LINK ANALYSIS

- Greatest innovation of the century
- Massive collection of **interconnected** hypertext docs.

Features are

- End user and Content creators are of diverse background.
- They have different motives for using the web.
- Link Spammers manipulate the results of search engine Commercial use/malicious

User

Content Creator

Spammers

SE getting more and more Sophisticated to avoid being victim of spam

Spammers are finding Innovative ways of defeating the purpose of these search Engine. like [Term spam]

To Avoid Spam : Analyze the hyperlinks and Graph structure of WEB for ranking of results. = **LINK ANALYSIS**	Spammers responded with the ways to manipulate the pageRank too is called as **LINK SPAM.**
LINK ANALYSIS is one of the many factors for calculating composite score of the web page for the given query. : GOOGLE used PageRank, MPR	
Techniques **Trust Rank –** Topic sensitive PageRank – HITs is used to detect LinkSpam	

History of Search Engine and Spam

Huge information on Internet is useless unless information can be discovered and consumed by users.

Taxonomies based Search Engine

- All web pages are organized in Hierarchical way based on category labels.

Full Text Search Engine

- Presented user with keyword based search UI
- Increase growth rate/ creating indexes. Creating indexes is difficult task.

- Creating accurate taxonomies requires accurate Classification techniques.
- Increasing size of Internet.. This becomes impossible.

Internet: E-selling, opinion, information pushing… web search engines began to play a major role.
People want their pages to be retrieved by search engines.

SPAM: web page owners want their pages to be high ranked in a search query.
So this can be done by manipulating the web page content …
Link Spamming: is a part of black hat SEO hidden text, keyword stuffing, cloaking, doorway pages

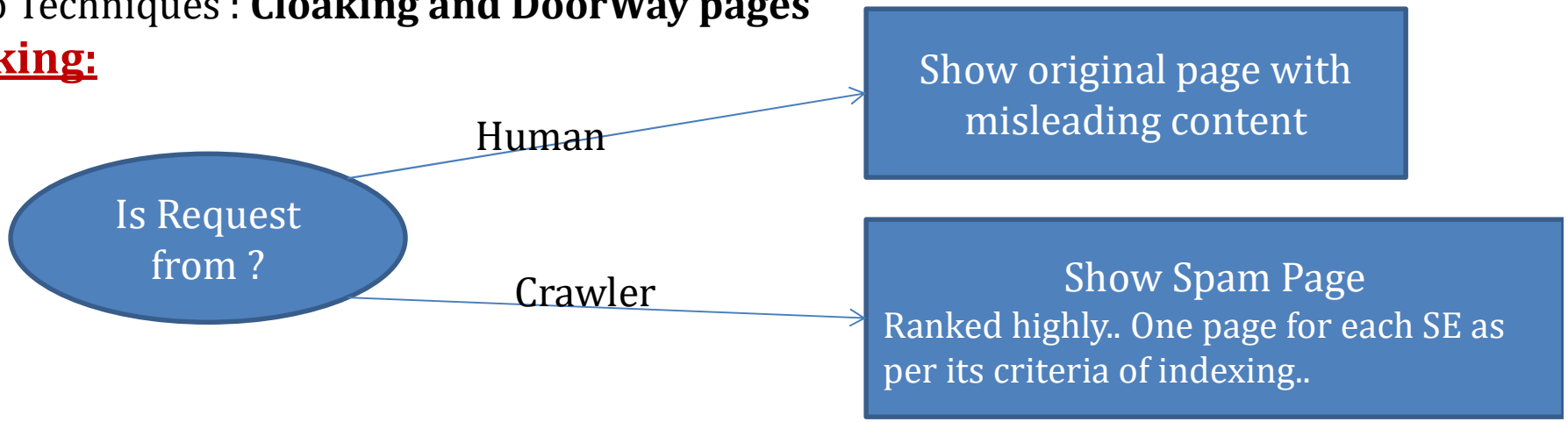
Spamdexing: Search Engine Spamming …. Spamming with Indexing

- SEO.. That increases the ranking
- SEO providers create web sites illegitimately indexed with high ranking in search engines.

Spamdexing: Search Engine Spamming …. Spamming with Indexing

➢Two Techniques : **Cloaking and DoorWay pages**

Cloaking:



➢Normal page for the URL is hidden from the engine.
➢It is cloaked.
➢Search Engine indexes this page under misleading keywords.
➢User will see a page that is totally different content to what is indexed by SE.

