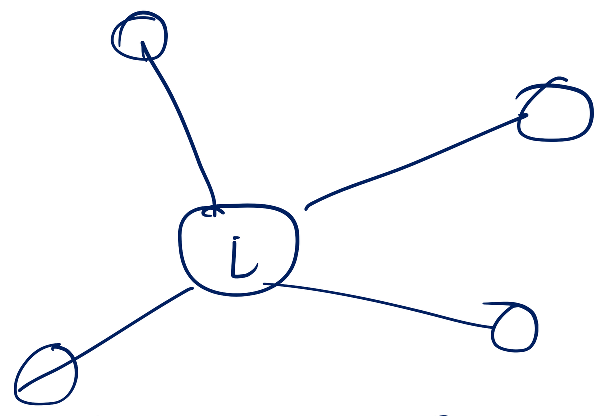
Centrality Measures

1) Degree Centrality

2). Eigen vector Centrality



If A is the adjacency matrix of an undirected network then $A_{ij} = 1$ if nodes i & j are connected

$A = \begin{bmatrix} & \\ & \end{bmatrix}$

Let $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_i \\ x_n \end{bmatrix}$ be the vector that denotes the importance of the nodes 1 to n

So importance of node i given as $x_i = \sum_{j=1}^{n} A_{ij} x_j$

$\vec{x}' = A\vec{x}$

$A_{i\cdot}$

$\vec{x}'$ is the Eigen vector Centralities of the nodes in the network.

$\vec{x}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

$\vec{x}^{(1)} = A\vec{x}^{(0)}$

$\vec{x}^{(2)} = A\vec{x}^{(1)} = A^2\vec{x}^{(0)}$

$\vec{x}^{(3)} = A\vec{x}^{(2)} = A^3\vec{x}^{(0)}$

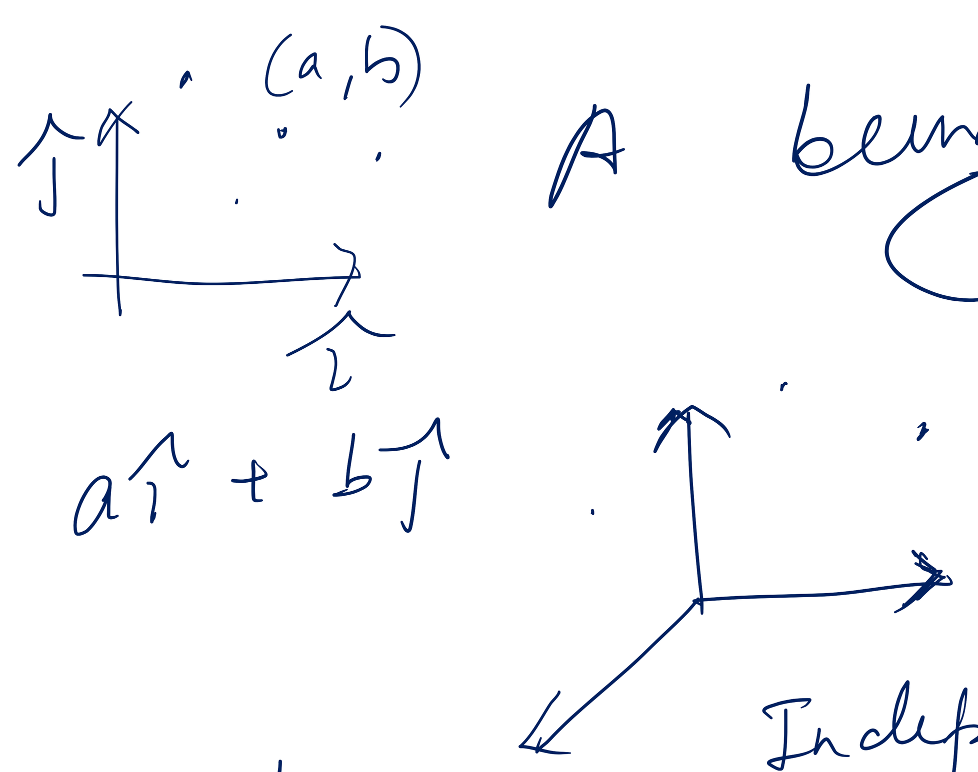
$\vdots$

$x^{(t)} = A^t x^{(0)}$

After repeated iterations

$$|x^{(t+1)} - x^{(t)}| < \epsilon$$

Let $\vec{x}^{(0)}$ be a linear combination of the Eigen vectors of A.



Column space

Independent vectors

Basis

A being a symmetric matrix with real values. $A \in \{0,1\}^{n \times n}$

- There will be n Eigen vectors
- These Eigen vectors will be orthogonal to each other.

