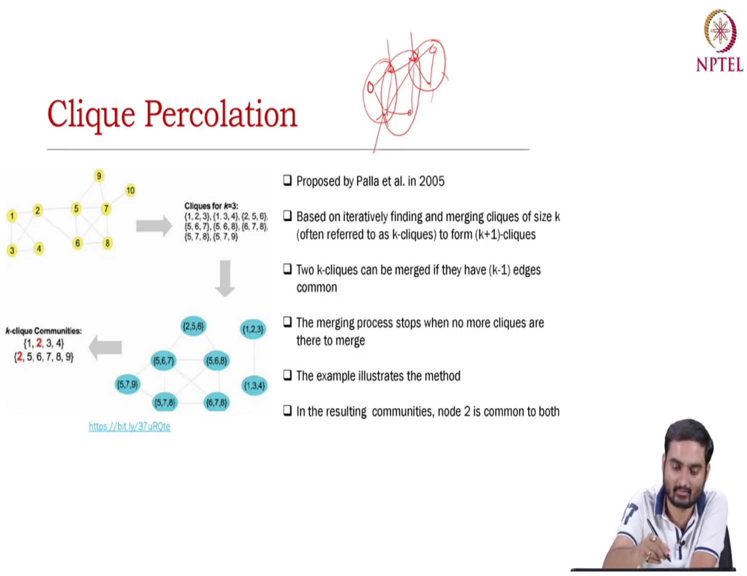
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 05
Lecture - 07
Social Network Analysis

We have already discussed you know modularity, permanence, and you know how we can use modularity and permanence for disjoint community detection. We have also seen the limitations, right. So, now, we start you know to discuss techniques for overlapping community detection, ok.

Now, what is overlapping communities? We already discussed.

(Refer Slide Time: 00:43)



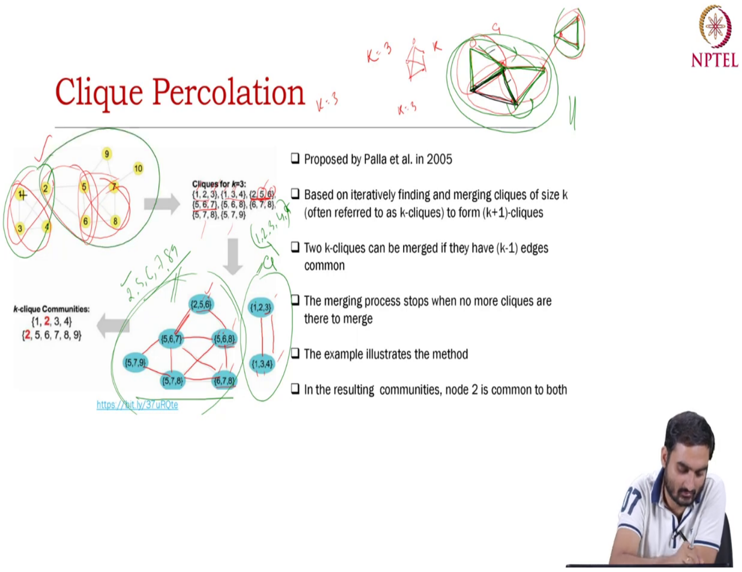
In overlapping communities, a node can be a part of multiple communities, right. For example, if you think of this as a network, this can be one community, this can be another community, this can be another community. You see that this node belongs to two communities, these 2 nodes also belong to two communities and so on, ok.

And we also mentioned that why overlapping community detection is even difficult than disjoint community detection, ok because of the exponential number of possible solutions.

So, the first technique that we will discuss is called clique percolation method. And this is I think this was the first method proposed I think way back 2005 by Palla et al. And this was published in Nature, and with this method you can detect overlapping communities, ok.

So, let us see the you know the technique here.

(Refer Slide Time: 01:47)



So, essentially let us assume that you have a graph like this what you do, you define something called clique of size k. What is clique?We know. Clique is a completely connected graph. And the size of the clique is k meaning there are k nodes, ok.

And what is the task here? The task is first to identify this k-cliques, cliques of size k, k-cliques. And then we basically club this cliques into one community, we keep clubbing clips into one community and we will detect overlapping communities, ok.