➢**Examples**

1. **Business selling writing instruments..**

   SE: text indicating history of writing instruments.. Etc..
   Human: page with images/ flash ads of writing instruments to users visiting the page

2. **Stuffing relevant extra text of keywords into a page:** Add the text like music into the page thousand times increases its ranking for music. Any query for music would lead the user to the writing instruments site.

Spamdexing: Search Engine Spamming …. Spamming with Indexing

- Two Techniques : **Cloaking and DoorWay pages**

DoorWay Pages

- Low quality dummy web pages with little content.
- It is stuffed with similar keywords and phrases to increase rank
- Search Engine indexes this page high rank.
- Serve no purpose for user.
- When browser requests the doorway page, it is redirected to other page that of for the commercial use for creator.
- Doorway pages are ugly and not interesting to the user.
- They may have a frame and in that frame they display their commercial pages.
- Software can create thousand of pages for a single keyword in minutes.

- **Meta-tag stuffing**
- **Scraper sites**
- **Article spinning:** Rewriting the existing articles. Created thru automated methods and human unreadable.

As a concerted effort to defeat spammers who manipulate the text of web page, newer search engines try to exploit the link structure of the web – a technique known as **link analysis.**
e.g. Google

Pagerank is a link analysis algorithm typical for google,.

Pagerank is a complex algorithm where a **weight** is assigned to a link and a **rank** given to a page based on the weight of the backlinks that page gets.

This rank spreads from **0 to 10** with 0 being a page with no important backlinks and 10 an almost unreachable rank only assigned to pages of great importance like Google and Microsoft.

This algorithm is not created by Google but what Google was based on.

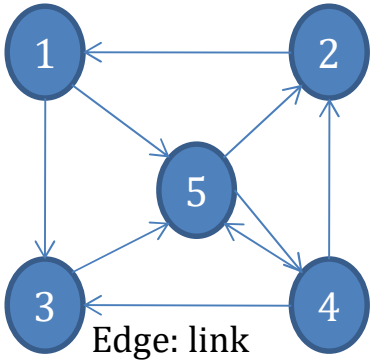
The pagerank algorithm is created by Larry Page and Sergey Brin, creators of Google.

PageRank
Link based Analysis

Query → 

Analyze the **hyperlinks**
And **Graph** Structure.
+
Other factors
= composite score for a web page

Web page: node



Edge: link
Inbound links important

A hypothetical Web Graph

1. The random surfer may visit from current page A to new page
2. **PageRank: Pages with large number of visits are more important than pages with few visits.**
3. Ranking terms used in this page +
   terms used in near links to that page.
4. The near (pointing page) may not be of that spammer

The algorithm based on these concepts is initiated by Google for the first time is called as **PageRank**.

PageRank works by counting the number of links to the page.
by seeing the quality of links
Rough estimate of how important the website to others
More IMPORTANT websites get more links from other websites.

➢**PageRank = Link Analysis function.**

➢It assigns numerical weight to each element of WWW. PR(E)

➢Measures the relative importance of the document in a given set.

➢PR(Element).. Higher the value means more relevant the page.

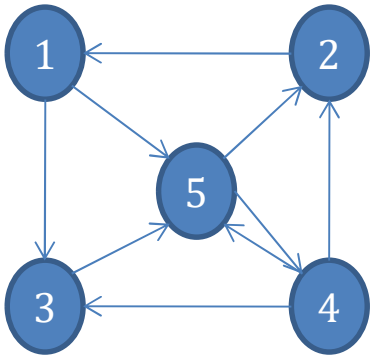
Web page Pair (Pi, Pj) is

M(I, j) = 1/K

K = number of outlinks from page Pj

And one of those is to page Pi

Otherwise **M(i, j) =0**



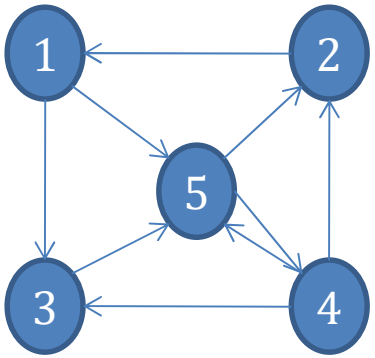
M5=

	1	2	3	4	5
1	0	1	0	0	0
2	0	0	0	1/3	1/2
3	1/2	0	0	1/3	0
4	0	0	0	0	1/2
5	1/2	0	1	1/3	0

5 X 5 matrix

Transition matrix: probability of random surfer reaching the node from the current node.

1. Sum of entries of a column = 1
2. All entries are either 0 or greater than 0.
3. **=Markov Chain Process/matrix**
4. Next node of Markov process depends on the current node he is visiting.



