

# Combating Web Spam with TrustRank

## **Authors:**

Gyöngyi, Garcia-Molina, and Pederson

## **Published in:**

Proceedings of the 30th VLDB Conference

## **Year:**

2004

**Presentation by:** Rebecca Wills

**Date:** April 4, 2007

## Questions we will answer today include...

- Why should we care about this paper?
- What is TrustRank?
- Is TrustRank mathematically sound?

# Why should we care about this paper?

The screenshot shows a Netscape browser window with the address bar containing the URL <http://www.google.com/search?hl=en&q=trustrank&btnG=Google+Search>. The search results page displays the Google logo and the search term 'trustrank'. The search results are categorized under 'Web' and show the following results:

- TrustRank Algorithm : SEO Book.com**  
A buddy of mine pointed me to a white paper by Zoltan Gyongyi, Hector Garcia-Molina, & Jan Pederson about a concept called **TrustRank**(PDF). ...  
[www.seobook.com/archives/000661.shtml](http://www.seobook.com/archives/000661.shtml) - 27k - [Cached](#) - [Similar pages](#)
- TrustRank - Wikipedia, the free encyclopedia**  
**TrustRank** is a link analysis technique for semi-automatically separating useful webpages from spam. (Gyöngyi et al. 2004) ...  
[en.wikipedia.org/wiki/TrustRank](http://en.wikipedia.org/wiki/TrustRank) - 17k - [Cached](#) - [Similar pages](#)
- [PDF] Combating Web Spam with TrustRank**  
File Format: PDF/Adobe Acrobat - [View as HTML](#)  
**TrustRank** algorithm "refines" the original scores given by ... on a real web graph the **TrustRank** algorithm is still able to ...  
[www.vidb.org/conf/2004/RS15P3.PDF](http://www.vidb.org/conf/2004/RS15P3.PDF) - [Similar pages](#)
- Slashdot | Google TrustRank**  
Philipp Lensen writes "Google registered a trademark for the word "**TrustRank**", as Search Engine Watch reveals. Is this a sign we can expect a follow-up to ...  
[slashdot.org/article.pl?sid=05/04/26/1157212&from=rss](http://slashdot.org/article.pl?sid=05/04/26/1157212&from=rss) - 115k - Mar 28, 2007 - [Cached](#) - [Similar pages](#)
- trustrank: See what people are saying right now on Technorati**  
Everything in the known universe tagged **trustrank**. Everything; Blog Posts - Videos ... How To Make Google **Trustrank** Trust You. No one has claimed this blog ...  
[technorati.com/tag/trustrank](http://technorati.com/tag/trustrank) - 15k - [Cached](#) - [Similar pages](#)

On the right side of the page, there are sponsored links:

- Google Sitemaps**  
Which URL on your site has the highest PageRank? Find out more.  
[www.google.com/webmasters/sitemaps](http://www.google.com/webmasters/sitemaps)
- Track your Rankings**  
All-in-one SEO Tools for DIY SEO  
Free 2-week Trial  
[www.SoloSEO.com](http://www.SoloSEO.com)
- Trustrank**  
Learn about the **TrustRank** algorithm. Free tips & info.  
[SeoBook.com](http://SeoBook.com)

The search results summary at the top right of the page is circled in red: "Results 1 - 10 of about 489,000 for trustrank. (0.05 seconds)".

At the bottom of the browser window, the Windows taskbar is visible, showing the start button and several open applications: MA591R, Class\_Project..., Adobe Acrob..., Microsoft Pow..., Netscape Tool..., and trustrank - Go... The system clock shows 8:08 PM.

Image captured:  
week of March 25 - 31

# Why should we care about this paper?

Netscape Internet Service HOME E-MAIL Search Go MY ACCOUNT OPTIONS HELP SIGN OFF

trustrank - Google Scholar - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://scholar.google.com/scholar?q=trustrank&hl=en&lr=&btnG=Search

Getting Started Latest Headlines

Google Scholar BETA Web Images Video News Maps more »

trustrank Search Advanced Scholar Search Scholar Preferences Scholar Help

Scholar All articles Recent articles Results 1 - 10 of about 115 for trustrank. (0.04 seconds)

All Results

[Z Gyongyi](#)  
[H Garcia-Molin...](#)  
[J Pedersen](#)  
[B Wu](#)  
[R Guha](#)

[Combating web spam with TrustRank - group of 18 »](#)  
Z Gyongyi, H Garcia-Molina, J Pedersen - Proceedings of the 30th International Conference on Very ..., 2004 - nblavoie.com  
Page 1. Combating Web Spam with TrustRank ... 4. We introduce the TrustRank algorithm for determin- ing the likelihood that pages are reputable. ...  
Cited by 76 - [Related Articles](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

[Topical TrustRank: using topicality to combat web spam - group of 2 »](#)  
B Wu, V Goel, BD Davison - Proceedings of the 15th international conference on World ..., 2006 - portal.acm.org  
Page 1. Topical TrustRank: Using Topicality to Combat Web Spam Baoning ... TrustRank is a recent algorithm that can combat web spam. However ...  
Cited by 4 - [Related Articles](#) - [Web Search](#)

[CITATION] Combating web spam with TrustRank  
H García-Molina, Z Gyöngyi, J Pedersen - Proceedings of the Thirtieth International Conference on ..., 2004  
Cited by 3 - [Related Articles](#) - [Web Search](#)

[CITATION] Combating spam with trustrank  
Z Gyongyi, H Garcia-Molina, J Pederson - n Proceedings of the 30th International Conference on Very ..., 2004  
Cited by 2 - [Related Articles](#) - [Web Search](#)

