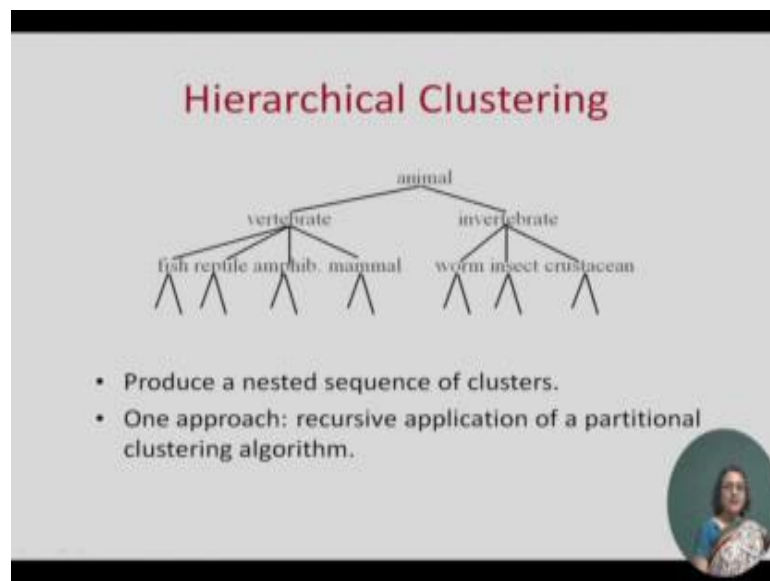


**Introduction to Machine Learning**  
**Prof. Sudeshna Sarkar**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Kharagpur**

**Module - 9**  
**Lecture - 39**  
**Agglomerative Hierarchical Clustering**

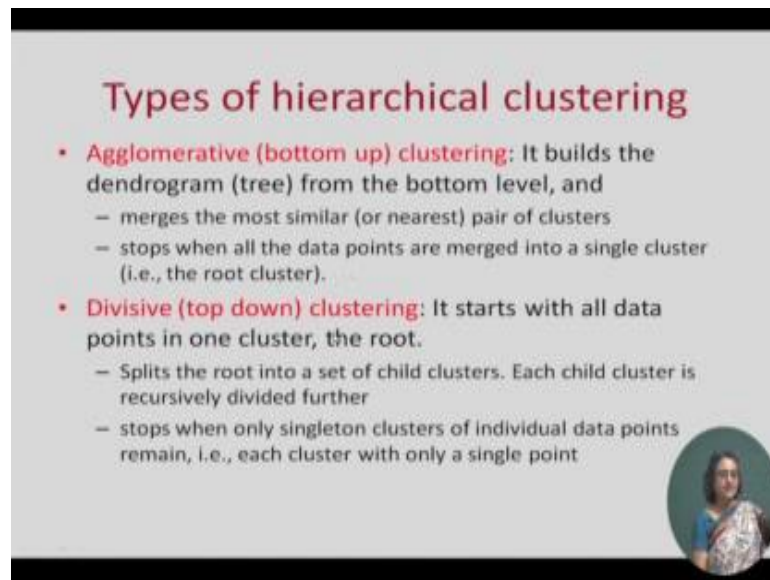
Good morning. Today we come to the last lecture of this week, or the last lecture of the Clustering, in fact the last lecture of this class. We will talk about Hierarchical Clustering; rather we talk about a specific Bottom up Hierarchical Clustering Algorithm. Earlier we have seen k means algorithm which is an example of a top down clustering method. Today we will look at a Bottom up Clustering Method.

(Refer Slide Time: 00:47)



Now, hierarchical clustering is very common. For example, when scientists look at the animal kingdom, they divide them hierarchically; animals are vertebrates and invertebrates, within vertebrate, your fish reptiles' amphibians and mammals and so on. So, this produces a nested sequence of clusters. In order to get a hierarchical cluster we can recursively use a partition clustering algorithm like k means. We discussed the k means algorithm in the last class, and the k means algorithms can be recursively used to do a hierarchical clustering.