V0=
Col Vector

1/5
1/5
1/5
1/5
1/5

1. The surfer can be at any of theses pages with Probability 1/n
2. Vector V = probability distribution for current location
3. Next state X = M X Vj .. Surfer at node j
4. Slowly visit frequency converges to fixed and steady state quantity.
5. For Markov chain to reach equilibrium, two conditions have to b met. Strongly connected graph and no dead ends.
6. Present in www. Normally true for www.

nX1
n=number of web pages

Multiply v0 by M5 repeatedly,….. After about 60 iterations (60 to 80 iterations for X_k to converge)

1/5	1/5	1/6	0.4313
1/5	1/6	13/60	0.4313
1/5	1/6	2/15	0.3235
1/5	1/10	11/60	0.3235
1/5	11/30	3/10	0.6470

……

0	1	0	0	0
0	0	0	1/3	1/2
1/2	0	0	1/3	0
0	0	0	0	1/2
1/2	0	1	1/3	0

MV0 M(MV0) M(M(MV0)……M^{60}(MV0)

Distribution Vector for next state

Modified PageRank

INITIAL ASSUMTIONS WERE WRONG

Study in 2000 by IBM, AltaVista, Compaq.. (200 million pages, 1.5 billion links)

- **Where no matter you start, very soon you will reach the entire web .. Is not right**
- Entire web is strongly interconnected.
- Older models of web topology..

- Cluster of sites connected to other clusters… all forming
  Strongly Connected Component. (SCC)
- SCC = sub graph of a graph where path exists between every pair of nodes.

- But results of study presented different picture of web.