[CITATION] Combating Web Spam with TrustRank  
Z Gyngyi, H Garcia-Molina, J Pedersen - International Conference on Very Large Data Bases  
Cited by 2 - [Related Articles](#) - [Web Search](#)

[CITATION] Combating web spam with trustrank  
Z Gyöngyi, H Garcia-Molina, J Pedersen - Proceedings of the 30th International VLDB Conference, 2004  
Cited by 1 - [Related Articles](#) - [Web Search](#)

Transferring data from scholar.google.com...

start MA591R Class\_Project... Adobe Acrob... Microsoft Pow... Netscape Tool... trustrank - Go... 8:13 PM

Image captured:  
week of March 25 - 31

## Why should we care about this paper?

Google registered the trademark for “TrustRank” on March 16, 2005.

The algorithm receives human assistance.

The authors state, “We believe that our work is a first attempt at formalizing the problem and at introducing a comprehensive solution to assist in the detection of Web spam.”

## What is Web spam?

The term refers to hyperlinked webpages that are created to mislead search engines.

Example 1: webpages containing numerous words having nothing to do with the webpage using text invisible to humans but observed by search engines

Example 2: webpages receiving links from numerous other real or phony webpages for the sole purpose of increasing PageRank

## Research Goal of the Authors

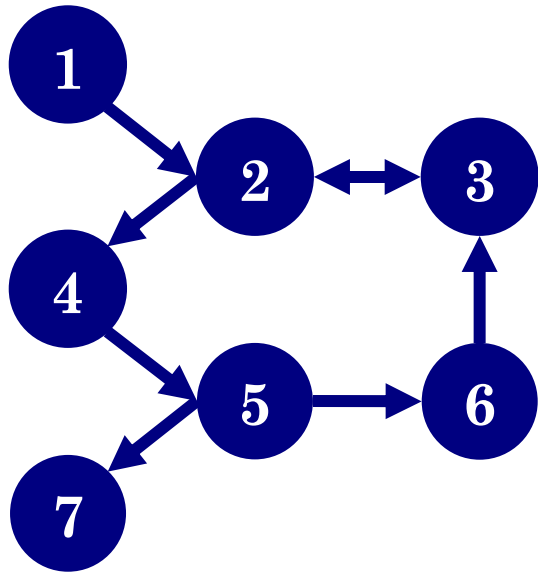
“Our research goal is to assist the human experts who detect web spam. In particular, we want to identify pages and sites that are likely to be spam or that are likely to be reputable.”

## What is TrustRank?

An algorithm that creates a personalization vector to be used in the PageRank computation for the purpose of combating Web spamming



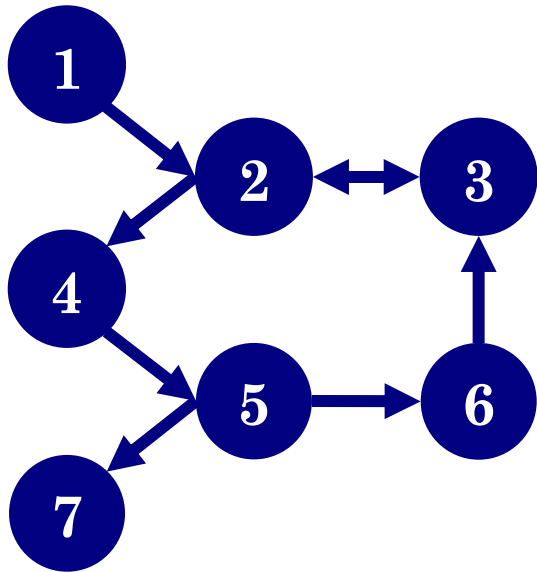
# Example Web Graph



$$V = \{1, 2, 3, 4, 5, 6, 7\}$$

$$E = \{(1,2), (2,3), (2,4), (3,2), (4,5), (5,6), (5,7), (6,3)\}$$

# Example Web Graph



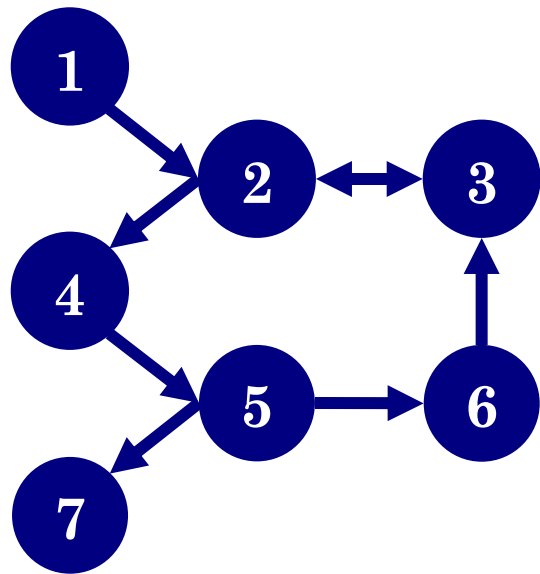
$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Basic PageRank

$$r^T = \alpha (r^T H) + \frac{(1-\alpha)}{n} e^T$$

$$0 \leq \alpha < 1, \quad n = |V|, \quad \text{and} \quad e^T = (1 \quad 1 \quad \dots \quad 1)$$

# Example Web Graph



$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Basic PageRank

$$\mathbf{r}^T = \alpha \left( \mathbf{r}^T \mathbf{H} \right) + \frac{(1-\alpha)}{n} \mathbf{e}^T$$

Note: The authors do not directly address the dangling node issue. They do mention the “biased PageRank” definition, where  $\mathbf{v}$  is any probability vector, but they compare their results to the “regular PageRank” definition.

