The Linear Algebra Aspects of PageRank

Ilse Ipsen

Thanks to Teresa Selee and Rebecca Wills
More PageRank More Visitors
Two Factors

Determine where Google displays a web page on the Search Engine Results Page:

1. **PageRank (links)**
   - A page has high PageRank if many pages with high PageRank link to it

2. **Hypertext Analysis (page contents)**
   - Text, fonts, subdivisions, location of words, contents of neighbouring pages
PageRank

An objective measure of the citation importance of a web page  
[Brin & Page 1998]

- Assigns a rank to every web page
- Influences the order in which Google displays search results
- Based on link structure of the web graph
- Does not depend on contents of web pages
- Does not depend on query
PageRank

... continues to provide the basis for all of our web search tools http://www.google.com/technology/

- “Links are the currency of the web”
- Exchanging & buying of links
- BO (backlink obsession)
- Search engine optimization
Overview

- Mathematical Model of Internet
- Computation of PageRank
- Sensitivity of PageRank to Rounding Errors
- Addition & Deletion of Links
- Web Pages that have no Outlinks
- Is the Ranking Correct?
Mathematical Model of Internet

1. Represent internet as graph
2. Represent graph as stochastic matrix
3. Make stochastic matrix more convenient $\Rightarrow$ Google matrix
4. dominant eigenvector of Google matrix $\Rightarrow$ PageRank
The Internet as a Graph

Link from one web page to another web page

Web graph:
Web pages = nodes
Links = edges
The Web Graph as a Matrix

Let $S = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$.

Links = nonzero elements in matrix
Elements of Matrix $S$

Assume: every page $i$ has $l_i \geq 1$ outlinks

If page $i$ has link to page $j$ then $s_{ij} = 1/l_i$
else $s_{ij} = 0$

Probability that surfer moves from page $i$ to page $j$
Properties of Matrix $S$

- Stochastic: $0 \leq s_{ij} \leq 1, \quad S\mathbf{1} = \mathbf{1}$
- Dominant left eigenvector:
  $$\omega^T S = \omega^T \quad \omega \geq 0 \quad \|\omega\|_1 = 1$$
- $\omega_i$ is probability that surfer visits page $i$

But: $\omega$ not unique
if $S$ has several eigenvalues equal to 1

Remedy: Make the matrix more convenient
Google Matrix

Convex combination

\[ G = \alpha S + (1 - \alpha) \mathbf{1} \mathbf{1}^T \]

- Stochastic matrix \( S \)
- Damping factor \( 0 \leq \alpha < 1 \)
  - e.g. \( \alpha = .85 \)
- Column vector of all ones \( \mathbf{1} \)
- Personalization vector \( \mathbf{v} \geq 0 \)
  - \( \|\mathbf{v}\|_1 = 1 \)
Models teleportation
Properties of Google Matrix $G$

$$G = \alpha S + (1 - \alpha) \mathbb{I} v^T$$

- Stochastic, reducible
- Eigenvalues of $G$:
  $$1 > \alpha \lambda_2(S) \geq \alpha \lambda_3(S) \geq \ldots$$
- Unique dominant left eigenvector:
  $$\pi^T G = \pi^T \quad \pi \geq 0 \quad \|\pi\|_1 = 1$$
PageRank

Google Matrix

\[ G = \alpha S + (1 - \alpha) \mathbb{1} \mathbb{1}^T \]

Links \hspace{1cm} Personalization

\[ \pi^T G = \pi^T \hspace{1cm} \pi \geq 0 \hspace{1cm} \|\pi\|_1 = 1 \]

\( \pi_i \) is PageRank of web page \( i \)

PageRank \( \Rightarrow \) dominant left eigenvector of \( G \)
How Google Ranks Web Pages

- **Model:**
  Internet $\rightarrow$ web graph $\rightarrow$ stochastic matrix $G$

- **Computation:**
  PageRank $\pi$ is eigenvector of $G$
  $\pi_i$ is PageRank of page $i$

