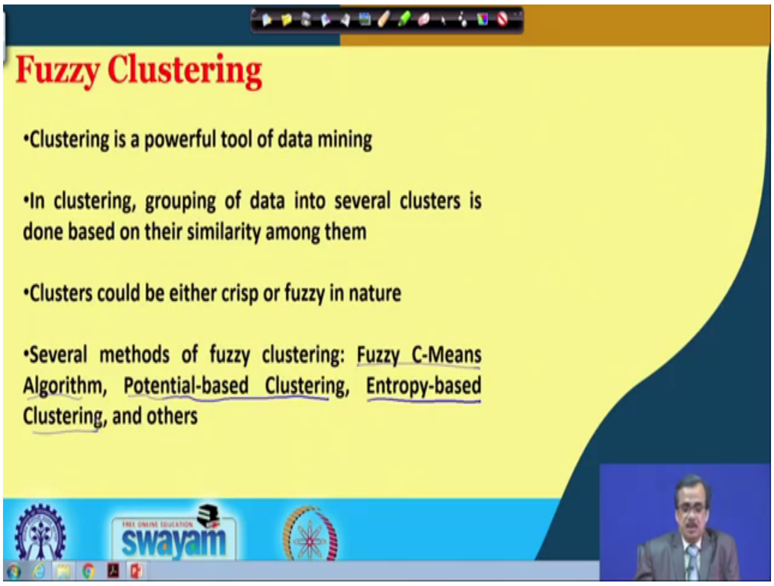
Fuzzy Logic and Neural Networks
Prof. Dilip Kumar Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 13
Applications of Fuzzy Sets (Contd.)

We are going to start with another potential applications of Fuzzy sets, and that is the form of fuzzy clustering.

(Refer Slide Time: 00:24)



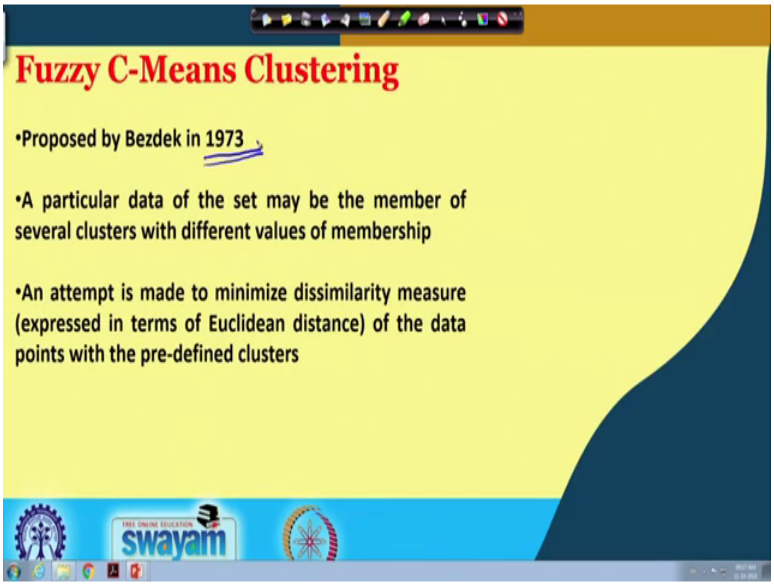
Now, this clustering is a powerful tool for data mining, and the purpose of data mining is to extract useful information from a data set. Now, actually what you do in clustering, the clustering is done based on the concept of similarity; that means, the two similar point should belong to the same cluster and two dissimilar points should belong to two different clusters.

Now, if you see the literature, the clustering could be either crisp or fuzzy in nature. Now, for the crisp cluster, there should be well-defined boundary, but for the fuzzy clusters the boundaries could be vague. Now, let me take one very practical example very simple example. Now, supposing that say you are staying in a hostel and in the hostel there are say 1000 students. Now, if you see, these 1000 students they will move in a few clusters they will form the clusters based on their similarity and might be these 1000 students will move in say 10 or say 12 clusters and you can see that.

