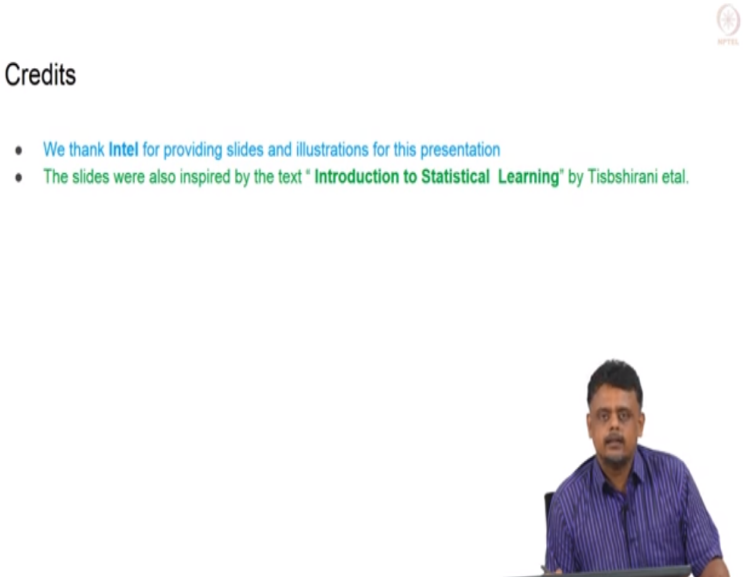
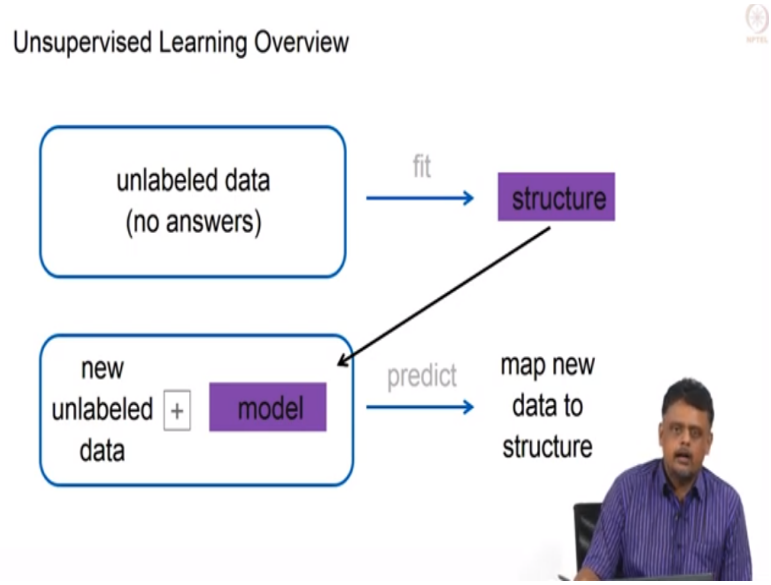
Machine Learning for Engineering and Science Application
Professor Ganapathy Krishnamurthi
Department of Engineering Design
Indian Institute of Technology Madras
Unsupervised Learning & (Kmeans)

(Refer Slide Time: 00:15)



Hello and welcome back in this video on the next couple of videos we will look at some topics in unsupervised learning specifically at the most popular K-means algorithm and as well as the hierarchical agglomerative clustering algorithm, so some of this slide are provided by Intel software based on their curriculum offering and many of the content much of the content is also inspired by the text introduction to statistical learning as well as the elements of statistical learning text.

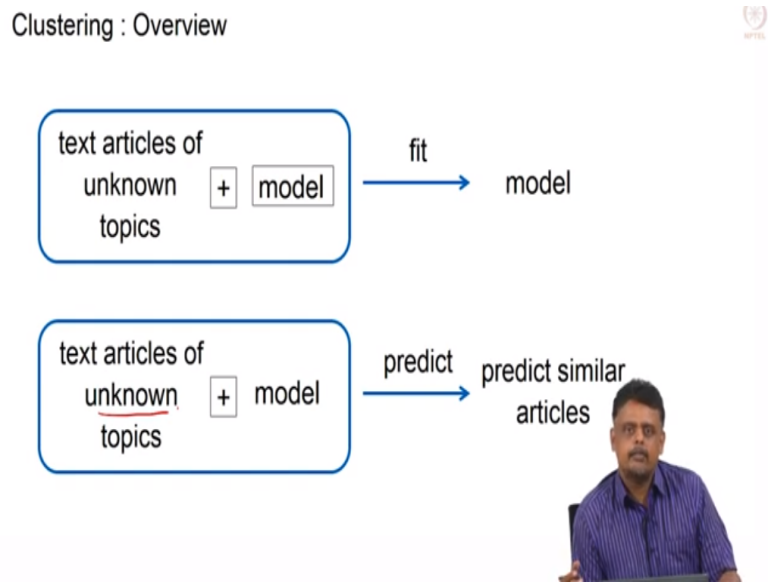
(Refer Slide Time: 00:48)



