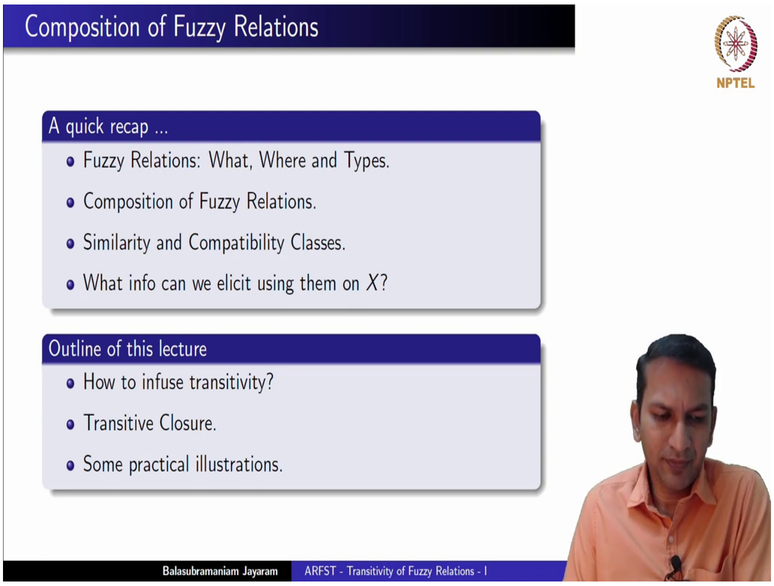
Approximate Reasoning using Fuzzy Set Theory
Prof. Balasubramaniam Jayaram
Department of Mathematics
Indian Institute of Technology, Hyderabad

Lecture - 26
On the Transitivity of Fuzzy Relation - 1

Hello and welcome to the 4th of the lectures in this week 5 of the course titled Approximate Reasoning using Fuzzy Set Theory, a course offered over the NPTEL platform.

(Refer Slide Time: 00:38)



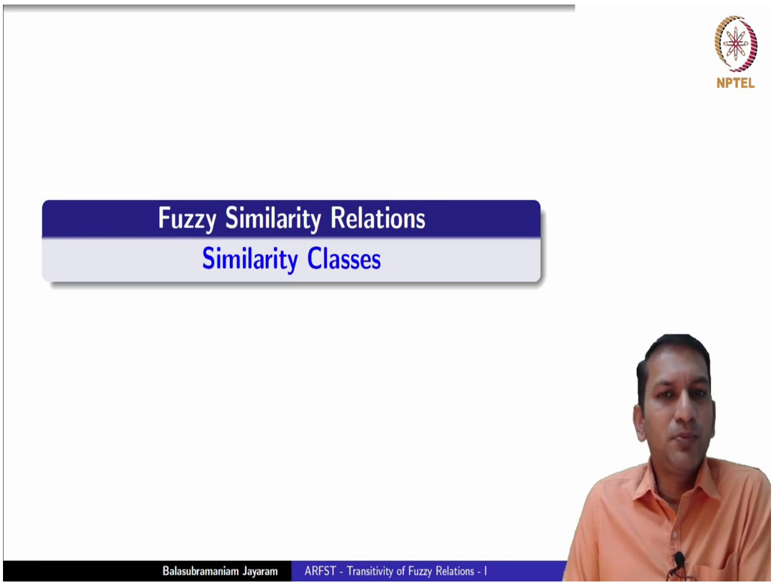
So far this week, we have seen what fuzzy relations are, where they can arise, and a few different types of fuzzy relations. We have also seen how to compose fuzzy relations. We have seen here there are many possibilities and interpretations. We have also seen that from the similarity and compatibility relations, we can extract similarity and compatibility classes. And we have seen the utility or usefulness in the last lecture.

It is our quest using these relations, special relations, we would like to unearth some interesting information about the elements of X itself. In this quest, we will continue and we have seen that transitivity plays an important role if you would like to actually come up with some kind of a partition on the underlying set X. If you would like to group objects based on their similarity levels, then transitivity plays an important role.

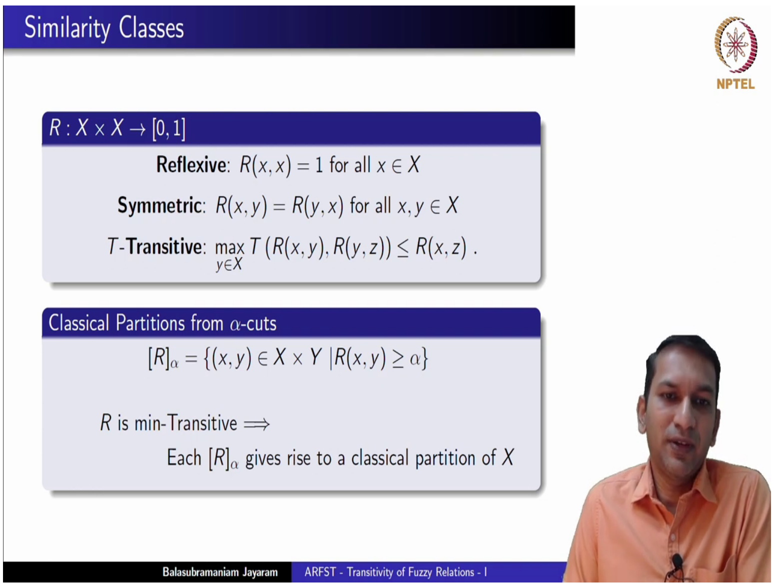
In this lecture, we will see how to infuse this transitivity to a given fuzzy relation. We will define what is a transitive closure of a fuzzy relation with respect to a t-norm T. And we will also come up with a very simple algorithm to actually calculate or find determine this transitive closure.

We will end with a few practical illustrations of what we have seen in the previous class, previous lecture and also in this lecture.

(Refer Slide Time: 02:23)



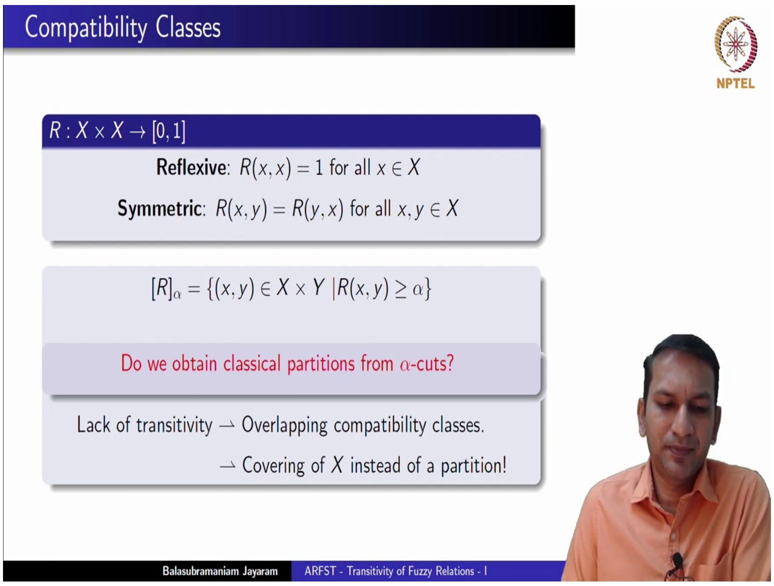
(Refer Slide Time: 02:28)



Let us begin with a quick recap of what similarity classes are. They are fuzzy relations which are reflexive, symmetric, and transitive. We know that because of the transitivity if we consider the alphabets if the fuzzy relation is min transitive, then we can obtain partitions on the set X for different resolutions of alpha.

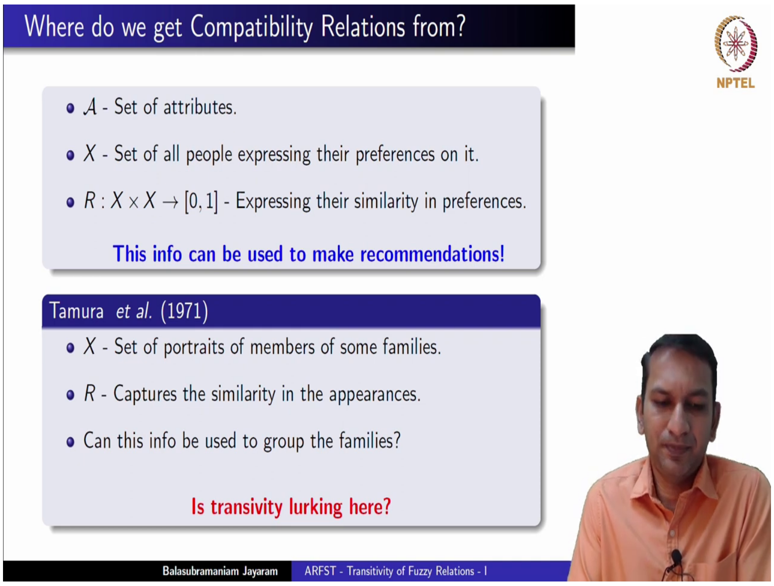
(Refer Slide Time: 02:53)

(Refer Slide Time: 02:55)



Compatibility classes are like similarity classes except that they do not have transitivity. And we found that this was a singular reason why we were not able to obtain partitions from the different alphabets, instead we were only obtaining overlapping compatibility classes. Of course, they were covering the set X, but not partitioning it.

(Refer Slide Time: 03:19)



We stopped with this question in the last lecture where do we get these compatibility relations from. Let us look at least two scenarios. First of them let A denote a set of attributes. An X, a set of all people expressing their preferences on this set of attributes. You could think of the set of attributes as probably just a set of movies and people giving their preferences.

Now, we may construct a fuzzy relation on X binary fuzzy relation on X which expresses the similarity between the preferences of the different people. Now, note that even in the example that we are given of ranking movies or giving preferences from a set of movies, we know that this is the kind of information that can be made use of to make recommendations.

There was another interesting work done by Tamura and others, way back in 1971 in which they took a set of portraits of members of some families and they wanted to capture the similarity in the appearances between any two photographs. Now, these similarities themselves that cannot be generated automatically. That means, humans human subjects were involved in it and they were given pairs of them and asked to measure the similarity between these two pictures. This is how they had constructed the similarity relation R.

