Fundamentals of Artificial Intelligence

Prof. Shyamanta M. Hazarika
Department of Mechanical Engineering
Indian Institute of Technology – Guwahati

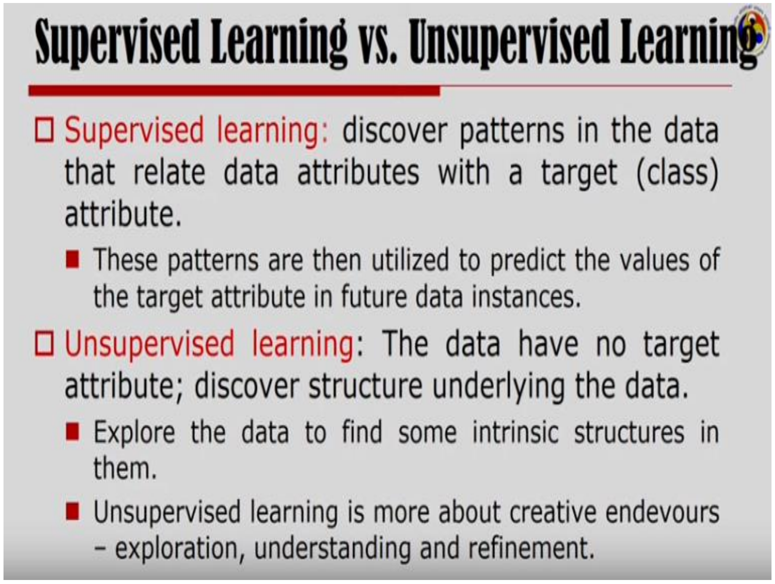
Lecture – 31
Unsupervised Learning

Welcome to fundamentals of artificial intelligence we continue our discussion on machine learning. Today we will look at unsupervised learning. Having introduced the area of machine learning in the first lecture we moved on to look at learning from observation. We looked at learning a decision tree and thereafter focused on linear regression. Support vector machines one of the most popular supervised machine learning algorithms was introduced in the last lecture.

We looked at the formulation of a linear support vector machine and thereafter discussed what is called the kernel trick wherein we could deal with linearly non-separable data by using what is called the kernel trick and we looked at how we could use a support vector machine for nonlinear classification. Today we will look at unsupervised learning we would introduce the idea of clustering.

Now unsupervised learning is more challenging than supervised learning for if you recall unsupervised learning is one where we do not have any labels on the data that is coming to us.

(Refer Slide Time: 02:43)



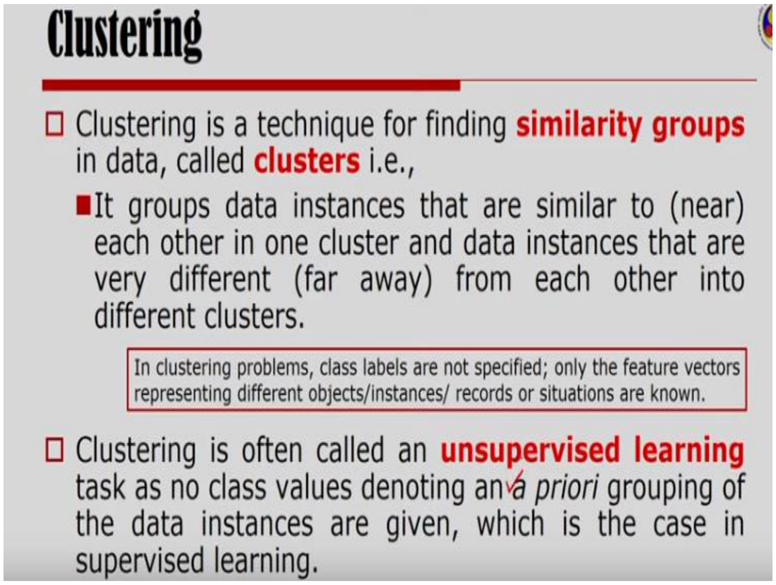
Supervised Learning vs. Unsupervised Learning

- □ Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- □ Unsupervised learning: The data have no target attribute; discover structure underlying the data.
  - Explore the data to find some intrinsic structures in them.
  - Unsupervised learning is more about creative endevours – exploration, understanding and refinement.

Supervised learning on the other hand was about discovering patterns in the data that relate the data attributes with a particular target that is referred to as a class and then these patterns are utilized to predict the values of the target attribute in future data instances. On the other hand, unsupervised learning the data have no target attribute. What we are interested in is about discovering the structure underlying the data.

And we want to somehow explore the data to find some intrinsic structures in them. Now unsupervised learning is more about creative endeavours, exploration, understanding and refinement.

(Refer Slide Time: 03:35)



Clustering

□ Clustering is a technique for finding **similarity groups** in data, called **clusters** i.e.,

- It groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.

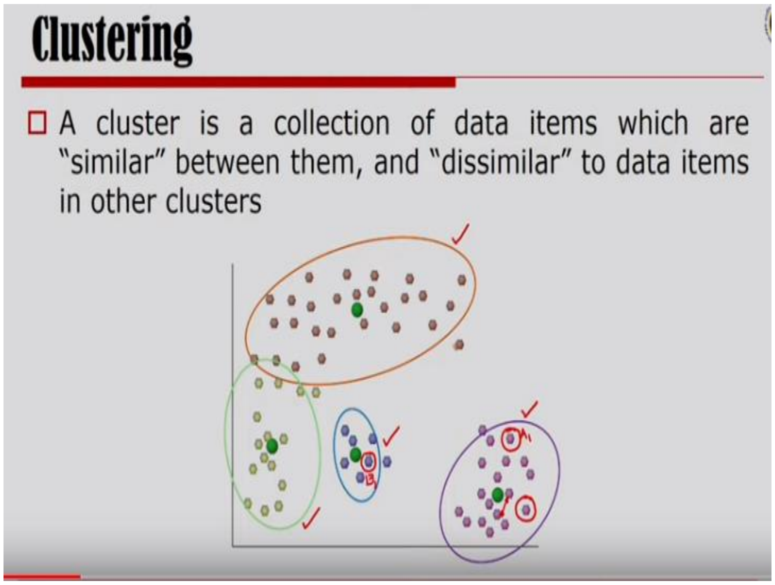
In clustering problems, class labels are not specified; only the feature vectors representing different objects/instances/ records or situations are known.

□ Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.

Let us look at clustering and try to understand unsupervised learning. Our focus here today is more on understanding the intuitive idea underlying this concepts rather than mathematical regard. So clustering is a technique for finding similarity groups in data which are referred to as clusters. So it groups data instances that are similar or near to each other and tasks of data instances to be put in different groups if they are different or far away from each other.

Now in clustering problems as I have been emphasizing class labels are not specified what is given is only the feature vector representing different object instances records or situations. Clustering is often synonymous with unsupervised learning and this is precisely because it takes no class values denoting apriori grouping of the data instances as opposed to what we see in supervised learning.

(Refer Slide Time: 05:00)



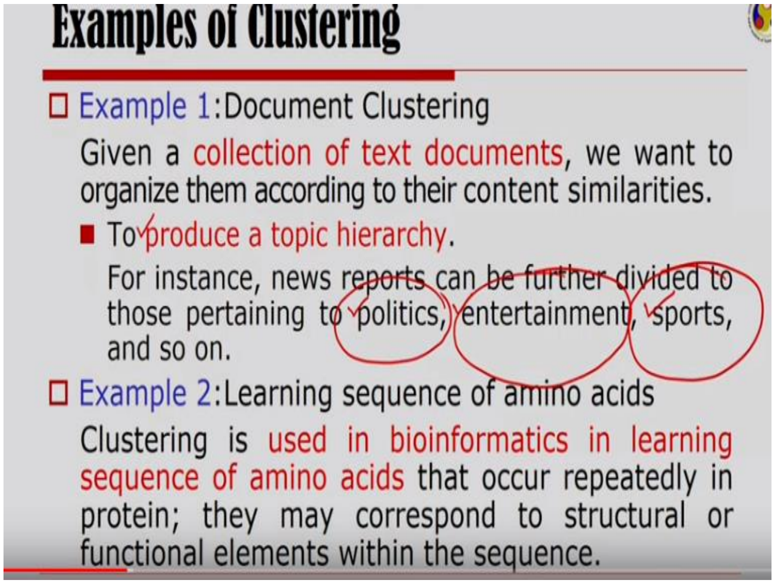
Clustering

□ A cluster is a collection of data items which are "similar" between them, and "dissimilar" to data items in other clusters

Let us look at more closely by what we mean by a cluster. On your screen are shown different colour hexagonal points and if by some measure we could think of identifying distances between them then a little consideration will show that the distance between these points vis a vis the distance between points that are of different colour here are going to be distinctly different. So in a sense if we see all these points that have been put together within this group or this one within this group here or for that matter this group here and this group here.

These data items are similar between themselves whereas they are distinctly dissimilar between the data items in the other crystal. So if you take a point here and if you take a point here these 2 data items are similar whereas if you take the point A1 and the point which is another data item in this group point B1. They are distinctly different so a cluster is a collection of data items which are similar between them and dissimilar to data items in other clusters and it is appropriate that we now take a minute and try to understand the clusters that we encounter in our daily life.

(Refer Slide Time: 07:14)



Examples of Clustering

- □ Example 1:Document Clustering

  Given a collection of text documents, we want to organize them according to their content similarities.

  - ■ To produce a topic hierarchy.

    For instance, news reports can be further divided to those pertaining to politics, entertainment, sports, and so on.

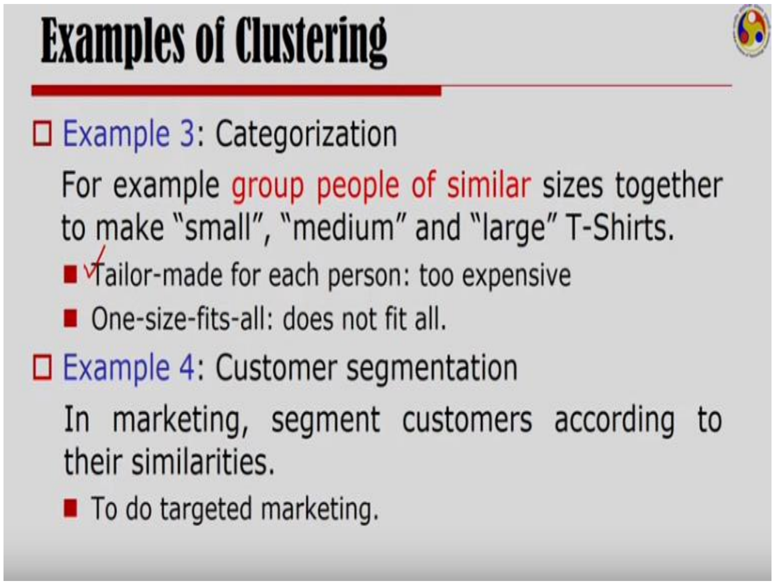
- □ Example 2:Learning sequence of amino acids

  Clustering is used in bioinformatics in learning sequence of amino acids that occur repeatedly in protein; they may correspond to structural or functional elements within the sequence.

So some common cluster examples could be the document clustering. So if you are given a collection of text documents and we want to organize them according to their content similarity so we will end up producing a topic hierarchy. Let us take a minute to understand what we mean by this for instance you can think of all the news reports. That you see in a newspaper or elsewhere.

And all these newspaper reports can be divided into those pertaining to politics to entertainment sports and so on. So each of these is actually a cluster of all the text documents of the newspaper reports that I have collected. Another example of clustering which is hugely changing the way learning in bioinformatics is happening is learning sequence of amino acids. So clustering is used in bioinformatics in learning sequence of amino acids that occur repeatedly in protein. Now they may correspond to structural or functional elements within the sequence.

