Graph Theory and Linear Algebra of Google's Pagerank

Chip Jackson, Cari Jamieson, Tyler Suronen

Fall 2013

Background

Google introduced in 1998 by Stanford graduate students Sergey Brin and Larry Page.

Goal was to eliminate "junk" results by looking at the hyperlink structure of the internet.

Developed PageRank algorithm that calculated the importance of a webpage based on the number of links pointing to it.

PageRank still in use today.

The Web as a Graph

A Simple Model

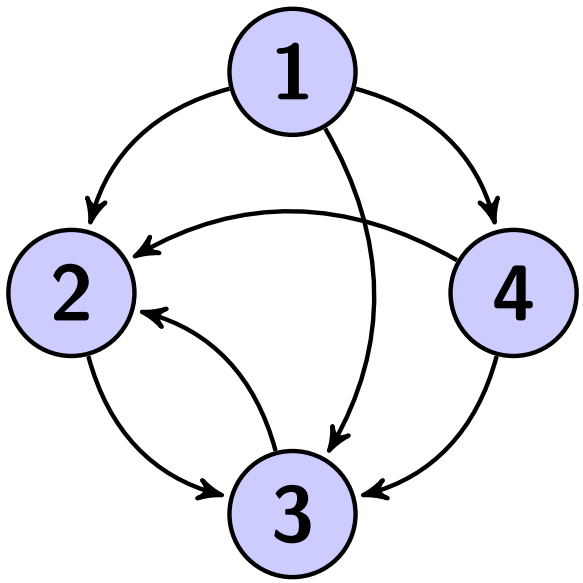
The internet is comprised of web pages and these pages link to one another. A simple model for the web is to represent the web by a directed graph.

The Web as a Graph

A Simple Model

The internet is comprised of web pages and these pages link to one another. A simple model for the web is to represent the web by a directed graph.

Ex: A web with four pages.



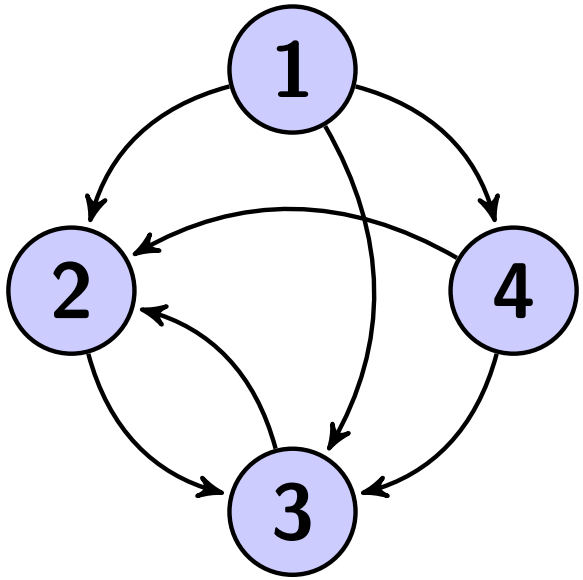
The Web as a Graph

Terminology

- Links to page A are called **backlinks for** A.
- A page with no outgoing links is called a **dangling node**.
- A graph is called **connected** if you may begin at any vertex and reach any other vertex by traversing edges.
- A graph is called **strongly-connected** if you may begin at any vertex and reach any other vertex by traversing edges *only along the direction they point*.
- E.g. A graph with a dangling node is not strongly-connected.

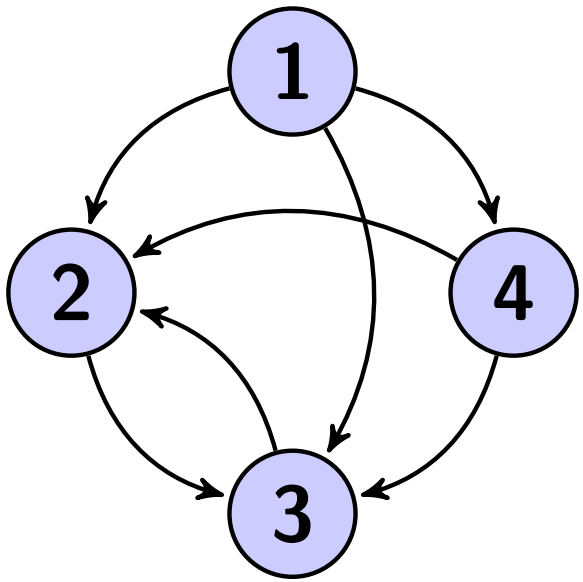
The Web as a Graph

Our previous example:



The Web as a Graph

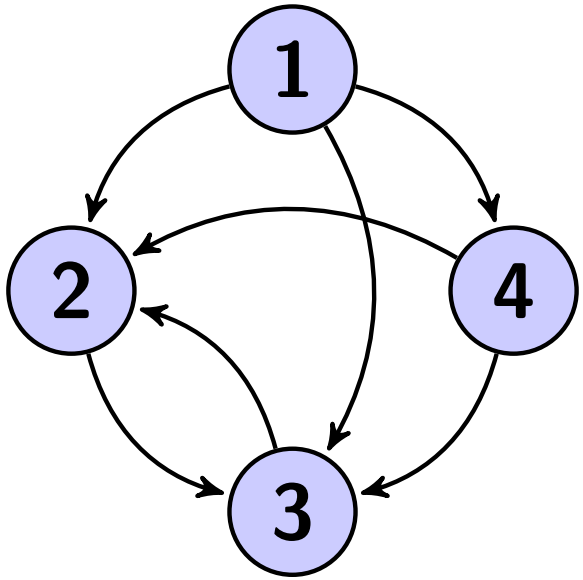
Our previous example:



- Each vertex has at least one outgoing link, so the graph has no dangling nodes.

The Web as a Graph

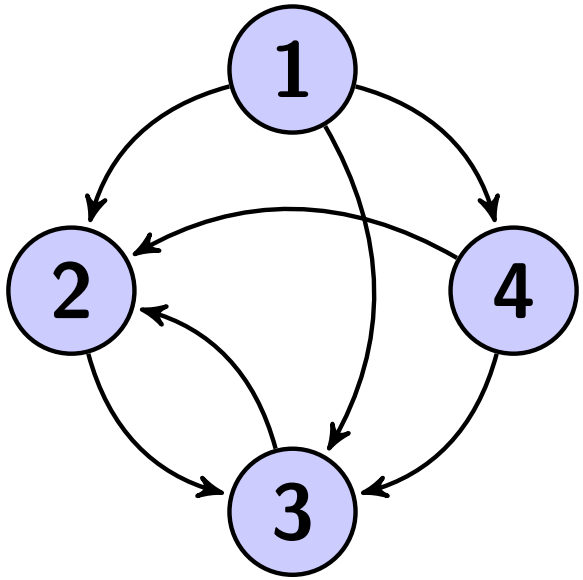
Our previous example:



- Each vertex has at least one outgoing link, so the graph has no dangling nodes.
- The graph is connected.

The Web as a Graph

Our previous example:



- Each vertex has at least one outgoing link, so the graph has no dangling nodes.
- The graph is connected.
- The graph is **not** strongly-connected.

PageRank Concept

Basic idea: PageRank(x) = number of pages that link to x

Problem: Pages that link to tons of other pages have too much influence.

Fix: Divide a page's vote evenly among all the pages it links to.

Problem: Some pages should have a larger vote than others (i.e. yahoo.com vs. wwu.com)

Fix: Weight a page's voting power by its own PageRank.

PageRank Calculation

L_k = the set of pages that link to page k
n_k = the number of pages that are linked to by page k

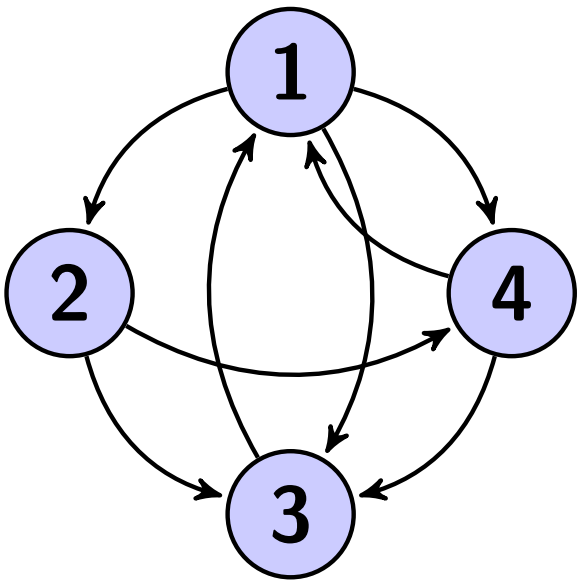
The PageRank of page k is

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j}$$

PageRank Algorithm

Build matrix A, where

$$A_{ij} = \begin{cases} \frac{1}{n_j} & j \in L_i \\ 0 & j \notin L_i \end{cases}$$