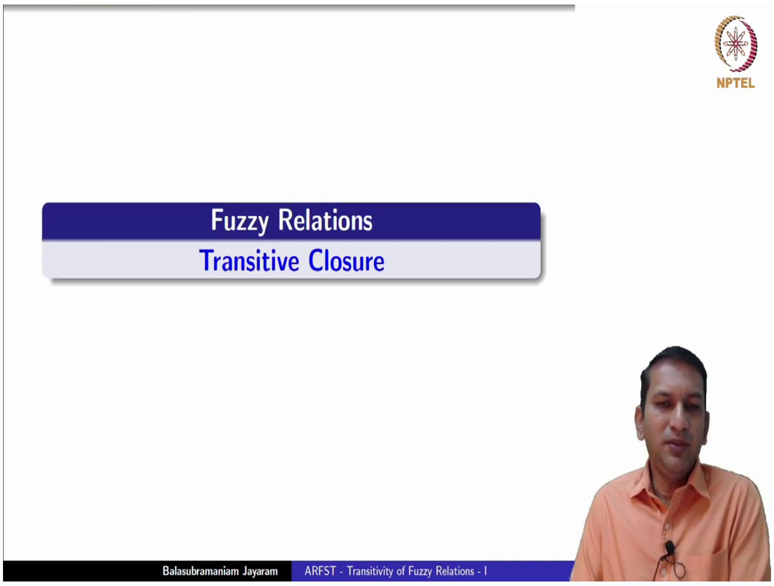
It is clear now that it is way too much to expect this relation to be transitive also. Clearly, it is symmetric, because it does not matter which way you are comparing a pair of photographs.

But it is unlikely to be transitive, and especially min transitive. However, what they wanted to do was they wanted to make use of this relation and see whether the portraits could be grouped into families. Now, what give them this confidence or where are they basing this reason on? Is transitivity lurking here? Yes, definitely.

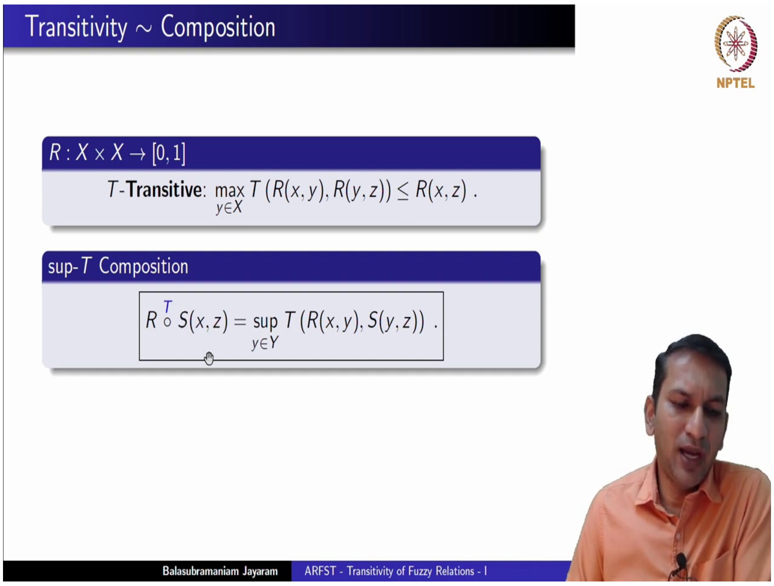
They know that the parents, the father and mother, they may not look similar, however, their similarity perhaps is related to through their offspring's. So, this is how their similarity can be thought of as being transitive through their offspring's.

So, you see here in these two scenarios that we have, we have actually extracted relations which are fuzzy relations and which are unlikely to be transitive. However, they are symmetric and they could easily be made reflexive. For instance like, when we are talking about preferences, take two people who are given preferences on the same set of attributes, if x and y are same then the preference level matches completely and you could give a similarity value of one.

(Refer Slide Time: 06:33)



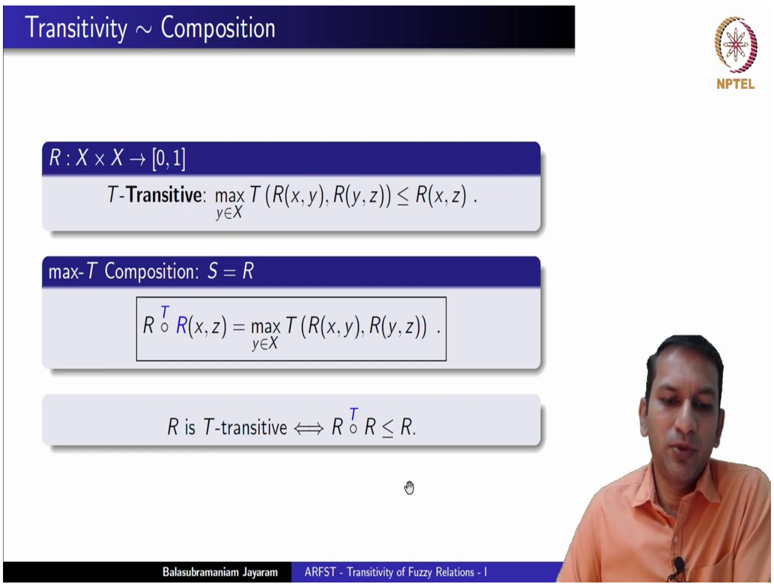
(Refer Slide Time: 06:37)



Well, let us move on. Let us look at what is transitive closure. Before that the concept of transitivity itself is related to some kind of composition. Let us look at what is T-transitive.

So, we said that a relation fuzzy relation is T-transitive if this inequality holds. But now look at the sup-T composition. So, when R is on X x Y and S is on Y x Z, we compose R and S like this. If instead of taking S to be from y cross z, if you instead take S to be R itself what we have is essentially, this composition coming in here.

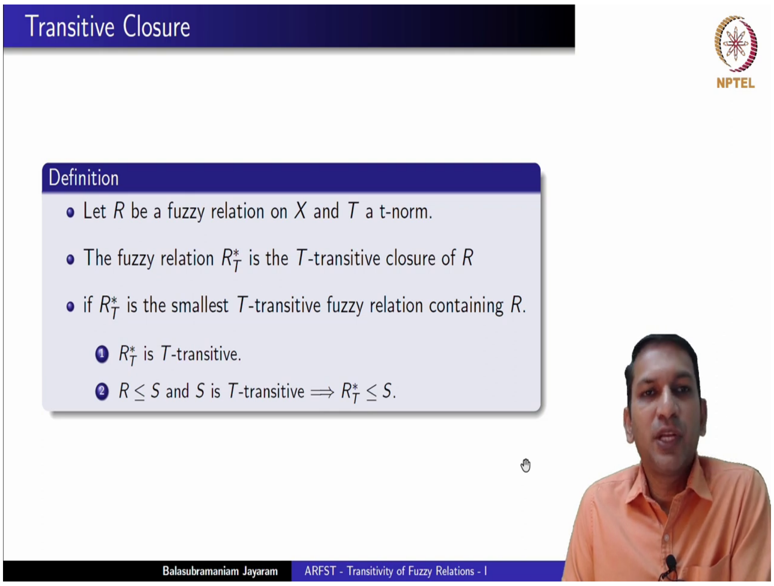
(Refer Slide Time: 07:16)



So, from these two we could also define transitivity in an alternate, but equivalent way. We say R is T-transitive if and only if R circle T R is less than or equal to R. So, you see here the transitivity of a fuzzy relation with respect to T is in fact, related to the sup-T or max-T composition. Let us for the moment assume that X is only finite, and so it is easy to take supreme to the maximum.

Let us define what we mean by transitive closure.

(Refer Slide Time: 07:56)

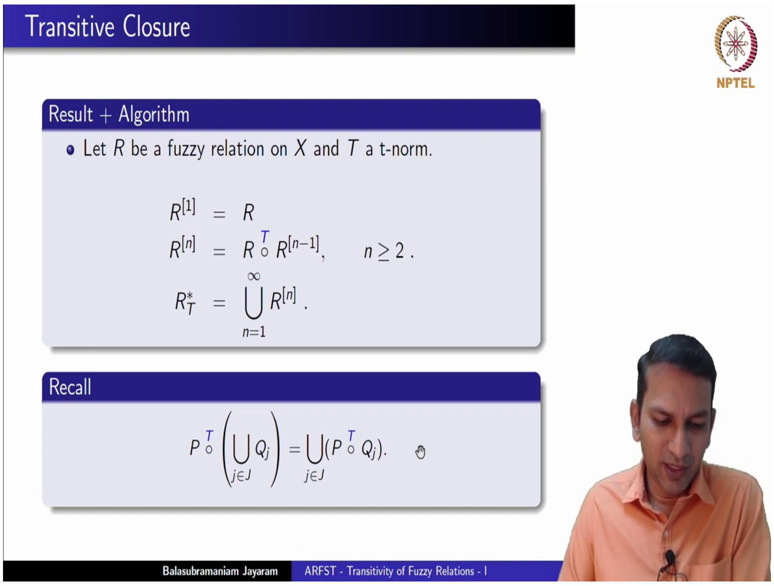


For that let us consider a relation, fuzzy relation R on X and T a t-norm. We say the fuzzy relation denoted by R_T* is the T-transitive closure of R if R_T* is the smallest T-transitive fuzzy relation containing R.

Now, let us decode this a bit. So, what do we expect R_T* to be? First of all R_T* should be T-transitive. And if S is some relation which contains R and is also T-transitive, then we expect that S is bigger than R_T*, sorry R_T* is T-transitive and the smallest such transitive relation T-transitive relation containing R.

Well, this is the definition. But how do we actually find the transitive closure of a fuzzy relation R with respect to a t-norm T.

(Refer Slide Time: 09:07)



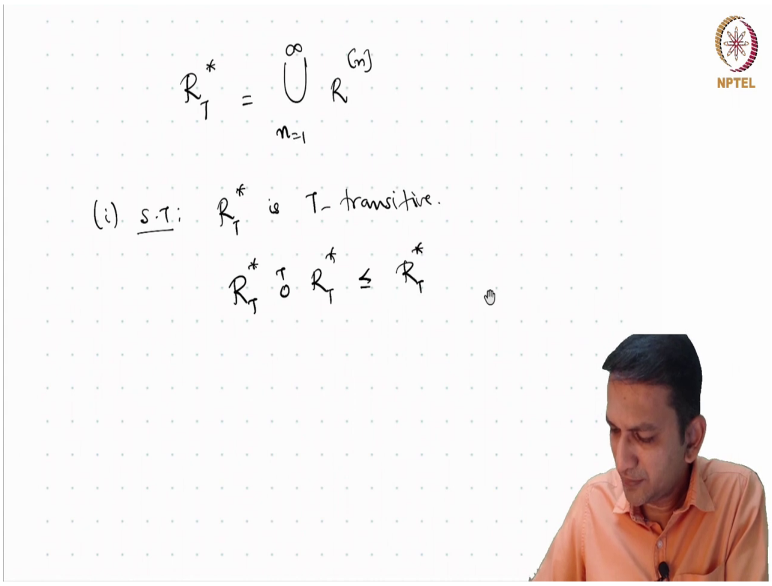
Fortunately, we have a result and it is not only a result it is also an algorithm. Let us be given a fuzzy relation R and a t-norm T.