So brief overview of unsupervised learning's order so far we have looked at a variety of technique which can be classified as supervised learning algorithm in the sense that you are provided with a bunch of inputs primarily features as we going to call them and we have a corresponding label to each one of those inputs so it is either a category label or something's it is just a real number which means more like a regression problem in all those cases that we have seen the for a given X which is just a raw input or some features which have been extracted from the input there is a corresponding Y which is either label or some real valued number in the case of regression however in the case of unsupervised learning we are just given plain unlabeled data now this is the case in most of the time in the real world because label data is hard to come by.

So we will be given a bunch of unlabeled data and the idea use to develop a model from the data itself based on some metric that we define with some symmetric we develop a model which automatically separates the data into different classes or different bins if you can like to call them and that is it of it determines the structure they are the unsupervised learning algorithm determine the structure from the model leading to the model and whenever there is new unlabeled data we run it to the model to see in which one of those bins or clusters that it falls or se extract an underlying structure based on the model, so this is the general flow of a unsupervised learning algorithm.

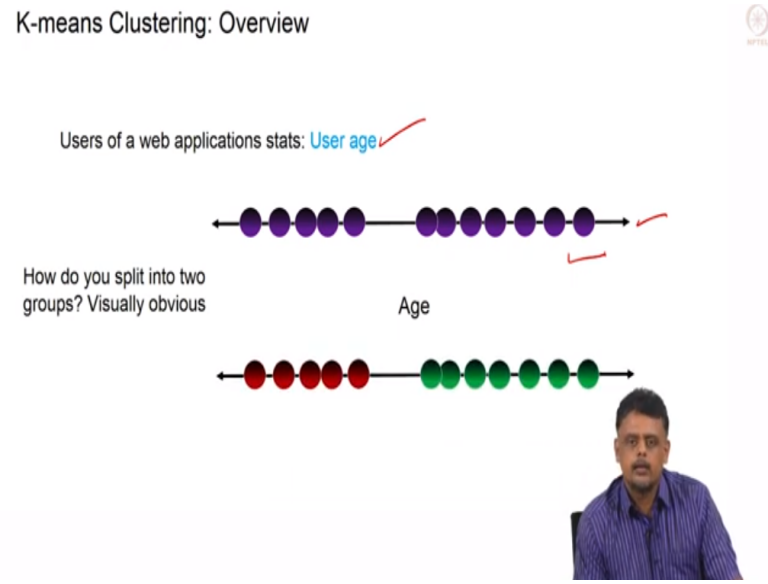
(Refer Slide Time: 02:26)



So for instance topic modeling so we are given a bunch of text article may be recent newspaper articles consisting of unknown topics, so based on the text in the article we come up with a model which tells you which separates the article into several topics so owe do not what the topics are primarily but because they are themselves are extracted from the given text however when a new articles comes in which is basically we have a new are from with an unknown topic we run it to the model and then it bins it with similar articles that is it predicts what to do what class it would deplete typically belong to.

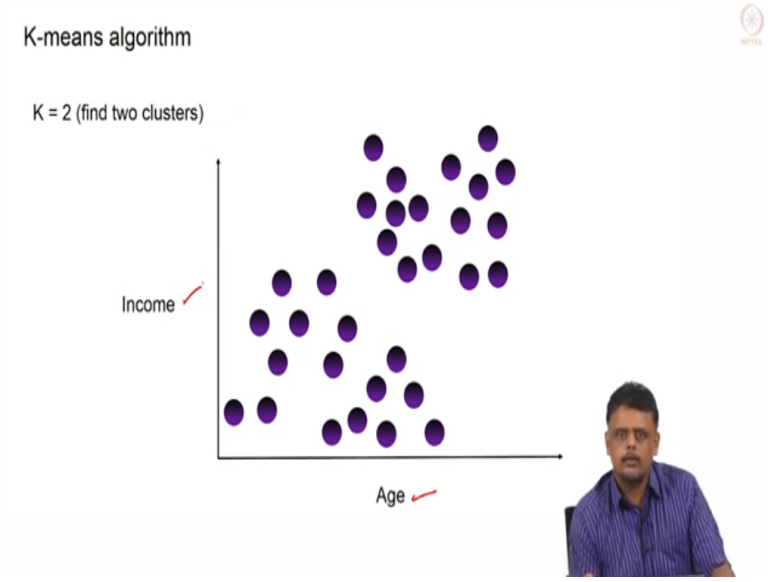
So this kind of structure what is commonly in commonly called as unsupervised learning wherein we really do not know what the classes are or what he underlying structure of the data is except that we are given the raw data and then we try to developed a model based by finding some underlying structure to the data, so the structure is again dependent on some metric that we define as we will see for the K-means algorithm as well as the clustering algorithm other clustering algorithm.

