



AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

19CSE337 Social Networking Security

Lecture 11

A vertical sidebar on the left side of the slide, featuring a grid of various social media and technology icons. The icons include a television, a camera, a lightbulb, a hand holding a device, a speech bubble, a padlock, a shopping cart, a smartphone, a person icon, a Twitter bird, and a large 't' logo. The background of the sidebar is a dark blue grid.

Topics to Discuss

- PageRank
- Katz Centrality



PageRank

- PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results.
- PageRank was named after Larry Page, one of the founders of Google.
- PageRank is a way of measuring the importance of website pages.
- According to Google: PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.



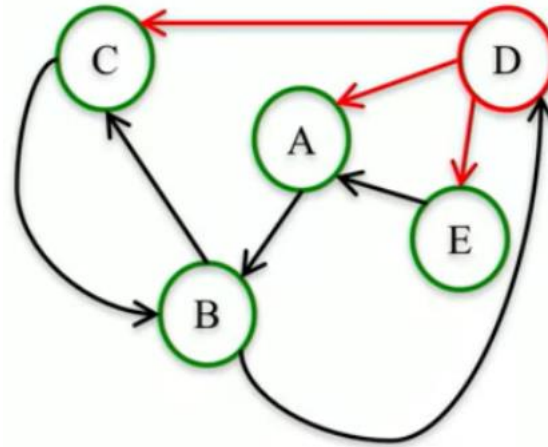
PageRank

- PageRank is a variant of Eigenvector centrality.
- The difference from eigenvector is that PageRank takes link direction and weight into account to calculate centrality.
- So, the links can pass influence only in one direction and pass different amounts of influence.
- Each node in a network is assigned a score based on its number of incoming links (its indegree) or in other words based on their connections, and their connections' connections.
- These links are also weighted depending on the relative score of its originating node.
- The nodes with many incoming links are influential, and nodes to which they are connected share some of that influence.

- **What does PageRank centrality tell me?**
 - Like Eigenvector Centrality, PageRank can help uncover influential or important nodes whose reach extends beyond just their direct connections. It's especially useful in scenarios where link direction is important.
 - Understanding citations (e.g; patent citations, academic citations).
 - Visualizing IT network activity.
 - Modeling the impact of SEO and link building activity.

PageRank Computation

- Calculate the page rank of the following graph.



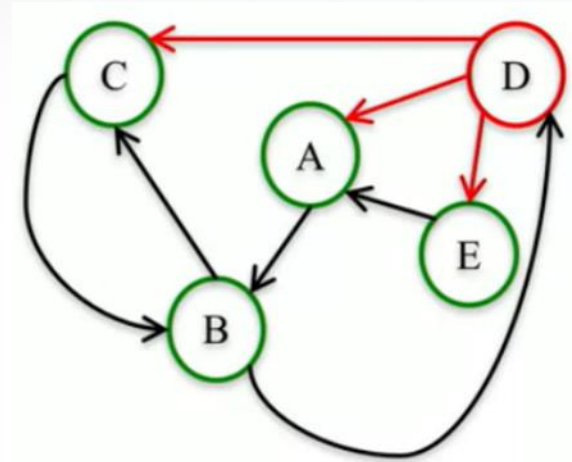


PageRank Computation

- Assume each node has a page rank value of $(1/n)$, where n is the total number of nodes in the graph.
- So, in our example, initially each node has a page rank value of $(1/5)$.
- By applying basic page rank updating rules K times, we will arrive at final page rank value of every node.

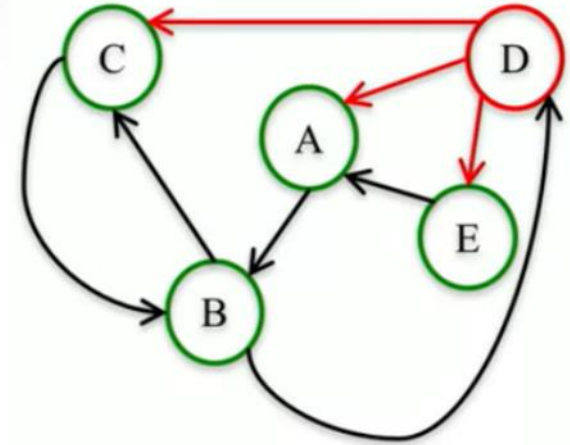
PageRank Computation

- Let's start with node A.
- Nodes D and E point to node A, so A is going to get PageRank from D and E.
- Now let's think about how much **PageRank A** is going to receive from each one of those two nodes.
- If we look at **D**, D has three edges, that points to three different nodes, C, A, and E.
- Therefore, A is going to receive $1/3$ of the current page rank that D has.
- D currently has $1/5$ PageRank, and so A is going to get $1/3$ of that $1/5$ PageRank that D has.



