



# CSCE 670 - Information Storage and Retrieval

## Lecture 5: Link Analysis (PageRank)

Yu Zhang

[yuzhang@tamu.edu](mailto:yuzhang@tamu.edu)

September 9, 2025





Course Website: <https://yuzhang-teaching.github.io/CSCE670-F25.html>

## Recap: BM25

$$\text{BM25}(q, d) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{\text{TF}(t, d) \cdot (k_1 + 1)}{\text{TF}(t, d) + k_1(1 - b + b \cdot \frac{|d|}{\text{avgdl}})}$$

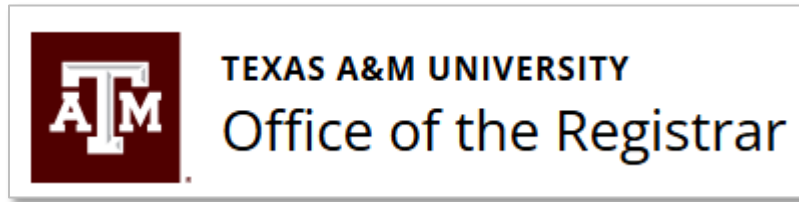
- $k_1$  controls term frequency scaling
  - $k_1 = 0$ : binary model
  - $k_1$  very large: raw term frequency
- $b$  controls document length normalization
  - $b = 0$ : no document length normalization
  - $b = 1$ : relative frequency (full document length normalization)
- Typically,  $k_1$  is set between 1.2 and 2;  $b$  is set around 0.75
- $|d|$  is the length of  $d$  (in words);  $\text{avgdl}$  = average document length (in words)

# Our Plan: Ranking

-  Why is ranking important?
-  What factors impact ranking?
- Two foundational text-based approaches
  -  TF-IDF
  -  BM25
- Two foundational link-based approaches
  - PageRank
  - HITS
- Machine-learned ranking (“learning to rank”)

## Recap: What factors impact ranking?

- Query: “TAMU 2025 Fall Break”
- Document 1: <https://registrar.tamu.edu/academic-calendar/fall-2025>



- Document 2: A social media post written by an account with 10 followers mentioning the time of TAMU 2025 Fall Break
- Document 1 should be ranked higher than Document 2 because it has a higher “reputation”.
  - But how can we know the “reputation” of a website?

# Web as a Directed Graph

- **Nodes:** Webpages

(Yu's Homepage)

*I am teaching  
CSCE 670 in Fall  
2025 ...*

(670 Webpage)

*CSCE 670 office  
hours are in the  
Peterson Building ...*

(CSE Webpage)

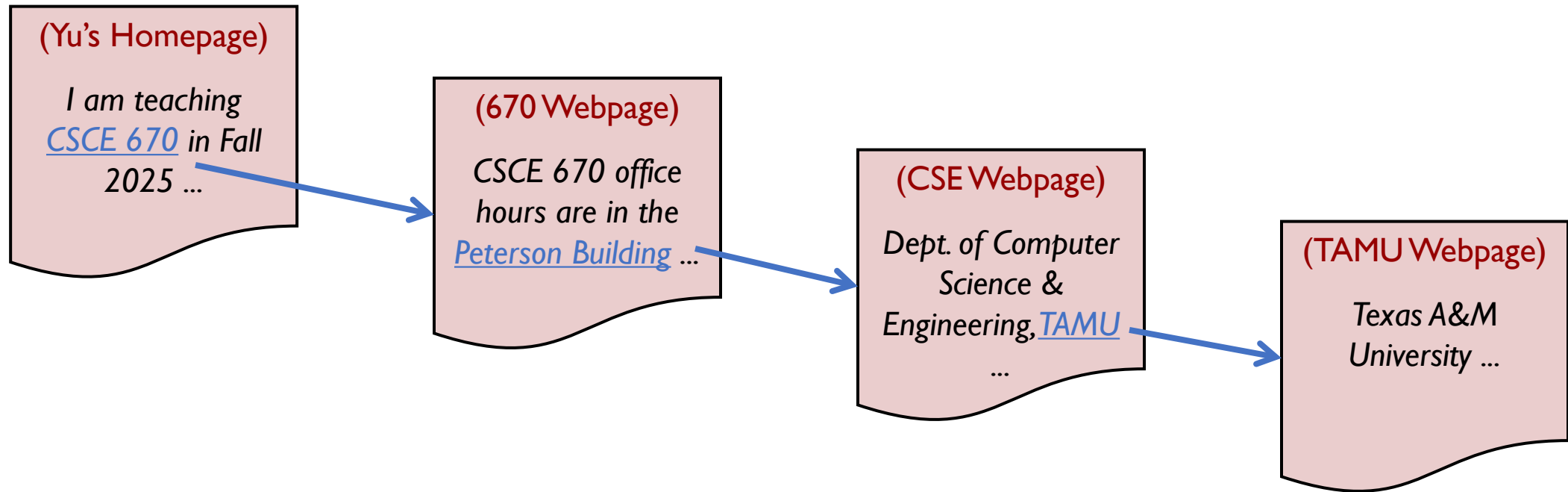
*Dept. of Computer  
Science &  
Engineering, TAMU  
...*

(TAMU Webpage)

