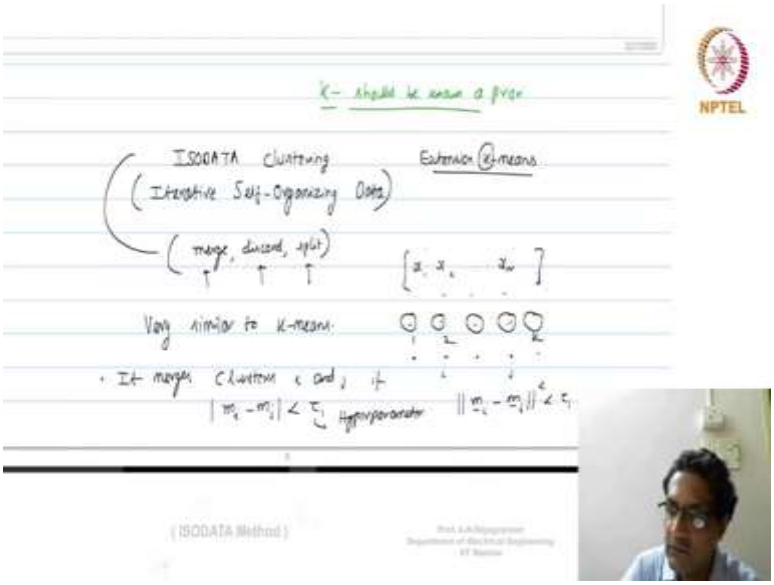
Image Signal Processing
Professor A. N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology, Madras
Lecture 62
ISODATA Method

(Refer Slide Time: 00:30)



So, we have seen K-Means and one of the main limitations of K-Means; although this is attractive in so many ways it has its strengths. One of the main weaknesses of K-Means is that you need to know K; K should be known a priori. It should be known a priori and exact knowledge is K there could be several applications where we do not have the exact knowledge of K. We might have an estimate of K; we might roughly know as to know probably K should be about this number but then it could be the eventual number of classes could be greater than that or could be a little less than that and so on.

In order to handle this shortcoming of K-Means, the other algorithm there is an extension of this, what is called ISODATA, ISODATA clustering. ISODATA clustering, and what is ISODATA stands for? It stands for Iterative Self-Organizing Data clustering, Self-Organizing Data clustering. ISODATA I will remove this fine, so ISODATA is Iterative Self-Organizing Data; this is a clustering method.

Again it is kind of it is an extension of K-Means; it is very similar to K-Means. In fact utilizes K-Means along the way but has its additional flexibility that K need not be needed. K can be lucky is allowed to change along the way as see iterations go on. Now, it hinges on 3 main things; it allows for merge, it allows for discard and it allows for split. I will I will explain what kind of each of these things mean.

So actually means that it can allow two kind of clusters to merge if their means are too close. It allows the cluster to be dropped, if it has too few samples in it and then it also allows the clusters to be split. I mean if it finds that the variance within that (())(02:40) within that particular cluster is way too high. So, I will write all these steps and then we will kind of see what they mean.

So, very similar to K-Means assume that you start with some value for K. Suppose we say that when we starting with K groups; so 1, 2 right up to k. But, then as I said we are we will be able to change K along the way; but initially we have some estimate we start with K. Now, what it does is the following, so once you have run through K-Means so you have these 3 data samples x1, x2 all the way up to xn. And just as we did in K-Means, we will start with some random initial assignment of the pattern centers.

Go through the, I mean go through a clustering process and then eventually end up with some kind of groups here. But, then what you need to do is even in the very first iteration so as you to get onto group, just as you would do in K-Means. With the first iteration, what do you do we start with some with some random centers which is simply pick from this data set randomly. And then you start assigning x1, x2 up to xn to let us say each of these data centers.

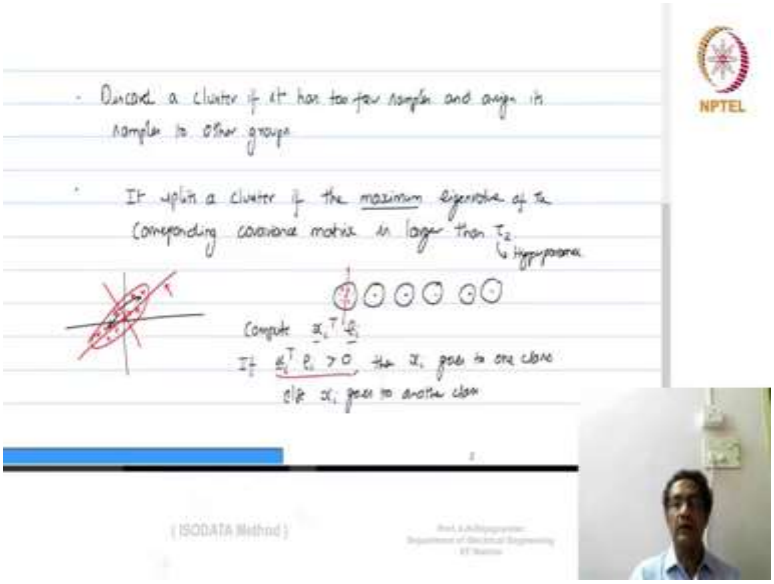
And then after you have gotten these data centers; you be a then you re-compute the means. When you re-compute the means, what you do is now you sort of say take a call; right you take a call whether we should decrease K or increase K or should we just stay with K. What is that kind of based on? So, at this point it and what it does is; so an ISODATA it merges clusters i and j.

