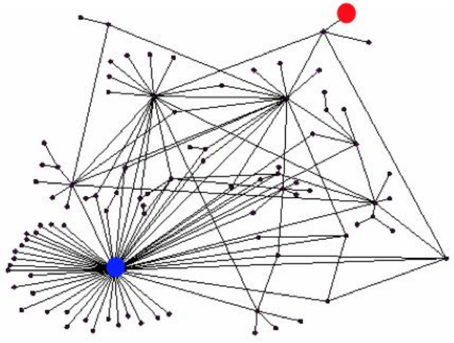


Link Analysis

Jay Urbain, Ph.D.
Electrical Engineering and Computer Science Department
Milwaukee School of Engineering

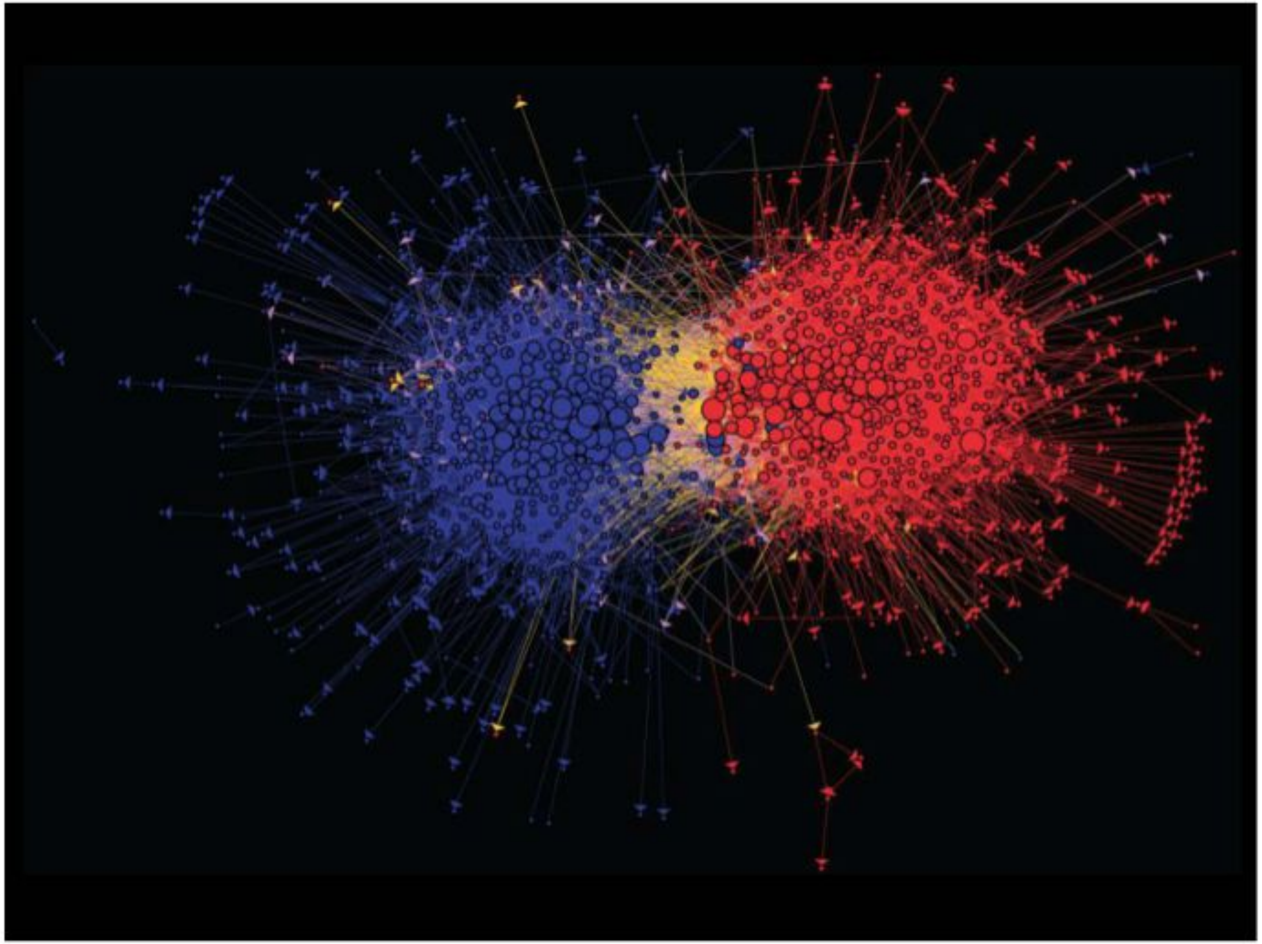
Credits:
Page and Brin, “The Anatomy of a Large-Scale Hypertextual Web Search Engine.“
Manning, “Introduction to Information Retrieval.”
J. Leskovec, A. Rajaraman, J. Ullman (Stanford University) “Mining of Massive Datasets.”

