

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

Chapter - 04
Lecture - 23
Lecture - 04

So, we discussed today about PageRank which is one of the metrics that we discussed in the 1st lecture, but briefly. So, today we will try to understand the intuition behind PageRank and why PageRank is useful and we also look at the limitations of PageRank ok. Remember the link analysis matrix that we have discussed so far. Those are useful in the context of node prediction node link prediction node classification these kind of tasks.

And these are also useful when you know we want to measure the importance of nodes or importance of any edge ok. And PageRank is one of such metrics where in which we basically look at how importance of a node flows from one node to another node and that in turn boosts the importance of the other nodes ok.

(Refer Slide Time: 01:20)

PageRank: Intuition

- ❑ Outgoing hyperlink from a page is termed as out-edge or **forward link**
- ❑ Incoming hyperlink to a page from the second one is termed as an in-edge or **backward link**
- ❑ With every forward link a page establishes,
 - ❑ it transfers some of its **importance/rank influence** to the forward page
- ❑ If a highly important node points to a lesser important one,
 - ❑ there is an **enhancement in the status** of the latter node
- ❑ **Importance** of each node is determined by its in-edges/backward links

Navadiya and Garg (2011)



So, the history I already discussed in the chapter 1 or chapter 2, but basically the idea here is that you know let us consider a directed network where a link indicates you know either some sort of citations from one webpage to another by page or whatever one citation one scientific

paper to another site scientific paper. And our task is to predict our task is to measure the importance of nodes ok.

So, let us look at some you know some notations. We use the term forward link to indicate a link from A to B for example. So, there is a link from C to D and this is a directory graph this is a forward link this is a forward link with respect to C right. This is page C this is page D. So, with respect to page C this particular link is a forward link with respect to page D this is a backward link ok.

So, outward edges indicate forward links inward edges indicate basically backward links ok. So, an importance of a node right basically spreads through its outward edges right through its you know forward links. Say for example, the importance of node C would basically move from C to D from C to F from C to E right and this in turn would increase the importance of page webpage D web page E and web page F right.

(Refer Slide Time: 03:14)

PageRank: Simple Ranking

For a node w , let F_w be the set of nodes that w points to (Forward links) and B_w be the set of node that points to w (Backward links). Further, let $N_w = |F_w|$, the number of forward links from w . Then, the simple ranking of w , denoted $R(w)$, is given by

$$R(w) = \sum_{b \in B_w} \frac{R(b)}{N_b}$$

$R(w) = \sum_{b \in B_w} \frac{R(b)}{N_b}$

 $N_b = \text{number of edges pointing to } b$

 $= |F_b|$

- The underlying web graph is assumed to be a **connected component**
- There could be pages that neither refer to any other page nor are referred to by any other page
- In a scenario where **no hyperlinks exist in the network**
 - Each page is assumed to be equally (un)important with a uniform rank given by

$$R(p) = \frac{1}{\text{\#Webpages}}$$

So, this is a very simple metric. So, basically the idea is that and basically the idea is that you know we assume that the important the importance of a node you know basically divides uniformly across these outward edges and that importance would further boost the importance of the other nodes. And we denote F_w . So, F_w is the set of forward links ok basically nodes. So, set of nodes which are which basically you know that the node w links to.

So, this is w and say w is connected to these 4 nodes say $A B C D$. So, F of w is a comma B comma C comma D ok. And similarly, we define say $E F G$. So, we define something called B_w which correspond to backward edges right. So, B_w would be $G E$ and F . Remember if w and B_w contain nodes not edges ok. So, now, if we assume that the importance or a rank right or prestige for example, of node w is R_w ok.

So, what would happen essentially is that node w acquires importance or prestige from its you know from its inward right or whatever backward neighbors right. So, w basically receives prestige from E from F and from G ok. So, so R_w is basically of sum of all the backward nodes of w which is denoted by B_w and small b is one such node R_b is the importance of one of the backward nodes b and N_b is the degree of the backward node b right.

Let us say in this case B is E ok. And say E has degree 3 and remember this degree N_b . So, N_b is the neighbors right N_b is the neighbors of b . This is basically outward edges of b . So, N_b is essentially forward edges forward nodes ok. It means that the importance of node E is equally divided into 3 and one such division one such component moves from E to w .

