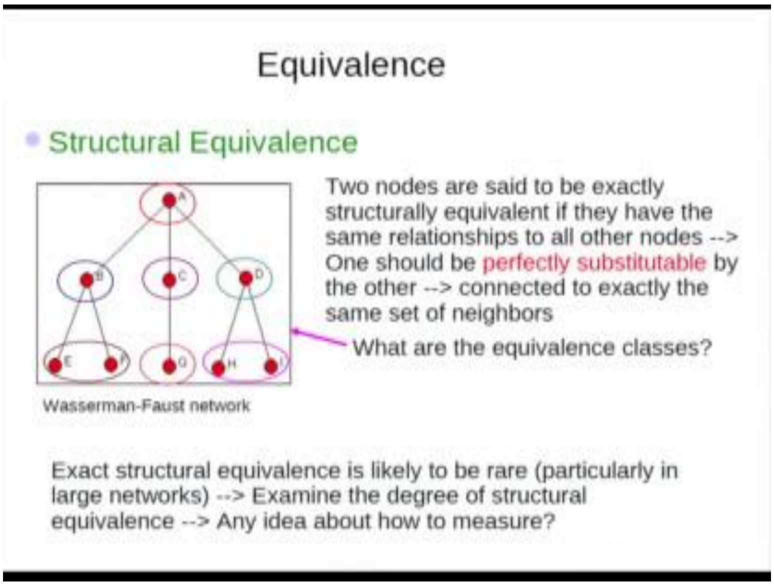
Complex Network: Theory and Application
Prof. Animesh Mukherjee
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 10
Social Network Principles – III

In the last few lectures, we have been talking about Social Network Properties mainly and the immediate previous lecture, we talked a little bit about social roles and the idea of equivalences. And we also introduced the specific case of structural equivalence where we said that two nodes are structurally in equivalent if they are farcically substitutable in the network that is they have the exactly same set of neighbors as we saw in the last example, if you look at the slides.

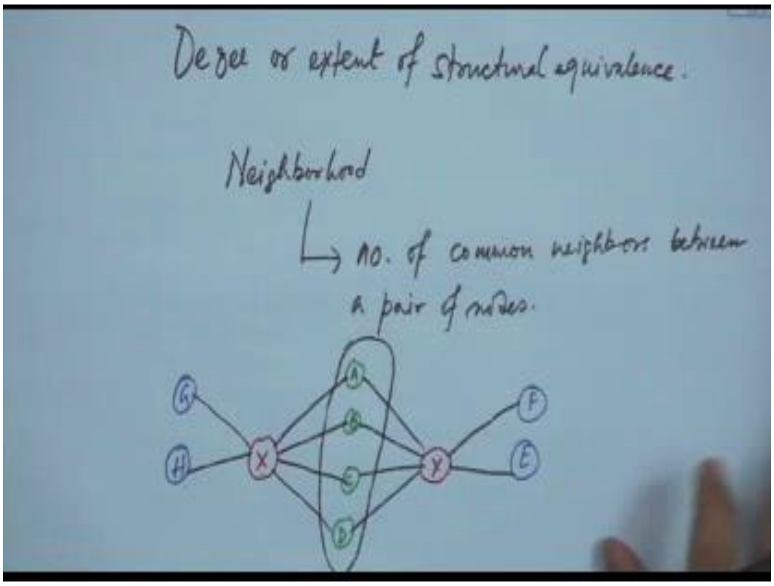
(Refer Slide Time: 00:55)



We had this same Wasserman-Faust graph and then we said that nodes E F and H I are structurally equivalent to one another and that is why you see they define a equivalence class in themselves.

Now, the point is that in many cases it is very difficult to find, in many real graphs it is very difficult to find exact structurally equivalent pairs of nodes. So, what we basically resort to is some sort of degree or extent of structural equivalence.