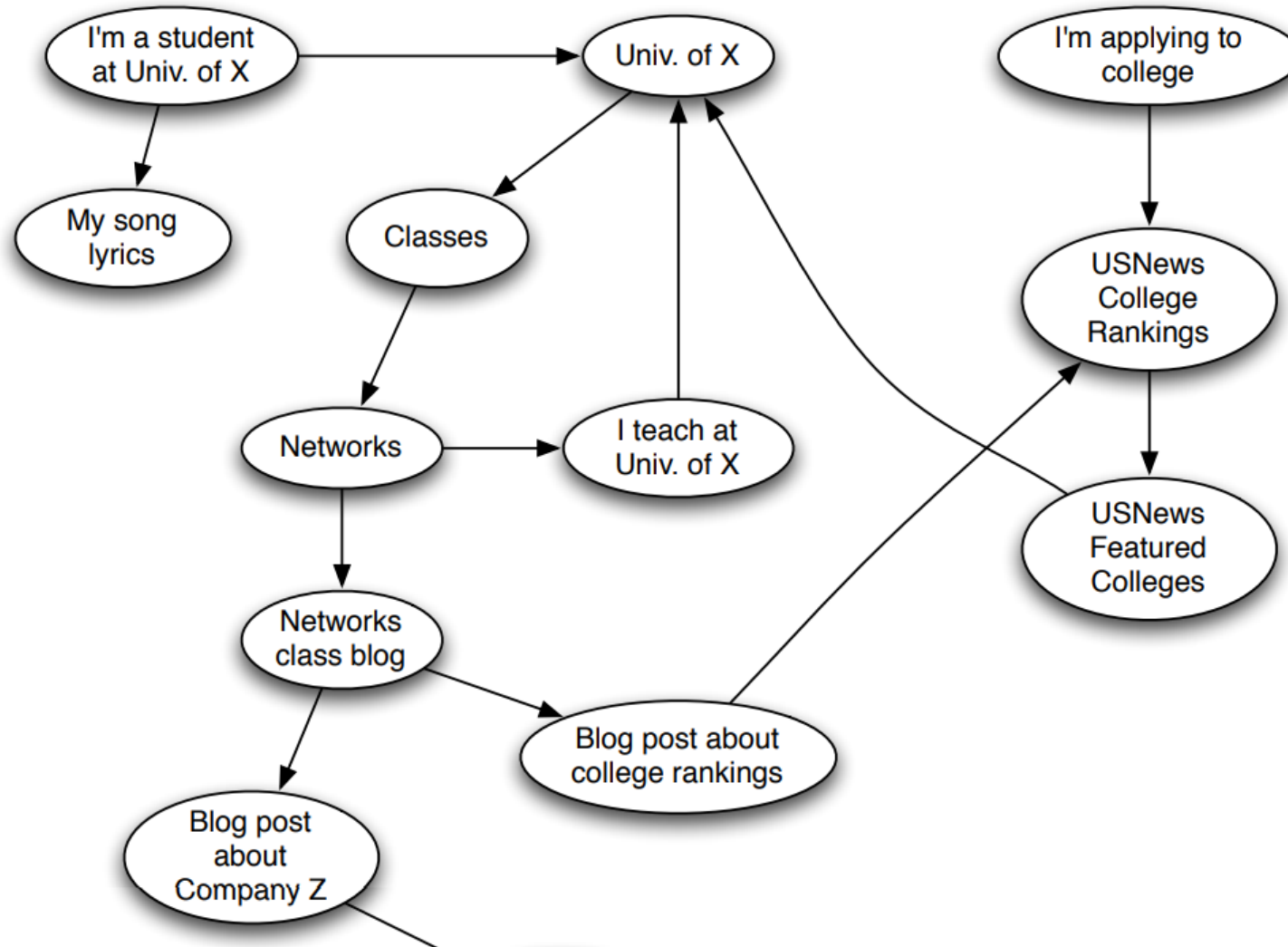
*Texas A&M  
University ...*

# Web as a Directed Graph

- **Nodes:** Webpages
- **Edges:** Hyperlinks



# Web as a Directed Graph

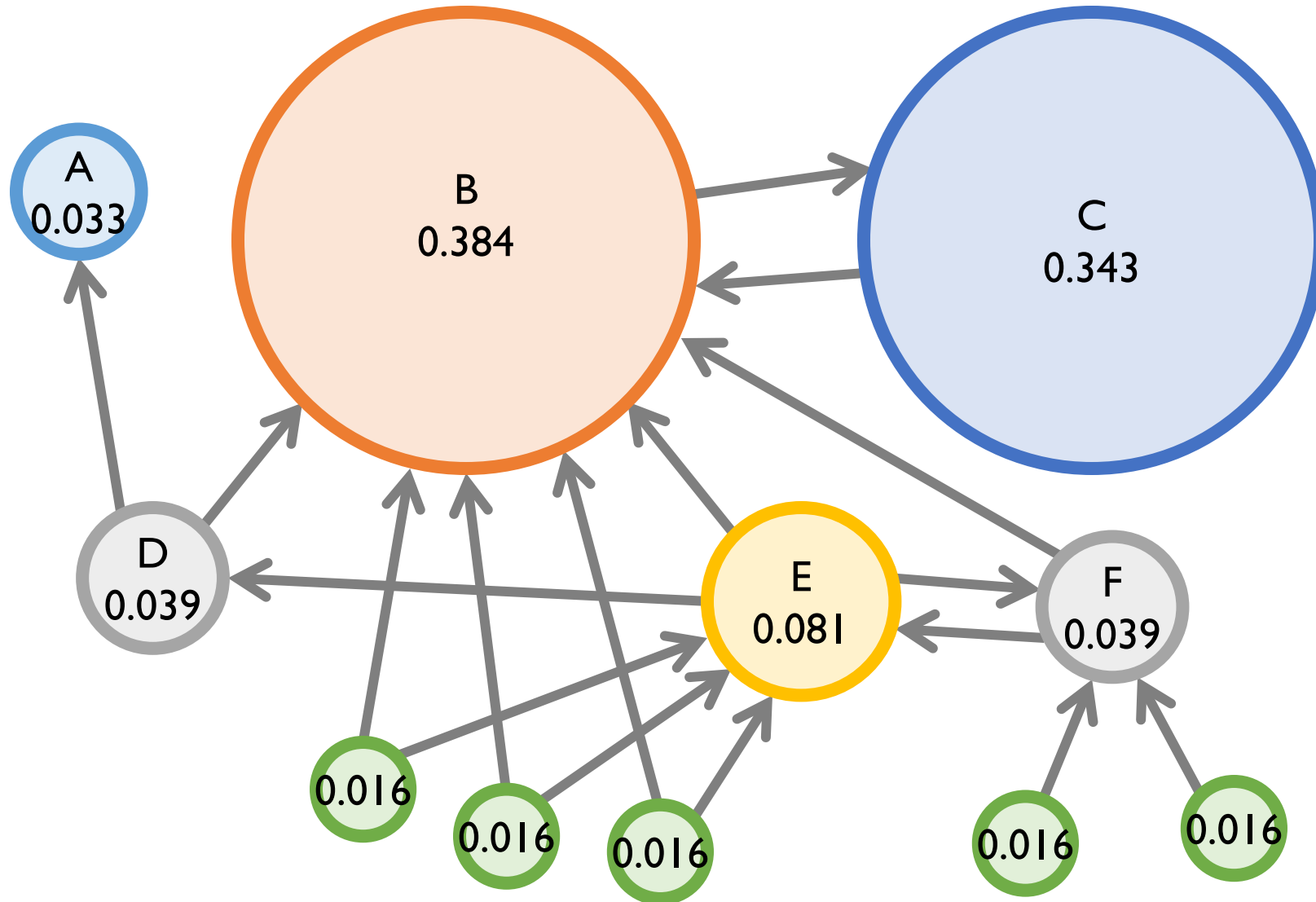


# Links as Votes

- **Rough Idea:** A webpage is more important if it has more links
  - In-coming links? Out-going links?
  - Out-going links can be easily manipulated by the webpage creator.
- Think of in-links as votes:
  - [www.stanford.edu](http://www.stanford.edu) has 23,400 in-links
  - [www.joe-schmoe.com](http://www.joe-schmoe.com) has 1 in-link
- Are all in-links equal?
  - Links from important webpages count more.
  - Recursive question!

