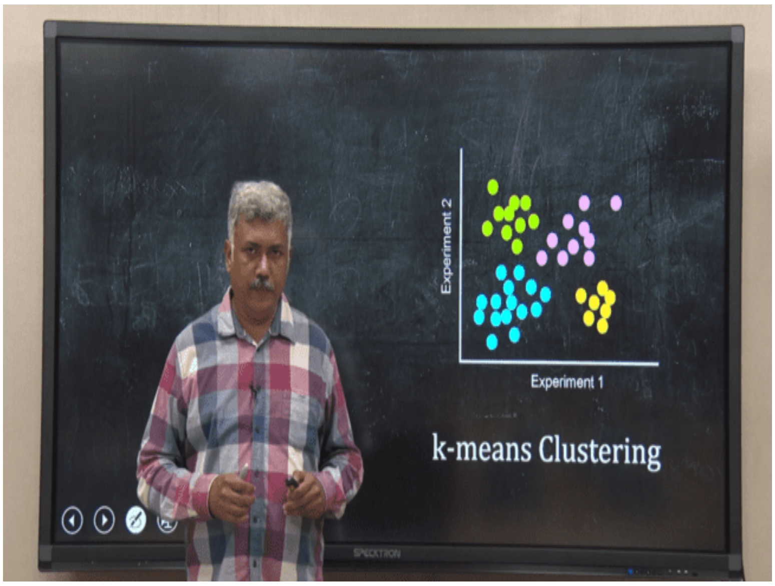
Data Analysis for Biologists
Professor Biplab Bose
Department of Biosciences and Bioengineering
Indian Institute of Technology Guwahati
Lecture 38
K-Means Clustering

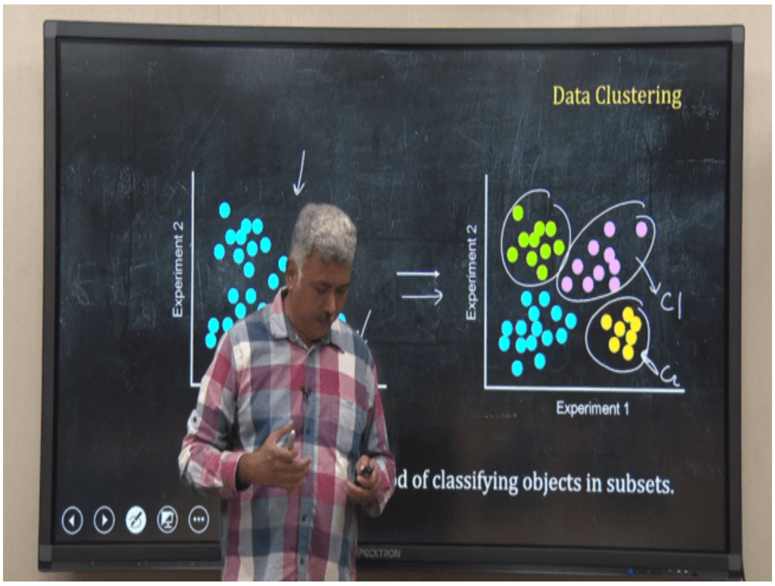
(Refer Slide Time: 00:29)



k-means Clustering

Hello everyone. In this lecture, we will discuss about k-means clustering.

(Refer Slide Time: 00:39)



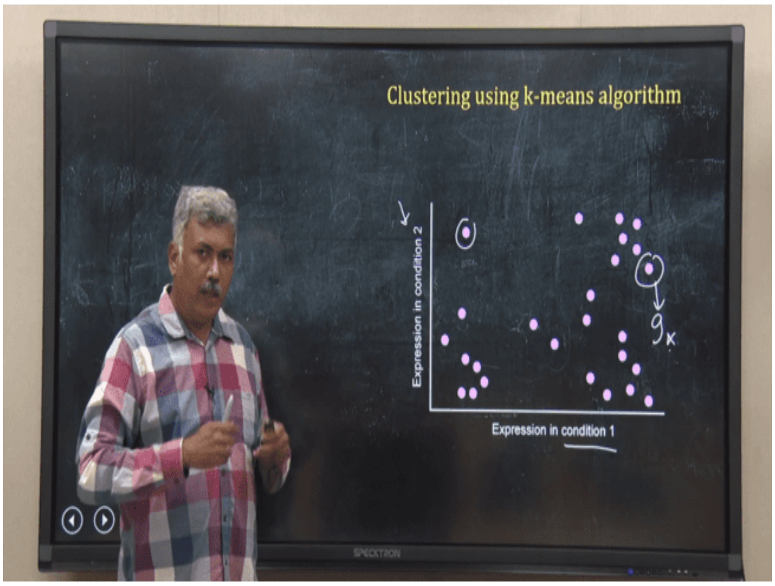
Data Clustering

d of classifying objects in subsets.