By this notation R^[1], we define, we denote the relation R itself. We recursively define what is R^[n], it is nothing but R composed with R^[n – 1] with respect to the sup-T composition, look T is fixed here, sup-T composition. So, R^[2] is nothing, but R circle R. R^[3] is R circle R^[2] which is R circle, R composed with R composed with R.

Now, what is the transitive closure of an R with respect to T? It can simply be written as the union of all such recursive compositions. That means it is R union R^[2] union R^[3] so on and so forth.

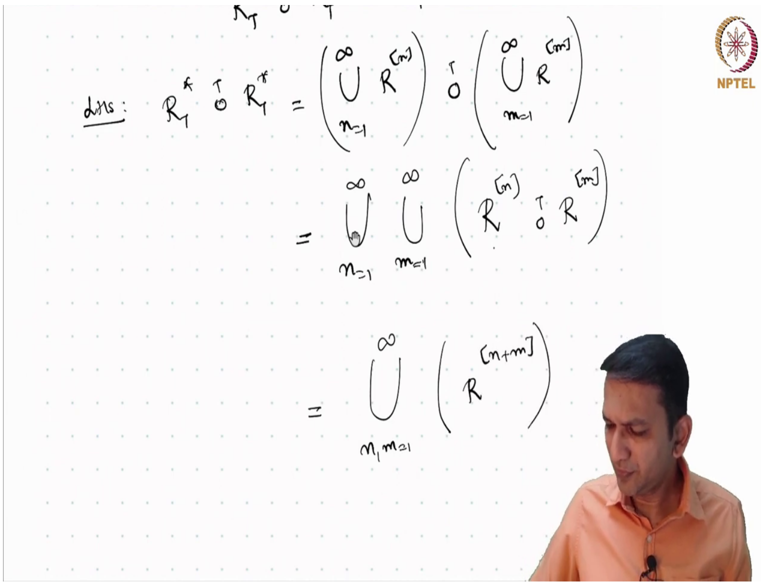
Now, the formula looks very simple, but it really does the job for us. That means, R_T* is T-transitive and in fact, it is the smallest T-transitive relation containing R.

(Refer Slide Time: 10:30)



Let us look into the proof of this. What we want is, what we are given is R_T* is defined like this. We need to prove two things. Firstly, show that R_T* is in fact, T-transitive. What does this mean? This means in terms of the composition R_T*, circle T, R_T* is in fact less than or equal to R_T*. Let us star with this.

(Refer Slide Time: 11:06)

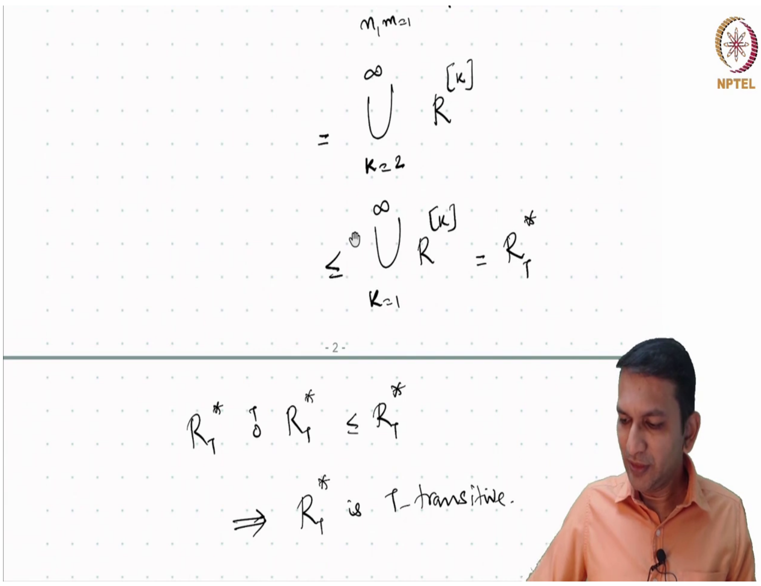


Let us start with the LHS. What is R_T* composed sup-T composition with R_T*? It is nothing, but union n is equal to 1 to infinity R^[n], sup-T composition with union, let us use a different index here R^[m]. Now, what is this? You might recall from the previous lecture that if we are discussing sup-T composition with relations appropriately defined and j being the index here, we see that P circle T union Q_j is nothing, but union of P certainty Q_j.

That means, we are able to pull out the union operation which is essentially the max operation here outside of this bracket. This once again follows because T and max distribute to each other. As I said, we will have an entire lecture discussing different functional equations, at that time we will also visit distributivity of fuzzy logic connectives or (Refer Time: 12:23).

So, using this property, we could write we could pull out both of them and write this as union n is equal to 1 to infinity meaning m is equal to 1 to infinity, R n is equal to R n. But now, this could also be written as union n comma m is equal to 1 to infinity. Note that by the recursive definition that we have here, this is nothing but R power n plus m.

(Refer Slide Time: 13:05)

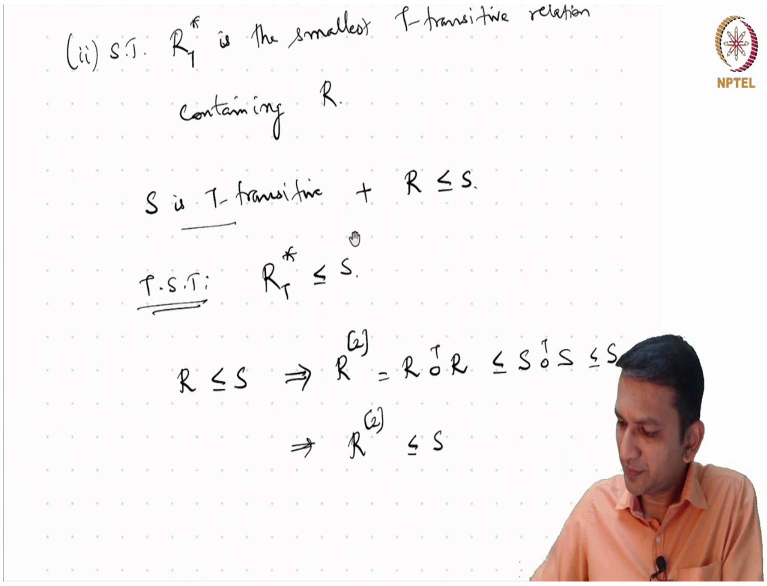


Now, as n and m vary from 1 to infinity, what we get is union k is equal to 2 to infinity R^[k], because n and m they start from 1, which means they start with 2. So, k is equal to 2 to infinity.

Now, we know that this is definitely less than or equal to union k is equal to 1 to infinity. But what is this? This is just R_T*. So, what we have shown is that R_T sup-T composition that itself is in fact, less than or equal to R_T* which implies R_T* is in the T-transitive.

Now, this is the first part. We need to prove also that it is the smallest such transitive relation containing R. So, we need to prove so, to this end.

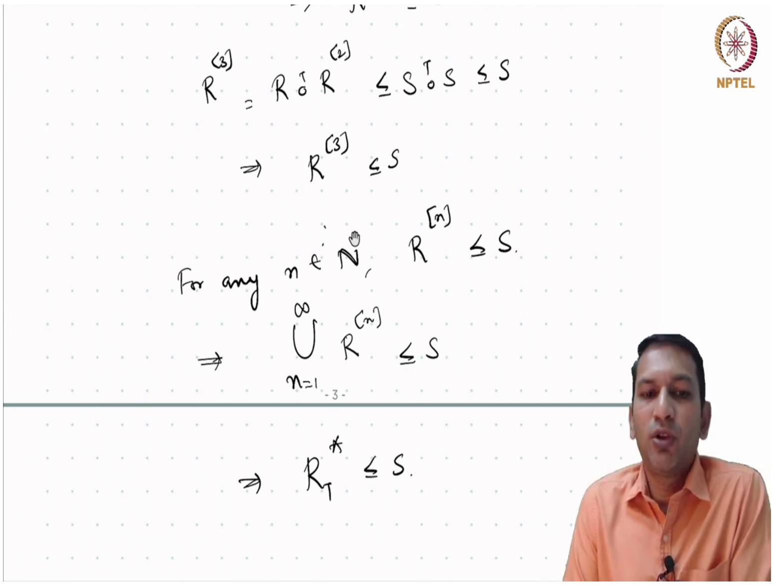
(Refer Slide Time: 14:37)



Let us consider an S which is T-transitive plus R is less than or equal to S. We need to show that R_T* in fact, is smaller than S. This is what we need to show. So, that means, it is smaller than any other T-transitive relation containing R.

Now, R is less than or equal to S implies look at R^[2]. What is R^[2]? That is R circle T R, but we know by the monotonicity of this composition that this is actually smaller than S circle S. But since S is T-transitive, we know that this is less than or equal to S. That means, from here we get R^[2] is in fact, less than or equal to S.

(Refer Slide Time: 15:36)

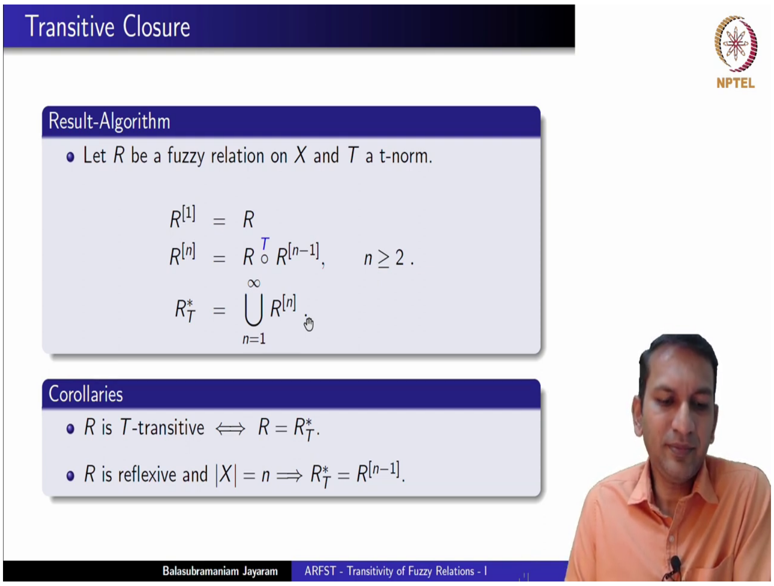


Now, what happens to R 3? R 3 is nothing, but R circle R 2, but R is less than or equal to S and R 2 is less than or equal to S. So, this is less than or equal to S circle T S, but which is again less than S because S is T-transitive. So, this implies once again that R 3 is also less than or equal to S.