(Refer Slide Time: 01:32)

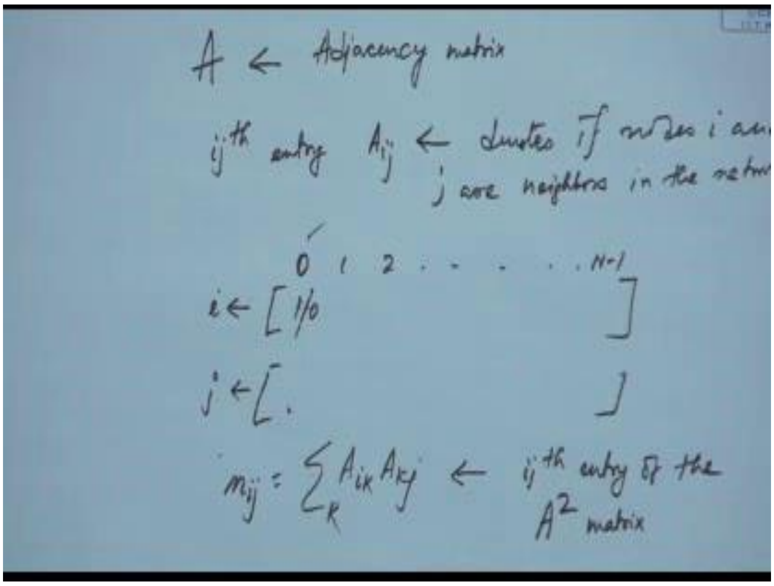


So, basically the idea is that if although we do not have many cases were two nodes are perfectly six substitutable, but are there cases which are like almost substitutable, the question is like are there cases which are almost substitutable. So, in order to do this in order to have a quantitative understanding for this, what we have to basically look into as we saw for structural equivalence we have to look into say the neighborhood, and the proxy for structural equivalence will be the count of number of common neighbors between a pair of nodes.

So, basically let us take a small example and see say, for instance, you have two nodes here one named X and the other named Y. Now, say also they have a bunch of neighbors A, B, C, D, G and H. Now, suppose the neighborhood structure is defined like this by the black edges that I am drawing. So, you see if you look at these example what you find is that num node X and node Y has a common neighborhood of four other nodes, A, B, C, D whereas, X has two other neighbors G and H which are not connected to Y and Y, similarly has two other neighbors F and E which are not connected any way to X.

So, basically we quantify the extent of equivalence or the extent of structural equivalence in terms of the number of common neighbors between every pair of nodes in the graph. Now, how to estimate this? So, if you look at the adjacency matrix.

(Refer Slide Time: 04:21)



So, imagine the adjacency matrix of a graph. So, if I call that adjacency matrix A, every entry i j th entry. So, this is the adjacency matrix. Now, i j th entry which we sometimes referred to as A i j is basically denotes if nodes i and j are neighbors in the network. In this way, if you look at the adjacency matrix you can find out all the neighbors of i. So, what will these be, the row corresponding to i will give you all the neighbors of i.

So, basically i will have a row in the adjacency matrix of size 1 to n minus 1, if there are n nodes in the graph actually 0 to n minus 1 including i. So, if there are n nodes in the graph and then you have 0, 1, 2 so on and so forth. So, if node i is a neighbor of node 0 then you put A 1, if it is not a neighbor of node 0 then you put A 0. In this way you make a vector of neighbors of vector representing the neighborhood of i.

Similarly, you can construct a vector representing the neighborhood of j. Now, given these two vectors you can very easily find out the common neighborhood, what will be this? Now, the common neighborhoods, basically if you look into these two vectors, the points where both of the entries in these two vectors are 1, indicate that both of them have this particular node common as their neighbor. So, if you just simply take the product of the two vectors you get a notion of the common neighborhood basically, that is what.