Clustering is a method by which we take a set of objects or set of data points and then we divide them into different subsets or groups. As I have seen in this diagram, I have largely heterogeneous data coming from two experiments, each point represents one data point or object. This is quite heterogeneous and I want to separate these objects or data points into some subsets like this green subset, this pink subset, this yellow one, in which the each of these subsets are quite homogeneous. So, each of these subsets are my cluster. So, this one may be cluster 1 and this one may be cluster 2 and so, on.

So, what I have done in this case using some algorithm I took a heterogeneous data and I have broken it down into four clusters where in each cluster data is quite homogeneous or close to each other. There are many algorithms for clustering. In this lecture, I will discuss k-means clustering, which is very easy to understand implement and quite widely used in different data types.

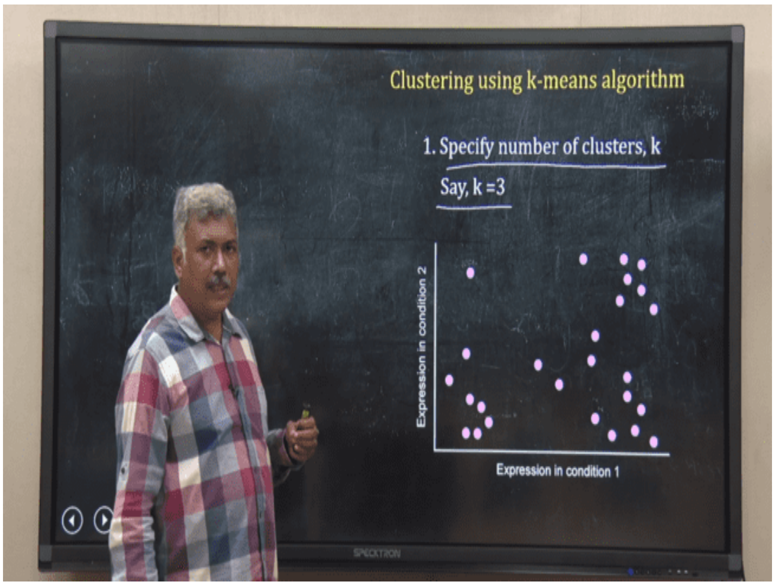
(Refer Slide Time: 01:50)


Clustering using k-means algorithm

So, let us take an example of gene expression experiment. So, suppose I have two experimental condition. Condition 1 and condition 2 maybe two different drug doses, you have treated cells with the drug and then you may have measured the gene expression of hundreds or thousands of them and I have shown only around 25 or 26 of such genes and each of these point is one gene. Suppose gene x.

So, I want to create clusters out of this data, I have a heterogeneous gene expression population, you can see gx has quite different expression from this one. So, I want to find clusters of genes in this data who are behaving in the same fashion whose expressions are same that means in one cluster if three genes are in one cluster, their expression pattern in these two experimental conditions are similar, may not be same. So, how do I perform k-mean algorithm on this? Let us do it step by step visually.

(Refer Slide Time: 02:53)



