

# Link Analysis: TrustRank and WebSpam

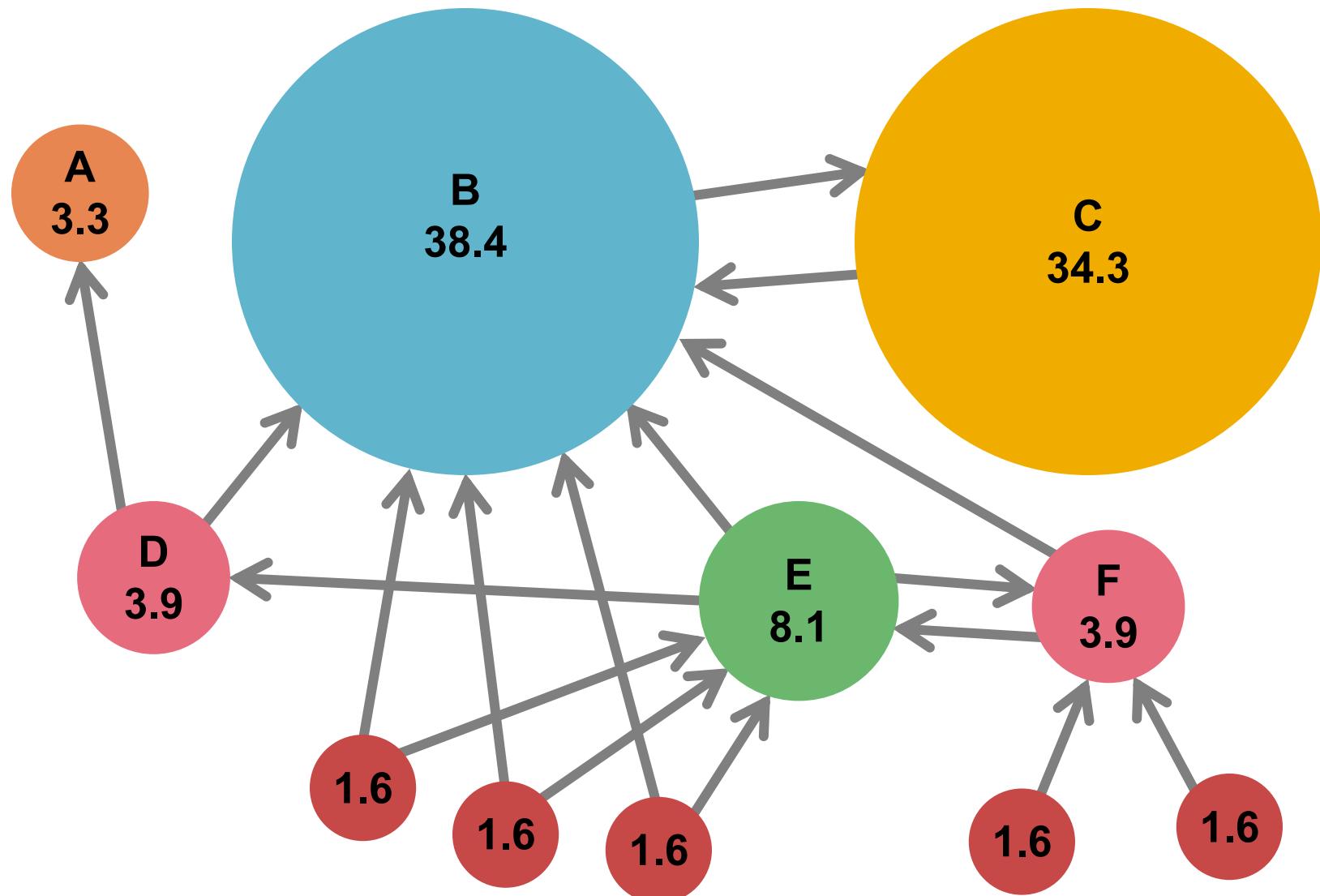
CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

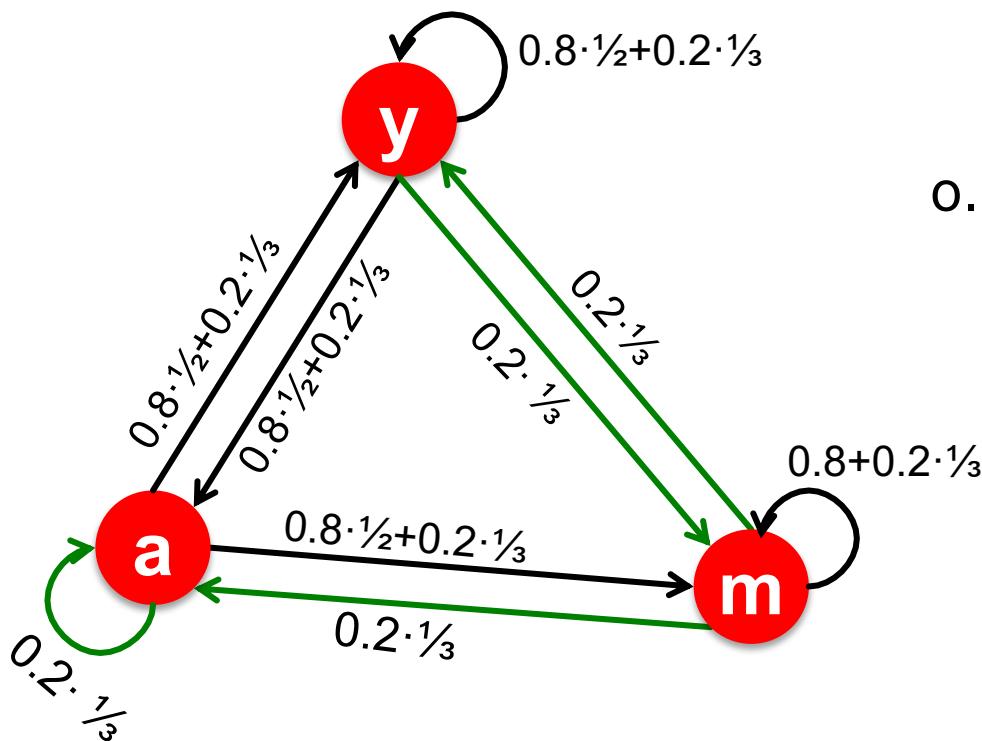
<http://cs246.stanford.edu>



# Example: PageRank Scores



# Random Teleports ( $\beta = 0.8$ )



$$\begin{array}{c}
 M \\
 \left[ \begin{array}{ccc} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{array} \right] \\
 + 0.2 \left[ \begin{array}{ccc} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{array} \right] \\
 [1/N]_{NxN}
 \end{array}$$

$$A \\
 \left[ \begin{array}{ccc} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{array} \right]$$

$$\begin{array}{llllll}
 y & 1/3 & 0.33 & 0.24 & 0.26 & 7/33 \\
 a = & 1/3 & 0.20 & 0.20 & 0.18 & \dots & 5/33 \\
 m & 1/3 & 0.46 & 0.52 & 0.56 & & 21/33 \\
 r = & Ar & & & & &
 \end{array}$$

# PageRank: The Complete Algorithm

- Input: Graph  $G$  and parameter  $\beta$ 
  - Directed graph  $G$  (can have spider traps and dead ends)
  - Parameter  $\beta$
- Output: PageRank vector  $r$

- **Set:**  $r_j^{(0)} = \frac{1}{N}, t = 1$
- **Do:**  $\forall j: r'_j = \sum_{i \rightarrow j} \beta \frac{r_i^{(t-1)}}{d_i}$   
 $r'_j = 0$  if in-degree of  $j$  is 0
- Now re-insert the leaked PageRank:  
 $\forall j: r_j^{(t)} = r'_j + \frac{1-S}{N}$  where:  $S = \sum_j r'_j$
- $t = t + 1$
- **while**  $\sum_j |r_j^{(t)} - r_j^{(t-1)}| < \varepsilon$

If the graph has no dead-ends then the amount of leaked PageRank is  $1-\beta$ . But since we have dead-ends the amount of leaked PageRank may be larger. We have to explicitly account for it by computing  $S$ .