(Refer Slide Time: 03:38)



So for gaming's trusting is a brief overview so here the statistics of so this is basically we gather statistics let us say there is a website web application and we gather statistics of the users that use the web publication so one such statistic is the age of the user and let us say we have like several user in this case about five or thirteen or fourteen user and we just look at me plot that age along an axis that right there and it is very obvious from the plot itself that there are two visibly very visually obvious groups in the users age based on the users age so the we can see here the green is one group and the one from on another group.

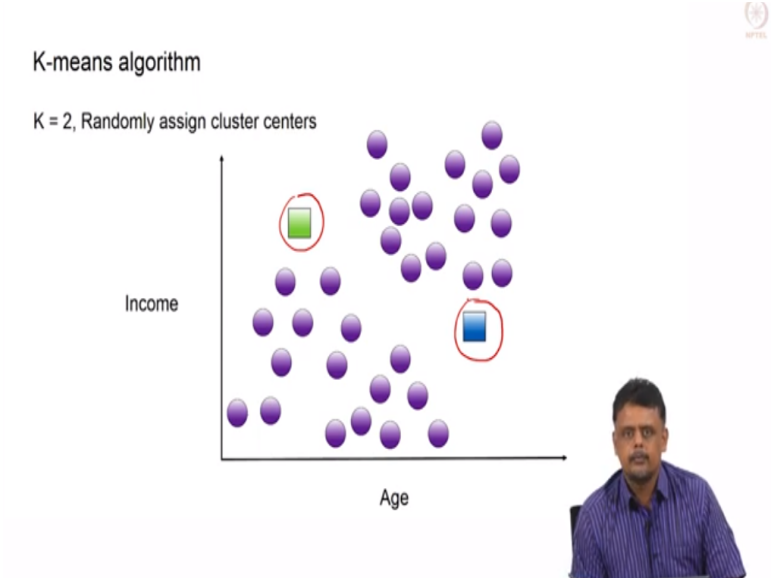
(Refer Slide Time: 04:26)



So let us consider a slightly more complicated in the case 2D, now typically in an unsupervised setting or in this case K-means algorithm setting you will have a lot more categories I this case income and age are feet just so you will have a lot more in input dimensionality would be typically higher but for the case of illustration and understanding the algorithm we will just consider two features which are the income and the age and the income right, so right now what we will says that we say that there are two clusters.

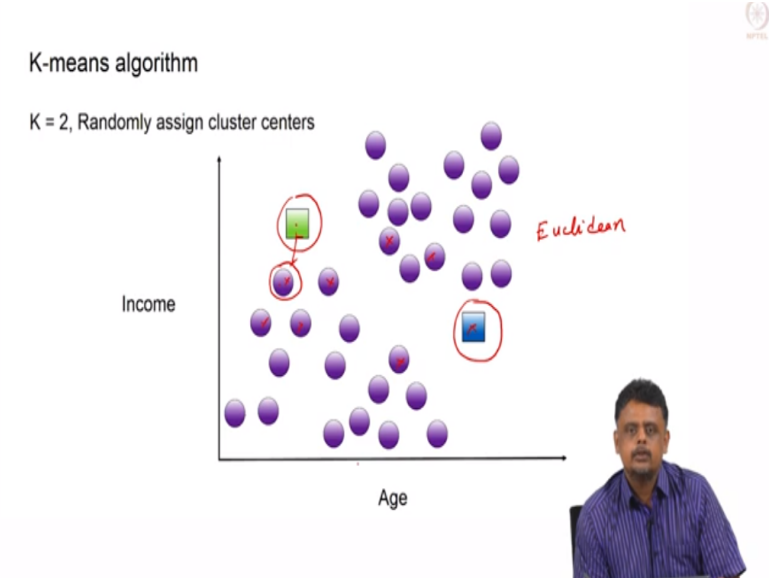
So when you plot the data it becomes our base those are two clusters very obvious two clusters but in again in many unsupervised learning algorithm the number of such clusters is also unknown especially for K-means where ever you this scheme means the number of clusters is also typically unknown, so by that is another kind of hyper parameter that you would have to determine in this case.

(Refer Slide Time: 05:21)



Let us say we want to separate two cluster, so what we do so we initialize two clusters centers so to speak right so these are the two cluster centers that we have initialized correct and so this is random and again we can make some very clever guesses of the initial status based in the data itself but right now we just say we will pick two cluster centers at random and of course when you say at random it depends on the range of your data, so you have to pick that again something which is rational irrational choice of the cluster centers.