# Why is the definition okay?

Biased PageRank

$$\mathbf{r}^T = \alpha (\mathbf{r}^T \mathbf{H}) + (1 - \alpha) \mathbf{v}^T$$

Class Definition of PageRank (with dangling node fix  $\mathbf{v}^T$ )

$$\begin{aligned} \pi^T &= \pi^T \left( \alpha [\mathbf{H} + \mathbf{a} \mathbf{v}^T] + (1 - \alpha) \mathbf{e} \mathbf{v}^T \right) \\ &= \alpha (\pi^T \mathbf{H}) + \underbrace{\left[ \alpha (\pi^T \mathbf{a}) + (1 - \alpha) \right]}_{\text{Scalar}} \mathbf{v}^T \end{aligned}$$

$$\text{So, } \pi^T (\mathbf{I} - \alpha \mathbf{H}) = \left[ \alpha (\pi^T \mathbf{a}) + (1 - \alpha) \right] \mathbf{v}^T$$

$$\Rightarrow \pi^T = \left[ \alpha (\pi^T \mathbf{a}) + (1 - \alpha) \right] \mathbf{v}^T (\mathbf{I} - \alpha \mathbf{H})^{-1}$$

## Why is the definition okay?

Biased PageRank

$$\mathbf{r}^T = \alpha (\mathbf{r}^T \mathbf{H}) + (1 - \alpha) \mathbf{v}^T$$

Class Definition of PageRank (with dangling node fix  $\mathbf{v}^T$ )

$$\text{So, } \pi^T (\mathbf{I} - \alpha \mathbf{H}) = \left[ \alpha (\pi^T \mathbf{a}) + (1 - \alpha) \right] \mathbf{v}^T$$

$$\Rightarrow \pi^T = \left[ \alpha (\pi^T \mathbf{a}) + (1 - \alpha) \right] \mathbf{v}^T (\mathbf{I} - \alpha \mathbf{H})^{-1}$$

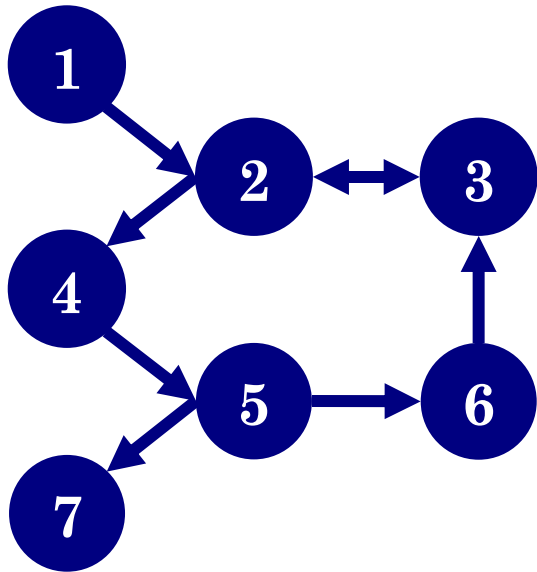
---

$$\mathbf{x}^T (\mathbf{I} - \alpha \mathbf{H}) = \beta \mathbf{v}^T, \quad \pi^T = \frac{\mathbf{x}^T}{\mathbf{x}^T \mathbf{e}}, \quad \beta > 0$$

produces the PageRank vector.

[Langville & Meyer, 2006, Thm. 7.3.1, pages 73 - 74]

# Example Web Graph



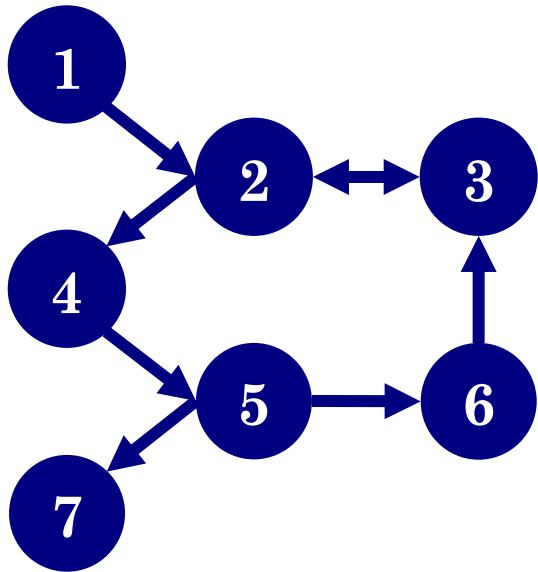
$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

