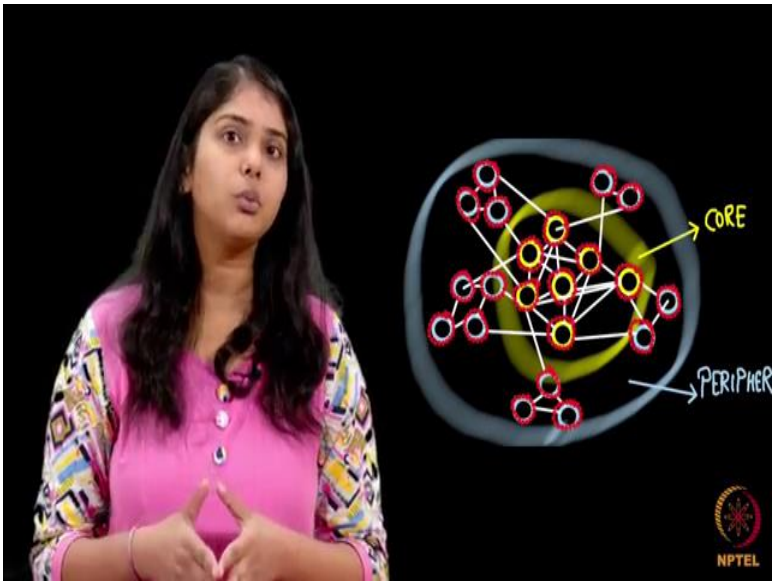
Social Networks
Prof. S. R. S. Iyengar
Department of Computer Science
Indian Institute of Technology, Ropar

How to go Viral on Web
Lecture - 162
Finding the right key nodes (the core)

So, here I tell you the answer how should you choose these key people. Do you remember we talked about this structure in one of our chapters in a small world phenomenon? We talked about this structure and it is called a core periphery structure.

So, we have looked that there are certain people in this world some small fraction of people who are rich, who have access to better resources, who can travel throughout the world and such people can manage more connections and rather connections to different-different parts of the world and they lie at the centre of a network something like this.

(Refer Slide Time: 00:45)

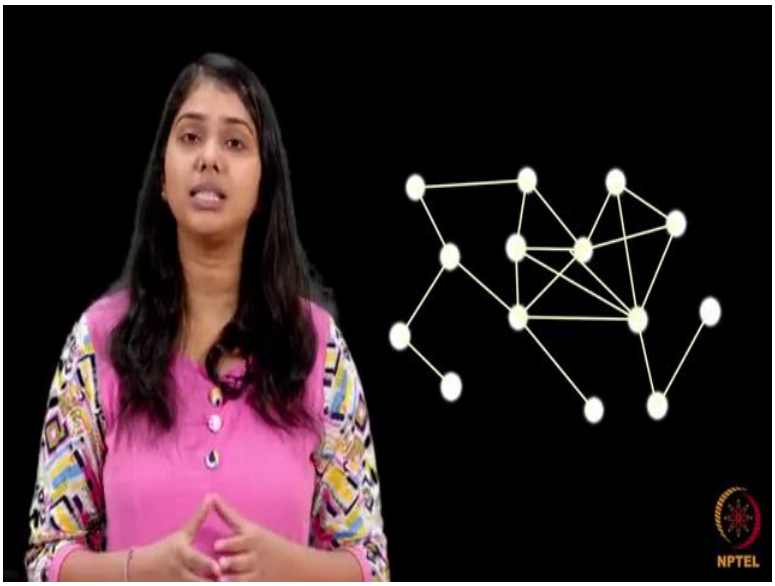


And rest of the people like a periphery are towards the periphery of this network they are towards the boundary. So, we talked about this structure before. This structure has a very big role to play here. So, it has been shown that if you infect one of these core nodes here you see what happens. So, if you closely look at this figure, you will see that these core nodes here they are very densely connected to the other core nodes.

So, if you infect one of these core nodes, it will quickly tell another core node, which will quickly tell another core node and soon this entire core will come to know of your idea. And we know that this entire core is now very nicely connected to all the people in the world to all the periphery. So, once this component is infected, your mean will go and reach all the periphery nodes as well. So, this should be the core nodes which you should be infecting ok.

