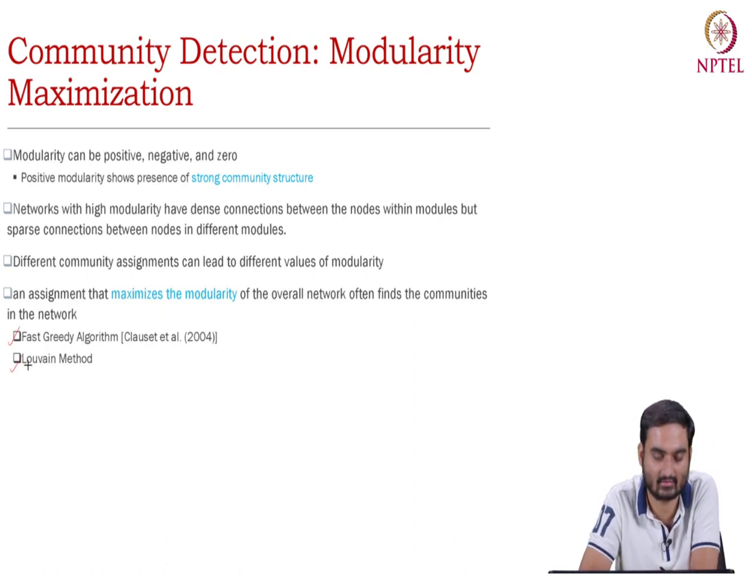
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 05
Lecture - 05
Social Network Analysis

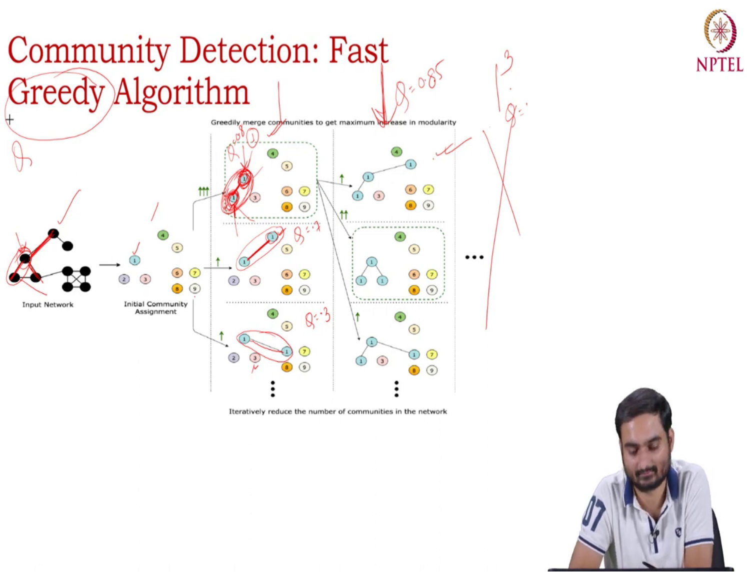
So, since we have understood what is modularity, let us see how we can use modularity for coming the detection, ok. So, you know it is, so there are actually lots of algorithms which used modularity for community detection, be it decision community or overlapping community.

(Refer Slide Time: 00:43)



But here we will discuss only two algorithms which are very famous. One is called Fast Greedy Algorithm and another is Louvain Method, ok.

(Refer Slide Time: 00:52)



So, fast greedy algorithm is proposed by again Newman and his team. So, here the idea is very simple. You have a network like this, and remember our quality metric is Q. We try to maximize Q, ok. And how to start? So, we basically start by assigning you know all nodes with individual communities, ok. You see here the colours indicate different communities; each of the node belongs to a separate community. So, there are 9 nodes and 9 different communities, ok.

And what we do? We then start clubbing nodes into groups, right. And how do we club? We how do we decide when to club, when not to club? So, we decide based on this modularity formula, ok. So, let us say we start from 1, ok. So, we start from 1 and we know that the you know the node we start from 1, right, this is node 1 and we know that this is connected to this one, this one and this one, ok.

So, this node is connected to this one, this node, and let us say this node, let us say this node, ok, does not matter. So, what we will do? We will actually place this edge, we will add this edge. So, we start by you know by taking only the nodes, right we start by taking only the nodes without any edges, right. So, nodes are there and nodes belong to different communities. And then we start from a seed node. So, let us say this is a seed node, ok.

And then we see that the seed node, the seed nodes is connected to which other nodes. So, it is connected to this one, it is connected to this one, and it is connected to this one, ok. So, we add this edge, ok. And we assume that the other node, right also belongs to the same community where node i, node 1 belongs to, right as if you are grouping these two nodes.