# Some Problems with PageRank

- **Measures generic popularity of a page**
  - Will ignore/miss topic-specific authorities
  - **Solution:** Topic-Specific PageRank (**next**)
- **Uses a single measure of importance**
  - Other models of importance
  - **Solution:** Hubs-and-Authorities
- **Susceptible to Link spam**
  - Artificial link topographies created in order to boost page rank
  - **Solution:** TrustRank

# Topic-Specific PageRank

# Topic-Specific PageRank

- Instead of generic popularity, can we measure popularity within a topic?
- Goal: Evaluate Web pages not just according to their popularity, but also by how close they are to a particular topic, e.g. “sports” or “history”
- Allows search queries to be answered based on interests of the user
  - Example: Query “Trojan” wants different pages depending on whether you are interested in sports, history, or computer security

# Topic-Specific PageRank

- Random walker has a small probability of teleporting at any step
- **Teleport can go to:**
  - **Standard PageRank:** Any page with equal probability
    - To avoid dead-end and spider-trap problems
  - **Topic Specific PageRank:** A topic-specific set of “relevant” pages (**teleport set**)
- **Idea: Bias the random walk**
  - When the walker teleports, she picks a page from a set  $S$
  - $S$  contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic/query
  - For each teleport set  $S$ , we get a different vector  $r_s$

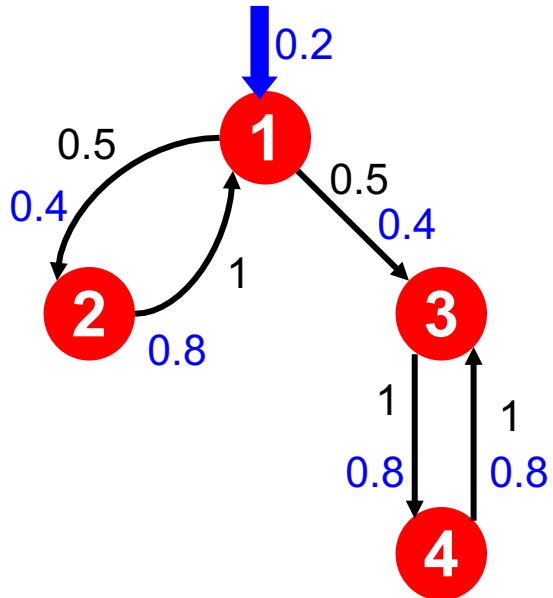
# Matrix Formulation

- To make this work all we need is to update the teleportation part of the PageRank formulation:

$$A_{ij} = \begin{cases} \beta M_{ij} + (1 - \beta)/|S| & \text{if } i \in S \\ \beta M_{ij} + 0 & \text{otherwise} \end{cases}$$

- $A$  is a stochastic matrix!
- We weighted all pages in the teleport set  $S$  equally
  - Could also assign different weights to pages!
- Compute as for regular PageRank:
  - Multiply by  $M$ , then add a vector
  - Maintains sparseness

# Example: Topic-Specific PageRank



Suppose  $S = \{1\}$ ,  $\beta = 0.8$

Node	Iteration	0	1	2	...	stable
1	0	0.25	0.4	0.28		0.294
2	0	0.25	0.1	0.16		0.118
3	0	0.25	0.3	0.32		0.327
4	0	0.25	0.2	0.24		0.261

$S = \{1, 2, 3, 4\}$ ,  $\beta = 0.8$ :

$r = [0.13, 0.10, 0.39, 0.36]$

$S = \{1\}$ ,  $\beta = 0.9$ :

$r = [0.17, 0.07, 0.40, 0.36]$

$S = \{1\}$ ,  $\beta = 0.8$ :

$r = [0.29, 0.11, 0.32, 0.26]$

$S = \{1\}$ ,  $\beta = 0.7$ :

$r = [0.39, 0.14, 0.27, 0.19]$

$S = \{1, 2, 3\}$ ,  $\beta = 0.8$ :

$r = [0.17, 0.13, 0.38, 0.30]$

$S = \{1, 2\}$ ,  $\beta = 0.8$ :

$r = [0.26, 0.20, 0.29, 0.23]$

$S = \{1\}$ ,  $\beta = 0.8$ :

$r = [0.29, 0.11, 0.32, 0.26]$

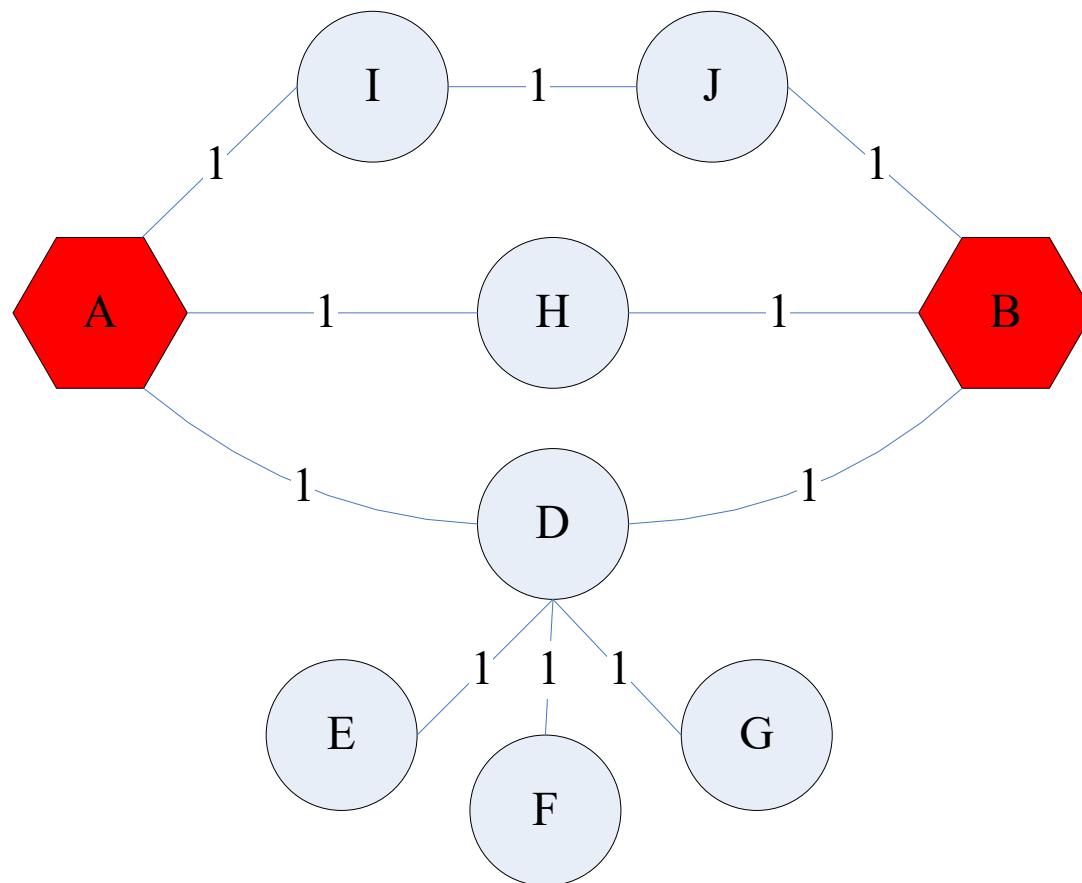
# Discovering the Topic Vector S

- **Create different PageRanks for different topics**
  - The 16 DMOZ top-level categories:
    - Arts, Business, Sports,...
- **Which topic ranking to use?**
  - User can pick from a menu
  - Classify query into a topic
  - Can use the **context** of the query
    - E.g., query is launched from a web page talking about a known topic
    - History of queries e.g., “basketball” followed by “Jordan”
  - User context, e.g., user’s bookmarks, ...

# **Application to Measuring Proximity in Graphs**

Random Walk with Restarts: set  $S$  is a single node

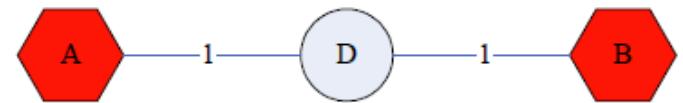
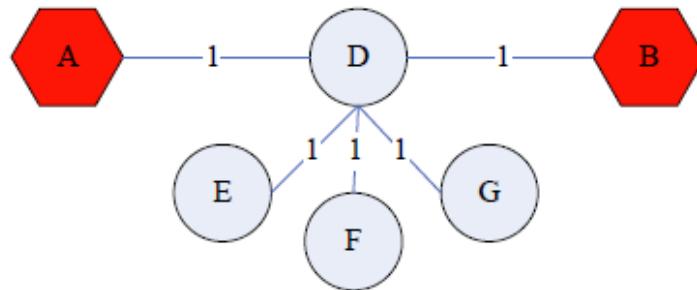
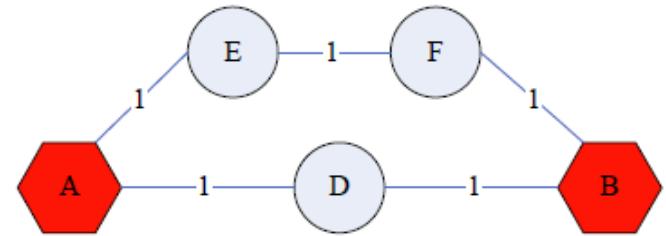
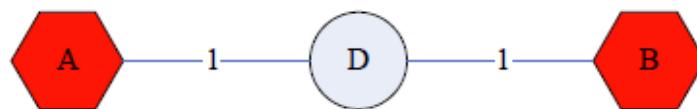
# Proximity on Graphs



a.k.a.: Relevance, Closeness, 'Similarity'...