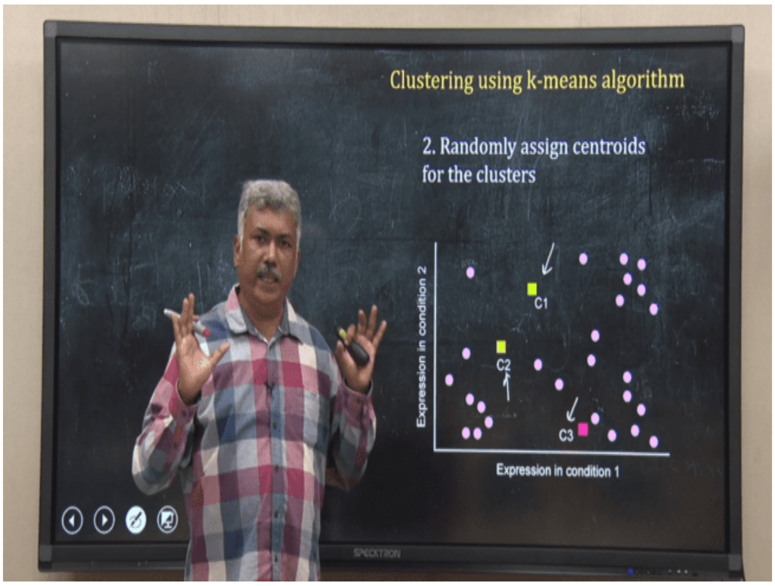
Clustering using k-means algorithm

1. Specify number of clusters, k

Say, k =3

The first step in k-means algorithm is that you have to specify number of clusters. There is a number of clusters that you want. Remember, k-means clustering is an unsupervised statistical learning method, I do not have labelled data, I do not know the classes. So, I do not know number of clusters present in my data set. So, how should I specify k? Okay, I will come to that. For the time being consider you know the number of clusters or you know how many clusters you want in this analysis. So, for example, let me assume I consider k equal to 3. So, k is the number of clusters and I have assumed in this data, I have 3 clusters.

(Refer Slide Time: 03:43)

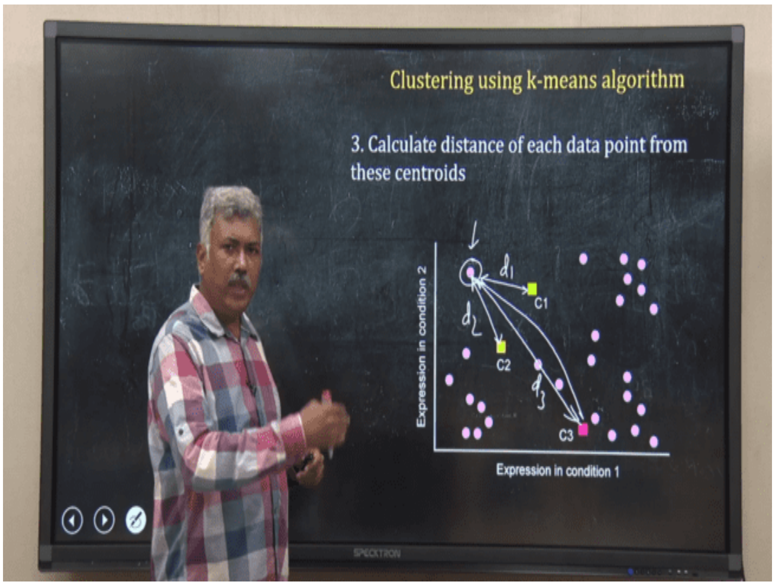


Clustering using k-means algorithm

2. Randomly assign centroids for the clusters

So, the next step is you randomly assign the centre or we call centroid for these clusters. So, I have three cluster k equal to 3. So, that is why I have three centroids C1, C2 and C3 and I have randomly positioned them in this 2d space because I do not know where the clusters are and so, I do not know where their centroid or central position is. So, I have just randomly seeded the centre, three centres.

(Refer Slide Time: 04:11)



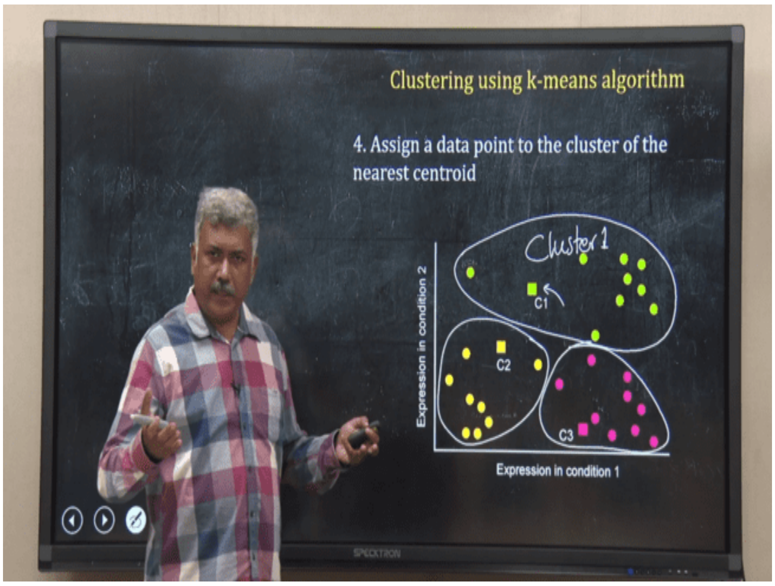
Clustering using k-means algorithm

3. Calculate distance of each data point from these centroids

Now, what you do is a third step. For each data point for example this one, for each data point you calculate the distance from these centroids. So, I can calculate the distance sorry this should be straight line . So, I have distance 1, distance 3 and distance 2. So, do that for all the data points, take a data point and calculate the distance of that data point from each of the centroids, three centroids.