So, a particular student may belong to more than one clusters and there is another possibility. So, a particular student may leave a cluster and he may join another cluster. So, these are all examples of actually the fuzzy clusters, and for each of the , there will be a leader and that is nothing but the cluster centers. Now, if you see the literature on fuzzy clustering, we have got a large number of methods for example, say we have got Fuzzy C-clustering, then we have got the potential-based clustering, we have got entropy-based clustering, and so on.

So, there are a few other methods, but out of all the methods of fuzzy clustering, the Fuzzy C-Means algorithm is the most popular one and we are going to , in detail, the working principle of this Fuzzy C-Means clustering.

(Refer Slide Time: 03:02)

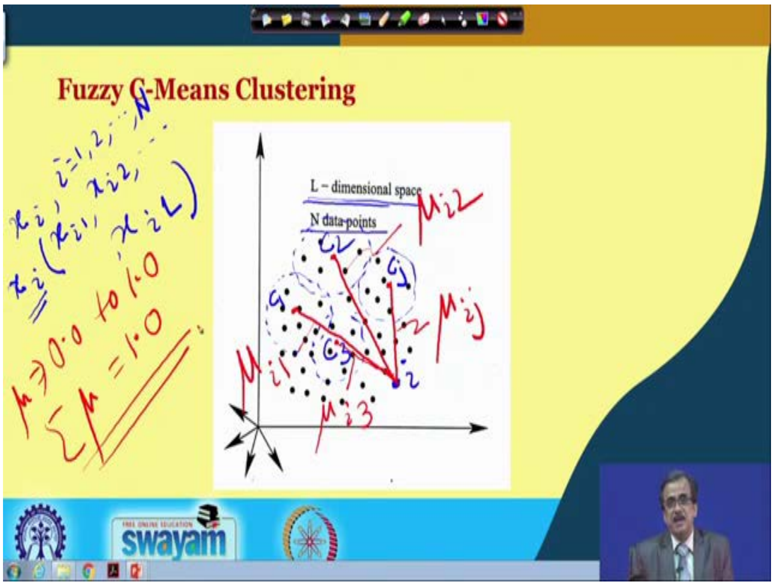


Now, if you see the Fuzzy C-Means clustering. So, this technique was proposed in the year 1973 by Bezdek and after that, actually a lot of modifications have been incorporated into the Fuzzy C-Means algorithm. Now, here, the way the clustering is done is as follows:

Now, supposing that I have got a large number of data points, and I have got a few predefined clusters. So, a particular data point may belong to different clusters with different numerical values of membership, and some of the membership function values will become equal to 1.0. Now, here, if I consider say a particular data point, and how much close is that particular data point with respect to the cluster centre.

So, what you do is, like we try to measure the Euclidean distance between that particular data point and cluster and the more the value for this Euclidean distance, the less will be the similarity and the more will be the dissimilarity, and our aim is to minimize this particular dissimilarity. So, that the data point can be brought very near to the cluster and it may belong to the cluster. Now, this particular principle has been used mathematically just to design and develop the Fuzzy C-Means algorithm.

(Refer Slide Time: 04:58)



Now, I am just going to discuss its principle, in details and after that, I will be solving one numerical example, just to make it more clear. Now, here I am just going to take a very typical example, now this example is as follows. Supposing that in the higher dimensional space, say L dimensional space, say I have got a large number of data points, say capital N number of data points. Now, each data point is denoted by x_i, where i varies from 1,2, up to capital N. And, our aim is to the form the clusters the fuzzy clusters of these particular data points based on their similarity.

