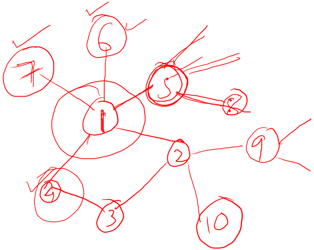
EIGEN VECTOR CENTRALITY (Refer 7.2 of Newman text)
KATZ CENTRALITY AND (Refer 7.3 of Newman text)
JACCARD COEFFICIENT

Eigen Vector Centrality



Not all neighbors of a vertex are equivalent.

The importance of a vertex increases due to those neighbors that are important

So, give a score proportional to the sum of the scores of its neighbours.

Generalizes the degree centrality by incorporating the importance of the neighbours.

$Ax = \lambda x$.

$$\text{Degree}(1) = \frac{\boxed{\deg(1)}}{n-1}$$ ← for normalizing.

$$= \frac{5}{6}$$

Perron - Frobenius Theorem : Asserts that a [real square matrix](#) with positive entries has a unique largest real [eigenvalue](#) and that the corresponding [eigenvector](#) can be chosen to have strictly positive components

Adj. matrix

$$x_v = \sum_{j \in N(v)} x_j = \sum A_{vj} x_j$$

Centrality Value for node v

$$x_v = \frac{1}{\lambda} \sum_{j \in N(v)} A_{vj} \cdot x_j$$

$$\lambda x_v = \sum A_{vj} \cdot x_j$$

$$\lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \end{bmatrix} = A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix}$$

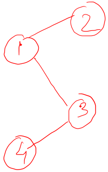
Node

$$\begin{array}{c|c} 1 & \\ 2 & \\ 3 & \\ 4 & \end{array} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & & & \end{bmatrix}$$

$$\lambda x = A x$$

λ = eigenvalue

$$\lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} = A \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}$$



$A = A^T$

Undirected

$$x_1 = \frac{1}{\lambda} \sum A_{1j} x_j$$

$$x_2 = \frac{1}{\lambda} \sum A_{2j} x_j$$

$$x_3 = \frac{1}{\lambda} \sum A_{3j} x_j$$

Let x_v be the centrality value of v.

Then $x_v = \sum_{j \in N(v)} x_j = \sum A_{vj} x_j$

($\because$ we need to proportionately consider nbrs of v while computing centrality)

We can have summation bound

$$x_v = \frac{1}{\lambda} \sum_j A_{vj} x_j$$

Normalizing constant

$$\therefore \lambda x_v = \sum_j A_{vj} x_j$$

If we generalize and use matrix notaⁿ

$$\lambda \underbrace{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}_{x} = A \underbrace{\begin{bmatrix} \\ \\ \\ \end{bmatrix}}_{x}$$

Thus,

$\lambda x = A x$ → Eigen vector formulaⁿ

Various eigen values are possible.

We chose non-negative eigen values.

Preferably, the maximum eigen value.

We choose eigen vector corresponding to largest eigen value due to Perron-Frobenius Theorem.

(Unique largest eigen value and corresponding eigen vector will have all positive entries)

Compute the eigenvalues of A
Select the largest eigenvalue
Get the corresponding eigen vector Ce

Which are wide-reaching influential nodes ?

Example :

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

$$\lambda x = A x$$

$$(A - \lambda I) x = 0$$

$$\begin{bmatrix} 0-\lambda & 1 & 0 \\ 1 & 0-\lambda & 1 \\ 0 & 1 & 0-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0.$$

$$\det (A - \lambda I) = 0$$

$$\lambda = ? \Longrightarrow (-\sqrt{2}, 0, +\sqrt{2})$$

$$(0-\lambda)\left((0-\lambda)^2 - 1\right) - 1(0-\lambda) + 0 = 0$$

$$-\lambda^3 + 2\lambda = 0.$$

$$\lambda(2 - \lambda^2) = 0$$

$$\lambda = 0 \; ; \; \lambda = -\sqrt{2} \; ; \; \lambda = +\sqrt{2}$$

$$\begin{bmatrix} -\sqrt{2} & 1 & 0 \\ 1 & -\sqrt{2} & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$v_2 \quad \begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1/2 \\ \sqrt{2}/2 \\ 1/2 \end{bmatrix}$$

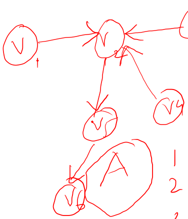
Norm

Left eig vect	Right eig
$\lambda x = Ax$	$\lambda x = A^T x$
In-deg.	Out-deg.