Now, if you remember in one lecture, we have discussed defined distance measures right you can use different distance measure as per your requirement but mostly in k-means algorithm people use Euclidean distance. So, what I am showing d1, d2 and d3 maybe the Euclidean distance of this particular data point from this three centroid C1, C2 and C3. Now you have done the same thing for all the 25 or 26 data points that we have, genes that we have in my data.

(Refer Slide Time: 05:16)



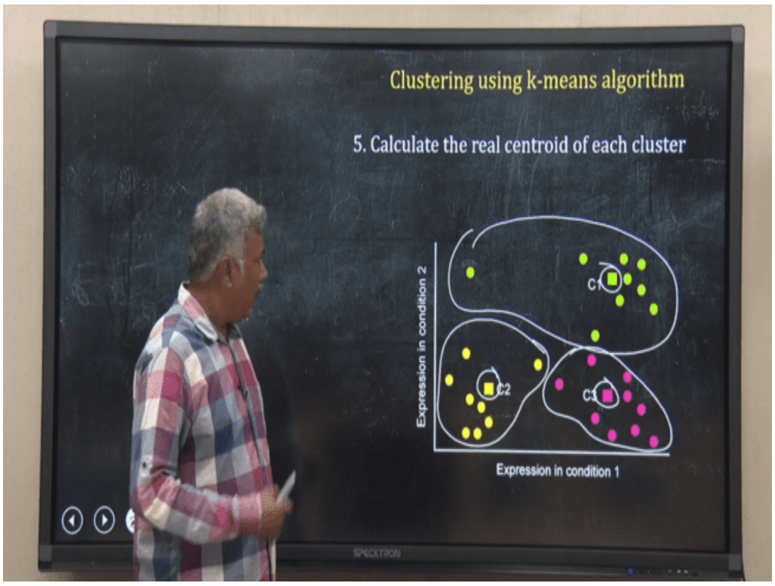
Clustering using k-means algorithm

4. Assign a data point to the cluster of the nearest centroid

Now what we have to do? The fourth step, I have to assign every data point to the cluster of its nearest centroid. That means, what do I mean by that? Let us take the same example this data point. Now, I know the distance from C1, I know the distance from C2 and I know the distance from C3. You can easily see C1 is the closest to my data point. So, what I will do? I will take this data point and I will assign that to the cluster of C1 that is cluster 1. I will do the same thing for all the data points. So, in this way my figure will change that means now all these green points, green data points belong to cluster 1 and the centroid was C1.

Similarly, this yellow data points belong to cluster 2 because they are closest to C2, the centroid of cluster 2 and this pink one belongs to cluster 3. So, I have got three non-overlapping clusters. Now, remember this C1 and C2 and C3, the centroids are arbitrarily placed randomly seeded right they are not correct centroid.

(Refer Slide Time: 06:41)



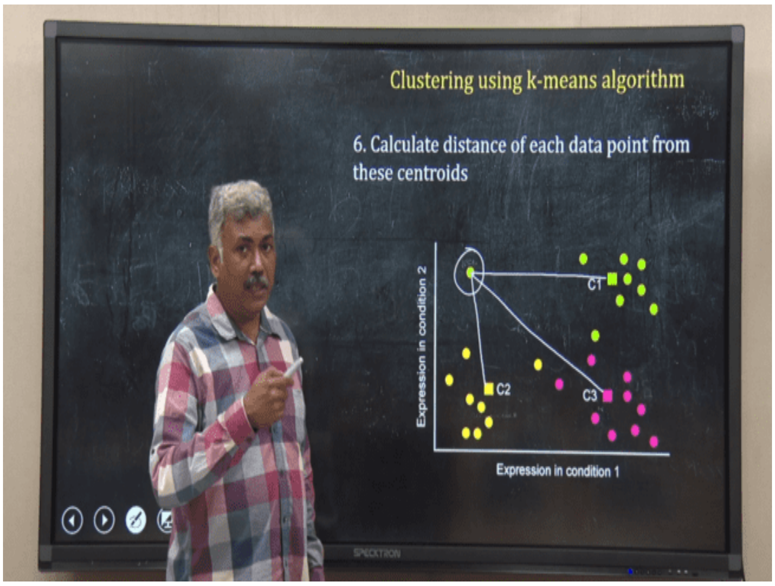
Clustering using k-means algorithm

5. Calculate the real centroid of each cluster

So, what I will do now, I will calculate the real centroid for each clusters. So, what is my first cluster this green data points on my first cluster. So, I will take the value of gene expression in a condition 1 and 2 for each of these green data points and calculate the centroid. That is the mean position.