Now, if I consider, a particular data point say x_i, now this is in L dimension. So, to represent this particular point I need to have the numerical values like x_i1, x_i2 and up to say x_iL. So, I need to have so, many numerical values, if I want to represent a particular data point say x i in L dimension. Now, let us see, how does it work, how does this particular Fuzzy C-Means algorithm work. Now I have already mentioned that to start with so, we will have to form a few clusters. That means, there should be some predefined clusters. Now, let me take some predefined clusters like this for example, say one fuzzy cluster could be something like this, another fuzzy clusters could be something like this and there could be overlapping also, because these are all fuzzy clusters. Another fuzzy clusters could be say something like this, another fuzzy clusters could be something like this, now supposing that I have defined say four clusters initially, and I am just going to take the decision whether a particular data point, say this particular data point will belong to cluster 1 or cluster 2 or cluster 3 and let me consider the cluster 4 as a general cluster say j-th cluster, say.

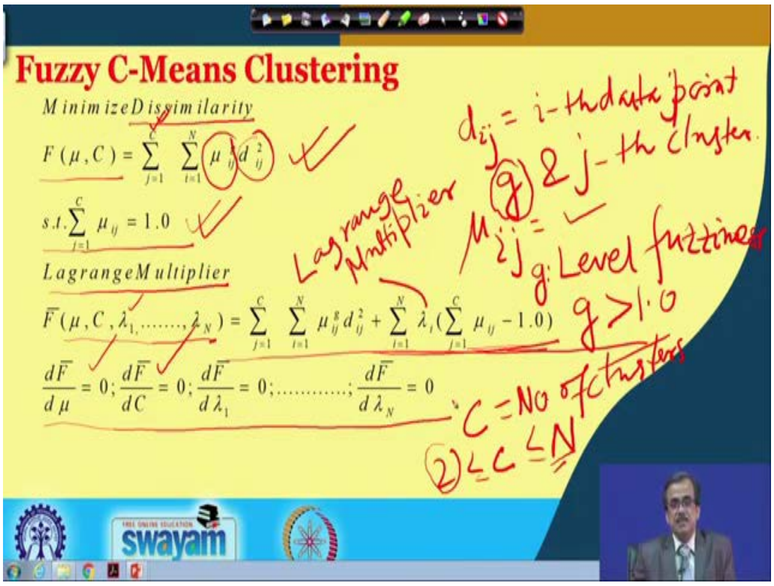
So, this is nothing but the cluster j. So, there could be some other clusters also. Now, how to take the decision whether this particular data point say the i-th data point should belong to any one of these clusters or not. Now, how to proceed, now the way we actually try to solve the problem is as follows. So, what you do is, as I told we try to find out your the Euclidean distance between this data point and the cluster centre, and that particular data the Euclidean distance as I told, the more the distance the less will be the similarity and the more will be the dissimilarity and our aim is to minimize this particular dissimilarity.

Now, if you see, say for the first cluster, supposing that I have got a cluster centre here and the second cluster, say cluster centre is here, the third cluster, the cluster centre could be here, the j-th cluster the cluster centre could be here and what I do is. So, from here this i-th point. So, we try to find out like what should be the membership value, supposing that the membership function value between the i-th data point and the first cluster centre is nothing but μ_{i1}, that is the membership value between the i-th data point and the first cluster. Similarly the membership value between the second cluster and i-th data point; so this is nothing but μ_{i2}.

Similarly, between the third cluster and the data point that is the i-th data point. So, μ is nothing but μ_{i3} and similarly your. So, this particular between i and j, j-th cluster. So, the membership function value is your μ_{ij}. So, μ indicates actually the membership function value and which varies from like 0.0 to 1.0. So, this is the range for the μ and here, a particular condition has to be fulfilled. So, that the sum of all the μ values becomes equal to 1.0.

So, this is actually a functional constraint let me repeat. So, with respect to this particular, and we have got different μ values with the different clusters and the sum of all the μ values should be equal to 1.0 and that is nothing but the functional constraint and what is our aim? Our aim is to find out the fuzzy clusters. Now, let us see, how do we proceed?

