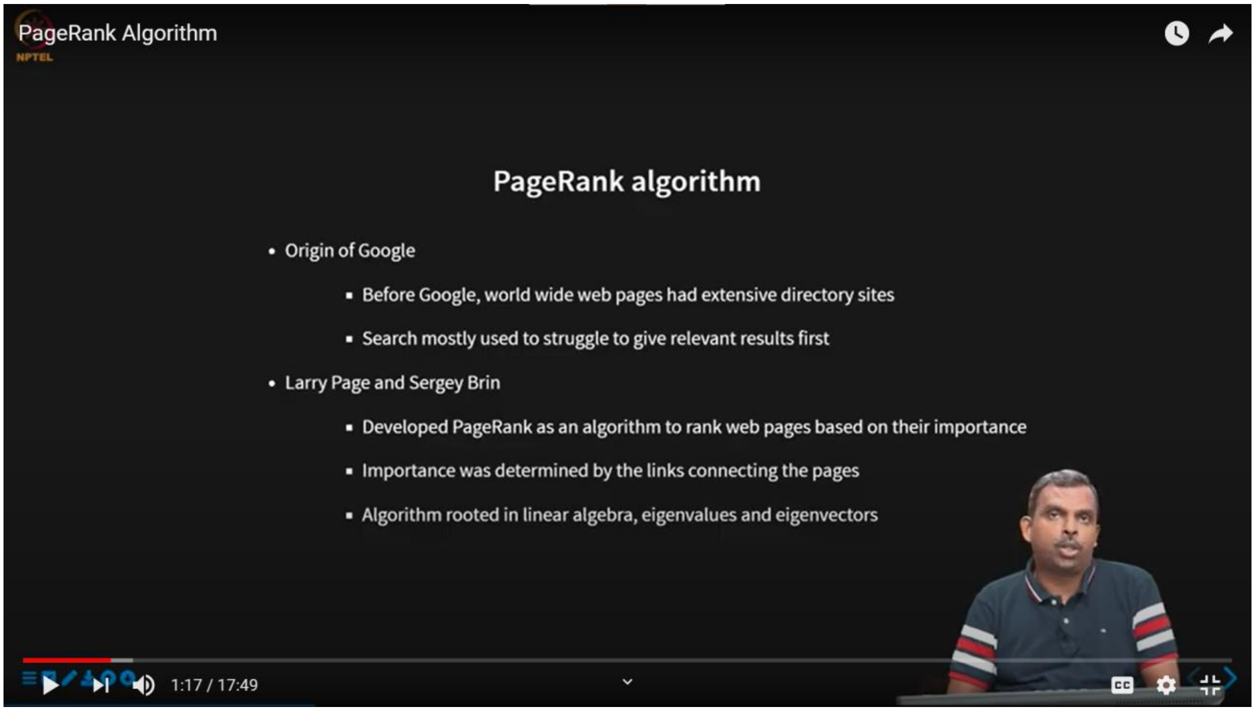
Applied Linear Algebra

Prof. Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras

Week 06
PageRank Algorithm

Hello and welcome. So in this lecture we're going to see one more application for eigenvalues, eigenvectors in Linear Algebra and that's called the page rank algorithm. It's a very, very popular algorithm today. The web is full of this algorithm and you'll see why this is one of the most interesting and very interesting algorithms that are out there. In fact it's a very lucrative algorithm. It makes a lot of money by the way. So you will see why that is so as we move along.

(Refer Slide Time: 01:17)


