**Unit-VI: Cluster Analysis**

| Introduction |
| --- |
| Types of Clustering & Types of Clusters |
| Basics of K-means |
| Additional issues:  --Different Issues  -- k-means and bisecting of K-means  -- K-Means and different types of clusters  -- Strengths and weakness  -- K-means as an optimization problem |

**🡪Introduction:**

--Definition

--Difference between Classification and Clustering

--Applications of Clustering

**Definition:**

“**Cluster analysis** or **clustering** is the task of grouping a set of objects or data points in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters).”

Or

“**Clustering** is the process of grouping objects into different groups which are meaningful, useful or both”

**Difference between Classification and Clustering:**

| **CLASSIFICATION** | **CLUSTERING** |
| --- | --- |
| We have a Training set containing data that have been previously categorized | We do not know the characteristics of similarity of data in advance |
| Based on this training set, the algorithms finds the category that the new data points belong to | Using statistical concepts, we split the datasets into sub-datasets such that the Sub-datasets have “Similar” data |
| Since a Training set exists, we describe this technique as **Supervised learning** | Since Training set is not used, we describe this technique as **Unsupervised learning** |
| **Example:** We use training dataset which categorized customers that are loyal. Now based on this training set, we can classify whether a customer will be loyal to our shop or not. | **Example:** We use a dataset of customers and split them into sub-datasets of customers with “similar” characteristics. Now this information can be used to market a product to a specific segment of customers that has been identified by clustering algorithm |

**Applications:**

--Biology

--Information Retrieval

--Medicine

--Statistics

--Climate

--Market Research

--World Wide Web

--Educational Institutions …

**🡪Types of Clustering & Types of Clusters:**

**Types of Clustering:**

--Hierarchical clustering Vs Partitional Clustering

-- Exclusive Clustering Vs Overlapping Clustering

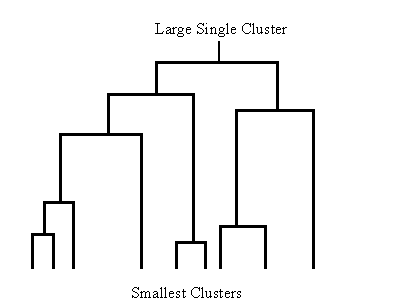
-- Fuzzy Clustering

-- Complete Clustering Vs Partial Clustering

**Hierarchical clustering Vs Partitional Clustering**

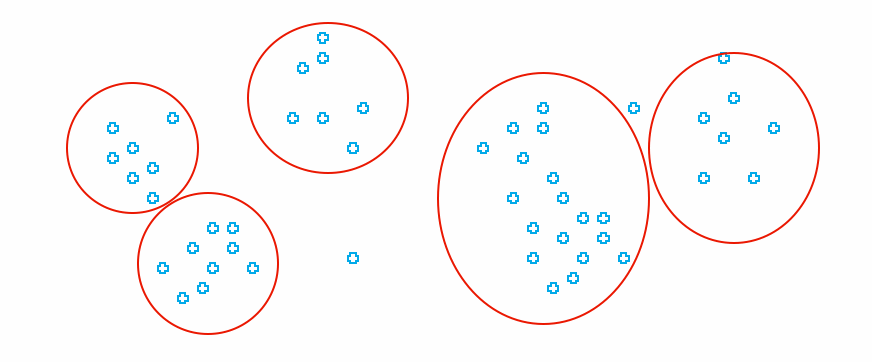
Hierarchical Clustering:

A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset. Each node (cluster) in the tree is the union of its children, and the root of the tree is the cluster contain



Partitional Clustering:

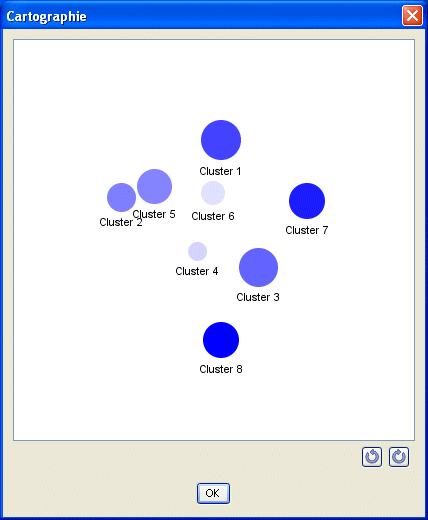
* + A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset



**Exclusive Clustering Vs Overlapping Clustering:**

Exclusive Clustering:

“In Exclusive clustering each data object is assigned to a single object”