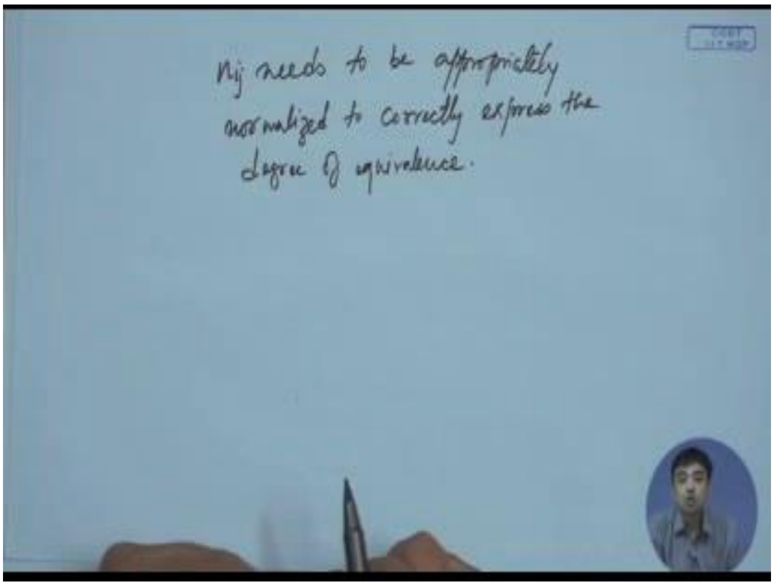
So, we call n i j the number of common neighbors and this can be calculated as using the following formula A i k A k j for all such case, where both A i k is 1 and A k j is 1, these value will be 1, otherwise it will be 0. If any 1 of them is 0 and the other is 1 or both of them are 0, this value will be 0 only when both of them are 1 that is both i is a neighbor of k and j is a neighbor of k. In all such cases, this factor this product will lead up to 1 and this will give you the count of the common neighbors between i and j between the pair i and j.

So, if you note carefully, this also expresses n i j also expresses i j th entry of the A square matrix. So, that is the adjacency matrix squared, the adjacency matrix made a product with itself. So, that the A square matrix every entry in the A square matrix is nothing, but n i j that is the number of common neighbors between i and j. So, in this way also n i j can be interpreted.

Now, see we are telling that we want to quantify the extent or the degree of structural equivalence and number of common neighbors is the proxy that we want to use, but then like if you have if there is a pair which out of 100 neighbors 100 total neighbors have a common neighborhood of 3 and there is another one which out of 10 common neighbors have a total neighborhood of 3 that makes a difference actually.

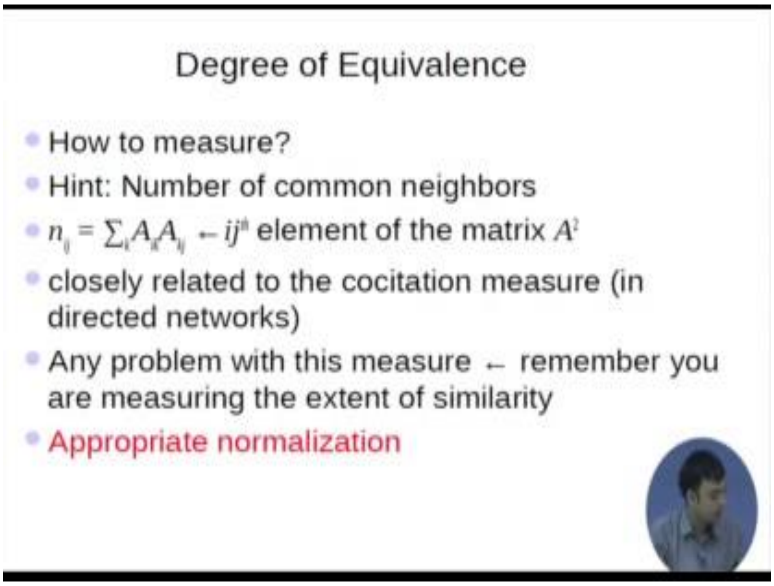
So, basically what I am trying to point out is that this particular factor needs to be appropriately normalized.

(Refer Slide Time: 08:45)



So, n i j needs to be appropriately normalized to correctly express the degree of equivalence fine.