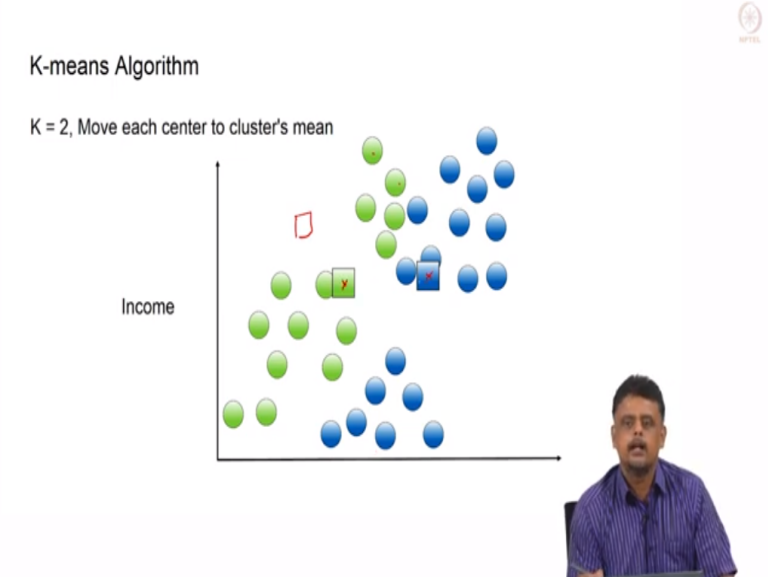
(Refer Slide Time: 05:55)



So we randomly assign it and what we do we move each center to the clusters mean right, so what does that mean here so we assign two randomly chosen cluster centers right and following which or we do is we calculate the distance of each one of these data point so there are several data points again initially we do not know what clusters they are in so we would calculate the distance in this case Euclidean distance let us say clearly and distance between the cluster center and each one of the rate of points and assign the data points to the cluster center which is closes.

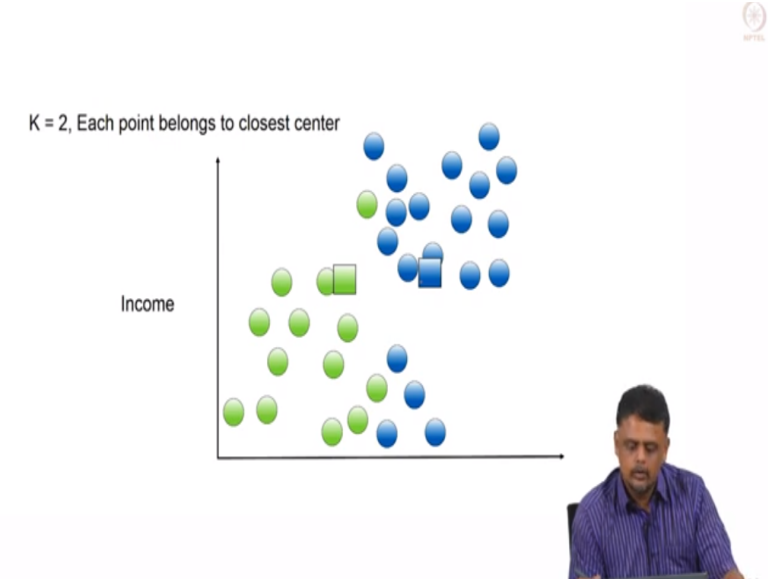
So for instance this particular data point is closes to this cluster center as opposed to that one and so we would assign this data point to this cluster center, so similarly we would go across E or go across the data set and assign each one of those points to the corresponding cluster centers which is closest to them.

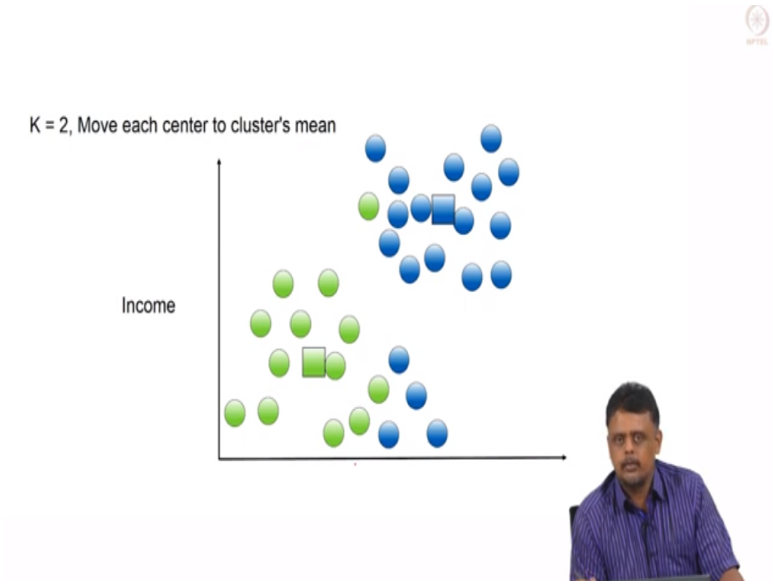
(Refer Slide Time: 07:09)



So then once we do that we would get to different sets of clusters green and blue in this case following which we would recalculate the cluster center based on the membership, so now we have all this green points which now been assigned to remember the somewhere here for the green and after we have assigned the closest point to that cluster center we would recalculate the mean of the cluster of the points in the cluster, so to get a new cluster center here these are the new cluster center for the green and blue.