TrustRank

$$r^T = \alpha (r^T H) + (1 - \alpha) v^T$$

where  $v^T$  is formed to combat Web spam

# Overall Idea



TrustRank

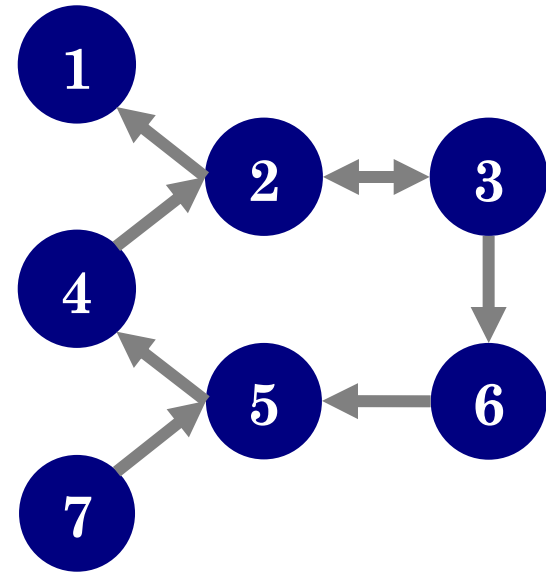
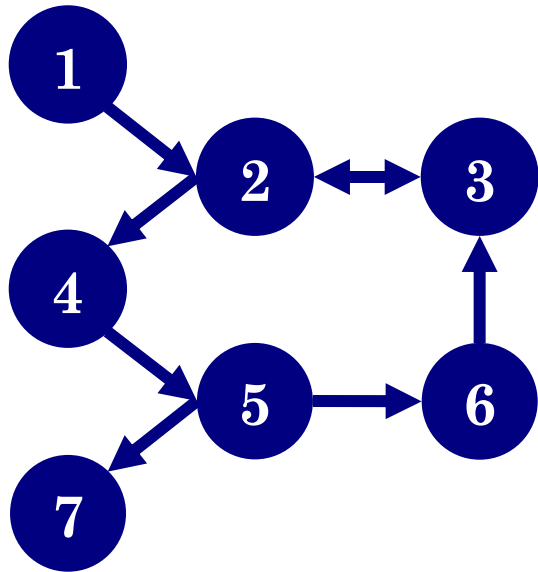
$$r^T = \alpha (r^T H) + (1 - \alpha) v^T$$

Step 1: Select a small “seed set” of webpages.

Step 2: Identify good webpages from the “seed set”.

Step 3: Create personalization vector based on identification of good webpages.

Step 1: Select a small “seed set” of webpages.



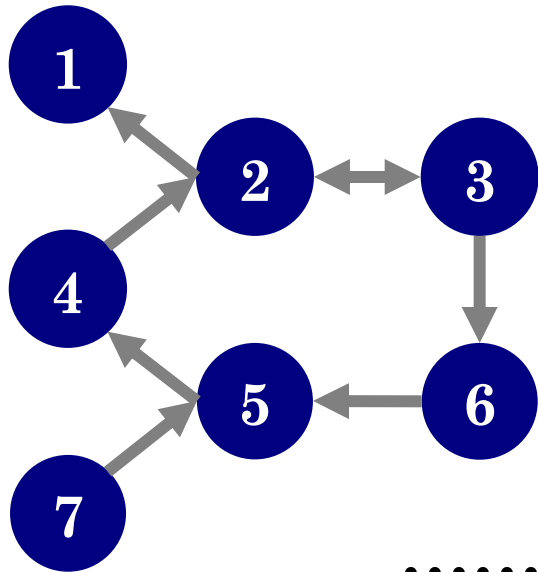
$$V = \{1, 2, 3, 4, 5, 6, 7\}$$

$$E = \{(1,2), (2,3), (2,4), (3,2), (4,5), (5,6), (5,7), (6,3)\}$$

$$E^{-1} = \{(2,1), (3,2), (4,2), (2,3), (5,4), (6,5), (7,5), (3,6)\}$$



Step 1: Select a small “seed set” of webpages.



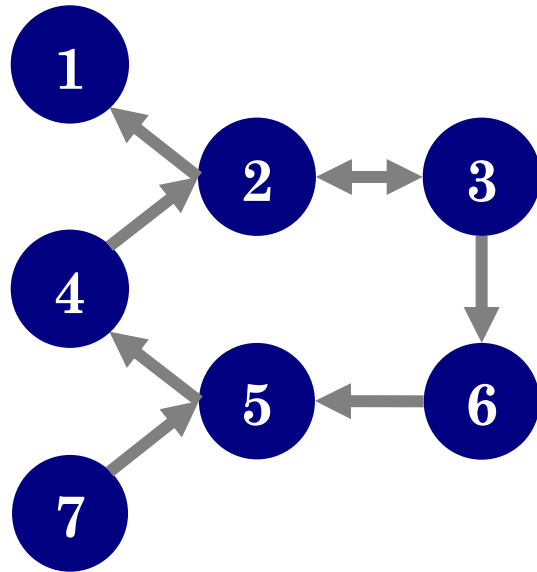
$$U = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Inverse PageRank

$$s^T = \alpha (s^T U) + \frac{(1-\alpha)}{n} e^T$$

Note: The authors emphasize that Inverse PageRank (named based on  $E^{-1}$ ) works well in practice.

## Step 1: Select a small “seed set” of webpages.



Function: **SelectSeed**

Initial iterate:  $\mathbf{s}_0^T = \mathbf{e}^T$

While:  $k \leq M$

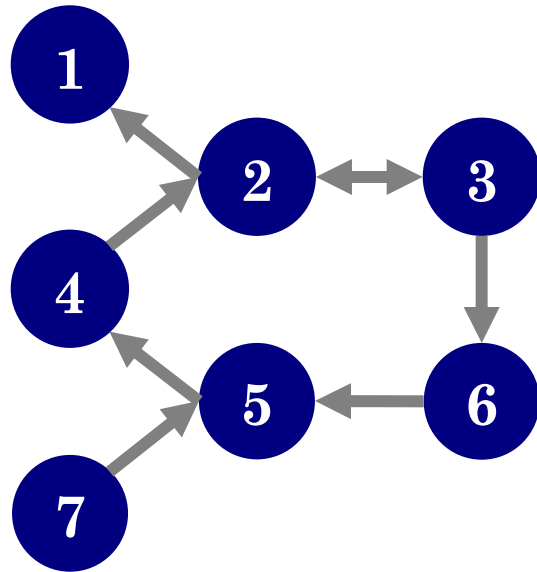
Do:  $\mathbf{s}_k^T = \alpha (\mathbf{s}_{k-1}^T \mathbf{U}) + \frac{(1-\alpha)}{n} \mathbf{e}^T, k \geq 1$

This is based on the belief that trust flows out of good seed webpages.