Let $\vec{v}_1, \vec{v}_2, \dots \vec{v}_n$ be the n Eigen vectors of A with corresponding Eigen values $K_1, K_2, \dots K_n$

Assume $K_1 > K_2 > K_3 \dots > K_n$

$\boxed{AX = \lambda X}$ $\lambda \to K$

$$x^{(0)} = c_1\vec{v}_1 + c_2\vec{v}_2 + \dots c_n\vec{v}_n$$

$$A x^{(0)} = A\left[c_1\vec{v}_1 + c_2\vec{v}_2 + \dots + c_n\vec{v}_n\right]$$

$$= c_1 A\vec{v}_1 + c_2 A\vec{v}_2 + \dots + c_n A\vec{v}_n$$

$$= c_1 K_1 \vec{v}_1 + c_2 K_2 \vec{v}_2 + \dots c_n K_n \vec{v}_n$$

$$= K_1\left[c_1\vec{v}_1 + c_2\frac{K_2}{K_1}\vec{v}_2 + \dots c_n\frac{K_n}{K_1}\vec{v}_n\right]$$

$$x^{(1)} = A x^{(0)}$$

$$x^{(2)} = A x^{(1)} = A^2 x^{(0)}$$

$$\boxed{AX = KX}$$

Then $A^2 X = KAX = K^2 X$

$$A^3 X = A \cdot A^2 X = A \cdot (K^2 X) = K^2 A X = K^3 X$$

$$A^t X = K^t X$$

$$A^{\{t\}}\vec{x}^{(0)} = A^{\{t\}}\left[c_1\vec{v}_1 + c_2\vec{v}_2 + \dots c_n\vec{v}_n\right]$$

$$= c_1 A^{\{t\}}\vec{v}_1 + c_2 A^{\{t\}}\vec{v}_2 + \dots + c_n A^{\{t\}}\vec{v}_n$$

$$= c_1 K_1^t \vec{v}_1 + c_2 K_2^t \vec{v}_2 + \dots + c_n K_n^t \vec{v}_n$$

$$= K_1^t\left[c_1\vec{v}_1 + c_2\left(\frac{K_2}{K_1}\right)^t\vec{v}_2 + \dots + c_n\left(\frac{K_n}{K_1}\right)^t\vec{v}_n\right]$$

So as $t \to \infty$, K_1 being the largest Eigen value

all $\left(\frac{K_i}{K_1}\right)^t \to 0$ for $i \neq 1$

Hence $A^t \vec{x}^{(0)} = c_1 K_1^t \vec{v}_1$

$A^t \vec{x}^{(0)} = c_1 k_1^t \vec{v}_1 = x^{(t)}$ (Eigen Vector Centrality)

~~The Eigenva~~ The Eigen vector centrality ~~is of~~ of the nodes is proportional to the Eigen vector of the adjacency matrix A, corresponding to its largest Eigen value.

$Ax^{(0)} \rightarrow A(Ax^{(0)}) \ldots A(A(Ax^{(0)}))$ - - - - -

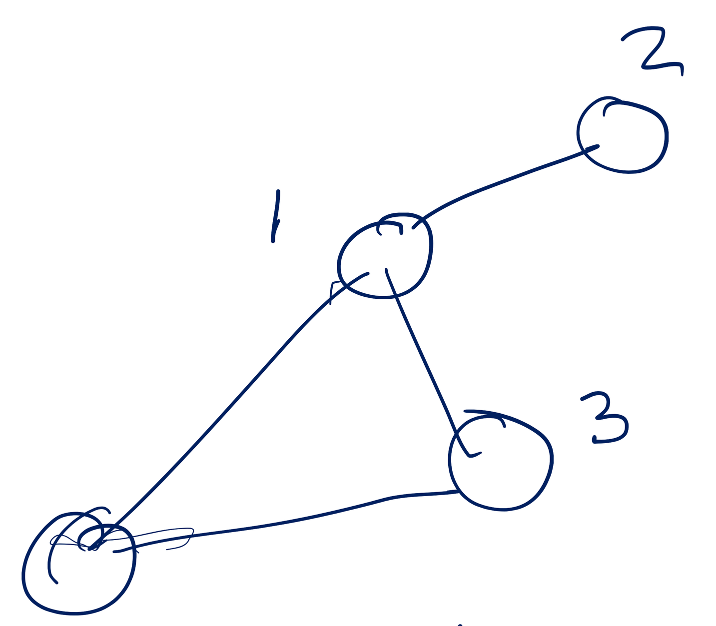
~~Reach at~~ Converge towards the Eigen vector, corresponding to largest Eigen value of A

This is also called as power iteration method to derive v_1 & k_1

Linear Algebra and its Applications (By Gilbert Strang)

~~$x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$~~

$x = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$



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

$A_{ij} = 1$

$\boxed{A^T}_{ij} = 1$

$i \rightarrow j$

$A_{ji} = 1$

$AX = \lambda X$

$A^T X' = \lambda' X'$

$Ax = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$

$A^2 x = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$

Problem of Eigen Vector Centrality

- Centrality of the nodes become 0, if the network is an acyclic directed network.

- $x_i = \sum_{\forall j} A_{ij} x_j$ (Eigen Vector Centrality)

(Katz Centrality)

$x_i = \alpha \sum_{\forall j} A_{ij} x_j + \beta$ (Katz Centrality)

$\vec{X} = \alpha A \vec{X} + \beta \vec{\mathbb{1}}$ $\quad \vec{1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ (vector of all 1's)

or $(I - \alpha A)\vec{X} = \beta \vec{1}$

or $\vec{x} = \beta (I - \alpha A)^{-1} \vec{1}$

What should be the value of α?