- **Display:**
  If $\pi_i > \pi_k$ then page $i$ may* be displayed before page $k$

  * depending on hypertext analysis
History

- The anatomy of a large-scale hypertextual web search engine
  Brin & Page 1998
- US patent for PageRank granted in 2001
- Eigenstructure of the Google Matrix
  Haveliwala & Kamvar 2003
  Eldén 2003
  Serra-Capizzano 2005
Statistics

- Google indexes *10s of billions of web pages*
- “3 times more than any competitor”
- Google serves $\geq 200$ million queries per day
- Each query processed by $\geq 1000$ machines
- All search engines combined serve a total of $\geq 500$ million queries per day

[Desikan, 26 October 2006]
Computation of PageRank

*The world’s largest matrix computation*
[Moler 2002]

- Eigenvector
- Matrix dimension is 10s of billions
- The matrix changes often
  250,000 new domain names every day
- **Fortunately:** Matrix is sparse
Power Method

Want: \( \pi \) such that \( \pi^T G = \pi^T \)

Power method:

Pick an initial guess \( \mathbf{x}^{(0)} \)
Repeat

\[
[\mathbf{x}^{(k+1)}]^T := [\mathbf{x}^{(k)}]^T G
\]

Each iteration is a matrix vector multiply
Matrix Vector Multiply

\[ x^T G = x^T \left[ \alpha S + (1 - \alpha) \mathbb{1} v^T \right] \]
An Iteration is Cheap

Google matrix \( G = \alpha S + (1 - \alpha) \mathbf{1} \mathbf{1}^T \mathbf{v}^T \)

Vector \( x \geq 0 \quad \| x \|_1 = 1 \)

\[
x^T G = x^T \left[ \alpha S + (1 - \alpha) \mathbf{1} \mathbf{1}^T \mathbf{v}^T \right] \\
= \alpha x^T S + (1 - \alpha) x^T \mathbf{1} \mathbf{1}^T \mathbf{v}^T \\
= \alpha x^T S + (1 - \alpha) \mathbf{v}^T
\]

Cost: # non-zero elements in \( S \)
Error in Power Method

\[ \pi^T G = \pi^T \quad G = \alpha S + (1 - \alpha) \|v\|^T \]

\[
[x^{(k+1)} - \pi]^T = [x^{(k)}]^T G - \pi^T G
= \alpha [x^{(k)}]^T S - \alpha \pi^T S
= \alpha [x^{(k)} - \pi]^T S
\]

\[
\underbrace{\|x^{(k+1)} - \pi\|}_{\text{iteration } k+1} \leq \alpha \underbrace{\|x^{(k)} - \pi\|}_{\text{iteration } k}
\]

Norms: 1, \infty
Error in Power Method

\[ \pi^T G = \pi^T \quad G = \alpha S + (1 - \alpha) \mathbb{1} v^T \]

Error after \( k \) iterations:

\[ \|x^{(k)} - \pi\| \leq \alpha^k \|x^{(0)} - \pi\| \leq 2 \]

Norms: 1, \( \infty \)  
[Bianchini, Gori & Scarselli 2003]

Error bound does not depend on matrix dimension
Iteration Counts for Different $\alpha$

bound: $k$ such that $2 \alpha^k \leq 10^{-8}$

Termination based on residual norms vs bound

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$n = 281903$</th>
<th>$n = 683446$</th>
<th>bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>.85</td>
<td>69</td>
<td>65</td>
<td>119</td>
</tr>
<tr>
<td>.90</td>
<td>107</td>
<td>102</td>
<td>166</td>
</tr>
<tr>
<td>.95</td>
<td>219</td>
<td>220</td>
<td>415</td>
</tr>
<tr>
<td>.99</td>
<td>1114</td>
<td>1208</td>
<td>2075</td>
</tr>
</tbody>
</table>