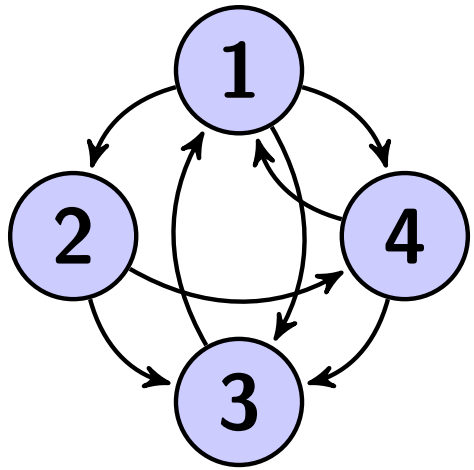


$$A = \begin{pmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

PageRank Algorithm

PageRanks will be the solutions to the equation $Ax = x$, or the eigenvectors with a corresponding eigenvalue of 1.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix}$$



A square matrix is *column-stochastic* if all entries are nonnegative and the entries in each column sum to one.

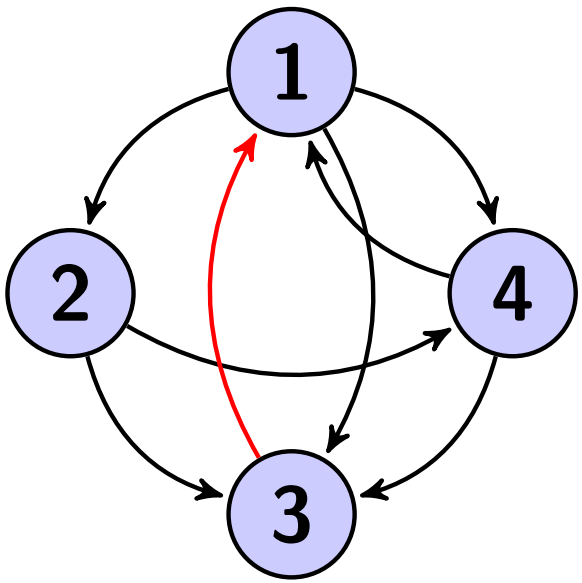
Proposition 1

Every column-stochastic matrix has 1 as an eigenvalue.

PageRank Algorithm

Complications...

Can't handle dangling nodes (no outgoing links)...



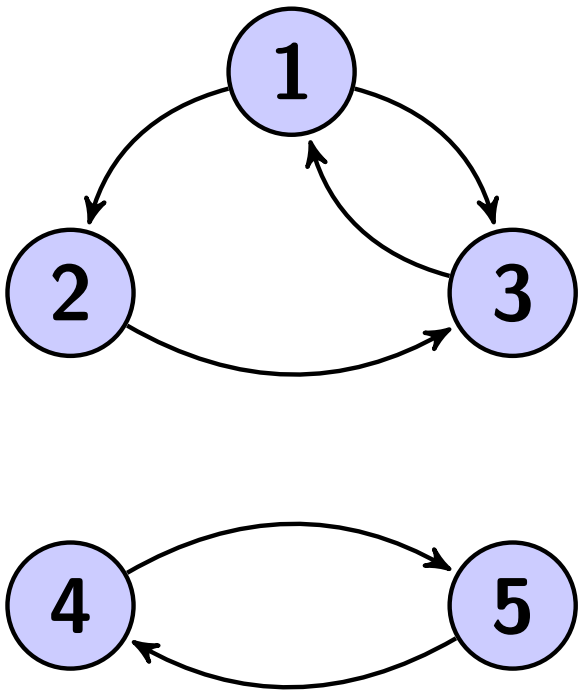
$$A = \begin{pmatrix} 0 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix}$$

$$\lambda = 0.56, 0, -0.28 \pm 0.26i$$

PageRank Algorithm

Complications...

Or disconnected graphs...