# Good proximity measure?

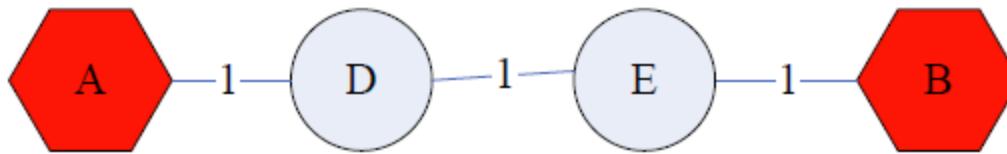
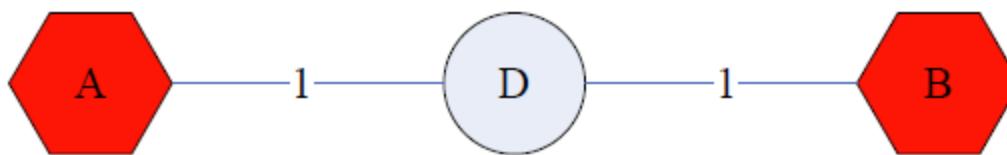
- Shortest path is not good:



- No effect of degree-1 nodes (E, F, G)!
- Multi-faceted relationships

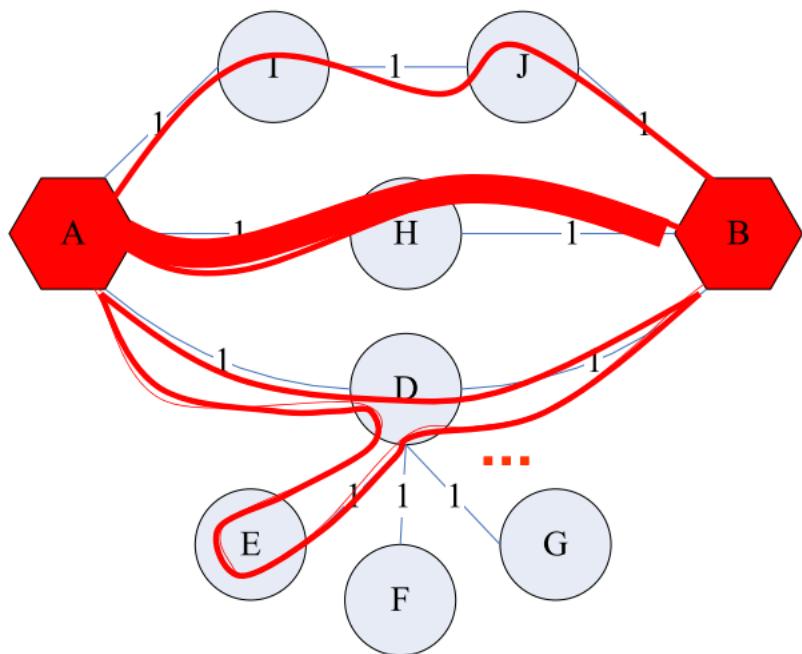
# Good proximity measure?

- Network flow is not good:



- Does not punish long paths

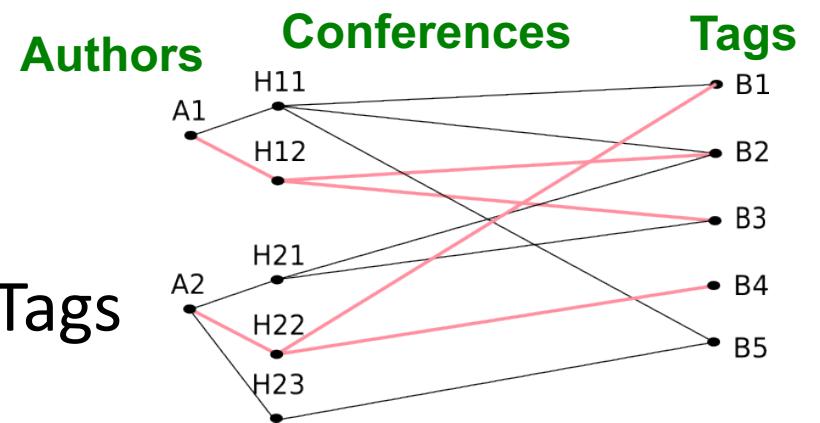
# What is a good notion of proximity?



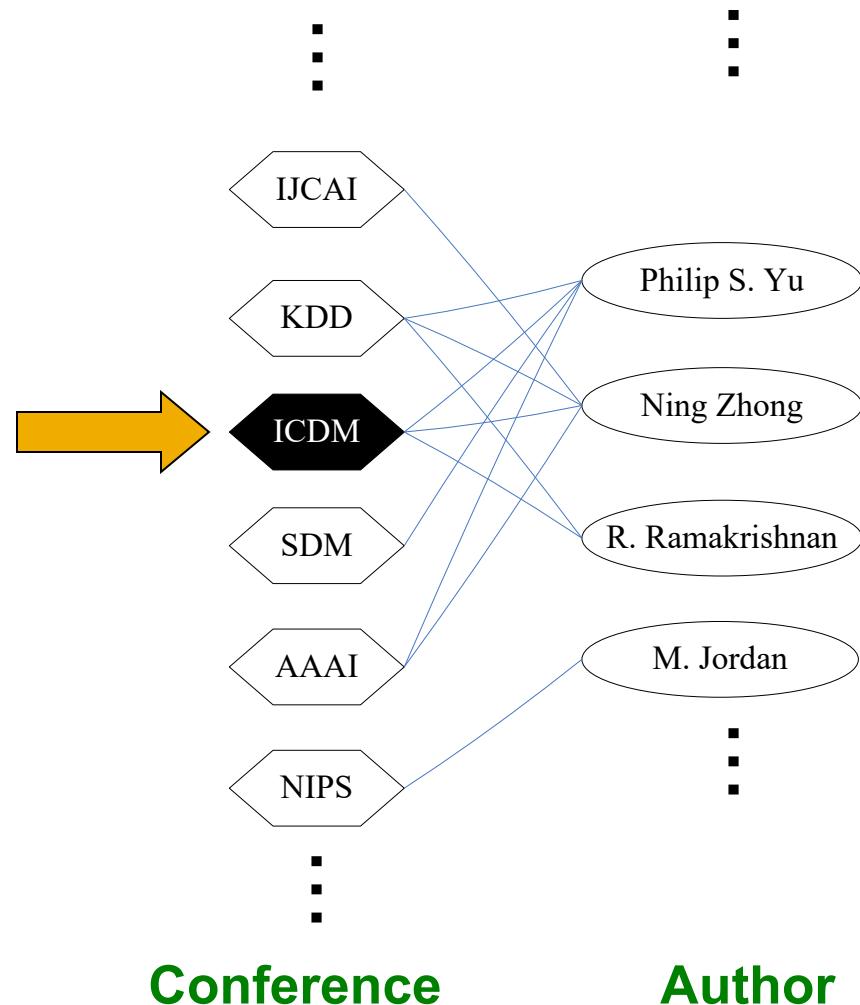
- Need a method that considers:
  - Multiple connections
  - Multiple paths
  - Direct and indirect connections
  - Degree of the node

# SimRank: Idea

- **SimRank:** Random walks from a **fixed node** on  $k$ -partite graphs
- **Setting:**  $k$ -partite graph with  $k$  types of nodes
  - E.g.: Authors, Conferences, Tags
- **Topic Specific PageRank** from node  $u$ : teleport set  $S = \{u\}$
- Resulting scores measure similarity/proximity to node  $u$
- **Problem:**
  - Must be done once for each node  $u$
  - Only suitable for sub-Web-scale applications