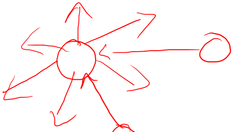
$A = A^T$ in digraph??

$A \neq A^T$ in digraph

$A \neq A^T$

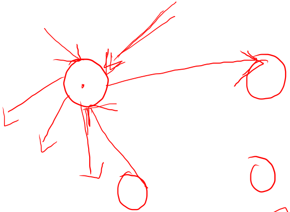


$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & \end{bmatrix}$$



A^T

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$



out-deg.

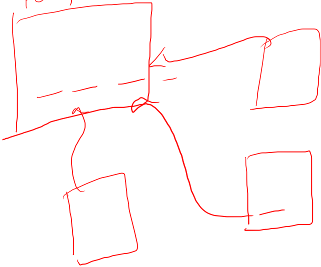
in-deg.

0.6
1.2
2.9
3.8

2.6
1.2
0.5
0.9

A $\quad$ A^T

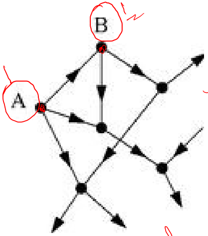
Portal



$A \,/\, A^T$

in-deg / out-deg.

Local as well as global.



global

Considering in-edges

Vertex A has eigencentrality 0 (const)
Vertex B also has eigencentrality 0 (const)

For directed graph, incoming edges are considered.

A and A.T(2 eigen vectors possible)

when G is weakly connected, Eigenvector Centrality fails.

KATZ CENTRALITY

$$x_v = \frac{1}{\lambda} \sum A_{vj} x_j + \beta$$

controlling

bias term

$$\alpha = 1/\lambda$$

$\alpha ; \beta$ are constants.

If $\alpha >>$; effect of β reduce.

$$x_v = \alpha \sum A_{vj} x_j + \beta$$

$\lambda \rightarrow$ max eigen value

$$0 < \alpha < 1/\lambda .$$

vector form

$$\underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{x} = \alpha A \underbrace{\begin{bmatrix} \\ \\ \end{bmatrix}}_{x} + \beta \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$x = \alpha A x + \beta \cdot 1$$

$$x (I - \alpha A) = \beta \cdot 1$$

$$x = (I - \alpha A^{T})^{-1} \cdot \beta \cdot 1 \Rightarrow \beta \cdot (I - \alpha A)^{-1} \cdot I .$$

KATZ Centrality

$$x = \beta \cdot \boxed{(I - \alpha A^T)^{-1}} \cdot I$$

If $\alpha = 0$

$$x = \beta \cdot (I)\, I \;;\; \beta \text{ is a constant.}$$

$\det(I - \alpha A^T) \neq 0$ (Invertible)

$\det(I - \alpha A^T) = 0$ (Non-invertible)

$\alpha = 1/\lambda$; Matrix will non-invertible.

$$\alpha = \frac{1}{\lambda}$$

$$0 < \alpha < 1/\lambda$$

So $0 < \alpha < 1/\lambda$ ← Max eigen value

$\alpha ; \beta$

Power Iteration

$$x^0 = 1$$

$$x^{t+1} = \alpha A x^t + \beta \;;\; \alpha \,\&\, \beta \text{ are constants.}$$

$\vdots$ x^t converges.

KATZ CENTRALITY

(To find score betn 2 vertices).

BASICS

Katz$_\beta$ (Exponentially Damped Path Counts)

- This heuristic defines a measure that directly sums over collection of paths, exponentially damped by length to count short paths more heavily.
- The Katz-measure is a variant of the shortest-path measure.
- The idea behind the Katz-measure is that the more paths there are between two vertices and the shorter these paths are, the stronger the connection.