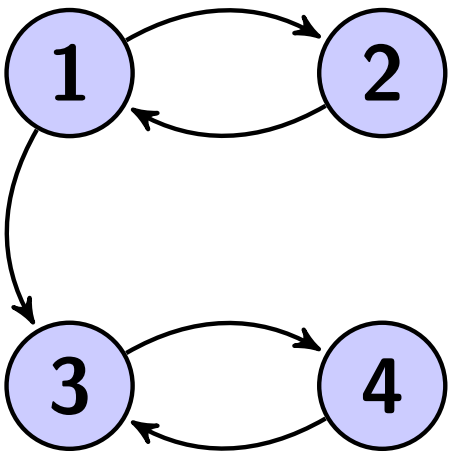
$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\lambda = 1, 1, 0, -1, -1$$

PageRank Algorithm

Complications...

Or graphs that aren't strongly connected...



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

$$x = [0, 0, 1, 1]$$

Modified PageRank

To resolve one of these issues we wish to make a modification of the original matrix which preserves the structure of the web while eliminating the issue of disconnected graphs. We replace the link matrix A with

$$M = (1 - m)A + mS$$

where S is the $n \times n$ matrix with all entries $\frac{1}{n}$ and $0 \leq m \leq 1$. This augmentation of A is a weighted average of A and S. For any $m \in [0, 1]$, M is column-stochastic. If $m = 0$ we get $M = A$; $m = 1$, $M = S$. Both of these cases are uninteresting. We expect a reasonable m to be small to preserve A. Initially, Google PageRank used $m = 0.15$.

Results

Let $V_1(M)$ denote the vector space of eigenvectors of M with eigenvalue 1.

Proposition 2

For any positive column-stochastic matrix M if $v \in V_1(M)$, then the components of v are all of one sign.

Proposition 2

Proof.
Suppose there exists an eigenvector, x, with positive and negative components.
Since $Mx = x$, we have $x_i = \sum_{j=1}^{n} M_{ij}x_j$ where the summands are of mixed sign.
By the triangle inequality,

$$|x_i| = \left|\sum_{j=1}^{n} M_{ij}x_j\right| < \sum_{j=1}^{n} M_{ij}|x_j|$$

We have:

$$\sum_{i=1}^{n} |x_i| < \sum_{i=1}^{n}\sum_{j=1}^{n} M_{ij}|x_j| = \sum_{j=1}^{n}\left(\sum_{i=1}^{n} M_{ij}\right)|x_j| = \sum_{j=1}^{n} |x_j|$$

Contradiction.

Other Results

Proposition 3

Let $v, w \in \mathbb{R}^m$ be linearly independent. Then $\exists\, s, t \in \mathbb{R}$ not both zero with $x = sv + tw$ having both positive and negative components.

Other Results

Proposition 3

Let $v, w \in \mathbb{R}^m$ be linearly independent. Then $\exists\, s, t \in \mathbb{R}$ not both zero with $x = sv + tw$ having both positive and negative components.

Proposition 4

If M is positive and column-stochastic then $V_1(M)$ has dimension 1.

HITS Method

HITS is an alternative to PageRank- a main difference being that it bases its results on a query, rather than calculating importance scores beforehand.
The main idea behind HITS is that that we can consider webpages to serve as both 'authorities' and 'hubs', where good authorities are pointed to from good hubs and good hubs point to good authorities. We calculate hub and authority scores iteratively by taking advantage of this relationship. For page i, given an initial authority score $a_i^{(0)}$ and an initial hub score $h_i^{(0)}$, then

$$a_i^{(k)} = \sum_j h_j^{(k-1)} \text{ for } e_{ji} \in E \text{ and } h_i^{(k)} = \sum_j a_j^{(k)} \text{ for } e_{ij} \in E$$

give the updated authority and hub scores, where E is the set of all directed edges in the graph.

HITS Method

Letting L be the adjacency matrix, where $L_{ij} = 1$ if $e_{ij} \in E$ and 0 if not, then we can rewrite the previous sums as

$$\overrightarrow{a}^{(k)} = L^T \overrightarrow{h}^{(k-1)} \text{ and } \overrightarrow{h}^{(k)} = L \overrightarrow{a}^{(k)}$$

where $\overrightarrow{a}^{(k)}$ is the vector containing the authority scores for each vertex on the kth iteration, and $\overrightarrow{h}^{(k)}$ defined similarly.