(Refer Slide Time: 08:59)



Examples of Clustering

- □ Example 3: Categorization

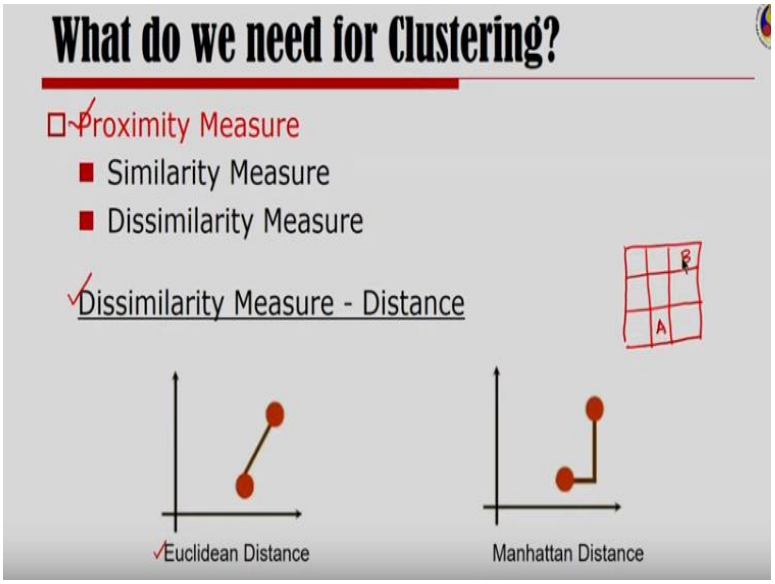
  For example group people of similar sizes together to make "small", "medium" and "large" T-Shirts.
  - Tailor-made for each person: too expensive
  - One-size-fits-all: does not fit all.
- □ Example 4: Customer segmentation

  In marketing, segment customers according to their similarities.
  - To do targeted marketing.

Other examples could be like why you have garments in typical sizes of small, medium and large and extra-large. Like for example if you are looking for a t-shirt all you will get is something which is small, medium, large or extra-large. So people are grouped in terms of their similarity in size and those are actually clusters. We do not have garments tailor-made for each person when I am buying it off the shelf that would be too expensive one size fit all also does not work and therefore what is done is you create clusters or groups of people of similar sizes.

Another example could be customer segmentation where in marketing you segment customers according to their similarities and this is particularly done for targeted marketing. So there are many examples of clustering or clusters that are used in day-to-day activities what we will today look at is what do we need in order to create clusters or what are the basic elements that go into creating a cluster.

(Refer Slide Time: 10:35)



What do we need for Clustering?

□ ✓Proximity Measure

- Similarity Measure
- Dissimilarity Measure

✓Dissimilarity Measure - Distance

✓Euclidean Distance

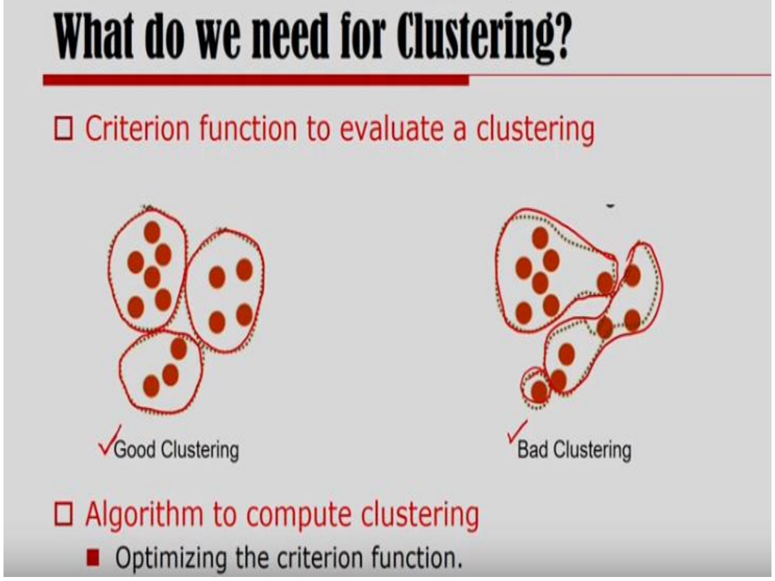
Manhattan Distance

So the first thing that we have been informally talking of is about some notion of similarity. Now it could be similarity or it could be dissimilarity measure as well where anything that is dissimilar I would love to put in different clusters and therefore we referred to that distance as proximity measure. So we would need some idea of a proximity measure to talk of cluster or in order to perform clustering.

So proximity measure could be a measure of similarity or it could be a measure of dissimilarity. Dissimilarity measure is some form of a distance that I can find out between the data points and the distance could be the Euclidean distance or the Manhattan distance, which is the distance that we measure by measuring the number of blocks. So the Euclidean distance is the distance between the point X and Y and the Manhattan distance is a distance in a blocks wall usually.

So if I have my whole space divided into blocks like this and I have a point A here and another point B here then the Manhattan distance would be the total of moving down here and then moving to A. So that is the Manhattan distance that we can use to find out dissimilarity.

(Refer Slide Time: 12:43)



What do we need for Clustering?

- □ Criterion function to evaluate a clustering

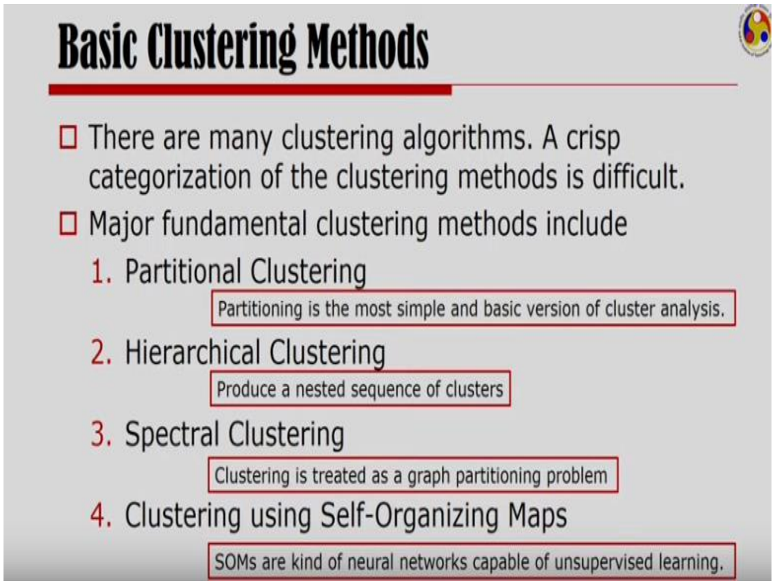
- □ Algorithm to compute clustering
  - Optimizing the criterion function.

Now other criteria that would be required is some function to evaluate a clustering like on your screen now I have a group of points and this one here I have grouped them in 3 clear groups one here one down there and a third here. Whereas if you look at the group of points on the right of your screen we have again grouped them but then now we had made these groupings where even points which are near each other have been put into different groups.

So this here is an example of good clustering whereas what we have here is bad clustering. So when we are trying to have a criterion function we should get the function to return us good values for clusters that are of the nature of good clusters rather than what I have on the right of the scheme which are bad clusters and this criterion function is actually been optimized by an algorithm to compute clustering.

So the clustering problem in its most simplest terms is about having some proximity measure. So as I can identify the similarity or dissimilarity between data points and thereafter have a function that can allow me to put these points together in one cluster and the clustering algorithm the idea is to optimize this criterion function. There are many clustering algorithms and a very crisp categorization of the clustering algorithms is really difficult.

(Refer Slide Time: 15:19)



Basic Clustering Methods

- □ There are many clustering algorithms. A crisp categorization of the clustering methods is difficult.
- □ Major fundamental clustering methods include
  1. Partitional Clustering

     Partitioning is the most simple and basic version of cluster analysis.
  2. Hierarchical Clustering

     Produce a nested sequence of clusters
  3. Spectral Clustering

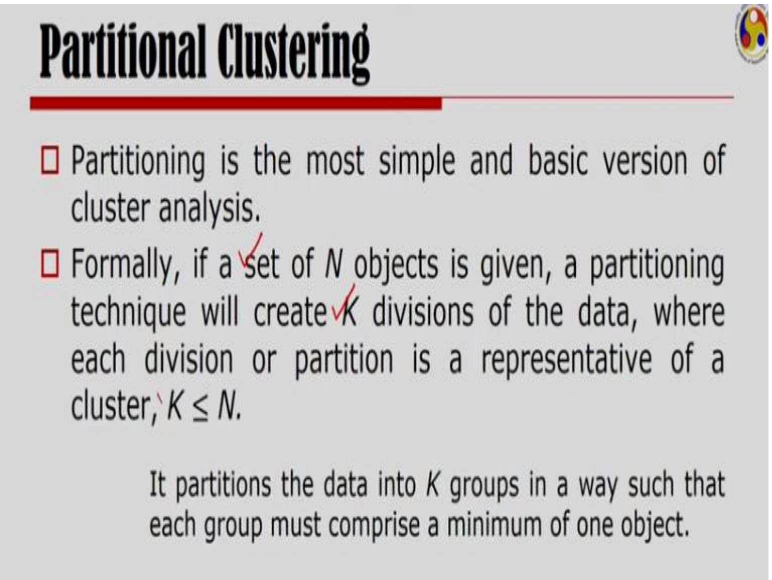
     Clustering is treated as a graph partitioning problem
  4. Clustering using Self-Organizing Maps

     SOMs are kind of neural networks capable of unsupervised learning.

However major fundamental clustering methods include one of the following four we have what is referred to as partitional clustering. Partitional clustering is possibly the most simple and basic version of cluster analysis. Thereafter we have what is called hierarchical clustering which produce a nested sequence of clusters. Spectral clustering is when you treat clustering as a graph partitioning problem and then we have clustering using self-organizing maps.

The self-organizing maps are kind of neural networks capable of unsupervised learning. We will now quickly look at partitional clustering. Now partitioning or partitional clustering as I have already mentioned is the most simple and basic version of cluster analysis.

(Refer Slide Time: 16:37)



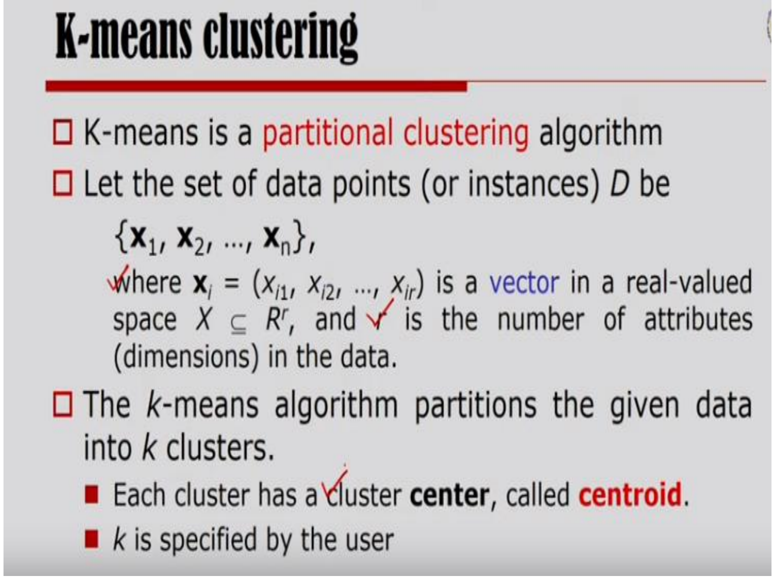
Partitional Clustering

- □ Partitioning is the most simple and basic version of cluster analysis.
- □ Formally, if a set of N objects is given, a partitioning technique will create K divisions of the data, where each division or partition is a representative of a cluster, $K \leq N$.

It partitions the data into K groups in a way such that each group must comprise a minimum of one object.

If I have a set of N objects a partitioning technique will create K divisions of the data and each division or partition is a representative of a cluster. Now obviously K can be only less than or equal to N it partitions the data into k groups in a way such that each group must comprise a minimum of one object.

