

**Note to other teachers and users of these slides:** We would be delighted if you found our material useful for giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. If you make use of a significant portion of these slides in your own lecture, please include this message, or a link to our web site: <http://www.mmds.org>

# Link Analysis: TrustRank and WebSpam

CS246: Mining Massive Datasets

Jure Leskovec, Stanford University

Mina Ghashami, Amazon

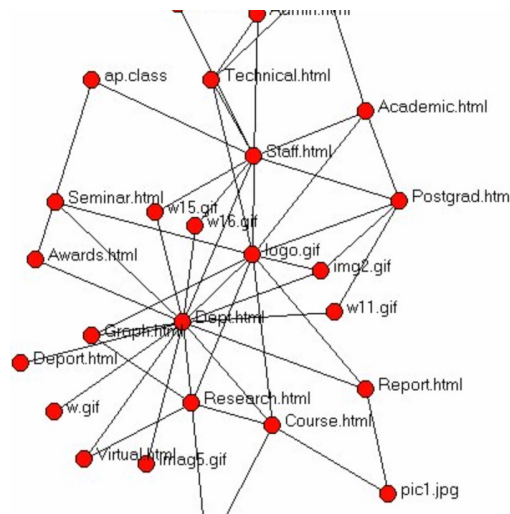
<http://cs246.stanford.edu>



# What We Saw So Far



Random surfer  
browsing the web



Random walker  
walking the graph

Transition matrix  $M$

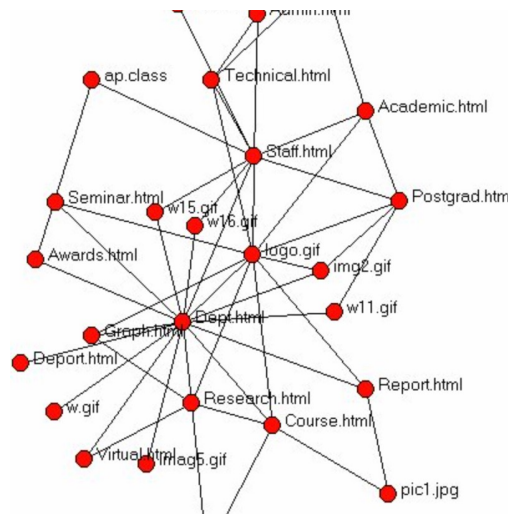
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix}$$

What is the stationary  
distribution of the random  
walker?

# What We Saw So Far



Random surfer  
browsing the web



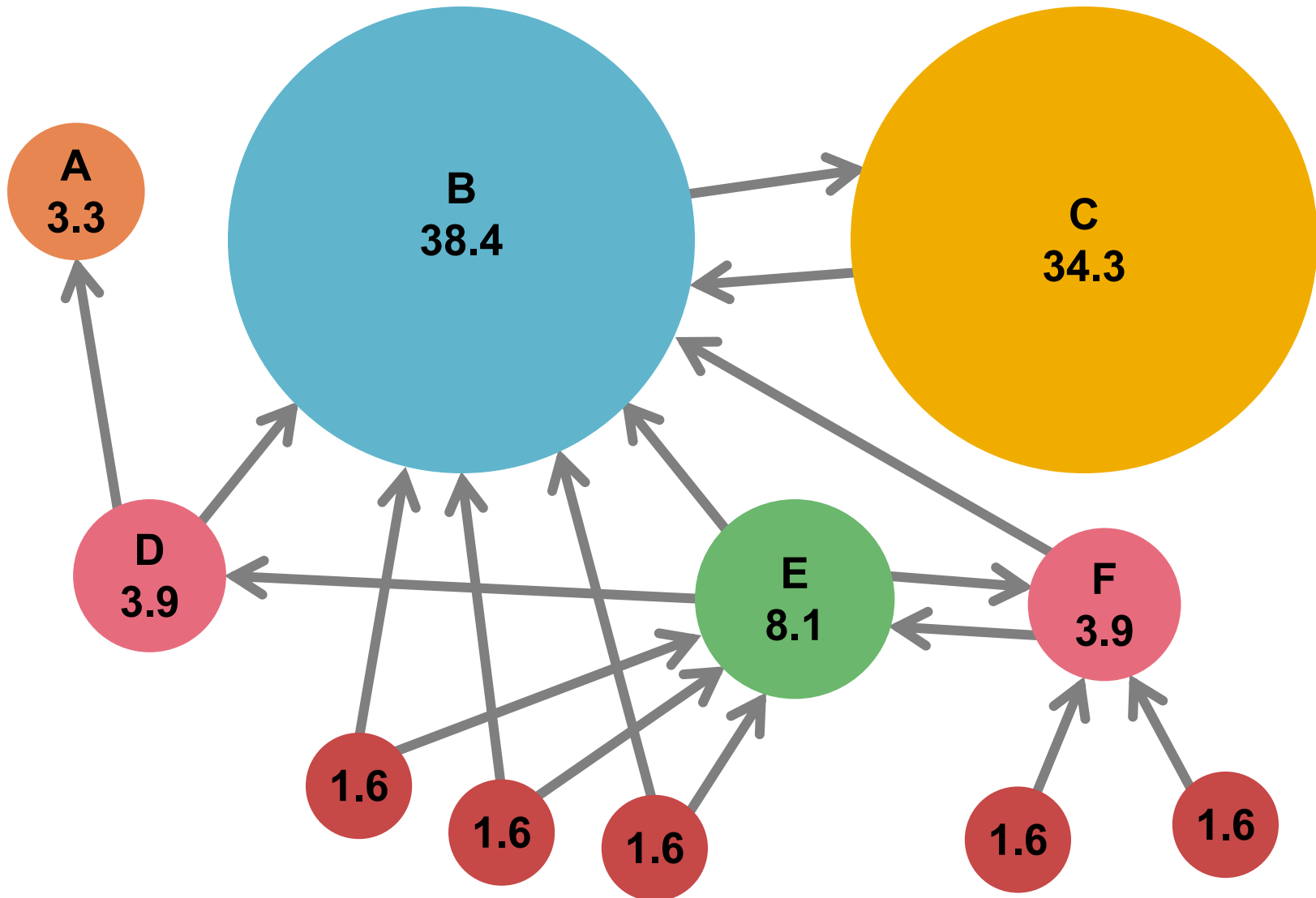
Random walker  
walking the graph

Power  
Method  $Mv = v$

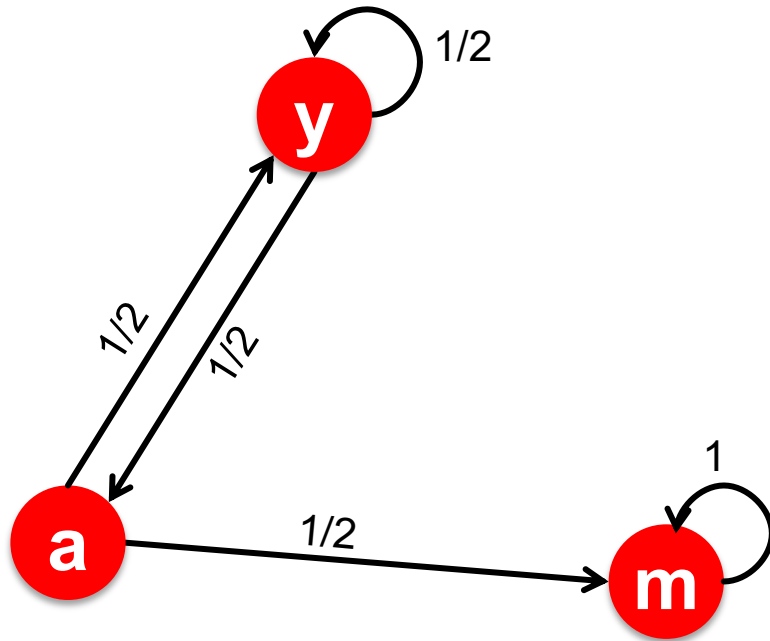
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix}$$