Inverse $(I - \alpha A)^{-1}$ must exist

$|I - \alpha A| \neq 0$

or $|(A - \alpha^{-1} I)| \neq 0$ — (1)

Seen this expression $A - \alpha^{-1} I$

Eigen vector $AX = \lambda X$

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

we want $x \neq 0$

so $|A - \lambda I| = 0$ ← Eigen value

So when λ is the Eigen value of A then $|A - \lambda I| = 0$

For Eqn (1), we want to avoid this condition

so α^{-1} should not be the Eigen value of A

~~So as α is increased~~

Let the Eigen values of A in ~~descend~~ descending order be $(K_1, K_2 \ldots K_n)$ ← largest, ← smallest

So as α increases

α^{-1} will move towards K_1

so $\alpha^{-1} > K_1$

$\alpha < 1/K_1$

A minor Extension to Katz Centrality

Rather than giving everybody equal amount of fee importance, ie β, give different amount $\vec{\beta}$

$$x_i = \alpha \sum A_{ij} x_j + \beta_i$$

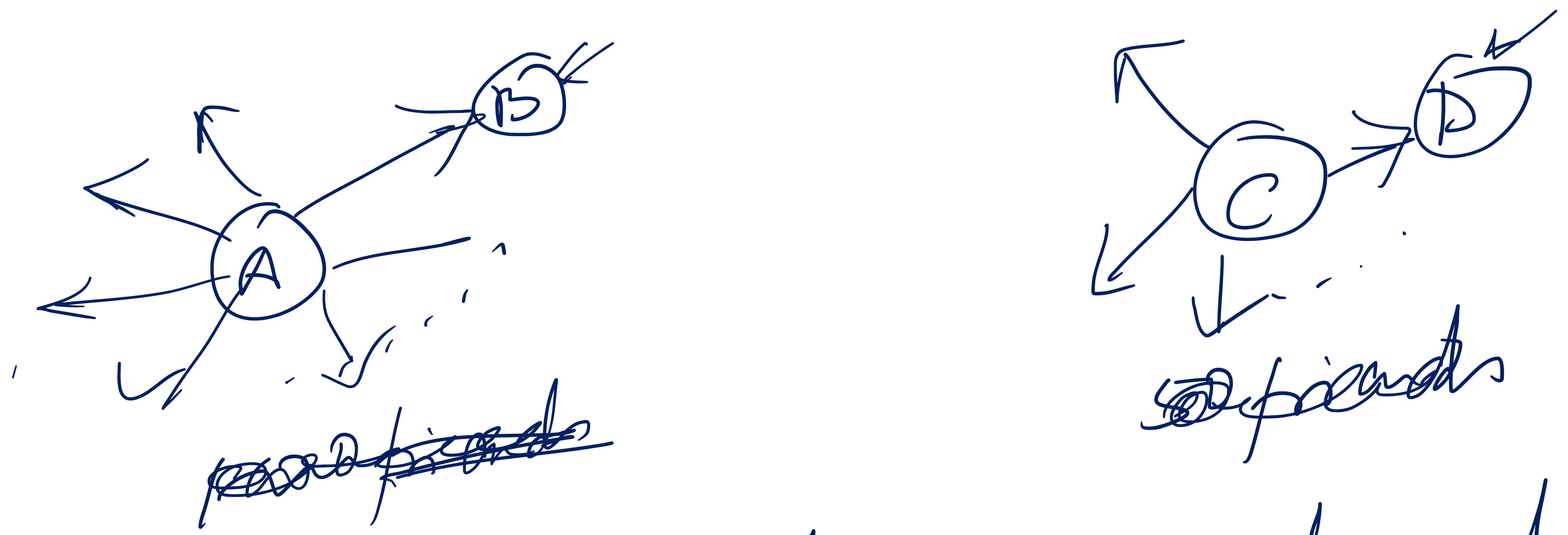
or $\vec{X} = \alpha A \vec{x} + \vec{\beta}$

or $(I - \alpha A)\vec{x} = \vec{\beta}$

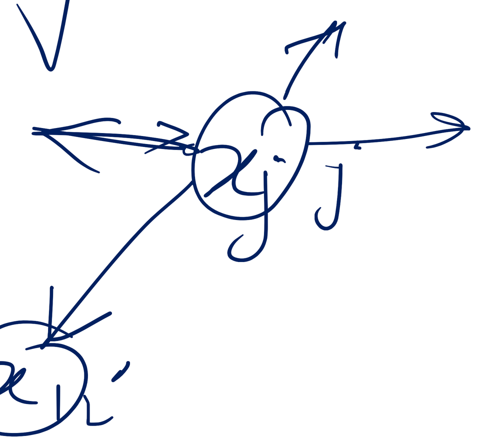
or $\vec{x} = (I - \alpha A)^{-1} \vec{\beta}$

Solves the problem of Eigen vector Centrality in directed networks.

- However one major issue



Let A has higher centrality as compared to C
but A has higher number of outgoing links as compared to C



Page Rank Centrality

$$x_i = \alpha \sum_{\forall j} A_{ij} \frac{x_j}{K_j^{out}} + \beta$$

$$D_{out} = \begin{bmatrix} K_1^{out} & 0 & 0 \\ 0 & K_2^{out} & 0 \\ 0 & 0 & \ddots \; K_n^{out} \end{bmatrix}$$

$$D_{out}^{-1} = \begin{bmatrix} 1/K_1^{out} & & \\ & 1/K_2^{out} & \\ & & \ddots \; 1/K_n^{out} \end{bmatrix}$$

$$\vec{x} = \alpha A D^{-1} \vec{x} + \beta \vec{I}$$

or $(I - \alpha A D^{-1})\vec{x} = \beta \vec{I}$

or $\cancel{\vec{x} = (I - \alpha A D^{-1})^{-1} \beta \vec{I}}$

or $\vec{x} = \beta (I - \alpha A D^{-1})^{-1} \vec{I}$

So what shout be the values of α

$\alpha < \frac{1}{K_1}$ where K_1 is the ~~Eigen~~ largest Eigen value of AD^{-1}

PageRank has been popularly used for ranking webpages.