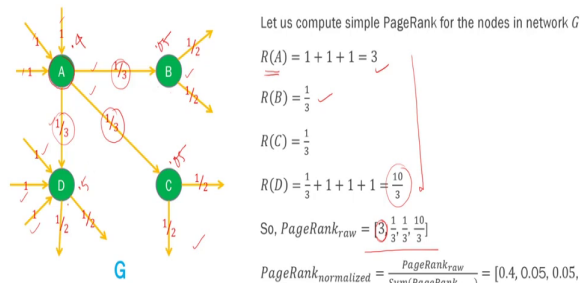
And that component would boost w 's importance right this is R_w . Now and we assume that the underlining graph is a connected graph because if this is not connected then there are some issues, but let us assume that this is connected we will we will see how we can tackle this problem. This you know disconnected components and so on ok. So, if there is no hyper if there is no hyperlink exists right. For example, say there is a there is a node w and node w does not have any inward links.

Now, remember this node w in this case. So, node w acquires prestige importance from all of its you know all of its in degree neighbors inward neighbors right in citations. Now if w does not have any incitation then how w is important will increase right. So, for those cases for those pages which do not have any inward any you know in boundaries we assume that they have some non negligible importance which is 1 by N . What is N ? N is the total number of web pages present in the graph.

So, if there are say hundred say thousand web pages. So, all the web pages initially would get 1 by 1000 1 upon 1000 prestige ok. So, therefore, for those nodes which do not have any you know inward edges they would also get some importance in terms of 1 by n ok.

(Refer Slide Time: 08:26)

Simple PageRank: Illustration



So, this is an example. So, you see that in this graph there are four nodes and let us say you know this node a right it has 3 outward edges 3 forward edges therefore, the prestige will divided equally 1 by 3 1 by 3 1 by 3. For B 2 outward edges. So, 1 by 2 and 1 by 2 and so on right.

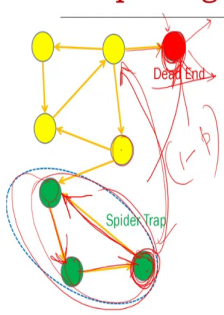
So, now, As prestige is basically 1 plus 1 plus 1 assuming that each of these edges contributes 1 unit of prestige right. So, R A is 3 what is R B? R B has only 1 inverted edges which is only 1 inverted and the important carries is 1 by 3. So, therefore, R B is 1 by 3. R C is also 1 by 3 RD is 1 plus 1 plus 1 plus 1 by 3 ok 10 by 3.

So, these are initial importance ok. So, this is a raw PageRank. So, raw PageRank of node a is 3 node B is 1 by 3 node C is 1 by 3 node D is 10 by 3 ok. What is the normalized PageRank? Normalized PageRank we want that the PageRank value or always ranges between 0 to 1. So, what we do after every stage after every iteration we take the sum of all the prestige values and we divide each component by the sum.


So, 3 would be divided by 3 plus 1 by 3 plus 1 by 3 plus 10 by 3 whatever is the number right and so on and then we will see that this would be the normalized PageRank 0.4, 0.05 0.05 0.5. You will see that the sum is 1. So, after every iteration we make sure that the sum is 1 if it is not 1 then we explicitly make it 1 by divided by the sum by the sum by divide each component by the sum ok.

And then we repeat the process again. So, now, for page A the prestige is 0.4, page B 0.05 and 0.5 ok. Then what would happen we repeat the same process again ok. And when we stop? We stop when we see that there is no you know there is no change in the there is no significant change in the PageRank values of all the pages ok.

(Refer Slide Time: 11:10)




The diagram shows a network of nodes. A red node at the top right is labeled 'Dead End' with a red arrow pointing to it. A group of three green nodes at the bottom is labeled 'Spider Trap' with a green circle around them. A red arrow labeled 'Teleportation probability' points from the spider trap area towards the dead end.



Simple PageRank: Drawbacks

- ❑ PageRank method follows a recursive approach
- ❑ Scores obtained at $(n - 1)^{th}$ iteration is used as input scores at n^{th} iteration
- ❑ The above process stumbles at two extra-ordinary situations shown in the diagram
- ❑ Scores of nodes at **dead ends** does not impact rest of the nodes in the network
- ❑ The nodes forming a **spider trap** can revise their scores indefinitely without having any impact on the rest of the nodes in the network



But what is the problem here? So, in this approach there are 2 problems the first problem is if you somehow you write you somehow reach a place reach a node which does not have any right outward edges right forward edges. What would happen? Right. You basically stop here you cannot move it further ok. So, this kind of node is called dead end right because this is a dead end you will not be able to move further there is no outward edges there is no forward edges ok.