(Refer Slide Time: 01:34)



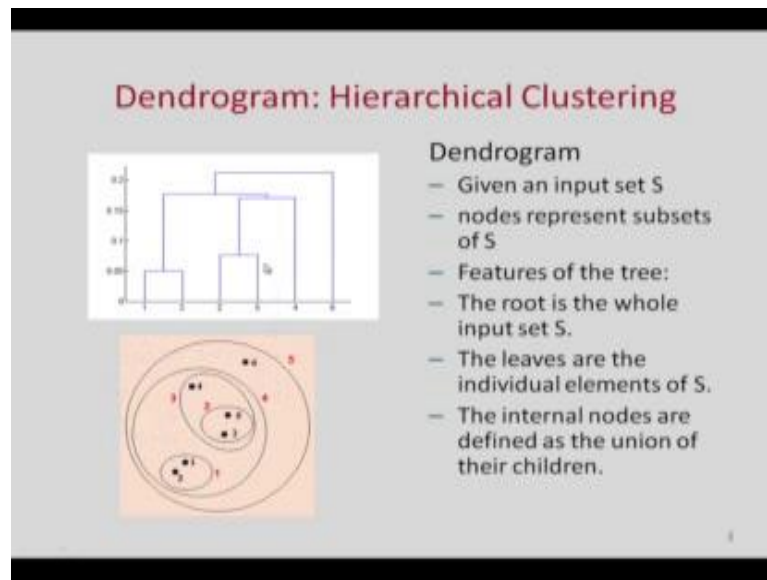
### Types of hierarchical clustering

- **Agglomerative (bottom up) clustering:** It builds the dendrogram (tree) from the bottom level, and
  - 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

But today we will talk about a bottom of clustering method called Agglomerative clustering. In agglomerative clustering we built the clusters from the bottom level that is we assume that every object initially given  $m$  objects we assume that every object is a cluster of its own. Then what we do is, we find the distance between each pair of clusters and those clusters that pair which is closest to each other is merged, so that at the next step we have  $m$  minus 1 clusters. Again we try to find the closest clusters and merged so that we have  $m$  minus 2 clusters. Like this we go on until all the clusters merged.

So, this is the gist agglomerative hierarchical clustering. We could also use Divisive clustering to make hierarchical clusters. We can start with all data in one cluster, take  $k$  equal to 2 divided into 2 clusters, then each clusters we can recursively keep dividing. This gives top down hierarchical clustering; however we will not discuss it further in this class.

(Refer Slide Time: 02:55)

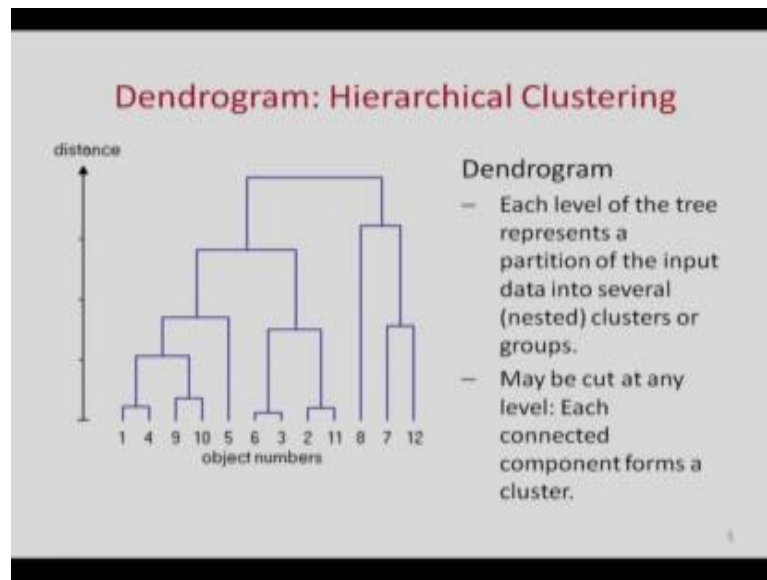


Now the bottom of the hierarchical clustering gives rise to a structure called the Dendrogram. So what is happening is that if you have the different objects, suppose these two objects are the most similar to each other they are merged at step 1. Then step 2, the next most similar objects are merged. This is step 1, this is step 2 and then step 3 the next most similar objects are merged. Let us say this is step 3. Step 4, next most similar objects are merged. In step 5, the next most similar objects are merged. Step 3, the next most similar objects are merged. In the step 7, the next most similar objects are merged.