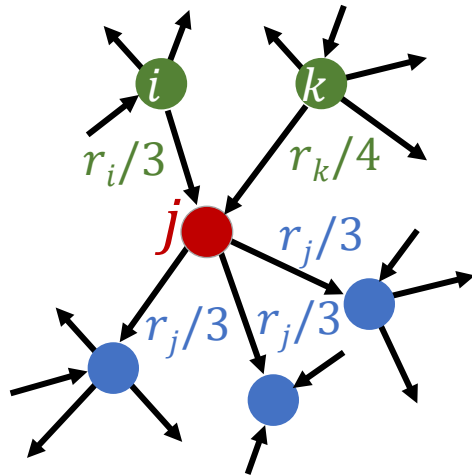


## Example: PageRank Scores



# Simple Recursive Formulation

- Each link's vote is proportional to the importance of its source page.
- If page  $j$  with importance  $r_j$  has  $n$  out-links, each link gets  $r_j/n$  votes
  - A vote from an important page is worth more.
- Page  $j$ 's own importance is the sum of the votes on its in-links.
  - A page is important if it is pointed to by other important pages



$$r_j = \frac{r_i}{3} + \frac{r_k}{4}$$

In general, 
$$r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$$

where  $d_i$  is the out-degree of  $i$

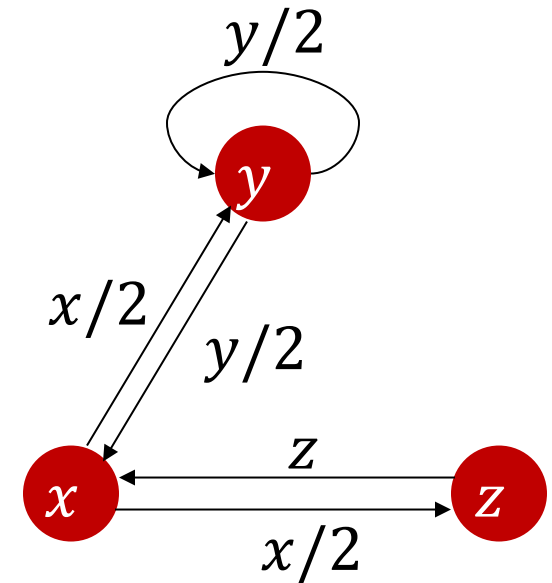
# Example

- $x = \frac{y}{2} + z$  (1)

- $y = \frac{y}{2} + \frac{x}{2}$  (2)

- $z = \frac{x}{2}$  (3)

- 3 equations, 3 unknowns. Looks like we can solve it!
- BUT if you add (1) and (2) together,
  - You will get (3).
  - Essentially, we have only 2 equations, so there exist infinitely many sets of solutions.
- Additional constraint forces uniqueness:
  - $x + y + z = 1$



# Example

- $x = \frac{y}{2} + z$  (1)

- $y = \frac{y}{2} + \frac{x}{2}$  (2)

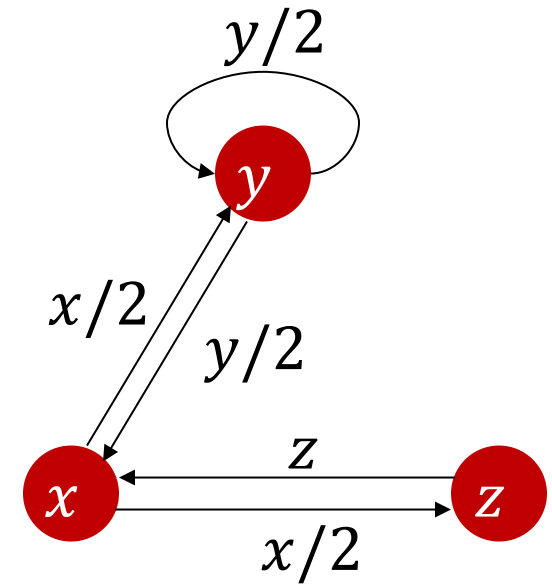
- $x + y + z = 1$  (3)

- Solution:

- $x = \frac{2}{5}, y = \frac{2}{5}, z = \frac{1}{5}.$

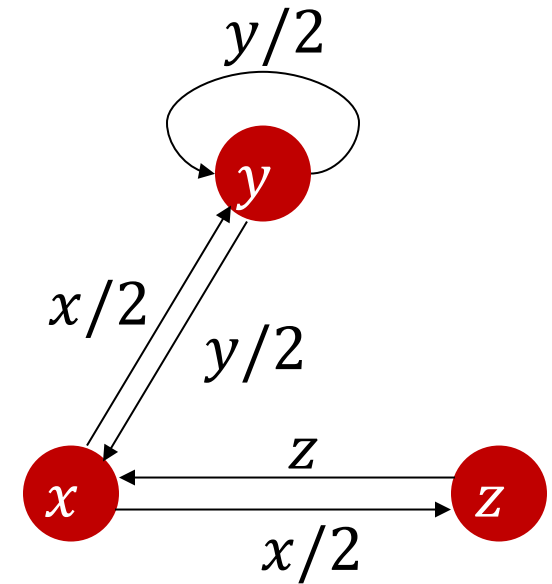
- Gaussian elimination method works for small examples, but we need a better method for large web-size graphs.

- We need a new formulation!