(Refer Slide Time: 09:28)

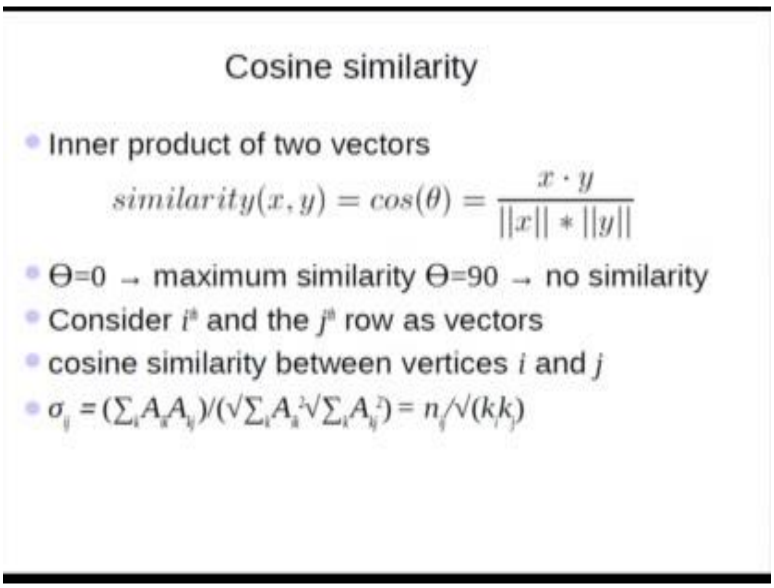


So, in order to do this appropriate normalization, we resort to various different methods. The first method that we will talk about is; please look at the slides. The first method that we will talk about is cosine similarity. So, what does this cosine similarity say? So, cosine similarity basically tries to measure the cosine angle between a pair of vectors. The cosine similarity tries to measure the cosine angle between a pair of vectors.

So, here as we said in the last part, that you can express every node in the graph as a vector of it is neighborhood. Now, if you take the cosine similarity between these two vectors, say the vector it for representing the neighborhood of i and the vector representing the neighborhood of j. Now, if you take the cosine similarity or the cosine angle between these two vectors then if this cosine angle is 0 that means the cosines in the angle is 0 means that they are perfectly aligned. These two vectors are perfectly aligned that is they are exactly the same so that means, cos theta in this case will be 1 whereas, if these two are completely orthogonal to each other then the theta angle will be 90 degree and cos theta will be 0.

So, basically perfectly similar cases we will have a cosine angle of 1, we will have a cosine angle of 0 with cos theta evaluating to 1 and perfectly orthogonal cases, we will have a cosine angle of 90 degree with cos theta evaluating to 90. Now, this actually is expressed by the formula that you see on the slide.

(Refer Slide Time: 11:12)

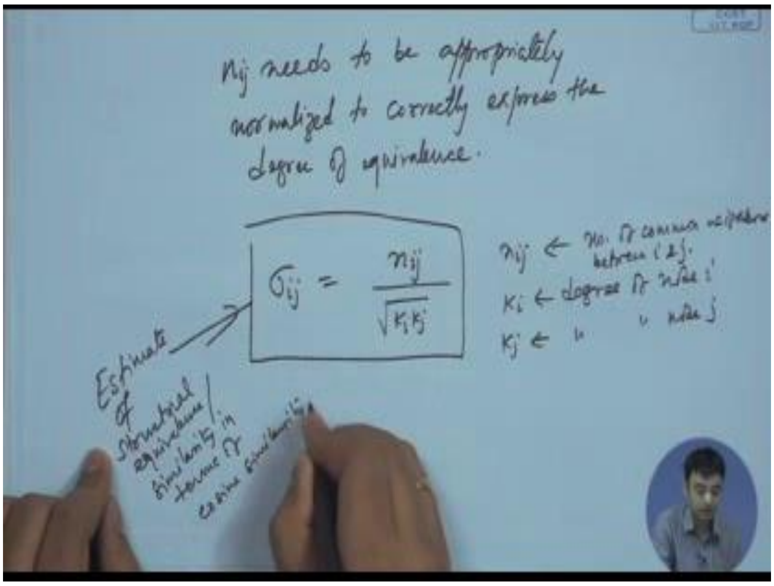


So, take two nodes x and y and say, x is the vector corresponding to node i and y is the vector corresponding to the node j. So, x represents the neighborhood vector for node i and y represents the neighborhood vector for node j.