KATZ

(u, v)

$$score(x, y) = \sum_{l=1}^{\infty} \beta^{l} |paths_{x,y}^{(l)}|$$

where $\beta > 0$ and $paths_{x,y}$ is the set of all length-l paths from x to y.

KATZ: EXAMPLE

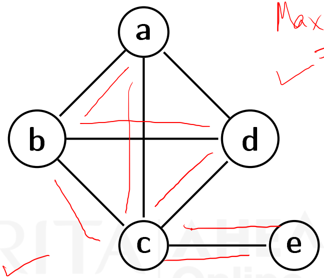
Linear graph

Max- 4

distance

Max path length = 4 if 5 vertices

n-1



(b, e)

0 - 1

$\beta = \frac{1}{2}$

$|path^1_{b,e}| = 0$, $|path^2_{b,e}| = 1$,
$|path^3_{b,e}| = 2$, $|path^4_{b,e}| = 1$

$$score(b,e) = \sum_{l=1}^{4} \beta^l |paths^{(l)}_{b,e}|$$

$$= \frac{1}{2} \times 0 + \frac{1}{4} \times 1 + \frac{1}{8} \times 2 + \frac{1}{16} \times 1$$

$$= \frac{9}{16}$$

$|path^1_{a,c}| = 0$, $|path^2_{a,c}| = 1$,
$|path^3_{a,c}| = 2$, $|path^4_{a,c}| = 2$

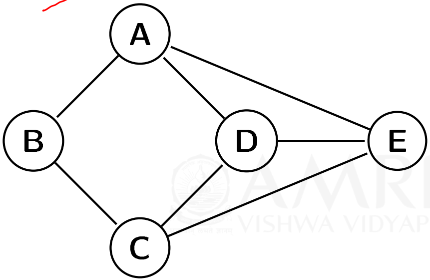
$$score(a,c) = \sum_{l=1}^{4} \beta^l |paths^{(l)}_{a,c}|$$

$$= \frac{1}{2} \times 0 + \frac{1}{4} \times 1 + \frac{1}{8} \times 2 + \frac{1}{16} \times 2$$

$$= \frac{5}{8}$$

JACCARD COEFFICIENT

- $\Gamma(x)$ is a set of neighbours of x

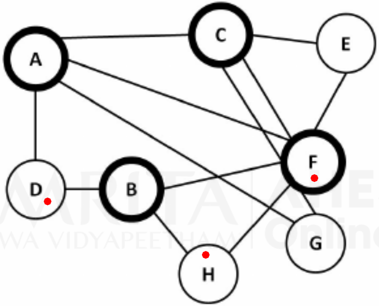


$\Gamma(A) = \{B, D, E\}$
$\Gamma(B) = \{A, C\}$
$\Gamma(C) = \{B, D, E\}$
$\Gamma(D) = \{A, C, E\}$
$\Gamma(E) = \{A, C, D\}$

- **Jaccard's coefficient:** Measure how likely a neighbour of x is to be a neighbour of y and vice versa

$$J(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

Jaccard Example



$\Gamma(A) = \{C, D, F, G\}$

$\Gamma(B) = \{D, F, H\}$

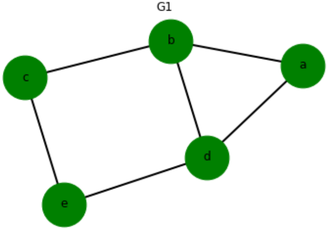
$J(A, B) = \frac{|\Gamma(A) \cap \Gamma(B)|}{|\Gamma(A) \cup \Gamma(B)|} = 2/5$