It gives preference to webpages from which many other webpages can be reached.

See Maple file for implementation of SelectSeed for this example.

## Step 1: Select a small “seed set” of webpages.



Function: **SelectSeed**

Initial iterate:  $\mathbf{s}_0^T = \mathbf{e}^T$

While:  $k \leq M$

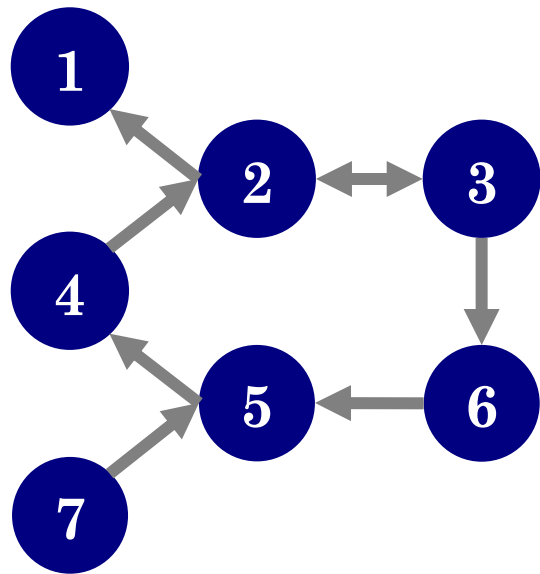
Do:  $\mathbf{s}_k^T = \alpha (\mathbf{s}_{k-1}^T \mathbf{U}) + \frac{(1-\alpha)}{n} \mathbf{e}^T, k \geq 1$

**Note:** This algorithm is the Jacobi Method applied to  $r^T(I - \alpha U) = (1-\alpha)/n \mathbf{e}^T$ .

The diagonal part of  $(I - \alpha U)$  is  $I$ , and the sum of the upper and lower triangular parts is  $\alpha U$ .

See Maple file for implementation of SelectSeed for this example.

## Step 1: Select a small “seed set” of webpages.



Function: **SelectSeed**

Initial iterate:  $\mathbf{s}_0^T = \mathbf{e}^T$

While:  $k \leq M$

Do:  $\mathbf{s}_k^T = \alpha (\mathbf{s}_{k-1}^T \mathbf{U}) + \frac{(1-\alpha)}{n} \mathbf{e}^T, k \geq 1$

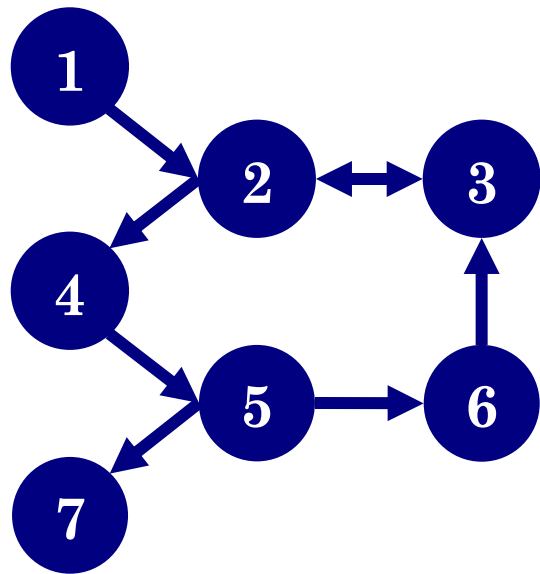
For  $\alpha = 0.85$  and  $M = 20$ , we obtain

$$\mathbf{s}_{20}^T \approx (0.08 \quad 0.14 \quad 0.08 \quad 0.10 \quad 0.09 \quad 0.06 \quad 0.02).$$

Suppose we want to check the top 3 webpages (in blue above).

Then, our seed set is  $S = \{2, 4, 5\}$ .

## Step 2: Identify good webpages from “seed set”.



Back to the original graph

$$V = \{1, 2, 3, 4, 5, 6, 7\}.$$

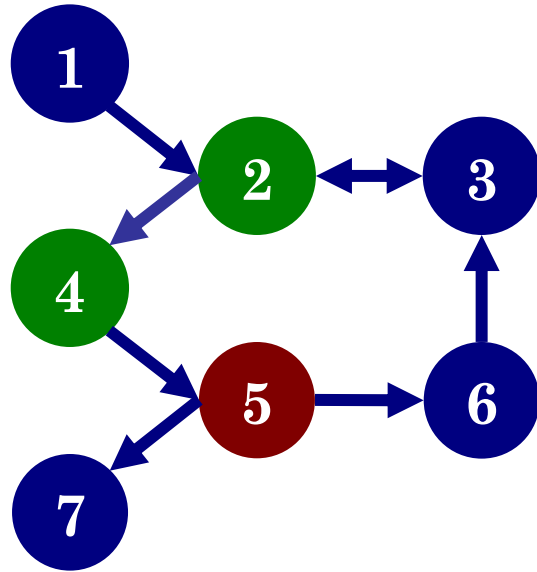
$$S = \{2, 4, 5\}.$$

Oracle function:

$$O(i) = \begin{cases} 1, & \text{if webpage } i \text{ is good} \\ 0, & \text{if webpage } i \text{ is bad.} \end{cases}$$

This is the step requiring human involvement, and it is, to some extent, subjective.