PageRank Algorithm

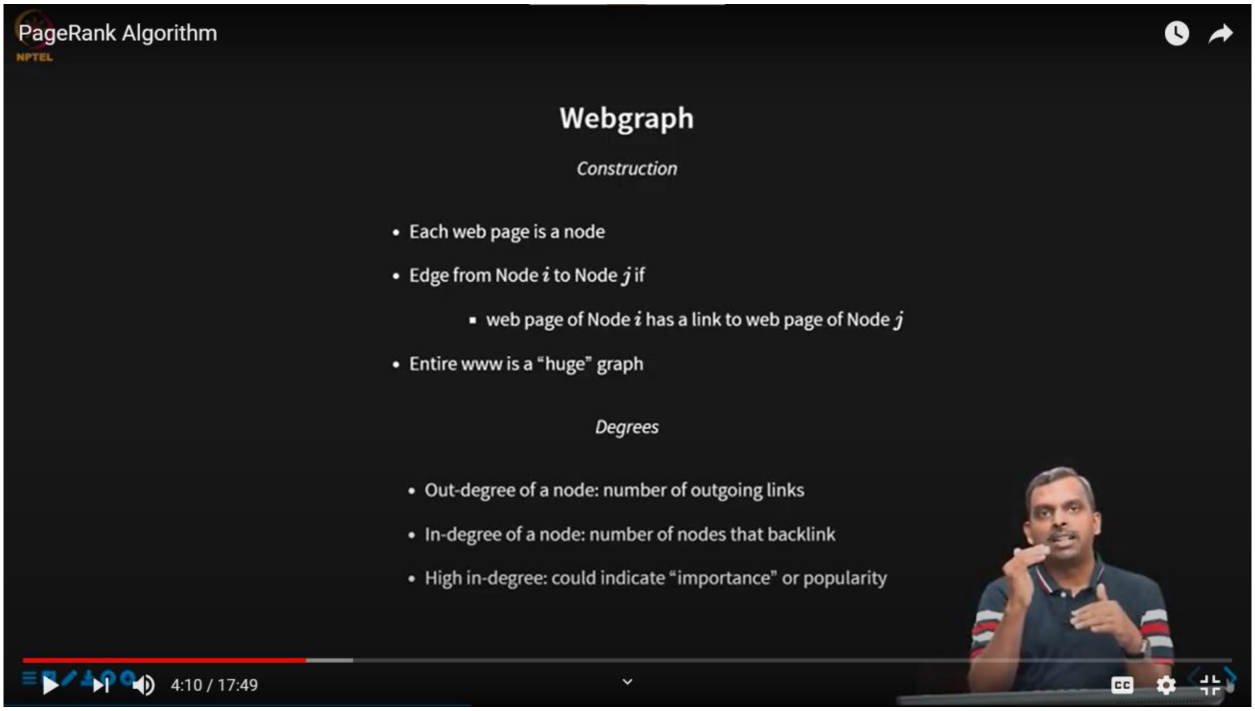
NPTEL

PageRank algorithm

- Origin of Google
  - Before Google, world wide web pages had extensive directory sites
  - Search mostly used to struggle to give relevant results first
- Larry Page and Sergey Brin
  - Developed PageRank as an algorithm to rank web pages based on their importance
  - Importance was determined by the links connecting the pages
  - Algorithm rooted in linear algebra, eigenvalues and eigenvectors

The recap is the same as before so I will skip it. We looked at quite a few ideas and then we are currently looking at applications of eigenvalues, eigenvectors and all that. So what is the story behind the page rank algorithm? It’s got to do with the origin of Google. Many of you might be very young and you may not know the world before Google, but there was a world like that and at that time there was the world wide web. The web was still there, but Google wasn't there and if you had to search or, you know, browse through the web etc... So you had these big directory sites. So you have to go through and see, you know this subject this area, that area do things like that and then find what you want. And quite often when you search, the results weren't that great. They were not very good. Most searches, you know, I mean you never used to get good relevant results. You have to really keep looking at... Then Larry Page and Sergey Brin came along and they developed this page rank as an algorithm to rank web pages and they could give really good results. The results from Google were spectacular and, I mean, you would really get what you wanted to find quite easily. And search became such a huge thing today, okay? It's become quite a huge thing today. It's been more than 25 years since then. But still it's huge and at the heart of it, whether or not you believe it, is linear algebra, eigenvalues and eigenvectors. In fact there is a very popular paper which says, like, 25 billion dollar eigenvector or something like that, which talks about this connection to page rank and how, you know, importance of web pages based on how they link with each other, how they are connected to each other can be done using eigenvectors, eigenvalues and linear algebra. So let us dive into it and more than anything I think this just tells you how you should really know, you know, the theory behind things, how you should not look down upon it, understand it, get to the basics of it know it intuitively and then also identify opportunities where you can pick up a problem and try and model it back into a more theoretical flavored problem and then solutions would magically present themselves, okay? So let us look at what this page rank does, okay?

(Refer Slide Time; 04:10)



Webgraph

Construction

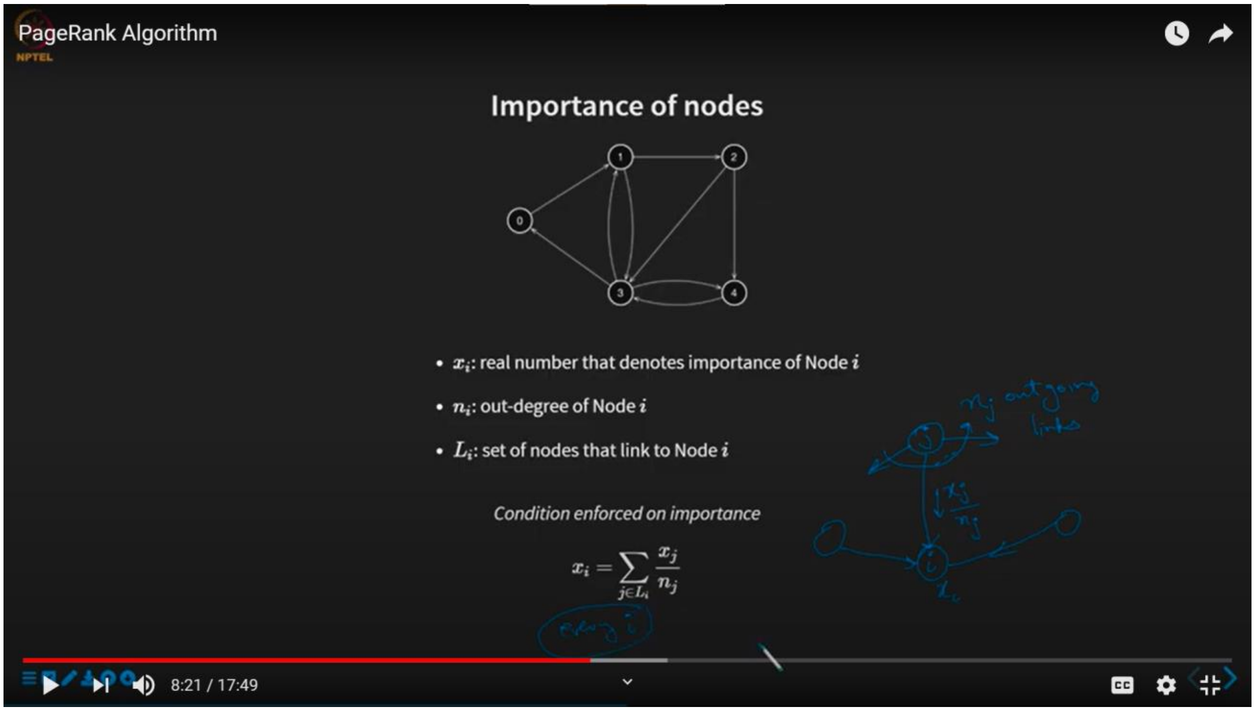
- Each web page is a node
- Edge from Node i to Node j if
  - web page of Node i has a link to web page of Node j
- Entire www is a "huge" graph