Now, if you go on like this for any n, we see that R of n is in fact, smaller than S. Not only for R, for every n this happens. That means, this implies even if you take the union, this will lie below S, means it is smaller than this. But what is this? It is nothing, but R_T*.

So, as I was to be proven if there existed an extra S a relationship which was T-transitive and contained R, we have shown that the T-transitive relation that we have come up with R_T* is in fact, smaller than S. Which means, it satisfies both the properties that is expected of transitive closure. And thus, R_T* is indeed the transitive closure of R.

(Refer Slide Time: 17:04)



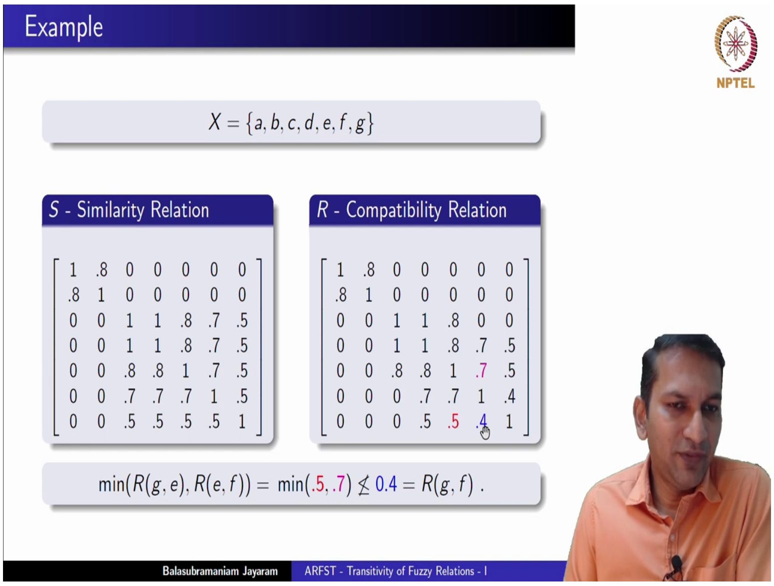
Now, we can come up with some interesting corollaries based on this result. As mentioned it is not only a result, it is also an algorithm because it allows us to actually determine the transitive closure.

Now, it is clear that a relation is T-transitive if and only if it is equal to its transitive closure because that is the smallest transitive relation containing it. And what is interesting is, one might wonder how long should we go on because we have to find R, then R 2, then take the union, again take the union with R 3, so on and so forth.

But interestingly if R is a reflexive relation, and if X is finite say it has only n elements, then it can be shown that R_T* is in fact, R power n minus 1. In fact, it depends on the value is there R_T* can actually be much smaller than that. It is just that there will definitely exist a k less than or equal to n minus 1, such that R_T* is in fact equal to R power k.

So, in the case of reflexive relations with a finite domain underneath, we see that this is a process which is not never ending and quickly converges at most in finite steps.

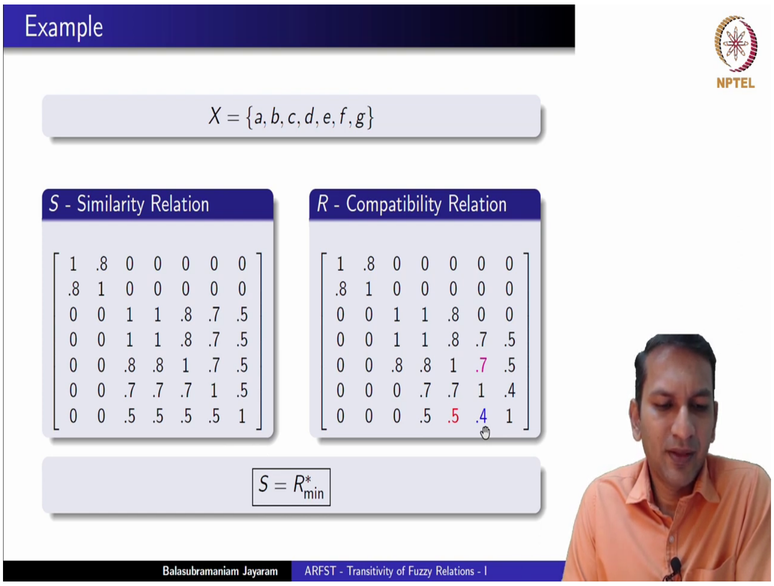
(Refer Slide Time: 18:28)



In the previous lecture, we have taken this set X consisting of these 7 elements and we have looked at two different relations on them. On the left norm, what you seen is a similarity relation. It is reflexive, symmetric, and min transitive. On the right, we have only a compatibility relation. We in fact showed that this is not min-transitive.

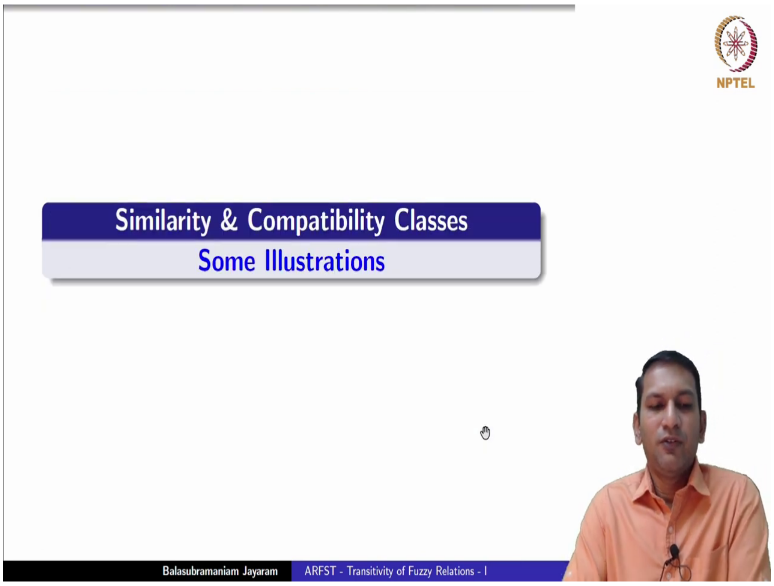
For instance, if you look at the colored numbers, you can see that minimum of R of g which is 0.5 and minimum of R of g, e and R of e, f which is 0.7 is in fact, 0.5 and not less than or equal to 0.4 which is R of g comma f. So, we said that the relation on the right R is only a compatibility relation which is not min transitive.

(Refer Slide Time: 19:30)



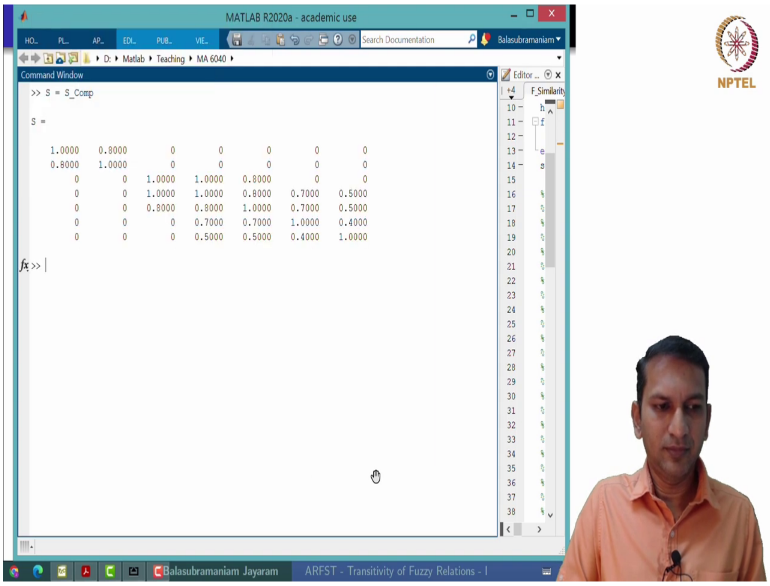
It might interest you to know that the relation on the left S is in fact, the transitive closure of R with respect to min obtained by this vary procedure that we have listed out.

(Refer Slide Time: 19:43)

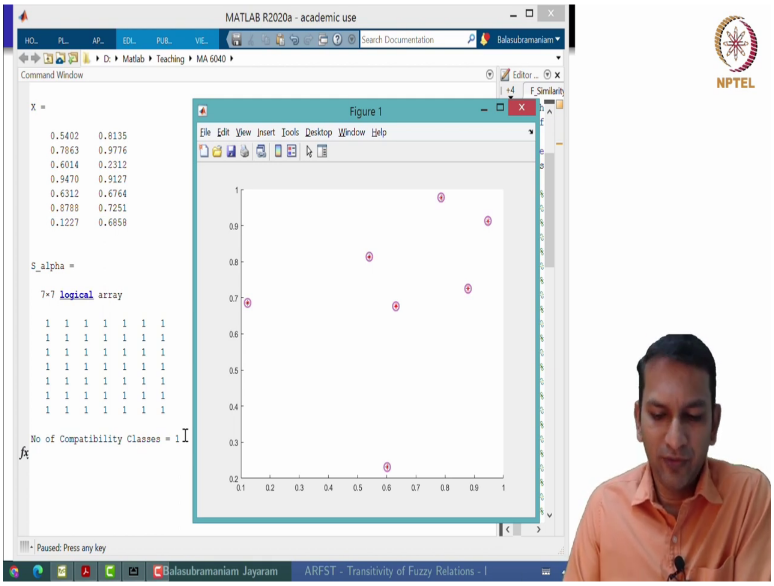


Well, now for some practical illustrations, I would like to show you through some simple MATLAB code, how you could visualize this. Well, let us look at some examples now.

