from book by M J Zaki on Data Mining Analysis

K-Means

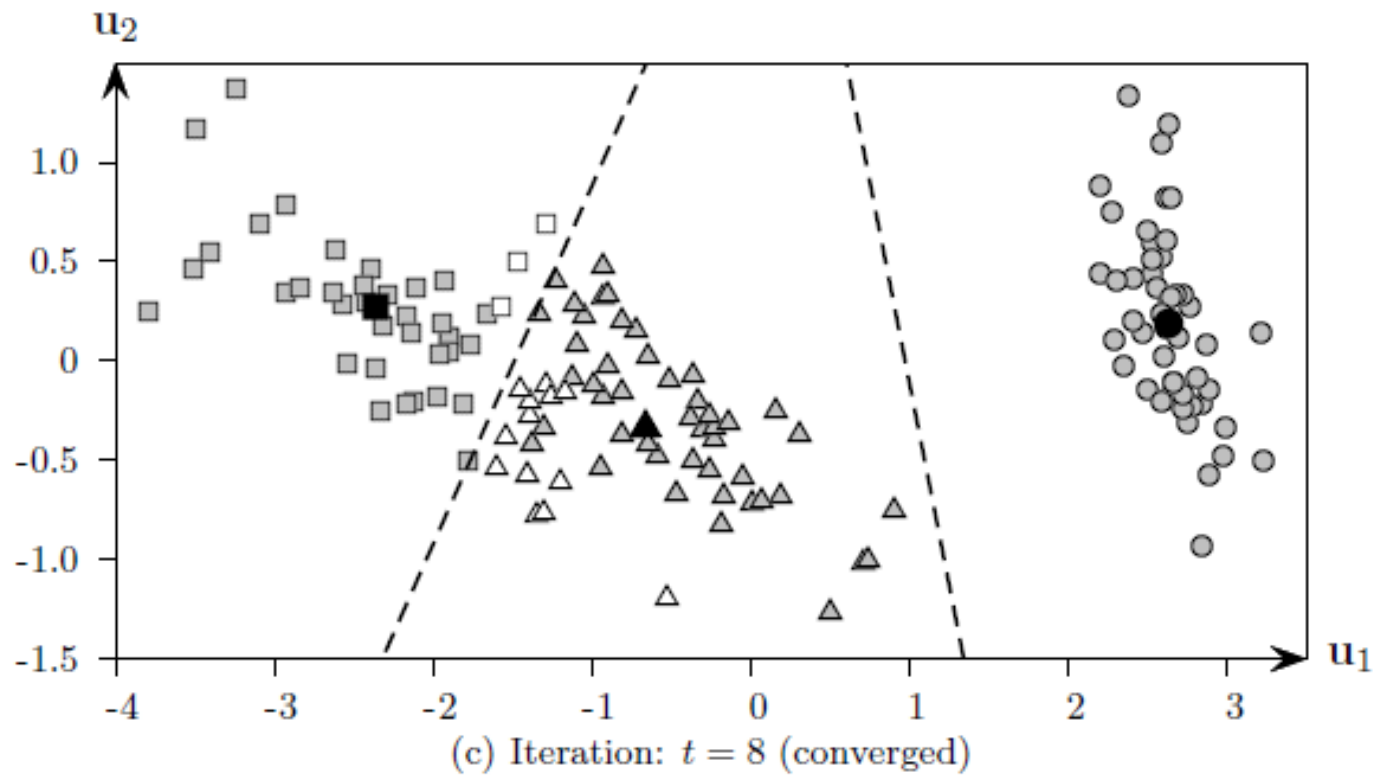


Figure 13.2: K-means in Two Dimensions: Iris Principal Components Dataset

K-Means