Principal Eigenvector  
of the transition Matrix

# Example: PageRank Scores



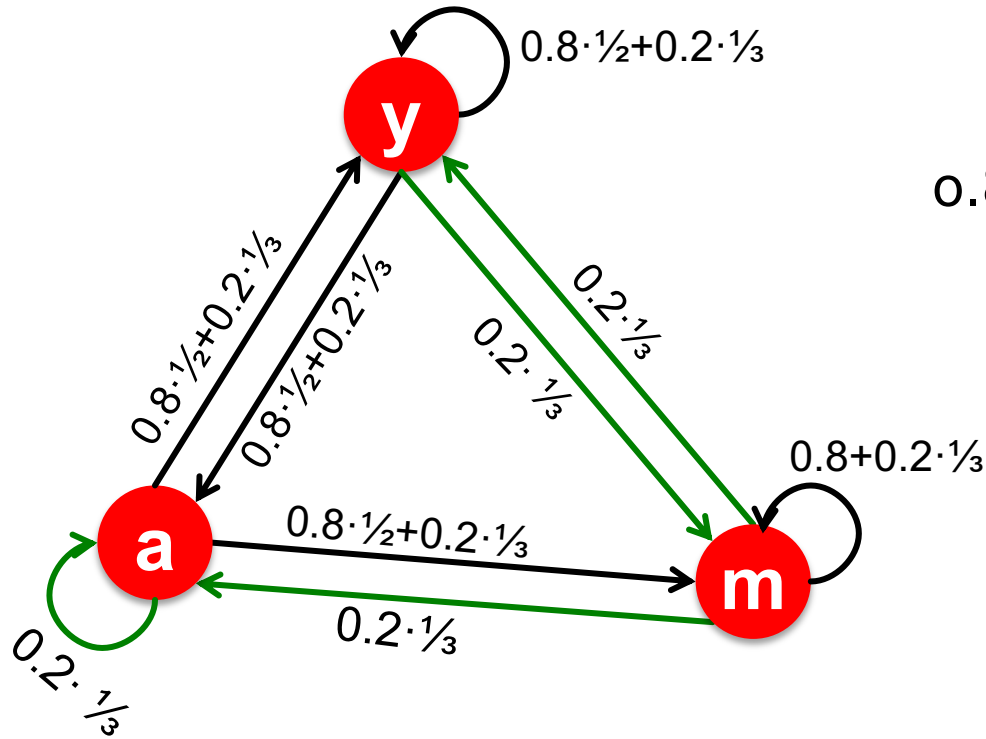
# Random Teleports ( $\beta = 0.8$ )



**M**

1/2	1/2	0
1/2	0	0
0	1/2	1

# Random Teleports ( $\beta = 0.8$ )



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

**A**

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 & 0.33 & 0.28 & 0.26 \\ 1/3 & 0.20 & 0.20 & 0.18 & \dots \\ 1/3 & 0.46 & 0.52 & 0.56 \end{matrix} \quad \begin{matrix} 7/33 \\ 5/33 \\ 21/33 \end{matrix}$$

$$\mathbf{r} = \mathbf{A} \mathbf{r}$$

# 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**  $\sum_j |r_j^{(t)} - r_j^{(t-1)}| < \epsilon$

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 importance 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 importance, can we measure importance within a topic?**
- **Goal:** Evaluate Web pages not just according to their importance, but also by how close they are to a particular topic, e.g. “sports” or “history”
- **Allows search queries to be answered based on the interests of a 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 randomly from the teleport 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  $r_S$

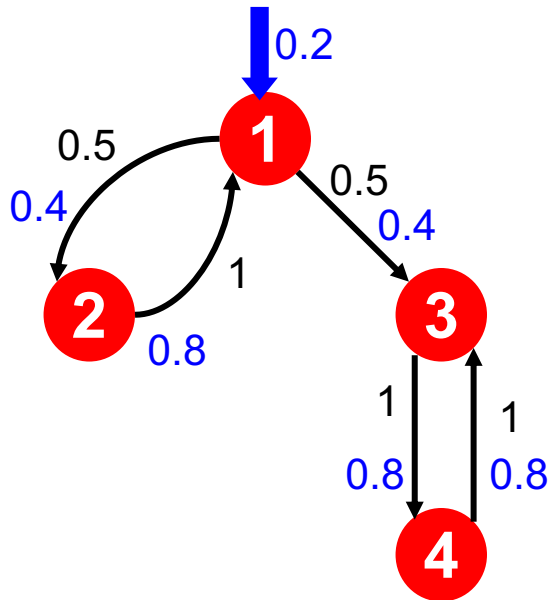
# Matrix Formulation

- 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 of  $(1 - \beta)/|S|$
  - Maintains sparseness

# Example: Topic-Specific PageRank



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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

