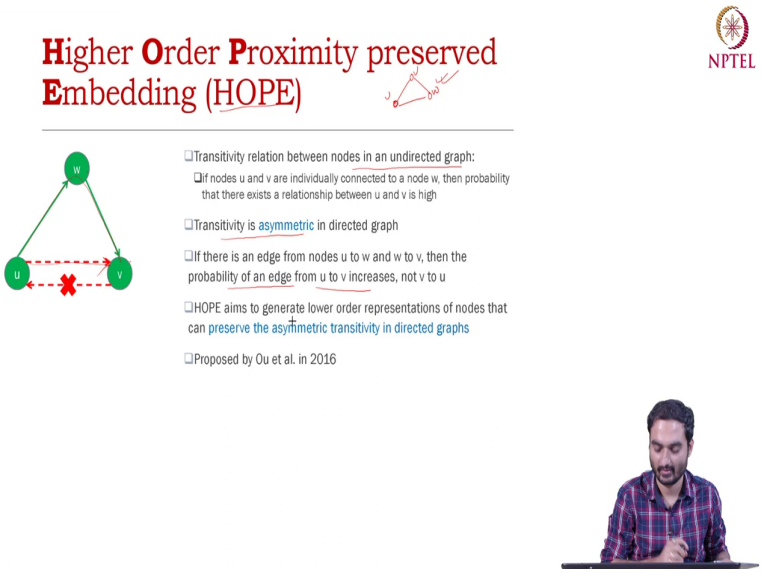
Social Network Analysis
Prof. Tanmoy Chakraborty
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology, Delhi

Chapter - 09
Lecture - 06

So, we are continuing with our discussion on graph reputation learning; particularly in the last lecture we have started discussing matrix factorization based approaches for graph learning. And we have seen GraRep which is one of such matrix factorization based approaches. And today we will discuss another algorithm which is called HOPE Higher Order proximity Preserving Embedding ok.

(Refer Slide Time: 00:52)



So, what is the intuition behind this algorithm? So, if you look at the network right. In general most of the networks follow some sort of transitivity relations right I mean if u is connected to v and v is connected to w there is high possibility that u and w would also be connected right. So, if you look at the algorithms which are proposed before HOPE right.

So, all these algorithms were mostly designed for undirected network which basically is I mean those kind of algorithms are ok when we have an undirected graph right. And naturally this kind of symmetric transitivity right preserves this kind of symmetric transitivity is

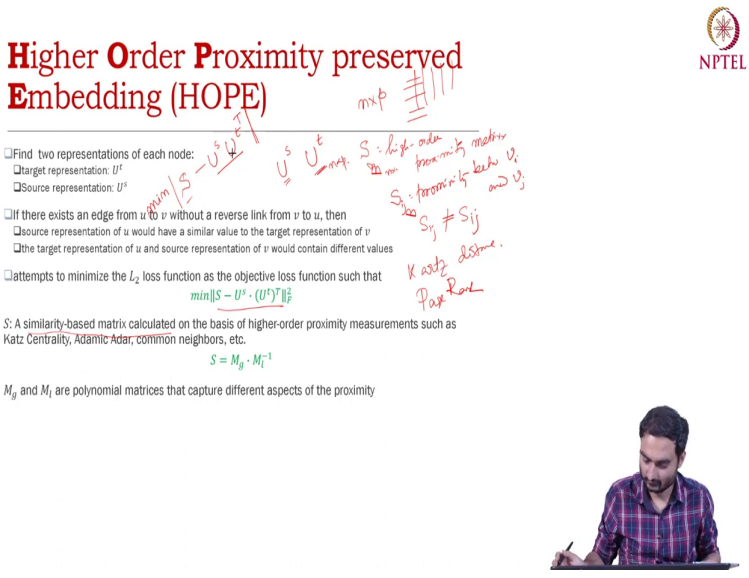
preserved in those I mean this kind of symmetric transitivity has been preserved by the algorithms.