So, suppose that is somewhere here, whereas for this red one, this cluster 3 suppose the mean maybe it means the centroid maybe the real centroid maybe somewhere here and for this yellow one the real centroid maybe somewhere here. So, now, I have calculated the real centroid of each of these three clusters. So, I record that means I moved this C1 to here, I moved the C3 to there and this C2 to this new calculated correct centroids. So, that is what I have done to visually represent it here. So, this is my cluster 1 green data points and the centroid the real centroid is now here and the real centroid for the second cluster is here and the third one here.

(Refer Slide Time: 07:59)

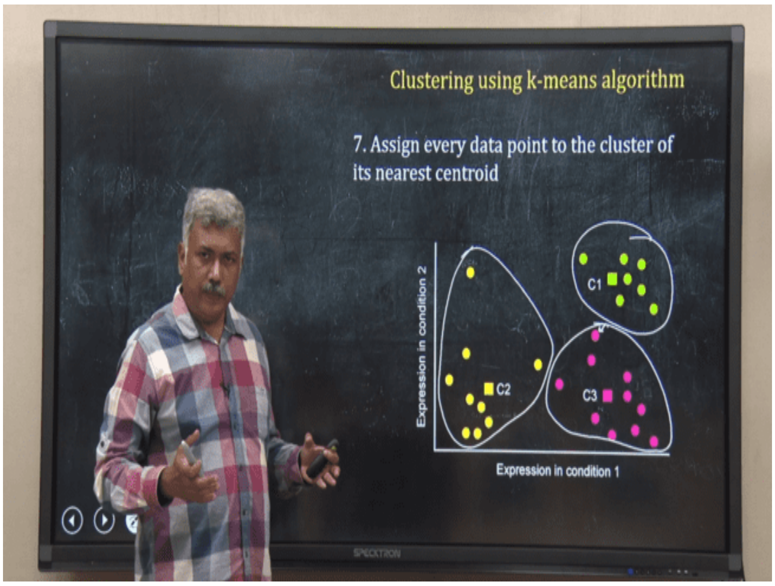


Clustering using k-means algorithm

6. Calculate distance of each data point from these centroids

Now, I come to the sixth step. Now, again for each data point, I will calculate the distance of each data point from each of these three centroids. Now, remember in this case, the centroids are the real centroid right they are not the randomly seeded one. So, take my same old example, this data point. I calculate the distance from C1 from C2 from C3. So, in this way, you calculate the distance of each of these data points from this newly assigned cluster centroid C1, C2 and C3 which are the real one. So, I have done that.

(Refer Slide Time: 08:36)



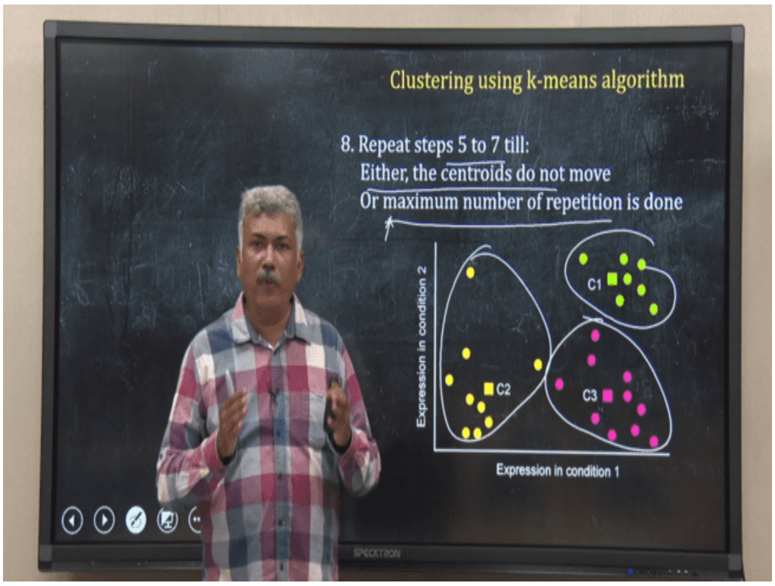
Clustering using k-means algorithm

7. Assign every data point to the cluster of its nearest centroid

Now, based on that distance, I will reassign the cluster to each of these data points. What do I mean by that? So, this is my example data point, I have calculated the distance from C1, C2, C3. For the time being if you remember and if you look into the colour, this data point belongs to cluster 1, the green cluster. But if you now look at the distance you will find this one is shorter than rest of the other distances. That means this data point this example data point that I have taken is closest to C2 not C1 or C3. That means, I have to assign these data points to cluster 2 not cluster 1. And that is what I have done and I have done that for all other data set data points.