- Based on the notion that all web pages w.r.t a query is not important.
  → Based on different measures like trustworthiness, popularity or relevance

PageRank uses a web graph link structure to rank the pages.

__webgraph__

↳ Links between webpages (WWW)

timesofindia.com
Page 1
iitp.ac.in

$\text{toi.com} \longrightarrow \text{iitp.ac.in}$

This forms the webgraph link structure

To determine the importance of these pages, 3 known methods

1) Page Rank
2) Personalized Page Rank
3) PageRank based on RandomWalk with Restarts.

__Page Rank__

If page i with importance r_i has d_i out links, the each link gets $\frac{r_i}{d_i}$ share of the importance of node i

so if node j is a directed neighbor of node i, then

$$r_j = \sum_{i \to j} \frac{r_i}{d_i} = \sum_{\forall i \in N_{in}(j)} \frac{r_i}{d_i}$$

If ~~can~~ we consider the edges of node j ~~and~~ and let M be the matrix such that

$$M_{ij} = \frac{1}{d_j} \qquad M = \begin{array}{c} j \\ \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix} \end{array}$$

where d_j is the degree of node j

Let $d_j = 3$ then

and let 1, 3, & 4 be its neighb

$$M = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \end{array} \begin{bmatrix} \frac{1}{3} \\ 0 \\ \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

So the sum of the columns is 1

So M is a column stochastic matrix.

If r_i is the ~~rank vector~~ importance of node i and $\vec{r}$ is the importance vector. Let $\sum r_i = 1$

~~Then~~ then $r_i = \sum_{\forall j \to i} M_{ij} \cdot r_j$

$$\vec{r} = M \cdot \vec{r}$$

How is Page Rank implemented?

- Using Random walk

  Imagine a Random web surfer

  - At any time t, the surfer is at some page i
  - At time t+1, the surfer follows an out link from node i, uniformly
  - Ends up at page j linked to i
  - Process repeats itself.

- Let $\vec{p}(t)$ be a vector whose i^{th} element is the prob that the surfer is at page i at time t

  So $\vec{p}(t)$ is a distribution over the pages

  ~~Pol~~ Pick a random link and walk.

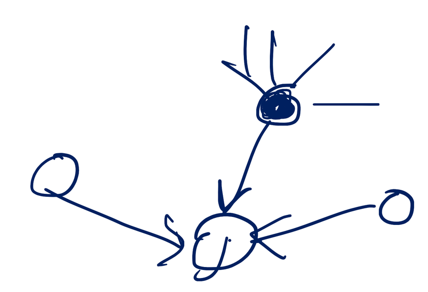
$$\vec{p}(t+1) = M \cdot \vec{p}(t)$$

If after some values of $t \to \infty$

$$\vec{p}(t+1) = M \cdot \vec{p}(t) = \vec{p}(t)$$

Steady State Condition

and $\vec{p}(t)$ is the vector with the page Rank.



Implementing Page Rank.

1. Given a web graph with N nodes where the nodes are the webpages and the links are the hyperlinks, then PageRank derives a rank vector $\vec{r}$

   a) Initialize $\vec{r}^{(0)} = \left[\frac{1}{N}, \frac{1}{N}, \frac{1}{N} \cdots \frac{1}{N}\right]^T$

   b) Iterate $\vec{r}^{(t+1)} = M \cdot \vec{r}^{(t)}$, i.e. $r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$

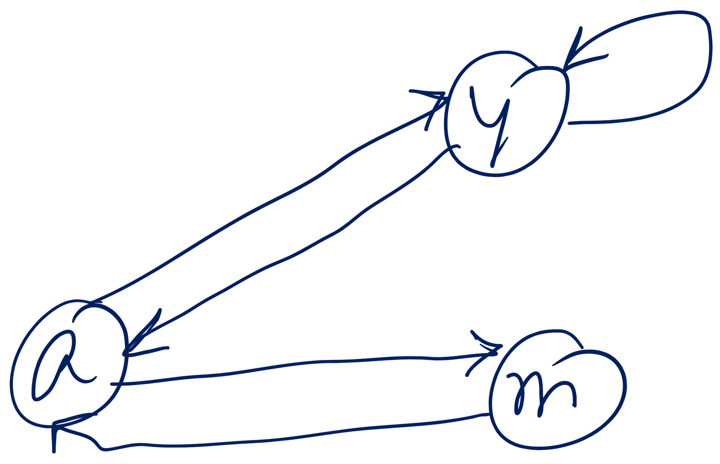
(c) Stop when $|\vec{r}^{(t+1)} - \vec{r}^{(t)}| < \epsilon$

This is L1 norm

~~It~~ One may use L_2 norm

$$|r_1^{(t+1)} - r_1^{(t)}| + |r_2^{(t+1)} - r_2^{(t)}| + \cdots |r_n^{(t+1)} - r_n^{(t)}|$$

E.g.



$$m = \begin{array}{c|ccc} & y & a & m \\ \hline y & 1/2 & 1/2 & 0 \\ a & 1/2 & 0 & 1 \\ m & 0 & 1/2 & 0 \end{array}$$

Set $\vec{r}^{(0)} = [1/3, 1/3, 1/3]^T$

Iterate $r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$

$$r_y^{(1)} = \frac{r_y^{(0)}}{2} + \frac{r_a^{(0)}}{2} = \frac{1}{3}$$

$$r_a^{(1)} = \frac{r_y^{(0)}}{2} + r_m^{(0)} = \frac{1}{2}$$

$$r_m^{(1)} = \frac{r_a^{(0)}}{2} = 1/6$$

Check If $|\vec{r}^{(1)} - \vec{r}^{(0)}| > \epsilon$ continue

$$\vec{r} = \begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} \to \begin{pmatrix} 1/3 \\ 1/2 \\ 1/6 \end{pmatrix} \to$$

Important Question

1) Does this converge?