(Refer Slide Time: 17:14)



K-means clustering

□ K-means is a partitional clustering algorithm

□ Let the set of data points (or instances) *D* be

$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

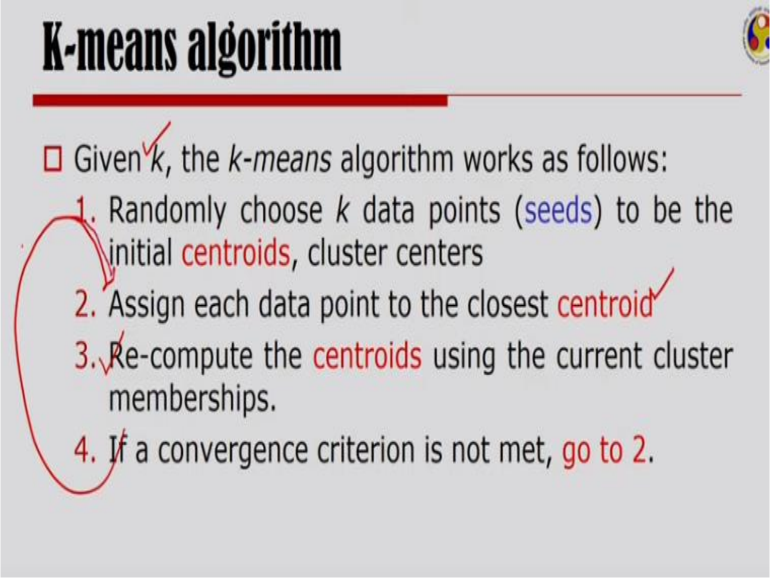
where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.

□ The *k*-means algorithm partitions the given data into *k* clusters.

- Each cluster has a cluster **center**, called **centroid**.
- *k* is specified by the user

K-means clustering that we will now look at is a partitioning clustering algorithm. Let the set of data points or instances be x1 to xn. Now each xi that I am talking of as a data point is a vector in a real-valued space and we have for each x some r number of attributes. The K-means algorithm partitions the given data into K clusters each cluster has what is called a cluster centre or a centroid or a centroid and the number of cluster that is required which is k is specified by the user.

(Refer Slide Time: 18:15)



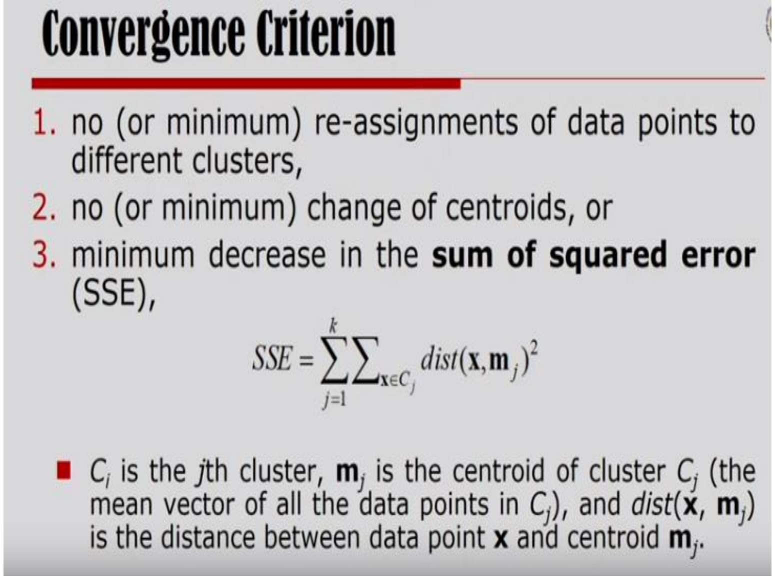
K-means algorithm

- □ Given *k*, the *k-means* algorithm works as follows:
  1. Randomly choose *k* data points (seeds) to be the initial centroids, cluster centers
  2. Assign each data point to the closest centroid
  3. Re-compute the centroids using the current cluster memberships.
  4. If a convergence criterion is not met, go to 2.

Now the K-means algorithm works as follows you are given k the number of clusters so you randomly choose k data points to be the initial centroids or the cluster centres and then you start assigning each data point to the closest centroid. Once that is done you recompute the centroids using the current cluster membership and then you have certain convergence criteria. If the convergence criteria is not met you go back to step 2 and do this process all over again.

That is you again start with assigning data points to the closest centroid and recompute the centroids using the current cluster membership and look for if the convergence criteria is being met.

(Refer Slide Time: 19:26)



Convergence Criterion

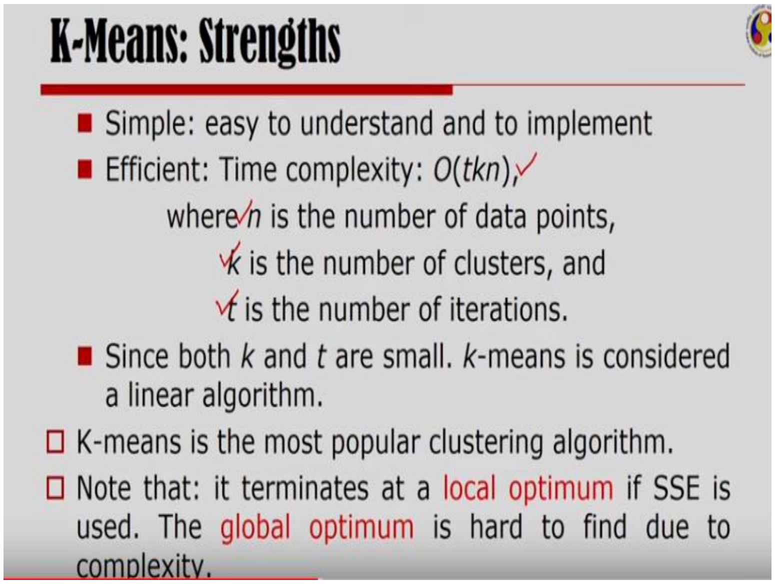
1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE),

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

- C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

The convergence criteria could be one of the following. It could be that no reassignment of data points to different clusters is required or a certain minimum beyond a predefined threshold is only required or it could be that there is no change of the centroid. One convergence criteria that is widely used is looking at the minimum decrease in the sum of squared error which is you find out the error for each point from the centroid of the cluster and get the sum of the square which is shown here and if there is a minimum decrease in there then it could be a sign of convergence.

(Refer Slide Time: 20:20)



K-Means: Strengths

- ■ Simple: easy to understand and to implement
- ■ Efficient: Time complexity: *O*(*tkn*),✓
  - where✓*n* is the number of data points,
  - ✓*k* is the number of clusters, and
  - ✓*t* is the number of iterations.
- ■ Since both *k* and *t* are small. *k*-means is considered a linear algorithm.

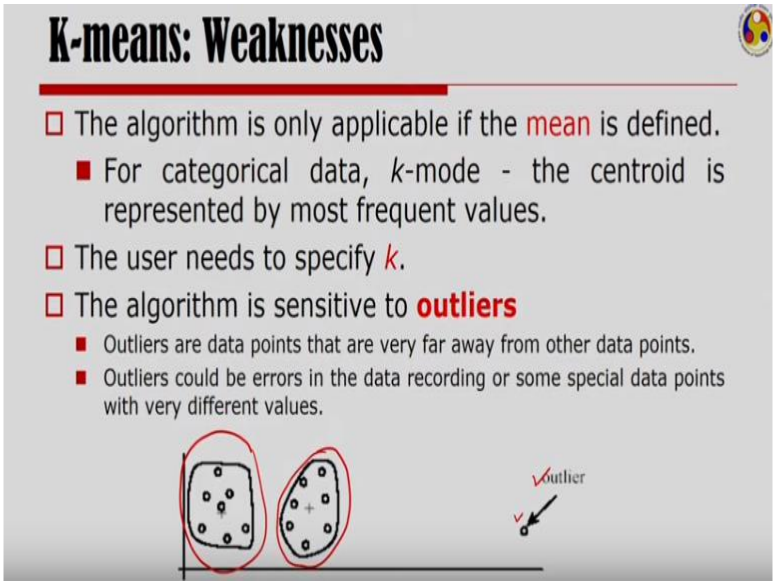
□ K-means is the most popular clustering algorithm.

□ Note that: it terminates at a local optimum if SSE is used. The global optimum is hard to find due to complexity.

Now K-means is a very simple easy to understand and implement algorithm. K-means is efficient the time complexity of K-means is of the order of tkn where n is the number of data points, k is the number of clusters and t is the number of iterations. Now since both k and t are small K-means is considered a linear algorithm and in fact K-means is one of the most popular clustering algorithms.

It terminates at a local optimum if the squared sum of error is used and the global optimum is hard to find due to the complexity and that is something which is problematic in K-means.

(Refer Slide Time: 21:22)



K-means: Weaknesses

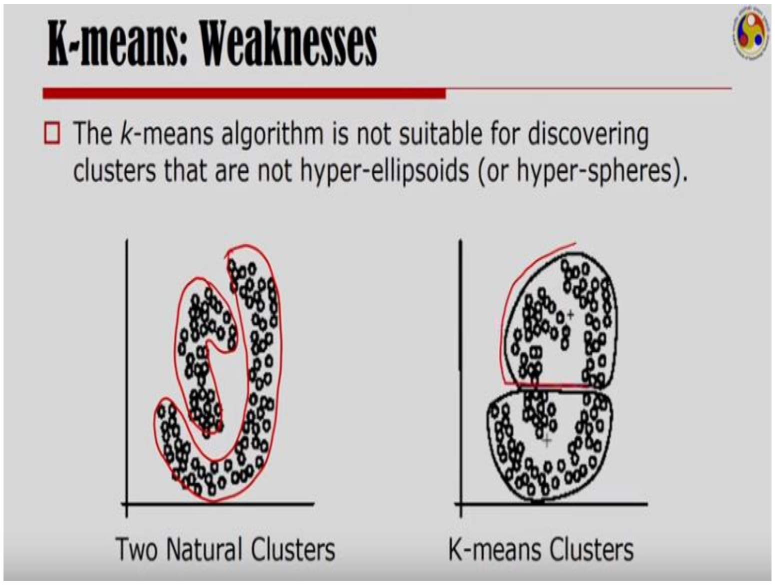
- □ The algorithm is only applicable if the mean is defined.
  - For categorical data, *k*-mode - the centroid is represented by most frequent values.
- □ The user needs to specify *k*.
- □ The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

In terms of its other weaknesses the algorithm is only applicable if the mean is defined. So for categorical data k mode is what is used the centroid is represented by the most frequent values K-means weakness also includes the fact that the user needs to specify the number of clusters to start with that is k. One more very serious problem it K-means is that the algorithm is sensitive to outliers.

Now what are outliers is data points that are far away from other data points. Now here is a data point that is an outlier and if I was looking for clustering here and somehow this point is included in the cluster I would end up having a very bad cluster whereas ideally these points should have been left out and I should have got my 2 very clear clusters here. Now these points could be errors in the data recording or some spatial data points with very different values it may be of interest to find these points as some amount of novelty detection.

But when we are talking of clustering then these points need not be put into a cluster and finally giving me a bad cluster. So this is something I will not be looking for and that is a problem K-means is sensitive to such points.

(Refer Slide Time: 23:22)

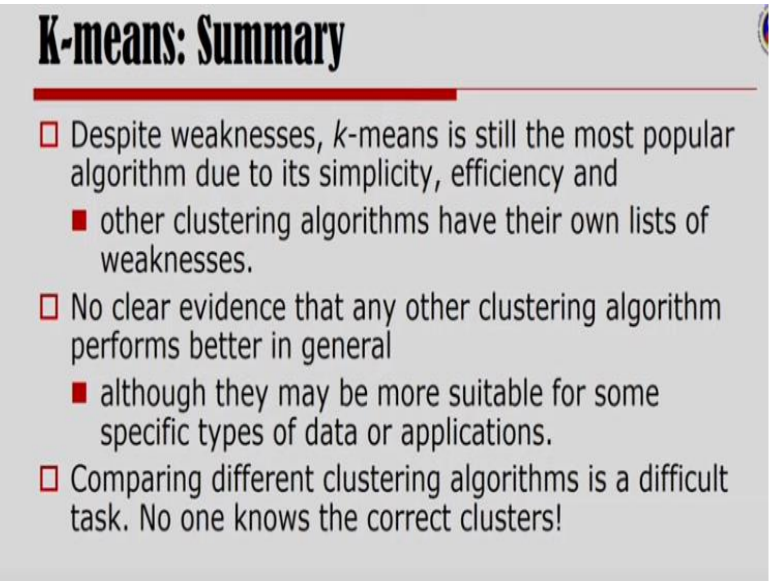


K-means: Weaknesses

□ The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres).