But if you think of a directed graph right. Say for example, here u is connected to w, w is connected to v there is a directed edge there is a directed edge from u to w and another directed edge from w to v right. Then if you imagine right it is highly likely that there might be a relation from u to v right because of this structure there might be a connection from u to v, but not v to u.

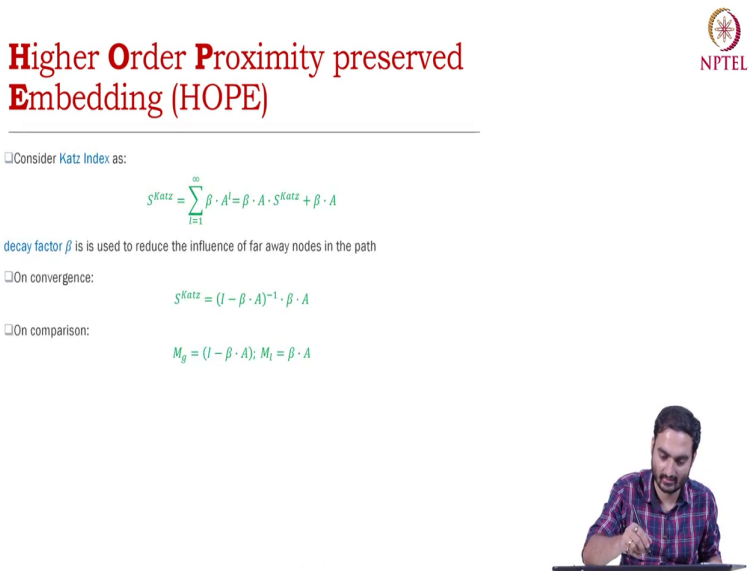
So, in a directed graph right transitivity property is asymmetric ok and this is the whole story behind this algorithm right; transitivity relation between nodes in an undirected graph is ok. But transitivity is asymmetric in a directed graph if there is an edge from node u to w as I mentioned here and w to v then the probability of an edge from u to v will increase, but not the other way around ok.

So, HOPE algorithm this aims to generate lower order representations of nodes that can preserve the asymmetric transitivity in directed graphs ok. So, this is the overall idea and this was proposed in 2016 in KDT ok.

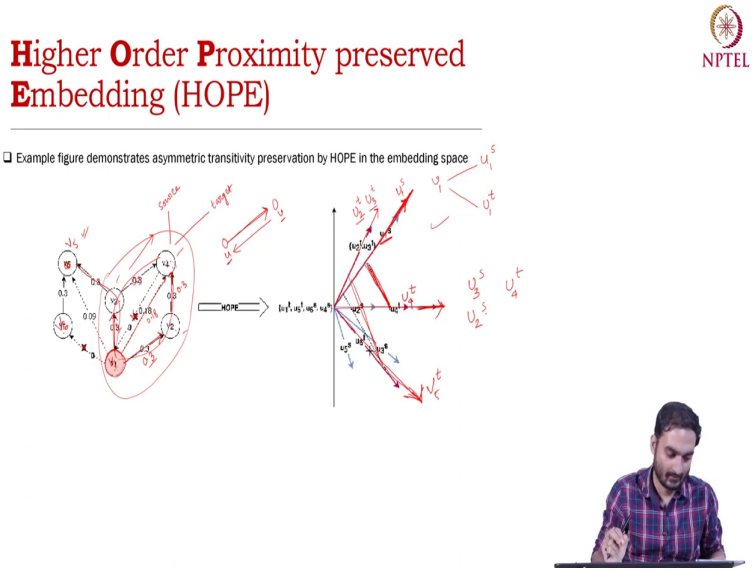
(Refer Slide Time: 03:21)



(Refer Slide Time: 03:28)



(Refer Slide Time: 03:30)



So, let me demonstrate it in using a figure ok. This figures may not be very visible, but I will try to explain what this figure indicates ok. So, you see here from v 1 there is an edge there is a directed edge right from to v 2 with certain weight 0.3. This weight is essentially some sort of proximity ok.