2) Does it converge in a meaningful way?

Problems

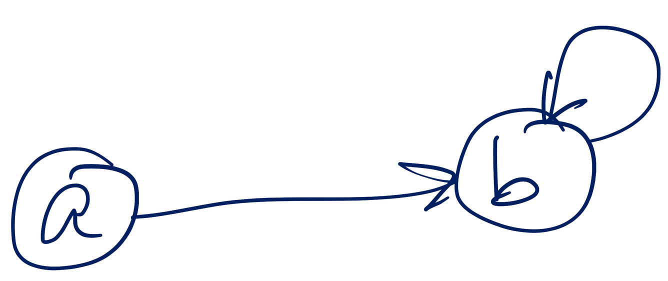
1) All pages may not have outlink (Dead End)

– ~~You~~ Once dead end is reached, importance of the nodes cannot be found out correctly.

2) A group of pages ~~for~~ forms a SPIDER TRAP, ie. all outgoing links are within the same group

– Nodes in the spider trap will have ~~all~~ high pagerank and the rest will have almost zero.

Spider trap problem



Iteration	0,	1,	2 .	3,	-
r_a	1	0	0	0	
r_b	0	1	1	1	

Dead End

Iteration	0	1	2	3	
r_a =	1	0	0	0	
r_b =	0	1	0	0	- - -

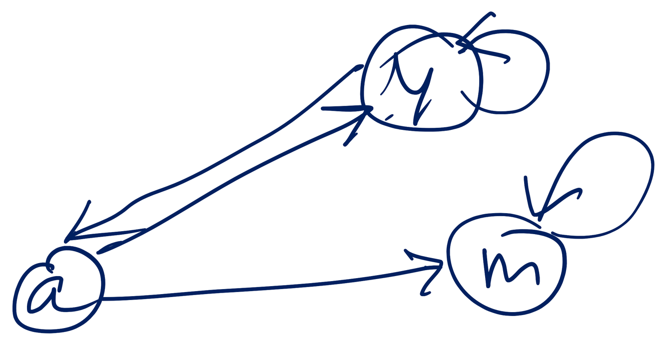
Solution

Teleportation

At each time step, the random surfer has 2 options

1) with prob β follow a random link
2) with prob $(1-\beta)$ jump to a random page

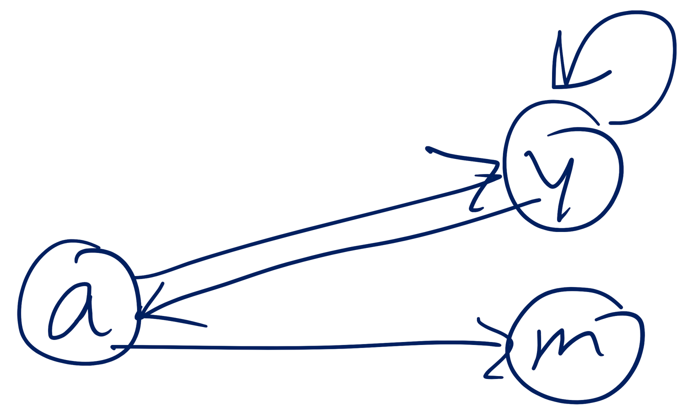
Keep the value of β high (0.8 or 0.9)



← spider trap

Teleportation helps to move out of the spider trap.

For dead Ends



$$M = \begin{array}{c|ccc} & y & a & m \\ \hline y & 1/2 & 1/2 & 0 \\ a & 1/2 & 0 & 0 \\ m & 0 & 1/2 & 0 \end{array}$$

mathematically deadends are a problem as All columns except the m^{th} column is column stochastic

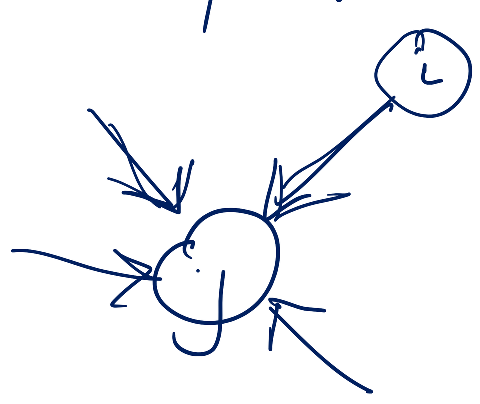
So make it column stochastic

$$M = \begin{array}{c|ccc} & y & a & m \\ \hline y & 1/2 & 1/2 & 1/3 \\ a & 1/2 & 0 & 1/3 \\ m & 0 & 1/2 & 1/3 \end{array}$$

← New matrix

Mathematically solution for random teleports.