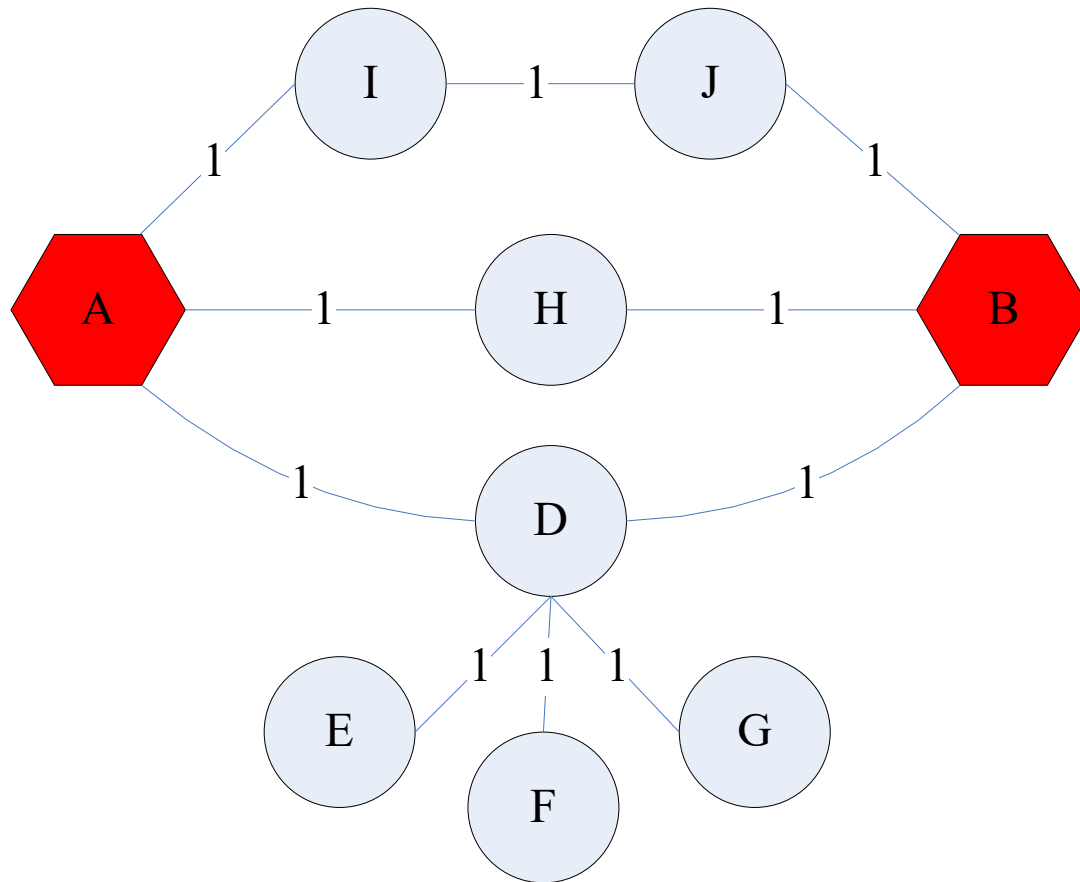
# Discovering the Topic Vector $S$

- **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

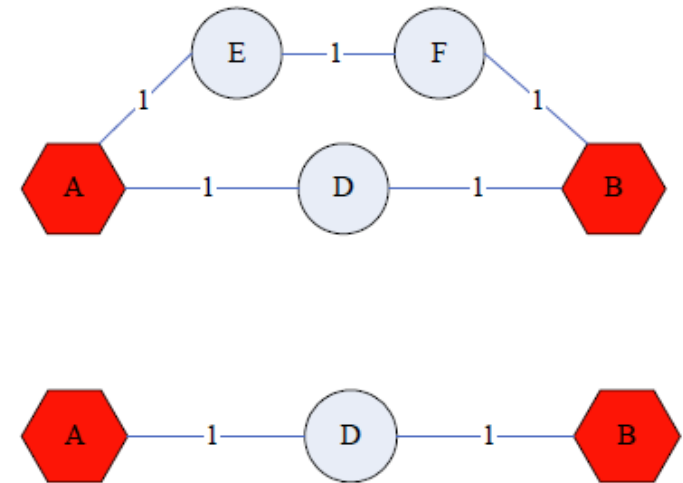
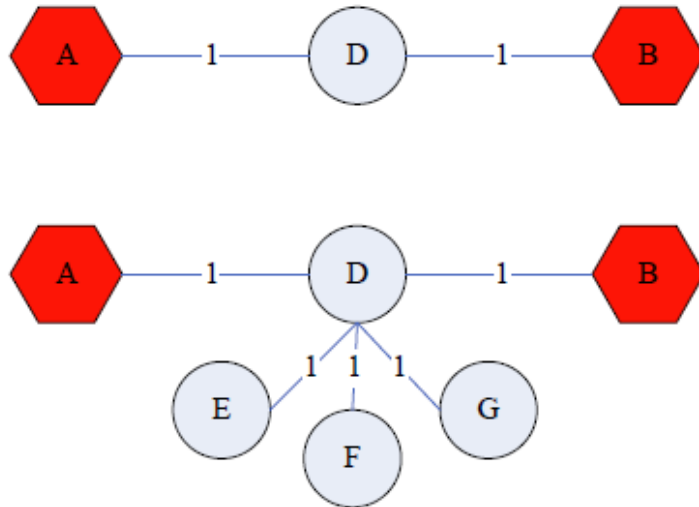


**a.k.a.: Relevance, Closeness, 'Similarity'...**



# Good proximity measure?

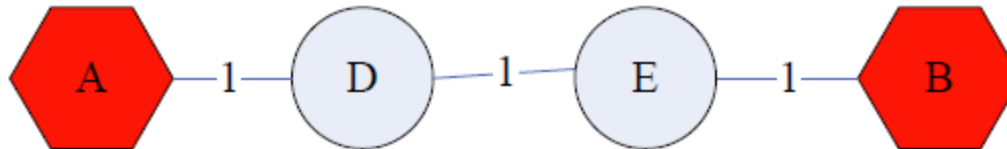
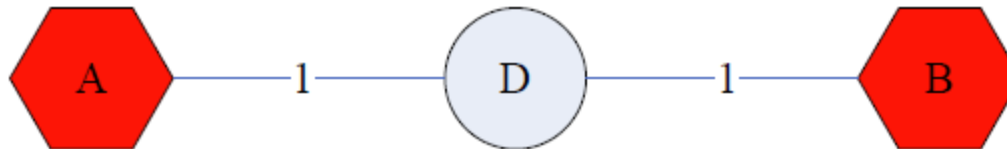
- Shortest path is not good:



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

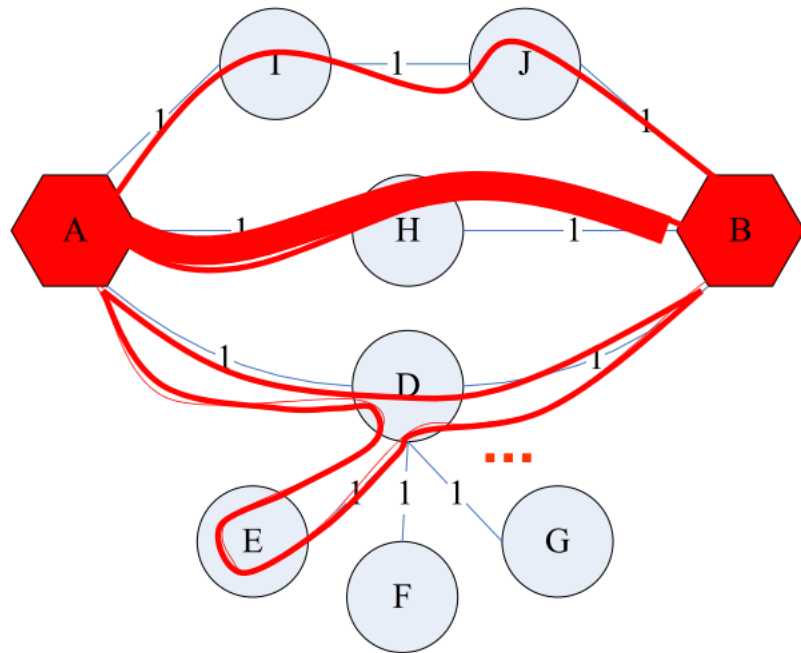
# Good proximity measure?

- Network flow is not good:



- Does not punish long paths

# What is a good notion of proximity?



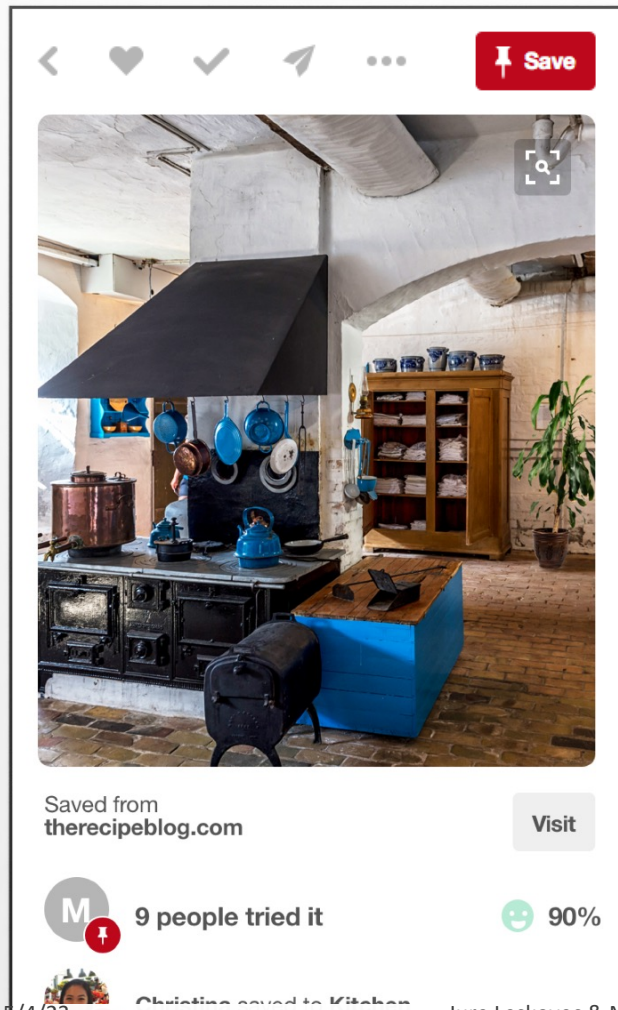
- **Need a method that considers:**

- Multiple connections
- Multiple paths
- Direct and indirect connections
- Degree of the node

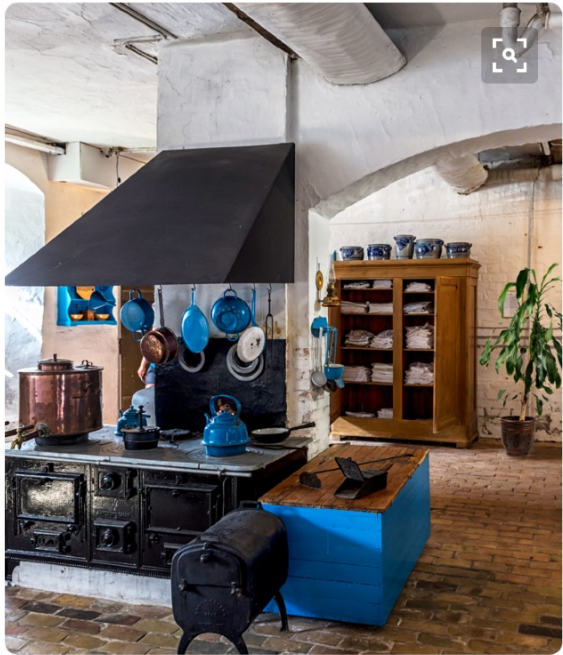
# Pixie: Random Walk-based Real-Time Recommender System at Pinterest

[https://labs.pinterest.com/user/themes/pin\\_labs/assets/paper/paper-pixie.pdf](https://labs.pinterest.com/user/themes/pin_labs/assets/paper/paper-pixie.pdf)

# Pinterest



Navigation icons: back, heart, checkmark, share, and menu. A red 'Save' button is in the top right.



Search icon in the top right of the image.