The K-means algorithm is not suitable for discovering clusters that are not hyper ellipsoids or hyper spheres. So here if you look at what I have on your screen is 2 very clear natural clusters one inside somehow the other but if I am applying the K-means clustering algorithm all I will end up is having some clusters of this nature and this is a serious weakness of the K-means algorithm.

(Refer Slide Time: 24:17)



K-means: Summary

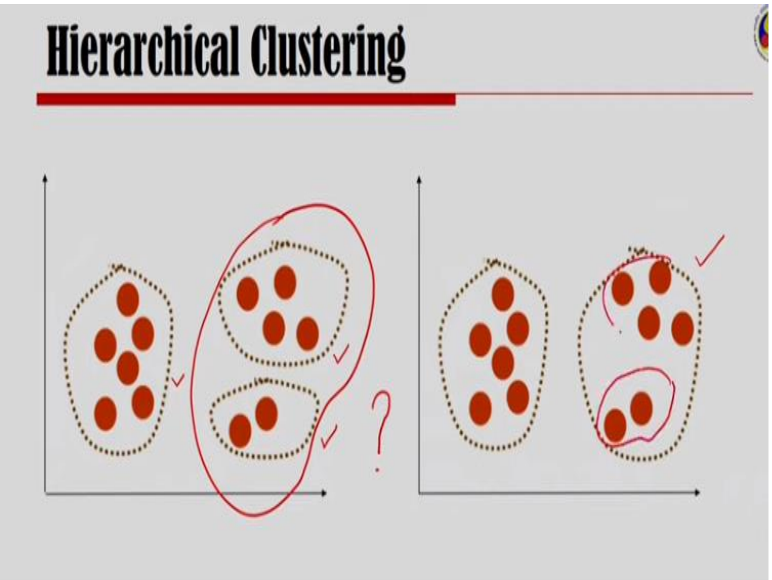
□ Despite weaknesses, *k*-means is still the most popular algorithm due to its simplicity, efficiency and
  - ■ other clustering algorithms have their own lists of weaknesses.

□ No clear evidence that any other clustering algorithm performs better in general
  - ■ although they may be more suitable for some specific types of data or applications.

□ Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Despite weaknesses K-means is still the most popular algorithm due to its simplicity, efficiency. No clear evidence is there that any other clustering algorithm performs better in general. Although many specialized algorithms may be suitable for a specific type of data or application. In fact comparing different clustering algorithms itself is a difficult task for no one knows which are the correct clusters. So let us now focus on the second clustering algorithm that we will introduce in a this is the hierarchical clustering.

(Refer Slide Time: 25:05)

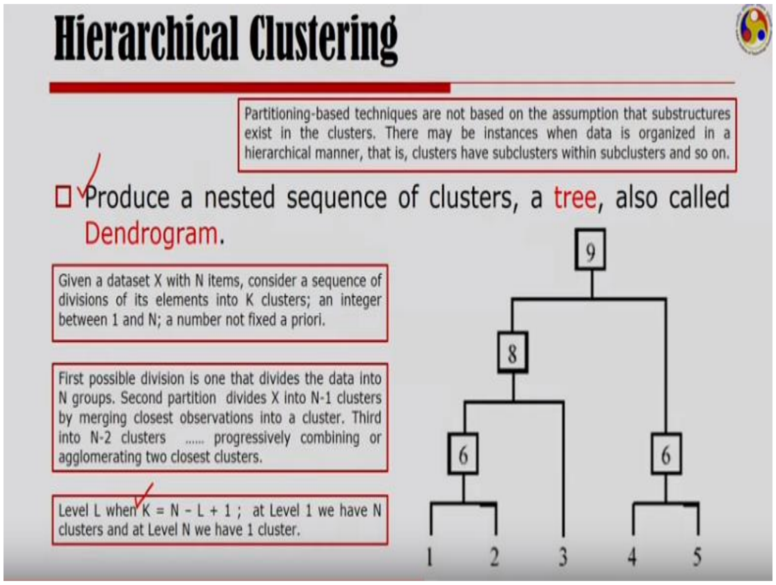


Hierarchical Clustering

So here on your screens I have shown a result of clustering which is either one given on the left or one given on the right. Now the question is which clustering result would you prefer. Given a choice here this is of course trying to identify 3 clusters, this of course has identified the very fact that these points are together and therefore has put it together in this cluster, but somehow lost the very idea that these also individually are very close and could have been put together that is the idea in hierarchical clustering.

In hierarchical clustering you would love to identify the sub clusters and then put the sub clusters together to get to the main cluster and that is what makes it very powerful clustering algorithm.

(Refer Slide Time: 26:18)



Hierarchical Clustering

Partitioning-based techniques are not based on the assumption that substructures exist in the clusters. There may be instances when data is organized in a hierarchical manner, that is, clusters have subclusters within subclusters and so on.

□ Produce a nested sequence of clusters, a tree, also called Dendrogram.

Given a dataset X with N items, consider a sequence of divisions of its elements into K clusters; an integer between 1 and N; a number not fixed a priori.

First possible division is one that divides the data into N groups. Second partition divides X into N-1 clusters by merging closest observations into a cluster. Third into N-2 clusters ...... progressively combining or agglomerating two closest clusters.

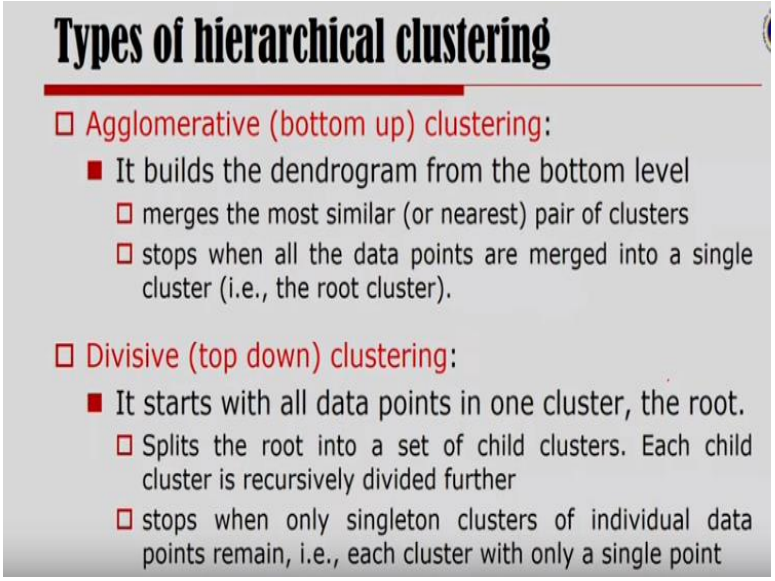
Level L when K = N - L + 1 ; at Level 1 we have N clusters and at Level N we have 1 cluster.

So partitioning based techniques are actually not based on the assumption that sub clusters exist in the cluster whereas there may be instances when data is organized in a hierarchical manner that is clusters have sub clusters within some clusters and so on as I have shown in the previous slide and therefore getting to this sub clusters is very important. This is what is exactly done in hierarchical clustering and it produces a nested sequence of clusters a tree called a Dendrogram.

So given a data set X with N items it consider a sequence of division of its elements into K clusters which is an integer between 1 and N but then that is not a fixed number apriori. The first possible division could be one that divides the data into N groups the second partition could divide X into N-1 clusters by merging the closest observation into a cluster. The third partition could be resulting in N-2 cluster and so on and so forth.

Progressively combining or what is referred to as a agglomerating 2 closest clusters. So if I am at level L, at level L K would be given as N-L+1 so if I am at level one we have N clusters whereas if I am at level N, I have finally a single cluster and this nested sequence of clusters is represented in form of a tree called the Dendrogram.

(Refer Slide Time: 28:18)



Types of hierarchical clustering

□ Agglomerative (bottom up) clustering:

- ■ It builds the dendrogram from the bottom level
  - □ merges the most similar (or nearest) pair of clusters
  - □ stops when all the data points are merged into a single cluster (i.e., the root cluster).

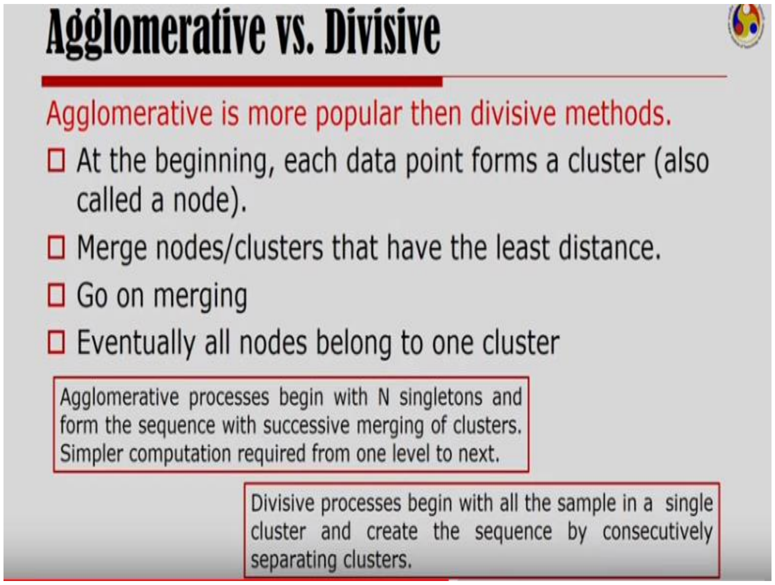
□ Divisive (top down) clustering:

- ■ It starts with all data points in one cluster, the root.
  - □ Splits the root into a set of child clusters. Each child cluster is recursively divided further
  - □ stops when only singleton clusters of individual data points remain, i.e., each cluster with only a single point

As I was telling you, the hierarchical clustering could proceed from combining sub clusters as it comes or a bottom-up process which is referred to as the agglomerative clustering or it could be about a top down process which is a divisive clustering idea. In agglomerative clustering, we build the Dendrogram bottom up that is merge the most similar or nearest pair of clusters and stop when all the data points are merged into a single cluster and that single cluster is the root cluster.

Whereas the divisive clustering starts with all data points in one cluster the root and then it starts splitting the root into set of child clusters and each child cluster is then again divided further and finally stops when only singleton clusters of individual data points remain that is each cluster with only a single point.

(Refer Slide Time: 30:24)



Agglomerative vs. Divisive

Agglomerative is more popular then divisive methods.

- □ At the beginning, each data point forms a cluster (also called a node).
- □ Merge nodes/clusters that have the least distance.
- □ Go on merging
- □ Eventually all nodes belong to one cluster

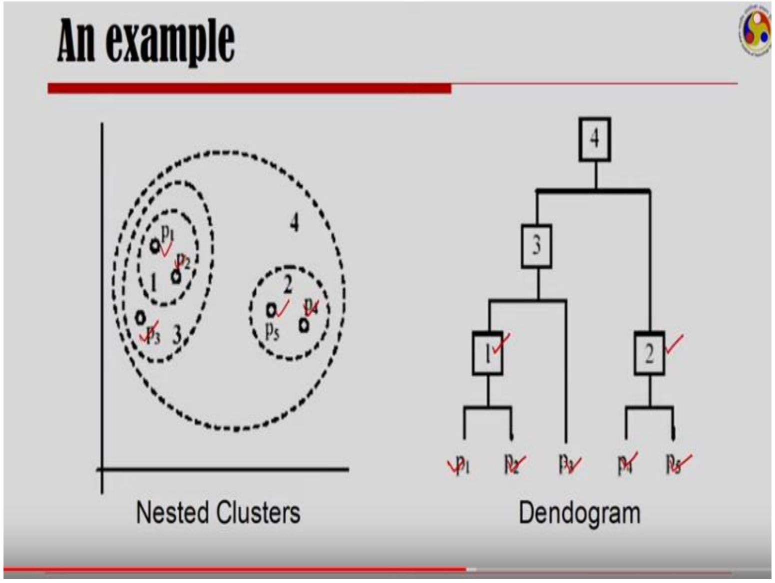
Agglomerative processes begin with N singletons and form the sequence with successive merging of clusters. Simpler computation required from one level to next.