So, cos theta is calculated as x dot y by the normalization, by the norm of x multiplied by the norm of y, this is in terms of the entries of the adjacency matrix. This cos theta, if it is if the theta angle is 0 then cos theta will evaluate to 1 and that is perfect alignment that is the two nodes are completely structural equivalent whereas, if theta is 90 degree then cos theta evaluates to 0 and they are not at all equivalent. Now, this in terms of the entries of the adjacency matrix will result into the formula that I write in the last pointer of this slide sigma i j. So, sigma i j is basically sum over k A i k k j which is nothing, but n i j as we have seen last time. So, this is the number of common neighbors between i and j, between the nodes i and j. This is also measured by this dot product here.

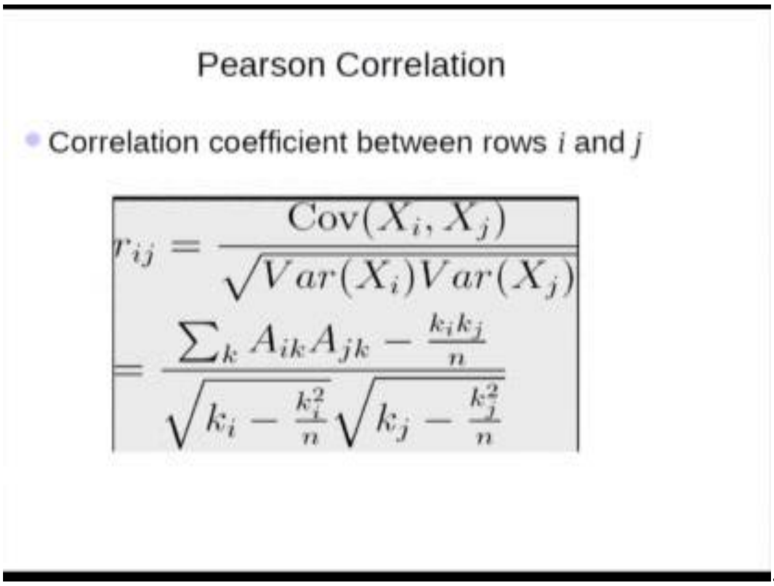
Whereas, the normalization goes like this, it is the square root of the sum of all k sum over all k A i k square into square root of sum over all k A k j square. So, this is nothing, but the square root of the product of the degrees of this two different matrices, assuming that the adjacency matrix is A 0 1 matrix if the adjacency matrix is the 0 1 matrix then sum of the square of it is elements will be nothing, but the degree of that node similarly, for i you have a degree for j you have a degree and the normalization factor is nothing, but the square root of the product of the degrees.

(Refer Slide Time: 13:20)



So, now finally, you have sigma i j that is the cosine similarity is equal to n i j which is the number of common neighbors by square root of k i k j where k i is the degree of node i k j is the degree of node j and n i j is the number of common neighbors between i and j between nodes i and j. So, this is the estimate of structural equivalence in terms of cosine similarity is the estimate of structural equivalence or similarity in terms of cosine similarity. So, similarly you can also design various other metric. The next metric that we will look into is the Pearson correlation coefficient.