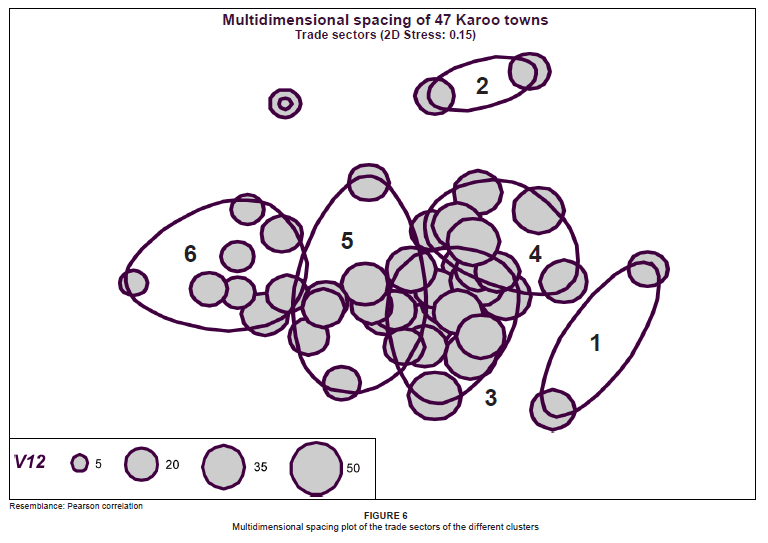


Overlapping Clustering:

--In non-exclusive or overlapping clustering, points may belong to multiple clusters.

--This clustering is used to represent that an object can belong to more than a single cluster.

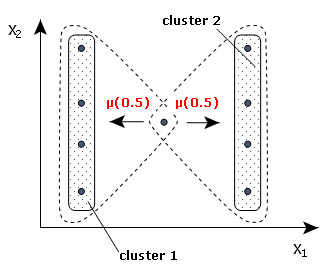
--For example, a person can act as student and faculty in the same organization.



**Fuzzy clustering:**

--In fuzzy clustering, every object belongs to a cluster with a membership weight that is between 0 (absolutely doesn’t belong) and 1 (absolutely belong).

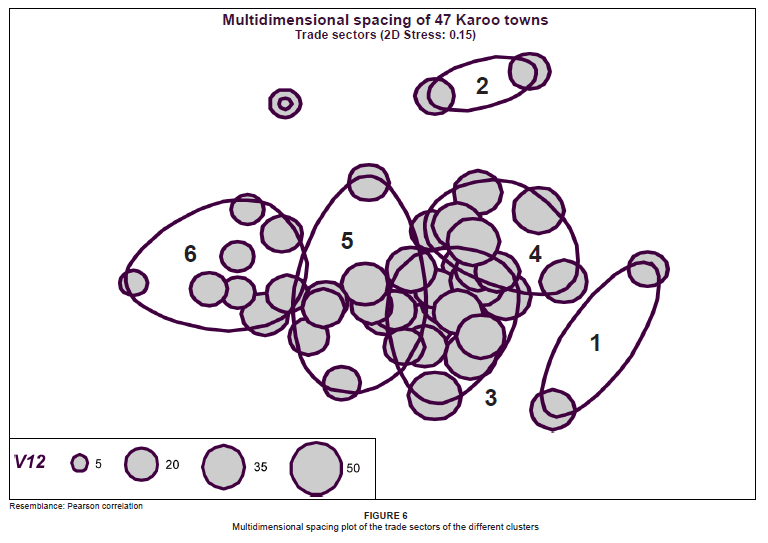
--In fuzzy clustering we often impose that the sum of weights of an object must be 1.



**Complete Clustering Vs Partial Clustering:**

Complete Clustering:

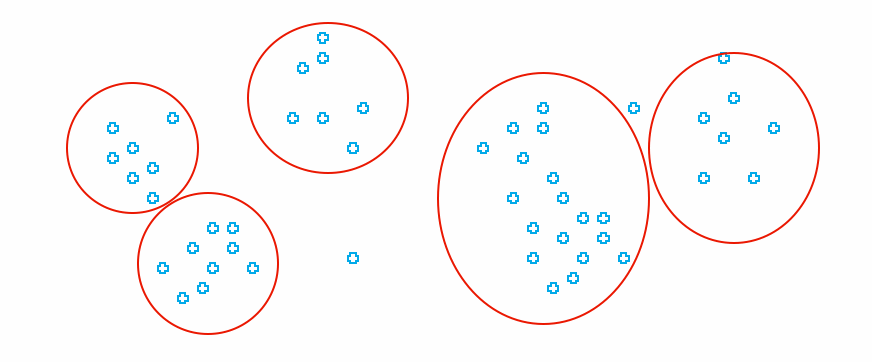
**--**A complete clustering assigns every object to a cluster.



Partial Clustering:

--In Partial clustering, some of the data points are left alone as outliers or noises.

--These types of clusters are used for outlier analysis



**Types of Clusters:**

--Well-separated clusters

--Center-based clusters

-- Contiguous clusters

-- Density-based clusters

--Property or Conceptual

Well Separated Clusters:

* + A Well Separated 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.