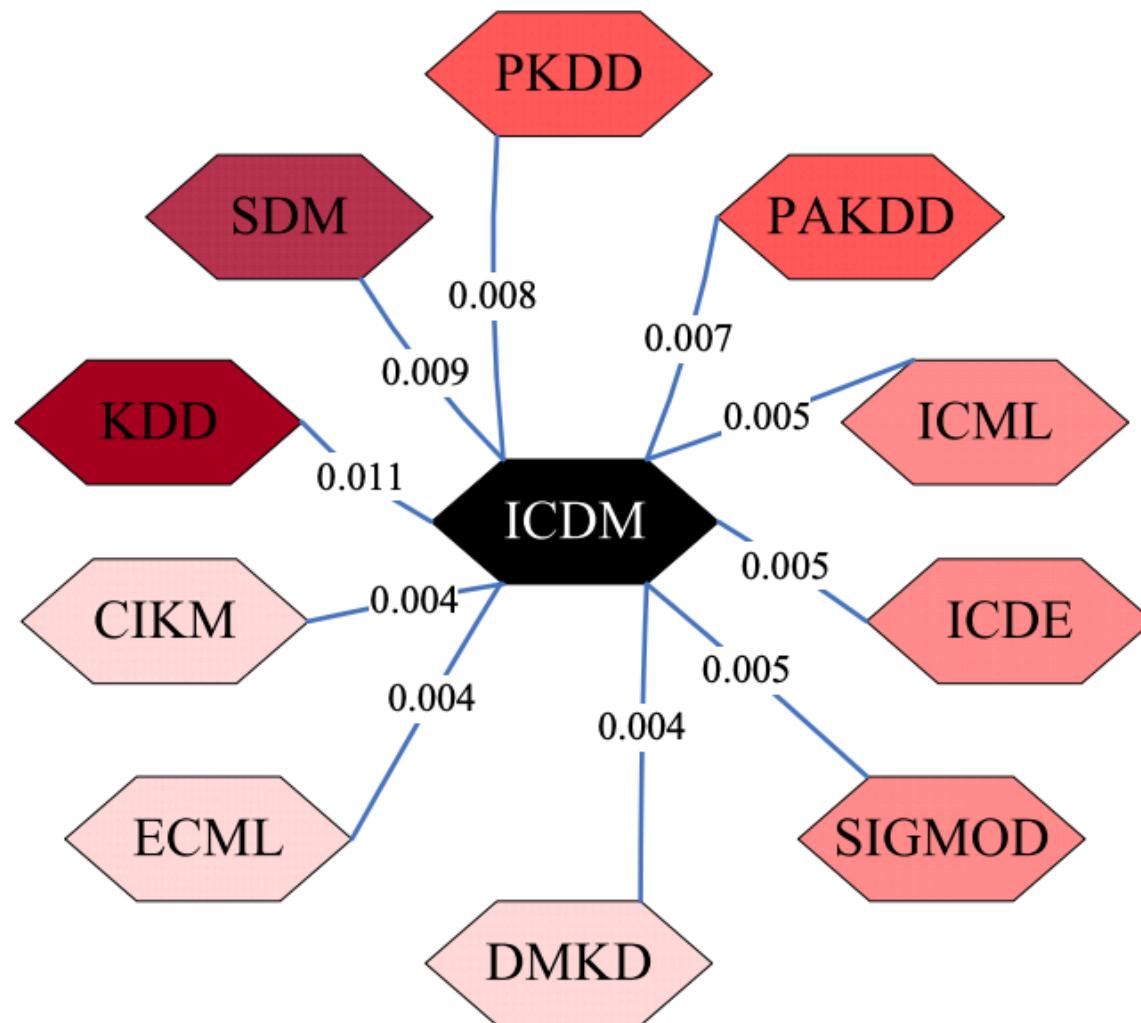
# SimRank: Example



**Q:** What is the most related conference to **ICDM**?

**A: Topic-Specific PageRank with teleport set  $S=\{ICDM\}$**

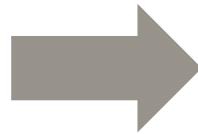
# SimRank: Example



# Pinterest: Pins and Boards



Pin

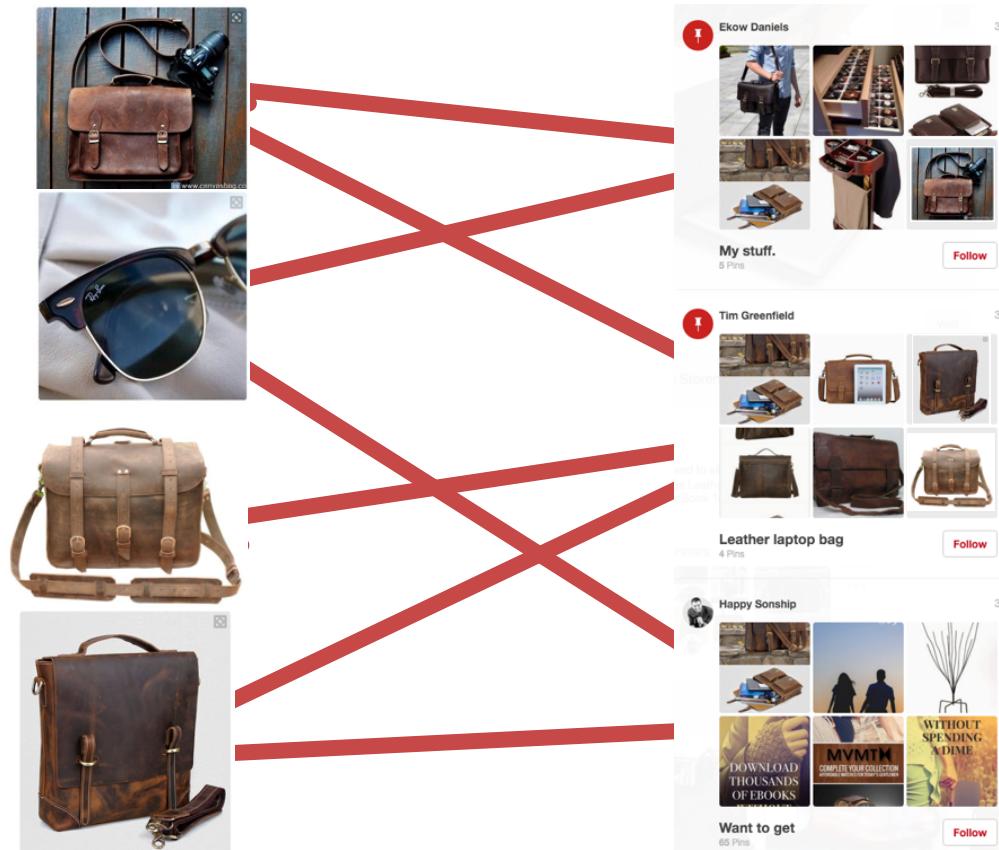


A screenshot of a Pinterest board titled 'Leather Messenger Bags'. The board features a grid of various leather messenger bags from different brands like EverHandmade, ugmonk, and Saddleback Leather Co. Each pin includes a small image, the item name, and a link back to the original source. The board has a light blue header and a white background for the pins.

Board

# Pinterest is a Giant Bipartite Graph

- Pins belong to Boards



# Pins to Pins Recommendations

Input:



HEALTHY CHOCOLATE STRAWBERRY SHAKE



Chocolate Strawberry Shake

249

This healthier chocolate strawberry shake is like sipping a...

One Lovely Life



Danielle Benzaia  
Strawberries

# Pins to Pins Recommendations

## Input: Recommendations:



**HEALTHY CHOCOLATE STRAWBERRY SHAKE**



Chocolate Strawberry Shake

This healthier chocolate strawberry shake is like sipping a...

One Lovely Life

Danielle Benzaia  
Strawberries



Chocolate Dipped Strawberry Smoothie  
Chocolate Dipped Strawberry Smoothie. Just in time for...  
Be Whole. Be You.  
Ed Todd  
Drinks- Smoothies



Tropical Orange Smoothie



Easy Breezy Tropical Orange Smoothie



**8 STAPLE SMOOTHIES**  
(THAT YOU SHOULD KNOW HOW TO MAKE)



8 Staple Smoothies You Should Know How To Make  
8 Staple Smoothies That You Should Know



The Perfect Vanilla Pumpkin Smoothie: A Quick &...

The perfect vanilla pumpkin smoothie recipe. Quick, easy and...

BabvSavers  
Marybeth @ Bab... Best Comfort Fo...



Spinach-Pear-Celery Smoothie  
drink this daily and watch the pounds come off without fuss...  
greenreset.com  
Spring Stutzman R - Drink Up



# Pins to Pins Recommendations

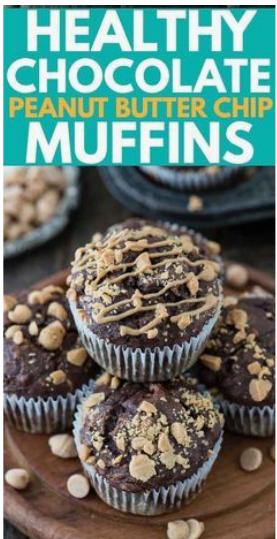
Input:



This healthier chocolate strawberry shake is like sipping a...

One Lovely Life

 Danielle Benzaia  
Strawberries



Healthy Chocolate Peanut Butter Chips Muffins

Healthy Chocolate Peanut Butter Chip Muffins made with greek...

The First Year  
 Katie - You Brew...  
 Healthy Recipes



The ULTIMATE Healthy Chocolate Chip Cookies -- so buttery...

Amv's Healthy Baking  
 Robin Guertin  
 healthy cooking

# Pins to Pins Recommendations

Input:



**Chocolate Strawberry Shake**

This healthier chocolate strawberry shake is like sipping a...

One Lovely Life

Danielle Benzaia  
Strawberries



**Healthy Chocolate Peanut Butter Chips Muffins**

Healthy Chocolate Peanut Butter Chip Muffins made with greek...