(Refer Slide Time: 14:44)



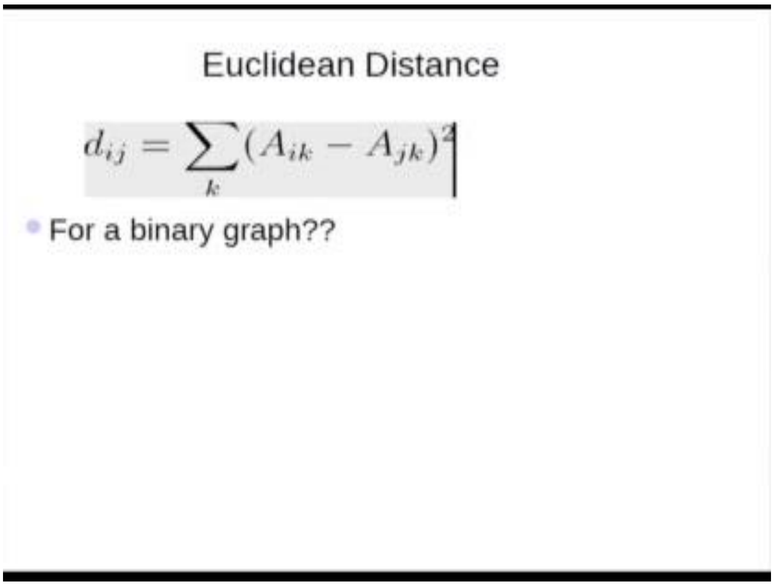
So, basically again as we saw earlier that node i and node j can be represented by neighborhood vectors like these. Now, given the neighborhood vector of i and j, you can basically find out. So, these are basically can be assumed as the terms or the variables and. So, there will be 1 vector containing all the neighbors of a containing a neighborhood information of i are all the neighbors of i whereas, there is another vector which contains all the neighbors of j.

Now, you can find out. So, these are you can treat these two set of numbers as a distribution in themselves. Now, you can find out basically the Pearson correlation coefficient between these two numbers. So, what I mean to say, suppose you have two nodes i and j and you have. So, with node number 0 i is connected j is not connected with node number 1 i is not connected j is connected and so on and so forth. Then may be two node number two; both of them are connected to node number, three both of them are connected and so on and so forth. So, given these two columns you can find out easily find out the Pearson correlation between the two columns, two sets of items and that is actually measured using this particular formula that you see on the slide.

So, it is basically the co variance of x i x j by square root of variance of x i into variance of x j. So, the square root of x i covariance of x i x j can be calculated as the product of A i k and A j k. So, that is nothing, but n i j as we have already defined minus as I already said that for correlation analysis you usually take out the term which is possible just by random chance. So, by random chance if two nodes having a particular degree becomes neighbors. So, we take out that factor we discount that factor. So, this is basically the co variance by the square root of k i minus k i square by n into square root of k j minus k k i square by n that is the square root of the variances of each of the distributions the distribution of neighborhood of i and the distribution of neighborhood of j. So, in this way you can compute the Pearson correlation coefficient between these two vectors.

Now, if the Pearson correlation is high; that means, if it is close to very close to 1 then this is an indication that the two nodes are structurally highly equivalent whereas, if this is low or negative then this would mean that the two nodes are not structurally very equivalent to each other. So, this is another way of defining the degree of structural equivalence.

(Refer Slide Time: 17:45)



There is yet another method where we use the Euclidean distance. So, again since these are binary vectors node the neighborhood can be represented as a binary vector like this and the neighborhood of j, can also be represented as a binary vector like this are also here same thing. So, then you can find out basically the Euclidean distance between this particular vector here and this particular vector here. So, since this is a 0, 1 vector both of them Euclidean distance is also proportional to the having distance basically.

So, you can basically find out the having distance between these two binary vectors, but again having distance or the Euclidean distance. So, they should be proportional in this particular case because it is a binary vector. So, once you have calculated this you have to also appropriately normalize. So, what could be a nice normalization technique here, so you remember in the first case, we normalized for the cosine similarity we normalized by square root of k i k j in the second case we normalized.