So, now, my clusters have changed. So, that example data point has now become part of cluster 2, the green cluster has become smaller. This one was earlier in cluster 1, but now it is in cluster 3. So, the clusters have been reassigned to the data points. Now, what I can do? I can repeat this whole process.

(Refer Slide Time: 09:54)



Clustering using k-means algorithm

8. Repeat steps 5 to 7 till:
Either, the centroids do not move
Or maximum number of repetition is done

That is I can repeat step five to seven that means again you calculate the distance of each data point from the centroids. Before that you calculate the centroid of each of these new clusters and you calculate the distance of each data point from those real centroids and then based on the distance from the centroid, you assign each of these data points to a particular cluster and you keep on repeating these three steps, step five to step seven.

Now, how long we should repeat it? You cannot repeat these eternally. So, your thumb rule would be once you see that actually the centroids are not moving, every time you are reassigning the clusters and you are calculating the new centroid, the value of the new centroid position is not changing the position on the centroid is not changing in this space, that mean the clusters are not changing.

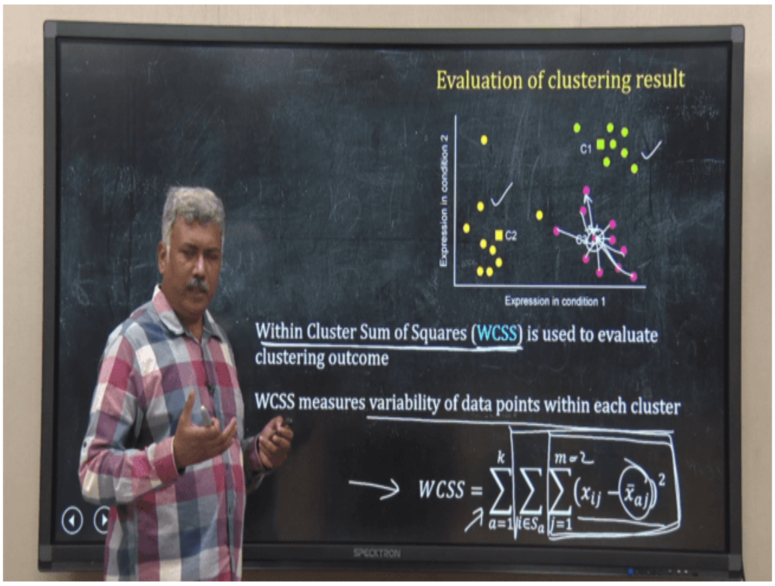
So, that means you have reached the end. But in some cases, it may not happen. You keep on doing it repeatedly 10 times 20 times 30 times still the clusters are reassigning and the centroids are moving around. So, you have to stop somewhere. So, that is why you may have a maximum number of repetition in your algorithm. So, if you have hit that maximum iteration, you will stop that and report the last result as your clustering result.

So, either you use the criteria the centroid do not move or use a maximum iteration criteria. Whatever you use your algorithm will eventually give you output by which I will see three clusters in this case suppose. But how do I know this clustering output is correct? Because remember this is not a classification problem.

I do not know beforehand to which class, cluster does each of these data point belongs to. This is completely unsupervised and if you do it yourself, remember at every time you are doing this using a k-means algorithm on your same data, your initial seeding is a random seeding. You are randomly putting the centroids, three centroids in this example.

That means every time the centroids will be different and there is a high chance that every time you run the algorithm, you may get different clustering results. So, how will you know that your clustering algorithm has given you the correct result? So, you have to evaluate your results.

(Refer Slide Time: 12:23)



Evaluation of clustering result

Within Cluster Sum of Squares (WCSS) is used to evaluate clustering outcome

WCSS measures variability of data points within each cluster

$$\rightarrow WCSS = \sum_{a=1}^{k}\sum_{i\in S_a}\sum_{j=1}^{m=2}(x_{ij}-\bar{x}_{aj})^2$$

$$WCSS = \sum_{a=1}^{k}\sum_{i\in S_a}\sum_{j=1}^{m}\left(x_{ij}-\overline{x_{aj}}\right)^2$$

So, that evaluation can be done in different way, I will explain one of them. The most popular method of getting the, calculating the quality of clustering result or evaluating the clustering result, it will look into the cohesiveness of these clusters that means how close these data points in this cluster 1. They should not be loose right they should not be far away from each other because we want them to be homogeneous.