Divisive processes begin with all the sample in a single cluster and create the sequence by consecutively separating clusters.

Agglomerative clustering is more popular than the divisive methods. At the beginning each data point forms a cluster you merge the clusters that have the least distance and go on merging eventually all nodes belong to one cluster. So agglomerative processes begin with N singletons and form the sequence with successive merging of clusters. Agglomerative clustering require simpler computation as we move from one level to the next.

The divisive process begins with all sample in a single cluster and create the sequence by consecutively separating out the clusters.

(Refer Slide Time: 30:27)



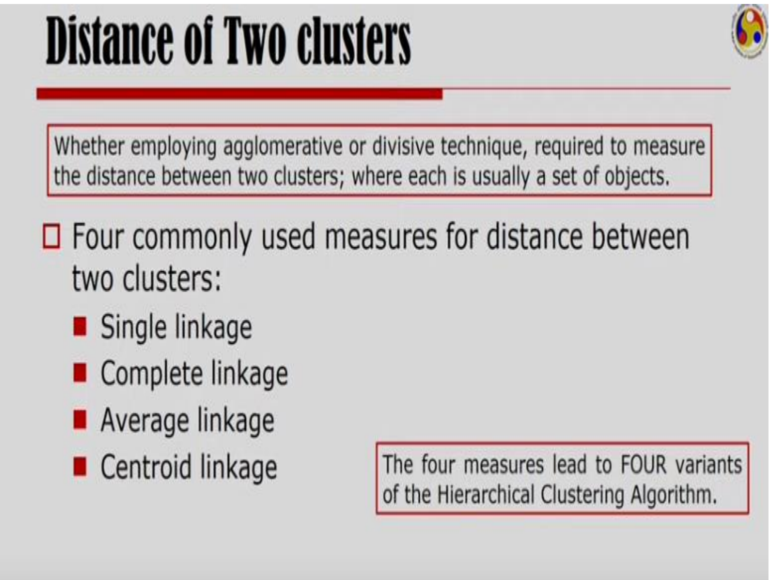
An example

Nested Clusters

Dendogram

Here is an example I have the next state cluster the points p1, p2, p3, p4 and p5. So those are each individually down there then p1 and p2 is put together p4 p5 put into one cluster. Then the cluster that I created reading p1,p2 merge with p3 to create cluster 3 this is shown here. Thereafter the cluster 2 and cluster 3 put together to have the complete cluster form. So here is the nested clusters on your left and the Dendrogram on your right and that is the idea of hierarchical clustering.

(Refer Slide Time: 31:31)



Distance of Two clusters

Whether employing agglomerative or divisive technique, required to measure the distance between two clusters; where each is usually a set of objects.

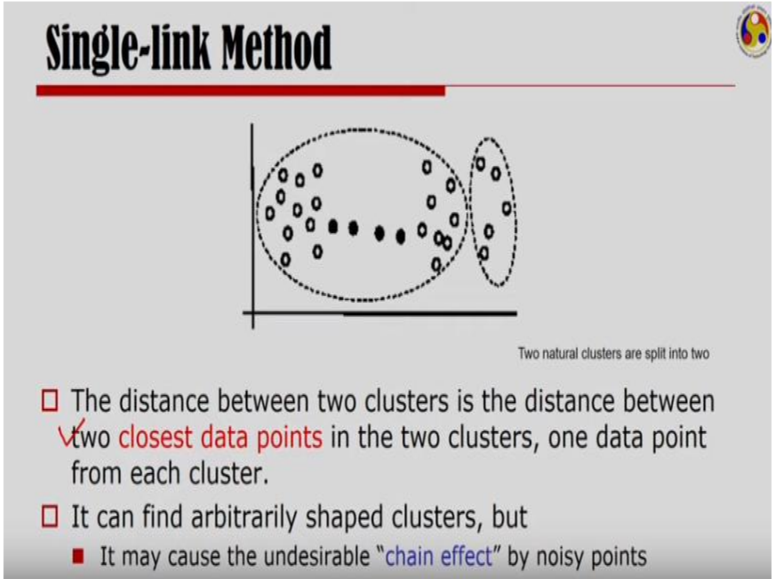
□ Four commonly used measures for distance between two clusters:

- Single linkage
- Complete linkage
- Average linkage
- Centroid linkage

The four measures lead to FOUR variants of the Hierarchical Clustering Algorithm.

Now when I am talking of hierarchical clustering whether I am employing agglomerative or divisive technique, I am required to measure the distance between 2 clusters where each is usually a set of objects. So there are 4 commonly used measures for distance between 2 clusters they lend up to different variants of the hierarchical clustering algorithm. The 4 commonly used measures are referred to as single linkage, complete linkage, average linkage and centroid.

(Refer Slide Time: 32:17)



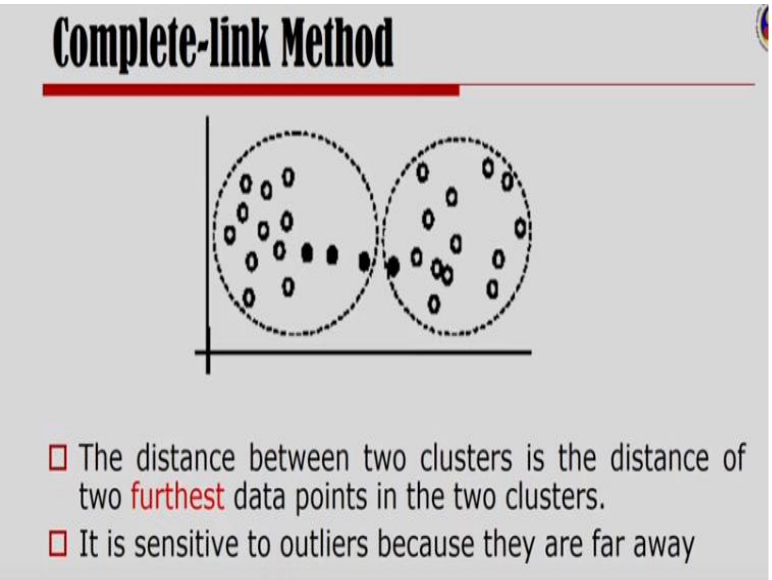
Single-link Method

Two natural clusters are split into two

- □ The distance between two clusters is the distance between two closest data points in the two clusters, one data point from each cluster.
- □ It can find arbitrarily shaped clusters, but
  - ■ It may cause the undesirable "chain effect" by noisy points

We will look at each of them one by one. So in single linkage method the distance between 2 clusters is the distance between 2 closest data points in the 2 cluster one data point from each. Now it can find arbitrarily shaped clusters but it may cause the undesirable chain effect if it had very noisy points. For I am trying to get a distance between the clusters by looking at the closest data points between the two of them.

(Refer Slide Time: 32:56)

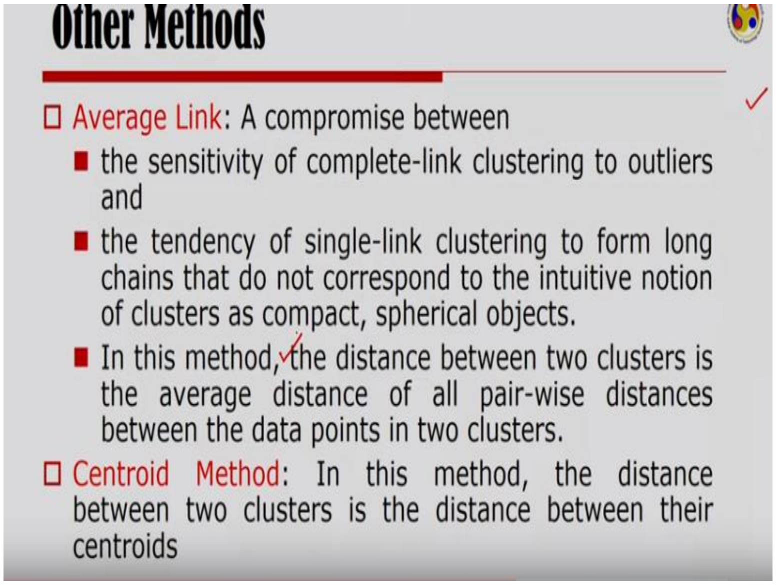


Complete-link Method

- □ The distance between two clusters is the distance of two furthest data points in the two clusters.
- □ It is sensitive to outliers because they are far away

The complete link method on the other hand is about the distance between 2 clusters where I am looking for the distance between the 2 farthest point in the 2 clusters. Now as you can see when I am thinking of using complete link method for agglomeration such a method trying to get the distance between two furthest data points is sensitive to outliers.

(Refer Slide Time: 33:30)



Other Methods

✓

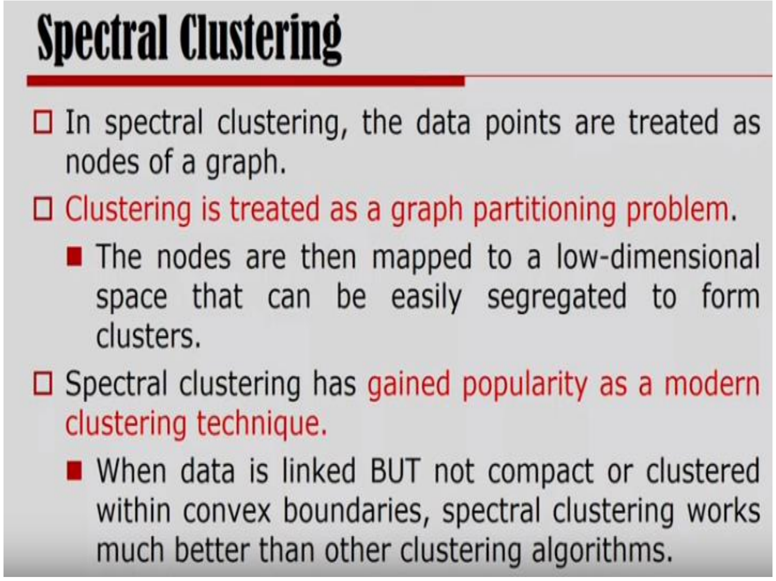
- □ Average Link: A compromise between
  - the sensitivity of complete-link clustering to outliers and
  - the tendency of single-link clustering to form long chains that do not correspond to the intuitive notion of clusters as compact, spherical objects.
  - In this method,✓the distance between two clusters is the average distance of all pair-wise distances between the data points in two clusters.
- □ Centroid Method: In this method, the distance between two clusters is the distance between their centroids

The average link method is a compromise between the sensitivity of the complete link clustering to outliers and the tendency of the single link clustering to form long chains that do not correspond to the intuitive notion of a cluster as certain compact spherical objects. Now let us take a minute to understand what we mean by that if you recall the single link method was about finding out the distance between two points which are nearest to each other.

So it could get this and thereafter it could again get to this as a single cluster and get a point here nearest to this and get this and get another point so it could create a chain. Whereas the complete link clustering it is obvious that since it is talking of farthest point distances and if I have outlines there. So I would be talking of complete link clustering where it would be very sensitive to such points which are the outliers.

In this method called the average link the distance between 2 clusters is the average distance of all pairwise distance between the data points and this looks to be a far better measure than a single link or a complete link measure of distance. The centroid method on the other hand is about finding out the distance between 2clusters in terms of the distance between their centroids. Now having looked at hierarchical clustering and K-means which is a partitional clustering let me quickly give you the intuition behind spectral clustering.