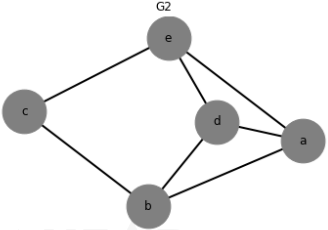
$\Gamma(B) = \{D, F, H\}$

$\Gamma(C) = \{A, E, F, G\}$

$J(B, C) = \frac{|\Gamma(B) \cap \Gamma(C)|}{|\Gamma(B) \cup \Gamma(C)|} = 1/6$

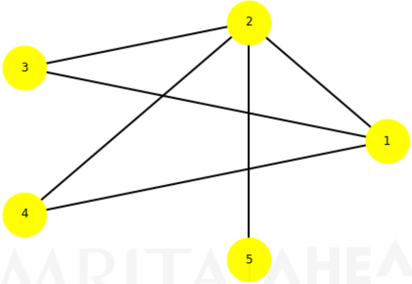
In NetworkX






```
G1=nx.Graph([('b', 'c'),('b', 'd'),('c', 'e'),('d', 'e'),
             ('a', 'b'),('a','d')])
jaccards = nx.jaccard_coefficient(G1, [('a', 'd')])
for u, v, p in jaccards:
    print(f"({u}, {v}) -> {p:.8f}")
(a, d) -> 0.25000000
jaccards = nx.jaccard_coefficient(G2, [('a', 'd')])
(a, d) -> 0.5000000
```

In NetworkX..



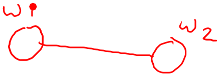

```
G3 = nx.Graph([(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (2,5)])
katz_centrality(G3, alpha=0.1, beta=1.0, max_iter=1000,
tol=1e-06, nstart=None, normalized=True, weight=None)
{1: 0.4713261328363417,
 2: 0.5063185248438331,
 3: 0.43204896443989,
 4: 0.43204896443989,
 5: 0.384916370862998}
```

Objectives

Link Analysis

- Random Surfing Model
- Simplified PageRank

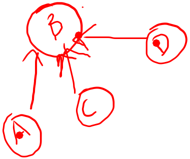
Link Analysis (1/1)

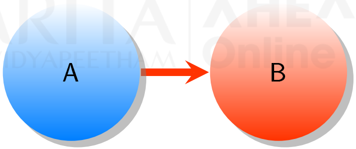


SEO

Why Link Analysis

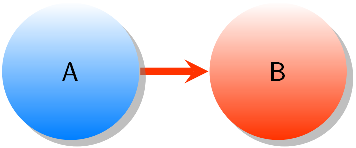
- The web is not just a collection of documents - its hyperlinks are important!
- A link from page A to page B may indicate:
    - A is related to B, or
    - A is recommending, citing, voting for or endorsing B





- Links are either
    - Referential - click here and get back home, or
    - Informational - click here to get more detail
- Links, effect the ranking of web pages and thus have commercial value.

Link Analysis ..



- A recommends B
- A specifically does not recommend B
- B is an authoritative reference for something in A
- A & B are about the same thing (topic locality)

What is Page Rank?

A method for rating the importance of web pages objectively and mechanically using the link structure of the web.

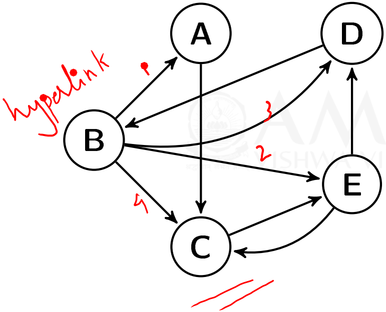
Page Rank

- PageRank was developed by **Larry Page** (hence the name Page-Rank) and **Sergey Brin**.
- In short PageRank is a vote, by all the other pages on the Web, about how important a page is.
- A link to a page counts as a vote of support.
- Iterative Algorithm

Random Surfer Model

- Model helpful for understanding PageRank
- The Random Surfer starts at a randomly chosen page and selects a link at random to follow
- PageRank of a page reflects the probability that the surfer lands on that page after clicking any number of links