Saved from [therecipeblog.com](http://therecipeblog.com) Visit

M 9 people tried it 90%

Christina saved to Kitchen



**Blue accents**  
219 Pins



**Vintage kitchen**  
377 Pins



**Fireplace**  
138 Pins

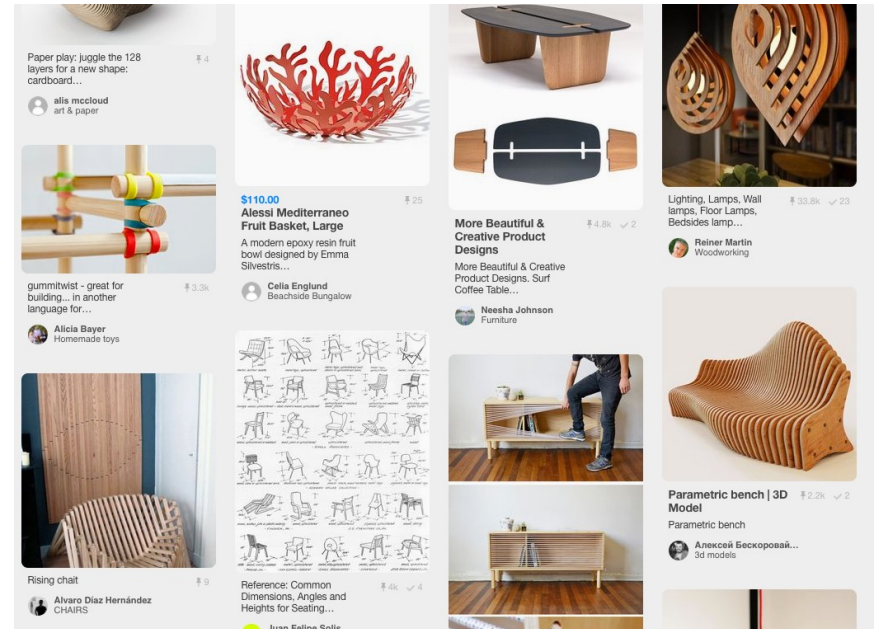
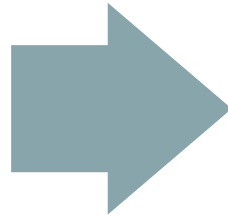
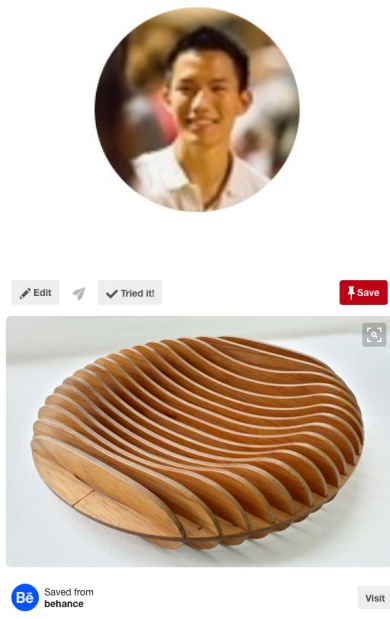
# Goal: Radical Personalization

- **Recommendations can be radically personalized.**
- Relevant recommendations
- Adapting in real-time(~50ms)
- Cheap to serve
- Easy to explain
- Highly scalable

# Recommendation problem

## How to provide relevant and responsive recommendations

- From 100B Pins to 1K Pins in **real-time (50ms, 200,000x/s)**



# From Pins to Pins

Input:



**HEALTHY CHOCOLATE STRAWBERRY SHAKE**



**Chocolate Strawberry Shake**

↑ 249

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

One Lovely Life



Danielle Berzaia  
Strawberries



# From Pins to Pins

## ■ Pins to Pins

Input:

Output:



**HEALTHY CHOCOLATE STRAWBERRY SHAKE**



**Chocolate Strawberry Shake** † 249  
This healthier chocolate strawberry shake is like sipping a...  
One Lovely Life  
Danielle Berzaia Strawberries



**Chocolate Dipped Strawberry Smoothie** † 5.3k  
Chocolate Dipped Strawberry Smoothie. Just in time for...  
Be Whole. Be You.  
Ed Todd Drinks- Smoothies



**Tropical Orange Smoothie**



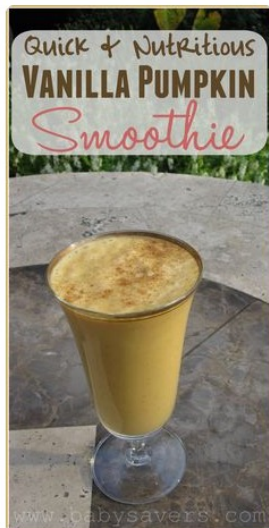
**Easy Breezy Tropical Orange Smoothie** † 80.1k



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



**8 Staple Smoothies You Should Know How to Make** † 5.2k  
8 Staple Smoothies That You Should Know



**Quick & Nutritious VANILLA PUMPKIN Smoothie** † 11.4k

**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** † 60  
drink this daily and watch the pounds come off without fuss...  
areenreset.com  
Spring Stutzman R - Drink Up



# From Pins to Pins

Input:



**Chocolate Strawberry Shake** † 249

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

One Lovely Life

 Danielle Benzaia  
Strawberries



**Healthy Chocolate Peanut Butter Chips Muffins** † 119

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** † 221

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

Amy's Healthy Baking  
 Robin Guertin  
healthy cooking

# From Pins to Pins

## Input:



**Chocolate Strawberry Shake** † 249  
 This healthier chocolate strawberry shake is like sipping a...  
 One Lovely Life  
 Danielle Benzaia Strawberries

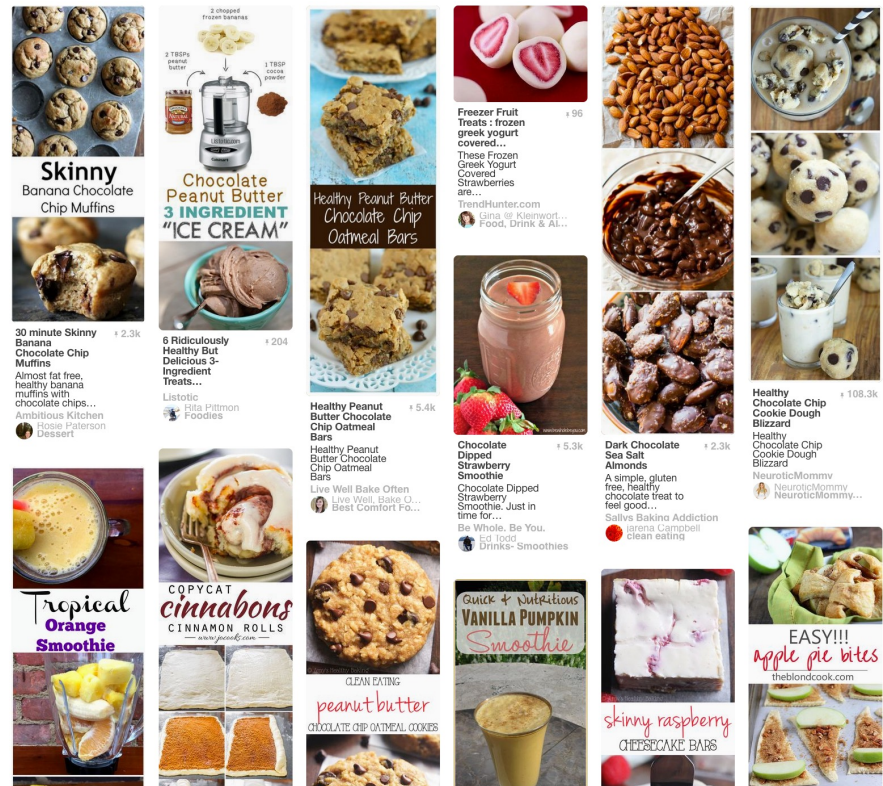


**Healthy Chocolate Peanut Butter Chips Muffins** † 119  
 Healthy Chocolate Peanut Butter Chip Muffins made with greek...  
 The First Year  
 Katie - You Brew ... Healthy Recipes



**The Ultimate Healthy Soft & Chewy Chocolate Chip Cookies** † 221  
 The ULTIMATE Healthy Chocolate Chip Cookies -- so buttery...  
 Amv's Healthy Baking  
 Robin Guertin healthy cooking

## Output:



**Skinny Banana Chocolate Chip Muffins** † 2.3k  
 30 minute Skinny Banana Chocolate Chip Muffins Almost fat free, healthy banana muffins with chocolate chips...  
 Ambitious Kitchen  
 Deise Patterson Unstap

**Chocolate Peanut Butter 3 INGREDIENT "ICE CREAM"** † 204  
 6 Ridiculously Delicious 3-Ingredient Treats...  
 Listotic  
 Hilda Pittmon Foodies

**Healthy Peanut Butter Chocolate Chip Oatmeal Bars** † 5.4k  
 Healthy Peanut Butter Chocolate Chip Oatmeal Bars  
 Live Well Bake Often  
 Best Comfort Fo...