Once you group these nodes, these two nodes you can measure Q. Now, because now you have this community and all the other nodes belong to different different communities. So, this is option 1. What is option 2? And let us say option 1 produces modularity 0.8. What is option 2? Option 2, you know in option 2 you now connect these two nodes because you know that 1 is connected to this one, so let us connect this one. And let us club this together.

You can measure Q. Say; let us say Q is 0.7. And you connect this to say this one, right then you club and Q would be say 0.3. So, the there are 3 options, ok because node 1, the this seed node has 3 connections, right. So, which one produces the best Q value? This one, right; so, I freeze this one. So, I freeze this group, right.

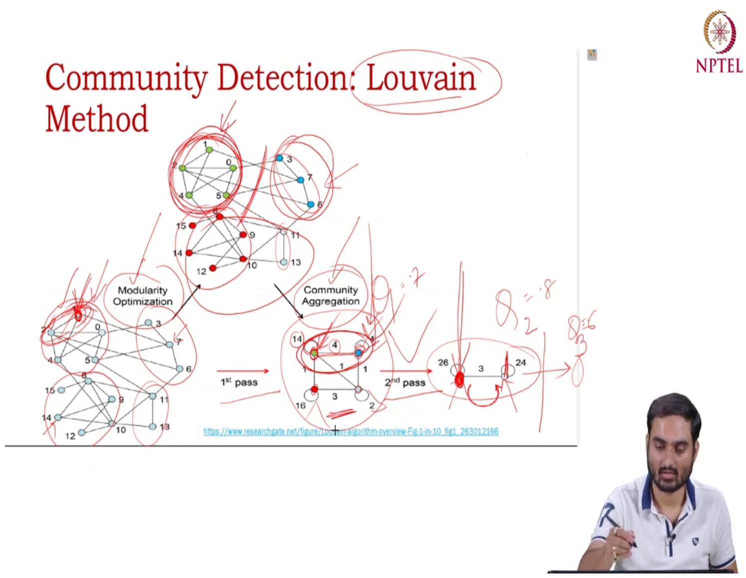
So, I freeze that this node now belongs to the same community where this node belongs to, ok. So, once we are done, then we will take the; now in our hand we have two nodes, right and their associated edges, their associated. So, now, these two nodes are clubbed, we also have, now we have their associated edges. We repeat the same process, right.

So, we take we connect an edge, we connect two nodes by an edge, group it, check the quality, and we freeze that group for which the Q is maximum, ok. Now, we keep on doing this thing until and unless we see that the Q value decreases. So, this is iteration 1, this is iteration 2, we will have iteration 3, right. So, iteration 2 we will see the maximum let us see, let us assume that the maximum modularity value is 0.8, 0.85, in iteration 3 you see that the maximum modularity value can be 0.7, right.

So, you will not proceed. You will stop here. So, you start from a particular seed node and you stop here and we take the best grouping which you obtain in this iteration, ok. So, this is the first step. Now, again let us look at, let us take another seed node, right which has not been chosen or not been encountered so far, ok.

So, we you, now choose another seed node and you repeat the process, and we basically we keep on repeating the process until and unless we see that you know all the nodes have been encountered and Q value reaches it is maximal value, right. Now, this is greedy algorithm, this is fast greedy, right this is greedy algorithm. So, there is no guarantee that it will actually reach the optimal solution, the maximum Q value, but you know in practice this works pretty well.

(Refer Slide Time: 06:28)



Now, as you understand first greedy you know has problems like this is kind of very ad hoc way and you also need to decide you have different realizations, right you have different iterations. So, this has problems. So, in 2008-2009 if I remember correctly this method was proposed Louvain method, Louvain algorithm which is I think by far one of the best methods for disjoint community deduction using modularity maximization, ok.

So, in Louvain method there are two steps. The first step is called modularity optimization and the second step is called community aggregation. So, at every iteration we will have these two steps, modulating maximization, community aggregation, right. So, let us see how this method works.

So, this is the network. What we do here. We basically you know start off by you know again adding assigning nodes into different communities, ok and then we again start with a particular node. Let us say we start with this node, ok. So, and what you do, the in the same way the way we actually did you know the grouping in case of fast greedy, here also we do the same thing. But here we will not remove all the edges and then we start adding. We will not do that. What we will do?

We will keep the actual graph, ok and let us say this is our seed node, ok. And then, we say you know say the seed node has this neighbour, this, this one and we have 3 neighbours. So, we group the this neighbour, we group this neighbour and we group this neighbours, right in the same manner. And then, we see that for which grouping the modularity value is maximum. Let us say, let us say you know with this grouping, with this grouping the modularity value is maximum, we freeze it, ok.