- At Each step the random surfer has 2 option
  a) with prob β, follow a random outgoing link
  b) with prob $1-\beta$ teleport to a random page



Page Rank Equation

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

Gives some measure of the random walker to be at node j at a given time

Matrix formulation

$$\vec{r}^{(t+1)} = \beta M \cdot \vec{r}^{(t)} + (1-\beta)\left[\frac{1}{N}\right]_{N\times N} \vec{r}^{(t)}$$

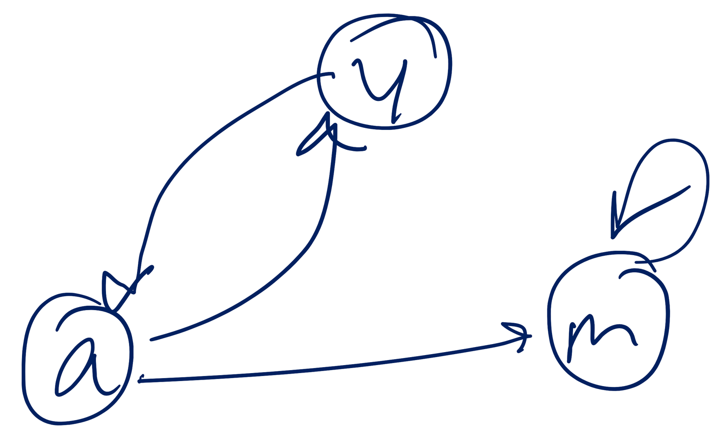
where $\left[\frac{1}{N}\right]_{N\times N}$ is ~~a matrix with~~ an $N\times N$ matrix with all values $1/N$.

$$\vec{r}^{(t+1)} = \left(\beta M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}\right)\vec{r}^{(t)}$$

$G \leftarrow$ Google matrix

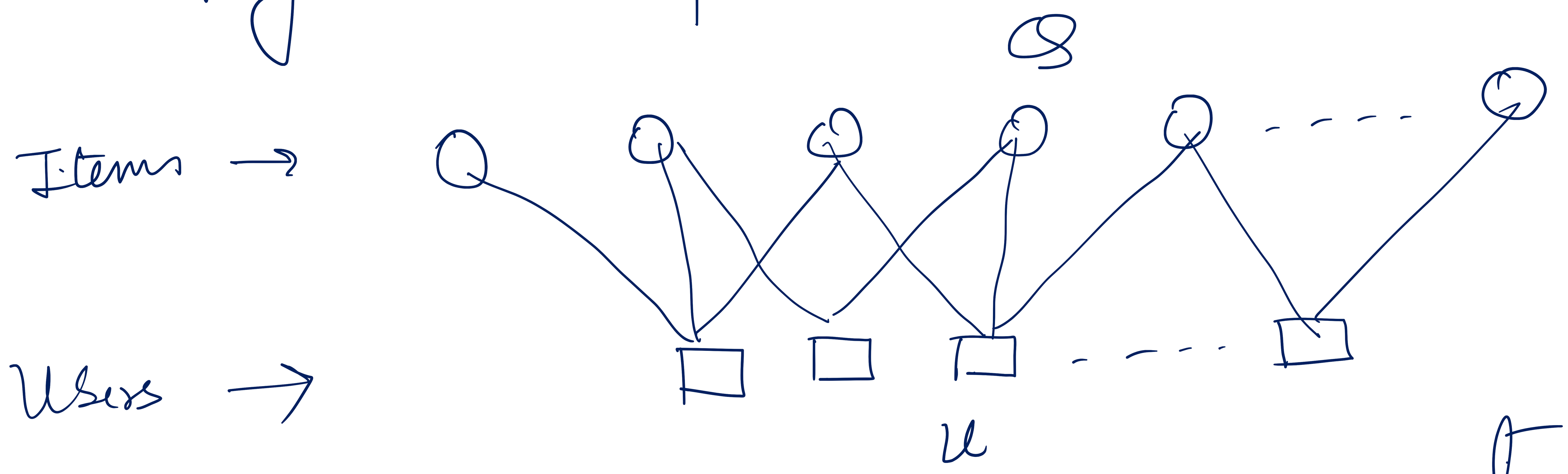
$$= G\,\vec{r}^{(t)} \longrightarrow \text{Solved by power iteration}$$

Solve by hand



Assume $\beta = 0.8$

PageRank for Recommendation



Given a bipartite graph representing user & item interactions

Goal – Which items should the system recommend to a user 'u' who has shown interest in a product 'Q'

(Find nodes proximity on graphs)

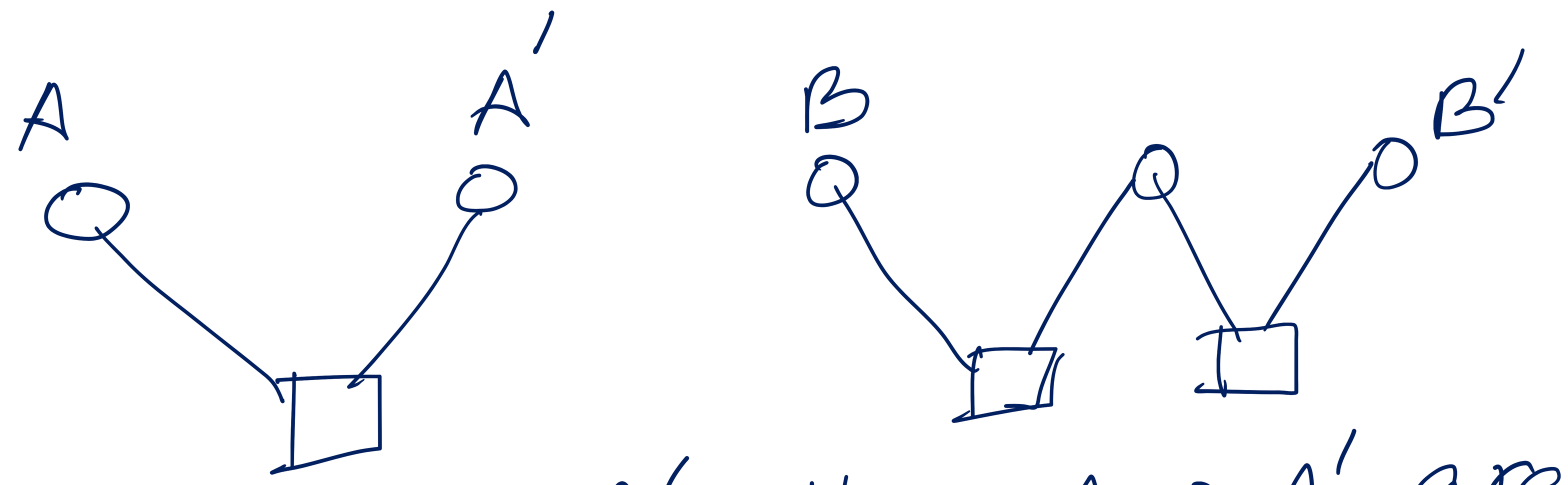
– If Q & P are interacted by similar users, then recommend P when users interact with Q.