(Refer Slide Time: 35:44)

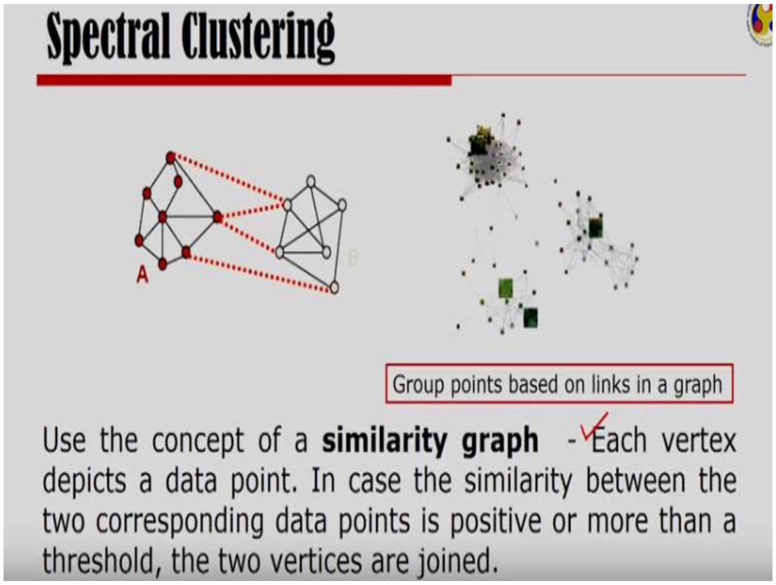


Spectral Clustering

- □ In spectral clustering, the data points are treated as nodes of a graph.
- □ Clustering is treated as a graph partitioning problem.
  - The nodes are then mapped to a low-dimensional space that can be easily segregated to form clusters.
- □ Spectral clustering has gained popularity as a modern clustering technique.
  - When data is linked BUT not compact or clustered within convex boundaries, spectral clustering works much better than other clustering algorithms.

This is because spectral clustering is increasingly gaining popularity as a modern clustering technique. In spectral clustering data points are treated as nodes of a graph and the clustering is treated as a graph partitioning problem. The nodes are then mapped to a low dimensional space that can be easily segregated to form clusters. Now when data is linked but not compact or clustered within convex boundaries, spectral clustering works much better than other clustering algorithms and this is why it is emerging as a very popular modern clustering technique.

(Refer Slide Time: 36:34)



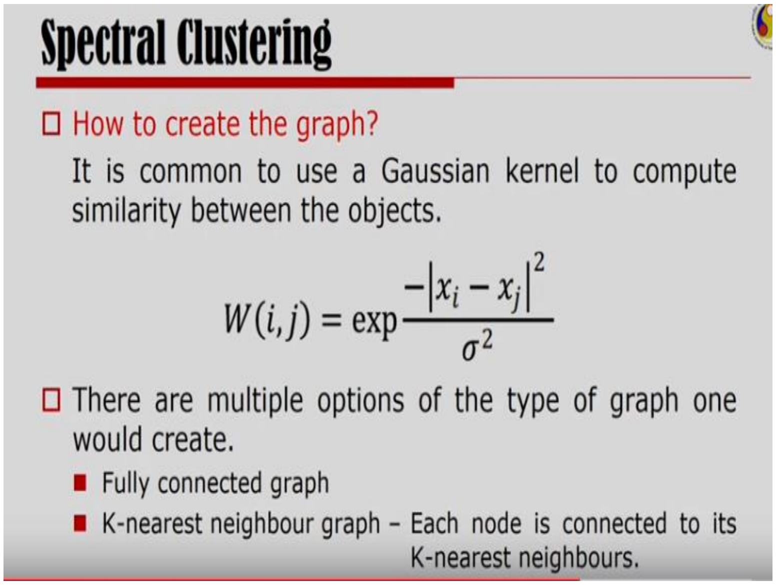
Spectral Clustering

Group points based on links in a graph

Use the concept of a **similarity graph** - Each vertex depicts a data point. In case the similarity between the two corresponding data points is positive or more than a threshold, the two vertices are joined.

Now here is what we have for spectral clustering, we group points based on links in a graph that is we use the concept of what is called a similarity graph where each vertex of the graph depicts a data point. In case the similarity between 2 correspondent data point is positive or more than a threshold the 2 vertices are joined and we seek partitions of the graph when edges between the various groups have extremely low similarity. This indicates dissimilarity between the points in the various clusters. So edges within the group have high similarity weights.

(Refer Slide Time: 37:30)



Spectral Clustering

□ How to create the graph?

It is common to use a Gaussian kernel to compute similarity between the objects.

$$W(i,j) = \exp\frac{-\left|x_i - x_j\right|^2}{\sigma^2}$$

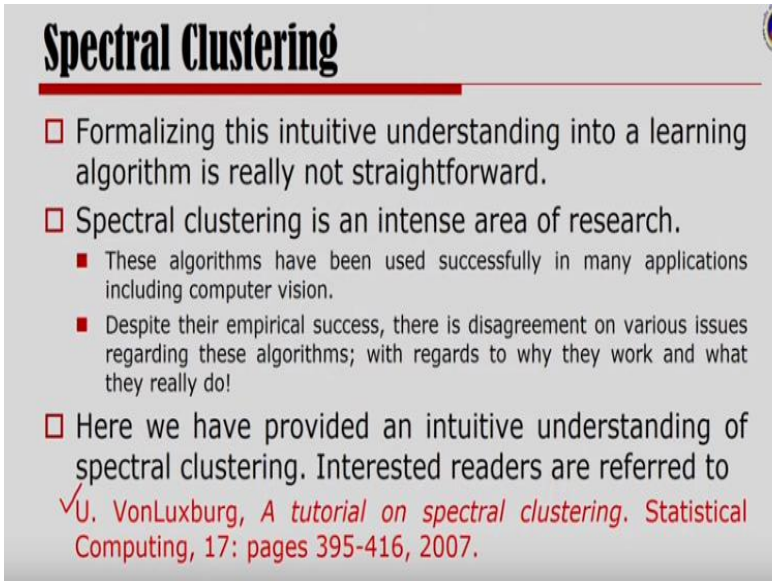
□ There are multiple options of the type of graph one would create.

■ Fully connected graph

■ K-nearest neighbour graph – Each node is connected to its K-nearest neighbours.

Question is how do we create the graph, so in order to create the graph we use a Gaussian kernel to compute similarity between the objects and there are multiple options of the type of graph one would create. One can end up having a fully connected graph or usually what is called the k nearest neighbour graph the k nearest neighbour graph is one in which each node is connected to its k nearest neighbours.

(Refer Slide Time: 38:12)



Spectral Clustering

□ Formalizing this intuitive understanding into a learning algorithm is really not straightforward.

□ Spectral clustering is an intense area of research.

■ These algorithms have been used successfully in many applications including computer vision.

■ Despite their empirical success, there is disagreement on various issues regarding these algorithms; with regards to why they work and what they really do!

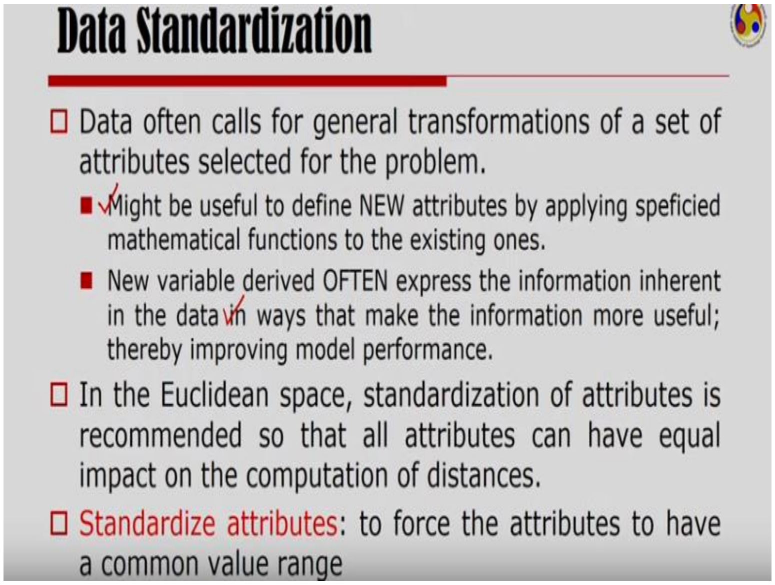
□ Here we have provided an intuitive understanding of spectral clustering. Interested readers are referred to

✓U. VonLuxburg, *A tutorial on spectral clustering*. Statistical Computing, 17: pages 395-416, 2007.

Now formalizing this intuitive understanding into a learning algorithm is really not straightforward. Spectral clustering is an intense area of research these algorithms have been successfully used in many applications including computer vision. Despite their empirical success there is no general consensus on why and how it works and that is why it is an area of intense research.

Here in our discussion today we have just given you an intuitive understanding of spectral clustering interested readers are referred to this brilliant tutorial on spectral clustering by VonLuxburg, which is published in statistical computing pages 395 to 416 of 2007. Now when we are talking of unsupervised learning and clustering it is important to take note of how one go about dealing with the data. So data standardization or what is called data transformation is a very important step here.

(Refer Slide Time: 39:47)

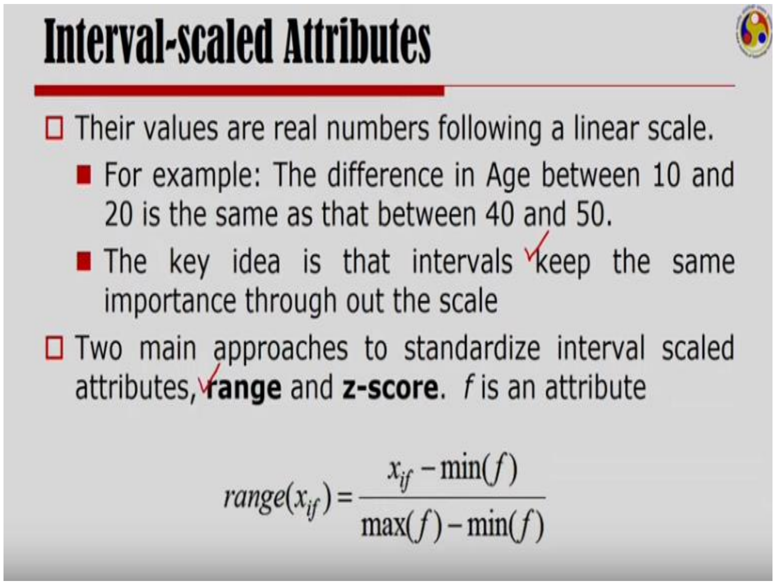


Data Standardization

- □ Data often calls for general transformations of a set of attributes selected for the problem.
  - ■ ✓Might be useful to define NEW attributes by applying speficied mathematical functions to the existing ones.
  - ■ New variable derived OFTEN express the information inherent in the data in ways that make the information more useful; thereby improving model performance.
- □ In the Euclidean space, standardization of attributes is recommended so that all attributes can have equal impact on the computation of distances.
- □ Standardize attributes: to force the attributes to have a common value range

Data often calls for general transformation of a set of attributes selected for the problem. So it might be useful to define a new attribute by applying specified mathematical functions to the existing ones and often the new variable derive express the information inherent in the data in ways that make the information more useful and thereby improving the model performance. In the Euclidean space standardization of attributes is recommended, so that all attributes can have equal impact on the computation of distances. Standardized attributes refer to forcing the attributes to have a common value range.

(Refer Slide Time: 40:45)



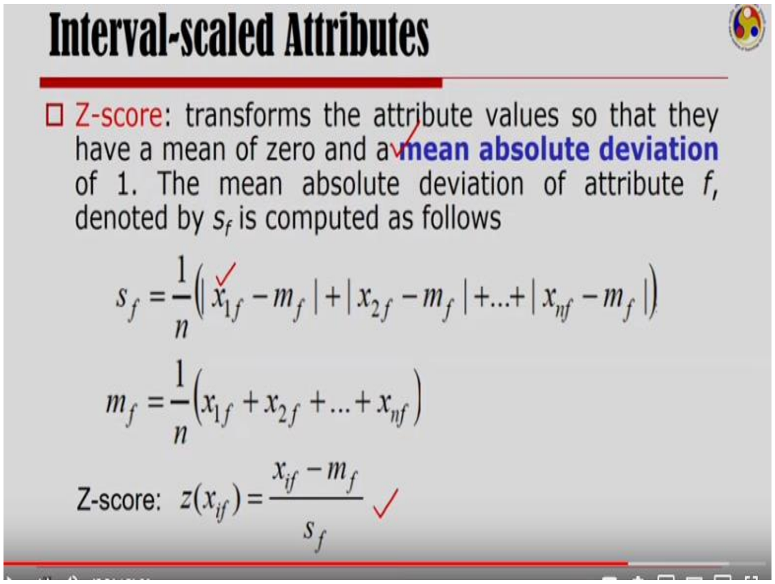
Interval-scaled Attributes

- □ Their values are real numbers following a linear scale.
  - For example: The difference in Age between 10 and 20 is the same as that between 40 and 50.
  - The key idea is that intervals keep the same importance through out the scale
- □ Two main approaches to standardize interval scaled attributes, **range** and **z-score**. *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)}$$

Now what we have are called interval scale attributes. Their values are real numbers following a linear scale for example the difference in age between 10 and 20 needs to be the same as that between 40 and 50, the key idea is that intervals keep the same importance throughout the scale. So 2 main approaches to standardised interval scale attributes are the range and the z score. So the range is about finding out the difference of that particular data point for that particular attribute with the minimum and then taking its ratio with the complete interval that is between the maximum and the minimum.