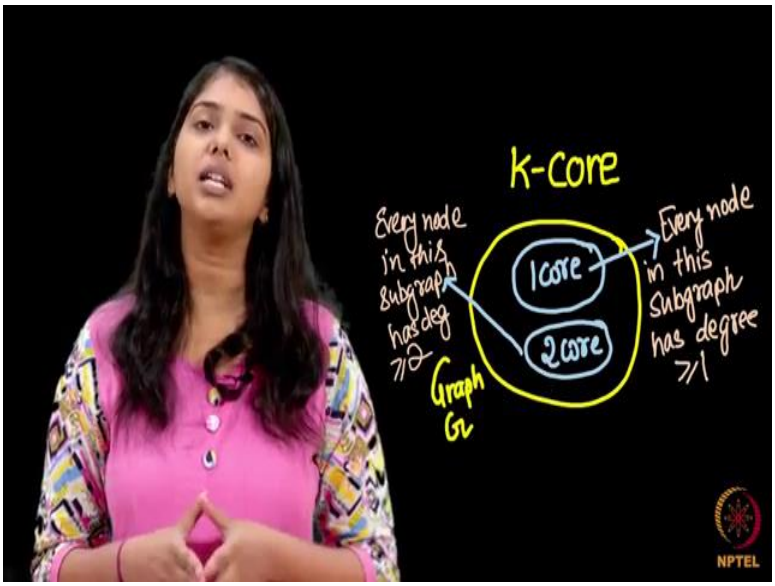
Till now, well and fine we come to know, we talk about the core periphery structure and we say that if this is the core which is infect my mean is going to become viral. Now I give you a network here can you tell me what is the core here?

(Refer Slide Time: 02:05)



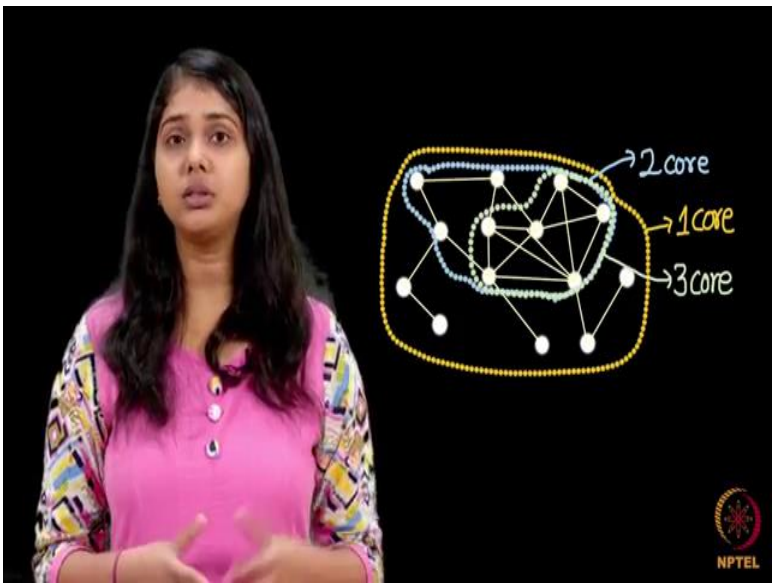
How do we even come to know what is core of a network, what lies in the core of a network? And then people have tried answering this question. And I will now discuss one of the most popular approach which you can use to identify these score nodes, identify these score nodes in this network. So, for that we will just switch to a definition a mathematical definition, but very simple and the definition is that of a K core.

(Refer Slide Time: 02:37)



So, given a graph let us say we are given a graph here, we say a node or we say not a node we take a sub graph from this graph and we say that the sub graph is 1 core. 1 core if every single node in the sub graph has the degree of at least 1. And similarly, you say a sub graph to be off 2 coress to be 2 cores if you look at every single node in this sub graph and every node has a degree of at least 2. So, for and so forth, so, you have 1 core, 2 cores, 3 cores, 4 core and so on. So, that is what defines a K core. You take a sub graph and every node in this sub graph should have a degree at least K ok.