(Refer Slide Time: 10:26)



Now to proceed further, what I will have to do is, I will have to formulate as an optimization problem, and this particular optimization problem has to be solved using some technique.

Now, what I do is we try to minimize the dissimilarity, as I mentioned, and this particular dissimilarity will be expressed in terms of the Euclidean distance value that is nothing but is your d_{ij}^{2}; now what is d_{ij}? d_{ij} is nothing but the Euclidean distance between the i-th data point and j-th cluster. So, what is our aim? Our aim is to minimize the dissimilarity; that means I will have to minimize d_{ij}^{2} and it is multiplied by μ_{ij}^{g}.

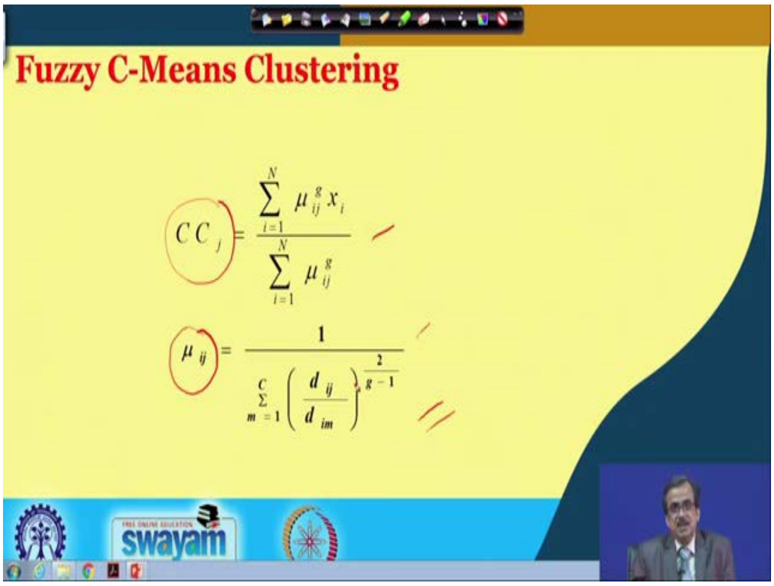
Now, what is μ_{ij}? μ_{ij} is nothing but the membership function value between the i-th data point and j-th cluster center, let me repeat μ_{ij} stands for the membership function value for i-th data point with respect to the j-th cluster centre and here, we put μ_{ij}^{g}, and this particular g is nothing but the level of the fuzziness. Now, if you remember that we discussed the power of a fuzzy set. So, this is almost similar to the power of the fuzzy set. So, this g is nothing but the level of cluster fuzziness and generally we consider g is greater than 1, ok.

Now, let us see how to write down this objective function? Now, our aim is to minimize the dissimilarity and that is, $F(\mu, C)$, μ is the membership function value, and this particular C actually it indicates the total number of clusters or the number of clusters to be made and C has got a range. For example, say C should be greater than equals to 2 and it should be less than equals to n, what does it mean? It means that the minimum number of clusters should be 2, it cannot be 1, if it is 1 then no clustering is done and this the maximum number of clusters cannot exceed n, that is the total number of data points. So, C is nothing but the number of clusters.

So, this $F(\mu, C) = \sum_{j=1}^{C}\sum_{i=1}^{N}\mu_{ij}^{g} d_{ij}^{2}$. So, this is the objective function, which I will have to minimize, subject to the condition that $\sum_{j=1}^{C}\mu_{ij} = 1.0$ and this is nothing but the functional constraint. Now, let us see, how to solve it using the traditional method of optimization and to solve this type of problem we take the help of Lagrange multiplier. So, what you do is. So, this objective function is written in a slightly different form and that is nothing but your $\overline{F}(\mu, C, \lambda_1, ...\lambda_N) = \sum_{j=1}^{C}\sum_{i=1}^{N}\mu_{ij}^{g} d_{ij}^{2} + \sum_{i=1}^{N}\lambda_i(\sum_{j=1}^{C}\mu_{ij} - 1.0)$ So, this is actually the functional constraint. So, this one you bring it to the left hand side. So, I will be getting this particular the expression and here. So, this particular λ_i is nothing but is your Lagrange multiplier. So, this is nothing but is your Lagrange multiplier, now I have got only one big expression for this objective function and I will have to solve using the method of calculus. Now, what we do is, we try to find out the $\frac{d\overline{F}}{d\mu} = 0$. Then, the $\frac{d\overline{F}}{dC} = 0$, then derivative of $\frac{d\overline{F}}{d\lambda_1} = 0$, then we write $\frac{d\overline{F}}{d\lambda_2} = 0$ and the last is $\frac{d\overline{F}}{d\lambda_N} = 0$.