(Refer Slide Time: 20:09)



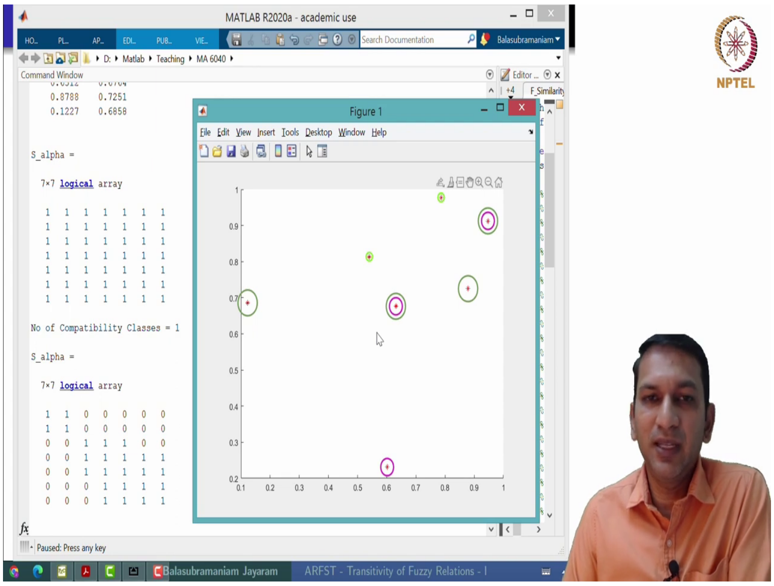
(Refer Slide Time: 20:28)



Consider the same relation that we have discussed in the previous lecture. Now, you will see this is the compatibility relation that we have taken now. Let us look at how to generate the corresponding similarity compatibility classes.

So, we have 7 elements a, b, c, d, e, f, g that is how we have listed them yesterday. However, on the plane here we have 7 points. These are the coordinates of these 7 points. And now if you consider for the alpha at 0.4, we know that all of them the indicator matrix consists of once everywhere and hence there is exactly one compatibility class. All of these 7 elements fallen (Refer Time: 21:04). So, as you can see the number of compatibility classes is actually equal to 1.

(Refer Slide Time: 21:17)

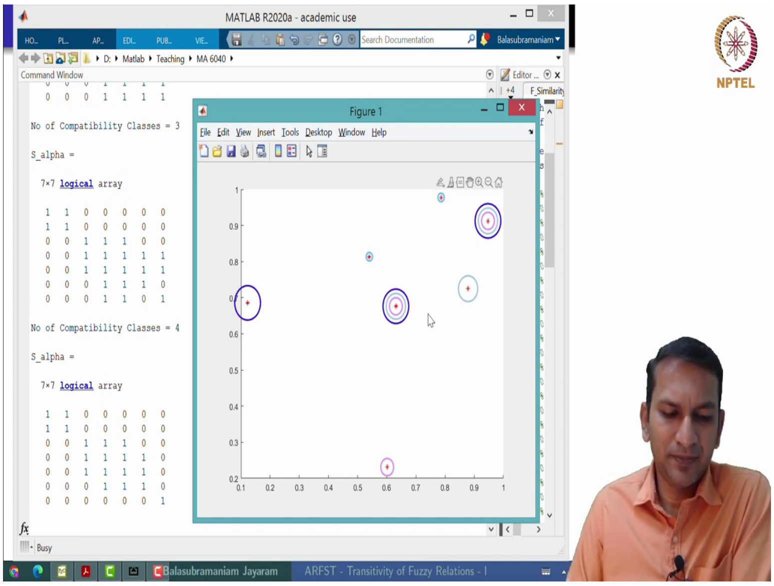


Now, let us take this next alpha cut. In this case alpha is equal to 0.5. Now, you see, we see that this is exactly the indicator matrix that we have got, the alpha cut of the relation R which is only compatibility relation. We have seen yesterday it does not have a pure block submatrix structure. However, in this case first we have a block here, a and b are related to each other and that is what gives rise to the first compatibility class, as pointing (Refer Time: 21:42) to elements here.

Next, we saw that these 3, that is c, d and e they form a compatibility class. So, here we see that these are the 3. Now finally, we have seen at this level of alpha that alpha is equal to 0.5, we see that there is an overlapping block here. That means, this compatibility class consists of d, e, f and g. Already d belongs to another compatibility class. It also belongs to this compatibility class of d, e, f and g.

And now that is what is indicated here. You see that this element is related both to the magenta class and also to the green comparability class and so, is the case with this. Because we had earlier c, v, e and now d, e and e both of them belong to two different comparative compatibility classes. So, as was mentioned yesterday, these compatibility classes do not form a partition, in fact, they are overrun. Of course, they form a cover.

(Refer Slide Time: 22:50)

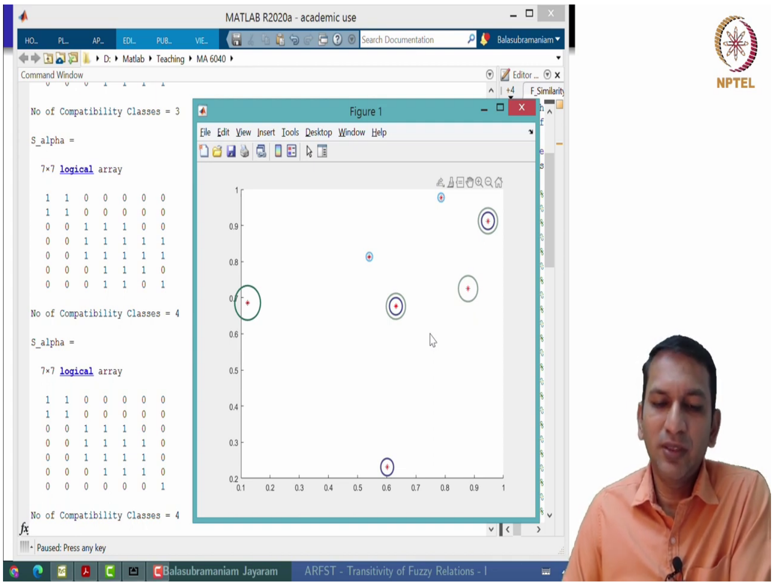


Now, let us look at a different resolution. So, at alpha is equal to 0.5, we have 3 compatibility classes.

Now, this is the alpha cut for alpha is equal to 0.7 like this. Once again you will see that these two elements are actually staying together now. Now, you see that again here you have c, d and e staying together. However, now things get little different. Now, you will see that these 3 that is d, e and f, they actually form a compatibility class, d, e and f. So, you see in d and e here, they belong to two different compatibility classes.

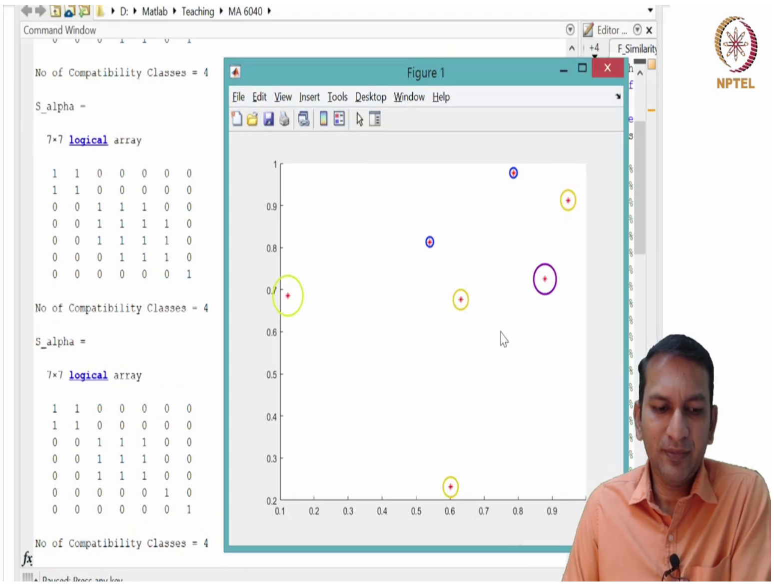
And finally, we also see that d, e and g, they in fact are related under this compatibility relation. And in the fourth compatibility class what you see is both d and e belong. So, now, you see these two elements they have 3 circles around them at this level. That means, they in fact belong to 3 compatibility classes. Let us go one level further up.

(Refer Slide Time: 24:02)



We see here that a and b stick together. Once again c, d, e they stick together. And now if we go further like this, we see that we have 4 compatibility classes, but now d and e belong only to 2 of them, not to the 3 of them as it did earlier.

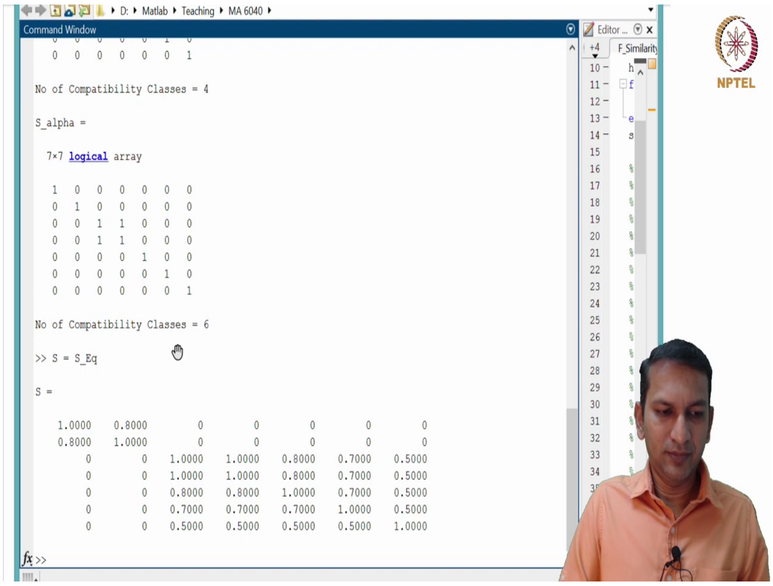
(Refer Slide Time: 24:32)



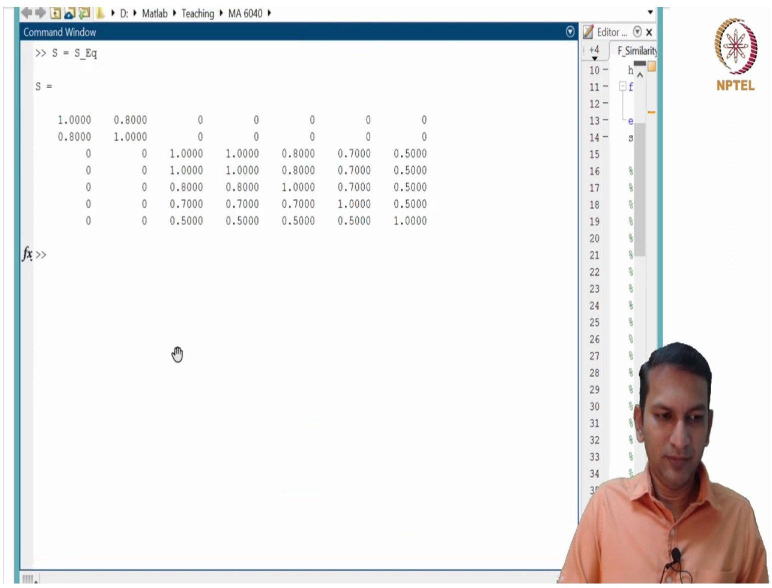
Finally, if this is at alpha level 1. We see that we have these 2, these 3, this is one separately and this is one separately. Of course, when alpha is equal to 1, what we get is a classical equivalence relation. So, we actually get a partition here.