Let us say i is somewhere here and probably j is somewhere there, if it is a mod of mi minus mj I mean this is if it is a scalar is less than some value tau 1. This will be a hyper parameter, there is something that we need to fix which is why ISODATA is also not so not so elegant. But, at the same time it gives you some flexibility; so the tau 1 is something that we need to fix. If it is a

vector then we will have to see norm mi minus mj, norm square we should say is less than some value tau 1.

If it is a scalar; if it is a vector so what does means is that, that these two groups we kind of believe that the means are so close that it does not make sense to create two separate groups for them, you simply merge them. You merge them and then what you would do is, you would actually create a new mean. After you merge them, then you get a new mean; and then your K will go down by norm because let us say two groups have been merged. If more than I mean if it so happens that there are more groups which tend to merge then we can actually merge them too.

(Refer Slide Time: 06:08)



Now, discard; right discard what it means is discard, so discard a cluster if it has too few samples, discard a cluster if has too few samples. So what this means is that I mean you do not even want to kind of consider that as an independent kind of cluster. Let us say that I have a cluster that has very few samples as compared to the number of samples in other clusters.

Now, I will simply remove this cluster from my entire grouping thing; so, it means that my K will come down come to by 1. And what I would do is I would reassign whatever samples here I would reassign that to the other clusters. Divide the discarded clusters if it has too few samples, and assign its samples to other groups. Again ofcourse use some kind of a similarity metric and Euclidian or something and then assign it to other groups.

Then I said that you can merge, you can discard, you can also split; now what you do? It splits a cluster. So, what it means is now let me first write it down and then I will explain, it splits a cluster if the maximum eigenvalue, so note this maximum, it is not like you are going to check all the eigenvalues; the maximum eigenvalue value of the corresponding covariance matrix, matrix is larger than let say tau 1, tau 2.

We use already tau 1, so again this will be a hyper parameter so we need to fix this. So, what does that mean, so merge is easy to understand, discard is easy to understand. What does it mean to split? So it means that after let us say some iterations is sitting here and then you suddenly find that this guy has a variance that is very high, now only the maximum eigenvalue.

So, if you go back to the way we did a PCA, at that time I drawn this diagram where I said that when you could have mean, if you do a zero mean and if you (make) may call it data sample zero mean; it could lie on a cluster like that, so a distribution like that. Now, you know that your data samples are probably situated like this, and then we say that the maximum variance is in this direction.

And then the next maximum is occurring in an orthogonal direction and so on, so each an eigenvector. Now, what this means is that if you find that the spread along the maximum eigenvalue that means with respect to this eigenvector which along which the maximum spread occurs. If the eigenvalue is higher than a threshold, it says that you split this cluster.

So, how do you how will you split this cluster? So you split this cluster such that; to split the cluster what you do? I mean suppose I take actually a data sample. Let me call this xi, then compute to check so what do you; compute xi transpose, let say the highest eigenvector is ei. If xi transpose ei that means if you take the inner product which is greater than 0; then xi goes to class 1.

Okay goes to let say one class else whatever it means that if it is less than or equal to zero; that means whether these points are lying on this side, or this side or whether these points are lying on this side else, xi goes to another class. So you splitting this cluster now by actually using the notion of maximum variance, direction of maximum variance.

And then and then right you simply finding out which of these samples should go to which go to will should go to what class. So, now you can think about this, so you have a situation like this where you have some groups there, some groups here. So, you got like maybe, now you could have more than K less than K already. And then if you look at this guy; this seems to have a higher variance, as we guys computed.

And therefore if we feel that, that these samples sitting inside this cluster; they are not really all that close. Because of the fact that the variance is high; that means this could be further split. So, that we can get two new groups wherein the data sample sitting inside each one of them could would look much more close. So, the idea is to split this into half and how to split this into half; is not is not immediately after it. How do you split the cluster and therefore you split the cluster in this manner.

And then similarly the K-Means, I say similar to K-Means you can, right I mean so you do the same thing, I mean you can kind of continue doing this until the labels are unchanged. Then after that you can again rerun it with another set of initializations and then try to pick the label assignment that has a minimum error. But, all this ofcourse we will require more computations; but that is okay at least this method gives you the flexibility to actually change K along the way.

So, typically you start with the K that you think there is a reasonable number, and then I say as this as this iterations go on K can decrease or increase depending upon what is happening inside your (())(12:23) data set, okay this is called ISODATA and this is very similar to K-Means, except that you can trigger look upon it an extension of K-Means which addresses the issue of lack of exact knowledge of K.