So, if you follow this process ok this kind of process. Then after this stage you will not be I mean when you come here you will not be able to proceed further ok. The second problem is called spider trap. Let us look at this structure ok. Say you move from this node to this node and then when you come here there is only one outward edge forward edge. So, you move through this, you come here, you move through this, you come here, you move through this, come here and you basically keep on rotating.

You keep on rotating you keep on you know circulating within this structure you will not be able to move further this is also dead end right, but you are not static here. In case of dead

end you start you get stagnant, but in case of a spider trap you basically you have got into a trap ok. So, how do we overcome these two problems?


To overcome these problems we use something called teleportation probability which we have we have already discussed, but this name is something which we have not encountered will use something called teleportation probability. What is teleportation probability? So, in normal cases what we do? At every node we basically choose one of its outward edges and then we move. And then we basically choose this outward edges uniformly at random, one of the forward edges uniformly at random.

This is one probability. So, with certain probability you choose one of the outward edges with the other probability 1 minus the given probability say with probability p you choose one of the outward edges with probability $1 - p$, you can jump from one node to any other nodes in the network ok. Say for example, you are here. So, with probability p you can choose this edge with probability $1 - p$ you can move from this node to any other nodes in the network you can directly jump from this to this for example.

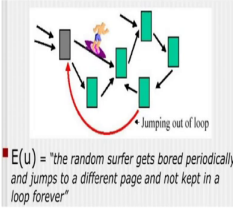
Although there is no directed there is no direct edge right. And this is called teleporting probability. So, every node at every node you have two chances; one chance you essentially you know sample 1 ; one of the outward edges right one of the forward edges in other probability you basically jump from that node to other node ok.

Therefore when you start here, there would be some probability some chance that you get out of this node right. Similarly when you trap here there is some probability you get out of the out of this node out of the structure. And this is called teleportation probability. So, through this teleporting probability we keep the random walk process going ok.

(Refer Slide Time: 15:17)



PageRank: Random Surfer Model



$E(u)$ = "the random surfer gets bored periodically and jumps to a different page and not kept in a loop forever"

<https://www.slideshare.com/tejaswini14/the-page-rank-algorithm-ranking-bringing-order-to-the-web>


- 1) Surfer starts a random page P_1 and moves to subsequent pages P_2, P_3, \dots, P_n in random order
- 2) Upon landing at a page P_i , the surfer choose either of the following
 - a) With probability α , jump to random page P_j and repeat step 2.
This random jump action is denoted $E = \frac{1}{N}$ where N is the number of pages in the network
 - b) With probability $1 - \alpha$, it continues in its course of following hyperlinks

❖ the more number of times the surfer visits a node during the above random surfing, the higher the importance of the node



So, this is the idea. And what is the teleportation probability? Teleportation probability is uniform is $\frac{1}{N}$ for all the nodes. Also you can make it different depending on your applications. We will discuss some variations of PageRank like personalized PageRank right topic sensitive PageRank where you can set this teleporting probability based on your need ok.

(Refer Slide Time: 15:42)



PageRank: Random Surfer Model

With the help of model and the analogy discussed here, the PageRank formulation is revised as:

$$R(w) = (1 - \alpha) \sum_{b \in B_w} \frac{R(b)}{N_b} + \alpha \left(\sum_{b \in B_w} \frac{R(b)}{N_b} + \frac{1}{N} \right)$$

$\sum_{i=1}^N R(i) = 1$

- The parameter α is a parameter that controls the balance between the importance of two components of the formulation above
- The random jump action is introduced in the revised PageRank method to deal with **Dead Ends** and **Spider Traps** in the network



So, now let us look at the combined PageRank equation. So, this part we have already seen right. And what is this E? This E is the is basically a vector right which contains teleportation

probability. So, in this case see since we assume that all the nodes are equally likely to be jumped to we say. So, this E would be 1 by N in our case in this particular case. So, we have this probability and this probability this is choosing one node one of the forward nodes uniformly at random and this is jumping from the given node to another to any of the nodes.


And these two things are controlled by this parameter alpha. This alpha is called the damping factor right. And we make sure that at after every iteration the PageRank value would be 1. The sum of the PageRank values would be 1 right and therefore, of course, the PageRank of individual node would be between 0 to 1 ok.