Link Analysis

- Analysis of **hyperlinks** and the **graph structure** of the Web has been instrumental in the development of Web search, social network analysis, and collaborative filtering.
- Primary factor for social network analysis.
- One of many factors considered by Web search engines in computing a composite score for a web page on any given query.

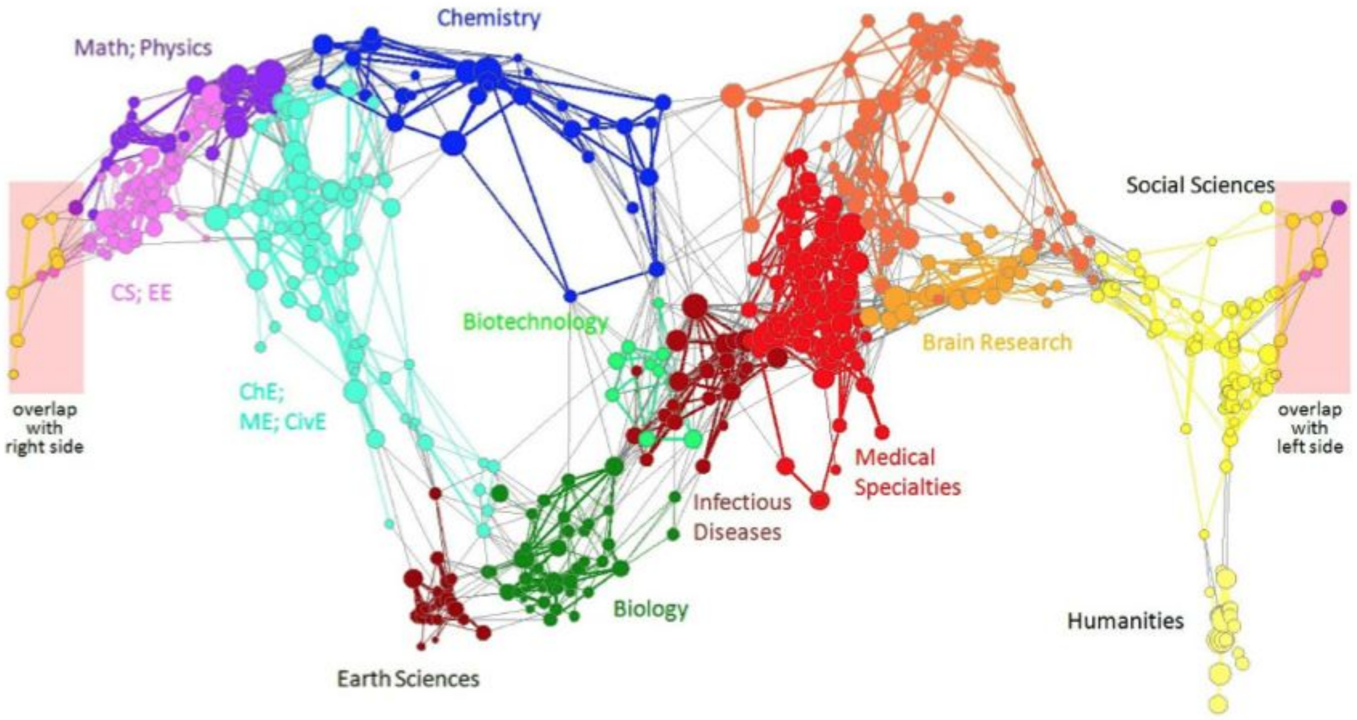
Facebook social graph

4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]



Connections between political blogs

Polarization of the network [Adamic-Glance, 2005]



Citation networks and Maps of science

[Börner et al., 2012]

Web Search: Challenges

1. **Web contains many sources of information. Who to "trust"?**
   - **Trick:** Trustworthy pages may point to each other!