Degrees

- Out-degree of a node: number of outgoing links
- In-degree of a node: number of nodes that backlink
- High in-degree: could indicate "importance" or popularity

So at the heart of it is something called the web graph where web graph is sort of a graph model for the world wide web or anything else like that, okay? So anything connected like that you can make a graph. How does that work? Here is how the construction works. Every web page, right, every web page is like a node in the graph, node or vertex in the graph. And if a web page has a link to another web page, then you draw an edge from the first web page to the second web page, okay? A directed edge is drawn like that, okay? So you can imagine the world wide web today has billions of pages. And so the graph would be a huge graph. It would have billions of nodes. And there'll be connections or edges from every node to every other node that it connects to, okay? So if you have a lot of links, it will all link up. Now there are a few interesting things about this web graph. First thing is this notion of degrees. There is an out degree to a node, right? So number of outgoing links from a web page. There is also an in degree to a node, okay? So this is not something that that webpage itself controls. How many other pages are pointing to you, okay? So that's also an interesting indication of how important your web page is, right? So high in degree could indicate, you know, some sort of importance of popularity. So in a search, if some page with high in degree or high popularity or high importance comes up, you want to show it first, right? So that you know it's an important page and people know that it comes up there and then they will go and you know that they'll find it useful, okay? So search that way needs to identify important web pages which also satisfy the search criteria, okay?

(Refer Slide Time: 08:23)



Importance of nodes

- x_i: real number that denotes importance of Node i
- n_i: out-degree of Node i
- L_i: set of nodes that link to Node i