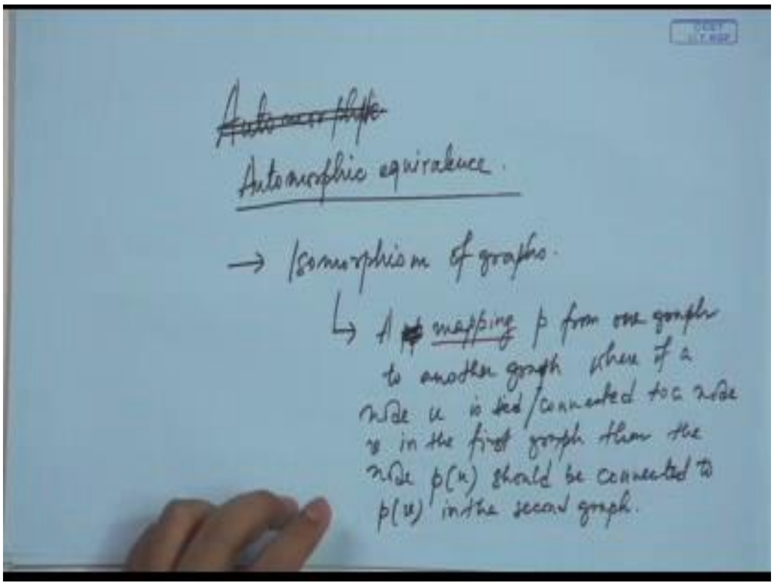
Again by the square root of the variance of x i and the variance of x j square root of the product of the variance of x i and the variance of x j similarly we need to find out normalization factor here. So, if you look carefully. So, the worst possible case is that none of the neighbors of i are similar to the neighbors of j. So, then in such a case what is the extreme Euclidean distance? The extreme Euclidean distance in this case is that the neighborhood overlap is 0 that means, the Euclidean distance would be the sum of the degrees of the two nodes basically if you take node i say it is degree is k i and you take node j say it is degree is k j.

Now, none of these nodes here are overlapping with these nodes here so that means, the distance in this case, is basically k i with plus k j because none of them overlap. So, there is no overlap between them. So, the straight forward the total distance between these two nodes neighborhood distance between these two nodes are overlap distance between these two nodes is k i plus k j. So, you can use this particular term as the normalization factor. So, you find out what is the overlap by the worst possible case.

This is how you can find out, this is the maximum distance that is possible and you find out what is the exact distance? Now, here the lower the distance the better you know the exact distance this if this distance is low normalized by this particular maximal distance then you say that these two nodes are structurally very similar to each other otherwise they are very different.

In this way, we quantify 1 of the very first types of social roles that is the structural equivalence. So, the next thing that we will introduce to is called the automorphic equivalence. So, in order to understand automorphic equivalence, we have to first identify and understand the concept of isomorphism of graphs. Now, this is a very simple concept. So, we will slowly define it using a set of examples. So, an isomorphism is defined as follows. So, this is a mapping.

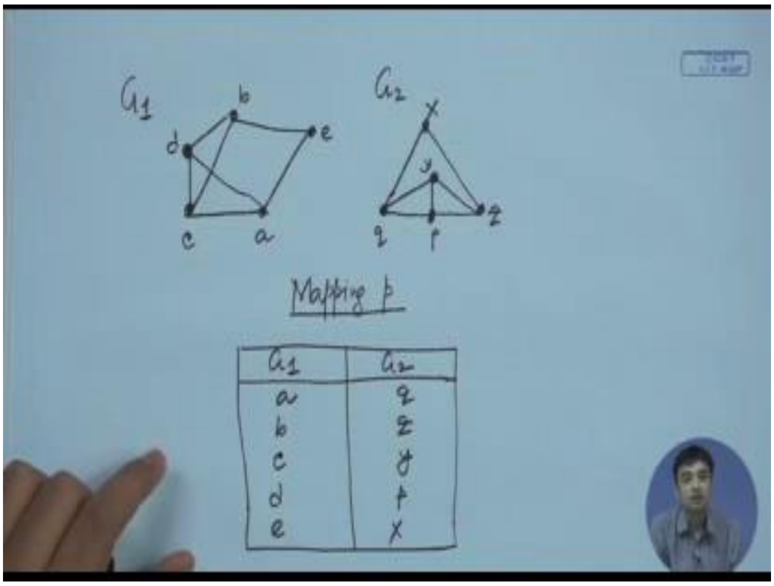
(Refer Slide Time: 21:07)



So, an isomorphism is a mapping. So, mapping means a function is a mapping p from 1 graph to another graph, whereas if a node u is tied or connected to a node v in the first graph then the node p of u should be connected to p v in the second graph. So, basically an isomorphism is a form of mapping. We say, we define this mapping as p from 1 graph to another graph such that if there is edge between two nodes u and v. In the first graph, there will be an edge in the fir between the nodes p u and p v in the second graph and this mapping is a bijection.