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 often a centroid, the average of all the points in the cluster, or a medoid, the most “representative” point of a cluster

Contiguous Cluster (Nearest neighbor or Transitive)

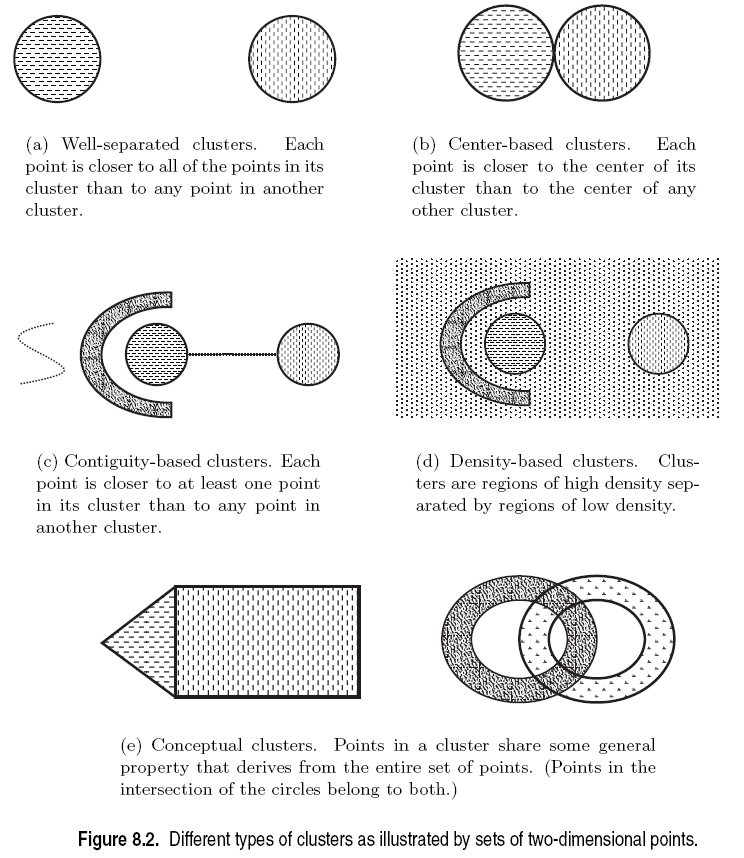
* + A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

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.

Shared Property or Conceptual Clusters

* + Finds clusters that share some common property or represent a particular concept.



**🡪Basics of K-means or Basic K-means Algorithm:**

--K-Means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem.

-- The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters).

-- The main idea is to define k centroids, one for each cluster.

--These centroids should be placed in a intelligent way because of different location causes different result.

-- So, the better choice is to place them as much as possible far away from each other.

--The next step is to take each point belonging to a given data set and associate it to the nearest centroid.

--When, no point is pending, the first step is completed and an early group is done.

--At this point it is necessary to re-calculate k new centroids as bar centers of the clusters resulting from the previous step.

-- After obtaining these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid.

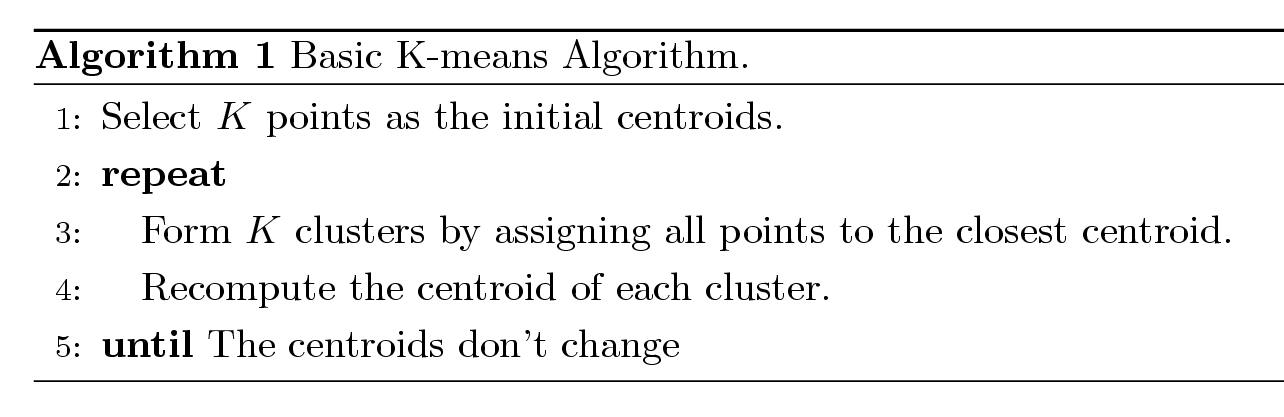
--A loop has been generated. As a result of this loop, one may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

--Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function:

http://docsdrive.com/images/ansinet/itj/2011/img1-2k11-478-484.gif

--Where, ||xi(j)-cj||2 is a chosen distance measure between a data point xi(j) and the cluster center cj, is an indicator of the distance of the n data points from their respective cluster centers.

**Algorithm:**



--In the K-means initial centroids are often chosen randomly.

--The centroid is (typically) the mean of the points in the cluster.