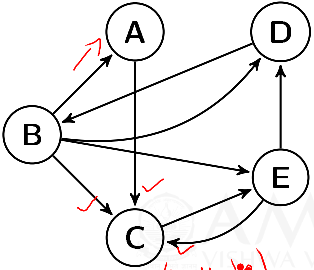
Example of Random Surfer Model



out links

- If Start at A:
  - 1 probability of going to C
- If Start at: B
  - 1/4 probability of going to A
  - 1/4 probability of going to C
  - 1/4 probability of going to D
  - 1/4 probability of going to E
- Choose: E
  - 1/2 probability of going to C
  - 1/2 probability of going to D

EXAMPLE OF RANDOM SURFER MODEL



- Give all pages same PR to start ($1/|P|$) & iteratively calculate new PR
- $PR(A) = PR(B) = PR(C) = PR(D) = PR(E) = 1/5 = 0.2$

0th itera^n

Epoch / itera^n

1st itera^n

inhale / outhale

$$
\begin{aligned}
PR(A) &= PR(B)/4 = 0.2/4 = 0.05 \\
PR(B) &= PR(D)/1 = 0.2 \\
PR(C) &= PR(A)/1 + PR(B)/4 + PR(E)/2 = 0.2 + 0.2/4 + 0.2/2 = 0.35 \\
PR(D) &= PR(B)/4 + PR(E)/2 = 0.2/4 + 0.2/2 = 0.15 \\
PR(E) &= PR(B)/4 + PR(C)/1 = 0.2/4 + 0.2/1 = 0.25
\end{aligned}
$$

RANDOM SURFER..

- Itration table shows the pagerank values for nodes
- Observe that by tenth iteration most of the nodes are getting converged

Iterations					
	A	B	C	D	E
0	0.2	0.2	0.2	0.2	0.2
1	0.05	0.2	0.35	0.15	0.25
2	0.05	0.15	0.225	0.175	0.40
3	0.0375	0.175	0.2875	0.2375	0.2625
4	0.04375	0.2375	0.2125	0.175	0.33125
5	0.059375	0.175	0.26875	0.225	0.271875
	..	..	..	..	..
	..	..	..	..	..
9	0.05857	0.1830	0.2475	0.240	0.27093
10	0.05859	0.1825	0.2470	0.235	0.27691

Simplified PageRank

- Pageranks share the probability space, ie between 0 and 1
- Initial pageranks for nodes will be $\frac{1}{|P|}$, where |P| is number of web pages.

$$\boxed{PR(P_i) = \sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|}}$$

B_{P_i} = Set of pages pointing P_i

$|P_j|$ = Number of outlinks from P_j

OBJECTIVES

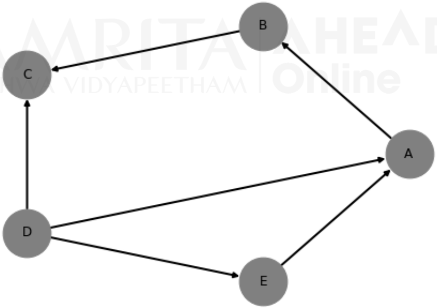
Google PageRank

- Dangling Nodes & Loops
- Teleporting

Page Rank

Rank Sinks

- Dangling Nodes
  - What if we go to C? We're stuck at a dead-end!
  - Solution: Teleport to any other page at random