Now, if we put all such things equals to 0. So, you will be getting a set of equations. And, those equations, if you solve then there is a possibility that you will be getting actually the expression like.

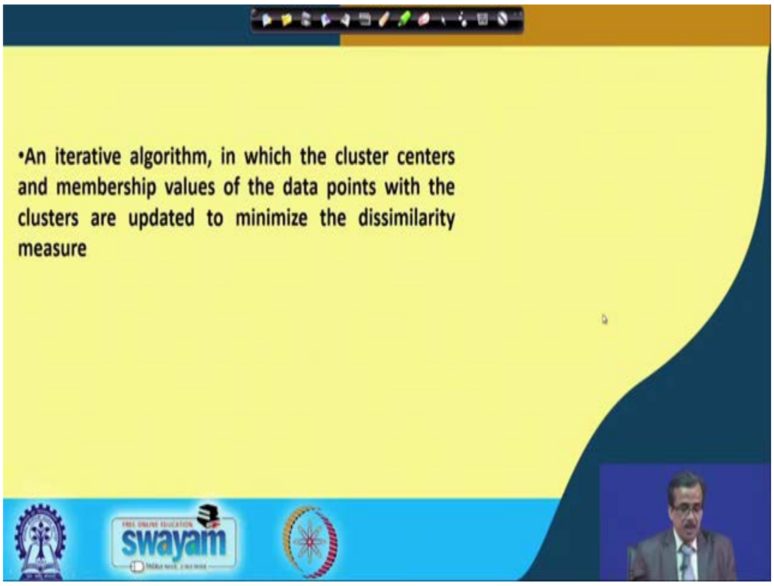
(Refer Slide Time: 16:11)



The CCj is nothing but the cluster centre corresponding to the j-th cluster. So, what should be the expression for the cluster centre? Now, $CC_j = \frac{\sum_{i=1}^{N} \mu_{ij}^{g} x_i}{\sum_{i=1}^{N} \mu_{ij}^{g}}$.