(Refer Slide Time: 41:43)



Interval-scaled Attributes

- □ Z-score: transforms the attribute values so that they have a mean of zero and a **mean absolute deviation** of 1. The mean absolute deviation of attribute *f*, denoted by s_f is computed as follows

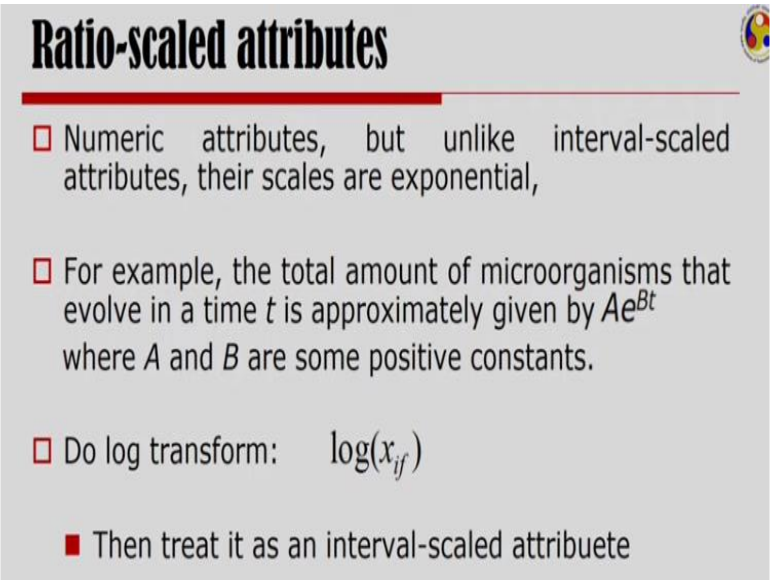
$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right)$$

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right)$$

Z-score: $z(x_{if}) = \frac{x_{if} - m_f}{s_f}$

The z score on the other hand transforms the attribute values so that they have a mean of 0 and a mean absolute deviation of 1. The mean absolute deviation of an attribute is computed by finding out the deviation from the mean for each of the data items and then taking the average of them and then we compute the z score and once we have the z score the attribute values are such that they have a mean of 0.

(Refer Slide Time: 42:28)

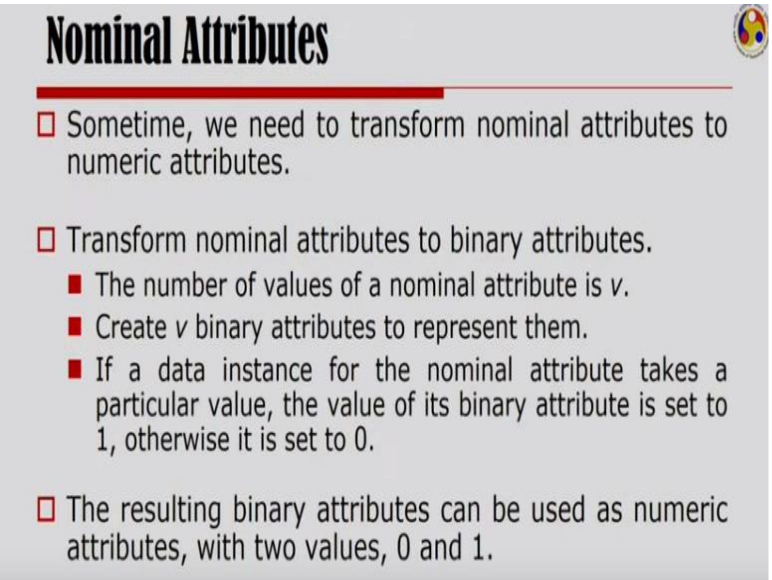


Ratio-scaled attributes

- □ Numeric attributes, but unlike interval-scaled attributes, their scales are exponential,
- □ For example, the total amount of microorganisms that evolve in a time *t* is approximately given by Ae^{Bt} where *A* and *B* are some positive constants.
- □ Do log transform: $\log(x_{if})$
  - ■ Then treat it as an interval-scaled attribuete

The ratio scale attributes on the other hand refer to numeric attributes but unlike interval scale attributes their scales are exponential. So for example we can think of the total amount of microorganisms that evolve in a time T is approximately given by A e to the power Bt where A and B are some positive constants. In that case you take a log transform and then once you take a log transform treat it as an interval scaled attribute.

(Refer Slide Time: 43:07)

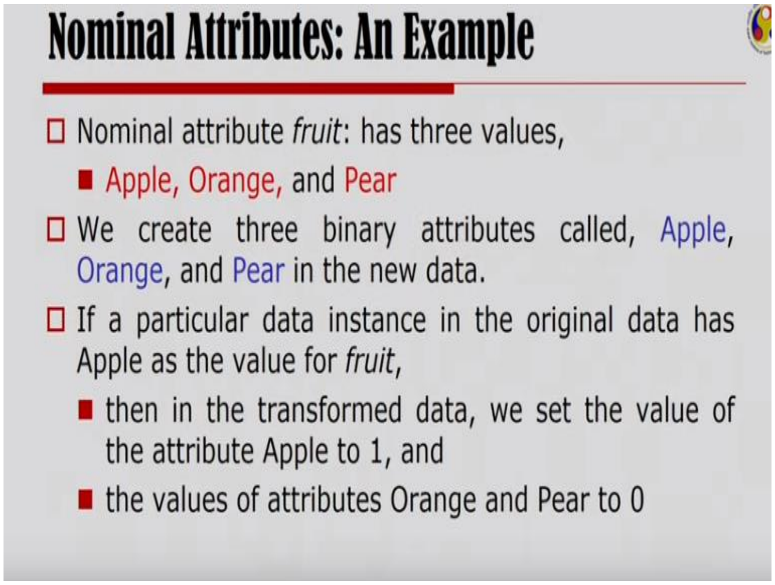


Nominal Attributes

- □ Sometime, we need to transform nominal attributes to numeric attributes.
- □ Transform nominal attributes to binary attributes.
  - The number of values of a nominal attribute is *v*.
  - Create *v* binary attributes to represent them.
  - If a data instance for the nominal attribute takes a particular value, the value of its binary attribute is set to 1, otherwise it is set to 0.
- □ The resulting binary attributes can be used as numeric attributes, with two values, 0 and 1.

Nominal attributes are those which need to be transformed to numeric attributes at times. So we transform nominal attributes to binary attributes. The number of values of nominal attribute let us say is v. We create v binary attributes to represent them and if a data instant for the nominal attribute takes a particular value, the value for its binary attribute is set to 1 otherwise it is set to 0. The resulting binary attributes can be used as numeric attributes with just 2 values 0 and 1.

(Refer Slide Time: 43:52)

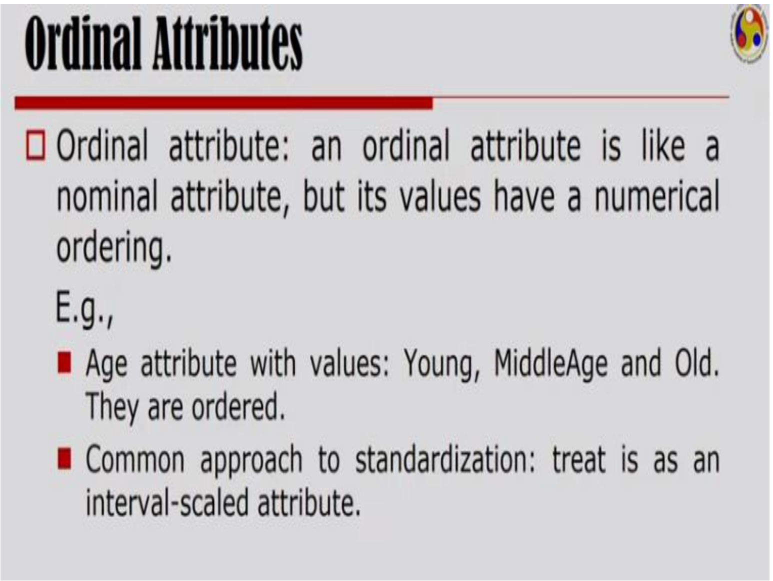


Nominal Attributes: An Example

- □ Nominal attribute *fruit*: has three values,
  - Apple, Orange, and Pear
- □ We create three binary attributes called, Apple, Orange, and Pear in the new data.
- □ If a particular data instance in the original data has Apple as the value for *fruit*,
  - then in the transformed data, we set the value of the attribute Apple to 1, and
  - the values of attributes Orange and Pear to 0

For example let us say we are talking of a nominal attribute called fruit and it has 3 values Apple, Orange and Pear. We create 3 binary attributes Apple, Orange and Pear in the new data and if a particular data instance in the original data has Apple as a value for fruit then in the transform data, we set the value of the attribute Apple to 1 and the values of Orange and Pear to 0 and thus we transform our nominal attribute to a numeric attribute in terms of its Boolean value.

(Refer Slide Time: 44:42)



Ordinal Attributes

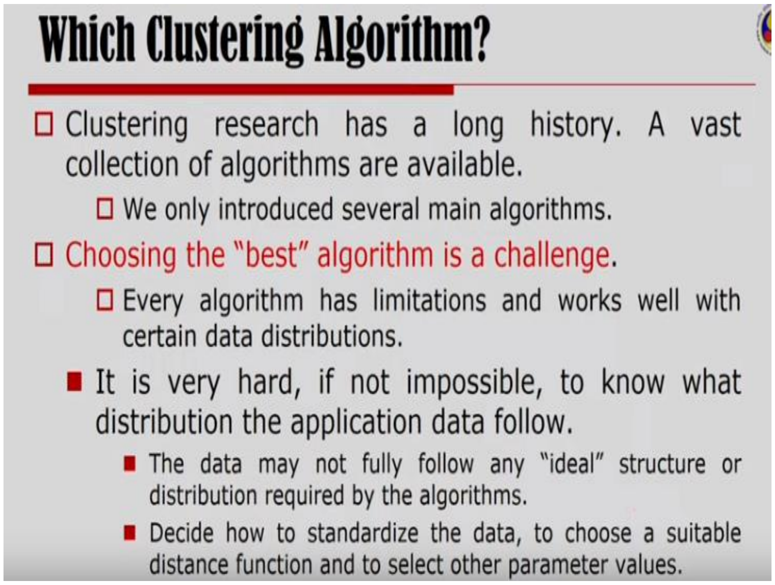
- □ Ordinal attribute: an ordinal attribute is like a nominal attribute, but its values have a numerical ordering.

  E.g.,

  - ■ Age attribute with values: Young, MiddleAge and Old. They are ordered.
  - ■ Common approach to standardization: treat is as an interval-scaled attribute.

Ordinal attributes is like a nominal attribute but its value have a numerical ordering. Like age attribute with values young, middle age and old they are ordered and the common approach to standardization is to treat it as an interval scale attribute.

