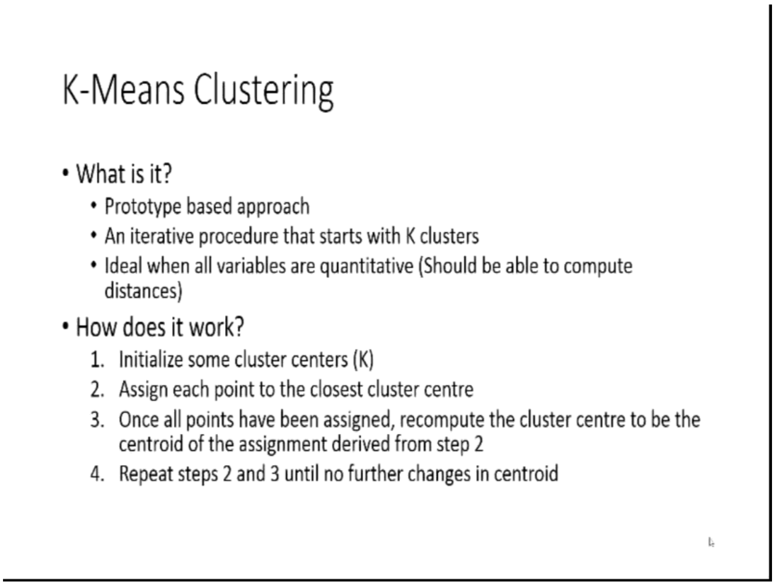
Introduction to Data Analytics
Prof. Nandan Sudarsanam
and Prof. B. Ravindran
Department of Management Studies
and Computer Science and Engineering
Indian Institute of Technology, Madras

Module - 08
Lecture - 42
Clustering Analysis – 2

Hello and welcome to our second lecture on Clustering Analysis. In the first lecture, we introduce the idea of clustering, differentiated it from say classification and other supervised learning techniques. And we explain, what clustering is, how it is useful, where it can used and also gave you brief overview of the different types of clustering and the different types of clusters.

In today's lecture we are going to introduce two very popular clustering techniques. The first is K mean clustering and the second is called the hierarchical clustering, where both these techniques are fairy old, this still enjoy immense popularity in terms of being actually used. The first one, the K mean clustering and this choice of K mean and hierarchical, you should find to be also fitting. The overall, the first dichotomy that we used when we talked about clustering approaches, which is partitioning based approaches and hierarchical approaches. So, the K mean clustering is essentially the partitioning based approach.

(Refer Slide Time: 01:15)



K-Means Clustering

- What is it?
    - Prototype based approach
    - An iterative procedure that starts with K clusters
    - Ideal when all variables are quantitative (Should be able to compute distances)
- How does it work?
    1. Initialize some cluster centers (K)
    2. Assign each point to the closest cluster centre
    3. Once all points have been assigned, recompute the cluster centre to be the centroid of the assignment derived from step 2
    4. Repeat steps 2 and 3 until no further changes in centroid

And, what we will do is, we will dive into the algorithm. So, we have already mentioned that is a partitioning based approach. So, it is one, where you partition the entire data set into K clusters essentially and you, it is also very useful to think of the K means as a prototype based approach, where that is also something that we discuss in terms of how clusters are formed. And the prototype based approach is one where, there is like this representative for each cluster and you use these representatives in some fashion to express, what a cluster is and to also form the clusters.

So, it is essentially this prototype based approach, where you create K clusters and it is also noteworthy that it is an iterative procedure. So, even right at the end of the first iteration you already have K clusters and they might not be good clusters, because it is just the first iteration. And then, like most optimization procedures like steepest descent, any of these other iterative producers you will find that over time you are just refining a solution and at some point, it does not make sense to refine any more. You are not, your clusters are not changing essentially prototypes or not changing either location or who they are and, so you stop at some point.

This procedure is ideal when all variables are quantitative, whether what we really mean is that you should be able to take each data point and you might have some other data point or some other location and your location is defined across the multiple attributes associated with the data point. So, a data point, again the conception you have in your mind is these rows and you are basically grouping data points, that is your job. And each data point is described by some attributes, which are these columns in a table and each column means something.

So, with the K means procedure, what you doing is you want to say that that you want to take this conception and you want to actually, you want actually be able to compute distances between two points across these dimensions, which means that attributes themselves need to be quantitative. This in a lot of ways to remind you of this K nearest neighbors, examples that we took up especially in the case of K nearest neighbors regression, where each of these dimensions for those each data point is a continuous quantitative variable.