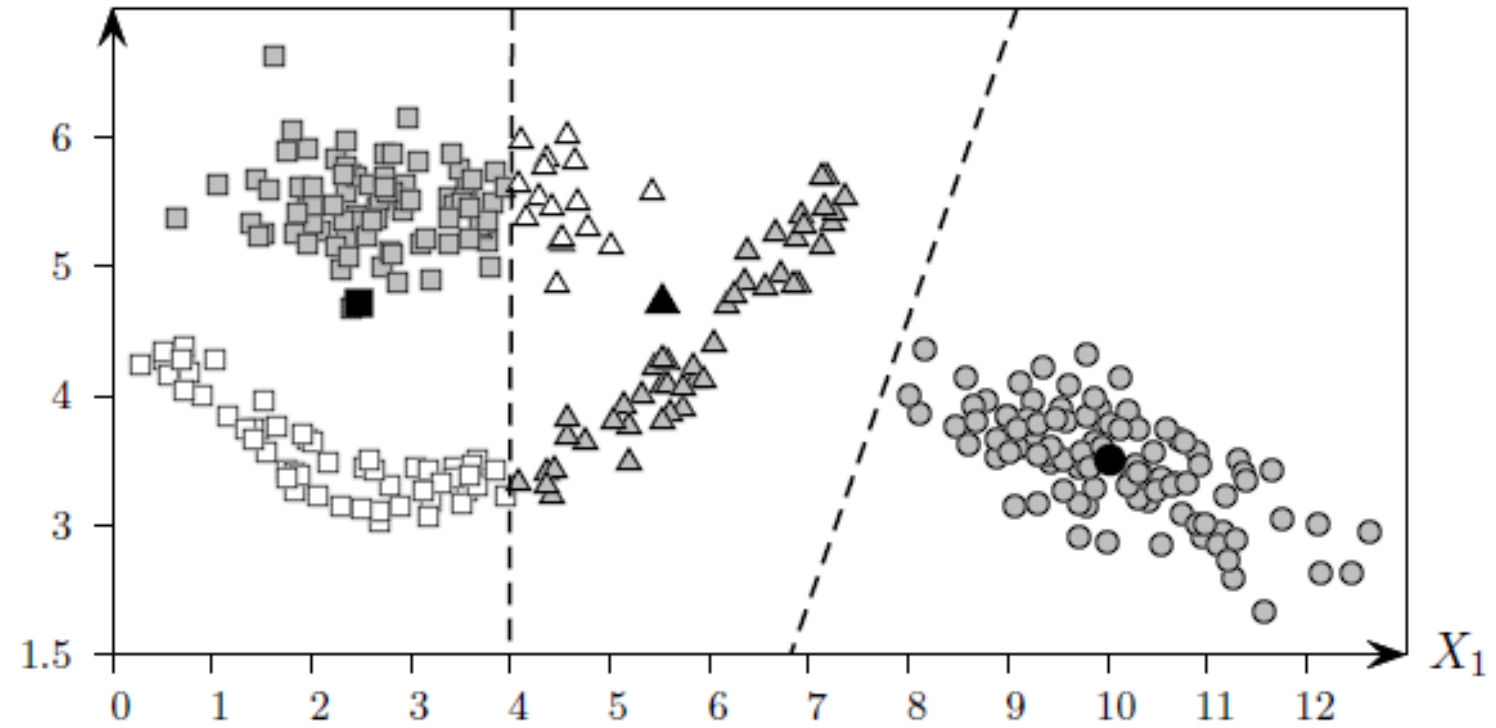


Figure taken from book by M J Zaki on Data Mining Analysis

Kernel K-Means

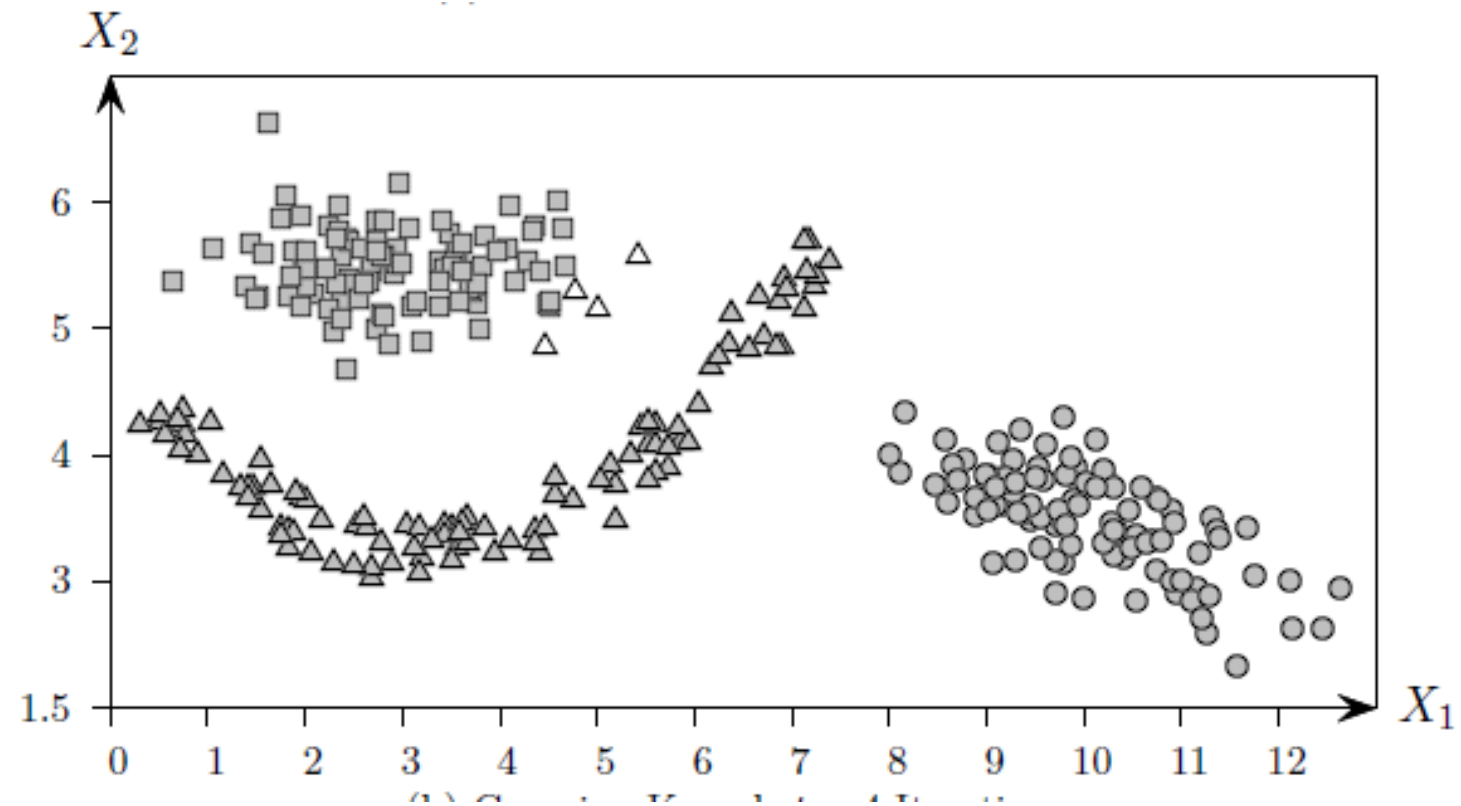


Figure taken from book by M J Zaki on Data Mining Analysis