0^{th} iter

$PR(C) = 0.2 = 1/5$

1^{st} iteration

$$PR(C) = \frac{PR(D)}{3} + \frac{PR(B)}{1}$$

$$= \frac{0.2}{3} + 0.2$$

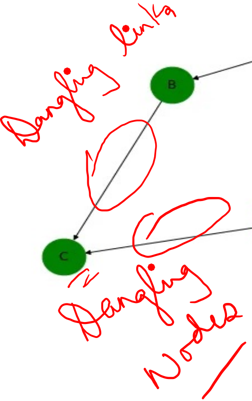
$$= 0.26$$

$PR(B) = PR(A) = 0.2$

$PR(D) = 0$

$PR(E) = \frac{0.2}{3} = 0.06$

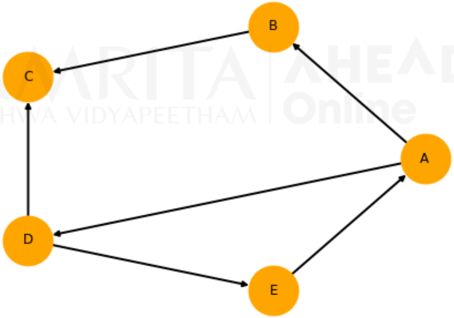
$PR(A) = 0.26$



Itr. No.	A	B	C	D	E
1	0.266667	0.200000	0.466667	0.000000	0.066667
2	0.066667	0.266667	0.666667	0.000000	0.000000
3	0.000000	0.066667	0.933333	0.000000	0.000000
4	0.000000	0.000000	1.000000	0.000000	0.000000
5	0.000000	0.000000	1.000000	0.000000	0.000000

Google PageRank

- What if we get stuck in an a cycle?
  - A->D->E->A
- Solution: Teleport to any other page at random



Google PageRank..

Teleport

- Solution is to add a teleportation probability α to every decision α % chance of getting bored and jumping somewhere else.
- (1-α)% chance of choosing one of the available links
- $\alpha = .15$ is typical

Google PageRank

0.2

0.02

0.15

PageRank Definition

Teleportation probability

Sum of PR of all pages pointing to P_i

$$PR(P_i) = \frac{\alpha}{|P|} + \boxed{(1-\alpha)} \cdot \boxed{\sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|}}$$

(0-1)

PageRank of page P_i

Total num of pages

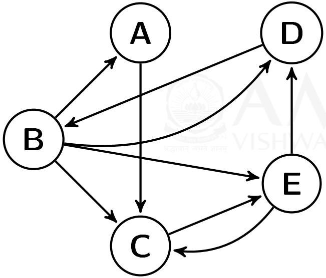
B_{Pi} is set of all pages pointing to P_i

Number of outlinks from P_j

Computation

- PageRank is computed over and over until it converges, around 20 times
- Notice that when $\alpha=0$, page rank equation reduces to $PR(P_i) = \sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|}$

Google PageRank: Example



- Give all pages same PR to start (1/|P|) & iteratively calculate new PR
- PR(A) = PR(B) = PR(C) = PR(D)= PR(E) = 1/5 =0.2
- α=0.15

0.15/5 -

$$
\begin{aligned}
PR(A) &= .03 + .85 \times PR(B)/4 = .0725 \\
PR(B) &= .03 + .85 \times PR(D)/1 = .2 \\
PR(C) &= .03 + .85(PR(B)/4 + PR(E)/2) \\
&= .03 + .85(.2/1 + .2/4 + .2/2) \\
&= .3275 \\
PR(D) &= .03 + .85(PR(B)/4 + PR(E)/2) \\
&= .1575 \\
PR(E) &= .03 + .85(PR(B)/4 + PR(C)/1) \\
&= .2425
\end{aligned}
$$

Example ..

Iterations			*PageRanks*		
	A	B	C	D	E
0	0.2	0.2	0.2	0.2	0.2
1	0.0725	0.2	0.3275	0.1575	0.2425
2	0.0725	0.1639	0.2372	0.1756	0.3508
	..	..	..	..	..
	..	..	..	..	..
	..	..	..	..	..
99	0.071	0.194	0.254	0.193	0.287
100	0.071	0.194	0.254	0.193	0.287

OBJECTIVES

PageRank in Matrix Form

- Simplified PageRank
- Google PageRank

Google PageRank as Matrix

Matrix equation

$$R = TR$$

where,
R is the vector of PageRank values
T or M is the matrix for transition probabilities
T_{ij} is the probability of going from page j to i:

$$T_{ij} = \begin{cases} \frac{\alpha}{|P|} + (1-\alpha)\frac{1}{|P_j|}; & \textit{if a link from page } j \textit{ to } i \textit{ exists} \\ \frac{\alpha}{|P|}; & \textit{otherwise} \end{cases}$$