# PageRank: Matrix Formulation

- Stochastic adjacency matrix  $M$ 
  - Assume page  $i$  has  $d_i$  out-links
  - If  $i \rightarrow j$ , then  $M_{ji} = \frac{1}{d_i}$ , else  $M_{ji} = 0$ .
  - Entries in each column of  $M$  sum to 1
  - Example:  $M = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$

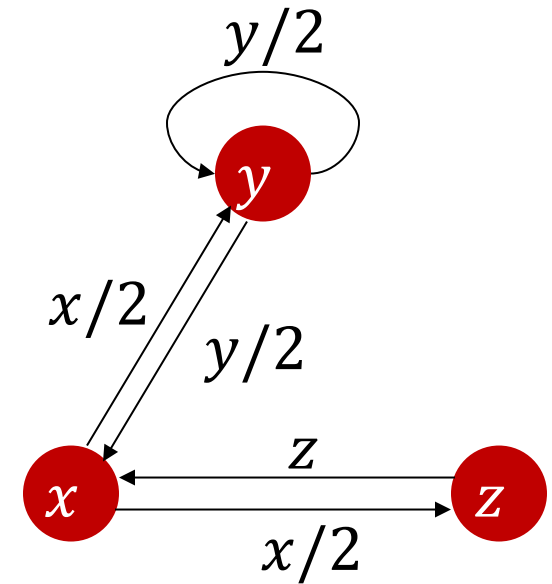


# PageRank: Matrix Formulation

- Rank vector  $\mathbf{r}$

- $r_i$  is the importance score of page  $i$
- Entries in  $\mathbf{r}$  sum to 1

- Example:  $\mathbf{r} = \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$



# PageRank: Matrix Formulation

- Equations:

- $r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$

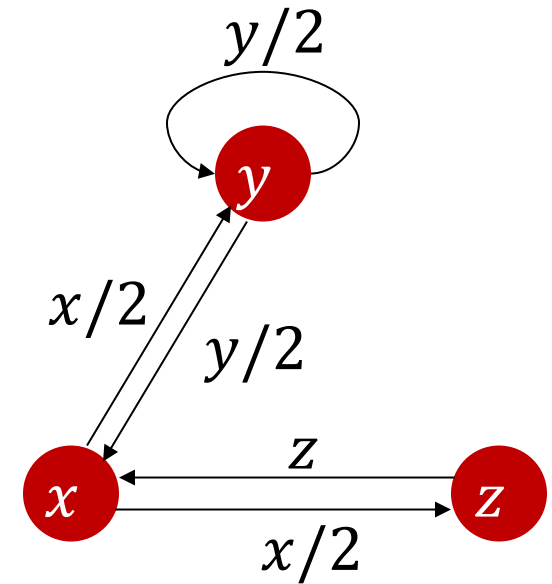
- Matrix form:  $\mathbf{M}\mathbf{r} = \mathbf{r}$

- Example: 
$$\begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix} = \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

- PageRank task:

- Given the stochastic adjacency matrix  $\mathbf{M}$ , we need to find a rank vector  $\mathbf{r}$  (whose entries sum to 1), so that

$$\mathbf{M}\mathbf{r} = \mathbf{r}$$



# Solving $\mathbf{M}\mathbf{r} = \mathbf{r}$ : Power Iteration Method

- *(Let's first assume this algorithm is correct. We will show why it works later.)*
- **Power Iteration**: a simple iterative scheme
  - Suppose there are  $N$  web pages in total
  - Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
  - Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
  - Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$  (a very small number, e.g., 0.001)
- If the algorithm stops, we have a good solution  $\mathbf{r}^{(t)}$ 
  - $\mathbf{M}\mathbf{r}^{(t)}$  is very close to  $\mathbf{r}^{(t)}$

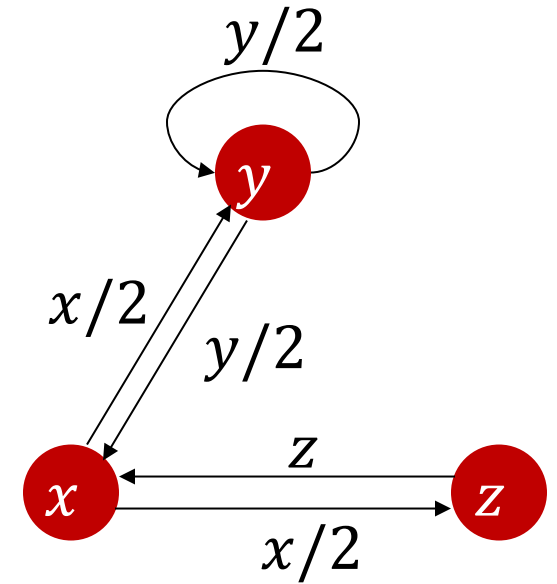


# Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$$



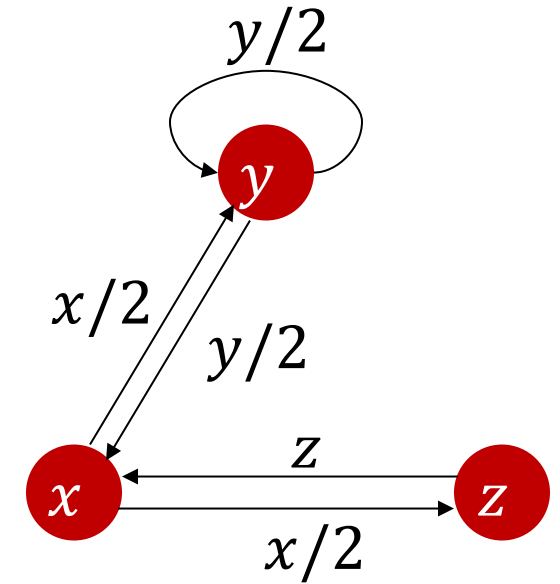
	$\mathbf{r}^{(0)}$
$x$	1/3 (0.33)
$y$	1/3 (0.33)
$z$	1/3 (0.33)

# Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$$



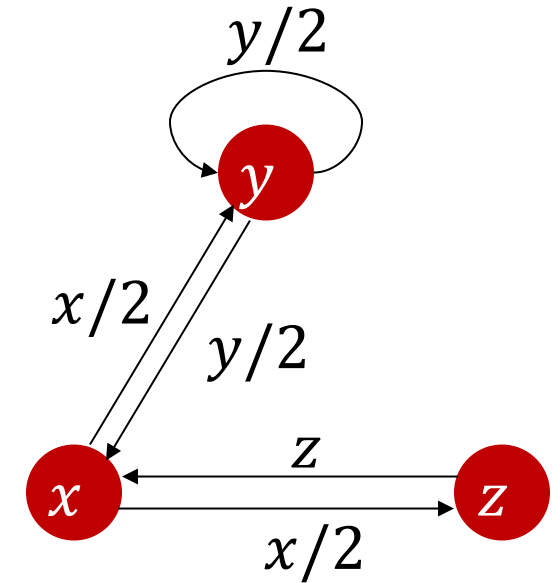
	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$
$x$	1/3 (0.33)	1/2 (0.50)
$y$	1/3 (0.33)	1/3 (0.33)
$z$	1/3 (0.33)	1/6 (0.17)

# Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$$



	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$	...	Finally
$x$	1/3 (0.33)	1/2 (0.50)	1/3 (0.33)	11/24 (0.46)	...	0.40
$y$	1/3 (0.33)	1/3 (0.33)	5/12 (0.42)	3/8 (0.38)	...	0.40
$z$	1/3 (0.33)	1/6 (0.17)	1/4 (0.25)	1/6 (0.17)	...	0.20

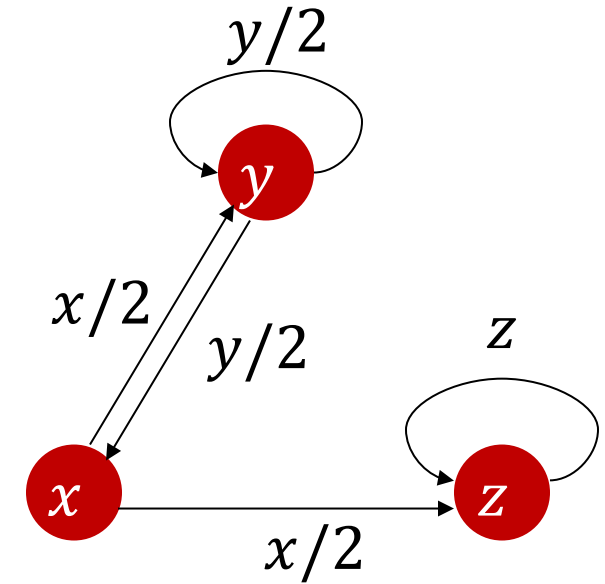
Questions?

# Another Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \end{bmatrix}$$



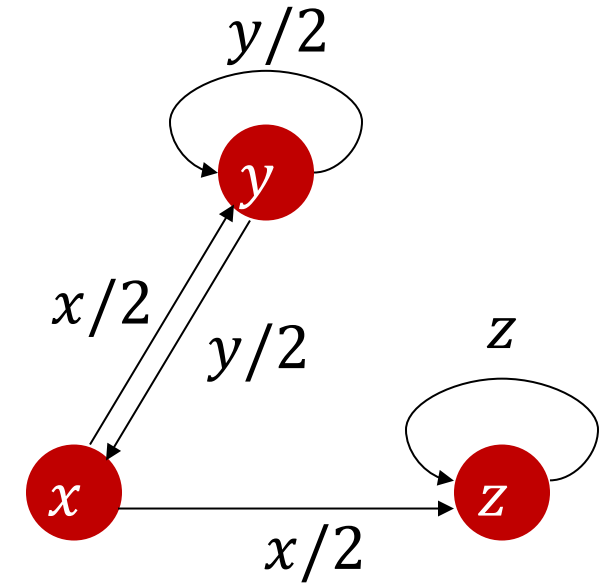
	$\mathbf{r}^{(0)}$
$x$	1/3 (0.33)
$y$	1/3 (0.33)
$z$	1/3 (0.33)

# Another Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \end{bmatrix}$$



All the PageRank scores get “trapped” in node z.

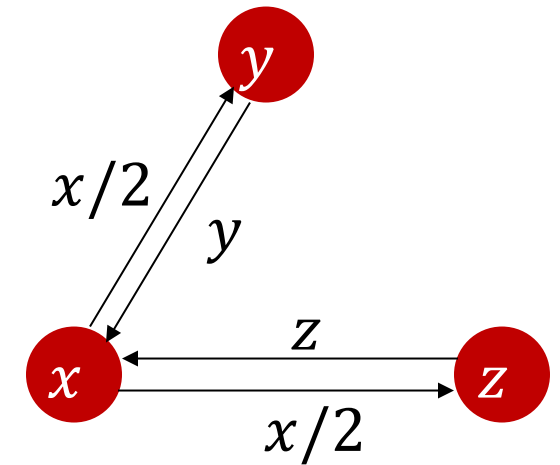
	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$	...	Finally
x	1/3 (0.33)	1/6 (0.17)	1/6 (0.17)	1/8 (0.13)	...	0.00
y	1/3 (0.33)	1/3 (0.33)	1/4 (0.25)	5/24 (0.21)	...	0.00
z	1/3 (0.33)	1/2 (0.50)	7/12 (0.58)	2/3 (0.67)	...	1.00

# An Even Worse Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 1 \\ 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$$



The algorithm falls into an infinite loop and will not terminate!

Root cause: the graph is bipartite.

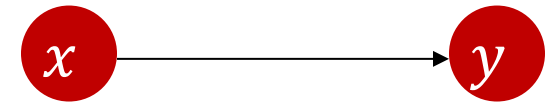
	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$	...	Finally
$x$	1/3	2/3	1/3	2/3	...	?
$y$	1/3	1/6	1/3	1/6	...	?
$z$	1/3	1/6	1/3	1/6	...	?

# Yet Another Even Worse Example

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate:  $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when  $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

$$\mathbf{M} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$



All the PageRank scores get “leaked”!

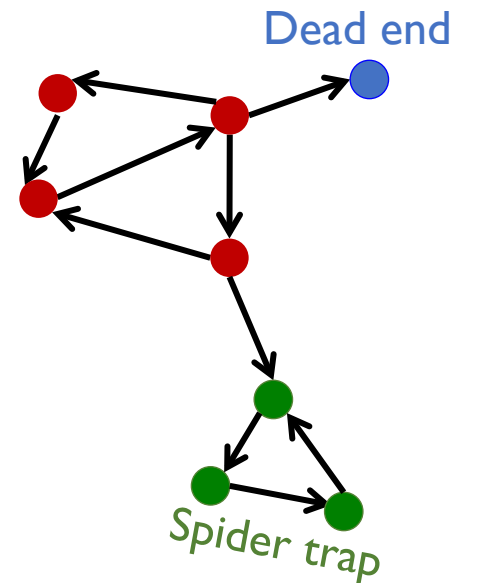
Root cause: the graph has a dead-end node (i.e., no out-links).