**Chocolate Strawberry Smoothie** † 5.3k  
 Chocolate Dipped Strawberry Smoothie. Just in time for...  
 Be Whole. Be You.  
 Ed Joda Drinks' Smoothies

**Dark Chocolate Sea Salt Almonds** † 2.3k  
 A simple, gluten free, healthy chocolate treat to feel good...  
 sains Baking Addiction  
 Jarena Campbell clean eating

**Healthy Chocolate Chip Cookie Dough Blizzard** † 108.3k  
 Healthy Chocolate Chip Cookie Dough Blizzard  
 NeuroticMommy  
 NeuroticMommy... NeuroticMommy...

**Tropical Orange Smoothie**

**Copycat Cinnabons Cinnamon Rolls**

**peanut butter chocolate chip oatmeal cookies**

**Vanilla Pumpkin Smoothie**

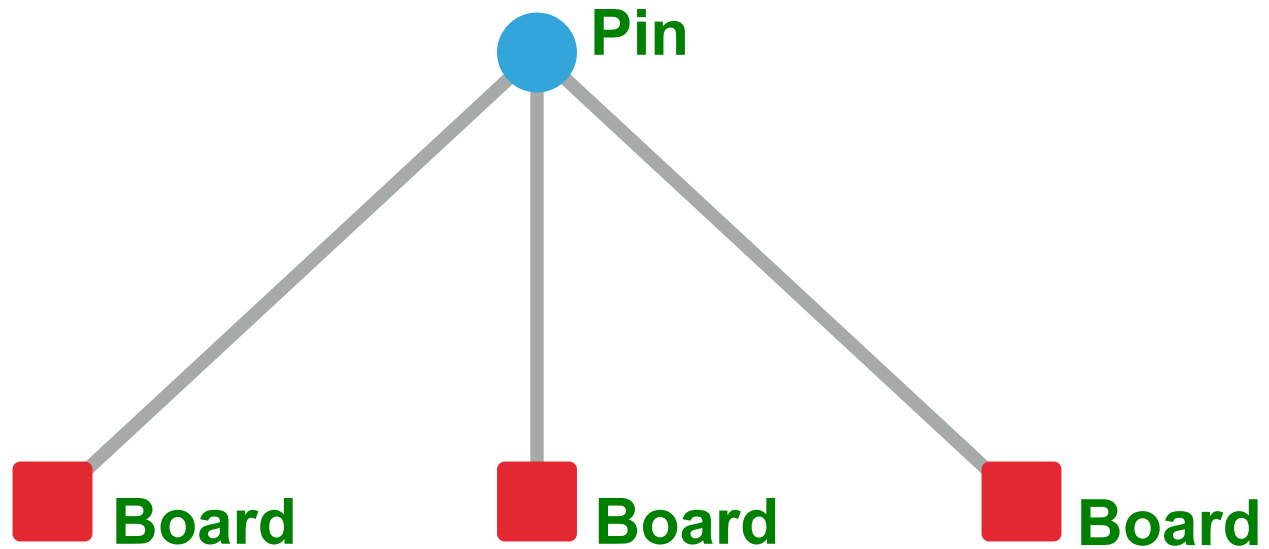
**skinny raspberry cheesecake bars**

**EASY!!! apple pie bites**

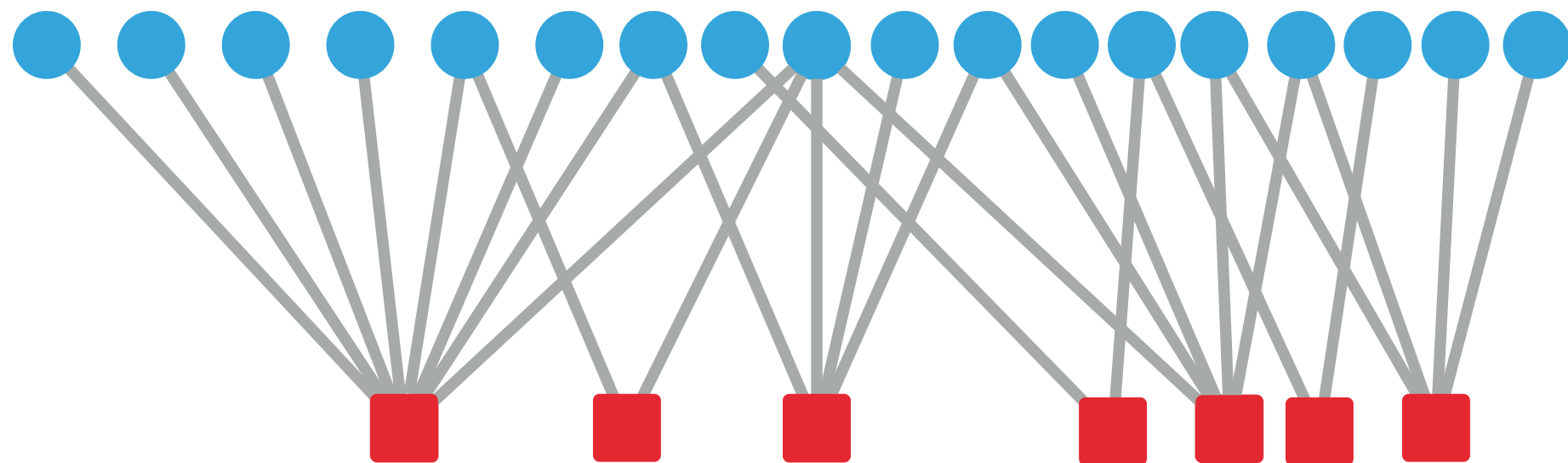
# Pinterest is a Giant Bipartite Graph



# Bipartite Pin And Board Graph

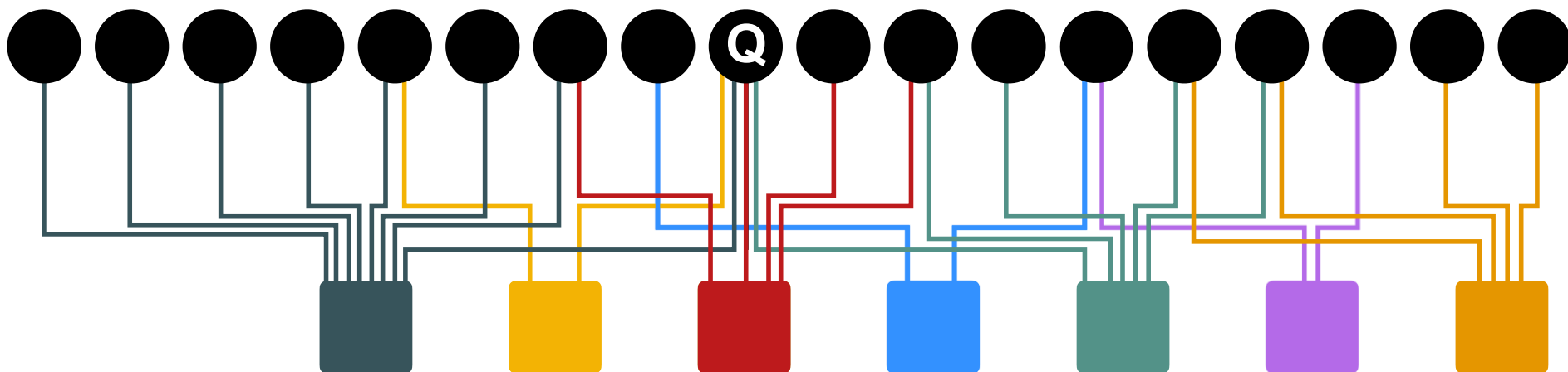


# Bipartite Pin And Board Graph



# 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:**

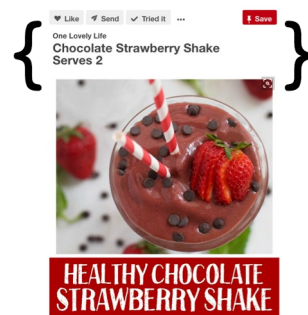


# Pixie Random Walk Algorithm

## ■ Proximity to query node(s) $Q$ :

ALPHA = 0.5

QUERY\_NODES =



```
pin_node = QUERY_NODES.sample_by_weight()
```

```
for i in range(N_STEPS):
```