This proximity can be a Katz proximity or an or a proximity which is already given say a weight right. There is another directed link from v 2 to v 4 with 0.3 right. Let us look at only

this part. There is a link from v 1 to v 3 with 0.3 v 3 to v 4 with 0.3. So, you see from v 1 to v 4 there are two paths there are two directed paths one through v 3 and another through v 2 ok.

So, it is natural that there might be a path there might be an edge from v 1 to v 4 which is highly likely, but from v 4 to v 1 there is an edge this is less likely ok. So, you may see an edge from v 1 to v 4 with say some proximity 0.18 these are dummy proximities ok, but from v 4 to v 1 there should not be any edge ok. So, if this is the graph if you look at the embedding space right.

In the embedding space what should happen? In the embedding space of course, v 3 and v 2 they would be closer to v 1 right v 4 would also be closer to v 1, but in a different aspect ok, I will discuss the aspect here. So, what they proposed in this algorithm is that every node right, for every node there should be two embeddings ok one embedding is called the source embedding and the other is called the target embedding. Why? Say there is an edge from u to v ok.

So, with respect to this edge u is acting as a source and v is acting as a target whereas, if you an edge from v to u then v is acting as a source and u is acting as a target ok. So, what they suggested is that every node is associated with two embeddings one represents that node as a source node other represents that node as a target node ok.

So, if you think of this case as I mentioned v 4 is also close to v 1, but v 4s target embedding is close to should be close to v 1es source embedding whereas, v 4s source and embedding should not be close to v 1s target embedding which basically indirectly says that there should be an edge from v 1 to v 4, but not v 4 to v 1 ok.

This is the embedding space as you see here these arrows are basically vectors right again dummy example right you see. So, this is the notation that they are following let us say v 1 is a node and u 1 s is a source embedding of node v 1 and u 1 t is the target embedding of node v 1 ok.

So, you see here let us say this edge ok this edge is u 1 s meaning v 1 s source embedding right. You see that this vector is closest to this vector ok and what is this vector? This vector is u 2 t and u 3 t meaning 2 and 3 2 and 3 as the target nodes right. Therefore, you see there is a directed edge from v 1 to v 3 and v 1 to v 2 right and both the proximities are same. Therefore, u 2 t and u 3 t should be closest to u 1 s right.

And you also see that v 3. So, v 3 and v 2s source embedding meaning u 2s u 3 s and u 2 s ok they should also be; they should also be closest to v 4 v 4s target embedding which is u 4 t right. Similarly v 4s target embedding right you see here this is v 4 right this is v 4s target embedding right this is also closer to u 1 s is also closer to u 1 s ok right. If you note it here another thing to note is very important. So, look at here. So, this is v 6 ok there is a v 5.

You see that there are two directed paths from v 1 to v 4 right, but if you think of here there is a path from v 1 to v 3 to v 5 there is a path from v 5 this is v 6 to v 5 right, but there is no paths there is no edge from v 1 to v 6 right. So, v 5 as a target is also close to v 1 right due to this path, but this is not as close as v 4. Why? Because v 4 is now connected to v 1 through two paths two shortest paths right so, this is v 5's target embedding ok.

So, if you look at v 1s source embedding this is v 1s source embedding ok this one and this is v 5s target embedding and this is v 4s target embedding. So, you see that this is closer than this one. So, the bottom line story is that if there are lots of such shortest paths between two nodes right between u and v although u and v are not directly connected they should be closer they should be close in the embedding space ok.