So, this structure gives rise to a tree which we call Dendrogram. And, as we see in that in the dendrogram the nodes represent the subsets of  $s$ . This a node represent these 3 points. This node represents this 6 point. This node represents these 3 points. So each node represents a subset of the points. And the characteristic of the tree is that, the root is the whole input set. The root of the tree is the entire set, and the leaves are the individual elements. And the internal nodes are defined as the union of their children.

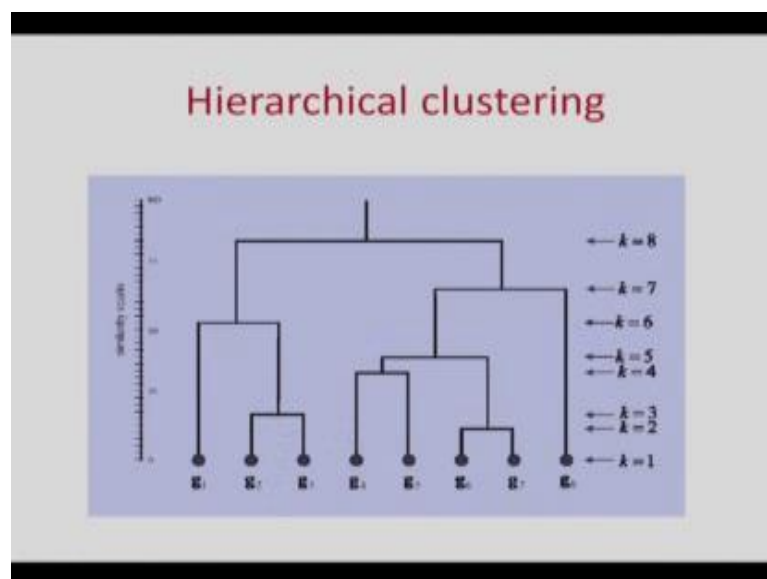
We can also represent this by this sort of diagram. This the one cluster, this is another cluster, this is a nested cluster, and these are nested clusters, so the objects are hierarchically nested which can be represented like this sort of a set diagram or this sort of a dendrogram.

(Refer Slide Time: 05:18)



So, this tree may be cut at different levels. If you cut at this level you will get only one cluster if you cut at the root. If you cut at a particular level, this will give you a certain number of clusters. For example, this cut gives you 1 2 3 4 clusters. This cut gives you 1 2 3 4 5 6 clusters. So cutting the dendrogram at different levels you get different numbers of clusters. You need not decide the number of clusters apriory; you can construct the dendrogram and cut it at the desired level to get number of clusters that you require.

(Refer Slide Time: 06:07)



So, for different values of  $k$  you get different cluster as it is illustrated by this picture.

(Refer Slide Time: 06:19)

### Hierrarchical Agglomerative clustering

- Initially each data point forms a cluster.
- Compute the distance matrix between the clusters.
- Repeat
  - Merge the two closest clusters
  - Update the distance matrix
- Until only a single cluster remains.

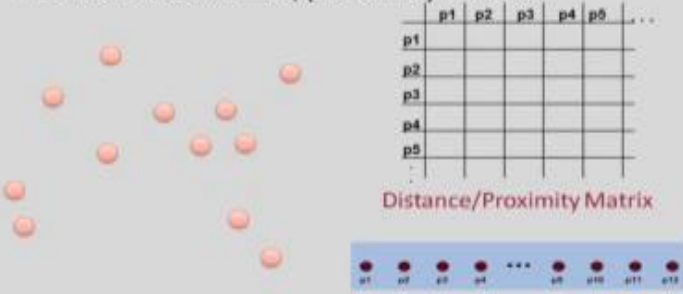
Different definitions of the distance leads to different algorithms.