--The ‘Closeness’ between the data points or the objects can be measured by using Euclidean distance, cosine similarity, correlation, etc.

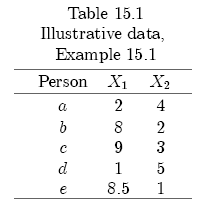
--We can specify the stopping condition as the number of clusters that we want.

--The Complexity of K-means is O( n \* K \* I \* d )

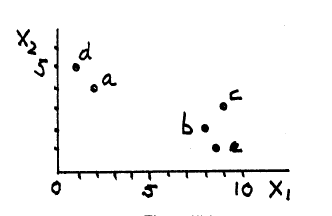
* + n = number of points, K = number of clusters,   
    I = number of iterations, d = number of attributes

**Example: Numeric Data**

**--**The daily expenditures on food (X1) and clothing (X2) of five persons are shown in Table 15.1.



--The data points can be allocated on the graph as:



**--**In order to calculate the distance between two data points we can use Euclidean distance:

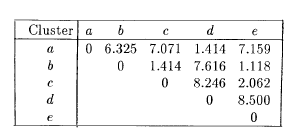
--The Euclidean distance can be measured by using:



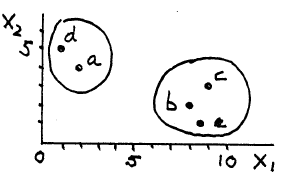
**--**For example, the distance between a and b is:



--After calculating the distance between all the data points, the data **proximity matrix** will look like:



--By using the **proximity matrix** we can divide the data points into different clusters as shown below::



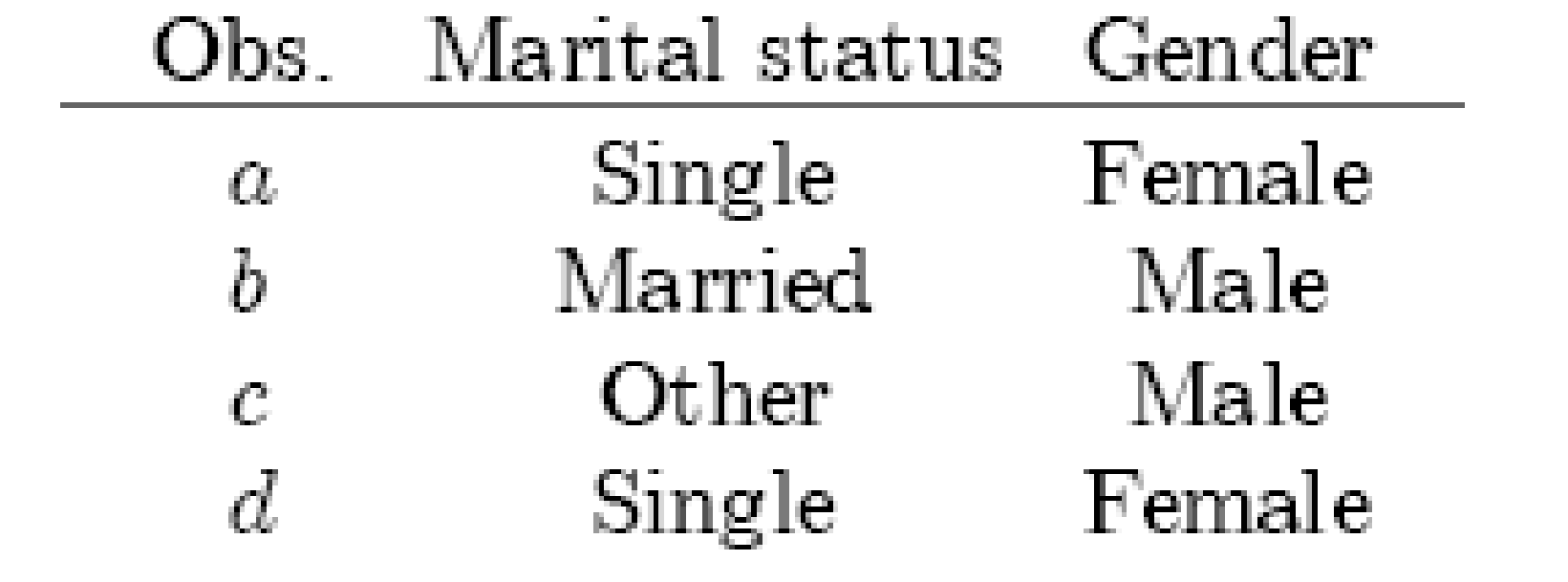
--We can use the following method to calculate the centroid:

For each cluster, add the values of all members. For example, if a cluster of data consisted of the points (2,4), (1,5) the sum of the values would be (3,6).

Divide the total by the number of members of the cluster. In the example above, 3 divided by 2 is 1.5, and 6 divided by 2 is 3, so the centroid of the cluster is (1.5, 3).