HITS Method

Letting L be the adjacency matrix, where $L_{ij} = 1$ if $e_{ij} \in E$ and 0 if not, then we can rewrite the previous sums as

$$\overrightarrow{a}^{(k)} = L^T \overrightarrow{h}^{(k-1)} \text{ and } \overrightarrow{h}^{(k)} = L \overrightarrow{a}^{(k)}$$

where $\overrightarrow{a}^{(k)}$ is the vector containing the authority scores for each vertex on the kth iteration, and $\overrightarrow{h}^{(k)}$ defined similarly. Rearrangement gives

$$\overrightarrow{a}^{(k)} = L^T L \overrightarrow{a}^{(k-1)} \text{ and } \overrightarrow{h}^{(k)} = L L^T \overrightarrow{h}^{(k-1)}$$

This is essentially a step in the power method of computing dominant eigenvectors.

HITS Algorithm

- Given a query, create a neighborhood graph N, consisting of webpages which contain terms in the query, or link to/from pages that contain the terms in the query
- Create adjacency matrix L from this graph
- From equations $\overrightarrow{a}^{(k)} = L^T L \overrightarrow{a}^{(k-1)}$ and $\overrightarrow{h}^{(k)} = LL^T \overrightarrow{h}^{(k-1)}$, calculate dominating eigenvector by considering $\lim_{k\to\infty} \frac{A^k x_0}{||A^k x_0||_1}$ where $A = L^T L$ or $A = LL^T$ respectively, given an initial vector x_0. These should converge on $\overrightarrow{a}^{(k)}$ and $\overrightarrow{h}^{(k)}$.
- Order the elements in the resulting vectors and return pages with largest hub and authority scores in two separate lists.

An Example

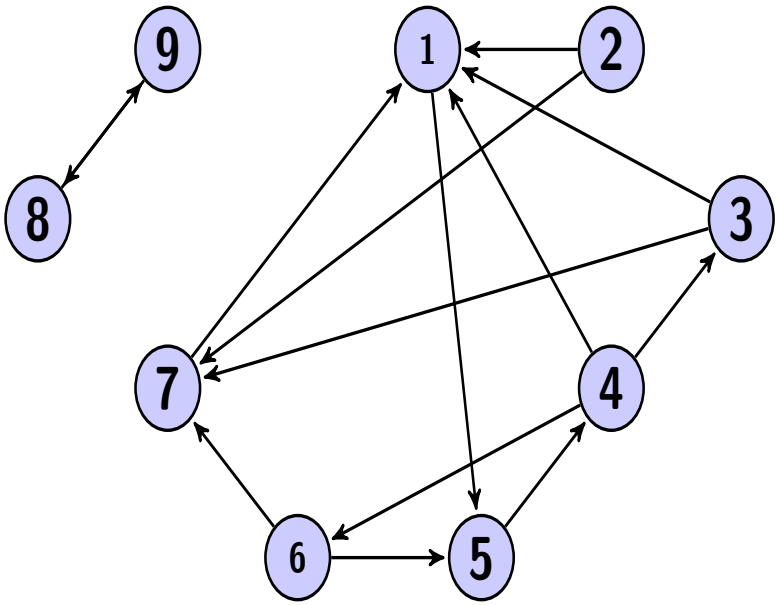
Consider the following 'Web' consisting of 9 webpages:

1. A History of Google
2. Representing Webpages with a Linear-Algebra Based Model
3. The Anatomy of a Large-Scale Hypertextual Web Search Engine
4. Efficient Crawling though URL Ordering
5. Queries and Computation on the Web
6. Mining Structural Information on the Web
7. Matrix Computations
8. Modeling Population Growth
9. Effect of Environmental Factors on Large Populations

Also consider the query 'using linear algebra to understand the Web'.