(Refer Slide Time: 03:29)

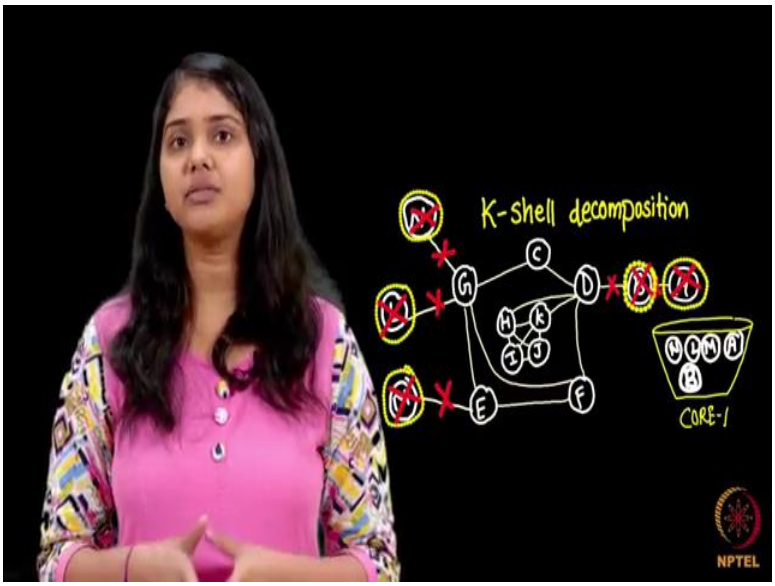


So, here is a graph. So, we can quickly see and try to figure out what is 1 core here, what is 2 cores, what is 3 coress. So, if we look at it and introspect it introspect on it a little bit we come to know that yes fine this is 1 core, this is 2 cores and this is 3 cores, but now what I am going to do? This is a small network. It was easy for you to look at it and then tell me what 1 core is, what is 2 cores, what is 3 coress.

(Refer Slide Time: 04:03)

I will give you a network having billions of nodes like Facebook or Twitter, I give you the network here and now can you tell me what is 1 core here, what is 2 cores, what is 3 cores and so on. It becomes very difficult. So, there is a very nice algorithm to do it. It is actually intuitive if you little bit think about it, you can converge to this algorithm on your own. And the algorithm is very interesting as well So, this algorithm determines what goes in 1 core, what goes in 2 cores, what goes in 3 cores and so on and it is known as K shell decomposition algorithm.

(Refer Slide Time: 04:39)



K shell decomposition algorithm because, it decomposes your network into multiple shells. Now, let us see how does this algorithm work. So, you look at your network. So, here is your network. And let us say your network is some instead of nodes I use the word ball here. So, your network has many balls and these balls are connected to each other.

Then just for some time I am using the word ball instead of the word node. And then I come with a bucket and I label this bucket as 1 core and put this bucket here. And then I look at this network, I look at all the balls, all the nodes which are having degree 1; which means which have only 1 friend, we also call them pendant vertices if you know graph theory otherwise you can just leave it.

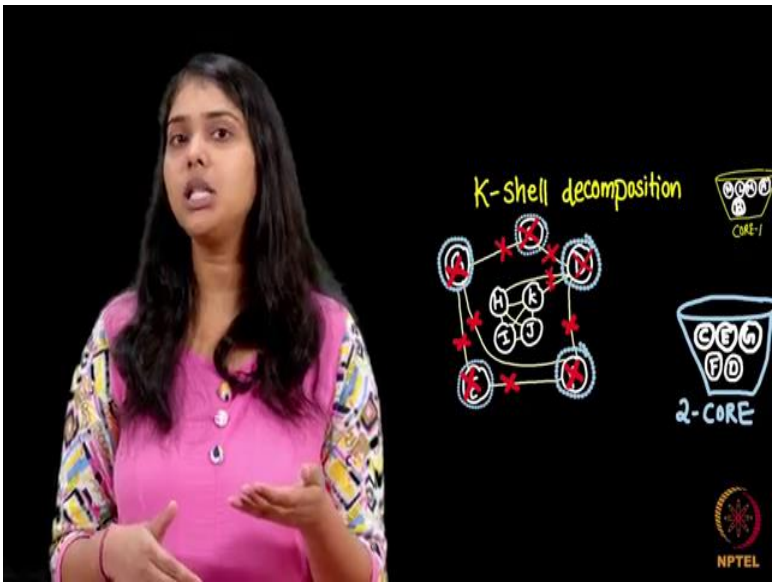
So, the just have degree 1 they are kind of attached to the network with just 1 edge. I take these nodes I plug them out and put them in this bucket. So, when I plug a node out please note that all the edges which are outgoing all the edges which are incident on this node also gets plugged out.