We started with a heterogeneous data and I want homogeneous clusters. So similarly, for cluster 3 data should not be dispersed too much. So, how should I measure? What should be my statistics to measure that homogeneity? That homogeneity is measured by within clusters sum of squares or WCSS, within cluster sum of squares. It is a measure that gives the variability of data points within each of these clusters.

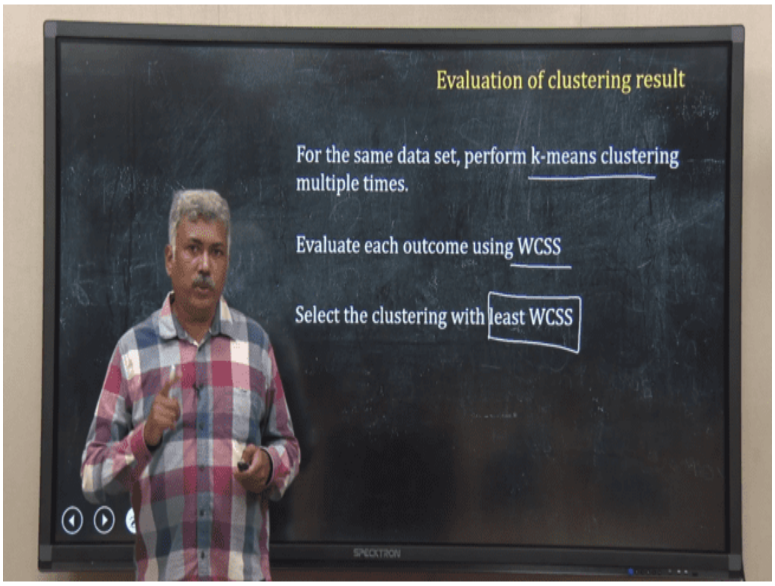
So, what do we do in that? Take C3 example. So, this is my cluster centre, the centroid. So, calculate the Euclidean distance of this data from the centroid and take a square of that. Similarly, take the other data, calculate the Euclidean distance for each of them and square those term and sum those square of Euclidean distances.

And now, you do that for all these three clusters, this one and this one and sum those all Euclidean distances' square that will be called within clusters sum of squares and I have written that in mathematical form notation here. So, this last part, this is the square of Euclidean distance, I have two-dimensional data here.

So, j will vary from 1 to m equal to 2. If you have three-dimensional data means three experiments or you have 10 dimensional data 10 experiments, aim should be j should be 1 to 10, 1 to 5 something like that. So, this whole thing is giving me the square of the Euclidean distance from the centroid, a particular cluster centroid and then you summing them for all the data in that cluster. And then you are summing all those for each of these three clusters. That is the last summation outermost summation.

So, this summation of summations give you something called WCSS. So, it is a measure of variability or measure of in other words is a measure of homogeneity within each of these clusters. So, what do I want? My good clustering outcome should have these WCSS lowest.

(Refer Slide Time: 15:14)



Evaluation of clustering result

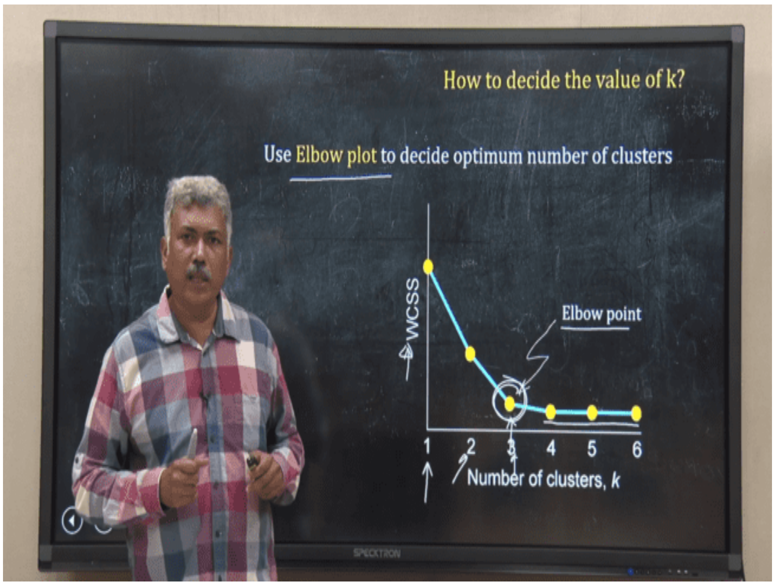
For the same data set, perform k-means clustering multiple times.