2. **What is the "best" answer to the query: "newspaper"?**
   - No single right answer
   - **Trick:** Pages that actually know about newspapers might all be pointing to many newspapers

Ranking Nodes on the Graph

- **All web pages are *not* equally "important"**
  - http://catvideooftheweek.com/ vs. www.msoe.edu
- There is large diversity in the web-graph node connectivity.
- **Can we rank pages using link structure???**
- **Yes! Link Analysis Algorithms**
  - Page Rank
  - Hubs and Authorities (HITS - Hyperlink-Induced Topic Search)
  - Topic-Specific (Personalized) Page Rank
  - SimRank
  - Web Spam Detection Algorithms
  - Many other variants

Link Analysis for Web search

- Intellectual antecedents in **citation analysis** (*bibliometrics*).
- Seek to quantify the influence of scholarly articles by analyzing the **pattern of citations** among them.
- Much as ***citations*** represent the ***conferral authority*** from a scholarly article to others, link analysis on the Web treats ***hyperlinks*** from a Web page to another as a ***conferral authority***.

Problem:

- Every citation or hyperlink does not imply such authority, so measuring the quality of a web page requires other measurements as well.
  - Otherwise *link spam!*

Web as a Graph!

1. Anchor text pointing to a page *B* is a good description of page *B*.
2. The hyperlink from page *A* to page *B* represents an endorsement of page *B*, by the creator of page *A*.
   - Not always the case. Many links are from common templates, e.g., corporate web page referencing contact or copyright.

Informative hyperlink – target has same description as hyperlink

- Jounal of the ACM

Informative hyperlink – hyperlink has correct meaning, target may not – IBM home page may not even have the word computer, but may have words like "solutions"