Then, in our plate now we have two nodes and they are neighbours we keep on grouping. We keep on grouping by maximizing modularity, right. And let us say let us assume that from this seed node we get a group like this, ok. Then, we again choose another seed node which has not been explored so far and we the then we keep on grouping. Choose another seed node and group; choose another seed node and group, ok. So, then we see that we reach the maximum modularity value, ok.

So, this is our grouping after modularity maximization, ok. So, you see that there are 4 colours indicating 4 groups, ok. In the 6 see in the second you know stage we use something called community aggregation. So, what is community aggregation? In community aggregation, we freeze a particular community into a node. We call this node as a super node. We create a super network; this is a super network, right a summarized network where each node indicates a community that we obtained earlier, right.

So, we have this community as a node here, this green node, we have this as a node here, this red community as a node here, and this this as a node here, ok. And how do we connect these nodes? So, number of edges within a community, intra community edges will basically be denoted by self-loop and the weight of this self-loop is the number of intra community edges within this green community.

And between this green and sky communities, right, so we have edges, right. And the this edge, and the edge has a weight, right and this weight is basically the number of inter community edges between green community and blue community, sky community. Similarly, we draw edges for the other network for the other nodes, other super nodes. So, this is our super community, super network.

Now, in this super network we will again repeat the same process modularity optimization and community aggregation. Let us say in the, so in the first pass we have two steps, in the second pass also we have two steps, right. Let us assume that after the second pass we have these kind of summarized super network, ok.

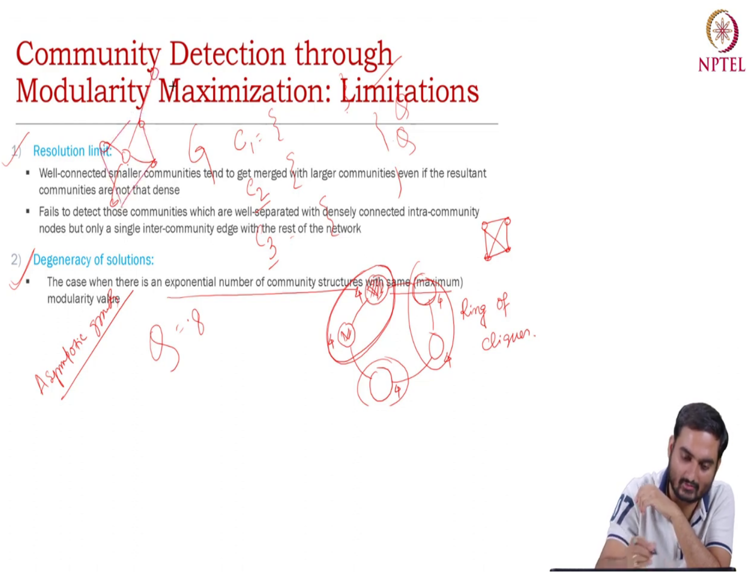
We again repeat the same process, right. Remember at every stage we have certain Q value, this is Q 1 modularity value; this is Q 2, right. After step stage 3, we have Q 3. When do we stop? We stop when we see that at two consecutive passes the modularity value will not increase. So, this is 0.7, this is 0.8, but say this is 0.6. So, we actually stop here.

So, we now we know that there are two communities, the green community and this you know ash colour community, ok. Then, we unfold it because we know this is a super node and this super node was formed based on say these two nodes in the previous iteration and these two nodes are formed based on these two communities, right.

So, it basically means that you know this community; this community should have this nodes and this nodes. This community should have this nodes and this nodes together, ok. You keep on aggregating, you stop when there is no further improvement, right you stop there and then you unfold it to get the actual community structure. Remember, using this process you also explode the hierarchical structure because you want to stop here.

Let us say you want to get 4 communities, right and the 4 communities have already been obtained. So, we stop, we can also stop here. But if you want to club you know club nodes into groups you know and so on and so forth, you keep clubbing, keep grouping, then you can go on until unless you see that modularity value will not increase further, ok. So, this is Louvain method.

(Refer Slide Time: 13:17)



Now, let us try to understand the limitations of modularity maximization. So, there is a very interesting paper, which talks about you know limits of modulating maximization, and there are 3 essentially 3 limitations. So, the first limitation is called resolution limit. What is resolution limit? Resolution limit is a problem of modularity maximization algorithm. Now, modularity maximization can be done using Louvain using fast greedy using some other methods, right. You can also do it exhaustively.