## Step 2: Identify good webpages from “seed set”.



$$V = \{1, 2, 3, 4, 5, 6, 7\}.$$

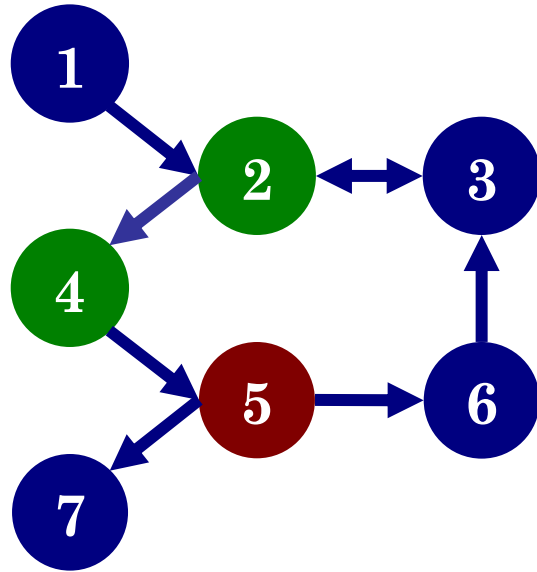
$$S = \{2, 4, 5\}.$$

Oracle function:

$$O(i) = \begin{cases} 1, & \text{if webpage } i \text{ is good} \\ 0, & \text{if webpage } i \text{ is bad.} \end{cases}$$

$$S^+ = \{2, 4\} \text{ and } S^- = \{5\}.$$

Step 3: Create personalization vector based on identification of good webpages.

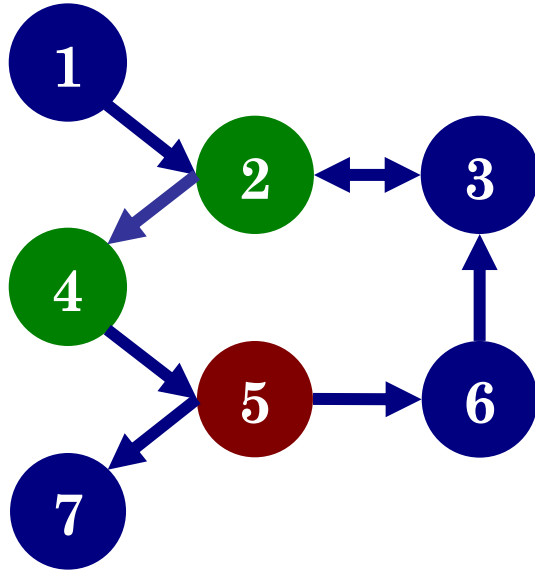


$$S^+ = \{2, 4\} \text{ and } S^- = \{5\}.$$

$$\mathbf{v}^T = \left( 0 \quad \frac{1}{2} \quad 0 \quad \frac{1}{2} \quad 0 \quad 0 \quad 0 \right)$$

**Comment:** I think it's interesting that the authors go through the trouble of making the personalization vector a probability vector even though the PageRank vector will not be a probability vector. Also, they do not use a probability vector to initialize the Inverse PageRank algorithm.

# Compute TrustRank



Function: **TrustRank**

Initial iterate:  $\mathbf{r}_0^T = \mathbf{v}^T$

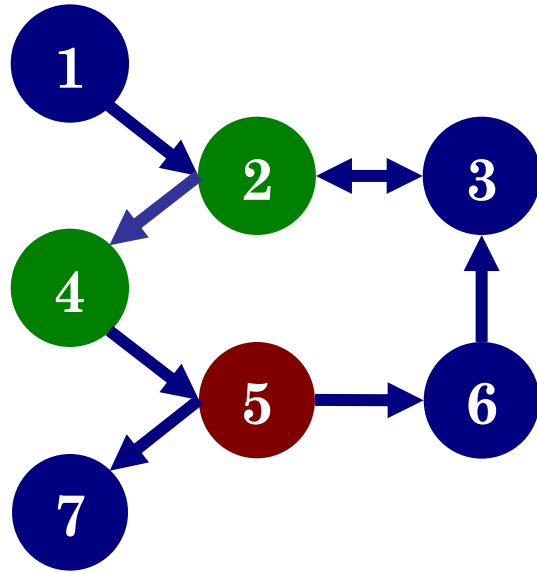
While:  $k \leq M$

Do:  $\mathbf{r}_k^T = \alpha (\mathbf{r}_{k-1}^T \mathbf{H}) + (1 - \alpha) \mathbf{v}^T, k \geq 1$

Note: This  $M$  and  $\alpha$  can be different from the ones used for Inverse PageRank.