And once you do that you just able to be compute distances, typically whenever it comes to computing distances we always use Euclidean distance, you could also use other measures of distance and you would actually, it would still be a K means procedures, but with different approach, so common once are Manhattan distances, Euclidean distances and so on. So, how does it really work? So, let us just go through the algorithm in some senses and then, we will see a graphical representation of it, the idea is to initialize some clusters centers K.

So, what you mean by initialize cluster centers? What we mean is, essentially think of this and it is easy to think of it in a graphical sense or you can think of it in a mathematical sense, but the idea is that each data point can be expressed in terms of an x axis, y axis, z axis, w axis, so on and so forth, depending on the number of attributes there are that represent each data points. Now, while each data point is represented based on these different attributes, you can create a cluster center also on these attributes.

And often it really make sense to put cluster centers, where you think the clusters are going to be form, but you see how even if you do not do that perfectly sometimes the clustering the K means clustering will adjust for it. But, the most important thing to take away from this is that, the K means cluster algorithm therefore, could be sensitive to where you place the cluster centers. If you do not place them in sometimes the right places, you could get to a solution, which is not necessarily a good solution.

So, the idea is that you initialize some cluster centers, so specifically you would initialize K cluster center, so you will, because it is K mean clustering. So, K is usually a number between 2 and may be 10 or 20 depending on the… The higher limit is a little more vague, most problems you looking at between 2 to 6 or 7 clusters at the most. But, there are contexts, where your data set is really large or what you intend to do with the clustering is such that you do not mind having more clusters.

So, the algorithms initially you drop in K cluster centers and then, what you do is you take each data point and if you figure out, which of this cluster centers is closest to the data point and then, you assign the data point to the closest cluster center. And, so essentially it is like if you kind of give names. So, just to give you some inside, this is where we are, where the second bullet point.

So, what we are doing is we basically saying if you wanted to give those K-cluster centers name as cluster 1, cluster 2, cluster 3, then what we would do is, we would go to each data point and say, which cluster center is closest to you and then, I am going to assign you to that cluster center. Now, once all of these points have been assigned, what we are going to do is we are going to recompute the cluster center to be essentially the centroid of this assignment of data points derived from step 2.

So, what we are going to do is essentially in step 2, what you did is you assigned a whole bunch of data points to different clusters. So, what we will do is, we then, for instance take all the data points that were assigned to cluster 1 to the centroid 1 essentially, to the cluster center 1. And we gave them all the labels that you are assigned, the data point x you are assigned to the cluster sector 1, data point y you are assigned to cluster center 1.

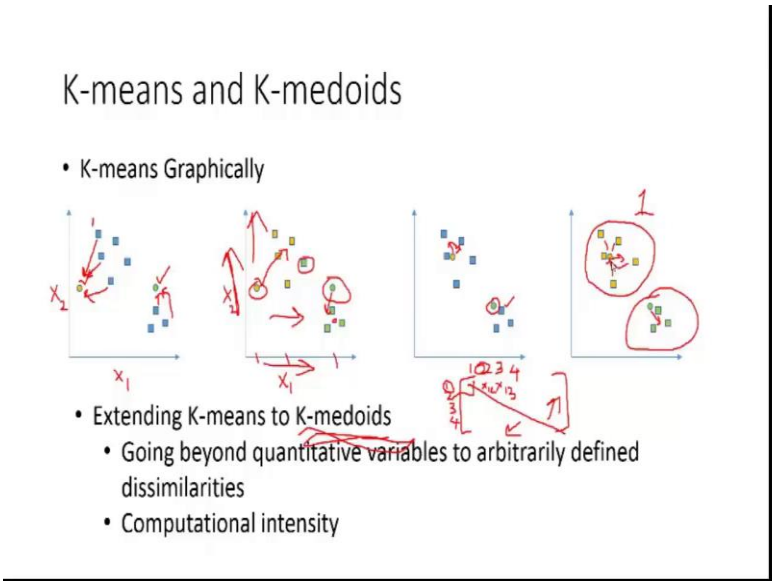
So, all of them that were assigned to cluster center 1 we take those data points and compute the centroid of those data points. Centroid essentially is in many senses like an average. It is, again it depends on exactly what measure of distance you choose, but if you are choosing Euclidean distances, the centroid is essentially like the average. And we say it is the average of these data points, but it is called the centroid, because it is the average across all those dimensions. The numbers of dimensions are the number of attributes, so across those dimensions, what is the center.