**Example 2:** Categorical ATTRIBUTES:

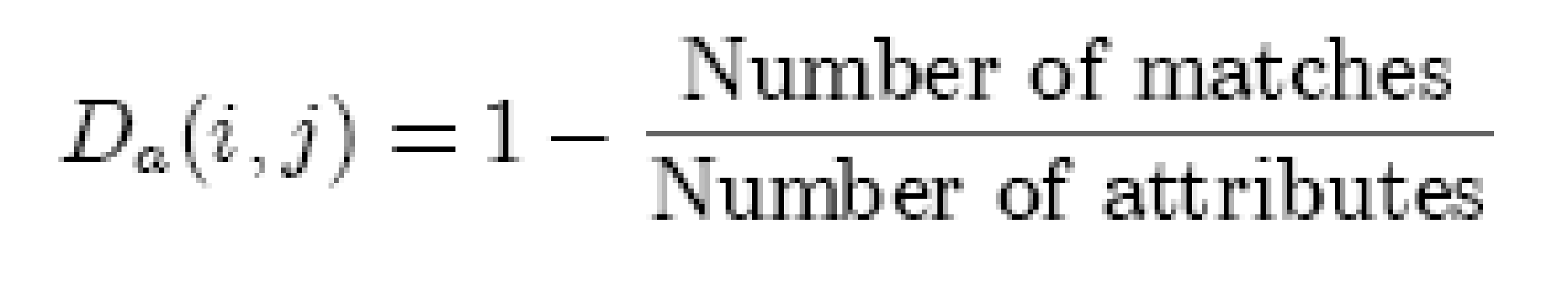
--Consider, for example, the following description of four persons according to marital status (single, married, divorced, other) and gender (male, female):

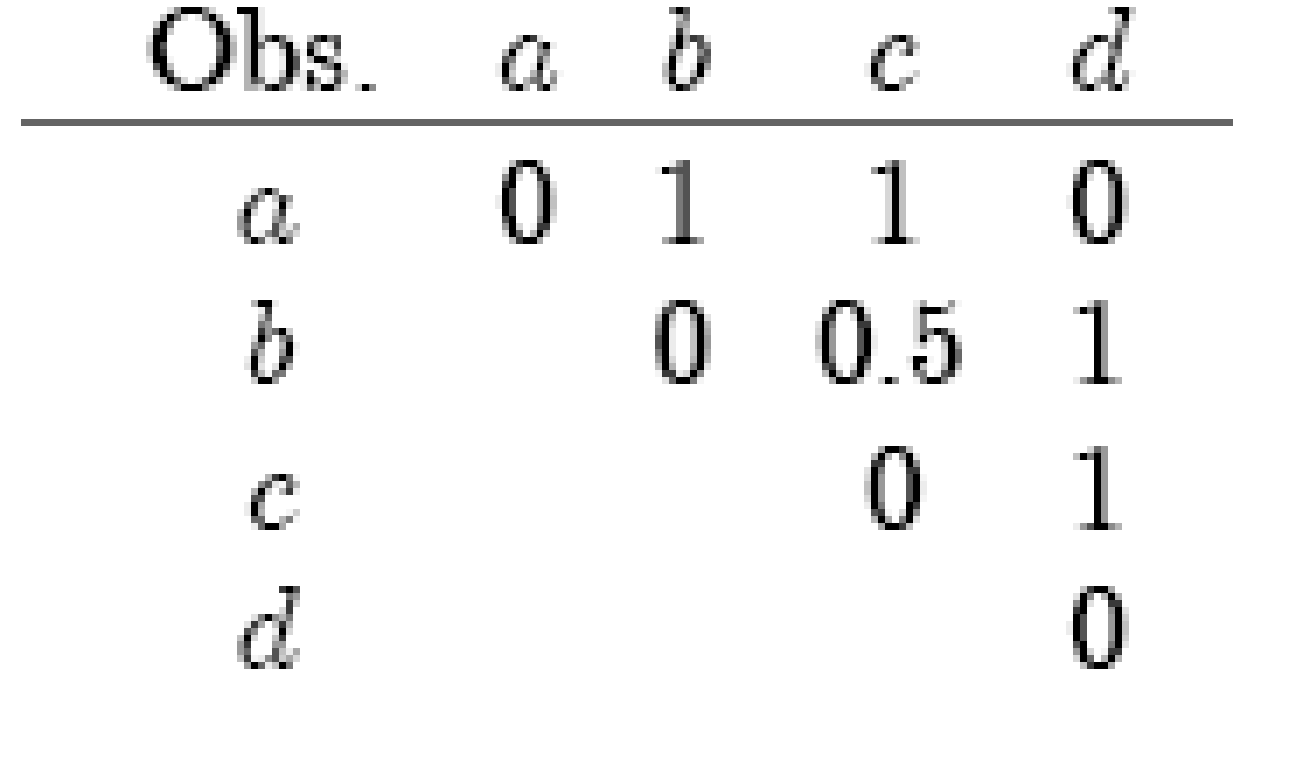


--A reasonable measure of the similarity of two observations is the ratio of the number of matches (identical categories) to the number of attributes. For example, since a and d are both single and female, the similarity measure is 2/2 or 1; b and c do not have the same marital status but are both male, so the similarity measure is 1/2. To be consistent with earlier measures,

however, we use instead as the measure of \distance" (dissimilarity) of two observations i and j.