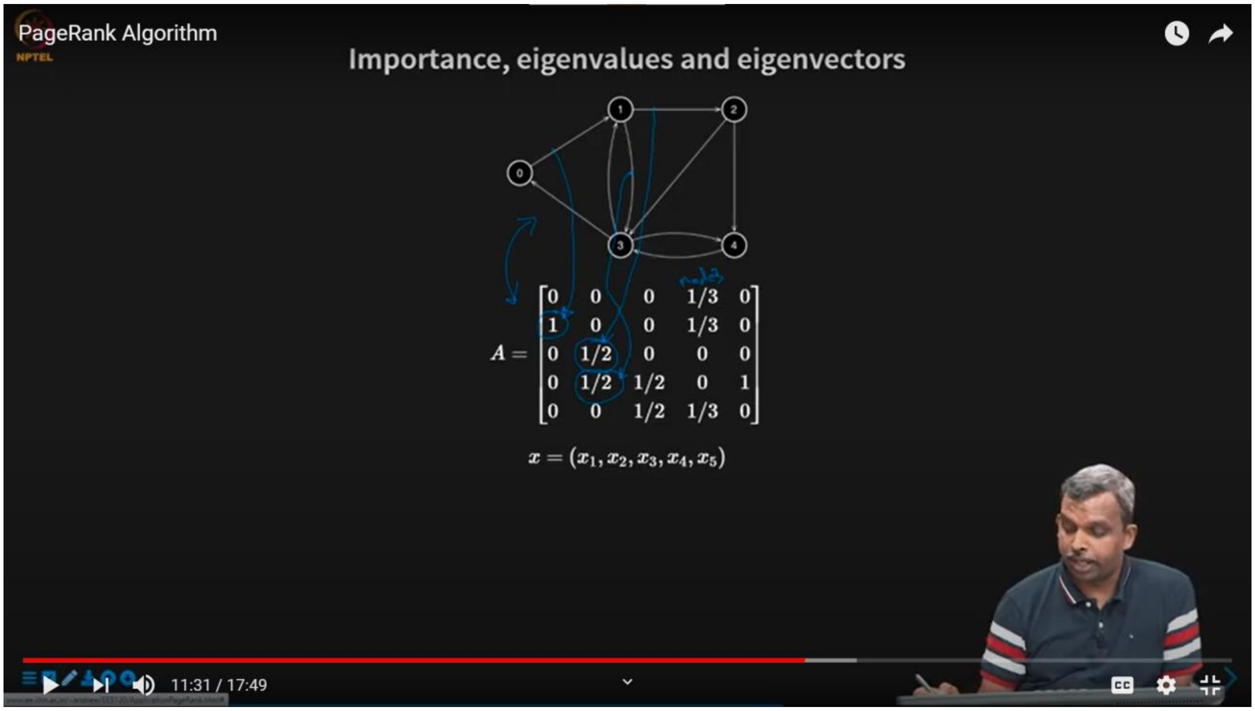
Condition enforced on importance

$$x_i = \sum_{j \in L_i} \frac{x_j}{n_j}$$

So how do you associate this importance? How do you do this importance? You use this web graph but then you do something more, okay? So what do you do? So how do you go about assigning some sort of importance to the node, okay? So you made a graph. There are a lot of edges coming in and going out. There is degree. Is degree enough or do you need more? It turns out degree alone is not a very strong indicator. There can be lots of other reasons where, you know, degree can be very high but it may not be very relevant to what you are looking for in some sense, you know? I mean, degree is not that good, okay? We can go into details later when, the details you can look it up or you can try to understand why degree may not be everything, but... So there is this other way in which you can associate importance to every page or every node in the web graph that you are looking at, okay? So let's say for node i, I want to associate something called importance which I will call x_i, okay? I do not even know how to do it at this point, but let us say I have some way of assigning it. So what is it that it should satisfy? How do we, how do we impose some intelligent conditions on it and then try to think about what it means, okay? So that's something we are going to do as we go along. But for now let us say we associate some number called importance to every node, okay? Now I'll use a few more notation. One notation I will use is for out degree of node i, okay? I will call it n_i, okay? And also the neighbors that link to node i, okay? What are all the other nodes that have a link to node i, okay? So that is L_i, okay? And notice there is this interesting little condition you can impose on these importance values x_i. You can say x_i, right, should be equal to, the importance of node i, should be connected to the, should come from the importance of the nodes that are pointing to it, right? So it's sort of, in one way, it's sort of all interconnected. But there should be this equation which connects the importances of all these nodes based on the connection. How does that work? This is the equation, okay? I mean this is just some intuitive equation at this point. But notice the very nice thing that it captures. Whatever value you assigned as importance to every node, they should satisfy this equation. So x_i for every node should be equal to sort of like the total sum of the incoming importances. What is the incoming importance? j, Every node j now belongs to L_i which means, right, so that is the sort of picture here. If you want you can have that in mind. So you have the ith node, you have the jth node here and this links to this. But it also has other links, okay? Like that. How many does it have? It has n_j links, n_j outgoing links, okay? So what you say is sort of the importance that is flowing in here is x_j/n_j, okay? The importance of node j itself. But it gets divided by n_j because it's sort of linking to so many others, right? So it's linking to n_j other guys. So x_j/n_j is sort of like the importance that node j is giving to node i, right? Because of that one link, okay? So when you add up all these other things that are connected to it and the importances that come from there, right, you should get the importance of x_i, okay? x_i should be sum of all the importances that are flowing in from the connected nodes, okay? They should first of all link to i, node i. And then the original importance of that node that is connecting gets divided by the out degree of that node, okay? So that sum has to be equal to the importance of x_a. Now this is like a, you know... Remember i also will be connected to all the other nodes. So this i will be on the right, sometimes come on the right hand side also, okay? So it's not just an equation for one i, this is for every i, okay? So you get a bunch of equations, all right?

Now first thing you have to worry about is: given an arbitrary graph, you know, you can imagine the web has a huge graph, can you even come up with an x_i like this, okay? That satisfies all these equations. Is it even possible? Does there exist an important assignment, okay? Is it possible to find these x_is? That's a question that you should ask. And if it is, it seems to be a reasonable metric, okay? So if you sort of associate importance with, you know, being connected to other important people and all that, important nodes and all that then this is a good metric for that situation, okay? But is there a vector like this? Can it be, you know, computed efficiently? All of those things one needs to look at. For that, that's where, you know, linear algebra and eigenvalues and eigenvectors will help you. Let's see this a little bit more closely, okay? So what you can do is you can define a matrix A in this fashion, okay? And define this vector of importances $(x_1, x_2, x_3, x_4, x_5)$ and write it out like this, okay? So I haven't exactly told you how this comes from. This matrix is associated with this graph by the way. This is the web graph and this matrix is associated with this graph, okay? So I put this graph here just for illustration. You can see for instance node 0 has an out degree of 1 and in degree of 1. Node 1 has an out degree of 2 and an in degree of 2. Node 2 has an out degree of 2, in degree of 1, like that, okay? So there are lots of things. But just by the looks of it, it appears that, you know, node 3 is important, you know? Doesn't it seem like that? Lots of things are happening around node 3. It seems to be something of importance maybe, you know? But let's see if this importance vector and the calculations will bear it out.

(Refer Slide Time: 11:31)