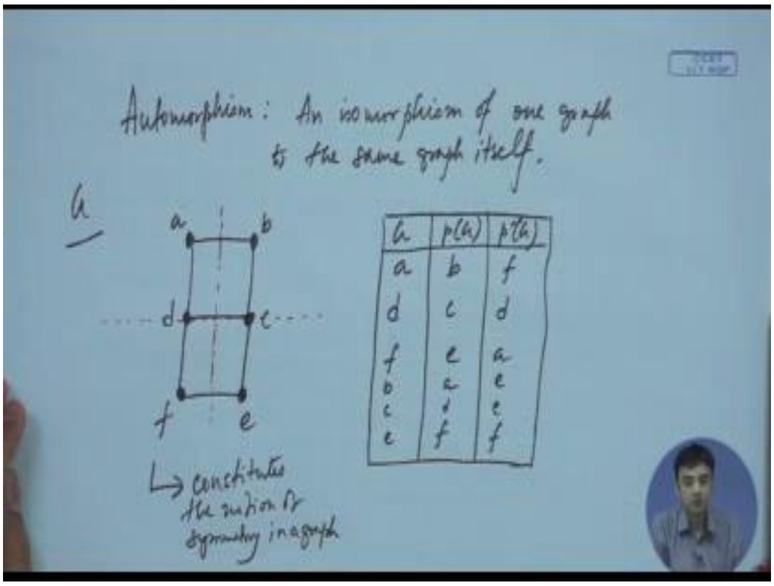
So, this mapping is a bijective mapping. We will take a typical example of isomorphism and then establish the concept of automorphism.

(Refer Slide Time: 24:55)



Let us take these two graphs. So, let us call these graphs G 1 and this graph G. Now, we can actually construct a mapping p between these two graphs as follows G 1, G 2. So, the node number a in the graph G 1 can map to node number q in G 2. Similarly, b in G 1 will map to z. Similarly c in G 1 will map to y d will map to p and e will map to x. So, in this way, you can actually establish a nice mapping relationship a bijective mapping between the two graphs G 1 and G 2 and therefore, G 1 and G 2 are called isomorphic to 1 another. So, using this concept we can now describe automorphism. So, automorphism is nothing, but an isomorphism from 1 graph to the same graph itself.

(Refer Slide Time: 26:12)



Automorphism can be defined as an isomorphism of 1 graph to the same graph itself fine. So, this is what is called automorphism, again let us take an example and see. So, let us take this graph here. So, we can again define mappings on the graph say let us call this graph G, we can have various forms of automorphic mappings here, at least to that we can immediately find out p (G) is by first kind, p prime G another kind of automorphism.

In 1 case, basically automorphism, actually constitutes the notion of symmetry notion of symmetry in a graph. So, basically here again you see this symmetry is very, very apparent. So, look at the line of symmetry indicated by this broken red line. Now, you can see that, if you take this as the line of symmetry in a is automorphic to b, d is automorphic to c, f is automorphic to e, similarly b is automorphic to a, c is automorphic to d and e is automorphic to f. On the other hand, you can also construct another line of symmetry across the horizontal plane.

Here, the automorphism or the isomorphism is established between a can map to f, d maps to d itself, f will map to a, b will map to e, c will map to c itself and e will map to f. So, in this way you can construct automorphism for a given graph. So, automorphism as I am telling you again, I am iterating the fact that automorphism actually constitutes the symmetric structure present in a graph. So, it actually captures the symmetry associated with the particular graph.

So, we will stop here.