So, when I plug out a node, I cannot have a hanging edge in this network right. So, that edge is remove and I put that ball in this bucket. And then I keep doing it, I look at another ball which is having degree 1, put it in the bucket, I keep doing with all the balls and I look at all the balls having degree 1 and put it in the bucket.

Did you notice something here? When I pull a ball having degree 1 another ball which was previously having degree 2 can now become a ball having degree 1. So, that is when you keep removing these balls with degree 1, new balls keep immerging in your network which has degree 1. So, if you look at this node A here, when I plug this node A out, keep it here, you see this node B, it has now become of degree 1.

So, what do you do? You also remove node B and put it in the basket. So, mainly you keep doing this business of plugging out the nodes and keeping them in the basket till there is no degree 1 ball in your network. Till all the nodes here they have degree 2 or a higher degree. And once you achieve such a graph which has nodes of degree 2 or higher degree you stop and you put your bucket aside then you bring a fresh bucket and then we label this bucket as 2 cores.

(Refer Slide Time: 07:21)



And you now can guess what I am going to do is. So, I put this bucket having label 2 cores here and I again locate all these balls in my network. And now I look at the balls which have degree which have degree 2 that is with 2 edges they are connected to this network.

So, for example, the node C here, you can see that it is connected with degree 2 to this network with 2 edges. So, I plug C out and keep it in the basket. I do it for all the balls which are having degree 2. I take all those balls and put those balls in this basket. And again did you notice the same thing happens when you plug a ball out which is having a

degree 2, a new ball having degree 2 might immerge and rather a new ball having degree 1 might immerge.

So, you see here when I remove this ball having this when I remove this ball E from this network having degree 2, this node F immerge is which has just degree 1. What I do, what should I do now with this degree one node? Shall I keep it in my previous bucket or shall I keep it in this bucket? So, I will tell you I keep it in this bucket. This market means the bucket label 2 cores. So, I keep removing the balls having degree 2 or lesser. So, in this iteration what all goes in this bucket 2 all the nodes having degree 2 or lesser.

(Refer Slide Time: 09:09)

And you keep plugging them out and new balls keep immerging which might have a degree 2 or lesser you plug those balls also out and put them in this same bucket and you keep doing; so till there are no nodes of degree 2 or lesser in the network. So now, in this network there will only be nodes having degree 3 or a higher degree.

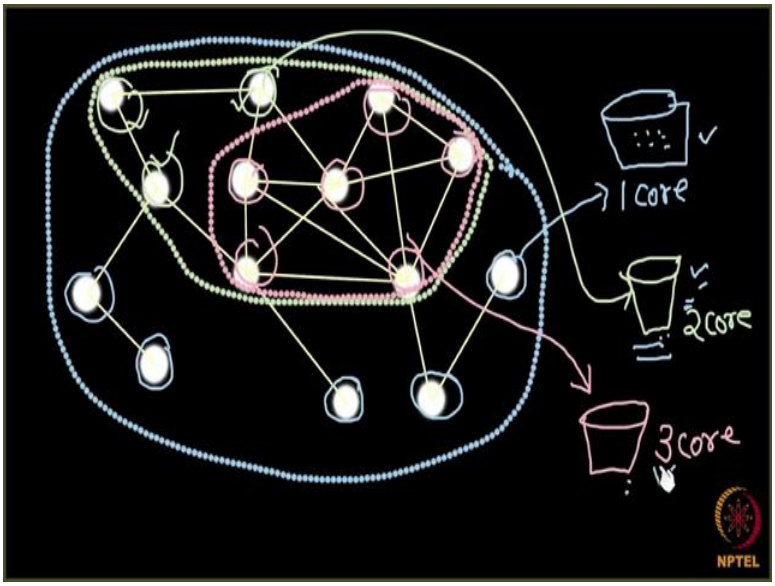
And when this is done, I pack this bucket and keep it aside and I bring the third bucket in picture, I label it as 3 cores and I start repeating the same procedure again. Plug all the nodes having degree 3 from this network and new balls might immerge which might be having a degree 3 or lesser and I keep all of them in the same bucket and I keep doing it. And we keep doing this procedure till my network becomes empty.