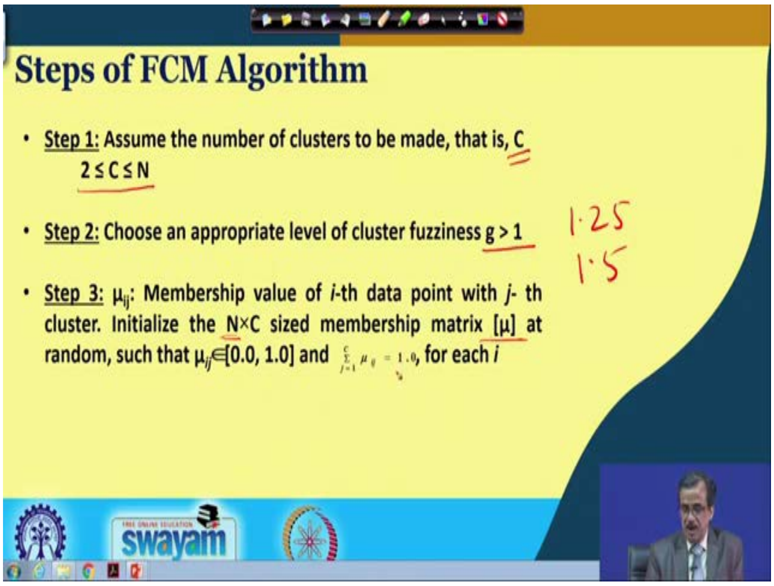
So, this is actually going to give the coordinate of the j-th cluster centre. Now, the j-th cluster centre, now as the data are in L dimensions, it will have L numerical values and accordingly, I can write down like your CC_jk, now k varies from 1 to l. So, I can find out the information for each of the dimensions of this particular the cluster centre. And, another expression you will be getting by solving those equations that is nothing but $\mu_{ij} = \frac{1}{\sum_{m=1}^{C} (\frac{d_{ij}}{d_{im}})^{2/g-1}}$. So, if we solve those equations, we will be getting these two particular expressions. One is actually how to determine the coordinate of the cluster centre and another is how to determine the membership value of a particular data point with respect to a cluster say j-th cluster and what is our aim? So, in this algorithm, what we do is, we try to update the cluster centre and the membership value of the data points with respect to the cluster centre iteratively. So, this is actually an iterative process. So, this algorithm is an iterative algorithm.

(Refer Slide Time: 18:32)



So, this I have already mentioned.

(Refer Slide Time: 18:35)



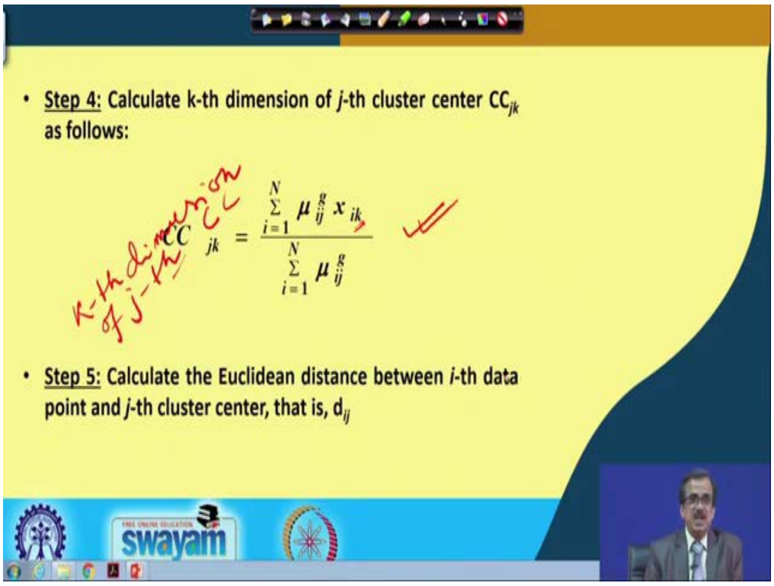
Now, I am just going to tell you the steps of this Fuzzy C-means algorithm like the FCM algorithm. So, one after another, now the step 1: we assume the number of clusters to be made and supposing that that is denoted by C and as I have already discussed the C is greater than equals to 2 and less than equals to N. Now, step 2, we select some appropriate level for cluster fuzziness. And so, this is nothing but g, generally it is considered g is greater than 1 for example, it could be say 1.25, 1.5 ,and so on.

Then, step 3: So, this μ_{ij} that is the membership value of the i-th data point with the j-th cluster centre. So, what we do is, we try to initialize at random and then, we try to modify it through a large number of iterations, now if there are N such data points. So, capital N number of data points and if we are generating say the C number of clusters. So, for each data point I should have C number of numerical values for the membership and I have got N number of data points. So, the size of the initial membership matrix that is μ, it will have the dimension like $N \times C$, N is the total number of data points and C is the total number of predefined clusters.

So, I will have to generate initially the membership matrix μ at random and what is the size of that particular matrix, that is, $N \times C$. Now, this I have already mentioned that μ varies from 0 to 1 and the sum of all the mu values corresponding to a particular point is nothing but 1.0.