Using PageRank

The graph of the web and the link matrix A is calculated:



$$A = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Results for PageRank

We calculate the matrix $M = 0.85 * A + 0.15 * S$, where S is the matrix of entries all $\frac{1}{9}$. Then use the power method to determine the dominating eigenvector with eigenvalue 1 (with initial vector x_0 where each page has equal rank $\frac{1}{9}$):

Results for PageRank

We calculate the matrix $M = 0.85 * A + 0.15 * S$, where S is the matrix of entries all $\frac{1}{9}$. Then use the power method to determine the dominating eigenvector with eigenvalue 1 (with initial vector x_0 where each page has equal rank $\frac{1}{9}$):

$$x = [0.173, 0.017, 0.068, 0.180, 0.192, 0.068, 0.081, 0.111, 0.111]$$

Results for PageRank

We calculate the matrix $M = 0.85 * A + 0.15 * S$, where S is the matrix of entries all $\frac{1}{9}$. Then use the power method to determine the dominating eigenvector with eigenvalue 1 (with initial vector x_0 where each page has equal rank $\frac{1}{9}$):

$$x = [0.173, 0.017, 0.068, 0.180, 0.192, 0.068, 0.081, 0.111, 0.111]$$

Thus pages 4 and 5 have high importance scores.

Using HITS

We select all pages except 8 and 9, and create the following neighborhood graph N and corresponding adjacency matrix L:

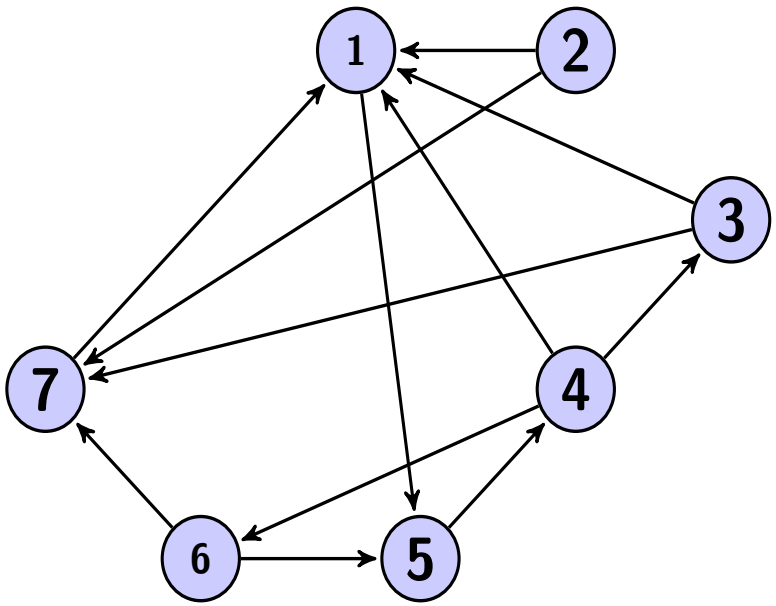


Figure: N

$$L = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Results of HITS

Calculate $H = L * L^T$ and $A = L^T * L$ and use power method to calculate dominant eigenvectors (with initial vector x_0 with all entries equal to 1):

Results of HITS

Calculate $H = L * L^T$ and $A = L^T * L$ and use power method to calculate dominant eigenvectors (with initial vector x_0 with all entries equal to 1):

$a = [0.477, 0.000, 0.131, 0.000, 0.000, 0.131, 0.262]$

Results of HITS

Calculate $H = L * L^T$ and $A = L^T * L$ and use power method to calculate dominant eigenvectors (with initial vector x_0 with all entries equal to 1):

$a = [0.477, 0.000, 0.131, 0.000, 0.000, 0.131, 0.262]$

$h = [0.000, 0.274, 0.274, 0.274, 0.000, 0.000, 0.177]$

Results of HITS

Calculate $H = L * L^T$ and $A = L^T * L$ and use power method to calculate dominant eigenvectors (with initial vector x_0 with all entries equal to 1):

$a = [0.477, 0.000, 0.131, 0.000, 0.000, 0.131, 0.262]$

$h = [0.000, 0.274, 0.274, 0.274, 0.000, 0.000, 0.177]$

Pages 1 and 7 are good authorities and pages 2, 3, and 4 are good hubs for this query.

References

- M.W. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval, 2nd Ed.* SIAM, Philadelphia, 2005. 77-88.
- S. Brin and L. Page. *The Anatomy of a Large-Scale Hypertextual Web Search Engine.* http://infolab.stanford.edu/ ∼ backrub/google.html (accessed November 2013)
- K. Bryan and T. Leise. *The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google.* http://www.rose-hulman.edu/∼ bryan/googleFinalVersionFixed.pdf (accessed November 2013)

Questions?