So, we are plugging nodes at every step. So, when we keep plucking nodes obviously one state should come when my network becomes empty and that might be at some K core at some K-th bucket, we do not know it can be a 10th bucket or it can be the 9th bucket.

Now, we have K cores. What we talk previously was about core and periphery; we were interested in finding the core of the network. So, what is the core of this network? So, when we do this K shell decomposition to find 1 core, 2 cores and 3 cores and so on. The nodes which are removed at the last step the most let me see the innermost core of the network, innermost shell of the network that is called the core.

So, in this network where we were having 1 core, 2 cores, 3 cores, the core of this network is this the 3 cores. And when we do k shell decomposition whatever goes in the last bucket, whichever nodes go in the last bucket form the core of that network. I want to tell something important related to K core, 2 cores and K shell decomposition.

(Refer Slide Time: 11:07)



So, we have looked at this network and, in this network,, we have looked at which node comprise of which nodes coming 1 core, which nodes coming 2 cores and which nodes coming 3 cores. So, we have seen that in this network the sub graph in which all the nodes have degree greater than or equals to 1 come in come in 1 core.

So, this entire network used to come in 1 core. So, this is the 1 core of this network. And then we have looked at 2 cores in which all the nodes have degree greater than or equals

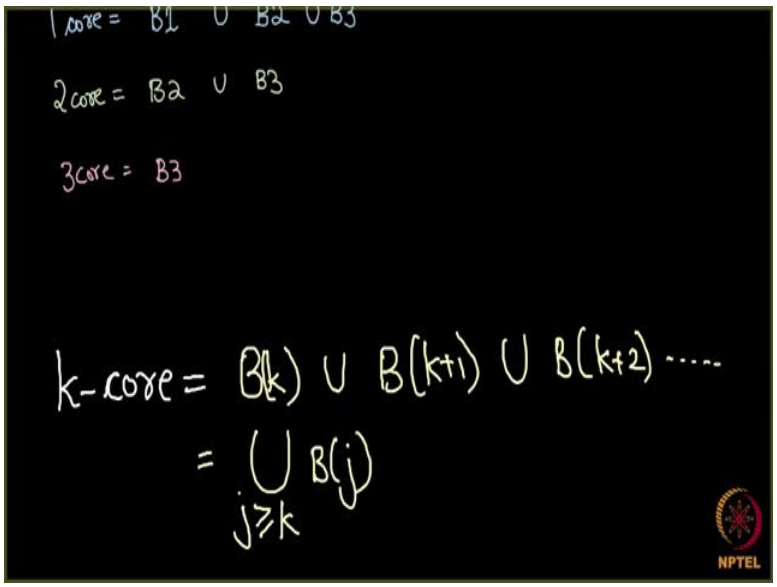
to 2. So, this particular portion in this network is the 2 cores. And then similarly we have looked at that this particular portion in this network is the 3 coress. And then next we will discuss the techniques to determine the 1 core, 2 cores and 3 coress in the network which was called K shell decomposition. Now what is important here is if you remember that in K shell decomposition, we made several buckets right. So, there was a bucket corresponding to 1 core.