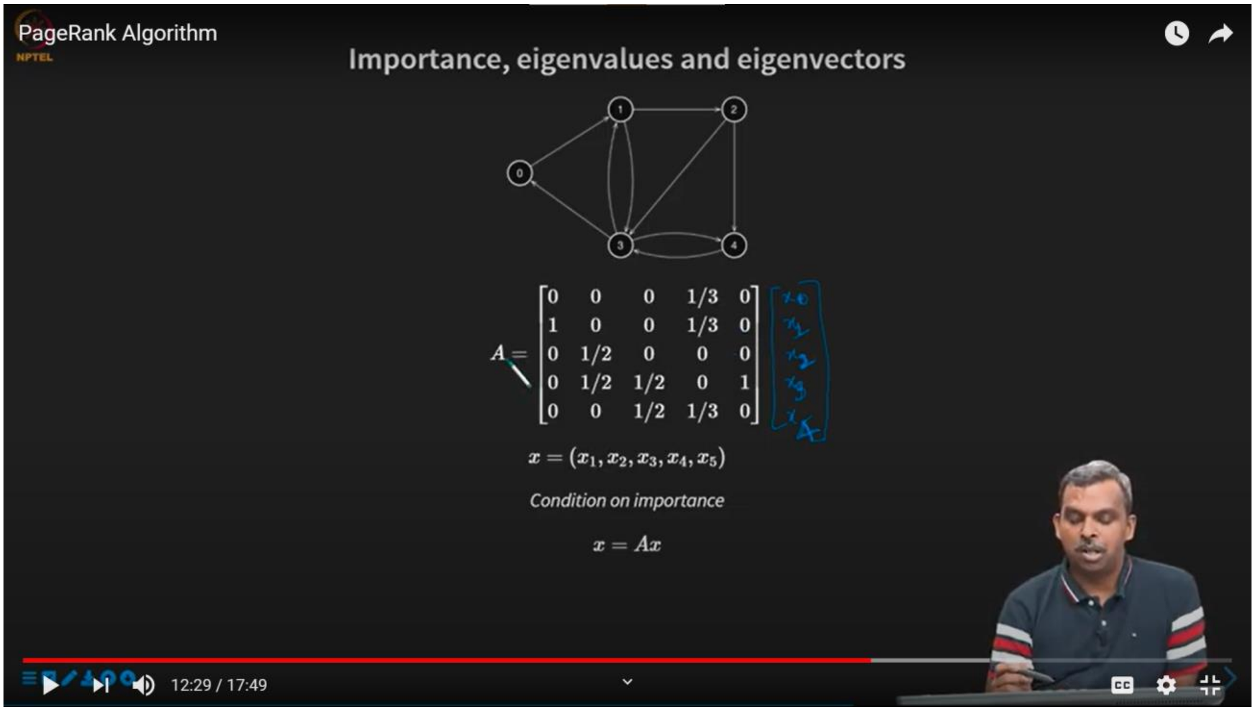
Importance, eigenvalues and eigenvectors

$$A = \begin{bmatrix} 0 & 0 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 1/3 & 0 \end{bmatrix}$$

$$x = (x_1, x_2, x_3, x_4, x_5)$$

What do I do here? What I have done here is the columns, right, the columns are representing each node. Rows also represent each node, okay? So you can see the first column. Node 0. Node 0 has an incoming link from, has an outgoing link to node 1, right? Node 0 has an outgoing link to node 1 and it is just 1 because there is only one outgoing link, okay? Now node 1 has two outgoing links. One to node 2 and another to node 3. But there are two of them. So you put a half weight on it. So this guy will come from this one and this half will come from this edge, okay? So you can see how this matrix has been made, okay? For every edge, there will be a number here in that position, okay? In the corresponding position from column to row, okay? So you can see from node 1 it went to node 3 and the weight half depends on the out degree of node 1, okay? So if there are two edges, you will have a weight ½ ½ . Likewise you can see this column, right? This column corresponds to node 3, okay? And you can see its $\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}$. It has three outgoing edges to node 0, node1 and node 4. And there is a node 0, node 1 and node 4 and weight is $\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3}$, okay? So that's how you make this matrix. Now what is interesting about this matrix is this condition that we had before, right? So you saw this condition, this condition that we had. x_i equals sum of all these conditions. This condition translates into a very simple equation with this matrix $x = Ax$. x is your importance vector that needs to be equal to Ax. You can check that this was exactly what it would mean, you know, if you multiply with the, you know, the importance vector on the right. So $(x_0\ x_1\ x_2\ x_3\ x_4)$. So you see x_0 has to be equal to $\frac{x_3}{3}$, right? x_0 will be $\frac{x_3}{3}$. So that is how this matrix works out and you can see x_1 has to be $\frac{x_3}{3} + x_0$, right? $\frac{x_3}{3} + x_0$. So it works out correctly. So this is just a matrix form. We made this matrix from the graph and then when you multiply this matrix on the right with the importance vector you should get the importance vector itself.

(Refer Slide Time: 12:29)



Importance, eigenvalues and eigenvectors

$$A = \begin{bmatrix} 0 & 0 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 1/3 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