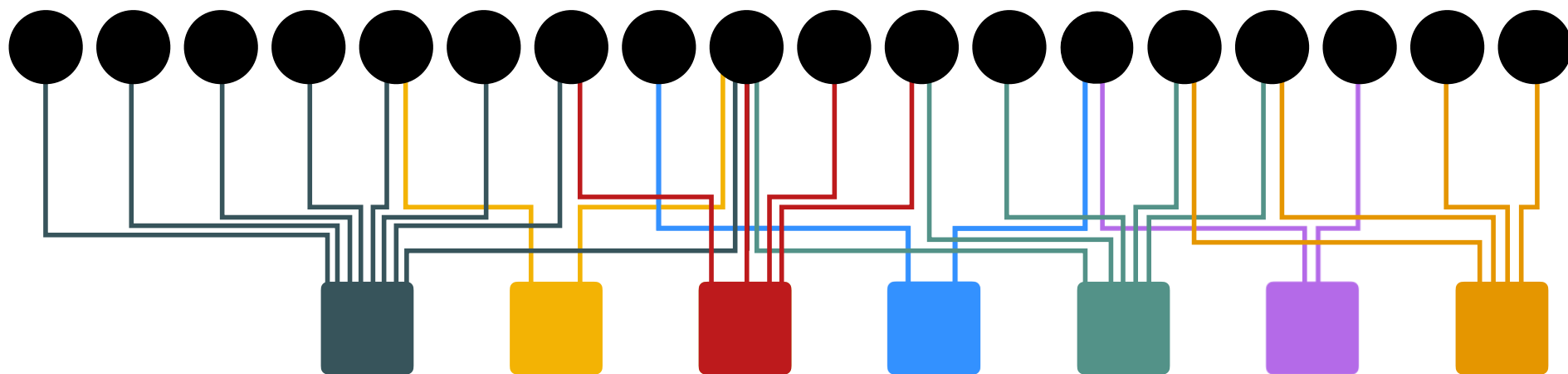
```
    board_node = pin_node.get_random_neighbor()
```

```
    pin_node = board_node.get_random_neighbor()
```

```
    pin_node.visit_count += 1
```

```
    if random() < ALPHA:
```

```
        pin_node = QUERY_NODES.sample_by_weight()
```



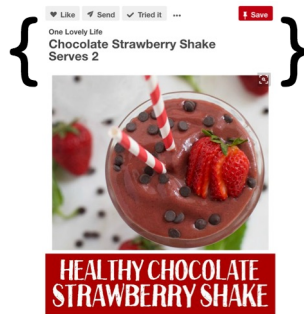


# Pixie Random Walk Algorithm

## ■ Proximity to query node(s) $Q$ :

ALPHA = 0.5

QUERY\_NODES =



```
pin_node = QUERY_NODES.sample_by_weight()
```

```
for i in range(N_STEPS):
```