So, now, look at the similarity relation that is given here, is exactly the similarity relation that we have here which is obtained as a min transitive closure of this particular compatibility relation.

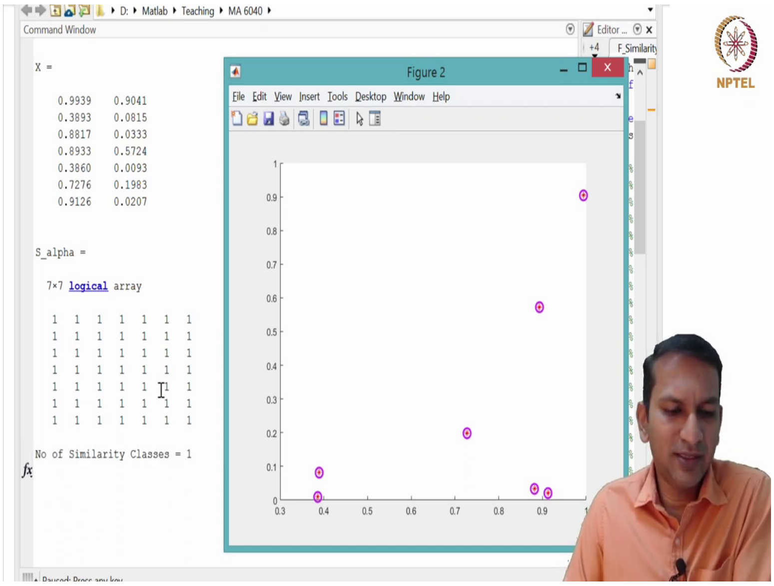
(Refer Slide Time: 25:10)



(Refer Slide Time: 25:18)

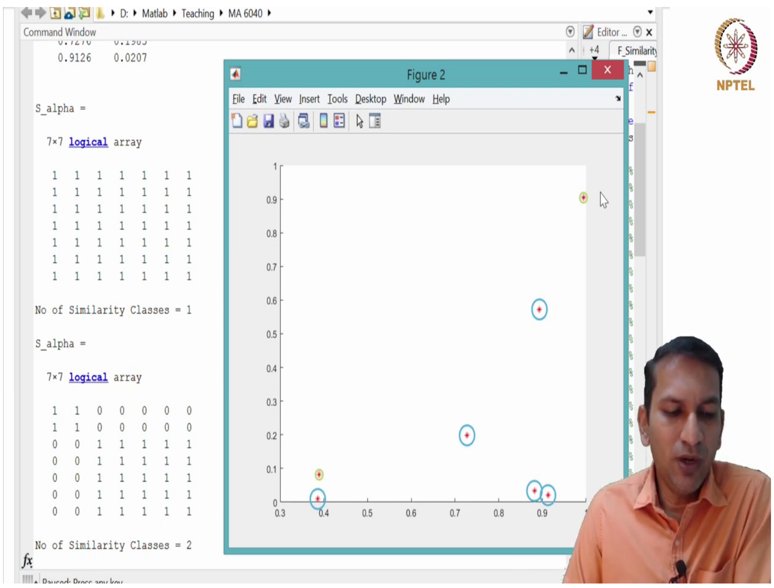


(Refer Slide Time: 25:28)



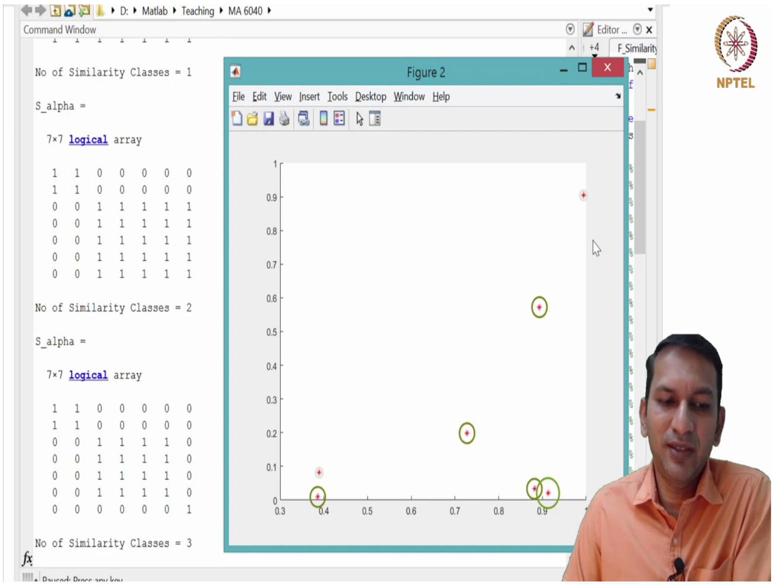
Let us run. Let us do the same thing as we did before. Now, you see here once again you have the 7 points are arbitrary generated on this plane. And at the level of alpha is equal to 0.5, we know that all of them are related to each other, and so, you have a single singularity class.

(Refer Slide Time: 25:54)



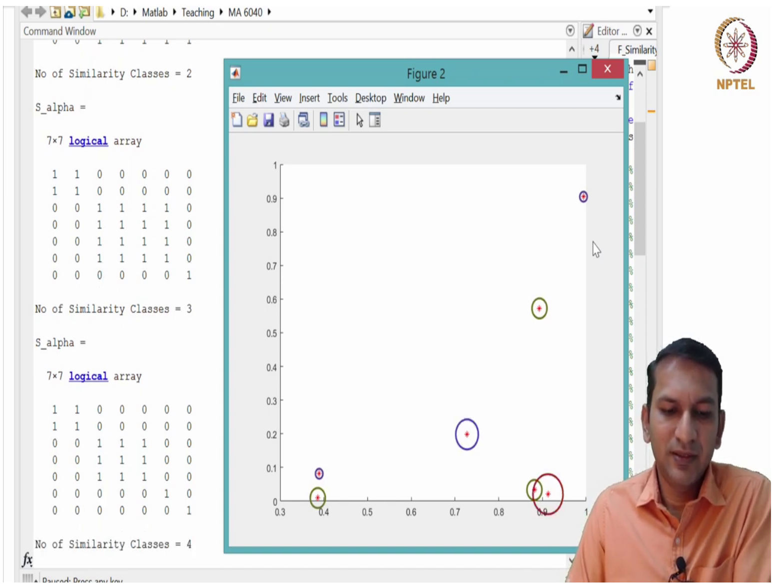
But now if you take the alpha at the level of 0.7, we see that we have a nice block matrix structure a and b are related to themselves and nobody else. And c, d, e f and g are related to each other and not to a or b. So, that is how you see the first similarity class consists of these two points which are a and b, and the other similarity class consists of the rest of the 5 points. Now, two similarity classes has indicated here.

(Refer Slide Time: 26:26)

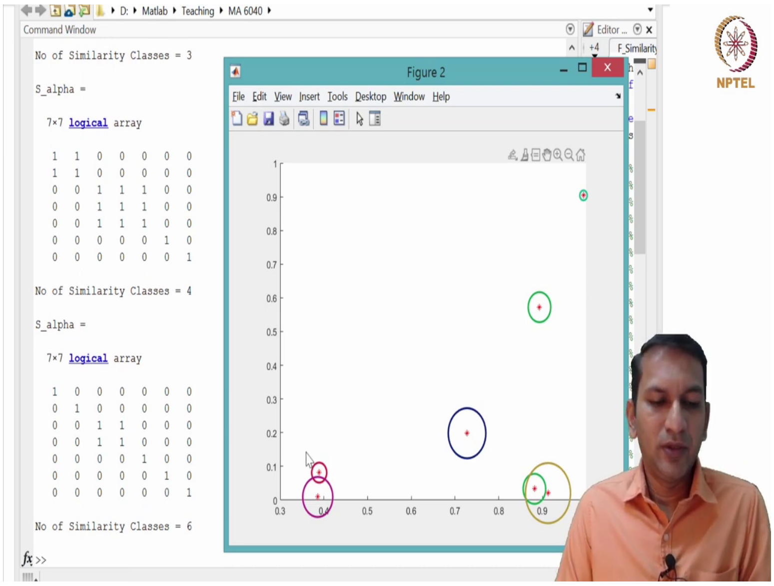


If you move up to the resolution level of alpha is equal to 0.8 we see that still the block structure is maintained, a and b are related to each other; c, d, e and f they form a similarity class, and g separates all. So, these are a and b for us, the second similarity class and this is the final one which is a single term. Now, at the level of 1, we still have some block structure here at the next level.