Simplified PageRank as Matrix

Matrix equation

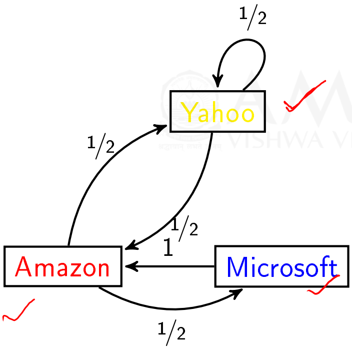
$$\boxed{R = TR}$$

where,
R is the vector of PageRank values
T or M is the matrix for transition probabilities
T_{ij} is the probability of going from page j to i
Substitute $\alpha = 0$ in Google PageRank formula

$$T_{ij} = \begin{cases} \frac{1}{|P_j|}; \ \textit{if a link from page } j \textit{ to } i \textit{ exists} \\ 0; \ \textit{otherwise} \end{cases}$$

Simplified PageRank: Example

$$T = \begin{array}{c} Y \\ A \\ M \end{array} \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} Y \\ A \\ M \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$



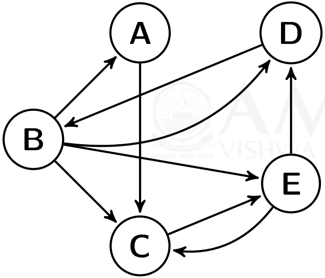
$$i. \quad \underbrace{\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}}_{R} = \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} \; R$$

$$ii. \quad \underbrace{\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}}_{R} = \begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} \; R$$

$$iii. \quad \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix}$$

$$iv. \quad \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 20/48 \\ 17/48 \\ 11/48 \end{bmatrix}$$

GOOGE PAGERANK: A CLOSER LOOK



PageRank Matrix Example

No link from C to A so value = 0.15/5

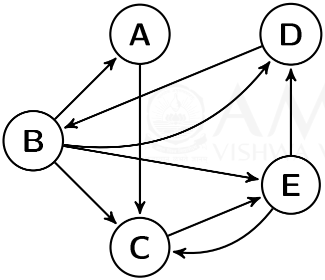
$$
\begin{array}{c}
A \\ \\ \\ \\ E
\end{array}
\begin{bmatrix} PR_A \\ PR_B \\ PR_C \\ PR_D \\ PR_E \end{bmatrix}
=
\begin{bmatrix}
. & . & .03 & . & . \\
. & . & . & . & . \\
. & . & . & . & . \\
. & . & . & . & . \\
. & .243 & . & . & .
\end{bmatrix}
\begin{bmatrix} .2 \\ .2 \\ .2 \\ .2 \\ .2 \end{bmatrix}
$$

$$T_{EB} = \frac{\alpha}{|P|} + (1 - \alpha)\frac{1}{|P_B|}$$

= 0.15/5 + (1 – 0.15)/4 = 0.243

Init PR values
1/|P|

Google PageRank: Example



$$\begin{bmatrix} PR_A \\ PR_B \\ PR_C \\ PR_D \\ PR_E \end{bmatrix} = \begin{bmatrix} .03 & .2425 & .03 & .03 & .03 \\ .03 & .03 & .03 & .88 & .03 \\ .88 & .2425 & .03 & .03 & .455 \\ .03 & .2425 & .03 & .03 & .455 \\ .03 & .2425 & .88 & .03 & .03 \end{bmatrix} \begin{bmatrix} .2 \\ .2 \\ .2 \\ .2 \\ .2 \end{bmatrix}$$

R_1	R_2	R_3	R_4		R_{10}
0.072500	0.072500	0.064823	0.068086		0.070966
0.200000	0.163875	0.179228	0.211854		0.194972
0.327500	0.237188	0.275570	0.236420	......	0.254335
0.157500	0.175563	0.213945	0.181320		0.193181
0.242500	0.350875	0.266433	0.302321		0.286546

References

Hagberg, A., Swart, P., and S Chult, D.
Exploring network structure, dynamics, and function using networkx.
Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Newman, M.
Networks.
Oxford university press, 2018.

Zinoviev, D.
Complex network analysis in Python: Recognize-construct-visualize-analyze-interpret.
Pragmatic Bookshelf, 2018.