- Big computer company

Non-informative hyperlink – link spam

- IBM

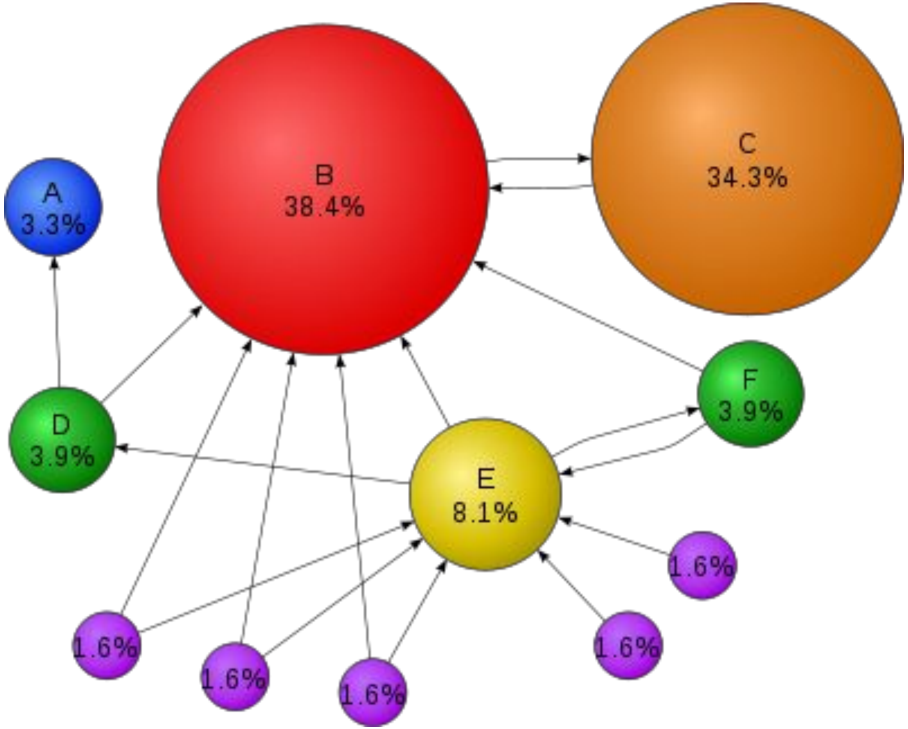
PageRank

- Assigns a weight between *0 and 1* to every node in the *web graph* relative to its "*importance*" within a hyperlinked set.
- Named after Larry *Page*.
- The *PageRank* of a node depends on the *link structure* of the web graph.
- Given a query, a search engine would combine *PageRank* with other similarity measurements to determine which Web pages are relevant.
- Assumes *random surfer* model.

PageRank

- The PageRank of a page is defined **recursively** (recurrence relation!), and depends on the *number* and *PageRank* metric of all pages that link to it ("**incoming links**").
- *A page that is linked to by many pages with high PageRank receives a high rank itself.*
- If there are no links to a web page there is no support for that page.
- Numerous academic papers concerning *PageRank* have been published since Page and Brin's original paper.
- In practice, the *PageRank* concept has proven to be vulnerable to manipulation.

PageRank



PageRank Algorithm

- PageRank is a ***probability distribution*** (random walk) representing the likelihood that a person randomly clicking on links will arrive at any particular page.
- Can be calculated for collections of documents of any size.
- Typically assumed that the distribution is evenly divided among all documents in the collection at the beginning of the computational process.
- PageRank computation:
  - Iterate: requires several passes through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.
- Solve:
  - Flow Model: Solve multiple simultaneous equations.
  - Matrix
- A probability is expressed as a numeric value between 0 and 1.
  - A 0.5 probability means there is a 50% chance that a person clicking on a random link will be directed to the document.

PageRank Algorithm

- Given a small universe of four web pages: **A**, **B**, **C** and **D**.
- The initial approximation of *PageRank* would be evenly divided between these four documents, i.e., begin with *PageRank* of 0.25.
- If pages **B**, **C**, and **D** each only link to **A**, they would each confer 0.25 PageRank to **A**.
- All PageRank **PR( )'s** in this system would confer a PageRank to **A** of 0.75

  PR(A) = PR(B) + PR(C) + PR(D)

PageRank Algorithm

- If page **B** has a link to page **C** as well as to page **A**, while page **D** has links to all three pages - the *value of the link-votes is divided among all the outbound links on a page*.
- Thus, page **B** gives a vote worth 0.125 (.250/2) to page **A** and a vote worth 0.125 to page **C**.
- Only one third of **D**'s PageRank is counted for A's PageRank (approximately 0.083 = .250/3).

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

PageRank Algorithm

- PageRank conferred by an outbound link is equal to the document's own PageRank score divided by the normalized number of outbound links **L(x)**.
- It is assumed that links to specific URLs only count once per document.

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

- More generally,

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$$

Problems:

- What if I just self-refer?
- What if I have a lapse in concentration and just jump to TikTok? I.e., not follow a hyperlink on my current MSOE Graph Machine Learning page?
- What if the page I'm on does not contain any outbound links? Am I stuck??? ;-)

Damping Factor

- Even an imaginary surfer who is randomly clicking on links will eventually stop clicking.
- The probability, at any step, that the person will continue is a damping factor ***d***.
- Various studies have tested different damping factors, and settled around 0.85. (PageRank values should sum to 1).

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

- Original :

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

Damping Factor

- A ***random surfer*** probably gets bored after several clicks and switches to a random page.
- The PageRank value of a page reflects the chance that the random surfer will land on that page by clicking on a link.
- Can be understood as a ***Markov chain*** in which the states are pages, and the transitions are all equally probable and are the links between pages.
- If a page has no links to other pages, it becomes a sink and therefore terminates the random surfing process.
- If the random surfer arrives at a sink page, it picks another ***URL*** at random and continues surfing again.