We declare two observations to be closer, the smaller this distance. The distances between all pairs of observations in our example are as follows:





**Drawbacks of K-means:**

--Need to know k (number of Clusters) in advance

--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



--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}

--Doesn’t have a notion of “outliers”

* + - Outlier problem can be handled by K-medoid or neighborhood-based algorithms

**🡪Additional Issues:**

--Different Issues

-- k-means and bisecting of K-means

-- K-Means and different types of clusters

-- Strengths and weakness

**Different Issues:**

--Evaluating K-means

--Problems with selecting Initial points

--Solution to Initial centroids problem

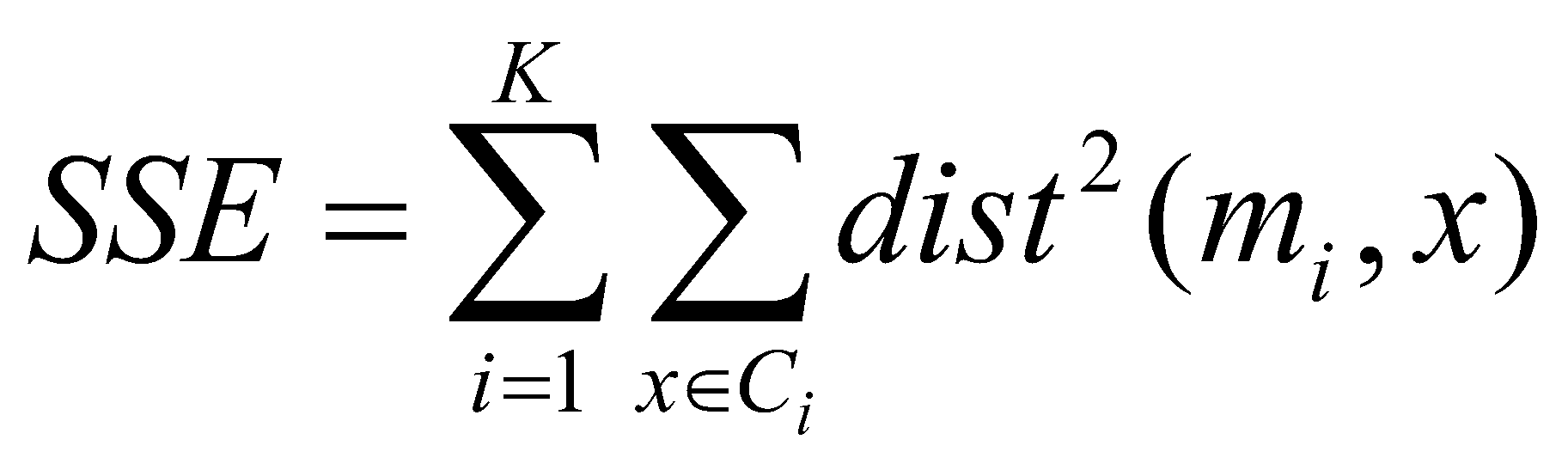
--Preprocessing & Post processing

**Evaluating K-means**

--In order to compare two different runs of the K-Means algorithm we have to be able estimate its quality

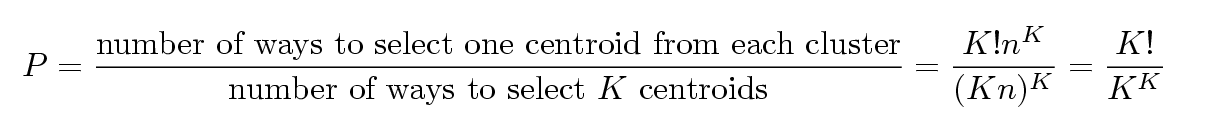
--There is no ground truth (we don't know the clusters/labels beforehand)

--Instead the sum of squared errors (SSE) is used:



* + *x* is a data point in cluster *C*i and *mi* is the representative point for cluster *C*i
    - can show that *mi*corresponds to the center (mean) of the cluster
  + Given two clusters, we can choose the one with the smallest error
  + One easy way to reduce SSE is to increase K, the number of clusters
    - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K
* --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



* + For example, if K = 10, then probability = 10!/1010 = 0.00036
  + Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t.