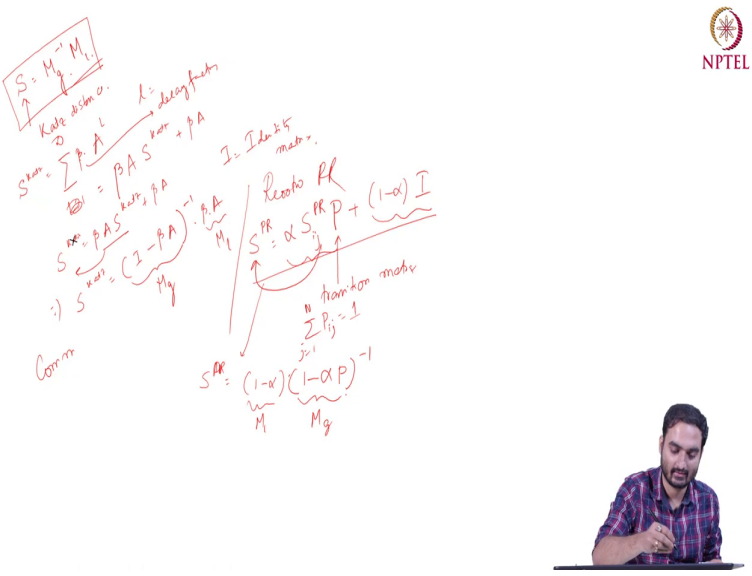
So, now this is the story now let us look at the formulation. So, what they suggest is as follows. So, they say that let us define let us define a high order proximity matrix yes this is a; this is a high order proximity matrix. Now how do you calculate it I am coming to that ok. So, this high order proximity matrix can be calculated using say PageRank. Now so, each entry S i j indicates the proximity between v i and v j. Remember this is not symmetric right.

Meaning v i j should be not equal S i j is not equals to S j i ok. So, now, this can be calculated in many ways. For example, this can be calculated using Katz distance, we discussed what is Katz centrality the same as Katz distance rise we can also use PageRank. We will discuss PageRank types algorithm we can also use common neighborhood say for example, right and so on and so forth.

So, they also defined two other matrices; one is called U s and other is U t. So, U s is the n cross p matrix whereas, n is the number of rows number of nodes and p is the number of features ok embeddings. So, U s consists of all nodes acting as consists of all embeddings of nodes acting as source nodes whereas, U t consists of embeddings of all nodes acting as target nodes same as this is also in cross p ok.

So, and S is n cross n right S is n cross n. So, what is the idea the idea is that you can factorize S into U s and U t ok of course, transpose right. And your target would be to factorize s meaning to minimize the distance between two right in order to obtain the embeddings this is the target ok.

(Refer Slide Time: 14:41)



So, then what they suggested is that every higher order matrix every higher order proximity matrix S can be decomposed into what they call two polynomials of matrices. One is M g inverse M l let us try to understand it let us focus on it.

So, let us say this S is a as I mentioned this is a proximity matrix and let us assume that this is Katz distance ok. What is Katz distance? We have already discussed S say S Katz is beta which is the decay parameter the damping factor right A to the power l where l equals to 1 to infinity l is what? l is the distance.

So, you are multiplying A with itself 1 time 2 times when you multiply A with itself A square meaning you are looking at 2 hop then 3 hop and 4 hop and so on right and you are looking at infinite distances. So, this can be decomposed into this, this can be written as beta A and what is beta? Beta is decay factor as you move along from 1 hop to 2 hops to 3 hops l l will regulate the you know decrease in importance ok.

So, plus beta A we can write it right recursive formulation and so, you have a S Katz is beta A beta A right you move this thing here and then you I mean calculate what you get? You get

S Katz equal to I, I is identity matrix right I minus beta A ok. As I mentioned earlier every S can be decomposed into this you see here this is your M g and this is M l ok you see that when S is defined in terms of Katz distance you can easily get this kind of formulation.

Let us see another such a higher order proximity notation I mean formation which is called rooted PageRank. So, rooted PageRank is same as random walk with PageRank ok essentially what it says is that is a. So, this rooted PageRank is the probability that a random walk from node v i is located at v j in the steady state ok and how do you define this? S of PageRank is alpha with alpha probability you basically move right.