So, let us assume that k equals to 3. So, we are interested to detect cliques of size 3. So, in this graph you know this can be clique of size 3, this can also be a clique of size 3, so 1, 2, 3; 1, 3, 4, then this one, right, then this one, this one, this one, right. So, you have 1, 2, 3, 4 5 6, 7, 8, there are 8 cliques of size 3. So, the first task is to detect cliques of size 3. Now, k can be 4, k can be 5, it is a it is a hyper parameter you know that you can set beforehand.

So, once you detect cliques, remember clique detection itself is challenging, right. And let us assume that we have the liberty we have the you know, the resource to detect cliques and we identify cliques.

What is the next stage? The next stage we will see that we will choose a pair of cliques in which two in which k minus nodes are common. We choose a pair of cliques in which k minus 1 number of nodes are common. So, this cliques are of size k. We choose a pair of cliques such that there are k minus nodes common in both the cliques, ok.

So, here k equals to 3. So, we choose pairs of nodes, pairs of cliques where 2 nodes are common. So, for example, 1, 2, 3; 1, 3, 4, you see that 1 and 2 they are common, right.

Then, 2, 5, 6; 2, 5, 6, and you know 5 6 7, 2 nodes are common 5 and 6. So, what we do, so once we identify cliques we create a graph, another graph where cliques are nodes. So, you have 8 nodes, you have 8 nodes and 2 nodes are connected, meaning two cliques are connected if they share two, if they share k minus 1 nodes. In this case, if they share 2 nodes, ok.

So, you see that these two cliques are connected because they share 2 nodes, 1, 3; 1, 3 are present in both the cliques. Similarly, 2, 5, 6, and 5, 6, 8; 5 and 6 are common 5, 6, 8; 6, 7, 8, you see 6 and 8 are common and so on and so forth. So, this is the graph that you can create based on the cliques and their common nodes, right.

So, what does it mean? It basically means that let us say let us say you have, right, let us say you have structure like this and let us say k equals to 3, ok. So, this is one clique, right. This is say, this is c 1, this is another clique, this is another clique, and this is another clique, ok. So, what do we mean by you know looking at two cliques which have k minus 1 common nodes?

It basically means that if you think of it graphically, it basically means that let us define, let us define you know a window, ok. A window which basically there is a template, a template like this, a template which is basically a k size clique. And then I rotate the template, I rotate the template, so that in the during the rotation the k minus 1 nodes will remain same.

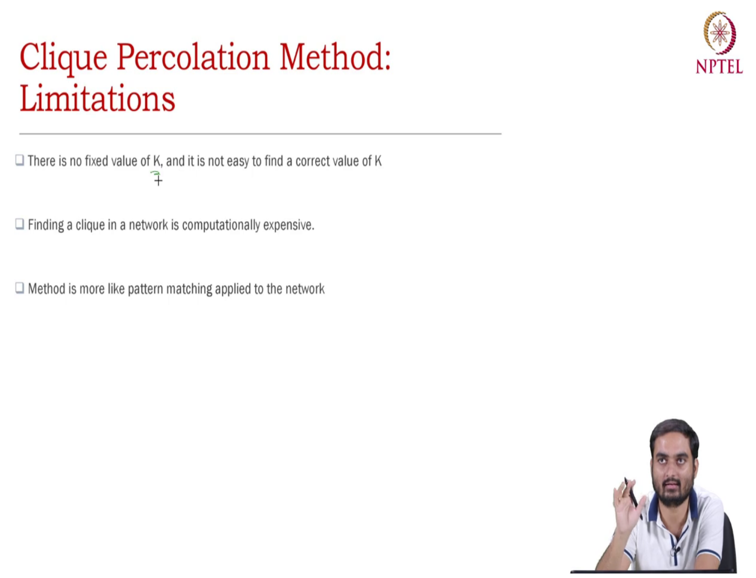
Meaning say this is the template now I rotate the template, ok. Then, the in the next, in the second stage the template will come here because between these two stages there are 2 nodes common, right. So, I rotate the template and I then you know capture some nodes and I group them. I keep on rotating. Then, so this is the second stage.

In the third stage, what you do? You further rotate it, right using the same constant. So, it will come here. You again group. You try rotating again, but you cannot rotate because now if you rotate there will be no such rotation where 2 nodes are common. So, you stop the rotation here, right. And the nodes that you have encountered that will form a community.