The approach

Quantify proximity

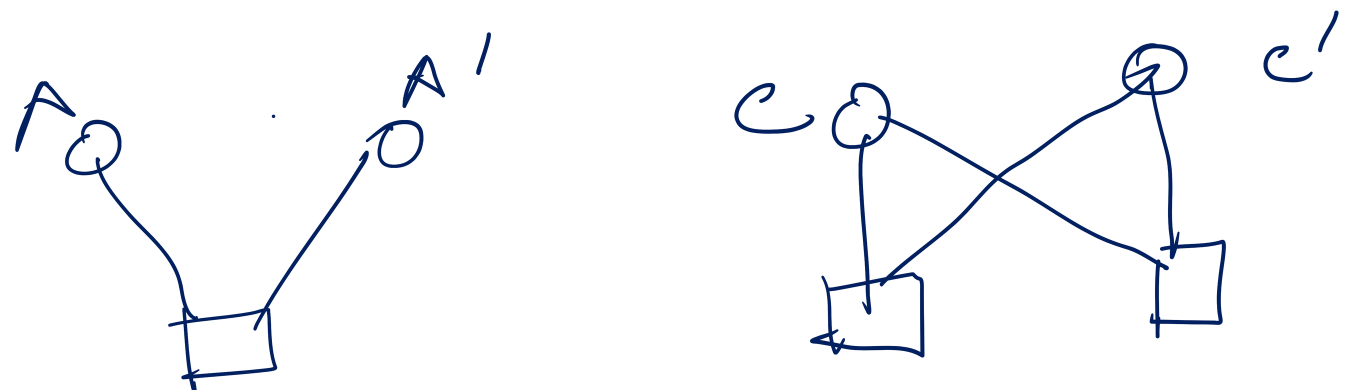
– Notions of proximity

1) Shortest path :–



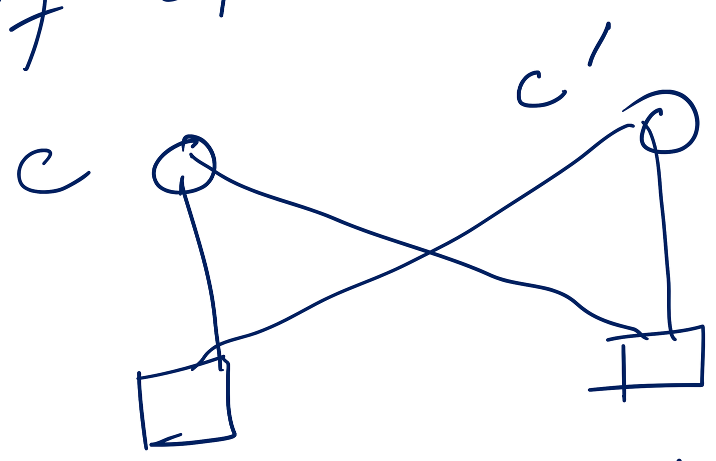
If A & A' have a shorter path than B & B', then A & A' are similar.