So, once all these points have been assigned you kind of compute a new cluster center based on the centroid. And it is essentially like you forget the old cluster center and now, this new cluster center is your cluster center and you do this for each assignment. If 10 data points were assigned to the cluster center 1, then you do this step for cluster center 1 data points. And then, you go to the next 15 data points set for perhaps assigned to the cluster center 2.

So, for these 15 data points find out the new centroid and call that the new cluster center. So, you got new K set of cluster centers. What do you do next? You iterate, for these new K cluster centers you go to each data point and it is almost like you forgot, which cluster center that data point belong to, because remember that data point was assigned to say cluster center 1 or 2 based on the old centroid, based on the old cluster center.

Now, because your cluster centers are moved based on the centroid, you now assigned them all over again and once you done the assignment all over again, you find the centroid. Once you find the centroid, you do the assignment that is how it is an interactive process. Essentially you repeating the steps 2 and 3 until the centroid is not moving any further or in some cases, you might say the centroid is moving by an amount smaller enough that is within your tolerance.

(Refer Slide Time: 10:03)



K-means and K-medoids

- K-means Graphically
- Extending K-means to K-medoids
  - Going beyond quantitative variables to arbitrarily defined dissimilarities
  - Computational intensity

So, this was fairly abstract in terms of bullet points, so let us actually take a graphical look at this if you taken an ultra simplistic example, where there are only two attributes. So, let say attributes is x_1, x_2 always remember that with clustering, which is unsupervised there is no y there is no output variables, what we doing is trying to grouped the data and here is the data. The data is this blue squares and you are trying to grouped this squares.

Now, fairly naive look at this can kind of make it obvious that perhaps this is one cluster and this is one cluster. So, the blue dots are actually the data points, now that is to the naked eye and the things to remember are that you know if for instance there were more dimensions that were there were more attributes beyond x_1 and x_2 it would be harder to show you this visually. So, that point the math kind of take over in 3, three dimensions I can show you x_1, x_2, x_3, but after that whatever I do if you have more attributes.

So, the squares are essentially the data points and let say the K means clustering algorithms started with two centroids here obviously; that means, K = 2. So, you initializes two centroids and you know this like I said the algorithms itself to some extend could be sensitive to how you initializes the centroid. So, there is no right way and the wrong way. Ideal would be if you could actually plot the centroid in the middle of the clusters, but often we do not know that yet that is why we are doing the clusters.

So, what is the first step in the clustering algorithms the first step is to take each data points and see, which cluster center is closest. So, let us say we take this data point 1 clearly the yellow cluster center is closest and at least hope for the set of data points is yellow is closest. Obviously, for another set it looks like the green is closest, so each data point basically gets an assignment either gets assigned the yellow color or a green color and that is essentially, what I have done. This diagrams are more conceptual not the scales.

So, I kind of high balled it and said it looks like this 4 data points are close to the yellow that is this 3 this 4 are close to the green, so that is how they have been assigned. What is the next step? The next step is given this assignment the new cluster center for the yellow data points is the centroid of the yellow. So, what is the centroid of the yellow then, so you can think of it as the essentially the average of the yellow and average; obviously, needs to be on this axis as well as it needs to be on this axis.

So, where you would centrally places yellow, so that is you know minimizing the squared deviation, which is Euclidean distances to each data point and that is the definition of centroid. And, so we move the yellow circle to be the new centroid and it is probably going to come somewhere here and the green circle, now the green circle has a little bit of problem and it just cannot move to the center of this space out here. Because, it is; obviously, has an assignment.

So, this is going to bias where the green moves, so perhaps it will move somewhere here and that is what I do in the next step actually move the yellow and the green. But, once I move it is like the assignments are completely lost you forgotten the assignment, because remember that the assignments were made with the old centroids.

Now, with the new centroids you just have the new cluster centers and no assignments you redo the process when you redo the process you see now, which blue dots are closest to the yellow and which ones are closest to the green and we repeat the process and as you can see we have already gotten pretty good results out here with the yellow and green classification, which matched our intuition about what the two clusters of this graphs was.