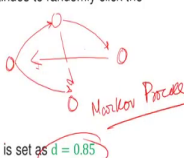
(Refer Slide Time: 16:58)


PageRank: Damping Factor

- PageRank theory centers around a random surfer who
 - randomly clicks on hyperlinks,
 - will eventually stop clicking, move to another random page, and
 - repeat the above sequence
- The damping factor d refers to the probability that the surfer continues to randomly click the current chain of hyperlinks
- We usually set $d = 1 - \alpha$
- Then the revised PageRank formula:

$$R_{i+1} = \frac{1-d}{N} + dR_iA$$
- We may set any value as damping factor; however, historically, it is set as $d = 0.85$







So, this damping factor alpha right I mean we will basically consider 1 minus alpha as damping factor. So, damping factor D is equal to 1 minus alpha, but does not matter I mean alpha and D they are basically kind of same right.

So, generally we set D as 0.85 there is no theoretical reasons behind this, but empirically it has been historically it has been set as 0.85 right. Now you can easily relate it to random walk process that I as I mentioned in the last day that you essentially start from a particular node and then and you are basically a random walker you follow one of the out wattages or you can jump from 1 node to you know any of the given nodes and so on and so forth. You keep on doing this thing until and unless things are things get stable.

Now, what do you mean by things get stable? So, when you keep on doing this thing in terms of matrix operation for example, in random work now you can map you can basically map the PageRank process using a Markov process right. In the Markov process what happens there are states Markov process. I am not going to going into details of this, but if you are interested you can look at it. In Markov process depending upon the you know say depending on the degree if it is zeroth order Markov process or first order Markov process second order Markov process the prior would be different right.

But let us assume that it is its a simple process where nodes are states and you can basically move from. So, the states are connected to transition probabilities right. So, you can move from one node to another node through this transition this transition edges with certain probabilities. See here also you can think of web pages as different states and hyperlinks as you know this edges to through which the transition is possible.


So, in the steady state. So, in the steady state there is something called steady state probability steady state probability. At the steady state probability what happens is that the ranking of web pages would not change further. You keep on you know keep on iterating the process it would not change further the normalized version ok. So, at the steady state probability as a steady state you know stage you get whatever probability you get that would be your PageRank ok.

(Refer Slide Time: 19:52)

RWR Random walk with Restart

Personalized PageRank

E = {0.0, 1/p, ..., 1/p, ..., 0}



- The vector E characterizes the **random jump** after surfing hyperlinks from a page
- The landing page need not be equally-likely for all the pages of the graph
- The surfer may be biased to return to one or more selective pages based on the search
 - Surfer may land a specific page on return (say, index page)
 - Surfer may land one of a set S of pages
 - Surfer may land on one of a list S_w of pages based on her search pattern
- The distribution of $E(S)$ or $E(S_w)$ will be different from being uniform distribution.
- The modified (Personalized) PageRank formula is as follows:

$$R(w) = (1 - \alpha) \sum_{b \in B_w} \frac{R(b)}{N_b} + \alpha E(S_w)$$


0

N-P

1/N

{P}

1/P



So, a variation of a PageRank is something called personalized PageRank ok. What is personalized PageRank? So, in case of PageRank we assume that a random walker can jump from that node to any of the nodes ok in the system. It may happen that you do not allow the random worker to jump to any of the nodes. You set a set of nodes you fix a set of nodes and say that hey if you really want to jump you can only jump to these nodes these set of nodes. Any of the I mean one of these set of nodes right.

So, therefore, the teleportation probability $\frac{1}{N}$ that we mentioned earlier would not remain as $\frac{1}{N}$ in this case right. So, what would happen say there are N number of nodes out of them P number of nodes P number of web pages have been selected by you and your random worker can only jump to one of this P web pages right. Remember there are two there are two possibilities; one in one possibility you can follow one of the outward edges and the other possibility you can jump.

Now, when you jump when you want to jump you cannot jump randomly. You either you can jump to any of the predefined set of pages ok. And this predefined set of pages are already selected based on the topic right based on the application for example. So, for these pages for this set of pages it would be uniform it would be $\frac{1}{P}$ or $\frac{1}{P}$ say and for other $N - P$ nodes it is 0.

So, when we write this E right this one. This would be is a vector it would be $0\ 0\ 0$, but $\frac{1}{P}$ for those pages which have been selected other would be 0 ok. So, now, you see this set of web pages S_w this for this cases your jumping probability teleportation probability will be nonzero for other cases this would be 0 ok. So, this is called personalized PageRank because you are essentially making the I mean entire PageRank process personalized right.

You are giving preference to a set of web pages for which you want your random worker to come again and again. This is also biased, biased PageRank. This is also called topic specific PageRank because the set of pages that we are going to choose these are based on the topics which you are interested in. For example, say you are running the PageRank algorithm on the citation network. And you want that your random walker random worker would basically jump again and again to the paper scientific papers related to AI ok.