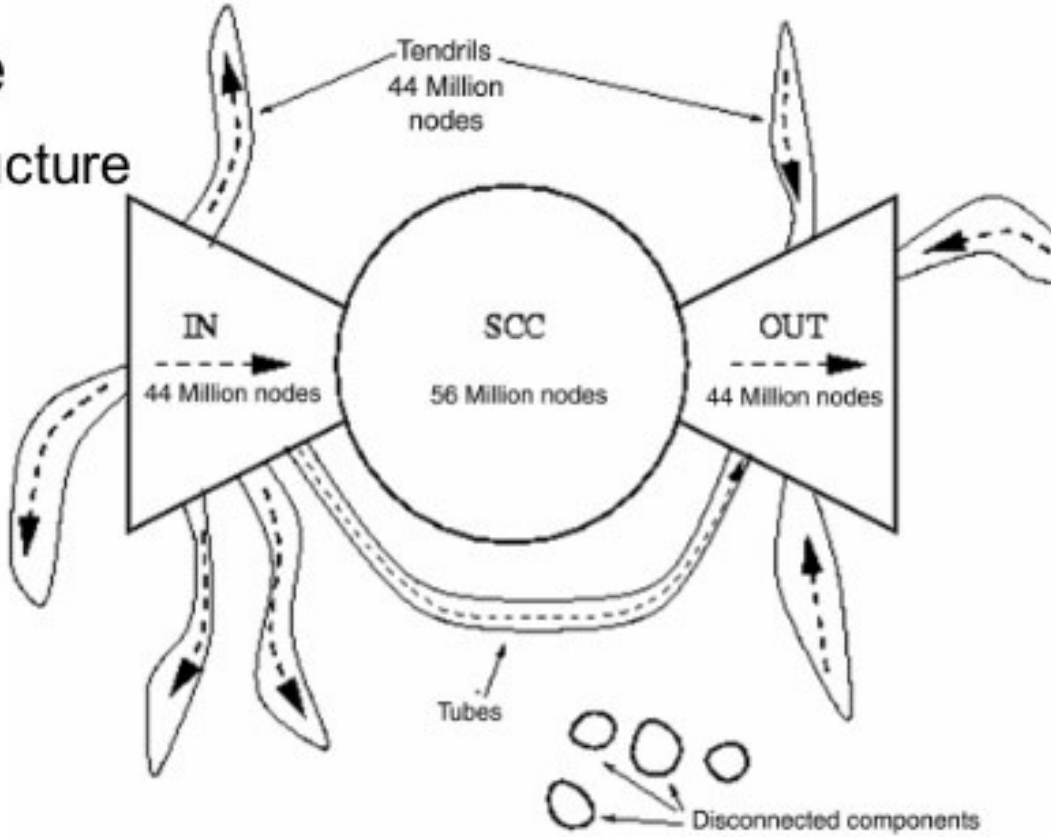
The Structure of Web

Measuring the Web

"Bow-tie" structure

Overall view of the structure of the Web

- SCC
- IN
- OUT
- Tendrils
- Tubes
- Disconnected

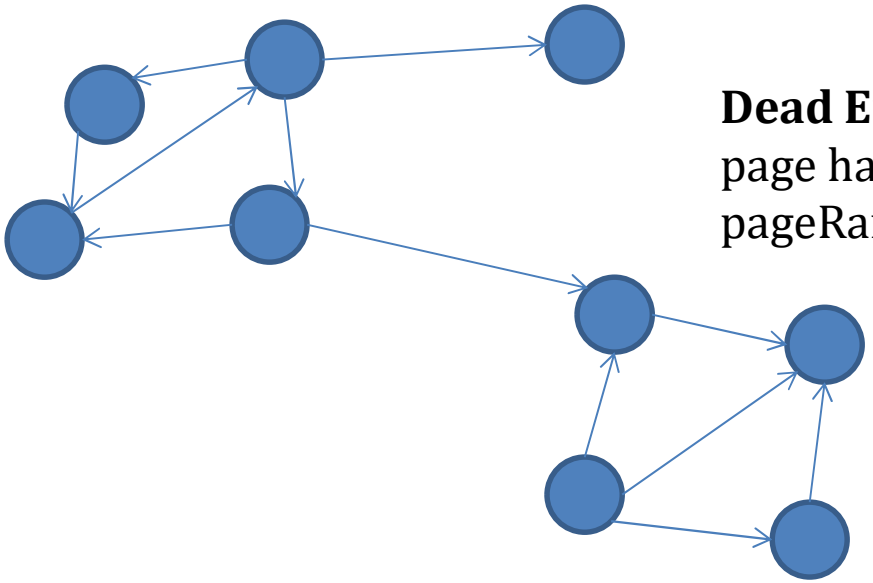


A Broder et al. Graph structure in the Web. Computer Networks. (2000)

The Structure of Web

Sizes: Core is only 1/3 rd of the total.
Originations and termination pages made up 1/4th of the web
Disconnected pages around 1/5th of the web.

As a result of Bow-Tie structure of web, the Markov process do not hold true.
For e.g. :
If surfer landed in OUT component.. He can never leave out.
Probability of surfer visiting SCC or IN component is zero.
This means pages from SCC or IN would end up as low PageRank.



Dead End: pages with no outlinks. No other page has probability to reach. It will lose all its pageRank eventually.

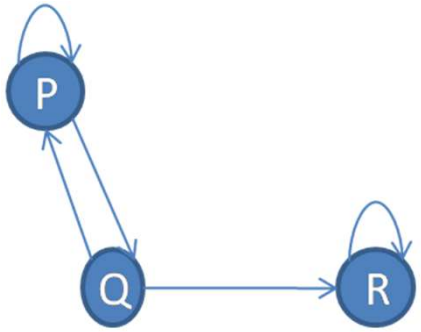
Spider Trap: Set of pages whose outlinks can reach to pages from That set only. Eventually only these set of pages will have any pageRank.

IN all the scenarios "**taxation**" cab help.
It allows the surfer to leave the web at any time and start randomly at new page.
<u>Problem of Dead End</u>

$M^k v$ results in V $\longrightarrow$ 0

$$\begin{bmatrix} X_p \\ X_Q \\ X_R \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 2/6 \\ 1/6 \\ 3/6 \end{bmatrix} \begin{bmatrix} 3/12 \\ 2/12 \\ 7/12 \end{bmatrix} \begin{bmatrix} 5/24 \\ 3/24 \\ 16/24 \end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

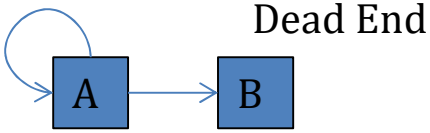
All the PageRank is trapped in R. Once a random surfer reaches R, he can never leave.



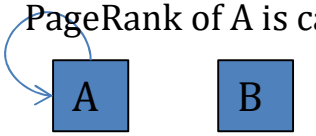
	P	Q	R
P	1/2	1/2	0
Q	1/2	0	0
R	0	1/2	1

Dealing with dead ends

1.Remove all pages with no outgoing links and remove their links too.
2. This may result into more dead ends. Recursively do 1.
3. Record the order of removal.