(Refer Slide Time: 26:45)

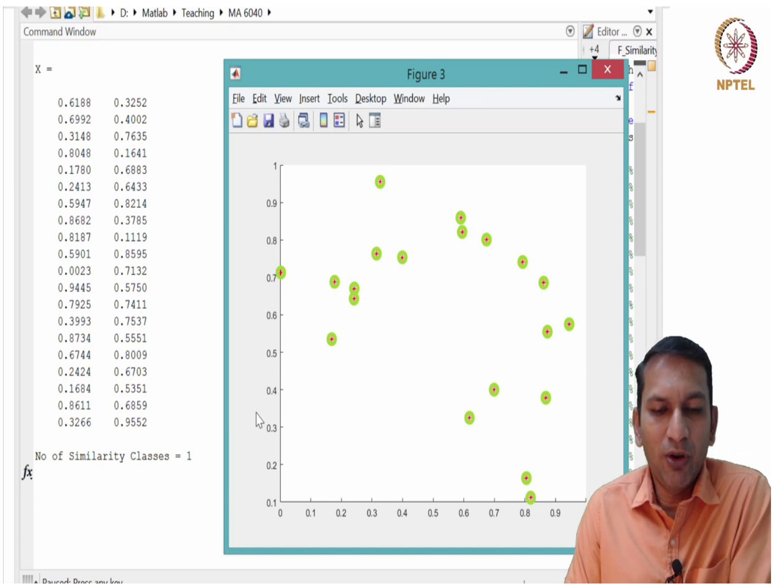


(Refer Slide Time: 26:56)



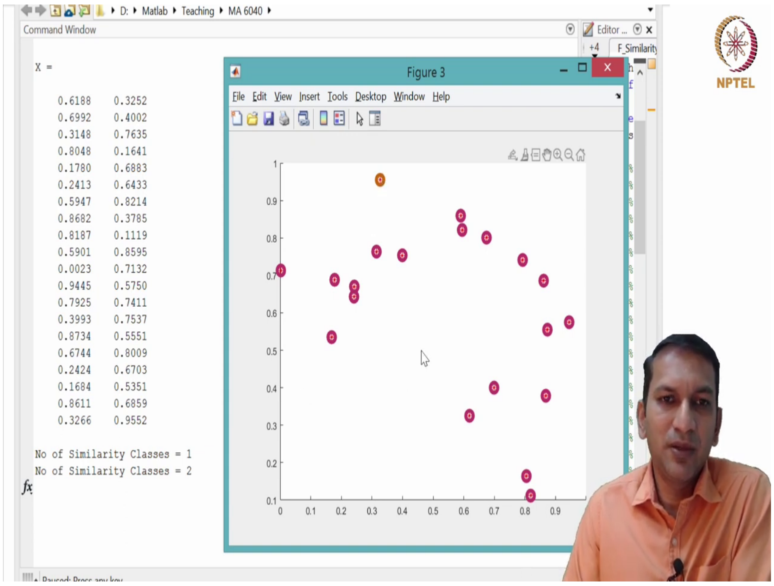
There are 4 similarity classes. And finally, when alpha is equal to 1, we see that because of this block structure here, we see that c and d are related to each other. They stay together under the same similarity class and rest of them become single time similarity classes. Now, allow me to stay with this a little more.

(Refer Slide Time: 27:19)



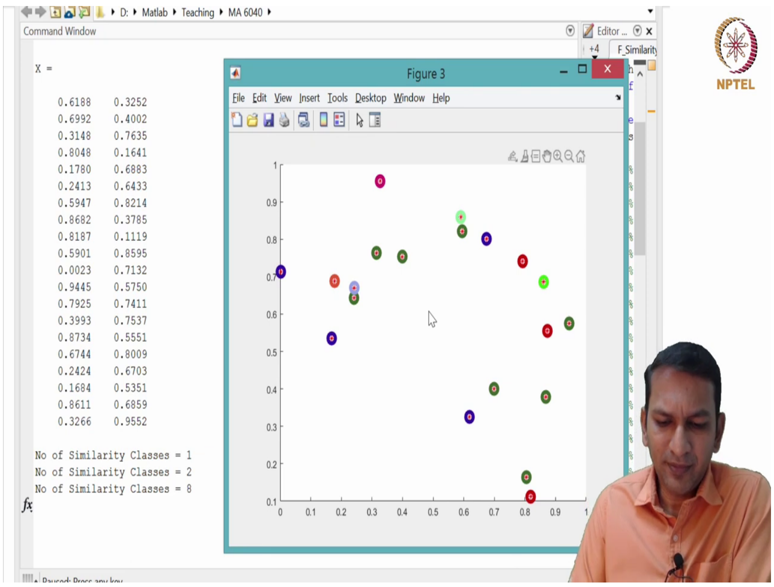
In here what we have are 20 points arbitrarily created and we have some kind of a similarity relation generated on these 20 points. Now, let us look at what happens when we start looking at alpha cuts at different alphas. Now, this is the smallest alpha. So, all of them belong to the same similarity class, right.

(Refer Slide Time: 27:52)



Alpha is equal to 2 and there are two classes. So, this is a slightly different color than what you see here. So, this fellow stays separately whereas, rest all of them belong to the same similarity class.

(Refer Slide Time: 28:05)



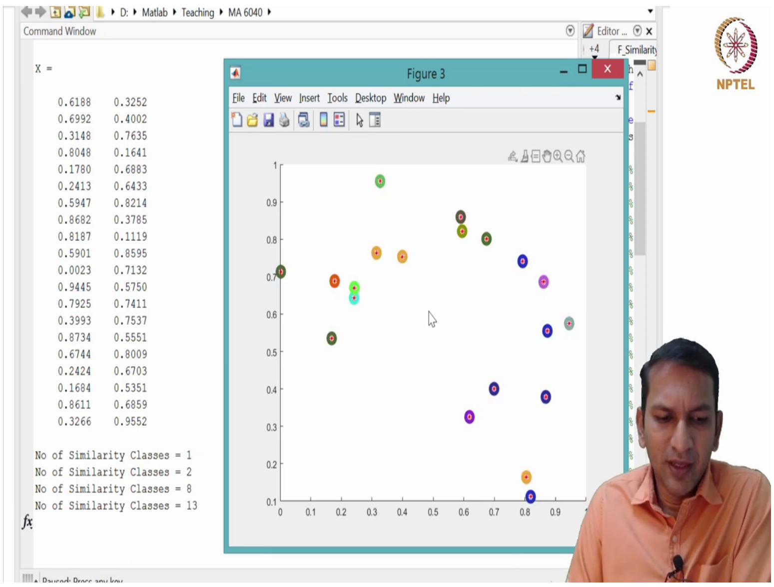
Now, the moment you increase the alpha, then we see that there are 8 similarity classes. Now, (Refer Time: 28:13) please do not go into this grouping or clustering based on visual distances between these objects, instead think of them as nodes on a graph. And each of those edges is weighted with the similarity value.

Now, what this algorithm has done because of transitivity that is in that is embedded in the relation, we are able to see that these points they are actually related to each other. Even though if you go with the distance, they are not.

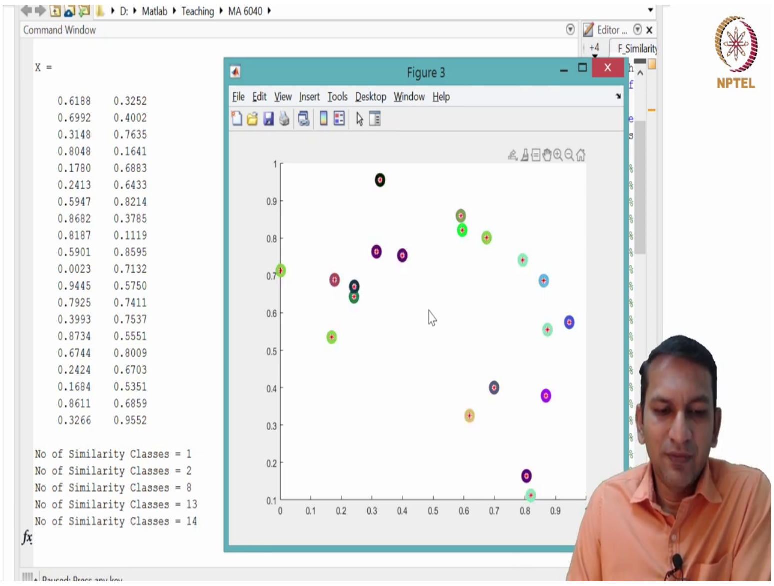
But based on the similarity relation that we have extracted or unearthed, in whatever scenario, it could be a set of people each of these nodes could be a set of people, it could be some somebody who is giving his preference on a set of attributes. And that is how we have generated the similarity relation or it could be each one of these nodes would be a portrait and the similarities between other portraits is what has given us the similarity relation. And now we are trying to group them, in some sense cluster them.

So, now, you see here even though they are physically apart, but they seem to be having the same likeness in terms of the attributes they prefer or in terms of the resemblances in terms of in their photographs.

(Refer Slide Time: 29:37)

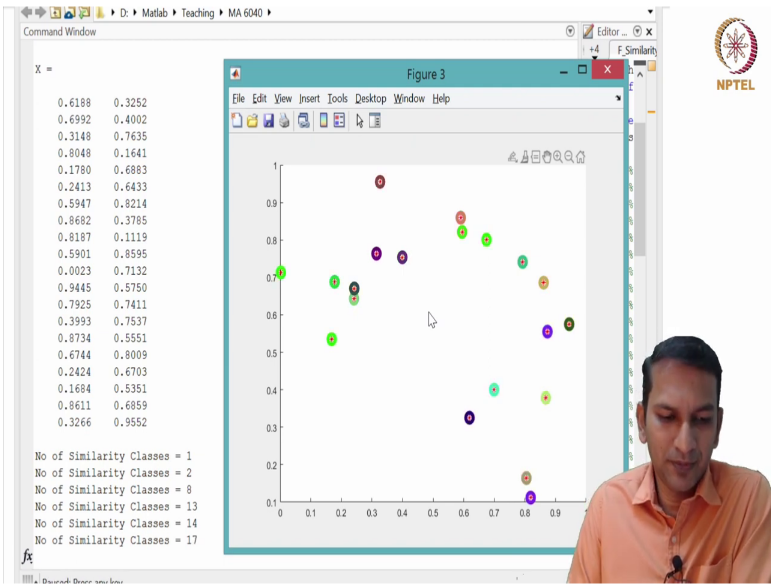


(Refer Slide Time: 29:39)

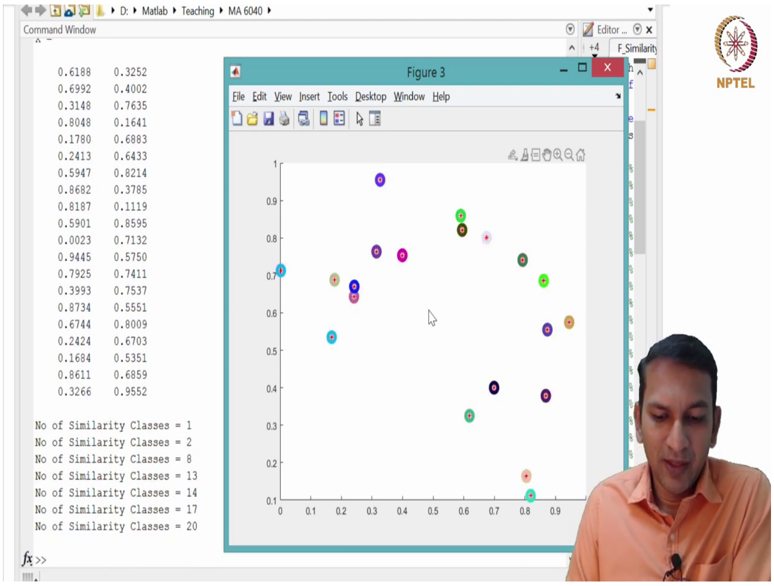


Now, if we go like this, we see that slowly at different resolutions as you increase the alpha, increase the threshold of acceptance to be similar, we see that we are getting a finer and finer groupings, means we get more groupings. And obviously, we get less number of people, less number of objects in each of those groups.