The algorithm is notionally very simple. The basic algorithm is very simple. Initially each data point forms a cluster. We compute the distance or proximity matrix between clusters. And we repeat the following steps; we merge the two closest clusters, after merging we update the distance matrix. And we continue these two steps until we get only a single cluster. Now the main thing about this algorithm is the distance measure depending on different definitions of the distance will give us different algorithms.

(Refer Slide Time: 07:08)

### Initialization

- Each individual point is taken as a cluster
- Construct distance/proximity matrix



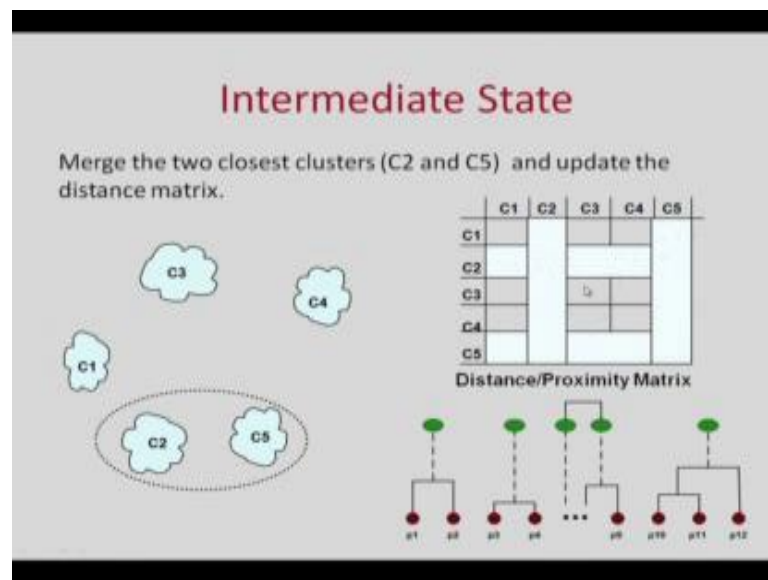
Distance/Proximity Matrix

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
⋮						

● p1 ● p2 ● p3 ● p4 ... ● p5 ● p6 ● p7 ● p8 ● p9 ● p10

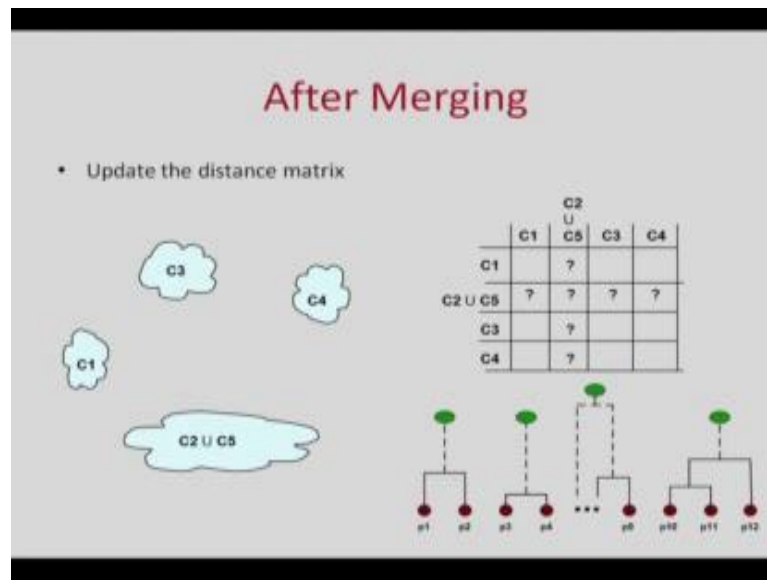
Now, initial step is every point. This is a set of objects; initially every object is a cluster by itself. So every point is a cluster and if there are  $m$  points we have  $m$  by  $n$  proximity matrix between each pair so there will be  $m$  square pairs, between each pair we can compute the distance and fill it up in this matrix.