So, this is the way actually the algorithm is working.

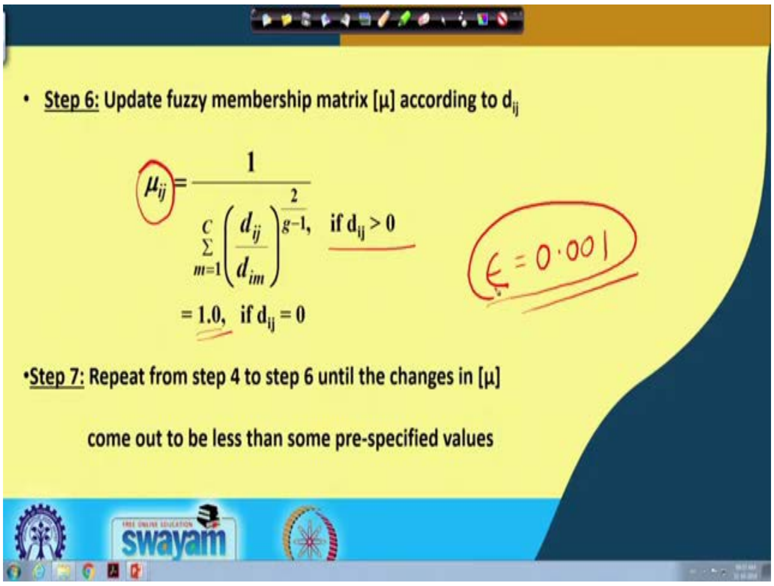
(Refer Slide Time: 20:53)



Now, then we go for step 4. Now, here actually what we do is, we try to update the cluster centre that is your the k-th dimension of the j-th cluster centre CC_jk is nothing but the k-th dimension of the j-th cluster centre j-th cluster centre CC. So, $CC_{jk} = \frac{\sum_{i=1}^{N} \mu_{ij}^{g} x_{ik}}{\sum_{i=1}^{N} \mu_{ij}^{g}}$.

So, this is the way actually we can calculate the k-th dimension of the j-th cluster centre. Now, in step five, we calculate the Euclidean distance between the i-th data point and j-th cluster centre and that is nothing but is your d_ij.

(Refer Slide Time: 22:03)



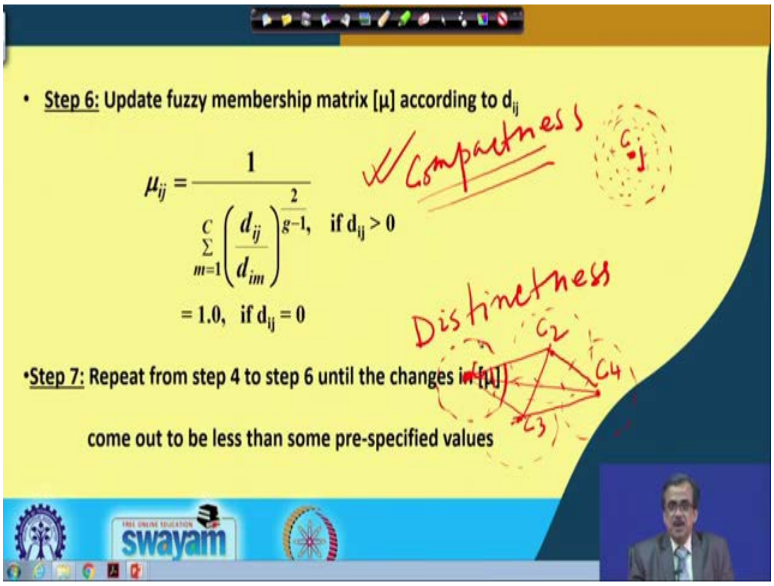
Then, in step 6, we try to update the membership value of the i-th data point with respect to the j-th cluster centre, that is your μ_{ij}. Now, $\mu_{ij} = \frac{1}{\sum_{m=1}^{C} (\frac{d_{ij}}{d_{im}})^{2/g-1}}$, if $d_{ij} > 0$. And, if d_ij is found to be 0, then what happens the distance between the two points is 0 ; that means, the similarity is the maximum and your mu_ij will become equal to 1.0.