Fewer iterations than predicted by bound
Advantages of Power Method

- Converges to unique vector
- Convergence rate $\alpha$
- Convergence independent of matrix dimension
- Vectorizes
- Storage for only a single vector
- Sparse matrix operations
- Accurate (no subtractions)
- Simple (few decisions)

But: can be slow
PageRank Computation

- **Power method**
  - Page, Brin, Motwani & Winograd 1999
  - Bianchini, Gori & Scarselli 2003

- **Acceleration of power method**
  - Kamvar, Haveliwala, Manning & Golub 2003
  - Haveliwala, Kamvar, Klein, Manning & Golub 2003
  - Brezinski, Redivo-Zaglia & Serra-Capizzano 2005

- **Aggregation/Disaggregation**
  - Ipsen & Kirkland 2006
PageRank Computation

- **Methods that adapt to web graph**
  Broder, Lempel, Maghoul & Pedersen 2004
  Kamvar, Haveliwala & Golub 2004
  Haveliwala, Kamvar, Manning & Golub 2003
  Lee, Golub & Zenios 2003
  Lu, Zhang, Xi, Chen, Liu, Lyu & Ma 2004
  Ipsen & Selee 2006

- **Krylov methods**
  Golub & Greif 2004
  Del Corso, Gullí, Romani 2006
PageRank Computation

- **Schwarz & asynchronous methods**
  Bru, Pedroche & Szyld 2005
  Kollias, Gallopoulos & Szyld 2006

- **Linear system solution**
  Arasu, Novak, Tomkins & Tomlin 2002
  Arasu, Novak & Tomkins 2003
  Bianchini, Gori & Scarselli 2003
  Gleich, Zukov & Berkin 2004
  Del Corso, Gullí & Romani 2004
  Langville & Meyer 2006
PageRank Computation

- Surveys of numerical methods:
  Langville & Meyer 2004
  Berkhin 2005
  Langville & Meyer 2006 (book)
Sensitivity of PageRank

How sensitive is PageRank $\pi$ to small perturbations, e.g. rounding errors

- Changes in matrix $S$
- Changes in damping factor $\alpha$
- Changes in personalization vector $\nu$
Perturbation Theory

For Markov chains

Schweizer 1968, Meyer 1980
Haviv & van Heyden 1984
Funderlic & Meyer 1986
Golub & Meyer 1986
Seneta 1988, 1991
Ipsen & Meyer 1994
Kirkland, Neumann & Shader 1998
Cho & Meyer 2000, 2001
Kirkland 2003, 2004
Perturbation Theory

For Google matrix

Chien, Dwork, Kumar & Sivakumar 2001
Ng, Zheng & Jordan 2001
Bianchini, Gori & Scarselli 2003
Boldi, Santini & Vigna 2004, 2005
Langville & Meyer 2004
Golub & Greif 2004
Kirkland 2005, 2006
Chien, Dwork, Kumar, Simon & Sivakumar 2005
Avrechenkov & Litvak 2006
Changes in the Matrix \( S \)

**Exact:**

\[
\pi^T G = \pi^T \quad G = \alpha S + (1 - \alpha) \mathbf{1} v^T
\]

**Perturbed:**

\[
\tilde{\pi}^T \tilde{G} = \tilde{\pi}^T \quad \tilde{G} = \alpha (S + E) + (1 - \alpha) \mathbf{1} v^T
\]

**Error:**

\[
\tilde{\pi}^T - \pi^T = \alpha \tilde{\pi}^T E (I - \alpha S)^{-1}
\]

\[
\|\tilde{\pi} - \pi\|_1 \leq \frac{\alpha}{1 - \alpha} \|E\|_\infty
\]
Changes in $\alpha$ and $\nu$

• Change in amplification factor:

$$\tilde{G} = (\alpha + \mu)S + (1 - (\alpha + \mu)) \mathbb{1} v^T$$

Error: $\|\tilde{\pi} - \pi\|_1 \leq \frac{2}{1-\alpha} |\mu|$

[Langville & Meyer 2004]