(Refer Slide Time: 07:38)



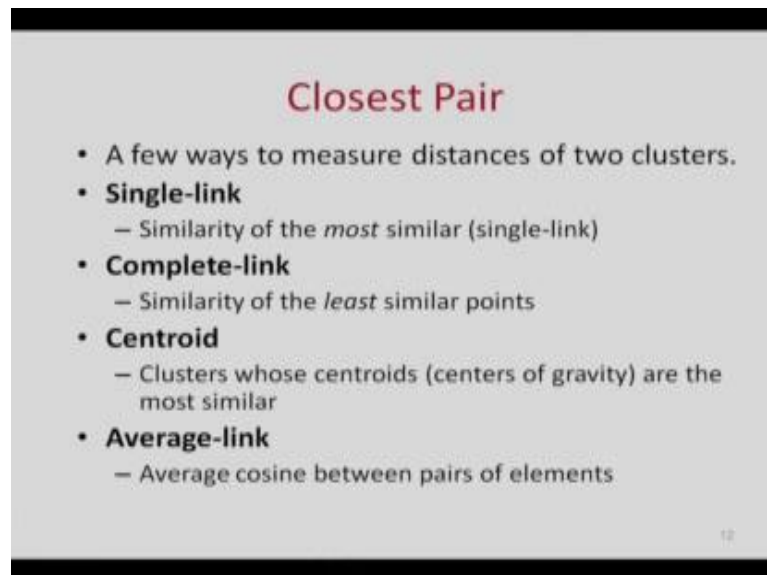
Then at an intermediate state we would have a number of clusters. And the proximity matrix will be defined, suppose there are  $k$  clusters there will be  $k$  square elements in the proximity matrix. Now, suppose in this matrix C2 and C5, so we have this proximity matrix and we compute and find the distance between this five C2 pairs and suppose we find C2 and C5 are the closest. Now what we will do is that we will merge C2 and C5 so that these 2 columns and these 2 rows in the proximity matrix will be collapsed into 1 row 2 column. The next step we will have 4 clusters, so 4 rows and 4 columns in the proximity matrix.

(Refer Slide Time: 08:37)



So after merging we will get these 4 rows and 4 columns and we have to find out how to fill up this column and this row. Now, how we fill up will depend on the distance measured that we will use. And we will talk about certain common measures that we use for this.

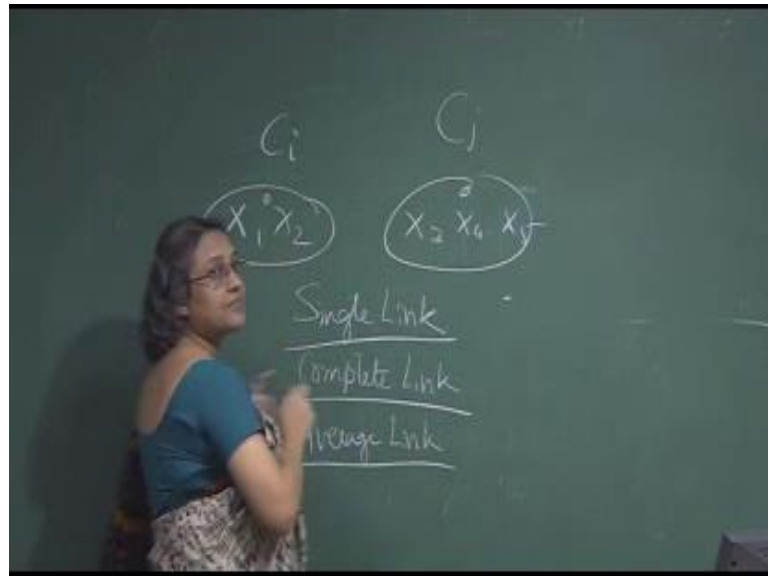
(Refer Slide Time: 09:03)



Now, when we have to find the closest pair as we said that there are different ways to measure distance of 2 clusters; some popular methods are - Single-link, Complete-link,

Average-link, Centroid. In single-link clustering when we look at the similarity between 2 clusters, suppose we have cluster  $C_i$  and cluster  $C_j$ .