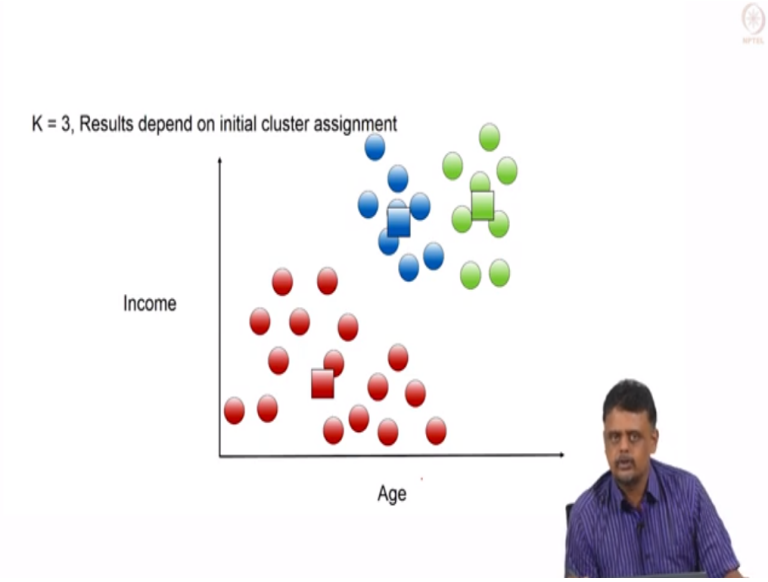
(Refer Slide Time: 07:49)





Once again we would redo this calculation that we did earlier that is reassigned by calculating the distance of each one of those points the new cluster center and then reassign the label accordingly, so then you see that the label has change through dramatically from here to there and then we would once again recalculate the cluster center based on the new labeling, so we continue this till the cluster centers do not change significantly and then we stop right there and so now we have the membership belonging to the clusters.

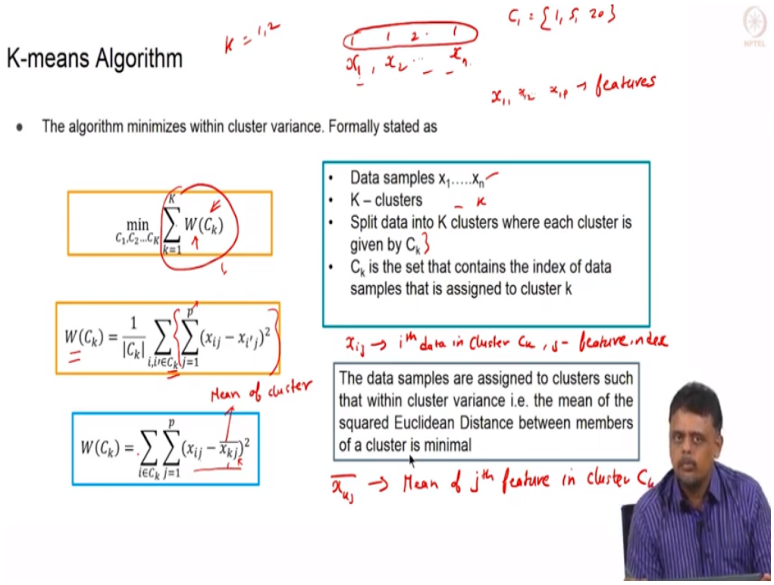
(Refer Slide Time: 08:24)



So we can also set K equal to three and do the same calculation as we outlined before and do the same calculation as we outlined before and we can get about three clusters in this case red blue and green and the problem is with K-means is that it is very sensitive to the initialization, so we can have a completely different initialization of the cluster centers and end up with a different se clustering set so it is a same data set so we had this red blue and green clusters and with the different initialization we will get something like this which is different from the previous clustering that we obtain.

So this is very sensitive to the initial condition all, so we will see later as to how we can resolve this problem that is how do we decide which one of the cluster and as is correct and which one we should discard, so here is one more example of K equal to three here once again K equal to three and we have a different starting point or the initial values of the cluster center and we got different set of clusters.