PageRank Computation

- Now A is also going to get PageRank from node E.
- Since E only points to A, then it's going to give all of its PageRank to node A.
- So A is going to get $1/5$ PageRank from node E.
- Therefore, in total, A is going to get $4/15$ PageRank from those two nodes.
- Hence, the new value PageRank of node A is $4/15$.
- Repeat the above procedure for every nodes.
- Repeat this process for K times until



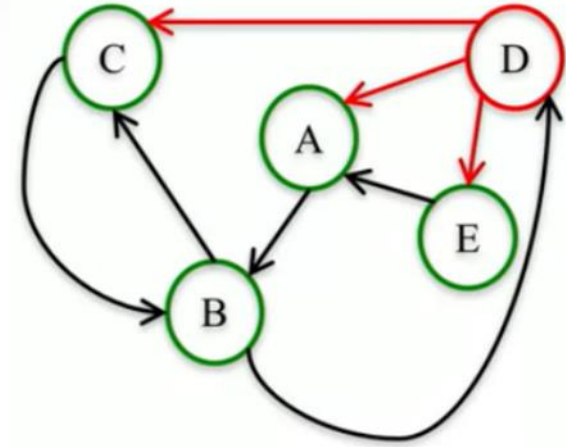
PageRank Computation

K=1

	A	B	C	D	E
Old PR Value	1/5	1/5	1/5	1/5	1/5
Updated PR	4/15	2/5	1/6	1/10	1/15
Value	0.267	0.4	0.16	0.1	0.067

Calculation

A	$(1/3) * (1/5) + (1/1) * (1/5) = (1/15) + (1/5) = (4/15)$
B	$(1/1) * (1/5) + (1/1) * (1/5) = (2/5)$
C	$(1/3) * (1/5) + (1/2) * (1/5) = (5/30) = (1/6)$
D	$(1/2) * (1/5) = (1/10)$
E	$(1/3) * (1/5) = (1/15)$



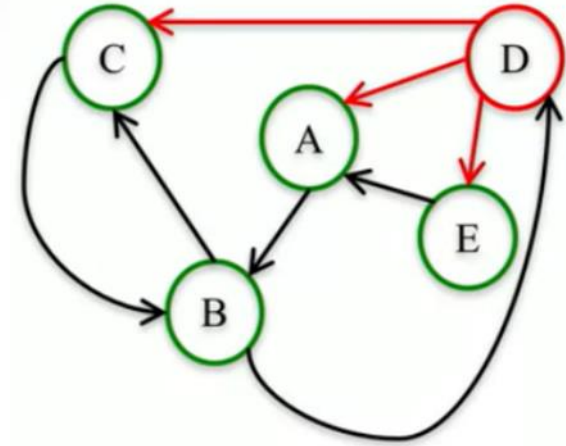
PageRank Computation

K=2

	A	B	C	D	E
Old PR Value	4/15	2/5	1/6	1/10	1/15
Updated PR	1/10	13/30	7/30	1/5	1/30
Value	0.1	0.43	0.23	0.2	0.033

Calculation

A	$(1/3) * (1/10) + (1/1) * (1/15) = (1/30) + (1/15) = (1/10)$
B	$(1/1) * (4/15) + (1/1) * (1/6) = (39/90) = (13/30)$
C	$(1/3) * (1/10) + (1/2) * (2/5) = (1/30) + (2/10) = (7/30)$
D	$(1/2) * (2/5) = (2/10) = (1/5)$
E	$(1/3) * (1/10) = (1/30)$





PageRank Computation

- From the results obtained after two steps we find that node B has the highest PageRank (0.43), followed by node C, then node D, node A, and E.
- Hence B is most important in this network.
- **What happens if we continue for another step?**
If we continue with more steps, we might notice that the values change a little bit, but they still have the same order and B is still the highest PageRank node.

PageRank using NetworkX

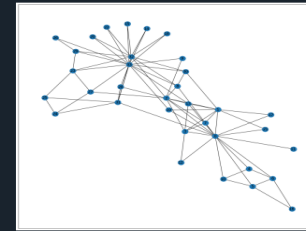
Spyder (Python 3.9)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\mails\.spyder-py3\BetwCen.py

```
1 from pylab import *
2 from operator import itemgetter
3 import networkx as nx
4 # G is the Karate Club Graph, the famous social graph published in 1977
5 G = nx.karate_club_graph()
6 plt.figure(figsize=(15, 15))
7 nx.draw_networkx(G, with_labels=True)
8
9 #calculate and print betweenness centrality
10 pr= nx.pagerank(G)
11 #To get top 5 nodes
12 sorted_degree = sorted(pr.items(),key=itemgetter(1),reverse=True)
13 for d in sorted_degree[:5]:
14     print(d)
15
```

C:\Users\mails\.spyder-py3



Help Variable Explorer Plots Files

Console 1/A X

Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

```
In [1]: runfile('C:/Users/mails/.spyder-py3/BetwCen.py', wdir='C:/Users/
mails/.spyder-py3')
(33, 0.1009179167487121)
(0, 0.09700181758983706)
(32, 0.07169213006588289)
(2, 0.05707842304763673)
(1, 0.052878391037427)
```

IPython console History

LSP Python: ready conda: base (Python 3.9.7) Line 15, Col 1 ASCII CRLF RW Mem 40%



ENG IN 21:53 18-01-2022



Katz Centrality

- Katz centrality is a measure of centrality in a network.
- It was introduced by Leo Katz in 1950.
- It is a variant of eigenvector centrality which overcomes the problems of eigenvector centrality.
- Katz centrality is mainly used for directed acyclic graphs.
- Katz centrality measures influence by taking into account the total number of walks between a pair of nodes.



Katz Centrality

- The adjacency matrix will be constructed based on the direction of links. The matrix will be asymmetric in nature.
- In directed networks, if a node is not pointed by any other nodes, its centrality will be zero and so its neighbours too!
- Katz centrality assigns a minimum score to all nodes and include this in calculation.



Katz Centrality

- Katz centrality of node i is $x_i = 1/\lambda \sum x_j + \beta$
- First part in the above formula is normal eigenvector centrality and second part is bias or free centrality score.
- Usually, we set $\beta=1$ for the computation.



Thanks.....