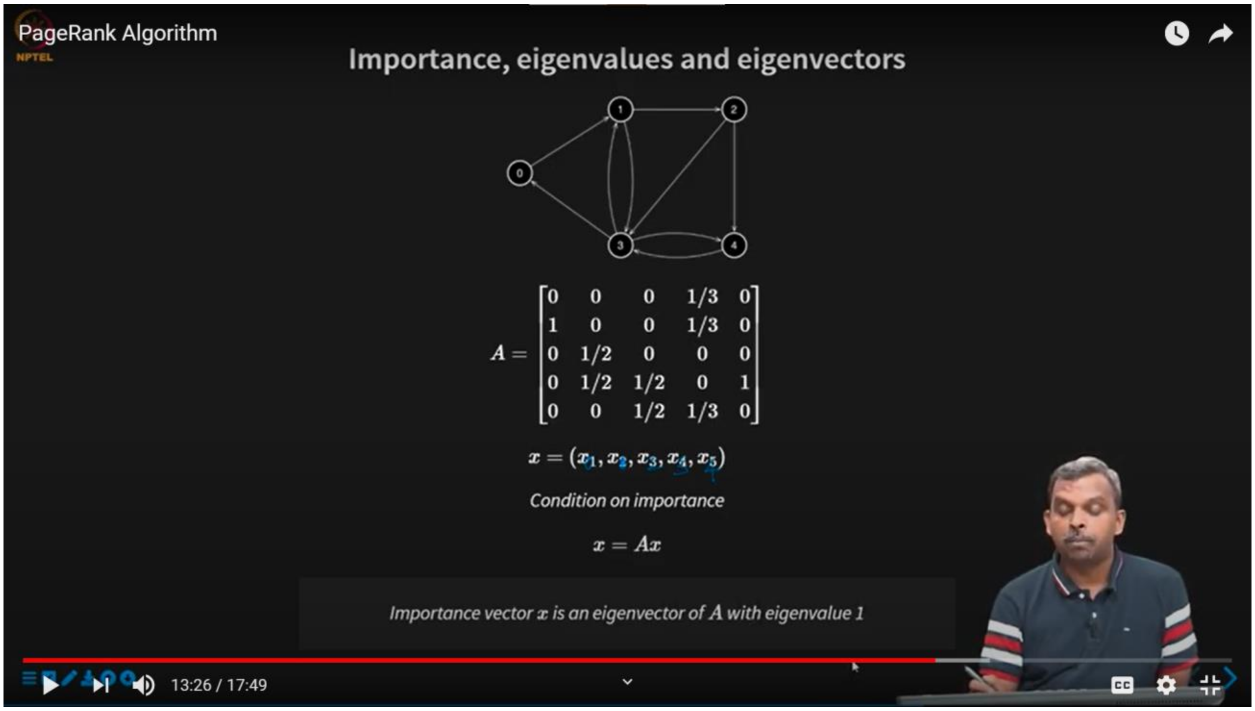
$$x = (x_1, x_2, x_3, x_4, x_5)$$

Condition on importance

$$x = Ax$$

Now what is this equation? You can identify this equation quite easily. This is an eigenvector equation, right? So the importance vector x, okay? So this x should be equal to Ax and from this equation we see that the importance vector, if it exists, will be an eigenvector of this matrix A and it will have an eigenvalue of 1, okay? So you see immediately these importances that we wanted to assign based on popularity or what is well connected etc. are connected in a wonderful way to matrices and eigenvectors and eigenvalues, okay? In particular a very special type of matrix that we constructed from the web graph and that matrix should have hopefully an eigenvalue 1 and it should have an eigenvector. Then you will have a valid importance vector that you can assign to this web graph, okay?

(Refer Slide Time: 13:26)



Importance, eigenvalues and eigenvectors

$$A = \begin{bmatrix} 0 & 0 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 1/3 & 0 \end{bmatrix}$$