PageRank of A is calculated

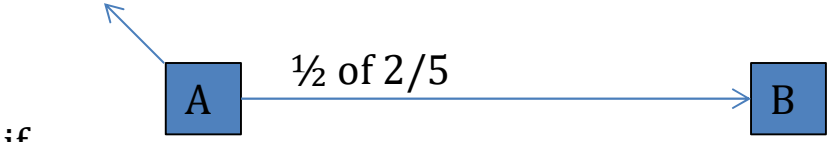


SCC

4. Compute pageRanks of nodes of G.
5. then Restore the graph in reverse order. Compute the pagerank of deadend added.

B added and it uses A to calculate Its pagerank.

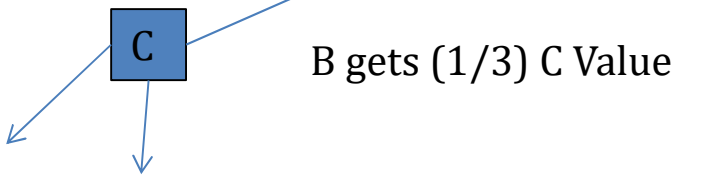
B gets (1/2) A Value



if
PR=2/5

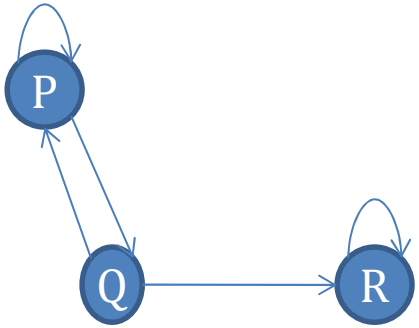
B gets (1/2) A Value

PR =1/2 * 2/5 + 1/3 * 2/7 = 31/105



if
PR= 2/7

B gets (1/3) C Value



	P	Q	R
P	1/2	1/2	0
Q	1/2	0	0
R	0	1/2	1

Avoiding Spider Traps

Teleport operation: random surfer jumps to any other node in web graph. With some probability.

V' = βMv + (1- β)e/n

β

0.8 *

	P	Q	R
P	1/2	1/2	0
Q	1/2	0	0
R	0	1/2	1

1/3
1/3
1/3

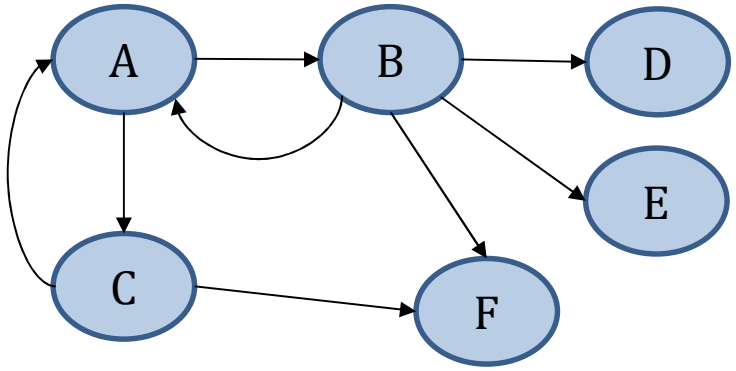
+ (0.2) *

P	1/3
Q	1/3
R	1/3

=

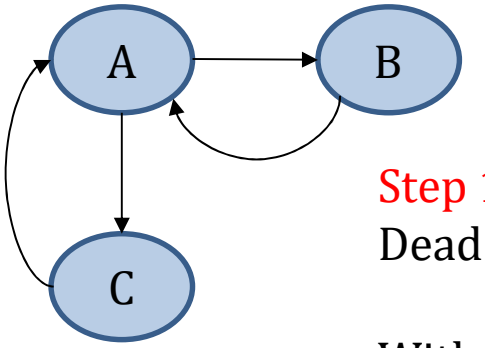
	P
P	
Q	
R	

$$\begin{bmatrix} X_P \\ X_Q \\ X_R \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1.00 \\ 0.60 \\ 1.40 \end{bmatrix} \quad \begin{bmatrix} 0.84 \\ 0.60 \\ 1.56 \end{bmatrix} \quad \begin{bmatrix} 0.776 \\ 0.536 \\ 1.688 \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix}$$



Teleport Factor = 0.8

1- Remove dead ends recursively
2 - For Resulting Graph find the transition Matrix Mv
3 - **V' = βMv + (1- β)e/n** Use this for spider trap handling
4. Calculate PageRank for 2 iterations
5. Add the removed nodes in reverse order and calculate their pagerank.



Step 1 – Remove Dead Ends
Dead end is a page with no link out.

With dead ends the matrix will no longer stochastic. [sum of all rows in column=1] because for dead end node all will be zeros

The nodes D,E,F are dead nodes. So remove them.

Check if removal of D,E,F results into any other dead end.

Finally the resulting graph have 3 nodes.