**Solutions to Initial Centroids Problem:**

Multiple runs

* + Helps, but probability is not on your side

Select more than k initial centroids and then select among these initial centroids

* + Select most widely separated

Bisecting K-means

* + Not as susceptible to initialization issues

**Pre-processing and Post-processing**

Pre-processing

* + Normalize the data
  + Eliminate outliers

Post-processing

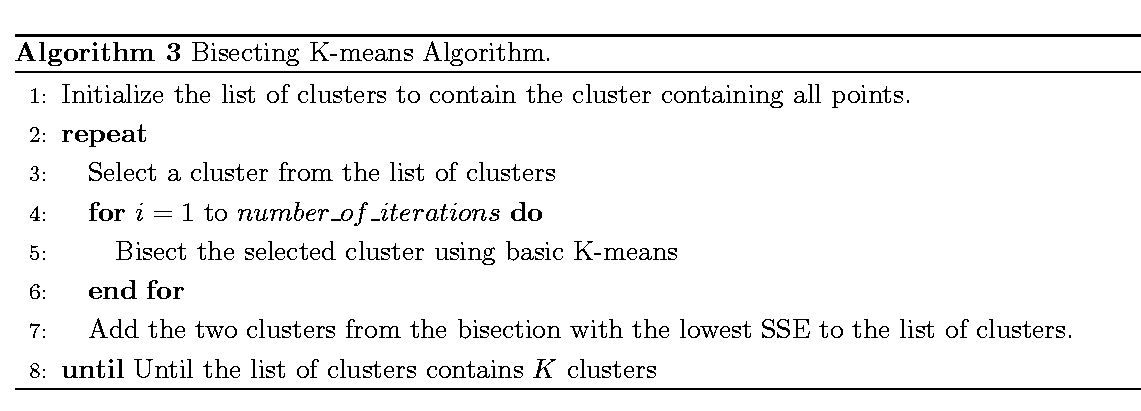
* + Eliminate small clusters that may represent outliers
  + Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  + Merge clusters that are ‘close’ and that have relatively low SSE

**K-means & Bisecting K-means:**

--Extension of the basic K-Means algorithm

-- Basic idea: Initially split the data into two cluster, then further split one of the clusters, and so on, until there are K clusters