So, you basically ignore other papers and you give more weightage is to the people related to AI this called this is also called topic specific PageRank. It may happen that this set is w right or this P set this contains only 1 page ok. So, what you are basically saying that either you

choose one of the outward edges forward edges or you jump right. So, this is called random walk with restart Random Work with Restart RWR ok.

So, the idea is that when you say you start from a particular node and you only allow your random worker to jump to that node right. So, the random walker is not allowed to jump to any other nodes it can only jump to the seed node from which you start you have already started your random work process. So, in that case what would be the value of E all the nodes the values of all the nodes would be 0 except the 1 from which you start and for that it would be 1 ok.

This is called random work with restart. The random work with this should be very important we will discuss later stages it is very important when you know when you understand an importance of a node with respect to its contexts. So, you do not want your random worker to move out of the context either you move through the context or you jump to the initial node ok.

(Refer Slide Time: 25:26)

PageRank: Advantages


- Vectorized system of equations fast to compute
- Guaranteed to converge to a unique solution
- Ranks can be pre-computed during indexing and re-used during query time
- Ranks are robust and stable as in-edges to a page are harder to manipulate than out-edges
- Conforms with the intuitive notion of importance of entities from the real world



Now, what is the advantage of PageRank kind of method? The advantage is that you can you easily vectorized it because ultimately you deal with matrix and vectors right. It is guaranteed to converge to an unique to a unique solution that is also good you can actually pre compute PageRank initially for a web graph right. And this PageRank value would act as a as an initial prestige for a page for a particular web page and then depending upon the content or query you can further modify, but the initial ranking can be obtained using PageRank right.


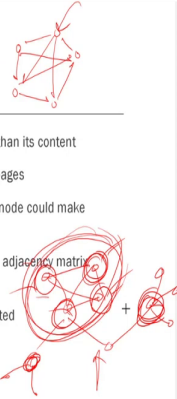
And the other important I mean other advantages is that it may be considered as a robust metric because it all depends on the inward edges right. You are moving because if a node has more involved edges its importance will increase. So, inward edges is something which is difficult to manipulate. So, therefore, this is kind of a robust.

(Refer Slide Time: 26:38)



PageRank: Disadvantages

- Prone to spamming, as it considers only the connections of node rather than its content
- A page can get high rank by connecting a lot of trivial (possibly dummy) pages
- Possibility of manipulation cannot be avoided completely as a malicious node could make hyperlinks with important pages and elevate its rank
- The basic PageRank system assumes a static system; no modification in adjacency matrix allowed during computation
- For dynamic systems, any modification requires all ranks to be re-computed
- Formulation has been extended for dynamic networks



The disadvantage is that since it is not depending dependent on the content of the webpage. One can actually create a crap webpage with spam content and add a lot of such hyperlinks and take part of the decision process ok. The second problem is that you know PageRank is mostly useful for static graphs ok. Although it its variations have already been proposed, but mostly it is useful for static graphs.

If the graph changes over time you need to compute PageRank again and again right. You can actually you know although I mentioned that this is robust, but there are some ways you can hack it. For example, you create a lot of web pages for example, and these web pages are linking each other like this right. It is also kind of a spider trap, but your now the size of the spider trap is huge. So, when your random walker comes here right it basically keeps moving within the within your collusive network ok.

Using your own gang automatically your PageRank will increase ok. This is one this is another such problem of PageRank the other PageRank problem the other limitation of PageRank is that it gives importance to nodes which are kind of redundant. For example, think of a network like this right something like this ok.

If you run PageRank what would happen is that maybe this page will be ranked first. This would be second this would be third this would be fourth this would be fifth and so on. If you look at the first four webpages they are very close by they are nearby right.

So, the amount of information that you can obtain from each of these nodes will be same right. For example, you basically want to get diversified information from a network right. So, if you run PageRank on this kind of network you will get these four nodes as top nodes. But these four nodes are part of the same context there is no need to return four nodes. You can only return one node and that node would you that node you give you that node would basically give you imp you know ideas about this page this particular portion.

Instead of this if you return one node from this component one node say this one and this node these four nodes at these three nodes at top three nodes then you basically cover the diverse parts of the network. So, diversity is not guaranteed in case of PageRank. So, we will discuss another matrix called div rank in the next lecture which basically takes care of both the importance as well as diversity ok.

Thank you.