# Compute TrustRank



Function: **TrustRank**

Initial iterate:  $r_0^T = v^T$

While:  $k \leq M$

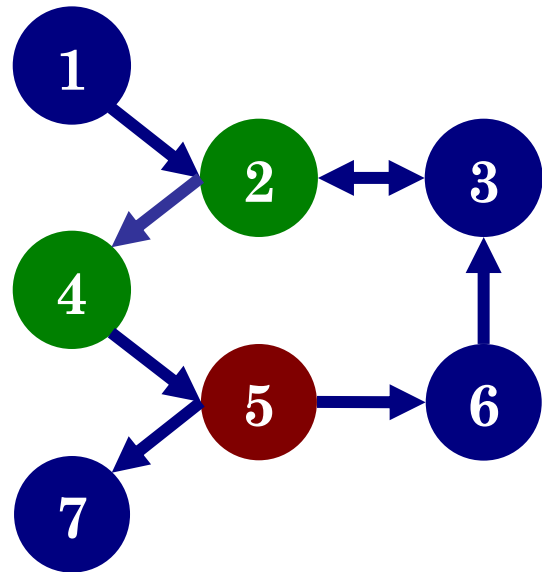
Do:  $r_k^T = \alpha (r_{k-1}^T H) + (1 - \alpha) v^T, k \geq 1$

**Note:** This algorithm is the Jacobi Method applied to  $r^T(I - \alpha H) = (1 - \alpha)v^T$ .

The diagonal part of  $(I - \alpha H)$  is  $I$ , and the sum of the upper and lower triangular parts is  $\alpha H$ .

See Maple file for implementation of TrustRank for this example.

# Compute TrustRank



Function: TrustRank

Initial iterate:  $r_0^T = v^T$

While:  $k \leq M$

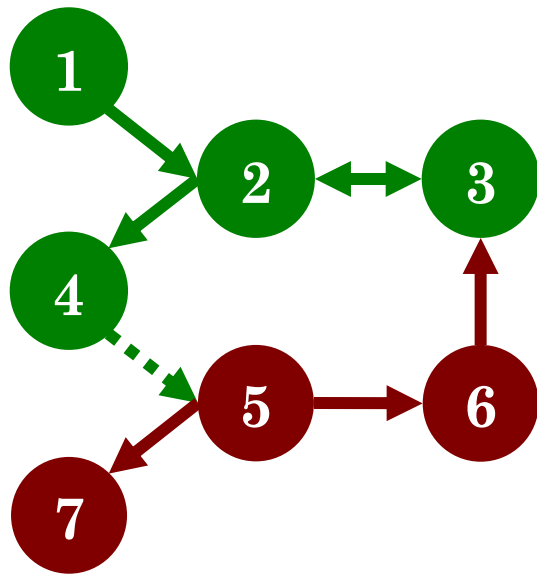
Do:  $r_k^T = \alpha (r_{k-1}^T H) + (1 - \alpha) v^T, k \geq 1$

For  $\alpha = 0.85$  and  $M = 20$ , we obtain

$$r_{20}^T \approx (0 \quad 0.18 \quad 0.12 \quad 0.15 \quad 0.13 \quad 0.05 \quad 0.05).$$

See Maple file for implementation of TrustRank for this example.

# Compute TrustRank



Interestingly, for the whole Web graph, the authors identified:

**Good webpages**,  $V^+ = \{1, 2, 3, 4\}$ , and  
**Bad webpages**,  $V^- = \{5, 6, 7\}$ .

The “Basic PageRank” algorithm ranked **good webpage 3** higher than **bad webpage 5**, but the TrustRank algorithm did not. *(Perhaps, they should have identified good and bad webpages differently for the example.)*

For  $\alpha = 0.85$  and  $M = 20$ , we obtain

TrustRank:  $\mathbf{r}_{20}^T \approx (0 \quad 0.18 \quad 0.12 \quad 0.15 \quad 0.13 \quad 0.05 \quad 0.05)$ .

Basic PageRank:  $\mathbf{x}_{20}^T \approx (0.02 \quad 0.26 \quad 0.21 \quad 0.17 \quad 0.16 \quad 0.09 \quad 0.09)$ .

See Maple file for implementation of PageRank for this example.

# Experiments

- Web data
  - Entire AltaVista index (August 2003)
  - Site-level Web graph
    - 31 million vertices
    - 13 million without inlinks
- Seed set
  - 25,000 candidates → reduced to 7,900 → then 1,250
  - 178 selected high-quality sites
- Evaluation sample
  - 1000 manually tagged sites
- Oracle: Gyöngyi (the first author of the paper)

# Experiments

- Manual evaluation took weeks
- Compared Inverse PageRank to other options for selecting the “seed set” (such as using webpages with high Basic PageRank scores)
- Observed that websites with highest inverse PageRank scores showed a heavy bias toward spam
- Removed all websites not listed in any major web directories
- Final filter – only selected websites with a clearly identifiable authority

# Experiments

## Section 6.2: Seed Set

The authors indicate that they compare inverse PageRank to high PageRank (seed set made up of pages with highest basic PageRank scores). They state:

“We describe these experiments in [4]. Due to space limitations, here we just note that inverse PageRank turned out to be slightly better at identifying useful seed sets. Thus, for the rest of our experiments, we relied on the inverse PageRank method.”

I searched the Web and never found [4].

The above statement reminds me of Fermat’s comment about his last theorem.

# Experiments

See slide 19 from below webpage.

Each bucket represents 5% of the total PageRank score. There are 20 buckets. TrustRank buckets have the same number of websites as PageRank buckets.

Picture of graph available at:

<http://infolab.stanford.edu/~zoltan/presentations/trustrank-vldb-2004-09-02.pdf>

# Experiments

See slide 20 from below webpage.

Picture of graph available at:

<http://infolab.stanford.edu/~zoltan/presentations/trustrank-vldb-2004-09-02.pdf>



## Also discussed: Assessing Trust

### Oracle Function:

$$O(i) = \begin{cases} 1, & \text{if webpage } i \text{ is good} \\ 0, & \text{if webpage } i \text{ is bad.} \end{cases}$$

### Trust function:

For webpage  $i$ ,  $T(i) = Pr[O(i) = 1]$ .