• Change in personalization vector:

$$\tilde{G} = \alpha S + (1 - \alpha) \mathbb{1} (\nu + f)^T$$

Error: $\|\tilde{\pi} - \pi\|_1 \leq \|f\|_1$
Sensitivity of PageRank $\pi$

$$\pi^T G = \pi^T \quad G = \alpha S + (1 - \alpha) \mathbf{1} v^T$$

Changes in

- $S$: condition number $\alpha/(1 - \alpha)$
- $\alpha$: condition number $2/(1 - \alpha)$
- $v$: condition number $1$

$\alpha = .85$: condition numbers $\leq 14$
$\alpha = .99$: condition numbers $\leq 200$

PageRank insensitive to rounding errors
Adding an In-Link

Adding an in-link increases PageRank (monotonicity)

Removing an in-link decreases PageRank

[Chien, Dwork, Kumar & Sivakumar 2001]
[Chien, Dwork, Kumar, Simon & Sivakumar 2005]
Adding an Out-Link

\[ \tilde{\pi}_3 = \frac{1 + \alpha + \alpha^2}{3(1 + \alpha + \alpha^2/2)} < \pi_3 = \frac{1 + \alpha + \alpha^2}{3(1 + \alpha)} \]

Adding an out-link may decrease PageRank
Justification for TrustRank

Adjust personalization vector to combat web spam
[Gyöngyi, Garcia-Molina, Pedersen 2004]

Increase $v$ for page $i$: $v_i := v_i + \phi$
Decrease $v$ for page $j$: $v_j := v_j - \phi$

PageRank of page $i$ increases: $\tilde{\pi}_i > \pi_i$
PageRank of page $j$ decreases: $\tilde{\pi}_j < \pi_j$

Total change in PageRank $\|\tilde{\pi} - \pi\|_1 \leq 2\phi$
Web Pages that have no Outlinks

- Technical term: Dangling Nodes
- Examples:
  - Image files
  - PDF and PS files
  - Pages whose links have not yet been crawled
  - Protected web pages
- 50%-80% of all web pages
- Problem: zero rows in matrix
- Popular fix: Insert artificial links
Dangling Node Fix

\[
\begin{bmatrix}
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{2} & 0 & \frac{1}{2} & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
\begin{bmatrix}
w_1 & w_2 & w_3 & w_4
\end{bmatrix}
\]
Inside the Stochastic Matrix $S$

Number pages so that dangling nodes are last

$$S = \begin{pmatrix} H \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \mathbf{1} w^T \end{pmatrix}$$

Links from nondangling nodes: $H$

Dangling node vector $w \geq 0$ $\|w\|_1 = 1$

Google matrix $G = \alpha \begin{pmatrix} H \\ \mathbf{1} w^T \end{pmatrix} + (1 - \alpha) \mathbf{1} \mathbf{v}^T$
Partitioning the Google Matrix

\[ G = \begin{pmatrix} G_{11} & G_{12} \\ 11 u_1^T & 11 u_2^T \end{pmatrix} \]

\[ (u_1^T, u_2^T) = \alpha \omega^T + (1 - \alpha) v^T \]

- **dangling nodes**
- **personalization**
Lumping

Separate dangling and non-dangling nodes.
“Lump” all dangling nodes into single node.

- **Stochastic matrices:**
  - Kemeny & Snell 1960
  - Dayar & Stewart 1997
  - Jernigan & Baran 2003
  - Gurvits & Ledoux 2005

- **Google matrix:**
  - Lee, Golub & Zenios 2003
  - Ipsen & Selee 2006
Example

→: real links
→: artificial links
Lumped Example
Google Lumping

1. “Lump” all dangling nodes into a single node
2. Compute dominant eigenvector of smaller, lumped matrix
   \[ \Rightarrow \text{PageRank of nondangling nodes} \]
3. Determine PageRank of dangling nodes with one matrix vector multiply
1. Lump Dangling Nodes