(Refer Slide Time: 09:50)



So, we have cluster  $C_i$  comprising of the points  $x_1, x_2$ , and cluster  $C_j$  comprising  $x_3, x_4, x_5$ . Now what is the distance between  $C_i$  and  $C_j$  we find the similarity of  $x_1$  with  $x_3, x_1$  with  $x_4, x_1$  with  $x_5, x_2$  with  $x_3, x_4$ , and  $x_5$ , and we find that pair which is the most similar among a pair we find that pair which so that one of them belongs here one of them belongs here and we take the most similar pair.

The similarity of the most similar pair we take as the similarity between the 2 clusters in single-link clustering. In complete-link which we will talk next, we take the least similar pair as a measure of similarity between 2 clusters. When we take the similarity between two clusters in single-link we take the value of the similarity between the most similar pair. In complete-link we take the similarity as the value of the least similar pair. In average-link, we take the average of their similarity. And in centroid, we take this to be represented its centroid, this to be represented its centroid. And look at the similarity between the 2 centroids.



(Refer Slide Time: 11:51)

### Distance between two clusters

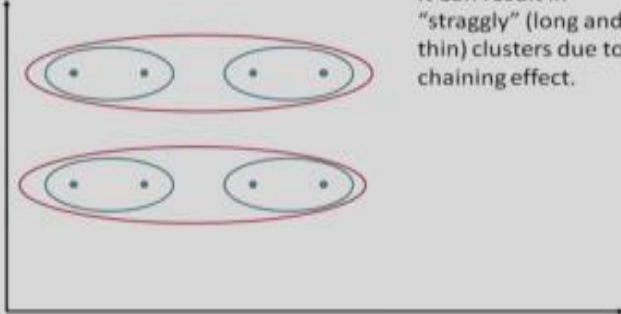
- Single-link distance between clusters  $C_i$  and  $C_j$  is the *minimum distance* between any object in  $C_i$  and any object in  $C_j$

$$sim(C_i, C_j) = \max_{x \in C_i, y \in C_j} sim(x, y)$$

So, in single-link the distance between clusters  $C_i$  and  $C_j$  is the minimum distance between any object in  $C_i$  and any object in  $C_j$ , or maximum similarity of one object in  $C_i$  with one object in  $C_j$ . Which is given by this formula:  $sim(C_i, C_j) = \max_{x \in C_i, y \in C_j} sim(x, y)$ .

(Refer Slide Time: 12:23)

### Single Link Example



It Can result in "straggly" (long and thin) clusters due to chaining effect.

For examples, if these are the points in single-link we will take the most similar points, then the next most similar points and so on, and then combine them. As we see that

single-link clustering can result in straggly, that is long and thin clusters due to chaining effect.

(Refer Slide Time: 12:54)

### Single-link clustering: example

- Determined by one pair of points, i.e., by one link in the proximity graph.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

So, in single-link clustering the similarities determined by one pair of points, that is one link in the proximity graph. If the proximity graph has these values, so what we need to do is that we have to look at the most similar pair and merge them.

(Refer Slide Time: 13:18)

### Complete link method

- The distance between two clusters is the distance of two furthest data points in the two clusters.