### Threshold trust property:

For webpage  $i$ ,  $T(i) > \delta \Leftrightarrow O(i) = 1$ .

### Signal function:

$$I(T, O, i, j) = \begin{cases} 1, & T(i) \geq T(j) \text{ and } O(i) < O(j) \\ 1, & T(i) \leq T(j) \text{ and } O(i) > O(j) \\ 0, & \text{otherwise.} \end{cases}$$

# Assessing Trust

Suppose  $X \subseteq V$  with  $m$  randomly selected elements.  
Let  $P = \{(i, j) \in X \times X: i \neq j\}$ . Then,  $|P| = m(m - 1)$ .

**Pairwise Orderedness:**

$$\text{pairord}(T, O, P) = \frac{|P| - \sum_{(i,j) \in P} I(T, O, i, j)}{|P|}$$

**Precision:**

$$\text{prec}(T, O) = \frac{|\{i \in X : T(i) > \delta \text{ and } O(i) = 1\}|}{|\{i \in X : T(i) > \delta\}|}$$

**Recall:**

$$\text{rec}(T, O) = \frac{|\{i \in X : T(i) > \delta \text{ and } O(i) = 1\}|}{|\{i \in X : O(i) = 1\}|}$$

# Assessing Trust

**Seed set:**  $S \subseteq V$  with  $L$  elements.

$S^+ = \{i \in S: O(i) = 1\}$  and  $S^- = \{i \in S: O(i) = 0\}$ .

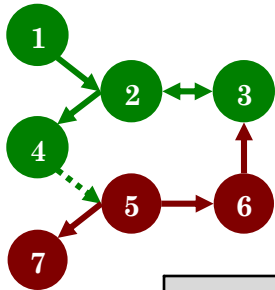
**Ignorant trust function:**

$$T_0(i) = \begin{cases} 1, & i \in S^+ \\ 0, & i \in S^- \\ \frac{1}{2}, & i \in V - S. \end{cases}$$

**M-step trust function:**

$$T_M(i) = \begin{cases} 1, & i \in S^+ \text{ or } i \notin S \text{ and } \exists j \in S^+ \text{ s.t. } d(i, j) \leq M \text{ and } d(i, k) > M \forall k \in S^- \\ 0, & i \in S^- \\ \frac{1}{2}, & i \in V - S. \end{cases}$$

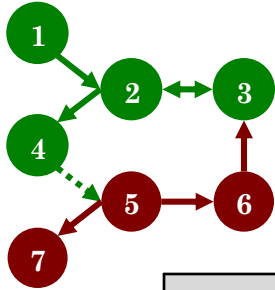
# Back to Example



Suppose  $S = \{1, 3, 6\}$ . Then,  $S^+ = \{1, 3\}$  and  $S^- = \{6\}$ .

Vertex $i$	Oracle $O(i)$	Ignorant Trust $T_0(i)$	1-step Trust $T_1(i)$	2-step Trust $T_2(i)$	3-step Trust $T_3(i)$
1					
2					
3					
4					
5					
6					
7					

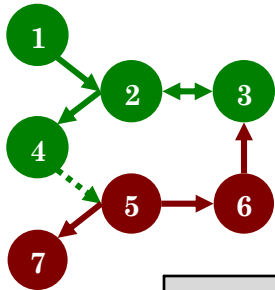
# Back to Example



Suppose  $S = \{1, 3, 6\}$ . Then,  $S^+ = \{1, 3\}$  and  $S^- = \{6\}$ .

Vertex $i$	Oracle $O(i)$	Ignorant Trust $T_0(i)$	1-step Trust $T_1(i)$	2-step Trust $T_2(i)$	3-step Trust $T_3(i)$
1	1				
2	1				
3	1				
4	1				
5	0				
6	0				
7	0				

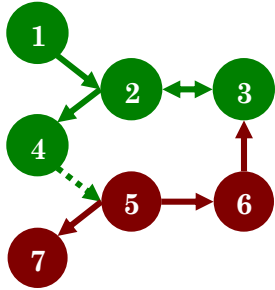
# Back to Example



Suppose  $S = \{1, 3, 6\}$ . Then,  $S^+ = \{1, 3\}$  and  $S^- = \{6\}$ .

Vertex $i$	Oracle $O(i)$	Ignorant Trust $T_0(i)$	1-step Trust $T_1(i)$	2-step Trust $T_2(i)$	3-step Trust $T_3(i)$
1	1	1	1	1	1
2	1	1/2	1	1	1
3	1	1	1	1	1
4	1	1/2	1/2	1	1
5	0	1/2	1/2	1/2	1
6	0	0	0	0	0
7	0	1/2	1/2	1/2	1/2

# Back to Example



Now, suppose  $X = V$ . Then,  $|P| = 7(6) = 42$ .

$M$	$\text{pairoid}(T_M, O, P)$	$\text{prec}(T_M, O)$ if $\delta = 1/2$	$\text{rec}(T_M, O)$ if $\delta = 1/2$
0	$(42 - 8)/42$ $= 17/21$	1	$1/2$
1	$(42 - 4)/42$ $= 19/21$	1	$3/4$
2	$(42 - 0)/42$ $= 1$	1	1
3	$(42 - 8)/42$ $= 17/21$	$4/5$	1

## Summary of Paper

- Formalizes the problem of Web spam and spam detection algorithms
- Introduces Inverse PageRank to select seed sets
- First use of an oracle to assess webpages
- Introduces the TrustRank algorithm (which is the PageRank algorithm with a carefully chosen personalization vector)
- Provides empirical results
- Defines trust assessing metrics