Clearly in the next step for instance is green will move into this cluster center the assignments themselves won’t change too much perhaps the yellow will move little bit out here. But, essentially after maybe one or two more steps the centroids will stop moving your assignment will probably the same. So, even if you stop the 1 or 2 stages earlier you would have gotten the clusters that you were interested in.

So, I hope this gives some idea, what the K means algorithms is, now another very popular prototype based approach, which can work and go beyond K means in some senses, which can go beyond a quantitative attributes. Because, remember for you for this whole algorithm to work you had to able to be compute distances and it is makes senses that you attributes x_1 and x_2 for quantitative and continue straight x_1 and x_2 there is a little there is a medium there is more and this is continuous quantitative variable, now many times you do not have that.

So, very useful alternative is K medoids in that case, now K medoids allows you to go beyond the quantitative variables. So, when you have categorical or rather more importantly nominal variables, where the variables are things like male, female and the attributes is like male female or things like that you can use K medoids. But, more importantly K medoids allows you to go beyond attributes altogether, where x_1 and x_2, where attributes to a point, where all you need is some dissimilarities matrices.

What do you mean by these dissimilarity matrices? What I mean is you have all this data points, let us start calling them 1, 2, 3, 4. Now, with K means the distances between and I am going give you the same data points in the columns, now with K means I can compute the distances between the data points 1 and data point 2 through some form of Euclidean distances and mark a value of x. But, what if, so the whole process of using Euclidean distances require that I go into each attributes look at how different it is and then compute that square distances takes the square roots.

But, what if I did not have that process, what if I had a process, where I gave you just the differences between each data points. So, I have something like a dissimilarities matrix, so when I have a dissimilarities matrix, that I am giving you and you can get the dissimilarities matrices through completely differences ways it can be user survey or it can be something extremely subjective, where you say I feel like 1 and 2 are different by this much and I can tell you how different 1 and 2 are and I can tell you how different 1 and 3 are you know and I can tell you how different one and four are and so on.

But, I cannot really break it down into five different attributes and give it to you that way. So, people do this lot of times in social science research whether some kinds of a question I have that is given to people to tell, so please tell me how different this two are. And people are able to say that there are not able to break it down into a set of quantitative variables and say how different they are in each of those dimensions. So, K medoids is very useful when you do not have these attributes, but you just have dissimilarities matrices of how each data points is different from every other data point.

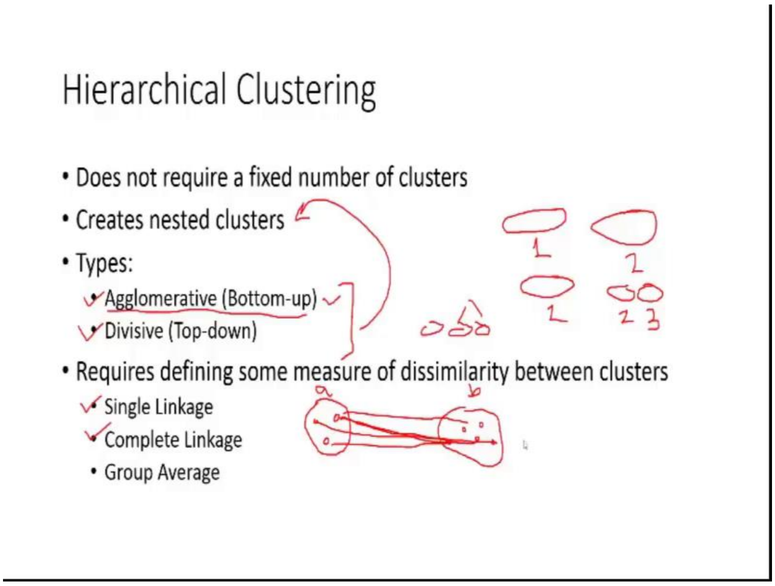
So, you will essentially just need some kinds of leading diagonal of data. So, the data either on this side or data is on that side, because what is different from two the same amount that two is different from one there is no different between those two. So, in cases like that K medoids primarily winds up not having this arbitrarily cluster center, because you really cannot any longer compute things like centroids there is no centroids there are no dimensions on which.

So, what winds up happening is you nominate A data points to be the prototype and you use the same core concept of K means in that you first nominate a data point and then, you do an assignment, where each data point chooses between the nominated data points. And once you have one assignment you find the new nominated medoids; that should become the representative data point. It is kind of like the idea of being the cluster center. Now, while this works an important point is that the computational intensity associated which such an approach.