(Refer Slide Time: 09:25)



So to summarize we put it in formally the K-means algorithm is obtain by actually minimizing a cost function so which is shown right here so that is the cost function, so that we are trying to minimize so it is W is what is called the inter class or the intra class variance, so we will see what that is soon so what does this trying to optimize the idea is to optimize the assignment of each of the data points to a cluster center, so that this cost function is optimized, so this is a combinatorial optimization problem.

So remember it is this cost function is minimized by assigning the appropriate label to each one of the data point, so let us say we have data points X1 X2 up to XN and we have K classes in this let us say K is like 1 2 we say two classes, so there are labels one and two, so the optimization problem is to figure out the right assignment, so we will say X1 is one ,X2 is one, X3 is two so on and so forth, XN is some class one again so this arrangement here is the outcome of the optimization so what is the correct arrangement of these labels, so that this inter class variance is optimized, so that is the problem that we are trying to solve.

So we will write this out little bit more detail so we see that the inter-class variance is defined by this formula it is nothing but here P in the summation is the number of feature in your input data, so X so if you have data point X1 right if there are P feature then each data point has X1 one two X1P features right, so given a particular class CK, so in this particular summation what you are

trying to do is determine the distance between, so this is XIJ minus XI prime J so where I and I prime or the data point labels inside a particular cluster.

So let us say we have an initial clustering and we are looking at only the elements in that cluster and we are calculating the sum of these distances between the elements in the cluster right and we do so for all the clusters and every time we would divide by the number of elements in that cluster, so this is the variance at our interact last variance that we are trying to minimize we can rewrite this formula as XIJ minus XK d bar where XK J bar is the mean of feature where is the mean of the cluster, which is also known as the cluster centroid, so it is a mean of the cluster K and J is the data the feature index.

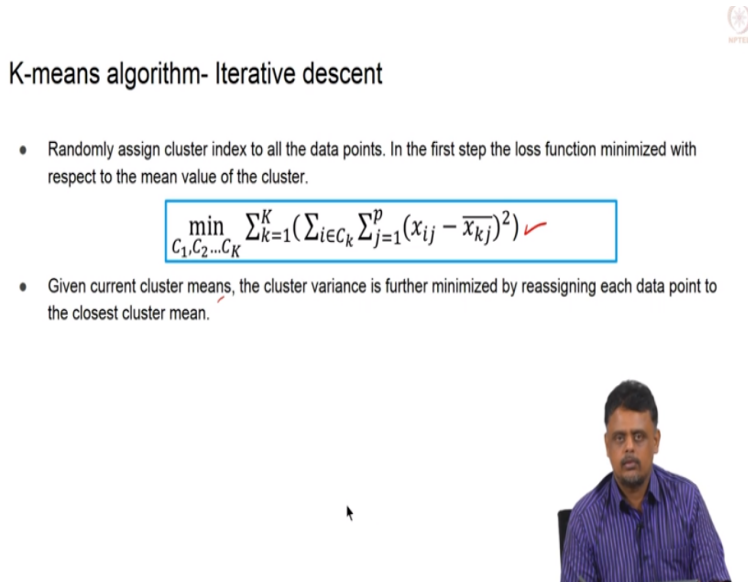
So X IJ is the I^{th} element in cluster CK and this jet feature and X KJ bar is the mean of the jet feature in cluster K, so that I hope that is clear so X IJ is the I^{th} data in cluster CK and J is the feature index, so similarly X KJ bar would be mean of J^{th} feature in the cluster CK, so that is what we trying to do here so it is basically we have you can rewrite this formula this way and this is the cost function we trying to optimize there will be one more summation if you bring this summation down there is one more summation here, so that is the cost function we are trying to optimize, so it is a combinatorial optimization problem so it is not-trivial to do.

So we use like a greedy approach or something called an iterative descent that is what is commonly referred to as the K-means algorithm, so just to clarify the notation again the data samples are indicated by X1 to XN there are n data samples there are K clusters and a cluster index is small K cluster index is small K so and we want to split the data into clay cluster where each cluster is denoted by CK, so CK is just a set of all the data point is this is which correspond to that cluster, so because the data point in this is range from X1 to XN, so class C1 cluster C1 would be just a set of indices of the data points so which basically, so let us say data point one five and 20 or in cluster C1.

So that says C1 right and there are other properties all the CK r do not have intersection, so one data point will only be strictly assigned to one cluster and the union of all the cluster will give you the entire data set right, so the idea again is to as I mentioned earlier these to assign the data to cluster centers such that within cluster variants that is the distance the squared distance

between the elements or the data point in that cluster is minimized, so that is how we that is the one problem that we are trying to solve in summarized algorithm.