Similarly, you place this template in another such clique and you rotate it. So, you cannot rotate further. So, this will become a clique. So, this is same as saying that you connect two cliques when there are there are k minus 1 number of nodes common, right.

So, in the resultant graph, the number of connected components are your communities. So, you have one connected component, another connected component, so you have two communities. So, in the first community, in this community c 1 you have 1, 2, 3, 4, 5 not; so 1, 2, 3, 4, 4 nodes; 1, 2, 3, 4. In the second you know this in the second community, you have 2, 5, 6, 7, 8 and 9, ok.

(Refer Slide Time: 08:28)

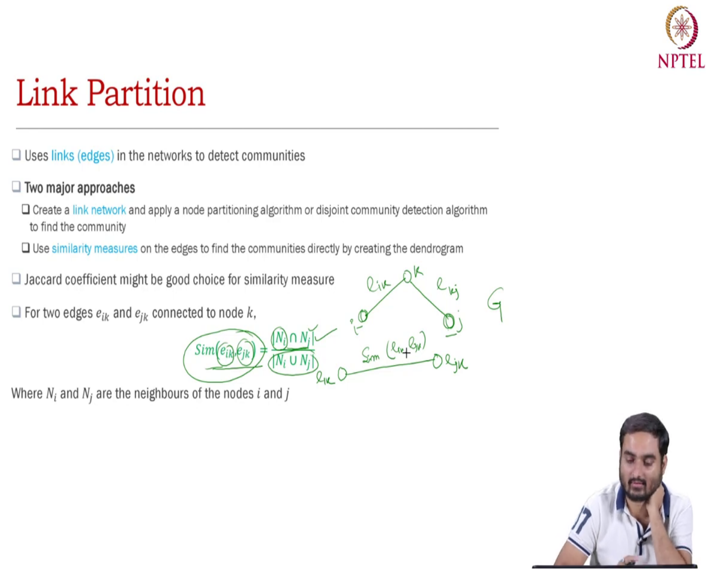


So, this is pretty simple. Basically the idea is that you start with a clique detection method, then you connect cliques, create a network, and you basically take the you know connected components as different communities and that would give you the overlapping community. Because you see here you know which nodes are common; here when which nodes are overlapping 2, node 2.

And what else? Only node 2, right. Yeah only node 2 is common, overlapping. So, node 2 is basically an overlapping node. You can basically get these two communities.

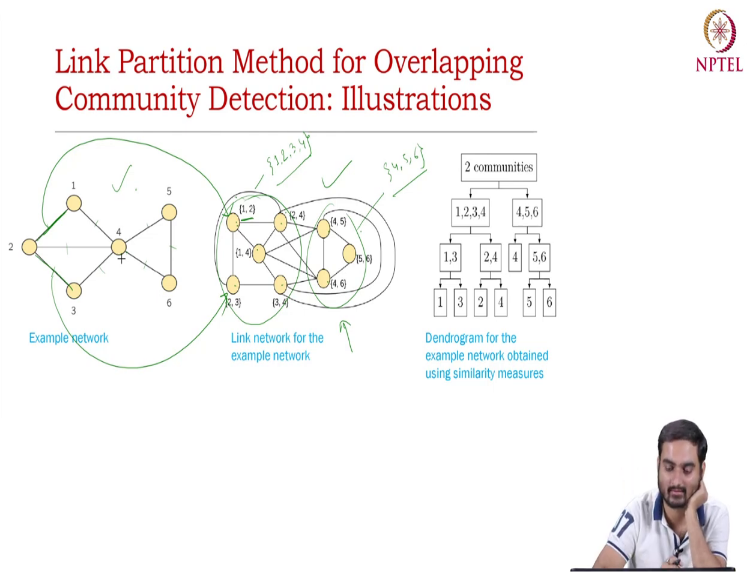
This is clique percolation. So, first problem in clique percolation is that you have to fix the value of k. The second problem is that you know clique, finding clique is also computationally expensive and it is essentially a pattern matching algorithm, because you are kind of trying to find out, cliques trying to find out a clique like patterns in a graph, right. And you are grouping them, ok.