Now, when you had 10 or 20 data points which had an assignment, so let us take all these data points. The computing the average through sum of square just was essentially like using sum of square minimization was essentially computing an average of this points across each of this dimensions. Now, you do not have something like that with K medoids once you have an assignment to choose which, for this assignment, what should be the prototype requires you to see the dissimilarities between each data point to each other.

So, the number of computations that you need to assess really grows. So, K medoids is often seen as a very computationally intensive approach. So, this should give you some idea about K means and K medoids.

(Refer Slide Time: 19:58)



Hierarchical Clustering

- Does not require a fixed number of clusters
- Creates nested clusters
- Types:
  - Agglomerative (Bottom-up)
  - Divisive (Top-down)
- Requires defining some measure of dissimilarity between clusters
  - Single Linkage
  - Complete Linkage
  - Group Average

The next approach that we are going to talk about is hierarchical clustering with hierarchical clustering you do not really windup fixing fix number of clusters. And essentially the approach itself starts with either one very large cluster and breaks it down step by step, so the first one, so one way of doing it is called the divisive way, which is you have one large cluster, which has all the data points in it is and then, after that I break it in to 2 clusters and then, after that I break it in to 3 clusters.

So, I look at the two clusters that were broken up as choose, how to break them down further. So, and that is called the divisive approach you also have the other approach, which is agglomerative and definitely the more popular approach, which is bottom up, where each data point becomes a cluster, so you have all these. So, you have the number of actual cluster you have is equal to the number of the data points, because of each data point is a cluster.

And then, after that you choose the two closest clusters and merge them together and we will talk about how closest is defined. So, you choose the two closest clusters and merge them together. So, the end of the first step you are essentially having n data points you having n - 1 data points, so the end of one step and after that the next step you have the n - 2. Because, now you have n - 1 cluster where all of them expect one have one data point and one clusters has two data points.

But, you are just treating them as clusters and you are saying, now order this n - 1, which are the two closest clusters that I can merge and that is seen as bottom of approach. Now, because of how we do this essentially you are going to have nested clusters. Think of this divisive approach if you created two clusters. Now, when you go to create three clusters in this divisive approach you are going to take either cluster 1 or cluster 2 and break it further and the others is going to be same, so one break up could be that the cluster 1 stay the same and cluster 2 gets broken up into two pieces.

So, in many ways the clusters that you are creating are nested within one another you can think of them is parent and daughter. And this same of approach goes for the agglomerative, where you started with many individuals clusters and you choosing to group them. It is never like you are breaking one grouping you rethinking an action that you did in the previous step you never in some senses going back in time and recorrecting a decision to either group clusters or break clusters, so in some sense you can think of this also partly greedy approach.

Now, we spoke about, how we are going take in with especially agglomerative, which is what we focus on the rest of lecture, because it is the more popular one and for very specific reason that we will talk about you need to take in the first step. For instance you have n data points the n clusters and you need to take two closest clusters and merge them, how are you defining closest clusters. And the way you would kind of define them is through some measure of dissimilarity between the clusters.

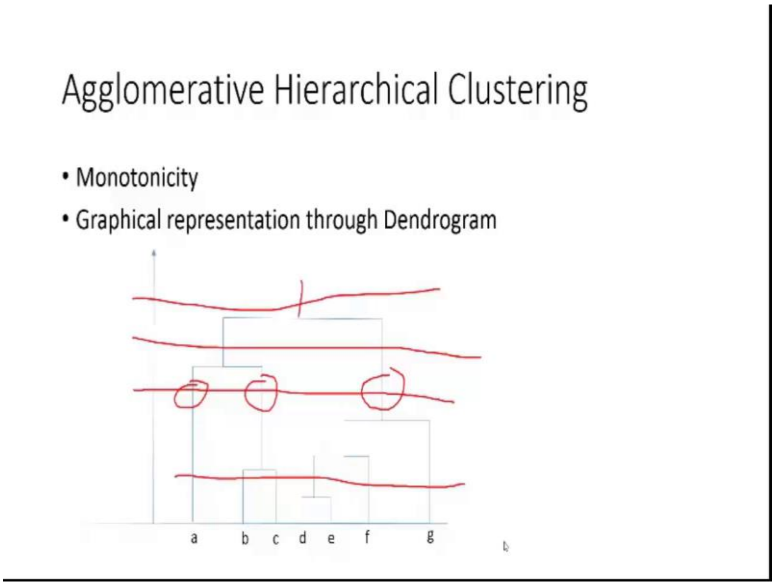
An example is for instance that there are many definitions of the dissimilarity one the three of them that are listed here really talk about it more in terms of come more from graph theory. And the first one is called single linkage and the idea behind single linkage as the means of talking about dissimilarity is that it is essentially when you take two clusters you take the minimum distances between two data points that can be in the two clusters.