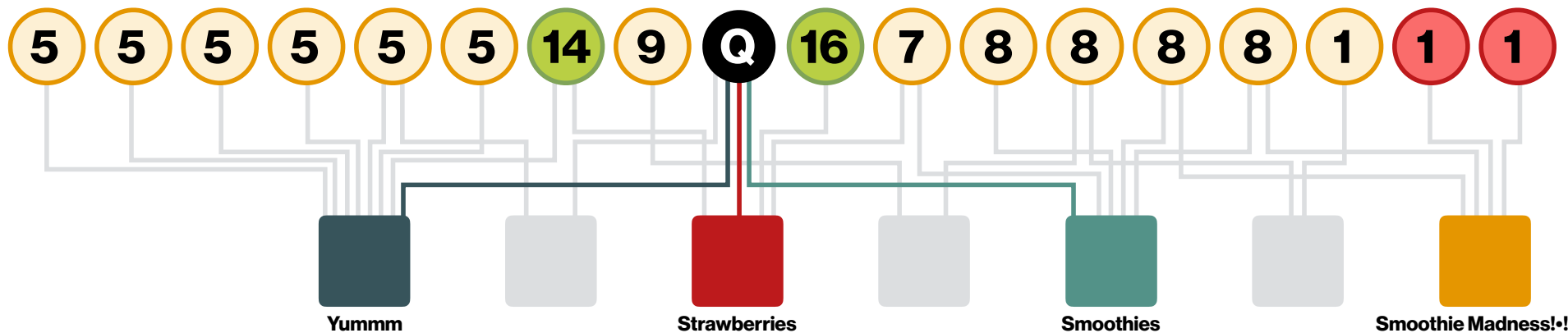
```
    board_node = pin_node.get_random_neighbor()
```

```
    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:**

- **1. Weighted edges:** The walk prefers to traverse certain edges:
  - Edges to pins in your local language
  - Personalized edge weights:
  - Pixie for different users and query pins can choose to bias edge selection dynamically based on user and edge features.
    - $\text{Weight} = \text{PersonalizedNeighbor}(E, U)$ , where  $E$  is edge and  $U$  is the user.

# Pixie Recommendations

## Extensions:

### ■ 2. Multiple query pins:

- Each query pin  $q$  gets a different importance  $w_q$
- Run PixieRandomWalk for each  $q$  in parallel.
- Combine visit counts.
- **Important insight:** The number of steps required to obtain meaningful visit counts depends on the query pin's degree
  - Scale the number of steps allocated to each query pin to be proportional to its degree

# Pixie Recommendations

## Extensions:

### ■ 3. Multi-hit Booster:

- For multi-pin queries we prefer recommendations related to multiple query pins  $q$ .
  - Candidates with high visit counts from multiple query pins are more relevant to the query than candidates having equally high total visit count but all coming from a single query pin.
- **Solution:** When combining visit counts use:

$$V[p] = \left( \sum_{q \in Q} \sqrt{V_q[p]} \right)^2$$

Note that when a candidate pin  $p$  is visited by walks from only a single query pin  $q$  then the count is unchanged. However, if the candidate pin is visited from multiple query pins, then the count is boosted.

# Pixie Recommendations

## Extensions:

### ■ 4. Early Stopping:

- Insight: We only care about top-1k most visited pins.
- So, we don't need to walk a fixed big number of steps
- We just walk until 1k-th most visited 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  $\sim 100k$  steps of the random walk)
- Single machine can run 1,500 walks in parallel (1500 recommendation requests per second).
- Fit entire graph in RAM of a single machine (17B edges, 3B nodes)
- Can scale it by just adding more machines

To learn more read: <https://cs.stanford.edu/people/jure/pubs/pixie-www18.pdf>

# Recommendations@Twitter

Joint work with many Twitter folks over several years:

<http://www2013.w3c.br/proceedings/p505.pdf>

<https://www.vldb.org/pvldb/vol9/p1281-sharma.pdf>



# Recommendations@Twitter

## Who to follow

Ramnath Balasubramanian and 3 others follow



**Jiasong Sun**  
@jiasong\_sun

Software Engineer @twitter

Follow

Gilad Mishne and 5 others follow



**David Burkett**  
@david\_burkett

Doesn't usually write well in the short form, but is glad that other people do.

Follow

David Gleich and 2 others follow



**Nelly Litvak**  
@nellylitvak

Professor in Applied Mathematics at University of Twente and Eindhoven University of Technology| complex networks| novelty in education| non-fiction author

Follow

Show more >



662 961 6,219



Elon Musk liked  
**DirtyTesla** Starlink Plz @Dirt... · 8h ...  
If you experience any kind of traffic like this, you need Autopilot. It makes the experience relaxing instead of stressful.



Elon Musk and 2 others

58 61 1,317



Mekka Okereke liked  
**Andrea Pitzer** @andrapitzer · 3h ...  
I'm skeptical of all politicians, because it's so much easier to say things than to do them. But it's such a relief that we now have a president who isn't actively using every public appearance to foment hatred and intolerance. It may be a low bar, but it still feels like a gift.

6 20 240

Show this thread

**Serena Williams** ✓  
@serenawilliams

Following

### Suggested

**Venus Williams** ✓  
@Venuseswilliams

Tennis player, big sister, grown up girl. Double Tap! ❤️ Be Well ❤️ #CoachVenus @elevenbyvenus workouts @ link in bio

Follow

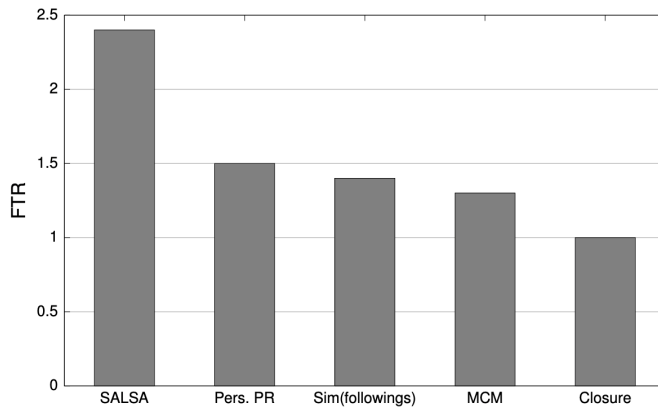
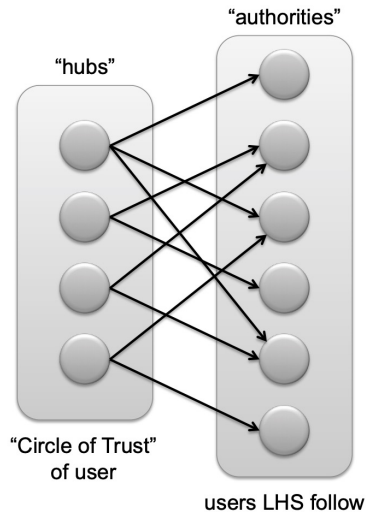
**Rafa Nadal** ✓  
@RafaelNadal

Tennis player

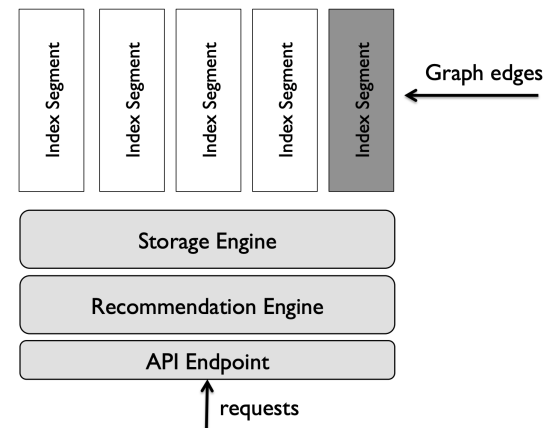
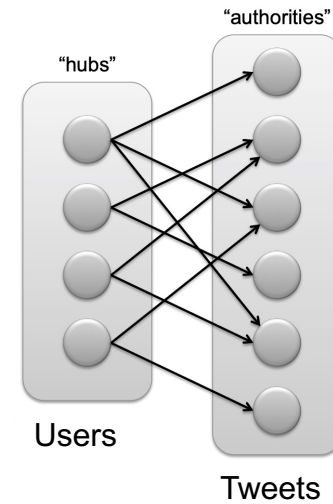
Follow

# SALSA for Recommendations

## User Recs



## Content Recs



# TrustRank: Combating Spam on the Web

# Web Search

- **How early search engines answer queries:**
  - Crawl the Web to collect pages
  - Build inverted index (word -> list of pages)
  - Given a search query, get intersection of pages containing those words
  - Rank pages and respond to user
- **Early page ranking:**