	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$
$x$	1/2	0	0	0
$y$	1/2	1/2	0	0



# Summary of the Challenges

- Spider traps
  - All out-links are within the group
  - Can have more than one node
  - Eventually spider traps absorb all importance
- Dead ends
  - The node has no out-links, therefore its importance score has nowhere to go
  - Eventually dead ends cause all importance to “leak out”
- Bipartite graph
  - If the graph is bipartite and the two partitions have different numbers of nodes, the algorithm will not converge.



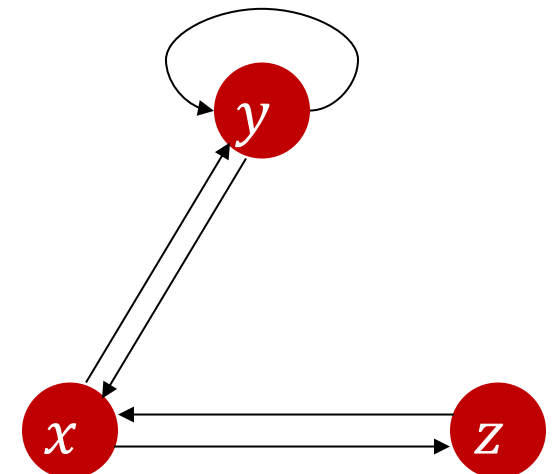
# PageRank: Google Formulation

- Google's solution for spider traps: **Teleportation!**
  - Each node must contribute a portion of its importance score and distribute it evenly to all other nodes.

- Without teleports,  $M = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$

- With teleports,  $M = \beta \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix} + (1 - \beta) \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$

- In practice,  $\beta = 0.8, 0.85, \text{ or } 0.9$

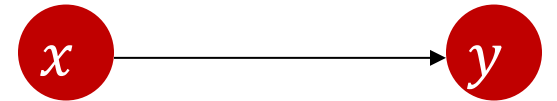


# How about dead ends?

- Dead ends must contribute **ALL** of its importance score and distribute it evenly to all other nodes.

- Without teleports,  $M = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$

- Without teleports,  $M = \beta \begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} + (1 - \beta) \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$



- Why do we call this solution “**teleportation**”?
  - Part of the importance score still flows according to the graph's defined neighborhoods
  - While the other part can instantly “**teleport**” to any node in the graph

# Why does teleportation solve the problems?

- **Spider traps**: with traps, PageRank scores are not what we want
  - Solution: Never get stuck in a spider trap by teleporting out of it
- **Dead ends**: the matrix  $M$  is no longer column-stochastic (entries in a column may sum to 0 rather than 1)
  - Solution: Make  $M$  column-stochastic by always teleporting when there is nowhere else to go
- Wait, how about the **bipartite-graph issue**?
  - Teleportation makes the graph fully-connected (with different edge weights) and naturally non-bipartite.

# PageRank: Google Formulation [Brin and Page, WWW 1998]

- Node-wise form:

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

- **Note 1:** Each node  $i$  in the graph teleports a score of  $(1 - \beta) \frac{1}{N} r_i$  to node  $j$ , so the total score node  $j$  receives through teleportation is exactly  $(1 - \beta) \frac{1}{N} \sum_i r_i = (1 - \beta) \frac{1}{N}$ .
- **Note 2:** This formulation assumes the graph has no dead ends. If there is a dead end, we can first link it to all the nodes (include itself).

# PageRank: Google Formulation [Brin and Page, WWW 1998]

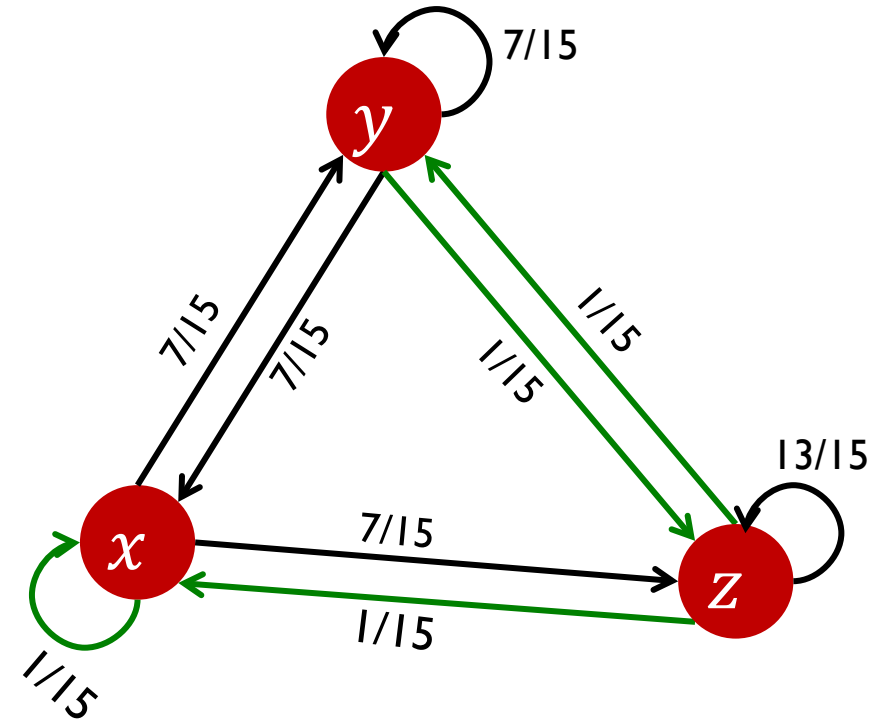
- Matrix form:

$$\mathbf{A} = \beta \mathbf{M} + (1 - \beta) \frac{\mathbf{1}}{N}$$

- **Note:**  $\mathbf{1}$  is an  $N \times N$  matrix where all entries are 1.
- Now we need to solve  $\mathbf{A}\mathbf{r} = \mathbf{r}$ 
  - We can still use Power Iteration

## Example ( $\beta = 0.8$ )

$$\begin{aligned}
 A &= 0.8 \times \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \end{bmatrix} + 0.2 \times \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \\
 &= \begin{bmatrix} 1/15 & 7/15 & 1/15 \\ 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 13/15 \end{bmatrix}
 \end{aligned}$$