Now, step 7: we repeat from step 4 to step 6, unless the change in μ values come out to be less than some pre-specified value, say it is denoted by epsilon; supposing that ε is say 0.001, a very small value. Now, I am running this particular algorithm, now after running this particular algorithm through a large number of iterations, there is a possibility that the μ values are going to reach some saturated level. And, after that, there may not be any significant change in the values of this particular the mu. And if this particular change is found to be less than equals to this, then we say that the algorithm has reached the optimal solutions and we try to terminate the program.

So, this is the way actually we terminate that particular the program. So, this shows actually this algorithm, the steps for this particular algorithm. Now, here, I just want to make one comment. Now, this algorithm is an iterative algorithm. Now, as it is an iterative algorithm, there is a possibility that the quality of the clusters will be updated and iteration-wise and there is a possibility that as the algorithm runs, we will be getting more and more compact clusters.

Now, I am just going to define, what do you mean by a compact cluster, and there is another objective, which we very frequently use, that is the distinctness of the clusters.

(Refer Slide Time: 25:00)



Now, here I am just going to define, what do you mean by the compactness of a particular cluster. So, the compactness of a cluster and another is your distinctness; distinctness of a particular cluster, now let me try to define, what do you mean by the compactness of a particular cluster?

Now, supposing that I have got say one fuzzy cluster something like this and say I have got a cluster centre, supposing that the cluster centre is denoted by c_j. Now, here surrounding this particular j-th cluster, there will be a large number of members, who are following actually the cluster centre, ok. Now, what we do is, we try to find out the Euclidean distance between the cluster centre and all the points surrounding it. For example, say here there are 200 points surrounding the cluster centre. So, what you do is, we try to find out the Euclidean distance between the j-th cluster centre and each of these two hundred data points surrounding that cluster centre. So, how many Euclidean distance values will be getting? We will be getting 200 Euclidean distance values and what you do is, you add them up and you find out the average. Now if the average Euclidean distance is less we define that this particular cluster is a very compact cluster.

So, this is the way actually we define the compactness of a particular cluster. Now, let us try to define the distinctness. Now, supposing that say I have got a large number of data points and we have got a few fuzzy clusters for example, say one clusters could be something like this, say this is the first cluster and its centre is your c_1, I have got another fuzzy cluster something like this and supposing that its cluster is c_2, then I have got another say might be I am here. So, I will be getting another clusters say c_3. I have got another cluster here and this is nothing but the cluster centre is say c_4 and so on.

So, what we do is, we try to find out the Euclidean distance between the cluster centers, for example, we try to find out the Euclidean distance between c_1 and c_2 then comes c_1 and c_3, then comes c_1 and c_4, we also try to find out what is Euclidean distance between c_3 and c_4, c_3 and c_2 and c_3 c_1, I have already calculated then we try to find out what should be the Euclidean distance between 2 and 4, and so on.

So, all the Euclidean distance values between the two of these possible clusters we determine and we try to find out what should be the average value. Now if the average Euclidean distance between the cluster centers if it is found to be more then we declare that this is a very distinct cluster. Now, if you use this Fuzzy C-Means algorithm, there is a possibility that you will be getting a very compact cluster, but at the cost of distinctness. So, you may not get a very distinct cluster using this particular the Fuzzy C-Means algorithm. But, what is our aim? Our aim is to get more compact as well as more distinct cluster, and how to get it. So, that I am going to discuss after some time.

So, these are actually the relative merits and demerits of Fuzzy C-Means algorithm.

Thank you.