(Refer Slide Time: 45:02)



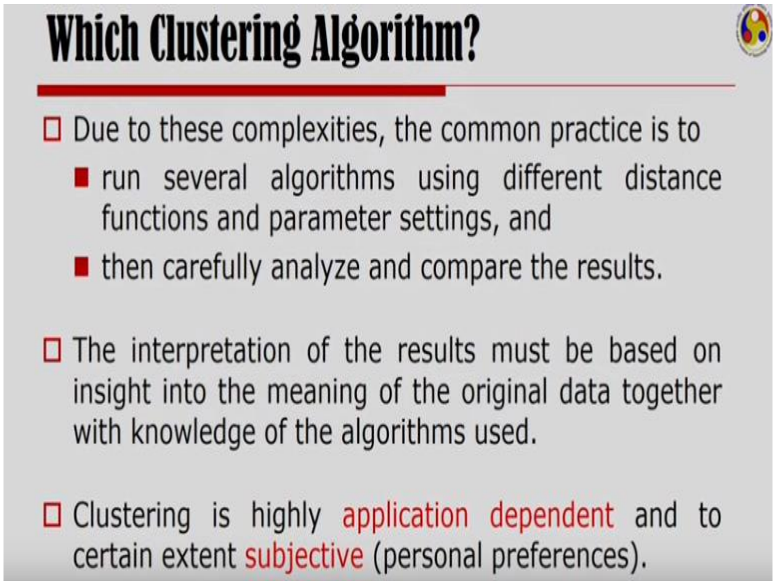
Which Clustering Algorithm?

- □ Clustering research has a long history. A vast collection of algorithms are available.
  - □ We only introduced several main algorithms.
- □ Choosing the "best" algorithm is a challenge.
  - □ Every algorithm has limitations and works well with certain data distributions.
  - ■ It is very hard, if not impossible, to know what distribution the application data follow.
    - ■ The data may not fully follow any "ideal" structure or distribution required by the algorithms.
    - ■ Decide how to standardize the data, to choose a suitable distance function and to select other parameter values.

Now the question is which clustering algorithm should one look for once we have a problem in hand. Now clustering research has a long history and a vast collection of algorithms are available choosing the best algorithm is a challenge. Every algorithm has limitation and works well with certain data distribution. Actually it is hard if not impossible to know what distribution the application data follow and the data may not fully follow any ideal structure or distribution required by the algorithms.

One needs to decide how to standardize the data to choose a suitable distance function and to select other parameter values.

(Refer Slide Time: 46:02)

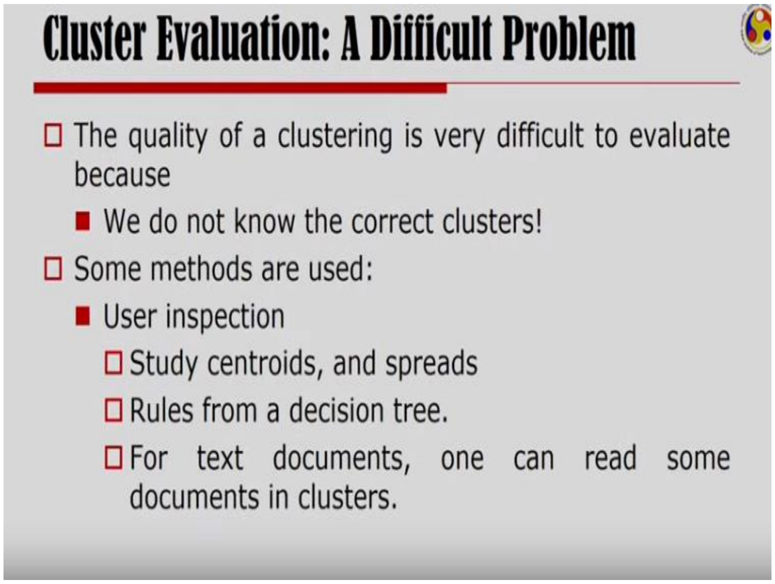


Which Clustering Algorithm?

- □ Due to these complexities, the common practice is to
  - ■ run several algorithms using different distance functions and parameter settings, and
  - ■ then carefully analyze and compare the results.
- □ The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used.
- □ Clustering is highly application dependent and to certain extent subjective (personal preferences).

Due to these complexities the common practice in deciding which clustering algorithm is to run several algorithms using different distance functions and parameter settings and then carefully analyse and compare the results. Now the interpretation of the clustering results must be based on the insight into the meaning of the original data and one needs to realize that clustering is highly application dependent and to certain extent very subjective.

(Refer Slide Time: 46:32)

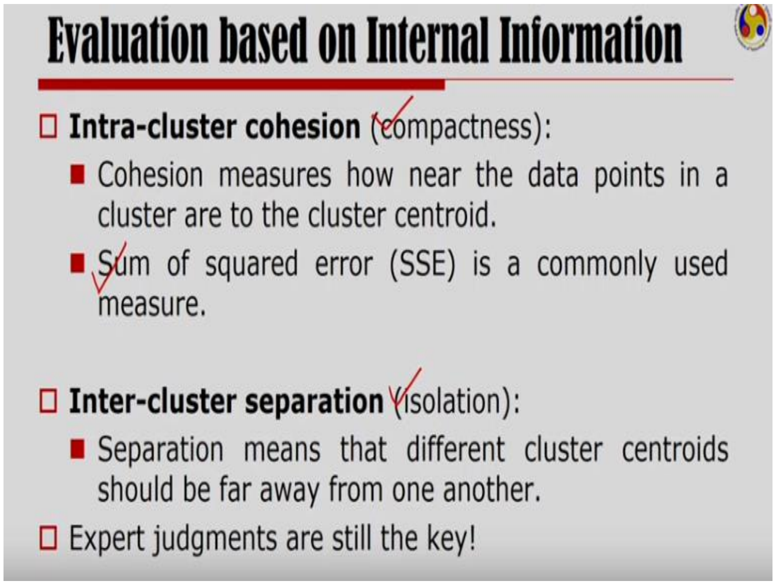


Cluster Evaluation: A Difficult Problem

- □ The quality of a clustering is very difficult to evaluate because
  - ■ We do not know the correct clusters!
- □ Some methods are used:
  - ■ User inspection
    - □ Study centroids, and spreads
    - □ Rules from a decision tree.
    - □ For text documents, one can read some documents in clusters.

And evaluating which clustering results are better than others is a very difficult problem. So the quality of a clustering is very difficult to evaluate because of the very fact that we do not know the correct cluster. Some methods are used and possibly the best ones are still the user inspection,you study the centroid or the spreads or you look at rules from a decision tree or for text documents one can read some documents in the cluster itself.

(Refer Slide Time: 47:16)



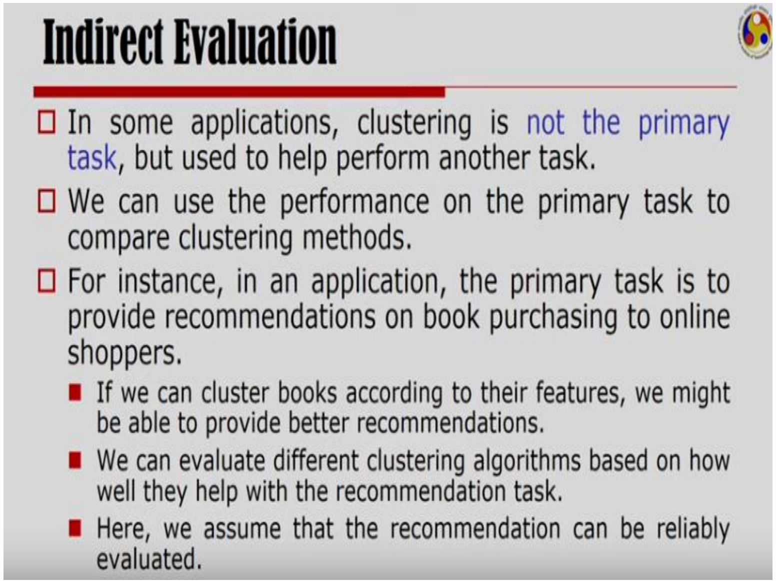
Evaluation based on Internal Information

- □ **Intra-cluster cohesion** (compactness):
  - ■ Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - ■ Sum of squared error (SSE) is a commonly used measure.
- □ **Inter-cluster separation** (isolation):
  - ■ Separation means that different cluster centroids should be far away from one another.
- □ Expert judgments are still the key!

Evaluation based on internal information is what is usually used. So we look at what is called the intra cluster cohesion and that is how compact the clusters are. Cohesion measures how near the data points in a cluster are to the cluster centroid and remember the sum of squared error that we have introduced in the beginning of this lecture today. The sum of squared error is a commonly used measure to find out the intra cluster cohesion and then we can use inter cluster separation which is referred to as isolation.

Now the separation means that different classes centroids should be far away from one another having said that expert judgments are still the key to finding out the quality of cluster.

(Refer Slide Time: 48:24)



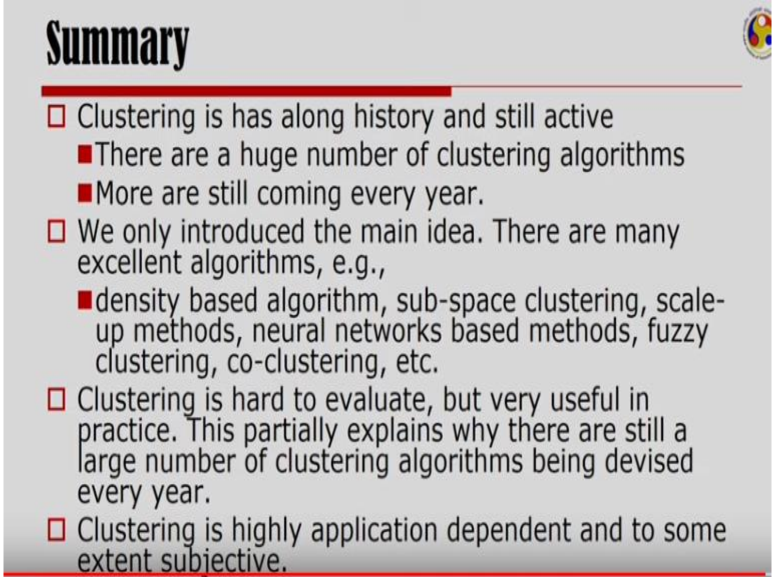
Indirect Evaluation

- In some applications, clustering is not the primary task, but used to help perform another task.
- We can use the performance on the primary task to compare clustering methods.
- For instance, in an application, the primary task is to provide recommendations on book purchasing to online shoppers.
  - If we can cluster books according to their features, we might be able to provide better recommendations.
  - We can evaluate different clustering algorithms based on how well they help with the recommendation task.
  - Here, we assume that the recommendation can be reliably evaluated.

There are also indirect evaluations possible. In some applications clustering may not be the primary task but used to help perform another task, we can use the performance the primary task to compare clustering methods. For instance in an application the primary task may be to provide recommendations on book purchasing to online shoppers. If we can cluster books according to their features we might be able to provide a better recommendation.

So we can evaluate different clustering algorithms based on how well they will help with the recommendation tasks. Here, we assume that the recommendations can be reliably evaluated so this is an indirect evaluation of the type of clusters that we get.

(Refer Slide Time: 49:25)



Summary

- □ Clustering is has along history and still active
  - There are a huge number of clustering algorithms
  - More are still coming every year.
- □ We only introduced the main idea. There are many excellent algorithms, e.g.,
  - density based algorithm, sub-space clustering, scale-up methods, neural networks based methods, fuzzy clustering, co-clustering, etc.
- □ Clustering is hard to evaluate, but very useful in practice. This partially explains why there are still a large number of clustering algorithms being devised every year.
- □ Clustering is highly application dependent and to some extent subjective.

So clustering as I was referring to has a long history and is still active. There are a huge number of clustering algorithms more are still coming every year we have only introduced the very basic idea of clustering today. There are excellent algorithms like the density based algorithm, the subspace clustering algorithm, scale-up methods, neural network based methods, fuzzy clustering and many others.

One realization that we have tried to hammer in is that clustering is hard to evaluate but very useful in practice. So this partially explains why there are still a large number of clustering algorithms being devised every year. Clustering is highly application dependent and to some extent very subjective. So it is very difficult to actually pinpoint which clustering algorithm is better than the other.

Having said that we have today introduced the fundamental concepts of partitional clustering and hierarchical clustering. We have also given an intuitive idea today on spectral clustering what I have not covered is self-organizing maps which are a form of neural network that do unsupervised learning including clustering. Thank you very much.