	$r^{(0)}$	$r^{(1)}$	$r^{(2)}$	$r^{(3)}$	...	Finally
$x$	1/3	0.20	0.20	0.18	...	0.15
$y$	1/3	0.33	0.28	0.26	...	0.21
$z$	1/3	0.47	0.52	0.56	...	0.64

**Extended Content**  
(will not appear in quizzes or the exam)



# Why does Power Iteration work?

- $Ar = r$
- In other words,  $r$  is an **eigenvector** of  $A$  with the corresponding **eigenvalue**  $\lambda = 1$
- Why does  $A$  necessarily have an eigenvalue of 1?
- How about other eigenvalues of  $A$ ?
- **Perron–Frobenius Theorem**: Let  $A$  be a square matrix with all entries **strictly positive**, and entries in each column sum to 1, then
  - $A$  has an eigenvalue of 1
  - 1 is the **unique “largest”** eigenvalue of  $A$ . That is, for all other eigenvalues  $\lambda$  of  $A$ , we have  $|\lambda| < 1$ .

# Why does Power Iteration work?

- Power Iteration:

- Initialize:  $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$

- Iterate:  $\mathbf{r}^{(t+1)} = A\mathbf{r}^{(t)}$

$$\mathbf{r}^{(1)} = A\mathbf{r}^{(0)}$$

$$\mathbf{r}^{(2)} = A\mathbf{r}^{(1)} = A(A\mathbf{r}^{(0)}) = A^2\mathbf{r}^{(0)}$$

$$\mathbf{r}^{(3)} = A\mathbf{r}^{(2)} = A(A^2\mathbf{r}^{(0)}) = A^3\mathbf{r}^{(0)}$$

...

- We have a sequence of vectors  $A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, A^3\mathbf{r}^{(0)}, \dots$
- We need to prove that this sequence converges to the eigenvector of  $A$  with the eigenvalue  $\lambda = 1$

# Proof

- Let's assume  $A$  has eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_N$ , where  $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_N|$
- The eigenvectors corresponding to  $\lambda_1, \lambda_2, \dots, \lambda_N$  are  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ 
  - Let's also assume that  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  are linearly independent
  - If  $\lambda_1, \lambda_2, \dots, \lambda_N$  are different from each other, this assumption always holds.
- $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  form a basis, so we can write  $\mathbf{r}^{(0)} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \dots + c_N\mathbf{x}_N$
- $$\begin{aligned} A\mathbf{r}^{(0)} &= A(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \dots + c_N\mathbf{x}_N) \\ &= c_1A\mathbf{x}_1 + c_2A\mathbf{x}_2 + \dots + c_NA\mathbf{x}_N \\ &= c_1\lambda_1\mathbf{x}_1 + c_2\lambda_2\mathbf{x}_2 + \dots + c_N\lambda_N\mathbf{x}_N \end{aligned}$$
- Repeated multiplication on both sides
- $$A^2\mathbf{r}^{(0)} = c_1\lambda_1^2\mathbf{x}_1 + c_2\lambda_2^2\mathbf{x}_2 + \dots + c_N\lambda_N^2\mathbf{x}_N$$
- $$A^k\mathbf{r}^{(0)} = c_1\lambda_1^k\mathbf{x}_1 + c_2\lambda_2^k\mathbf{x}_2 + \dots + c_N\lambda_N^k\mathbf{x}_N$$

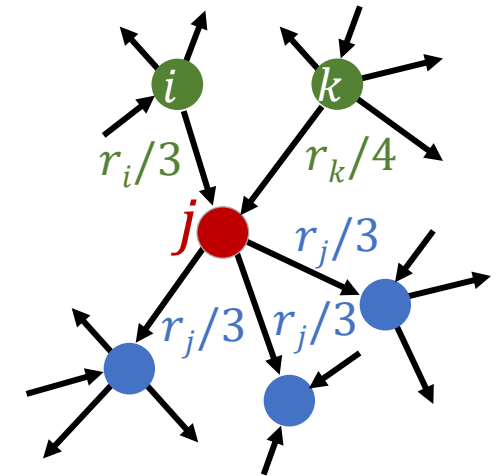
# Proof

- Let's assume  $A$  has eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_N$ , where  $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_N|$
- The eigenvectors corresponding to  $\lambda_1, \lambda_2, \dots, \lambda_N$  are  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$
- Repeated multiplication on both sides
- $$\begin{aligned} A^k \mathbf{r}^{(0)} &= c_1 \lambda_1^k \mathbf{x}_1 + c_2 \lambda_2^k \mathbf{x}_2 + \dots + c_N \lambda_N^k \mathbf{x}_N \\ &= \lambda_1^k \left( c_1 \mathbf{x}_1 + c_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{x}_2 + \dots + c_N \left( \frac{\lambda_N}{\lambda_1} \right)^k \mathbf{x}_N \right) \end{aligned}$$
- Note that  $\left| \left( \frac{\lambda_i}{\lambda_1} \right)^k \right| = \left| \frac{\lambda_i}{\lambda_1} \right|^k \rightarrow 0$  when  $k \rightarrow \infty$  (because  $|\lambda_i| < |\lambda_1|$ )
- Therefore,  $A^k \mathbf{r}^{(0)} \rightarrow \lambda_1^k (c_1 \mathbf{x}_1 + 0 + \dots + 0) = c_1 \mathbf{x}_1$ , which is the eigenvector of  $A$  with the eigenvalue  $\lambda_1 = 1$ .

Note: This proof does not apply to the case where  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  are NOT linearly independent, which may happen when  $A$  does not have  $N$  distinct eigenvalues.

# PageRank: Random Walk Interpretation

- Imagine there is a random web surfer
  - At time  $t$ , the surfer is on a page  $i$
  - At time  $t + 1$ , the surfer has two options
    - With probability  $\beta$ , it follows an out-link from  $i$  uniformly at random (i.e., ends up on some page  $j$  linked from  $i$ )
    - With probability  $1 - \beta$ , it jumps to a random page in the graph (can be  $i$ ,  $j$ , or any other node)
- The process repeats indefinitely
- Let  $\mathbf{p}(t)$  be the vector whose  $i$ -th coordinate is the probability that the surfer is at page  $i$  at time  $t$ 
  - So  $\mathbf{p}(t)$  is a probability distribution over pages



# The Stationary Distribution

- Where is the surfer at time  $t + 1$ ?