$$x = (x_1, x_2, x_3, x_4, x_5)$$

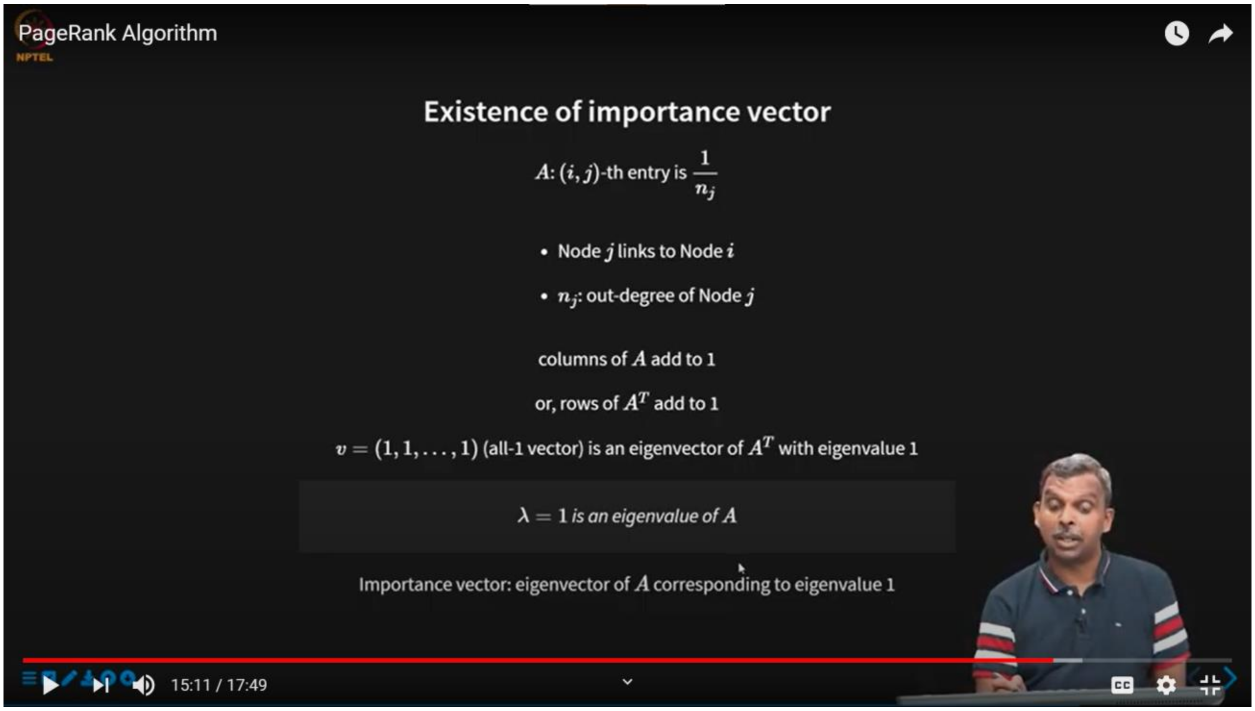
Condition on importance

$$x = Ax$$

Importance vector x is an eigenvector of A with eigenvalue 1

So let's see if this is possible or not. Does a matrix like this always have an eigenvector and eigenvalue 1? It turns out yes and that is because of the way in which we have assigned these numbers here, okay? So why is that? So if you see the matrix A, the way we constructed it, the $(i, j)^{th}$ entry means node j links to node i, okay? Right? node j links to node i and the weight or the number that you put there is $1/n_j$. And n_j is the out degree of node j, okay? So the property that A always satisfies is the columns of A add to 1, okay? So you can go back and check in the previous example also we will see that the columns add to 1. The columns of A always add to one. So that's nice, okay? The columns of A add to one. What's interesting when the columns add to 1? The rows of A^T add to 1, isn't it? If the columns of A add to 1, the rows of A^T add to 1. So when the rows are adding to 1, then all ones becomes an eigenvector of A^T with eigenvalue 1, okay? The all ones vector becomes an eigenvector for A^T because the rows are adding up to 1. Anytime you have a matrix with the rows adding up to 1, all the rows adding up to 1, you put all one vector on the right, you are simply doing sum of the rows, right? Sum of the values in the rows. And that will just give you all ones, okay? So the all ones vector is an eigenvector of A^T with eigenvalue 1. Now A^T has eigenvalue 1 which means what? A also has eigenvalue 1. I do not know about the eigenvector, but A has an eigenvalue 1, okay? So $\lambda = 1$ is an eigenvalue of A and the importance vector will be the eigenvector of A corresponding to eigenvalue 1, okay? You take the eigenvalue 1. I showed you just now that A has an eigenvalue 1 because of this property that columns add to 1, okay? And the eigenvector that you pick up from there will be assigned, can be assigned as the importance vector, okay?

(Refer Slide Time: 15:11)



Existence of importance vector

A: (i, j)-th entry is $\frac{1}{n_j}$

- Node j links to Node i
- n_j: out-degree of Node j

columns of A add to 1

or, rows of A^T add to 1

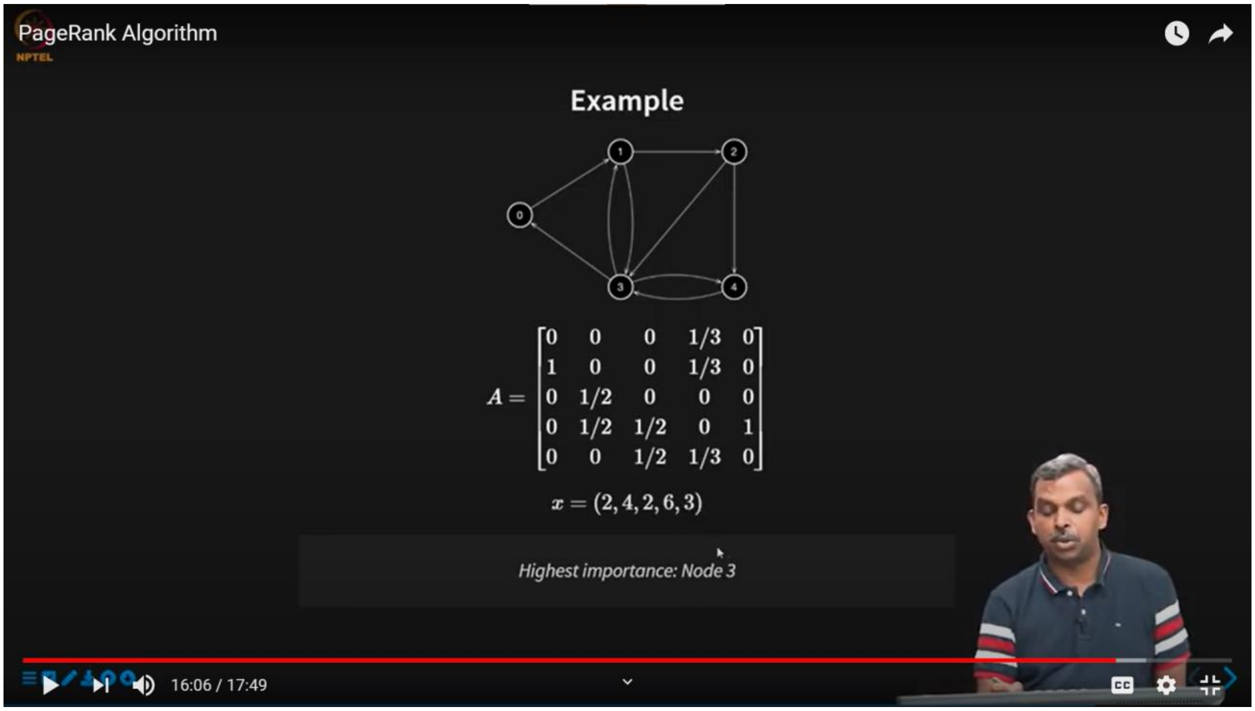
$v = (1, 1, \ldots, 1)$ (all-1 vector) is an eigenvector of A^T with eigenvalue 1

$\lambda = 1$ *is an eigenvalue of* A

Importance vector: eigenvector of A corresponding to eigenvalue 1

So let us go back to our toy example. In this example if you notice very carefully, you will see that $(2, 4, 2, 6, 3)$ is an eigenvector, okay? And you see from there that the one with the highest value of importance is node 3. And just like we thought it should happen. In this graph it looks very clearly that something is going on with 3, okay? Some very interesting thing is going on with 3. And that ends up being the important node as well, okay? After that it says it's node 1, okay? Node 1, yeah, sort of looks important. And then node 4 etc. etc., okay? So that's the interesting importance vector here. Maybe you can think about other interpretations. But at least for this simple graph we see that we can easily compute the importance vector and that ends up being the eigenvector, okay? So the eigenvector with eigenvalue 1. Nice connection to linear algebra, okay?