The First Year  
Katie - You Brew ...  
Healthy Recipes

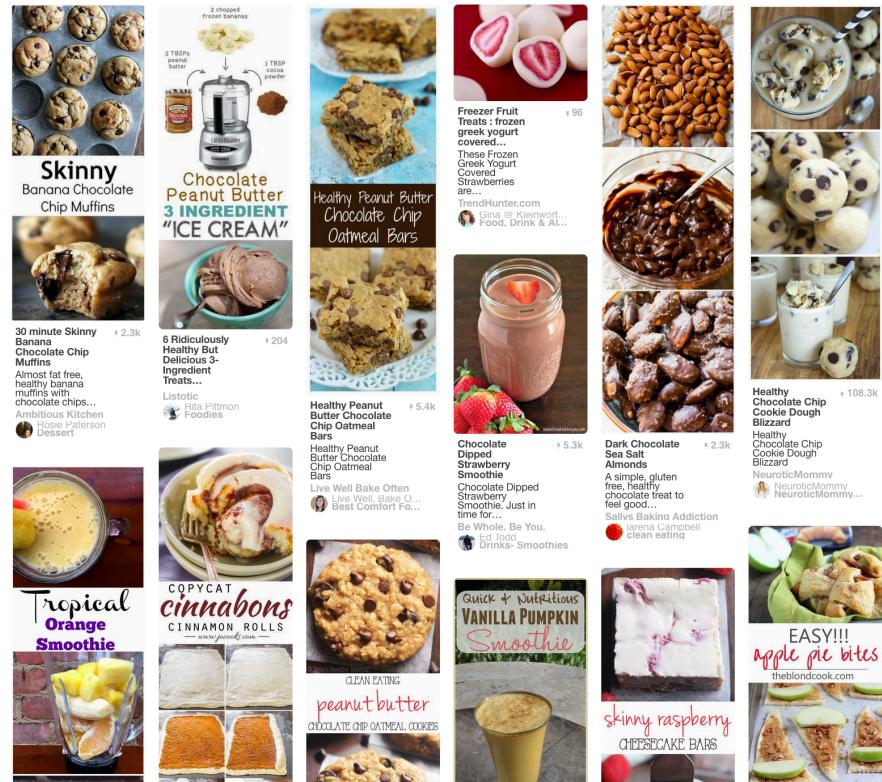


**The Ultimate Healthy Soft & Chewy Chocolate Chip Cookies**

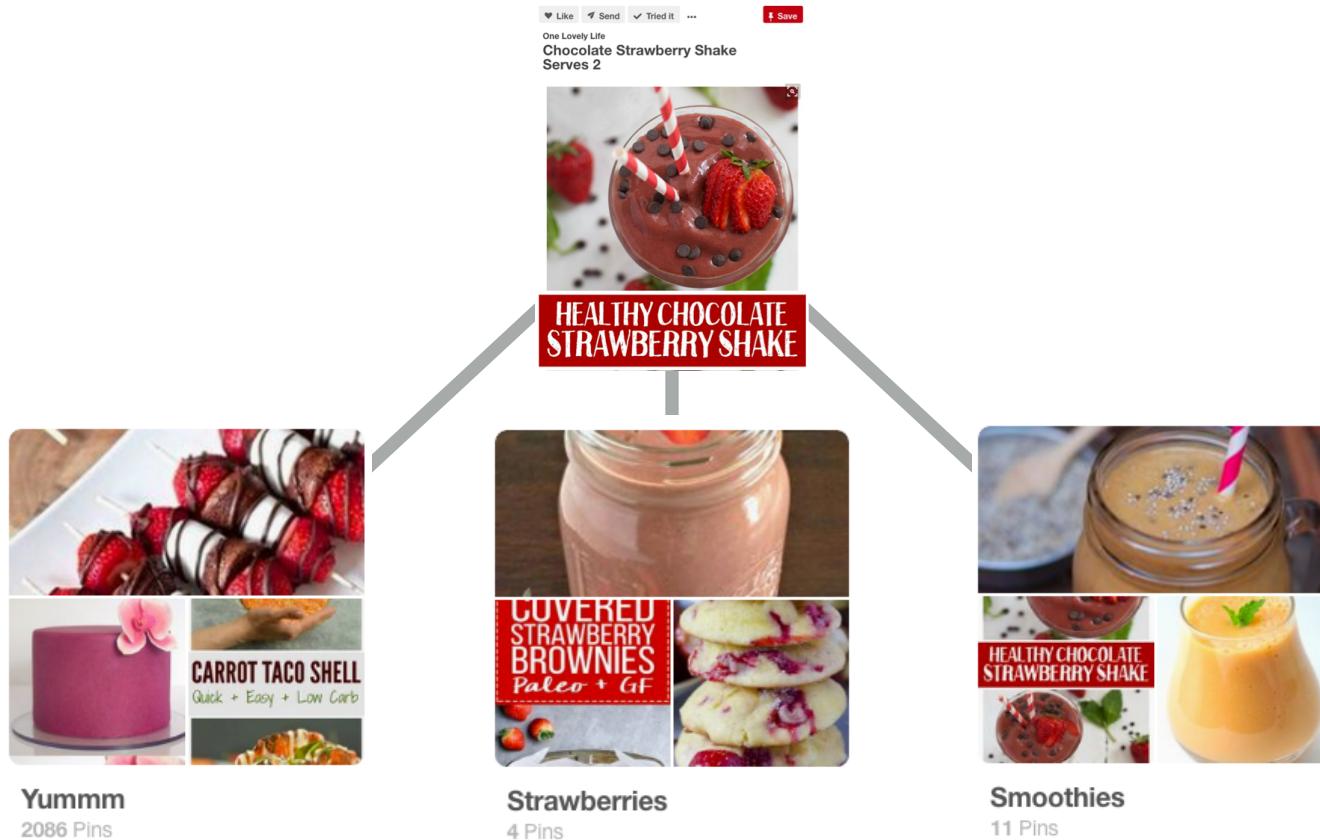
The ULTIMATE Healthy Chocolate Chip Cookies -- so buttery...

Amv's Healthy Baking  
Robin Guertin  
healthy cooking

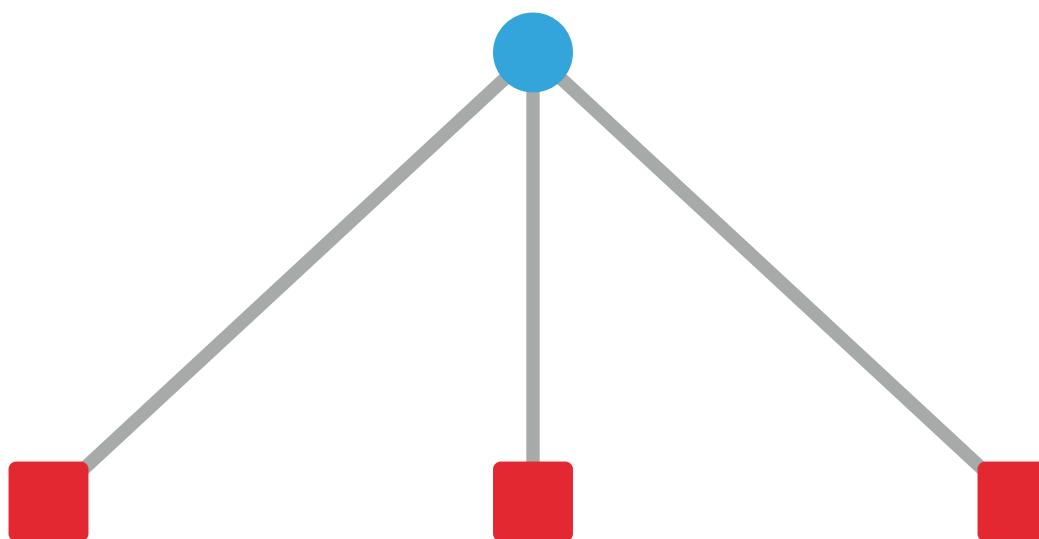
Recommendations:



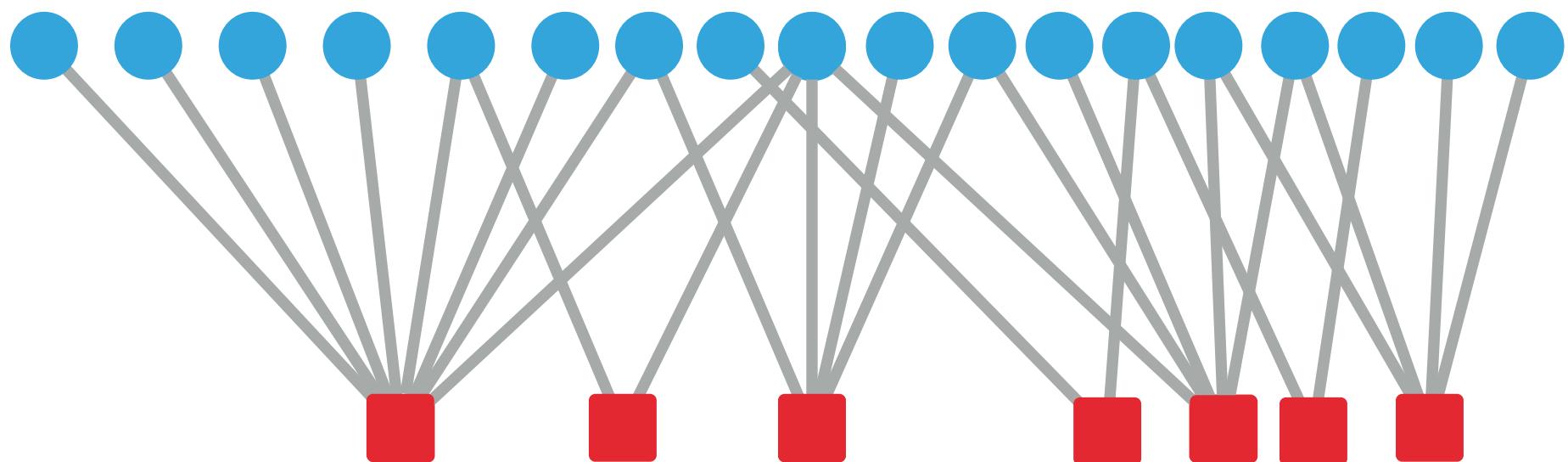
# Bipartite Pin And Board Graph



# Bipartite Pin And Board Graph

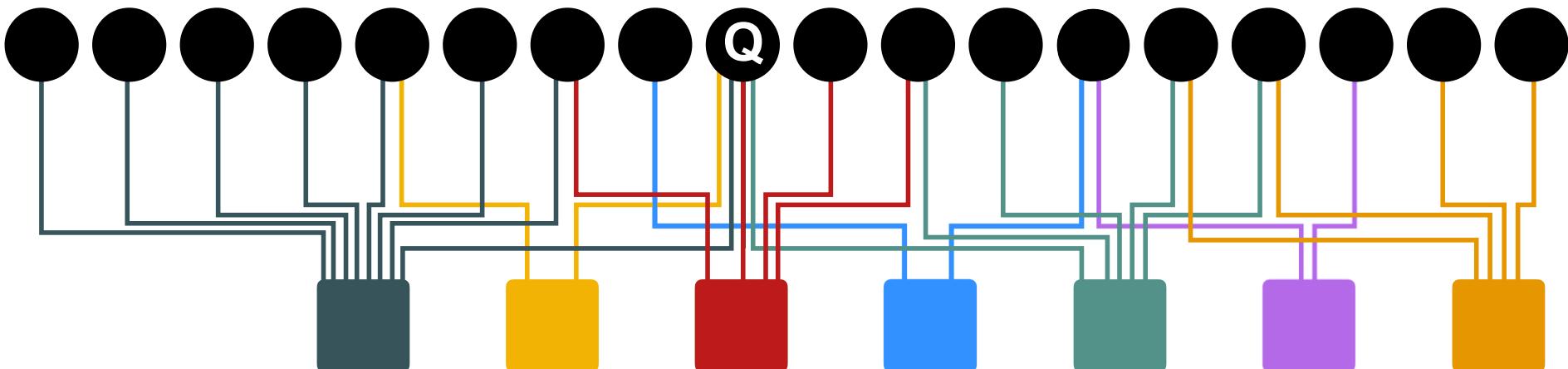


# Bipartite Pin And Board Graph



# Pixie Random Walks

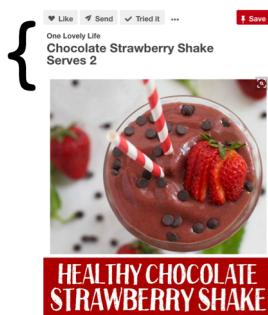
- Idea:
  - Every node has some importance
  - Importance gets evenly split among all edges and pushed to the neighbors
- Given a set of QUERY NODES Q, simulate a random walk:



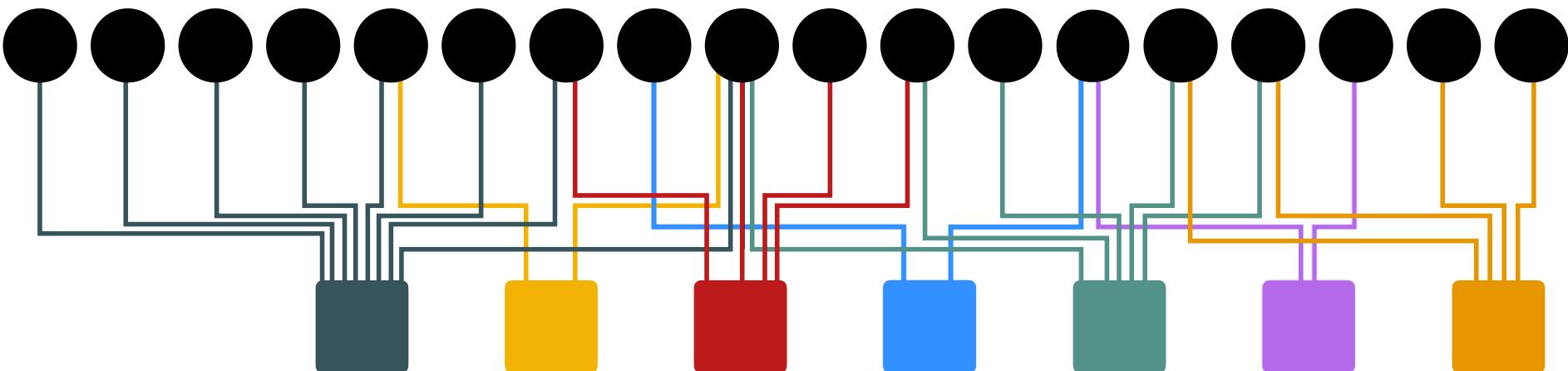
# Pixie Random Walk Algorithm

- Proximity to query node(s)  $Q$ :

```
ALPHA = 0.5  
QUERY_NODES =
```



```
{ } pin_node = QUERY_NODES.sample_by_weight()  
for i in range(N_STEPS):  
    board_node = pin_node.get_random_neighbor()  
    pin_node = board_node.get_random_neighbor()  
    pin_node.visit_count += 1  
    if random() < ALPHA:  
        pin_node = QUERY_NODES.sample_by_weight()
```



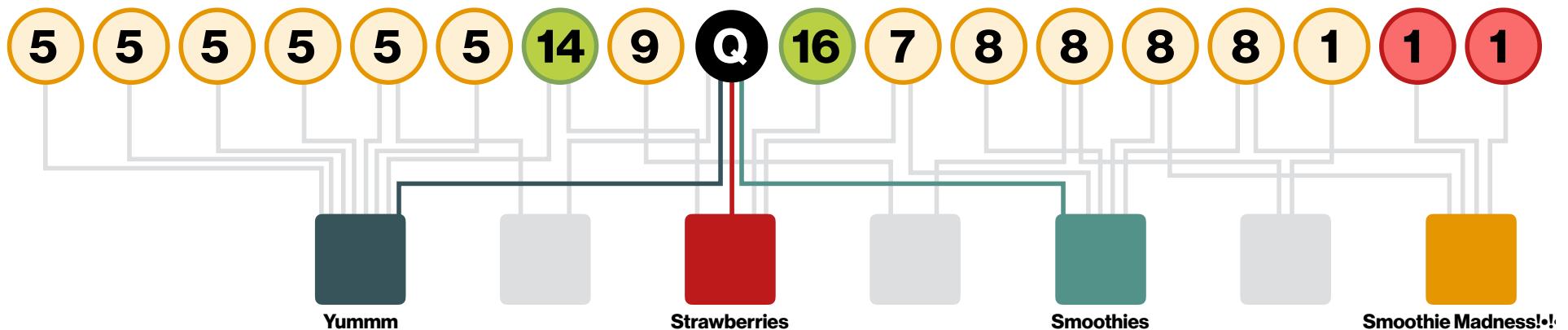
# Pixie Random Walk Algorithm

- Proximity to query node(s)  $Q$ :

```
ALPHA = 0.5  
QUERY_NODES =
```



```
{ } pin_node = QUERY_NODES.sample_by_weight()  
for i in range(N_STEPS):  
    board_node = pin_node.get_random_neighbor()  
    pin_node = board_node.get_random_neighbor()  
    pin_node.visit_count += 1  
    if random() < ALPHA:  
        pin_node = QUERY_NODES.sample_by_weight()
```



# Pixie Recommendations

- **Pixie:**
  - Outputs top 1k pins with highest visit count
- Extensions:**
  - **Weighted edges:**
    - The walk prefers to traverse certain edges:
      - Edges to pins in your local language
  - **Early stopping:**
    - Don't need to walk a fixed big number of steps
    - Walk until 1k-th pin has at least 20 visits

# Graph Cleaning/Pruning

- Pinterest graph has 200B edges
- We don't need all of them!
  - Super popular pins are pinned to millions of boards
    - Not useful: When the random walk hits the pin, the signal just disperses. Such pins appear randomly in our recommendations.
- What we did: Keep only good boards for pins
  - Compute the similarity between pin's topic vector and each of its boards. Only take boards with high similarity.