So, let us say this was a bucket corresponding to 1 core and then we had a bucket corresponding to 2 cores and 3 coress also. So, in this bucket corresponding to 1 core, which all were the nodes in K shell decomposition and if you will apply K shell decomposition over this network, you will note that these nodes over here were the ones which were present in this bucket corresponding to 1 core.

Similarly, if we look at the bucket corresponding to 2 cores then we saw that these nodes over here 1, 2, 3 1, 2, 3, so these 3 nodes were in bucket of 2 cores. And then similarly we had a bucket of 3 cores over here So, this is a bucket for 3 cores and all the nodes, so 1, 2, 3, 4, 5 and 6, the 6 nodes over here were in the bucket for 3 cores So, we had these 3 buckets.

Now, what is important here is please note that all the nodes which are present over here in the bucket for 1 core. So, if you are asked over this network what is the 1 core of this network?

(Refer Slide Time: 13:31)



So, 1 core of this network is according to K shell decomposition 1 core of this network is the nodes which are present in bucket corresponding to 1 core. Let us say that bucket 1 union bucket corresponding to bucket corresponding to 2 cores which is B 2 union bucket corresponding to 3 cores which is B 3 right.

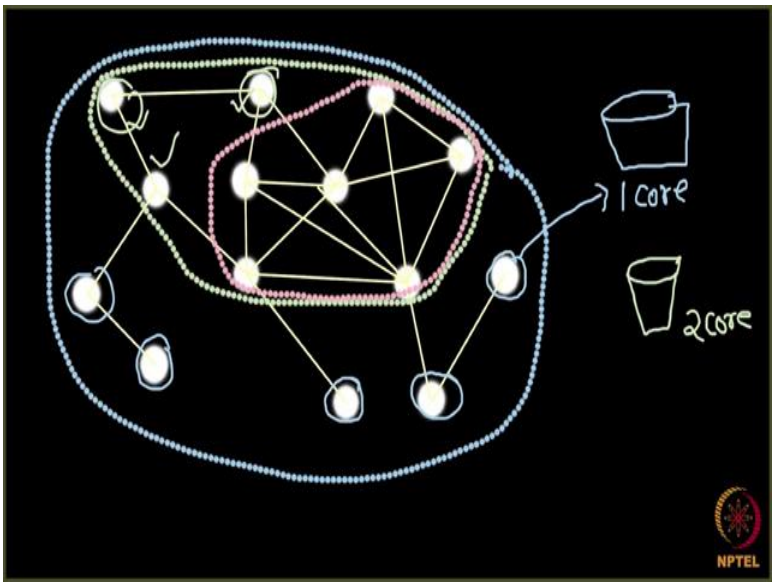
So, in this network if you see over here what is 1 core, so the 1 core in this network is all the nodes over here, union all the nodes over here, union all the nodes over here because, whatever is in this bucket for 2 cores is definitely is trivially present in the bucket for 1 core. So, here we have all the nodes which have degree greater than or equals to 2, so automatically this will be having degree greater than or equals to 1.

So, whatever is in this bucket is automatically in 1 core, whatever is there in this market is automatically in 1 core. Similarly, whatever is there in this bucket is automatically in 2 cores. So, the 2 cores of the network is, so if I want to write 2 cores in this network is so 2 cores is bucket 2 union bucket 3.

And if I want to write 3 cores in this network is 3 cores is my simply bucket 3. If you want to generalize this formula and write; if you want to generalize the formula and write it down we can say that when we try to determine the K core in a network with the help of K shell decomposition, we can say that K core of the network is, what is K core of the network is? It is bucket k, let us write it like this, bucket k union the nodes in bucket k plus 1 union the nodes in bucket k + 2 so on and so forth; that it is the $\bigcup_{j >= k} B(j)$.

So, that is how would how you can determine the K core in network with the help of K shell decomposition.

(Refer Slide Time: 15:41)



So, the overall motive of this video was to tell you that the buckets here do not correspond to the actual core numbers here; so, if you actual K core here. So, if you want to determine 1 core in this network, you cannot simply take the nodes which are present in bucket 1, you have to union it with the nodes in the bucket 2 and bucket 3 so on and so forth till the maximum bucket.