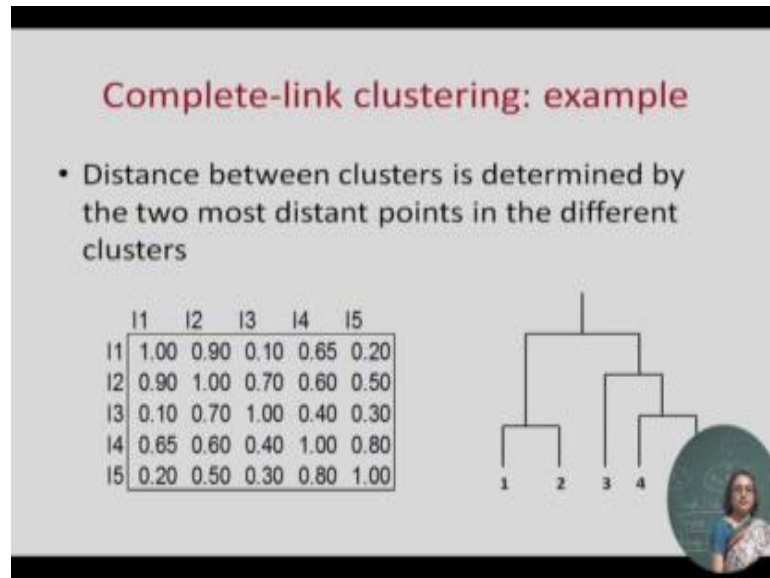
$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes "tighter," spherical clusters that are typically preferable.
- It is sensitive to outliers because they are far away

In a complete-link method, on the other hand the distance between 2 clusters is a distance between the least similar pair. Which is given by the following formula, sim of

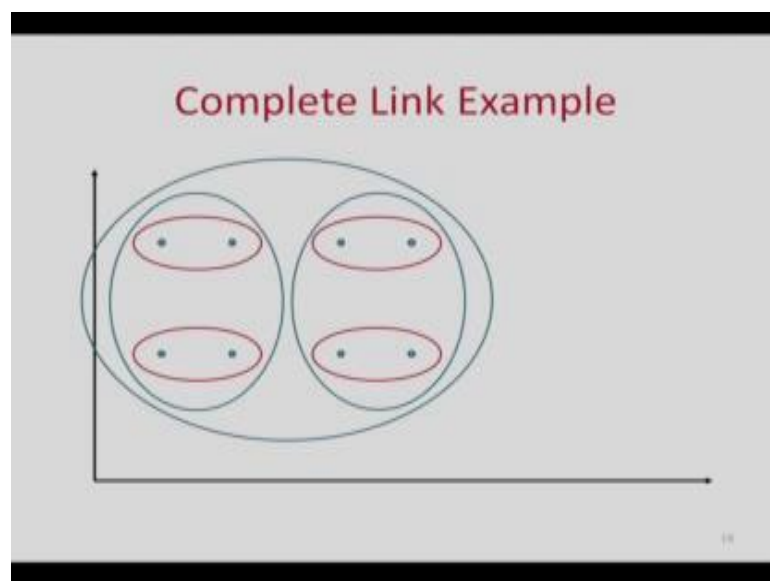
$C_i \cap C_j$  is  $\min x$  included in  $C_i, y$  included in  $C_j \text{ sim } y, x$ . And complete-link makes tighter more spherical clusters; unlike, single-link where you can have long thin clusters. However, complete-link clustering is sensitive to outliers because they are far away. If there is one object in a cluster which is very far away from the other objects it will influence similarity, which is not always desirable.

(Refer Slide Time: 14:12)