Calculating PR

- Can be computed iteratively or algebraically.

Iterative

- at t = 0, an initial probability distribution is assumed:

$$PR(p_i;0) = \frac{1}{N}$$

- At each time step, the computation:

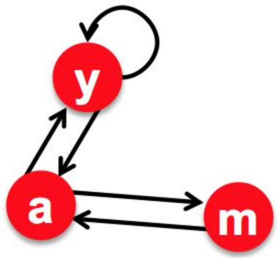
$$PR(p_i;t+1) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j;t)}{L(p_j)}$$

Matrix Formulation

- **Stochastic adjacency matrix**
- Let page *i* have d_i out-links
- If ***i* -> *j***, then $\boldsymbol{M_{ji}} = \frac{\mathbf{1}}{\boldsymbol{d_i}}$ else $\boldsymbol{M_{ji}} = \mathbf{0}$
- ***M*** is a **column stochastic matrix,** columns sum to **1**
- **Rank vector *r*:** vector with an entry per page $\sum_i \boldsymbol{r_i} = \mathbf{1}$
- $\boldsymbol{r_i}$ is the importance score of page $r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d_i}}$
- **The flow equations can be writter**

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

Flow Equations & Equivalent Matrix



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r = M \cdot r$$

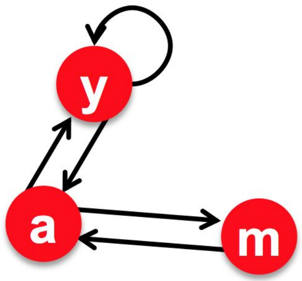
$$\begin{aligned} r_y &= r_y/2 + r_a/2 \\ r_a &= r_y/2 + r_m \\ r_m &= r_a/2 \end{aligned}$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

- **Power Iteration:**
  - Set $r_j = 1/N$
  - **1:** $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r = r'$
  - If not converged: goto **1**
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

Iteration 0, 1, 2,



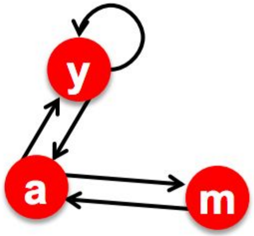
	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

- **Power Iteration:**
  - Set $r_j = 1/N$
  - **1:** $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r = r'$
  - If not converged: goto **1**



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

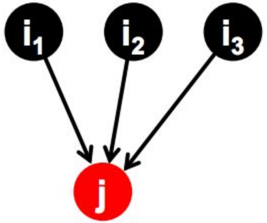
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{matrix}$$

Iteration 0, 1, 2,

Random Walk Interpretation

- **Imagine a random web surfer:**
  - At any time t, surfer is on some page i
  - At time $t+1$, the surfer follows an out-link from i uniformly at random
  - Ends up on some page j linked from i
  - Process repeats indefinitely
- **Let:**
  - $p(t)$ … vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
  - So, $p(t)$ is a probability distribution over pages

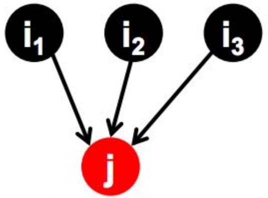


$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d_{out}(i)}}$$

Stationary Distribution

- **Where is the surfer at time *t+1*?**
  - Follows a link uniformly at random

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



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

- Suppose the random walk reaches a state

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

  then $\boldsymbol{p(t)}$ is **stationary distribution** of a random walk
- **Our original rank vector** $\boldsymbol{r}$ satisfies $\boldsymbol{r = M \cdot r}$
  - **So, r is a stationary distribution for the random walk**

Existence and Uniqueness

- **A central result from the theory of random walks (a.k.a. Markov processes):**

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

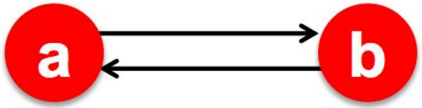
Page Rank Questions

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_\mathrm{i}}$$ or equivalently $$r = Mr$$

- **Does this converge?**
- **Does it converge to what we want?**
- **Are results reasonable?**

Does this converge?