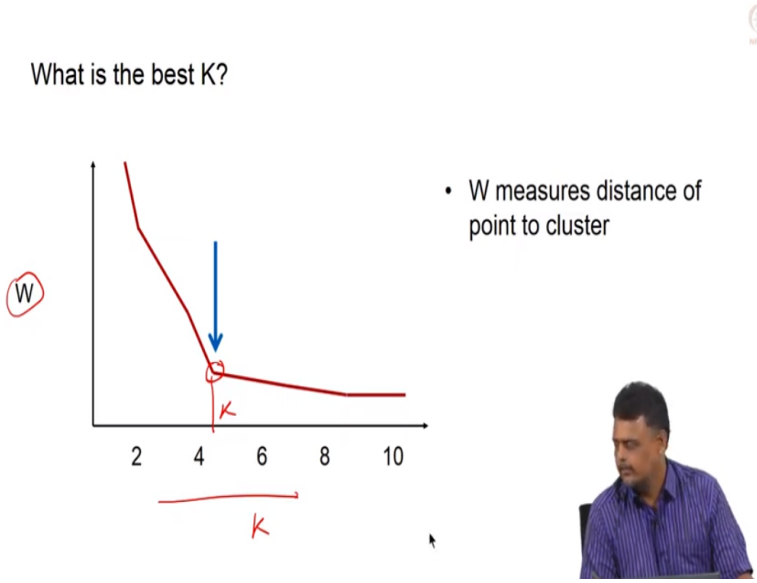
(Refer Slide Time: 15:37)



So I just rephrase it differently said that randomly assign cluster index to all the data points which is the same as saying you choose a cluster centroid first and based on the distance to that centroid you would assign a index cluster index to each one of the data point, so I the first step the last function that we had mentioned earlier and I mentioned earlier and have reproduces here for convenience is minimize with respect to the mean value of the cluster, so we would determine the mean value of the each of the clusters based on the data points in it and in the second step given cluster means you can minimize the cost function further by reassigning each point to the closest cluster mean.

So this done repeatedly and each and you can see that in each iteration the cost function the value of the last function will be minimize till point where in there is no appreciable change in the cluster center location you die trait but this is one of the more commonly used algorithm also refers to the K-means algorithm there are other ways of solving this optimization problem also.

(Refer Slide Time: 16:56)

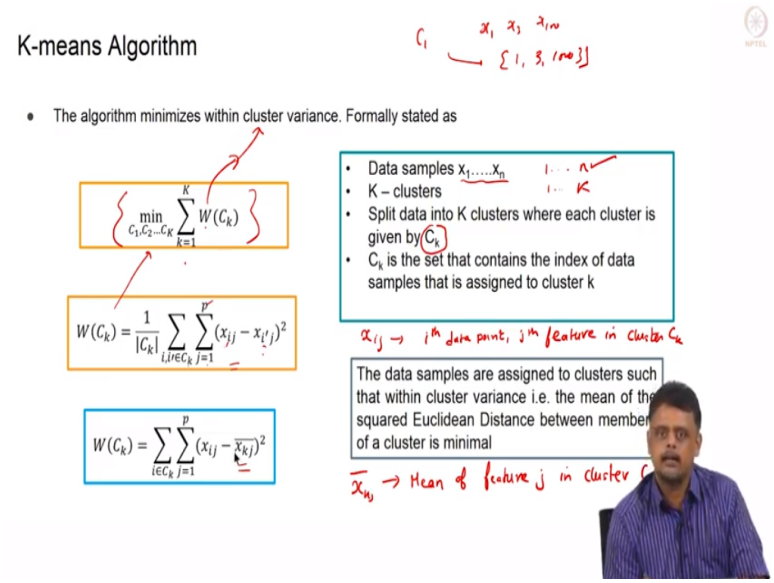


So the earlier question we had is what how do you determine the best possible value of , so that is not it is question is easy answered, so typically what one would do is to plot this W that we mentioned the interact the inter class vary within class variance that we calculate as part of the optimization we plot that for the final optimized value we plot that as a function of K here this is K all right and then typically you can notice this knee so we are in once beyond the point the W value decreases but not as much so you will have a knee in your plot this infection point and we choose the K corresponding to that inflection point.

So that is K that is, so there is another problem that we also talked about that is where in we were saying that if we had different initialization then we would end up with typically different clusters right, so then how do you figure out which is best possible cluster to which for best possible clustering that we have obtained now in order to do it again we would do the same strategy that is we look at the cost function the loss function that we calculated and choose the cluster with the least value or choose the clustering result with the least W value.

So that is the solution to that problem however the choice of K sometimes will be given and sometimes one has to discern it from the data itself, so it is basically very data dependent and so generally there is not too many too much work being done in this area because it is something that is kind of outside the algorithm itself.