Evaluate each outcome using WCSS

Select the clustering with least WCSS

So, for a particular data set, what you are supposed to do? You are supposed to run the same k-means clustering algorithm repeatedly on the same data set maybe 10 times 15 times. For each case you may have a different output, different result. And for each case you will calculate WCSS within clusters sum of square and take the result of that particular analysis which gives you least value for WCSS. That is how we try to evaluate my outcome of k-means algorithm clustering and take the best outcome. Now, there is another issue which we started with.

(Refer Slide Time: 16:00)



How to decide the value of k?

Use Elbow plot to decide optimum number of clusters

If you remember I said the algorithm starts with specifying k number of clusters. But the problem is it is an unsupervised technique, its unsupervised learning, I do not know how many clusters are there in my data. If you consider the ends the extreme would be all data belongs to one cluster that means k equal to 1 whereas, suppose you have 100 genes or in this case suppose we had 26 genes, I can assume there are 26 clusters. So, these are two extreme 1 and n, n is the number of data points but the real number should for k must be somewhere in between.

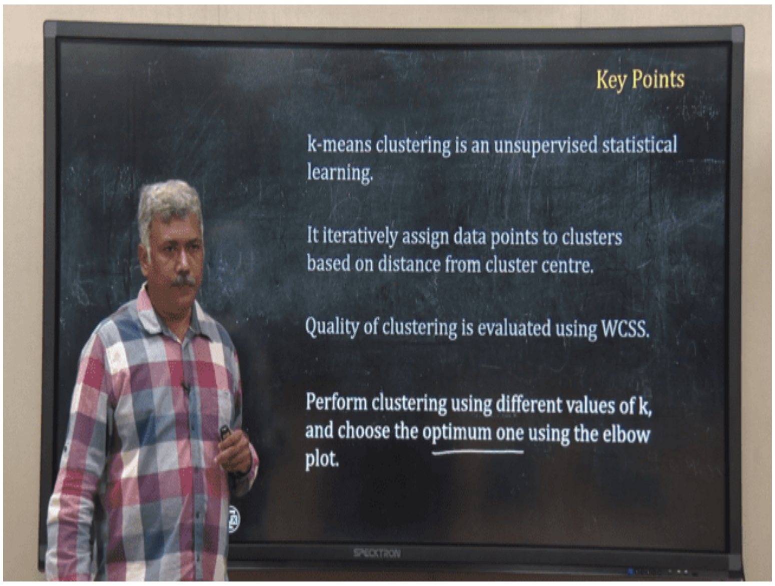
So, how should I decide that? There are many tests and trials we will do to decide the value of k. I will explain one which is very commonly used and we will use it in our further discussions. So, this is called the use of elbow plot. What is elbow plot? What you do now is that for the same data set, you initially decide that k equal to suppose 1.

You do k-means algorithm, use k-means algorithm get the WCSS within clusters sum of squared value for the result, then you repeat the whole thing but now you consider k number of cluster equal to 2. Again you will get a clustering data output and you will also calculate within cluster sum of square. In this way, you keep on increasing k to some reasonable value for example, in this case, I have shown up to 6 and you plot the data.

If you plot the data in most of the cases although not every case is you will see that after a certain value for example, in this example, at k equal to 3 the WCSS is flattening, if I increase k the WCSS or within cluster sum of squares is actually not reducing farther. So, this is that elbow point. Elbow point where I have a deflection right, it is changing. So, this inflection point is called elbow point. And this is my optimum value for k.

As I said this elbow point method is very commonly used but also in some cases, you may not find it very useful because in many times this elbow point is not so clear. But in most cases, you will find this method is very useful. And by using this method, you can actually decide what will be your optimum number of clusters. That is all for this lecture on k-means clustering. Let me jot down what we have learned the main point of this lecture.

(Refer Slide Time: 18:46)



Key Points

k-means clustering is an unsupervised statistical learning.

It iteratively assign data points to clusters based on distance from cluster centre.

Quality of clustering is evaluated using WCSS.

Perform clustering using different values of k, and choose the optimum one using the elbow plot.

So, we have learned k-mean clustering which is a unsupervised method of statistical learning. And what we do here, we iteratively assign data points to cluster based upon the distance from the cluster centre or centroids. So, it is an iterative process. And then what we have learned is that the quality of clustering is evaluated using within cluster sum of square and you can use the elbow plot with the vertical axis having WCSS and the horizontal axis having the number of clusters to decide the optimum number of clusters that you should use for your clustering problem. That is all for this video. See you in the next one. Thank you for being with me today.