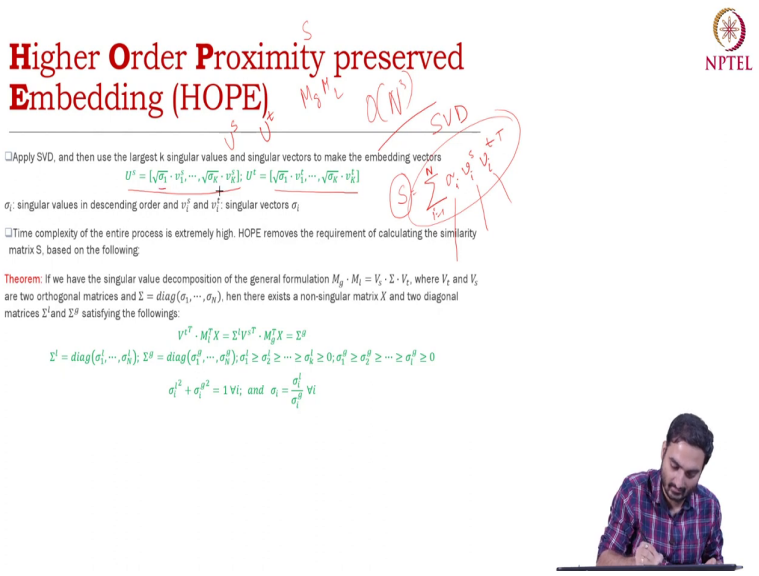
So, since we are only calculating between i and j. So, S i j pr right with transition probability P plus 1 minus alpha I this takes care of the restart probability and this takes care of the jumping probability right. What is P? P is the transition probability transition matrix, stochastic matrix ok. Transition matrix such as the this is stochastic. So, row the sum of row sum of entries of each row should be 1 right, say i j is equals to 1 to n should be 1 ok.

So, this part if you write in terms of matrix right this would be you see here you can move this thing out here and then the remaining part will is to be straight forward. So, sorry. So, this would be ok. So, here you see this is M g and this is M l. So, what they suggested is that.

So, they have actually looked at four such higher order proximity measurement formulation one is Katz other is rooted PageRank the other one. So, this is also rooted PageRank is very same as random work is registered also same as personalized PageRank right and so on.

They have also looked at common neighborhood right and they have also shown that you can decompose into S into M g M l and they have also looked at atomic at a distance ok. So, if this is true then what is the actual formulation right, let us try to understand it now. So, now, if this is this holds; that means, so; that means, you can essentially solve this problem using SVD right.

(Refer Slide Time: 20:48)



You can decompose S right you can decompose S right with SVD singular value decomposition we discussed what is singular value decomposition in the last lecture briefly.

So, it means that you will essentially have S equals to summation of right n number of nodes this is your singular value and then you have v t v i t transpose right this is singular value and these are singular vectors right, but SVD is extremely time consuming this is essentially order of n cube right. Now think of large network this is impossible to do right. Therefore, they came up with a theorem right which basically says that you can just get rid of SVD part.

If you just calculate M g and M l because all this all the proximity based you know proximity based formulations which essentially construct S this can be decomposed into M g and M l. So, M g if you calculate M g and M l which would which is which are easy to calculate compared to calculating S right the remaining part would be straight forward right.

And they also suggested some variations of SVD which they call as Jacobian Davidson type algorithm for some sort of partial generalized SVD I am not going into details of that right. But it turned out that using this kind of approximations, you can easily come ultimately your aim is to do get this one right once when you get this one then the remaining part is same because this ultimately what is the final task.

The final task is to get U s and U t these two matrices right. So, then U s would be this one and U t would be this one ok. So, what they suggested they you first calculate mg and ml

given a particular S given a particular higher order proximity formula you calculate M g and M l and then you use this Jacobian Davidson type algorithm right and then you obtain you obtain this singular values right and from singular values you can easily get U s and U t ok.

So, this is the idea behind HOPE in the next lecture we will discuss another such algorithm which is called line which is mostly useful for large network which also kind of you know takes care of the higher order proximity ok.

Thank you.