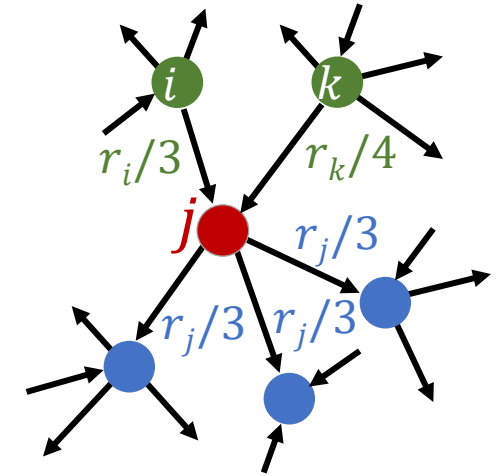
$$\mathbf{p}(t + 1) = \mathbf{A} \cdot \mathbf{p}(t)$$

- Suppose the random walk reaches a state

$$\mathbf{p}(t + 1) = \mathbf{A} \cdot \mathbf{p}(t) = \mathbf{p}(t)$$

then  $\mathbf{p}(t)$  is **stationary distribution** for the random walk

- The PageRank vector  $\mathbf{r}$  satisfies  $\mathbf{r} = \mathbf{A} \cdot \mathbf{r}$ 
  - So  $\mathbf{r}$  is a **stationary distribution** for the random walk



**A central result from the theory of random walks (Markov processes):**

For graphs that satisfy certain conditions (connected and non-bipartite), the **stationary distribution** is **unique** and **eventually will be reached** no matter what the initial probability distribution is at time  $t = 0$

# Back to the Broader Story of Ranking

Boolean + PageRank results for the query “university” [Page et al., 1999]

- With the rise of the Web, traditional **text-based signals** (e.g., TF-IDF and BM25) may not be sufficient.
- Many early web search engines relied on classic **text-based ranking** plus some rudimentary **link-based signals**.

The screenshot shows a web search engine interface with the query "university". The search bar at the top right contains "university" and a "Search" button. Below the search bar, there are tabs for "10 results", "clustering on", and "Search". The main content area displays a list of search results, each with a title, URL, and a score. The results are sorted by score, with the highest score at the top. The top result is "Optical Physics at the University of Oregon" with a score of 74.79%. Other results include "Stanford University Homepage", "Stanford University: Portfolio Collection", "University of Illinois at Urbana-Champaign", "Indiana University", "University of California, Irvine", "University of Minnesota", "Iowa State University Homepage", "The University of Michigan", "Mississippi State University", and "Northwestern University: NUInfo".

Query: university  
11 Results Returned  
Showing Results From 0 to 10

Stanford University Homepage  
http://www.stanford.edu/  
74.79% 4k - 2591993 - 010397

Stanford University: Portfolio Collection  
http://www.stanford.edu/home/administration/portfolio.html  
65.78% 3k - 2591993 - 010397

University of Illinois at Urbana-Champaign  
http://www.uiuc.edu/  
73.26% 13k - 129096 - 010397

Indiana University  
http://www.indiana.edu/  
68.38% 1k - 092396 - 010597

University of California, Irvine  
http://www.uci.edu/  
68.07% 3k - 129096 - 010397

University of Minnesota  
http://www.umn.edu/  
67.05% 0k - 129096 - 010397

Iowa State University Homepage  
http://www.iastate.edu/  
66.66% 3k - 129096 - 010397

The University of Michigan  
http://www.umich.edu/  
66.35% 1k - 2591993 - 010397

Mississippi State University  
http://www.msstate.edu/  
66.35% 3k - 2591993 - 010397

Northwestern University: NUInfo  
http://www.nyu.edu/  
66.15% 3k - 129096 - 010597

Optical Physics at the University of Oregon  
Oregon Center for Optics in Science and Technology. Department of Physics, University of Oregon, Eugene OR 97403. Research Groups: Carmichael Group....  
http://optics.uoregon.edu/ - size 1K - 16 Dec 96

Carnegie Mellon University - Campus Networking  
Departments. Data Communications. Data Communications is responsible for installing and maintaining all on campus networking equipment and all of...  
http://www.net.cmu.edu/ - size 4K - 19 Aug 95

Wesleyan University Computer Science Group Home Page  
Computer Science Group. Wesleyan University. Welcome to the home page of the Computer Science Group at Wesleyan University. We are administratively within.  
http://www.cs.wesleyan.edu/ - size 2K - 15 Apr 96

Keio University Shonan Fujisawa Campus (SFC)  
B\$3\$N%Z!EFnF#Bt%-%%s%Q%9 (B(SFC) \$B\$N (BWWW \$B% \$BcmOU=q\$- (B \$B\$FI\$s\$G\$/\$@5\$!\$# (B. Nihongo | English. SFC \$B>pJs (B. [ \$B%a%G%#% "%s%? !\*...  
http://www.sfc.keio.ac.jp/ - size 3K - 5 Feb 97

School of Chemistry, University of Sydney  
The School of Chemistry. School of Chemistry, University of Sydney, NSW 2006 Australia International Phone: +61-2-9351-4504 Fax: +61-2-9351-3329 Australia.  
http://www.chem.su.oz.au/ - size 4K - 25 Feb 97

Mankato State University  
The Campus Athletics, Campus Tour, Bookstore, Maps, Current Events... Admission & Registration Admissions, Financial Aid, Registrar's, Graduate...  
http://www.mankato.msut.edu/ - size 3K - 27 Nov 96

St. Ambrose University  
Main Index: Academic Departments. Administrative Services. Campus News. Computing Services. Galvin Fine Arts Center. Internet Connections. Library...  
http://www.sau.edu/ - size 2K - 4 Feb 97

University of Washington ECSEL Projects

# Back to the Broader Story of Ranking

- In practice, we will build a scoring function that considers many features.
- Typically, we have:
  - **Query-dependent features**: e.g., TF-IDF, BM25, # of times a query word occurs in a document, ...
  - **Query-independent features**: e.g., PageRank, # of in-links to a webpage, popularity of an album, ...
    - Many query-independent features are proxies for “reputation”
- **How to jointly consider these features?**
  - Week 5





Thank You!

Course Website: <https://yuzhang-teaching.github.io/CSCE670-F25.html>