- **The "Spider trap" problem:**



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_i}$$

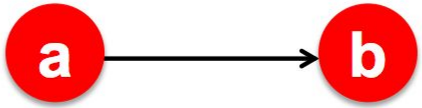
- **Example:**

$$\begin{array}{c} r_a \\ r_b \end{array} = \begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}$$

Iteration 0, 1, 2,

Does this converge?

- **The "Dead end" problem:**



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_i}$$

- **Example:**

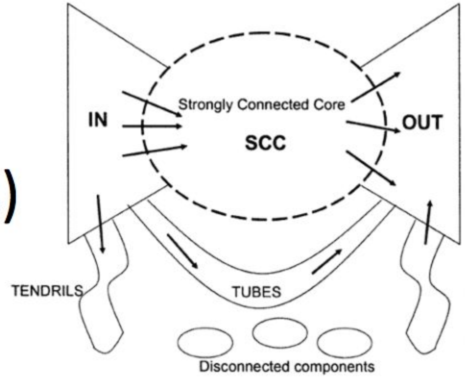
r_a	=	1	0	0	0
r_b		0	1	0	0

Iteration 0, 1, 2,

Does this converge?

<u>**2 problems:**</u>

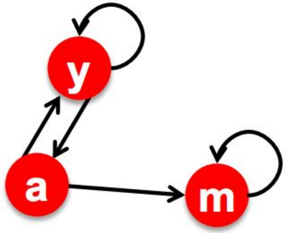
- **(1)** Some pages are **dead ends** (have no out-links)
  - Such pages cause importance to "leak out"



- **(2)** **Spider traps** (all out-links are within the group)
  - Eventually spider traps absorb all importance

Problem: Spider Traps

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2$$
$$r_m = r_a/2 + r_m$$

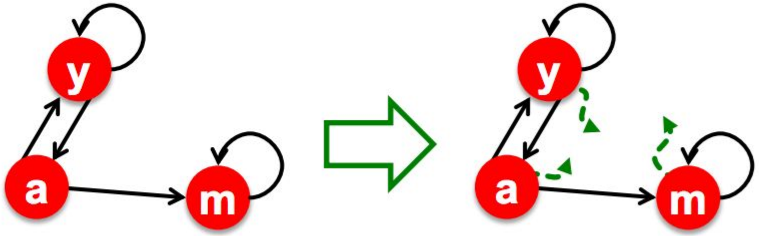
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

Iteration 0, 1, 2,

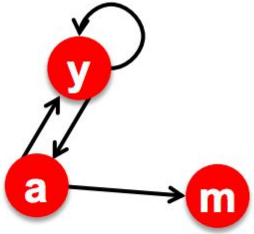
Solution: Random Teleports

- **The Google solution for spider traps: At each time step, the random surfer has two options**
  - With prob. β, follow a link at random
  - With prob. **1-**β, jump to some random page
  - Common values for β are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap within a few time steps**



Problem: Dead Ends

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2$

$r_m = r_a/2$

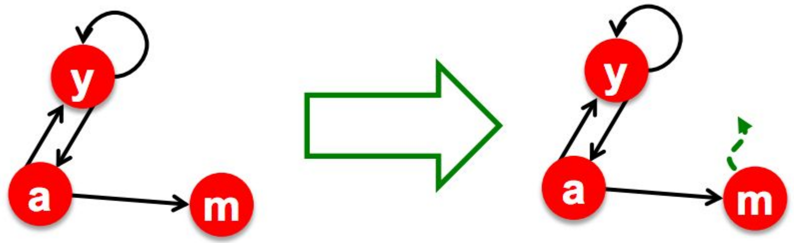
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{matrix}$$

Iteration 0, 1, 2,

Solution: Always Teleport

- **Teleports:** Follow random teleport links with probability **1.0** from dead-ends
  - Adjust matrix accordingly



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

- **Theory of Markov chains**

- **Fact:** For **any start vector**, the power method applied to a Markov transition matrix ***P*** will **converge** to a **unique** positive stationary vector as long as ***P*** is **stochastic**, **irreducible** and **aperiodic**.