So, do not think of it as much in terms in the first step with in each cluster there is one data points, so it is not a very interesting case. Now, think of a case, where you got this cluster and you got three data points; that is what I have shown here and let me give you concrete example, so here is one cluster and there are three data points within this cluster. Here is a cluster, four data points five data points. Single linkage basically takes each combination and sees, which combination is minimum, so in this case probably this would be minimum.

And, so you know it is for that reasons it is actually called min as a definition of dissimilarity. Complete linkage in contrast takes the maximum this is which, of this combination I need you take one from cluster A and you take from cluster B, which combination of points can I take get the maximum distances and I am probably guessing that this connection that is just I drew would have the maximum distances this group average, which basically takes every combination of every point to every other point and takes the overall average.

Now, in addition to that you also have some approaches that try the more prototype based ways and there you would for each cluster try to create a centroid and from that centroid you look at the distance from one centroid to the other and so on. And there are some other approaches there is a there is a wards method, so on, where again trying to take some kinds of a more prototype based approach to define the dissimilarity. But, essentially hierarchical clustering it is all of these still come under broad umbrella of hierarchical clustering.

(Refer Slide Time: 26:09)



Agglomerative Hierarchical Clustering

- Monotonicity
- Graphical representation through Dendrogram

So, there are two important things in connection with the hierarchical clustering as you can see, because it is doing this nested clustering, where you start from with agglomerative you start with each data points being a cluster and you keep merging them till you have this one mega cluster of all the data points. And in the other way around with this divisive you start this one mega cluster with all the data points and keep breaking it till each data points is it is own cluster.

So, therefore, you have not really committed to creating clusters of a specific size K and you can therefore, take you can kind of look at the results over all and make it make a decision in terms of what here size should be. One important property that we see with all agglomerative when you know in some divisive methods is that they possess this property called monotonicity. And the idea here is that dissimilarity between merged clusters is monotonically increasing with the level of merger and that should be fairly intuitive. When you think of the dissimilarity of a cluster which has just one data points there is no dissimilarity it is only when you have two data points you can say this two data points at different by this much. So, with agglomerative clustering in some senses the more number of data points that you have in to the clusters the greater of the amount of dissimilarity there and that monotonicity is strictly maintain with agglomerative clustering.

So, a very useful way of representing it therefore, especially when you have this monotonicity you can graph quickly represent the agglomerative clustering through something called the Dendrogram and what is shown on this slide is Dendrogram and it is essentially this binary tree, which is plotted. So, that the height of each node is proportional to the value of the dissimilarity between it is daughters. So, what is node here? Essentially you can think of each partition is being a node and the height of this node.

So, let us take something very simple height of this nod as to do with this height as to do with the degree of dissimilarity between b and c between see you are forcing b and c to be in a cluster now that is what you doing by this part of the graph that is what it is doing. This height as to do with dissimilarity of b and c, which is why for instance perhaps d and e was merge first. So, with agglomerative clustering think that you are going bottom up and you are sequentially making decisions.

And, so if you choose to put d and e together first; that means, d and e would have been less dissimilar to each other than b and c and that is why d and e was done first and then, b and c was separately done perhaps later. Now, the really good thing about this dendrogram is that, now you have a full picture you can, now choose to say I am interested in this situation where there are three clusters and you can essentially draw this horizontal line. And you have the three clusters cluster 1 is just data point a cluster 2 is the data point b and c cluster 3 is data point d e f and g you basically see what goes in each of this limbs and that is that is essentially your cluster.

So, at any level when you draw a horizontal line you can the number of vertical line cuts across is the number of clusters that you have in your thing. So; obviously, if you had a line out here you drew it here this is 1 cluster and here it is the 2 clusters does, so, on. So, this should hopefully give some idea about the two very popular algorithms K means clustering and hierarchical clustering.

Thank you.