(Refer Slide Time: 16:06)



PageRank Algorithm

NPTEL

Example

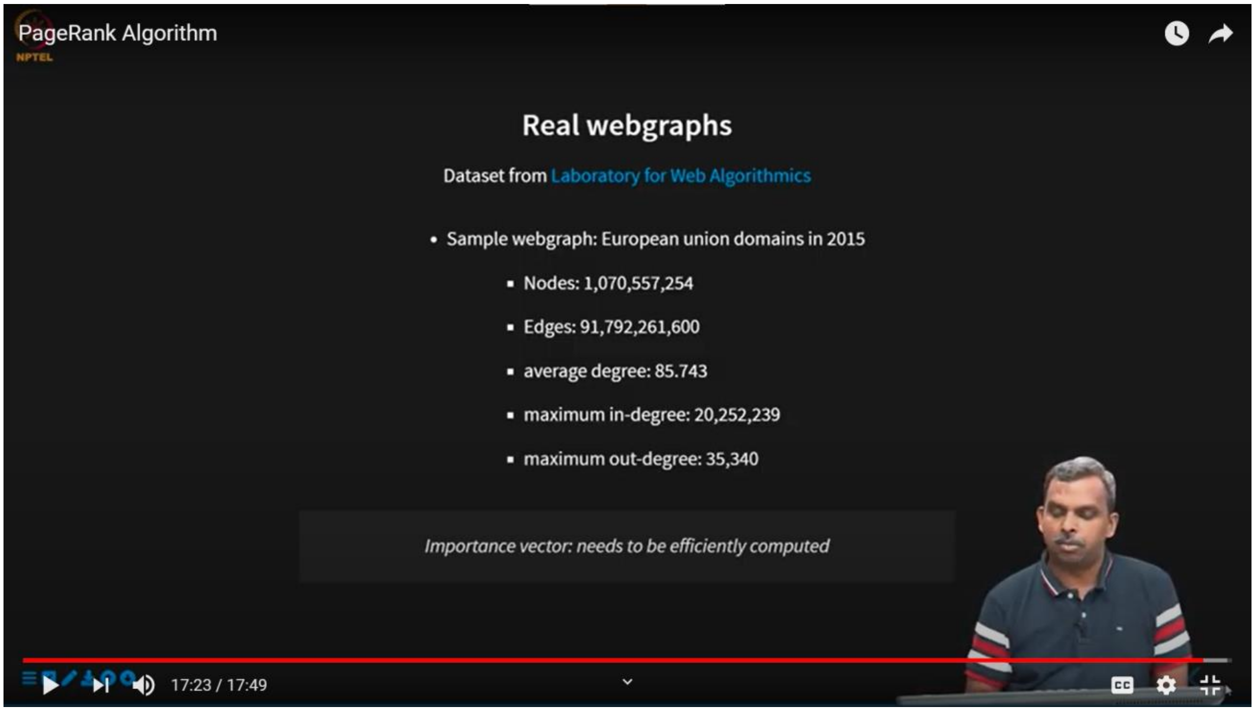
$$A = \begin{bmatrix} 0 & 0 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 0 & 1 \\ 0 & 0 & 1/2 & 1/3 & 0 \end{bmatrix}$$

$$x = (2, 4, 2, 6, 3)$$

Highest importance: Node 3

So if you want to look at real web graphs, there are websites, people maintain graphs from the, you know, actual internet by doing web crawling. They go from link to link, then figure it out. This is a website, you know, Laboratory for Web Algorithmics, I believe in Milan, has this, runs this data set. You can go there and look at the data. I looked at a few of the data. They have a lot of data. This is data that they did for the European Union domains. This is only domains with, you know, .it, .es, .fr, only the European Union domains in, some time in 2005. In that web graph they got more than a billion nodes, more than 91 billion edges and the average degree was about 85.73. There was a node with in-degree of more than 20 million, there was a node with an out degree of more than 35,000, okay? So this is the scale you're dealing with when it comes to the internet. And you ought to be able to compute eigenvectors and eigenvalues and other things at this kind of scale, okay? This kind of size. It's difficult to even imagine, so you need really efficient memory, really efficient, you know, processing power that's needed to maintain this importance factor and that's also possible. That was again a very interesting part of the page rank algorithm which we won't discuss in this lecture. But it gives you an idea, okay? So what is the kind of scale that's involved. So hopefully this lecture gave you a very, very interesting take on how these ideas based on linear algebra, eigenvectors and eigenvalues have revolutionized the world web and brought in technology that has made life easier for so many people, okay? Thank you very much.

(Refer Slide Time: 17:23)



Real webgraphs

Dataset from Laboratory for Web Algorithmics

- Sample webgraph: European union domains in 2015
  - Nodes: 1,070,557,254
  - Edges: 91,792,261,600
  - average degree: 85.743
  - maximum in-degree: 20,252,239
  - maximum out-degree: 35,340

Importance vector: needs to be efficiently computed