(Refer Slide Time: 30:00)

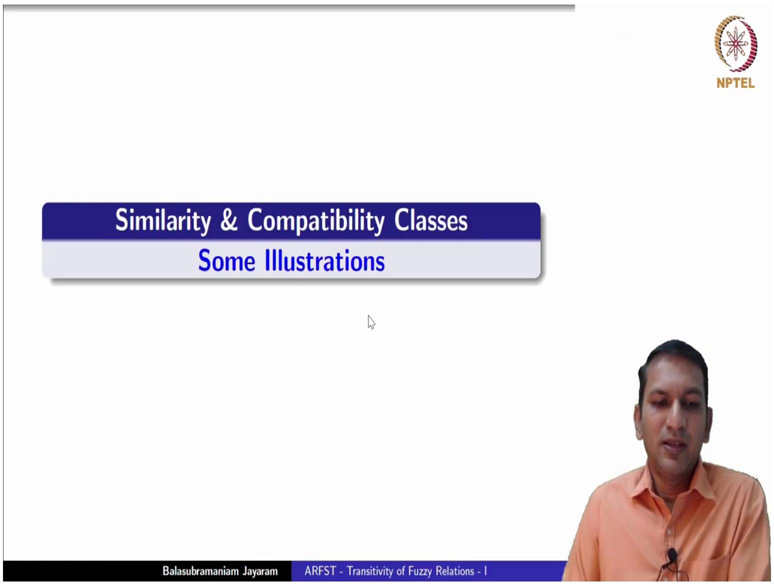


(Refer Slide Time: 30:04)



So, this is one way of doing clustering. It also goes under the name of relational cluster.

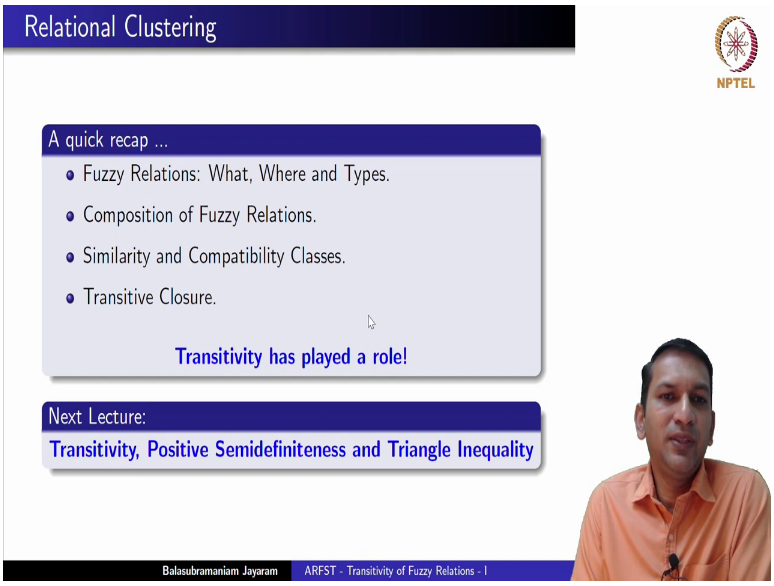
(Refer Slide Time: 30:13)



These are some illustrations which have shown you that if you take a compatibility relation, then you get overlapping compatibility classes. While if you have similarity relation, that means, you take a compatibility, apply the transitive closure, make it min transitive and get a similarity relation.

Then, by taking different alphabets at different resolutions, you are in fact able to get a partition on the underlying set which effectively turns out to be groups or groupings on the set of objects that you are considering.

(Refer Slide Time: 30:50)



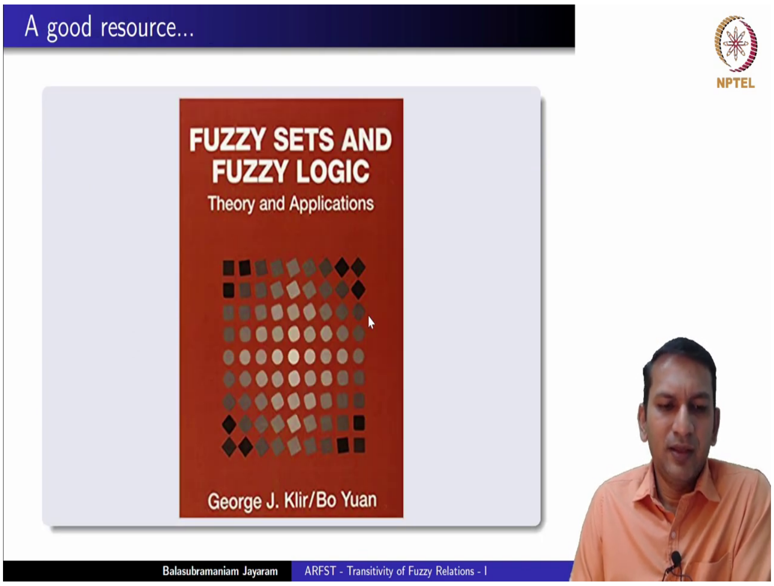
A quick recap of what we have seen so far. Said, we looked at fuzzy relations and in that we have looked at two different special types of binary fuzzy relations that of similarity and compatibility.

Note that transitivity plays a role here. When we discuss composition of fuzzy relations, we did not see them in terms of transitivity, but we now know that transitivity is related to composition and composition is related to transit. Finally, when we are discussing similarity in compatibility classes, we have seen the role, the key role played by transitivity there are, there again.

And today's lecture, we have seen how to infuse transitivity to a fuzzy relation if it does not have one to begin with. In all these things transitivity has played a role. What next? In the next lecture, we would like to see how the transitivity of fuzzy relation is related to the triangle inequality that you can get from a metric or in the language of kernels, the positive semi definiteness of a kernel.

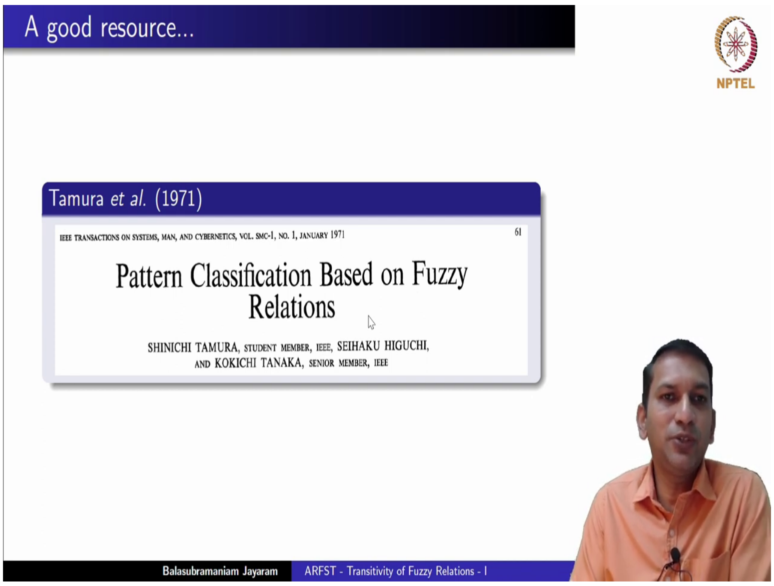
And you see that in that sense fuzzy relations can also be thought of as kernels or as being obtained from metrics themselves. In fact, if you have a similarity relation as was already mentioned, similarity and distance they can be thought of as dual of one another. This is what we would like to deal with in the next lecture.

(Refer Slide Time: 32:34)



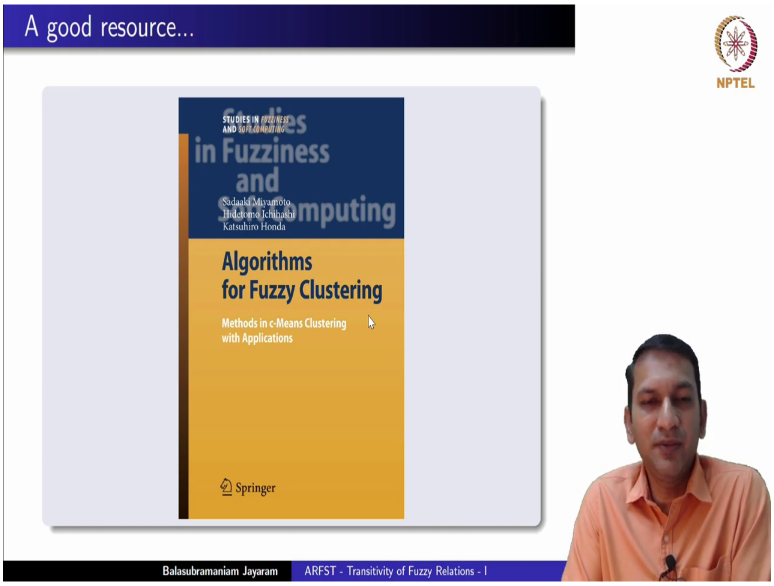
The algorithm that we have seen here, and some of the related properties are very well dealt with in this book like Klir and Bo Yuan. So, this can form a good source if you want to read up on the topics that we have discussed in this lecture.

(Refer Slide Time: 32:51)



This is the paper of Tamura and others published in 1971. In fact, the very first issue of a volume of IEEE Transactions on Systems, Man and Cybernetics, wherein they deal with a set of collection of photographs of members of some families and they try to actually group them into families based on relational clustering.

(Refer Slide Time: 33:15)



Finally, what we have seen as relational question based on similarity relations on set X, can also be thought of as hierarchical clustering, a particular form of hierarchical clustering. And if you would like to know more about this, a good source on this topic would be the book by Miyamoto and others titled Algorithms for Fuzzy Clustering.

Glad that you could join us in this lecture. Hope to meet you soon in the next one too.

Thank you.