Data Type	Number	Size	Memory
Pin Nodes	3 Billion	8 Bytes	24 GiB
Board Nodes	2 Billion	8 Bytes	16 GiB
Undirected Edges	20 Billion	8 Bytes	160 GiB
			208 GiB

# Benefits of Pixie

## ■ Benefits:

- **Blazingly fast:** Given Q, we can output top 1k in 50ms (after doing 100k steps of the random walk)
  - Single machine can run 1500 walks in parallel! (1500 recommendation requests per second)
  - Can fit entire graph in RAM (17B edges, 3B nodes)
  - Can scale it by just adding more machines
- 
- **Today about 70% of all the pins you see at Pinterest are recommended by random walks**

# PageRank: Summary

- “Normal” PageRank:
  - Teleports uniformly at random to any node
  - All nodes have the same probability of surfer landing there:  $S = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$
- Topic-Specific PageRank also known as Personalized PageRank:
  - Teleports to a topic specific set of pages
  - Nodes can have different probabilities of surfer landing there:  $S = [0.1, 0, 0, 0.2, 0, 0, 0.5, 0, 0, 0.2]$
- Random Walk with Restarts:
  - Topic-Specific PageRank where teleport is always to the same node.  $S=[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]$

# **TrustRank: Combating Spam on the Web**

# What is Web Spam?

- **Spamming:**

- Any deliberate action to boost a web page's position in search engine results, incommensurate with the page's real value

- **Spam:**

- Web pages that are the result of spamming
- This is a very broad definition
  - SEO industry might disagree!
  - SEO = search engine optimization
- Approximately **10-15%** of web pages are spam

# Web Search

- **Early search engines:**
  - Crawl the Web
  - Index pages by the words they contained
  - Respond to search queries (lists of words) with the pages containing those words
- **Early page ranking:**
  - Attempt to order pages matching a search query by “importance”
  - **First search engines considered:**
    - **(1)** Number of times query words appeared
    - **(2)** Prominence of word position, e.g. title, header

# First Spammers

- As people began to use search engines to find things on the Web, those with commercial interests tried to **exploit search engines** to bring people to their own site – whether they wanted to be there or not
- **Example:**
  - Shirt-seller might pretend to be about “movies”
- **Techniques for achieving high relevance/importance for a web page**

# First Spammers: Term Spam

- How do you make your page appear to be about movies?
  - (1) Add the word movie 1,000 times to your page
    - Set text color to the background color, so only search engines would see it
  - (2) Or, run the query “movie” on your target search engine
    - See what page came on top of result ranking
    - Copy it into your page, make it “invisible”
- These and similar techniques are term spam

# Google's Solution to Term Spam

- **Believe what people say about you, rather than what you say about yourself**
  - Use words in the anchor text (words that appear underlined to represent the link) and its surrounding text
- **PageRank as a tool to measure the “importance” of Web pages**

# Why It Works?

- **Our hypothetical shirt-seller loses**
  - Saying he is about movies doesn't help, because others don't say he is about movies
  - His page isn't very important, so it won't be ranked high for shirts or movies
- **Example:**
  - Shirt-seller creates 1,000 pages, each links to his with "movie" in the anchor text
  - These pages have no links in, so they get little PageRank
  - So the shirt-seller can't beat truly important movie pages, like IMDB

# Why it does not work?

Google™ Web Images Groups News Froogle Local more » miserable failure Search Advanced Search Preferences

**Web** Results 1 - 10 of about 969,000 for [miserable failure](#). (0.06 seconds)

[Biography of President George W. Bush](#)  
Biography of the president from the official White House web site.  
[www.whitehouse.gov/president/gwbbio.html](http://www.whitehouse.gov/president/gwbbio.html) - 29k - [Cached](#) - [Similar pages](#)  
[Past Presidents](#) - [Kids Only](#) - [Current News](#) - [President](#)  
[More results from www.whitehouse.gov »](#)

[Welcome to MichaelMoore.com!](#)  
Official site of the gadfly of corporations, creator of the film Roger and Me  
and the television show The Awful Truth. Includes mailing list, message board, ...  
[www.michaelmoore.com/](http://www.michaelmoore.com/) - 35k - Sep 1, 2005 - [Cached](#) - [Similar pages](#)

[BBC NEWS | Americas | 'Miserable failure' links to Bush](#)  
Web users manipulate a popular search engine so an unflattering description leads  
to the president's page.  
[news.bbc.co.uk/2/hi/americas/3298443.stm](http://news.bbc.co.uk/2/hi/americas/3298443.stm) - 31k - [Cached](#) - [Similar pages](#)

[Google's \(and Inktomi's\) Miserable Failure](#)  
A search for **miserable failure** on Google brings up the official George W.  
Bush biography from the US White House web site. Dismissed by Google as not a ...  
[searchenginewatch.com/sereport/article.php/3296101](http://searchenginewatch.com/sereport/article.php/3296101) - 45k - Sep 1, 2005 - [Cached](#) - [Similar pages](#)



**SPAM FARMING**

# Google vs. Spammers: Round 2!

- Once Google became the dominant search engine, spammers began to work out ways to fool Google
- Spam farms** were developed to concentrate PageRank on a single page
- Link spam:**
  - Creating link structures that boost PageRank of a particular page



# Link Spamming

- Three kinds of web pages from a spammer's point of view
  - Inaccessible pages
  - Accessible pages
    - e.g., blog comments pages
    - spammer can post links to his pages
  - Owned pages
    - Completely controlled by spammer
    - May span multiple domain names

# Link Farms

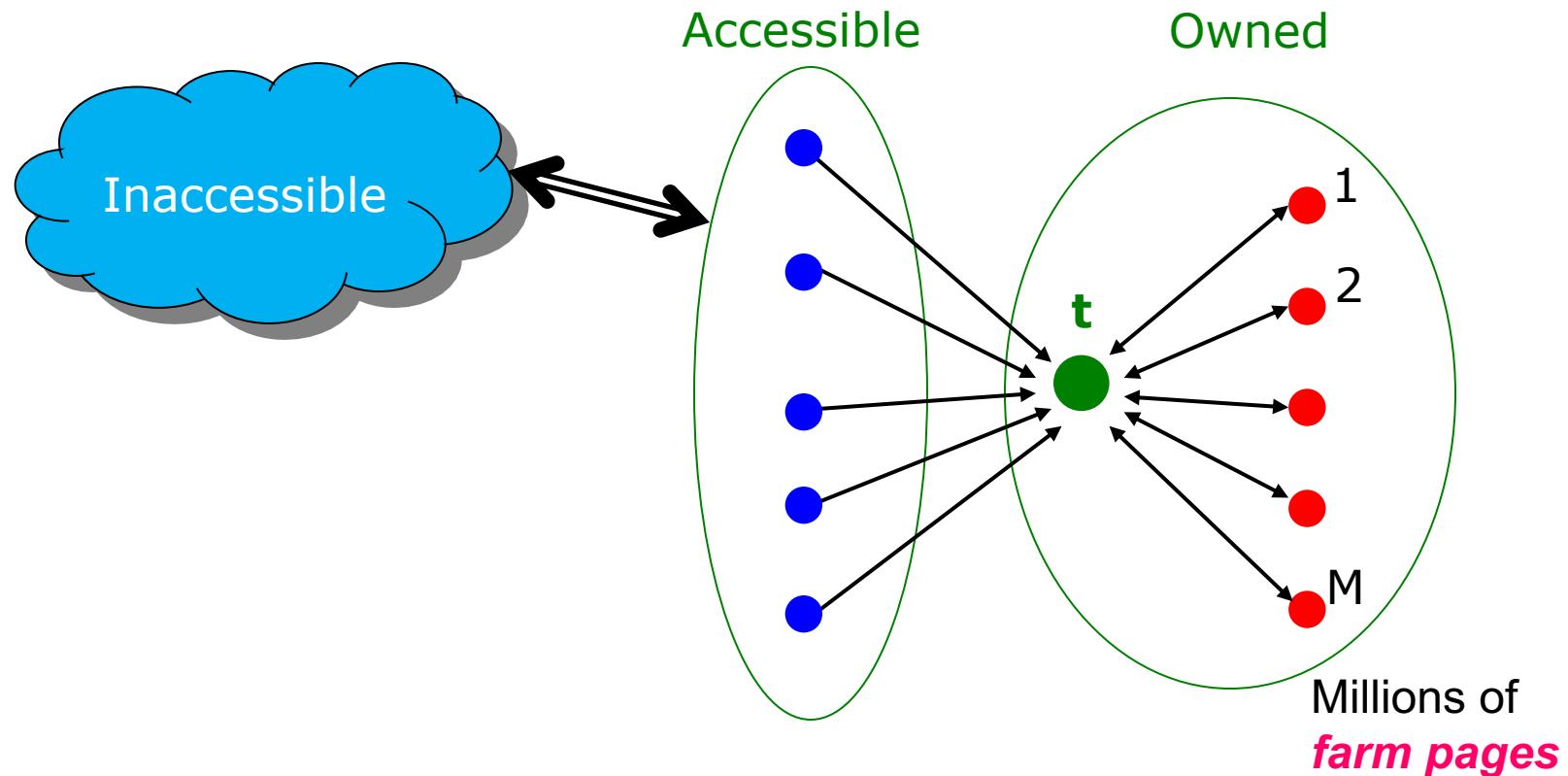
- **Spammer's goal:**

- Maximize the PageRank of target page  $t$

- **Technique:**

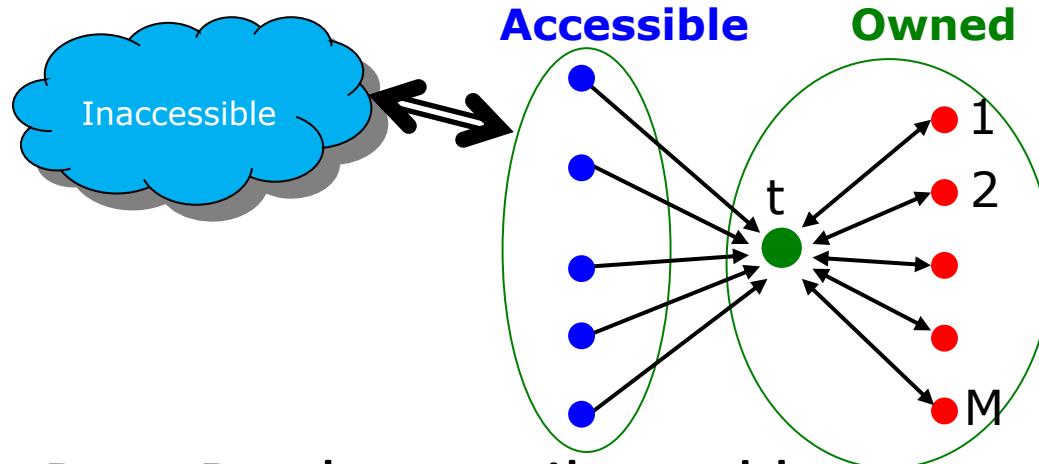
- Get as many links from accessible pages as possible to target page  $t$
  - Construct “link farm” to get PageRank multiplier effect