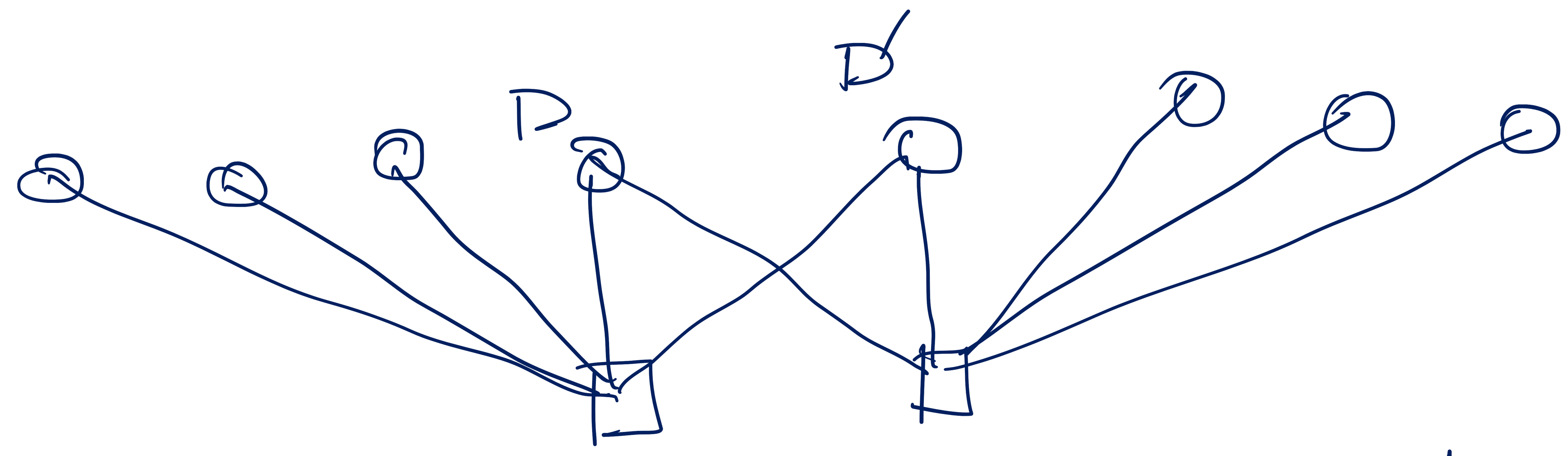
2) Common Neighbors :–



If C & C' have more common neighbors than A & A', then C & C' is more similar than A & A'

3) # of other interactions





Here D & D' is less similar than C & C' as the users have interacted with different items other than D & D'

Extension of PageRank

* PageRank
  - Ranked the nodes by their importance
  - Teleport to any node with uniform probability

* Personalized PageRank
  - Ranks the proximity of the nodes to a set of teleport nodes, S.
  - Answers, which is the most related item to item Q.
* PageRank with Restarts (Random Walk with Restarts)
  - Concept similar to that of personalized page rank
  - Considers only ~~on~~ the starting node as teleport node $S = \{Q\}$
  - Answers, which nodes are most closer to Q.

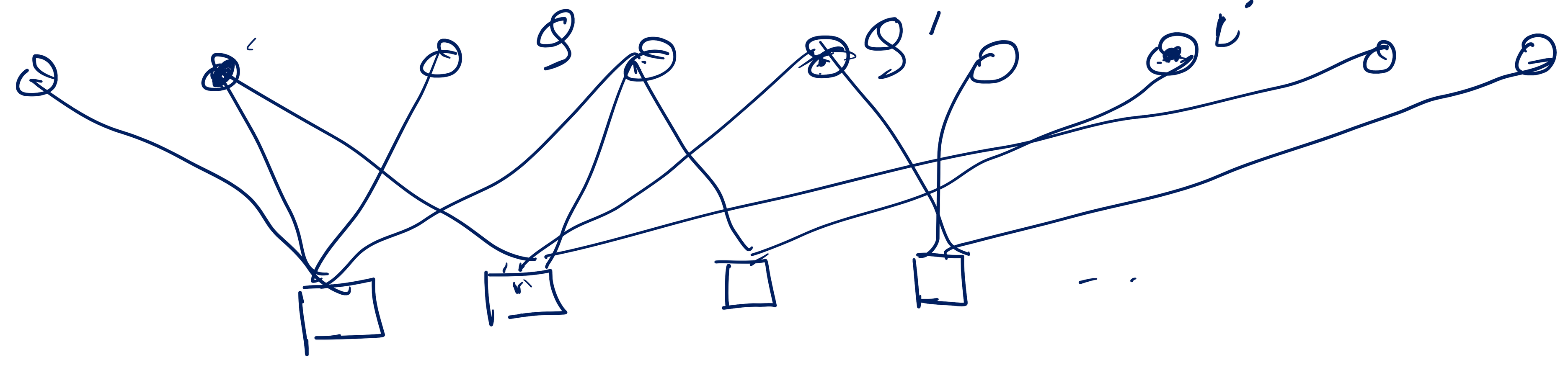
Personalized Page Rank Algorithm

Given a set of Query nodes, simulate a ~~a~~ random walk

a) make a step to a random neighbor and record the visit count.

b) with probability α, restart the walk at one of the Query nodes

c) The nodes with the highest visit count ~~with~~ will have the highest proximity to the query nodes.



Page Rank

Teleport to any node with uniform probability

$$\begin{array}{r|cccccccccc} \text{nodes} \rightarrow & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$

$$S_P = [0.1,\ 0.1\ \ 0.1\ \ 0.1\ \ 0.1\ \ 0.1\ \ 0.1\ \ 0.1\ \ 0.1\ \ 0.1]$$

for Personalized Page Rank $\quad S_{PPR}\ [0.1\ \ 0\ \ 0\ \ 0\ \ 0.2\ \ 0\ \ 0\ \ 0.4\ \ 0\ \ 0.3]$

for RW with Restart $\quad S_{RWR}\ [0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$

Hub & Authority Centrality

x_i = ~~Author~~ Authority Centrality of node i

y_i = Hub centrality of node i

$$x_i = \alpha \sum_j A_{ij}\, y_j \quad \& \quad y_i = \beta \sum_j A_{ji}\, x_j$$

$$\vec{X} = \alpha A \vec{Y} \qquad \vec{Y} = \beta A^T \vec{X}$$

or $\vec{X} = \cancel{\alpha A A}\ (\alpha\beta) A A^T \vec{X} \qquad \vec{Y} = \alpha\beta A^T A \vec{Y}$

or $(AA^T)\vec{X} = (\alpha\beta)^{-1}\vec{X} \quad \& \quad (A^TA)\vec{Y} = (\alpha\beta)^{-1}\vec{Y}$

So $\vec{X}$ is the Eigen Vector of (AA^T) with Eigen Value $(\alpha\beta)^{-1}$ | $\vec{Y}$ is the Eigen vector of (A^TA) with Eigen ~~value~~ value $(\alpha\beta)^{-1}$