The problem in modularity maximization is that these kind of algorithms tend to detect a large size communities, they tend to club small size communities into groups into bigger groups, right. A best example is ring of clicks, ok. Let us say let us say these are 5 clicks, remember these are clicks. So, you have, so you have nodes and they are completely connected clicks, ok. And let us say this is the size of the click is 4 for example. So, all these clicks are of size 4; 4 nodes and they are connected. So, this is a this is a 4 size click.

And 2 consecutive clicks are connected by a single edge, like this, ok. This is called ring of clicks, ring of clicks, ok. So, if you run modulating maximization on ring of clicks, what would happen is that the algorithm will group clicks, 2 consecutive clicks together to form a bigger community. The algorithm would not be able to detect this clicks.

Ideally, what would what should happen? Ideally, all these clicks should be detected as separate communities, right. But what the modularity will do? Modularity maximization will you know will club 2 consecutive clicks into 1 group, and you may end up having you know this kind of communities, 3 communities. This is called resolution limit.

You have a limit on the size of the, you have a limit on the size of the community structure. It would not be able to detect a community below a certain threshold, ok.

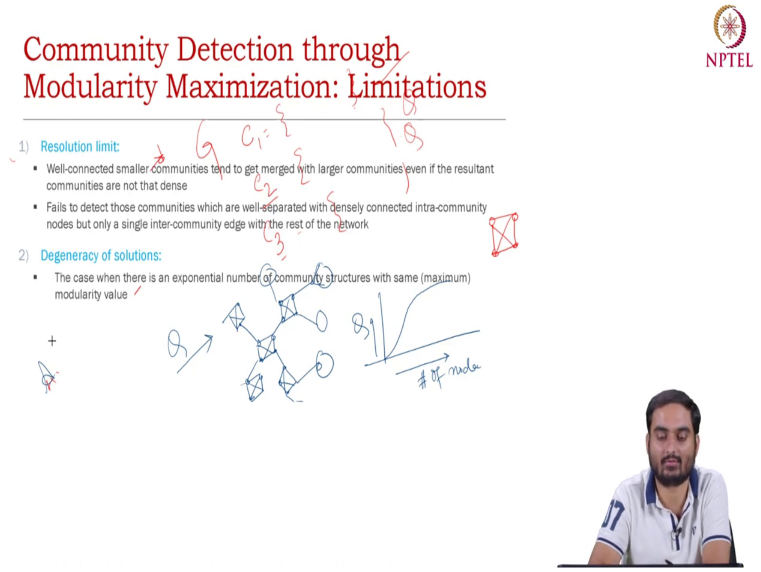
The second problem is called degeneracy of solution. Meaning, that given a particular modularity value Q, say let us say Q is 0.8, right given a maximum modularity value 0.8 say, you can have exponentially exponential number of solutions. Meaning, that is say you have a graph G, you can create community structure C 1, community structure C 2, community structure C 3.

When I say community structure meaning that you have communities within the community structure, you have a groups, right. C 2 has a different community structure; C 3 has a different community structure. You have different community structures with the same modularity value 0.8 say, right. You have exponential number of community structures with the same modularity value.

And there is no there is no point in choosing one, there is no preference of choosing one community over another because all the community structures are giving same value, same modularity values Q. So, there is no preference of choosing one community structure over another. Well, that should not happen, right.

So, it should always detect a unique solution ideally which also produces the maximum you know Q value. So, this is called degeneracy of solution. These two limits limitations are the major limitations. There is another limitation called asymptotic growth, ok asymptotic growth. What is asymptotic growth? It basically says that let us say you have a network, right a kind of a self-similar network like this, right like this. So, let us say you have a network like this. So, let me draw it afresh.

(Refer Slide Time: 17:55)



So, let us say you have a network like this where the network is growing symmetrically; the network is growing symmetrically like this, ok. You have same structure, click, and they are further growing, this is another click, another click, another 4 click, right like this, a 4 click, 4 click like this. The network is growing symmetrically.

Ideally, what should happen? Ideally, since it is growing symmetrically the Q value should not increase, ok. But what happens is that as you increase the size of the network, Q value also increases. So, you see that if you plot Q value, this is Q value and this is number of nodes, number of nodes, you will see that as the node size at the size of the network increases, the Q value starts increasing. It should not happen, right.

So, this is called the asymptotic growth of the value of the modularity, ok. So, these are the 3 limitations of modularity maximization. We stop here. So, in the next lecture, we will discuss other methods. We will see some other, some better metrics that I mean one can use to detect you know community structure.

Thank you.