# Link Farms



**One of the most common and effective organizations for a link farm**

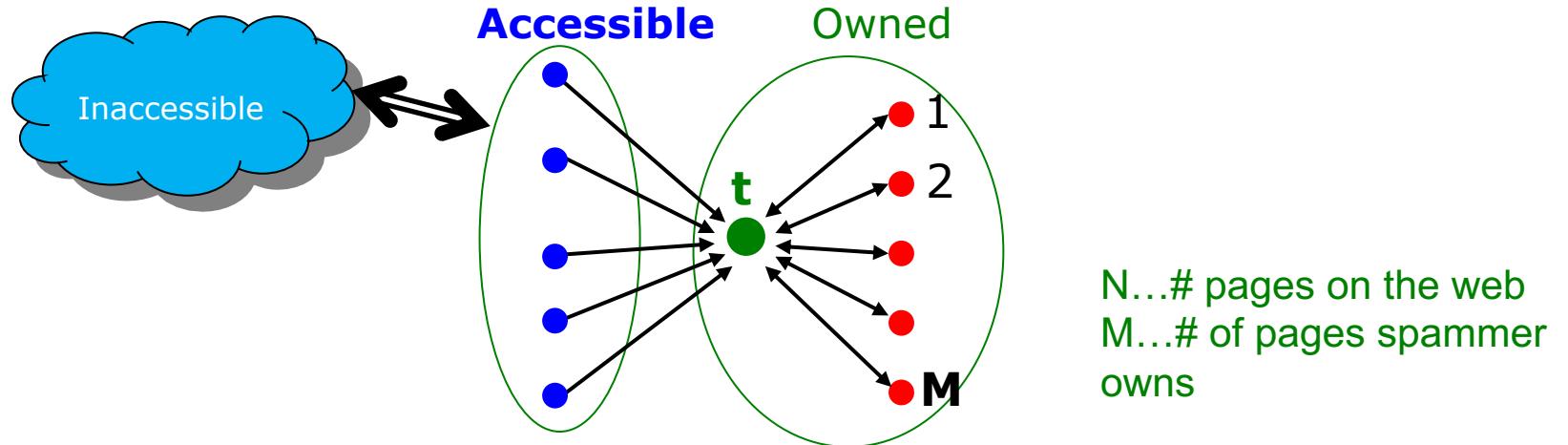
# Analysis



N...# pages on the web  
M...# of pages spammer owns

- $x$ : PageRank contributed by accessible pages
- $y$ : PageRank of target page  $t$
- Rank of each “farm” page =  $\frac{\beta y}{M} + \frac{1-\beta}{N}$
- $y = x + \beta M \left[ \frac{\beta y}{M} + \frac{1-\beta}{N} \right] + \frac{1-\beta}{N}$   
 $= x + \beta^2 y + \frac{\beta(1-\beta)M}{N} + \boxed{\frac{1-\beta}{N}}$ 
  - Very small; ignore
  - Now we solve for  $y$
- $y = \frac{x}{1-\beta^2} + c \frac{M}{N}$  where  $c = \frac{\beta}{1+\beta}$

# Analysis



- $y = \frac{x}{1-\beta^2} + c \frac{M}{N}$  where  $c = \frac{\beta}{1+\beta}$
- For  $\beta = 0.85$ ,  $1/(1-\beta^2) = 3.6$
- Multiplier effect for acquired PageRank
- By making  $M$  large, we can make  $y$  as large as we want

# **TrustRank: Combating Spam on the Web**

# Combating Spam

- **Combating term spam**
  - Analyze text using statistical methods
  - Similar to email spam filtering
  - Also useful: Detecting approximate duplicate pages
- **Combating link spam**
  - **Detection and blacklisting of structures that look like spam farms**
    - Leads to another war – hiding and detecting spam farms
  - **TrustRank** = topic-specific PageRank with a teleport set of **trusted pages**
    - Example: .edu domains, similar domains for non-US schools

# TrustRank: Idea

- Basic principle: **Approximate isolation**
  - It is rare for a “good” page to point to a “bad” (spam) page
- Sample a set of **seed pages** from the web
- Have an **oracle (human)** to identify the good pages and the spam pages in the seed set
  - **Expensive task**, so we must make seed set as small as possible

# Trust Propagation

- Call the subset of seed pages that are identified as **good** the **trusted pages**
- Perform a topic-sensitive PageRank with **teleport set = trusted pages**
  - **Propagate trust through links:**
    - Each page gets a trust value between **0** and **1**
- **Solution 1: Use a threshold value and mark all pages below the trust threshold as spam**

# Simple Model: Trust Propagation

- Set trust of each trusted page to 1
- Suppose trust of page  $p$  is  $t_p$ 
  - Page  $p$  has a set of out-links  $o_p$
- For each  $q \in o_p$ ,  $p$  **confers the trust** to  $q$ 
  - $\beta t_p / |o_p|$  for  $0 < \beta < 1$
- **Trust is additive**
  - Trust of  $p$  is the sum of the trust conferred on  $p$  by all its in-linked pages
- **Note similarity to Topic-Specific PageRank**
  - Within a scaling factor, **TrustRank = PageRank** with trusted pages as teleport set

# Why is it a good idea?

- **Trust attenuation:**

- The degree of trust conferred by a trusted page decreases with the distance in the graph

- **Trust splitting:**

- The larger the number of out-links from a page, the less scrutiny the page author gives each out-link
  - Trust is **split** across out-links

# Picking the Seed Set

- **Two conflicting considerations:**
  - Human has to inspect each seed page, so seed set must be as small as possible
  - Must ensure every **good page** gets adequate trust rank, so need make all good pages reachable from seed set by short paths

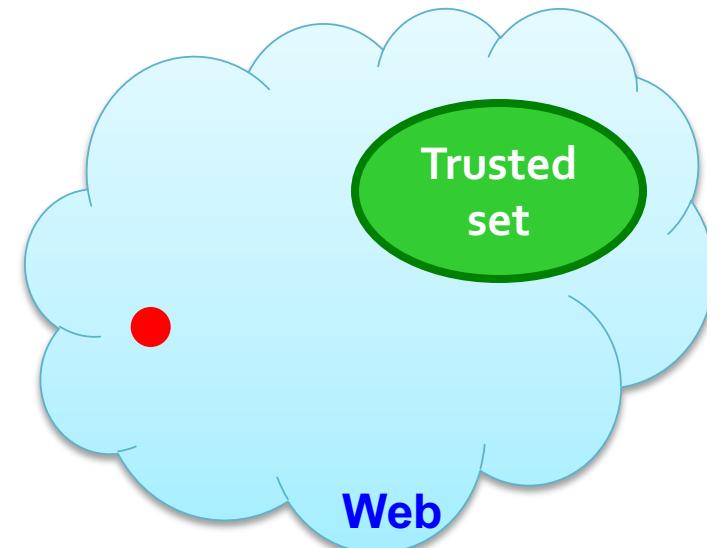
# Approaches to Picking Seed Set

- Suppose we want to pick a seed set of  $k$  pages
- **How to do that?**
- **(1) PageRank:**
  - Pick the top  $k$  pages by PageRank
  - Theory is that you can't get a bad page's rank really high
- **(2) Use trusted domains** whose membership is controlled, like .edu, .mil, .gov

# TrustRank

# Spam Mass

- In the **TrustRank** model, we start with good pages and propagate trust
- **Complementary view:**  
What fraction of a page's PageRank comes from **spam** pages?
- In practice, we don't know all the spam pages, so we need to estimate



# Spam Mass Estimation

## Solution 2:

- $r_p$  = PageRank of page  $p$
- $r_p^+$  = PageRank of  $p$  with teleport into **trusted** pages only
- Then: What fraction of a page's PageRank comes from **spam** pages?  
$$r_p^- = r_p - r_p^+$$

- **Spam mass of  $p$**  =  $\frac{r_p^-}{r_p}$ 
  - Pages with high spam mass are spam