--Results in hierarchical clusters

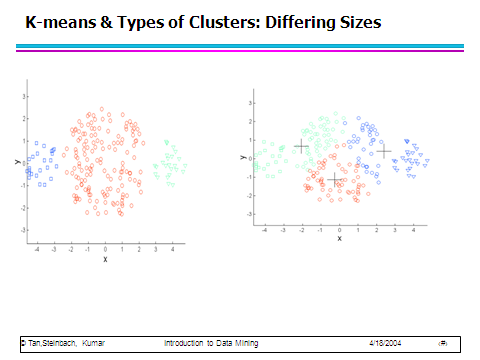


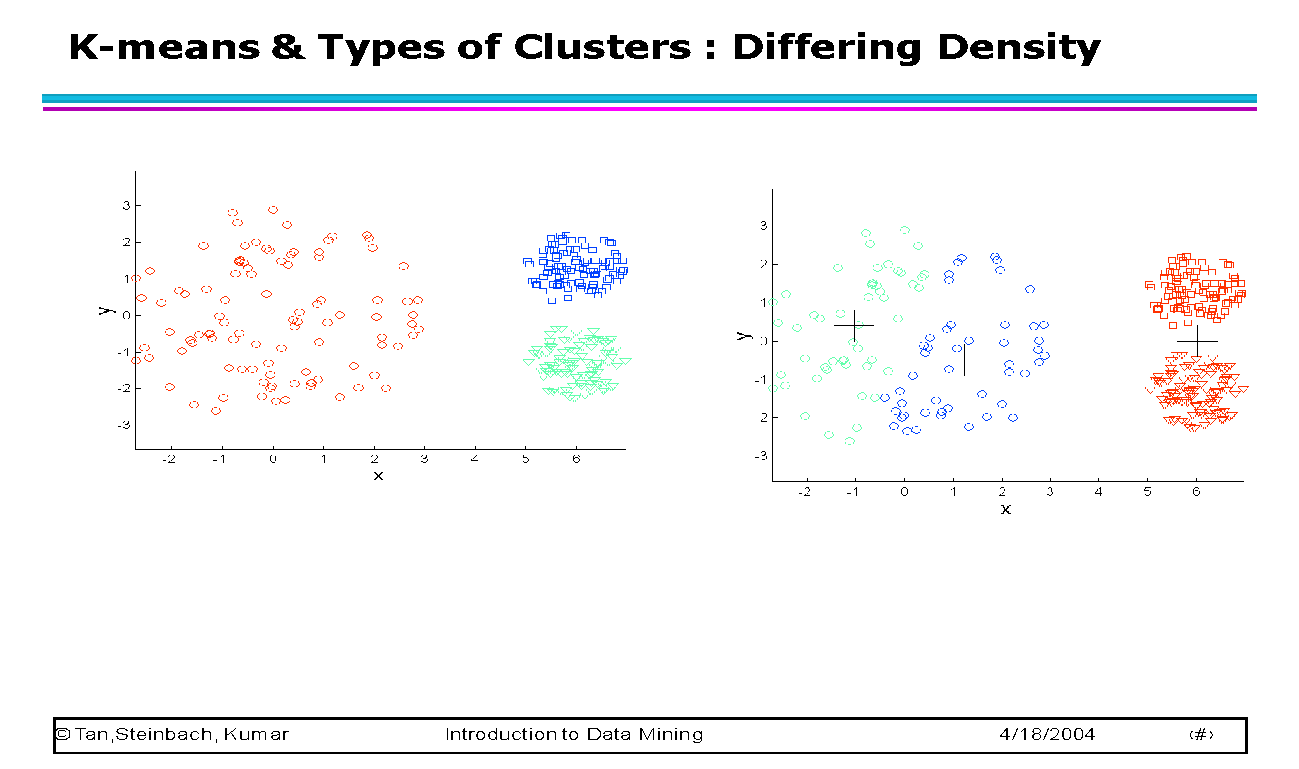
**K-means & Types of Clusters**

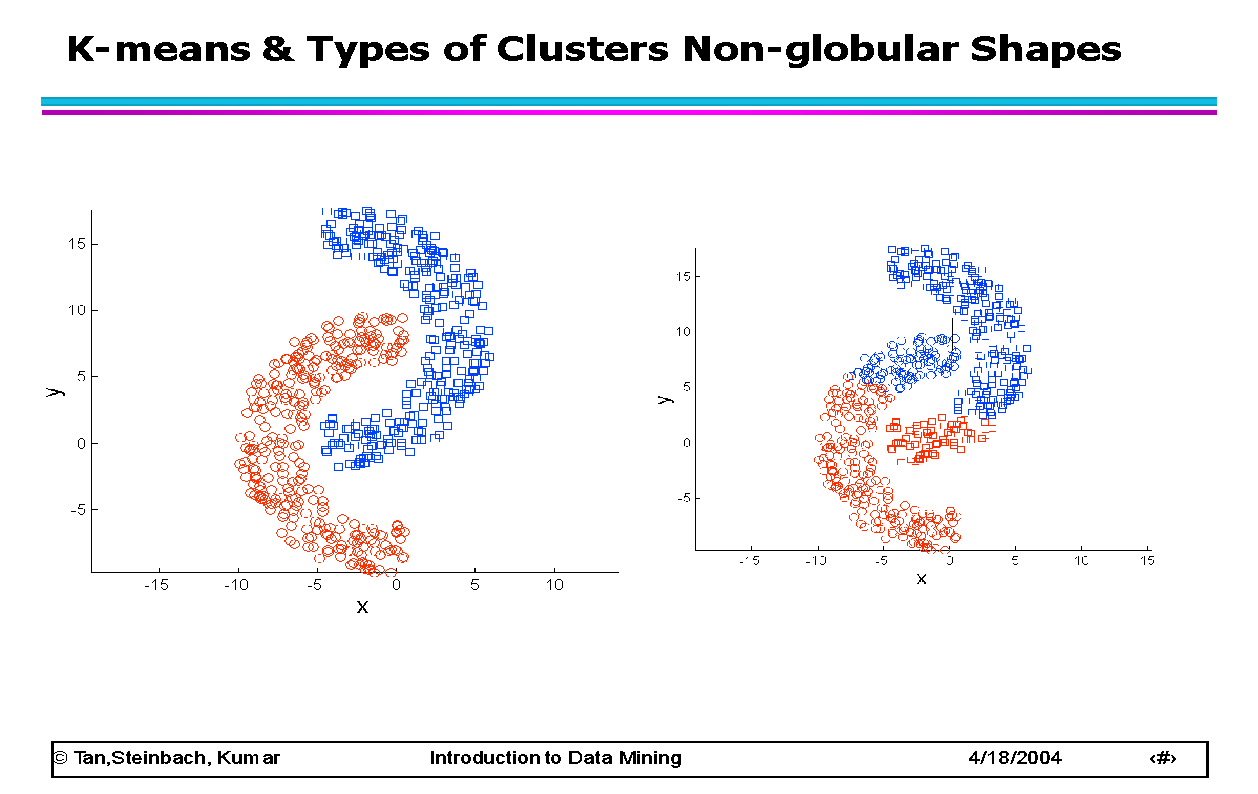
K-means has problems when clusters are of differing

* + Sizes
  + Densities
  + Non-globular shapes

K-means has problems when the data contains outliers.







**Strengths & Weakness of K-means:**

**Advantages or Strengths:**

1) Fast, robust and easier to understand.

2) Relatively efficient: O(tknd), where n is # objects, k is # clusters, d is # dimension of each object, and t  is # iterations. Normally, k, t, d << n.

3) Gives best result when data set are distinct or well separated from each other.

**Weakness:**

--The learning algorithm requires a prior specification of the number of clusters.

-- If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.

-- Randomly choosing the cluster center cannot lead us to the fruitful result.

-- Applicable only when mean is defined i.e. fails for categorical data.

-- Unable to handle noisy data and outliers*.*

-- Algorithm fails for non-linear data set.