Google Matrix $G$

Lumped matrix $L$
1. Lump Dangling Nodes

\[ G = \begin{pmatrix} G_{11} & G_{12} \\ \mathbb{1} u_1^T & \mathbb{1} u_2^T \end{pmatrix} \]

Lump \( n - d \) dangling nodes into a single node

\[ L = \begin{pmatrix} \mathbb{1} & \mathbb{1} \\ \mathbb{1} u_1^T & \mathbb{1} u_2^T \end{pmatrix} \]

Stochastic, same nonzero eigenvalues as \( G \)
2. Eigenvector of Lumped Matrix

\[ L = \begin{pmatrix} G_{11} & G_{12} \\ u_1^T & u_2^T \end{pmatrix} \]

Lumped matrix with \( d \) nondangling nodes

Compute eigenvector of lumped matrix

\[ \sigma^T L = \sigma^T \quad \sigma \geq 0 \quad \|\sigma\|_1 = 1 \]

PageRank of nondangling nodes: \( \sigma_{1:d} \)
3. Dangling Nodes

\[ G = \begin{pmatrix} G_{11} & G_{12} \\ 11 u_1^T & 11 u_2^T \end{pmatrix} \quad L = \begin{pmatrix} G_{11} & G_{12} 11 \\ u_1^T & u_2^T 11 \end{pmatrix} \]

Eigenvector of lumped matrix: \( \sigma^T L = \sigma^T \)

PageRank of dangling nodes:

\[ \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \]

One matrix vector multiply
Summary: Dangling Nodes

$n$ web pages with $n - d$ dangling nodes

- PageRank $\sigma_{1:d}$ of $d$ nondangling nodes: from lumped matrix $L$ of dimension $d + 1$
- PageRank of dangling nodes: one matrix vector multiply
- Total PageRank

$$\pi^T = \begin{pmatrix} \sigma_{1:d}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{pmatrix}$$
Summary: Dangling Nodes, ctd.

- PageRank of nondangling nodes is independent of PageRank of dangling nodes.
- PageRank of nondangling nodes can be computed separately.
- Power method on lumped matrix $L$: same convergence rate as for $G$ but $L$ much smaller than $G$.
- Speed increases with # dangling nodes.
Is the Ranking Correct?

\[ \pi^T = (0.23, 0.24, 0.26, 0.27) \]

- \[ [x^{(k)}]^T = (0.27, 0.26, 0.24, 0.23) \]
  \[ \|x^{(k)} - \pi\|_\infty = 0.04 \]
  Small error, but incorrect ranking

- \[ [x^{(k)}]^T = (0, 0.001, 0.002, 0.997) \]
  \[ \|x^{(k)} - \pi\|_\infty = 0.727 \]
  Large error, but correct ranking
Is the Ranking Correct?

After $k$ iterations of power method:
Error: $\|x^{(k)} - \pi\| \leq 2 \alpha^k$

But: Do the components of $x^{(k)}$ have the same ranking as those of $\pi$?

Rank-stability, rank-similarity: [Lempel & Moran, 2005]
[Borodin, Roberts, Rosenthal & Tsaparas 2005]
Web Graph is a Ring

\[ S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \]

[Ipse & Wills]
All Pages are Trusted

$S$ is circulant of order $n$, \( v = \frac{1}{n}1 \)

- **PageRank:** \( \pi = \frac{1}{n}1 \)
  All pages have **same PageRank**

- **Power method**
  \( x^{(0)} = v: \quad x^{(0)} = \pi \) correct ranking
  \( x^{(0)} \neq v: \quad [x^{(k)}]^T \sim \frac{1}{n}1^T + \alpha^k \left( [x^{(0)}]^TS^k - \frac{1}{n}1 \right) \)
  Ranking does not converge (in exact arithmetic)
Only One Page is Trusted

\[ v^T = (1 \ 0 \ 0 \ 0 \ 0 \ 0) \]
Only One Page is Trusted

PageRank decreases with distance from page 1

\[ \pi^T \sim (1 \alpha \alpha^2 \alpha^3 \alpha^4) \]
Only One Page is Trusted

$S$ is circulant of order $n$, $v = e_1$

- PageRank: $\pi^T \sim (1 \ \alpha \ \ldots \ \alpha^{n-1})$

- Power method with $x^{(0)} = v$:

  $[x^{(k)}]^T \sim \begin{pmatrix} 1 & \alpha & \ldots & \alpha^{k-1} & \frac{\alpha^k}{1-\alpha} & 0 & \ldots & 0 \end{pmatrix}$

  $[x^{(n)}]^T \sim \begin{pmatrix} 1 + \frac{\alpha^n}{1-\alpha} & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \end{pmatrix}$

Rank convergence in $n$ iterations
Too Many Iterations

Power method with $x^{(0)} = v = e_1$:

- After $n$ iterations:
  \[ [x^{(n)}]^T \sim \left( 1 + \frac{\alpha^n}{1-\alpha} \alpha \alpha^2 \ldots \alpha^{n-1} \right) \]

- After $n + 1$ iterations:
  \[ [x^{(n+1)}]^T \sim \left( 1 + \alpha^n \alpha + \frac{\alpha^{n+1}}{1-\alpha} \alpha^2 \ldots \alpha^{n-1} \right) \]

If $\alpha = .85$, $n = 10$: \[ \alpha + \frac{\alpha^{n+1}}{1-\alpha} > 1 + \alpha^n \]

Additional iterations can destroy a converged ranking.
Recovery of Ranking

$S$ is circulant of order $n$

- After $k$ iterations:

$$\left[x^{(k)}\right]^T = \alpha^k \left[x^{(0)}\right]^T S^k + (1 - \alpha) v^T \sum_{j=0}^{k-1} \alpha^j S^j$$

- After $k + n$ iterations:

$$\left[x^{(k+n)}\right]^T = \alpha^n \left[x^{(k)}\right]^T + (1 - \alpha^n) \pi^T$$

If $x^{(k)}$ has correct ranking, so does $x^{(k+n)}$
Any Personalization Vector

$S$ is circulant of order $n$

- PageRank: $\pi^T \sim v^T \sum_{j=0}^{n-1} \alpha^j S^j$

- Power method with $x^{(0)} = \frac{1}{n} 1$

$$[x^{(n)}]^T = (1 - \alpha^n) \pi^T + \frac{\alpha^n}{n} 11^T$$

For any $v$: rank convergence after $n$ iterations
Problems with Ranking

- Ranking may never converge
- Additional iterations can destroy ranking
- Small error does not imply correct ranking
- Rank convergence depends on: $\alpha$, $v$, initial guess, matrix dimension, structure of web graph
- How do we know when the ranking is correct?
- Even if successive iterates have the same ranking, their ranking may not be correct
Summary

- Google orders web pages according to: **PageRank** and **hypertext analysis**
- **PageRank** = left eigenvector of $G$
  \[ G = \alpha S + (1 - \alpha) 11^T \]
- Power method: simple and robust
- Error in iteration $k$ bounded by $\alpha^k$
- Convergence rate largely independent of dimension and eigenvalues of $G$
Summary, ctd

- PageRank **insensitive** to rounding errors
- Adding in-links **increases** PageRank
- Adding out-links **may decrease** PageRank
- **Dangling nodes** = pages w/o outlinks
  - Rank one change to hyperlink matrix
- **Lumping:**
  - PageRank of **non**dangling nodes computed **separately** from PageRank of dangling nodes
- **Ranking problem:** DIFFICULT
User-Friendly Resources

- **Rebecca Wills:**
  *Google’s PageRank: The Math Behind the Search Engine*
  Mathematical Intelligencer, 2006

- **Amy Langville & Carl Meyer:**
  *Google’s PageRank and Beyond The Science of Search Engine Rankings*
  Princeton University Press, 2006

- **Amy Langville & Carl Meyer:**
  Broadcast of On-Air Interview, November 2006
  Carl Meyer’s web page