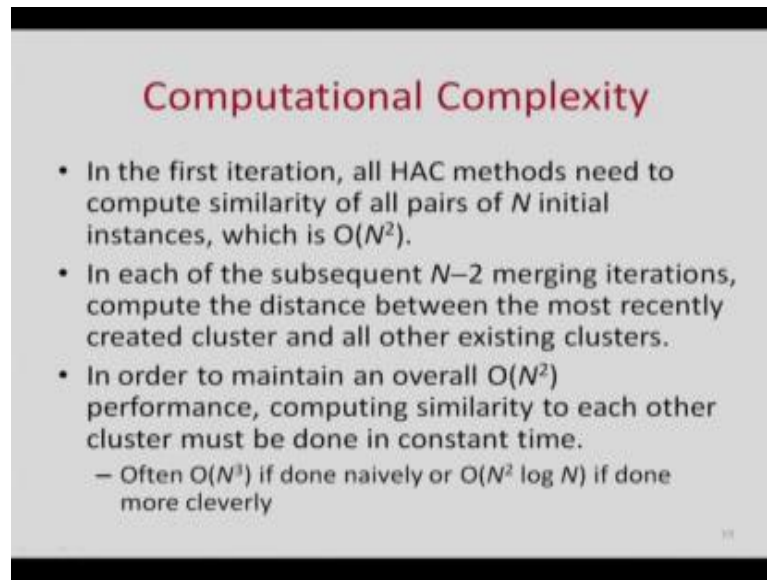
Again you have this matrix and you look at (Refer Time: 14:18) apply complete-link.

(Refer Slide Time: 14:18)



In the same example as we saw before, we can show that when we apply complete-link these are the different steps that we get. We notice that here the clusters are more tight than what we get in the single-link example.

(Refer Slide Time: 14:42)



**Computational Complexity**

- In the first iteration, all HAC methods need to compute similarity of all pairs of  $N$  initial instances, which is  $O(N^2)$ .
- In each of the subsequent  $N-2$  merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall  $O(N^2)$  performance, computing similarity to each other cluster must be done in constant time.
  - Often  $O(N^3)$  if done naively or  $O(N^2 \log N)$  if done more cleverly

Now, let us discuss the computational complexity of the clustering methods. So, hierarchical agglomerative clustering method, in the first step you need to find similarity of the objects, the  $N$  objects you require to find  $N$  squares similarity in the beginning. Then you have  $N$  minus 1 steps, so in each steps you are merging the clusters. And while merging you have to find out the most similar pair and update. Now, how much time you spend will depend on the type of clustering method that you are using; single-link, complete-link, average-link, centroid base, and it also depends on the type of data structure that you went in.

So, if you naively do this usually it is order of  $N$  cube, but in certain cases you can also achieve  $N$  square log  $N$  complexity.

(Refer Slide Time: 15:48)

### Average Link Clustering

- Similarity of two clusters = average similarity between any object in  $C_i$  and any object in  $C_j$

$$sim(c_i, c_j) = \frac{1}{|C_i||C_j|} \sum_{\bar{x} \in C_i} \sum_{\bar{y} \in C_j} sim(\bar{x}, \bar{y})$$

- Compromise between single and complete link. Less susceptible to noise and outliers.
- Two options:
  - Averaged across all ordered pairs in the merged cluster
  - Averaged over all pairs *between* the two original clusters

We will not describe the details. So, we talked about single-link and complete-link. In average-link clustering, actually average-link there people have described in two different ways. You can average across all ordered pairs in the merged cluster or over all pairs between the two original clusters. The second formulation is shown here, but there is other formulation of average-link clustering. So anyway, here we find similarity of cluster  $C_i$  and  $C_j$  is average of similarity of  $x, y$ ; where  $x$  belongs to  $C_i$ ,  $y$  belongs to  $C_j$ . It is a compromise between single and complete-link, it is less acceptable to noise and less acceptable to outliers it also produces spherical clusters.

(Refer Slide Time: 16:43)

### The complexity

- All the algorithms are at least  $O(n^2)$ .  $n$  is the number of data points.
- Single link can be done in  $O(n^2)$ .
- Complete and average links can be done in  $O(n^2 \log n)$ .
- Due the complexity, hard to use for large data sets.

Complexity already we have talked about.

With this we come to the close of today's lectures. There are other clustering methods like density based clustering. For example, the DB scan algorithm and several others, but we will not talk about this in this course. So, this brings us to end of the class on clustering and also the end of the course in machine learning. I hope you all have enjoyed this course.

Thank you very much for putting up through the entire course. All the best.