(Refer Slide Time: 09:36)



So, next method is called link partition. So, what is link partition? A link partition, we basically create a from a normal graph, we create a dual graph. What is a dual graph? In a dual graph the edges in the original graph will form nodes in the dual graph, ok. Let us take an example.

(Refer Slide Time: 10:04)



So, this is the original graph. And you have 1, 2, 3, 4 5 6, 7, 8, edges. In the dual graph you will have 8 nodes, right 1, 2, 3, 4 5 6, 7, 8, right. You see 1, 2, so this edge is this node, right. 2, 3 this edge is this node. So, this is called dual graph. Sometimes this is also called line graph, but this is basically dual graph.

And then what you do; you in the dual graph, how do you connect nodes? You basically connect 2 nodes, right based on the original structure. So, let us say this these are two edges e ik and e jk. These are two edges in the original graph. So, therefore, these are 2 nodes in the dual graph.

So, the similarity between 2 these 2 nodes will be computed by the neighborhood intersection. Meaning, that now what are these 2 nodes, rights think about it. So, these two edges you have i, k and j; e ik e kj, right. So, essentially you have these 2 nodes, i and j and k is a common node.

So, what I will see? I will see the neighborhood of i and the neighborhood of j, ok. And I will basically take the Jaccard coefficient, intersection of neighborhood of i and intersection of neighborhood of j in the actual graph G. Intersection of the neighborhood of i and j and the union of neighborhood of i and j, and that is the similarity, right.

And I use this similarity score as the weight in the dual graph, where I have e ik as a node and e jk as a node, and then you connect it, and the weight would be the similarity of e ik and e jk that you derive from the original graph, right. So, this is the graph structure.

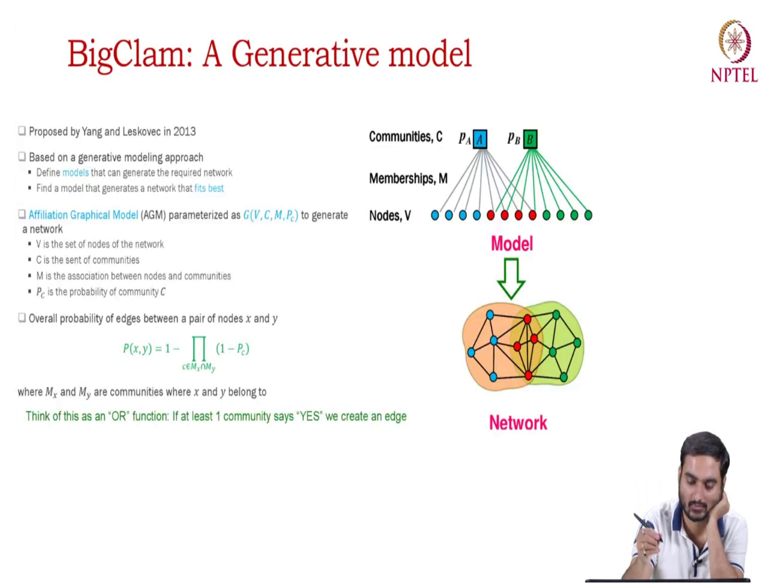
Now, in the resulting graph, in the dual graph, you run any disjoint community detection algorithm. You can run Louvain, you can run max form, you can run fast greedy. What it would do? It would basically group these nodes, right. So, if you group, say for example, if you group this 5 nodes together and these 3 nodes together, right in the dual graph now you unfold it. So, in the original graph, what would happen?

So, let us see what are the nodes which have been covered in this community. In this community, you will have 1, 2, right 3 and 4, in this community we have 4, 5 and 6, ok. So, these are the overlapping communities in the original graph. You see that this node 4 is basically overlapping, right.

So, what we have done? We have constructed a new graph called line graph or dual graph. On the dual graph you run a disjoint commutation algorithm, you group nodes, that would in turn group edges in the original graph, right, and then you unfold it.

This is you know a link partitioning method for community, for overlapping community detection, ok.

(Refer Slide Time: 13:43)



So, we will discuss in the next lecture we will discuss BigClam. Now, BigClam is so far I think one of the sophisticated methods for overlapping community detection. And this is also, I would say this is also new paradigm of algorithms which talk about generative methods for community detection. So, BigClam is a generative method for overlapping community detection.

So, in the next lecture, we will discuss BigClam.

Thank you.