(Refer Slide Time: 18:42)



So let us look at the cost function or loss function that is optimized by the K-means algorithm, so we will set up the problem first so initially we are given data samples X1 to XN, so one to N the indices of the data samples we have we decide that there are K clusters in the data typically sometimes this is given K is given and sometimes you have to discern from the data itself, so the cluster index goes from one to capital K and of course what we saw earlier was that we want to split the data into K clusters and each cluster is given by CK.

So what does CK contain so CK has the indices of the data points, so let us say data of X1 X3 or X100 if N is let us say thousand all belong to class C C1, so since C1 will have the indices one three and, so that how we would define the CK, so we want to split the data into K sets and the K sets do not have any intersection of any of the CK any to CK would be a null set and the union of all this CK would be the entire data set that you are using and how do we optimize how do we post this as an optimization problem.

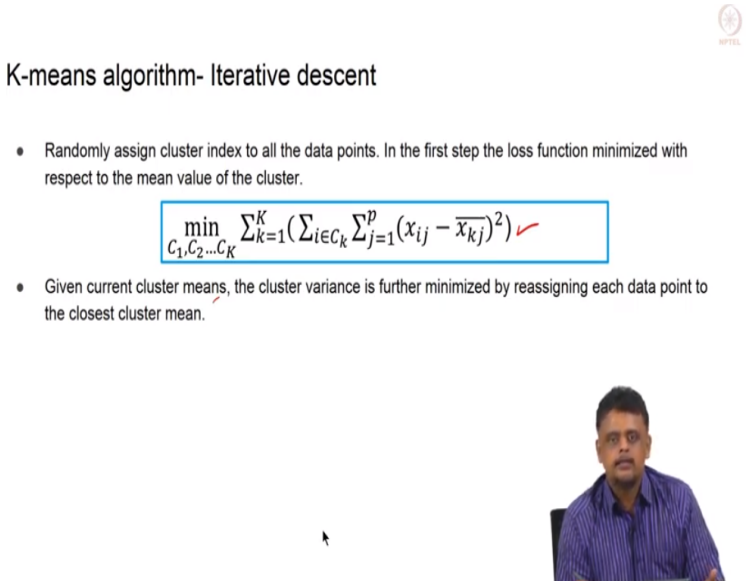
So we want to assign data samples to cluster such that the within cluster variance is minimized, so we can think of the within cluster may variance as the sum or the average distance between the members of the cluster between the data points in a cluster in the distance in terms of Euclidean distance right, so if we if you were to post this formally this is the optimization problem that you are trying to solve where W is the width in cluster variance so it is right here that is a variance and we want to take the sum over all the clusters K clusters, so for each cluster

we calculate this W which is the within cluster variance and we take the sum over all the cluster of course what are we trying to do in the process we are trying to figure out the assignment of the data point to the clusters that is what we are see we saw here earlier right such that this sum is minimized, so that is out optimization problem this W itself can be written in this form where in here the inner summation J runs through the feature so if it is one dimensional that is you are just given N data points and each data point is just a scalar value sometimes if you are N data points each data points can be a vector of values which the size length of the vector being P.

So in a some runs over the P features, so we calculate the distance you can distance between two data points I and I prime belonging to a particular cluster K and so for each data point spare of data points I and I prime we calculate the Euclidean distance between them by summing over it is feature distance and of course we take the mean value for every cluster right this is the cost function for this single cluster and for course and of course we have to sum this over all the a cluster it is possible to rewrite this particular cost function in this form, so here if we look at this XIJ in the summation where J is the feature index so XIJ is basically the height data point J^{th} feature in cluster CK and XKJ bar the average it is basically the mean of feature J in cluster CK correct so mean.

So we are just trying to as we saw earlier in the illustrations we just trying to find the distance of every data point within a cluster to the center or the cluster mean and we sum over all the distances inside a cluster and then we do the same for every cluster figure out and we add them all up and what the optimization problem solves is that we figure out the correct assignment of the data point to the clusters such that the summation that we calculate is optimized.

(Refer Slide Time: 24:06)



So if you were to post this as an algorithm and you are trying to solve it this is a combinatorial optimization problem and the way to solve it would be say an iterative techniques call it to dissent, so one way of looking at it is we randomly assign cluster index to all the data points, so let us say we have clay number of cluster K capital K then we would assign at random a cluster index to each one of the data points, so the first step is to minimize the loss function with respect to the mean value of the cluster.

So we know that once we the way to minimize the loss function is to figure out the distance of each one of those points from the cluster centers, that would minimize this cost function that we saw earlier, now once the cluster center is determined the second step would be to recalculate the distance of all the data point to each one of the cluster center and then reassign the data points to the appropriate cluster that the cluster to which the distance is the smallest, so we go back and forth between this two steps and till there is convergence in a sense that the cluster clusters stopped change.