  - Based on a notion of “importance”
  - **First search engines considered:**
    - (1) Number of times query words appeared
    - (2) Prominence of word position, e.g. title, header

# What is Web Spam?

- **People abused it in two ways:**
  - Term Spam
  - Link Spam
- **What is spamming?**
  - Any deliberate action to boost a web page's position in search engine results, incommensurate with the page's real value
- **What is a spam?**
  - Web pages that are the result of spamming
- This is a very broad definition
  - **SEO** (search engine optimization) industry might disagree!
- Approximately **10-15%** of web pages are spam

# First Spammers: Term Spam

- 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

- **Google's solution:**
  - Believe what people say about you, rather than what you say about yourself
  - Use words in the anchor text and its surrounding text
- **Measure “importance” of those Web pages via PageRank**



# Why Does It Work?

- **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 Does It NOT Work?



**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 farm:** A collection of pages whose purpose is to increase the page rank of a certain page(s)
- **Link spam:**
  - Create link structures that boost PageRank of a particular page



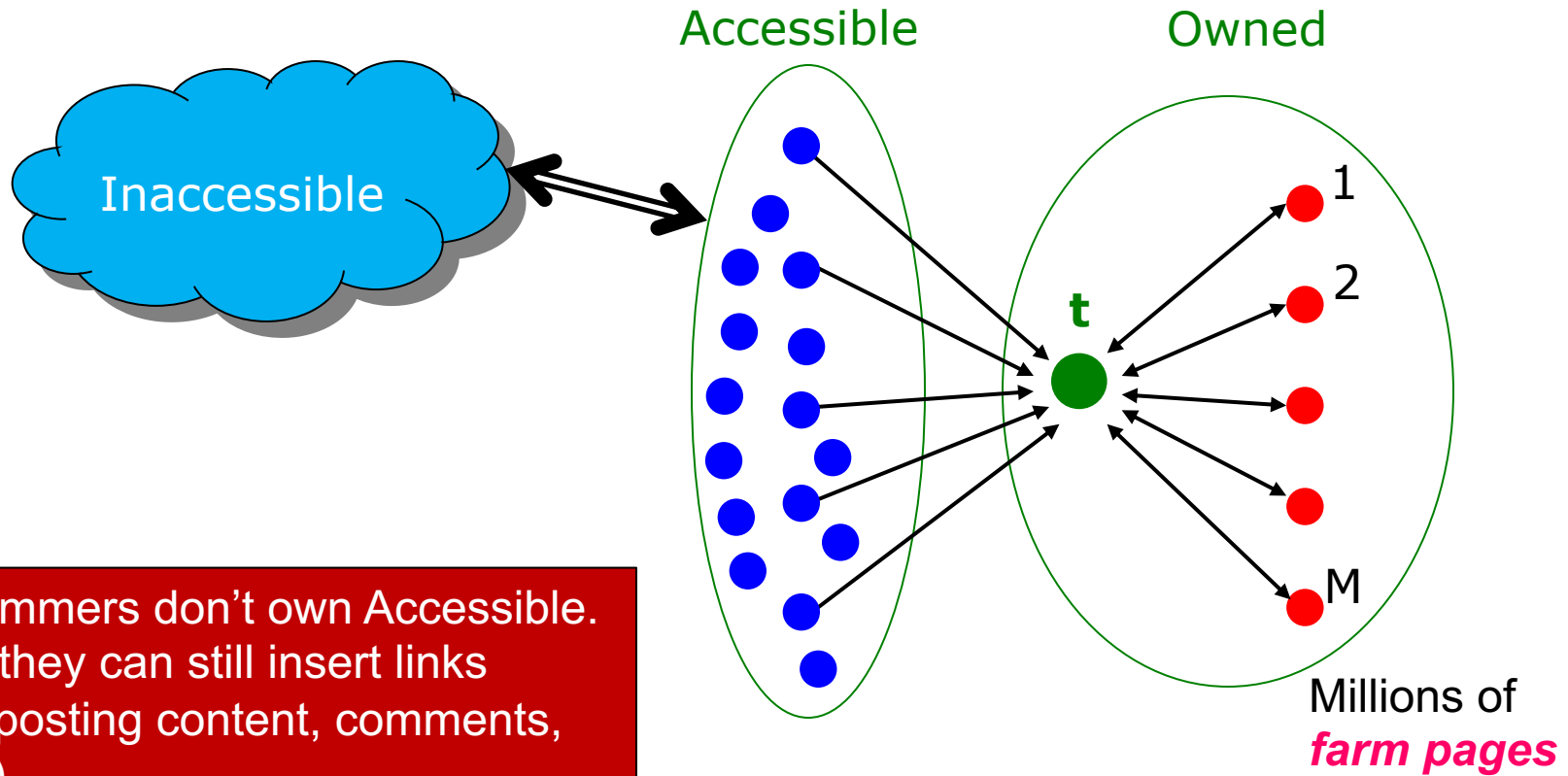
# Link Farms

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

# 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

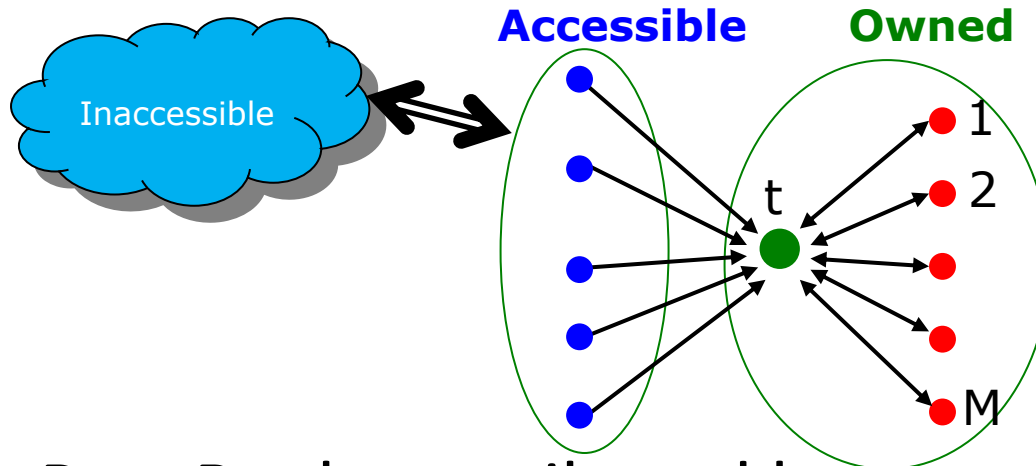
# Link Farms



Spammers don't own Accessible. But they can still insert links (by posting content, comments, etc.)

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 “owned” page =  $\frac{\beta y}{M} + \frac{1-\beta}{N}$

- $$y = x + \beta M \left[ \frac{\beta y}{M} + \frac{1-\beta}{N} \right] + \frac{1-\beta}{N}$$

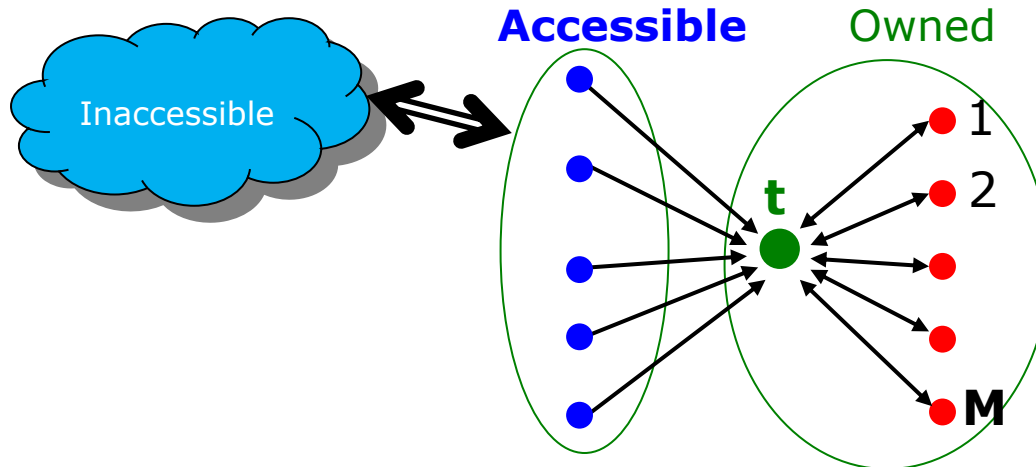
$$= x + \beta^2 y + \frac{\beta(1-\beta)M}{N} + \frac{1-\beta}{N}$$

Very small; ignore  
 Now we solve for  $y$

- $$y = \frac{x}{1-\beta^2} + c \frac{M}{N} \quad \text{where } c = \frac{\beta}{1+\beta}$$



# Analysis



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

- $y = \frac{x}{1-\beta^2} + c \frac{M}{N}$  where  $c = \frac{\beta}{1+\beta}$
- For  $\beta = 0.85$ :

- $y = 3.6 x + 0.46 \frac{M}{N}$

- 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 Link Spam

- **There are two approaches to detect and remove link spam:**
  - 1. Detection and blacklisting of structures that look like spam farms**
    - One page links to a very large number of pages, each of which links back to it.
    - Leads to another war – hiding and detecting spam farms
  - 2. TrustRank = topic-specific PageRank with a teleport set of trusted pages**
    - **Example:** .edu domains, .gov domains
    - similar domains for non-US websites

# TrustRank: Idea

- **TrustRank is topic-sensitive PageRank**
  - Topic = a set of pages believed to be trustworthy
  - The idea is that it is rare for a “good” page to point to a “bad” (spam) page
- To develop a suitable teleport set:
  1. Sample a set of **seed pages** from the web
  2. 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**

# Trust Propagation: Simple Model

- **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
  - Every time it is multiplied by  $\beta$

- **Trust splitting:**

- The larger the number of out-links from a page, the less scrutiny the page's author gives to 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 to make all good pages reachable from trusted set by short paths
    - **So the trusted set must be large**



# Approaches to Picking Seed Set

Suppose we want to pick a seed set of  $k$  pages

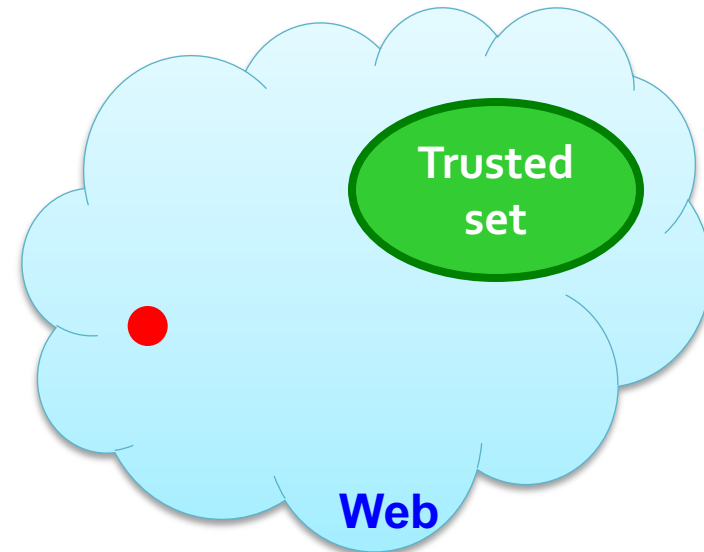
**How to do that?**

- **(1) PageRank:**
  - Pick the top  $k$  pages by PageRank
  - Theory is that bad pages can't get really high ranks
- **(2) Use trusted domains** whose membership is controlled, like .edu, .mil, .gov

# SpamMass

# Spam Mass

- The **TrustRank** model, propagates trust
- **SpamMass provides a 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



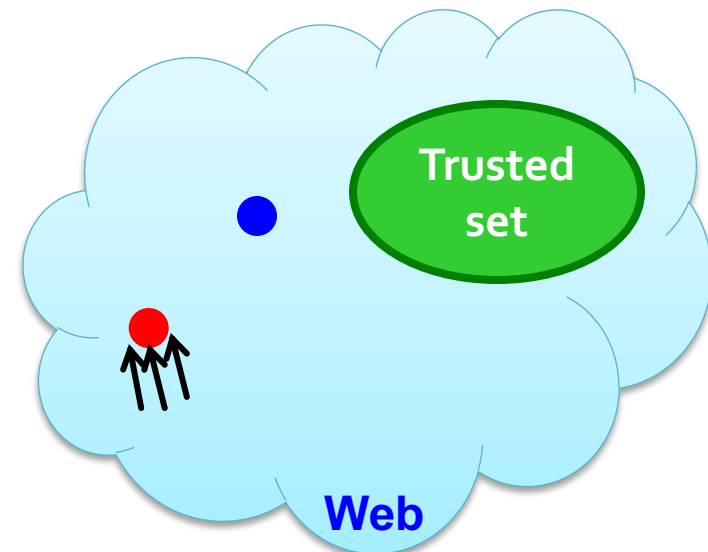
# Spam Mass Estimation

## 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 and thus removed from search engine index



# Summary of Today's lecture

- Topic specific PageRank
  - Custom teleportation vector
- Random Walk with Restarts
  - Recommendations
- Spam farming
- TrustRank and Spam Mass estimation