Applying Orthogonal/Power Iterations to Big Graph Mining : PageRank and Kempe-McSherry Algorithms

Sergiy Gogolenko

Ukrainian Catholic University, Lviv

2016, May 23

Outline

PageRank

What made them heros of top magazine cover?

PageRank: The model

Measure of importance

The importance of a Web page is an inherently subjective matter...But there is still much that can be said objectively about the relative importance of Web pages.

Page, L.; Brin, S.; Motwani, R.; Winograd, T. (1999) *The PageRank Citation Ranking: Bringing Order to the Web.* Tech.rep. Stanford InfoLab.

Probability of visiting page by idealized random Web surfer

Why this trick works?

- users of the Web "vote with their feet"
- users are more likely to visit useful pages

PageRank: The model

Measure of importance

The importance of a Web page is an inherently subjective matter...But there is still much that can be said objectively about the relative importance of Web pages.

Page, L.; Brin, S.; Motwani, R.; Winograd, T. (1999) *The PageRank Citation Ranking: Bringing Order to the Web*. Tech.rep. Stanford InfoLab.

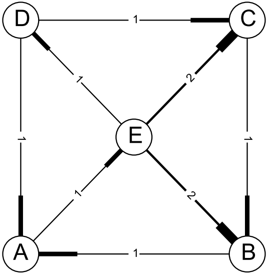
Probability of visiting page by idealized random Web surfer

Why this trick works?

- users of the Web "vote with their feet"
- users are more likely to visit useful pages

PageRank: The model

Random surfing as Markov process



Probabilities

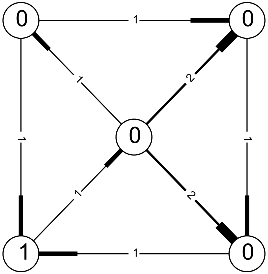
1. $p = (1, 0, 0, 0, 0, 0)$
2. $p = (0, 0, 0, 0, 0, 1)$
3. $p = (0, \frac{2}{5}, \frac{2}{5}, \frac{1}{5}, 0)$

Restrictions

- strongly connected graph
- no dead-ends

PageRank: The model

Random surfing as Markov process



Probabilities

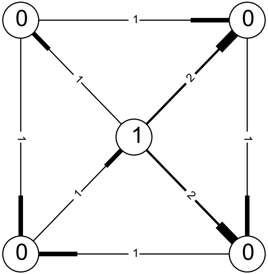
1. $p = (1, 0, 0, 0, 0, 0)$
2. $p = (0, 0, 0, 0, 0, 1)$
3. $p = (0, \frac{2}{5}, \frac{2}{5}, \frac{1}{5}, 0)$

Restrictions

- strongly connected graph
- no dead-ends

PageRank: The model

Random surfing as Markov process



Probabilities

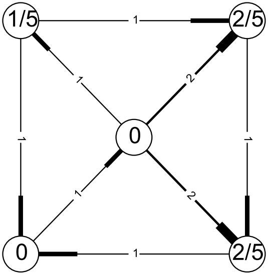
1. $p = (1, 0, 0, 0, 0, 0)$
2. $p = (0, 0, 0, 0, 0, 1)$
3. $p = (0, \frac{2}{5}, \frac{2}{5}, \frac{1}{5}, 0)$

Restrictions

- strongly connected graph
- no dead-ends

PageRank: The model

Random surfing as Markov process



Probabilities

1. $p = (1, 0, 0, 0, 0, 0)$
2. $p = (0, 0, 0, 0, 0, 1)$
3. $p = (0, \frac{2}{5}, \frac{2}{5}, \frac{1}{5}, 0)$

Restrictions

- strongly connected graph
- no dead-ends

PageRank: The model

Transition matrix of the Web

Restrictions

- strongly connected graph
- no dead-ends

Restrictions

- strongly connected graph
- no dead-ends

PageRank: The model

Transition matrix of the Web

Restrictions

- strongly connected graph
- no dead-ends

Restrictions

- strongly connected graph
- no dead-ends

PageRank: The model

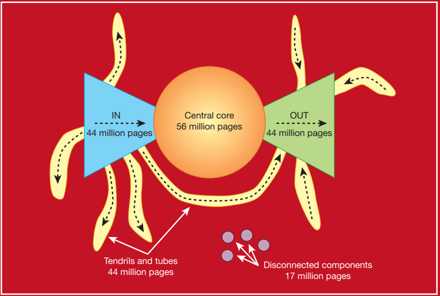
Structure of the Web

The web is a bow tie

A study of the web's structure, five times larger than any attempted previously, reveals that it isn't the fully interconnected network that we've been led to believe. The study suggests that the chance of being able to surf between two randomly chosen pages is less than one in four.

Researchers from three Californian groups — at IBM's Almaden Research Center in San Jose, the Altavista search engine in San Mateo and Compaq Systems Research Center in Palo Alto — have analysed 200 million web pages and 1.5 billion hyperlinks. Their results, which will be presented next week at the World Wide Web 9 Conference in Amsterdam, indicate that the web is made up of four distinct components.

A central core contains pages between which users can surf easily. Another large cluster, labelled 'in', contains pages that link to the core but cannot be reached from it. These are often new pages that have not yet been linked to. A separate 'out' cluster consists of pages that can be reached from the core but do not link to it, such as corporate websites containing only internal links. Other groups of pages, called 'tendrils' and 'tubes', connect to either the in or out clusters, or both, but not to the core, whereas some pages are completely unconnected. To illustrate this structure, the researchers picture the web as a plot shaped like a bow tie with finger-like projections.



©*Nature* 405, 113 (11 May 2000)

Web as "bowtie"

- in-component
- out-component
- tendrils
  - tubes
  - isolated components

Problems

- dead-ends
- spider traps

PageRank: The model

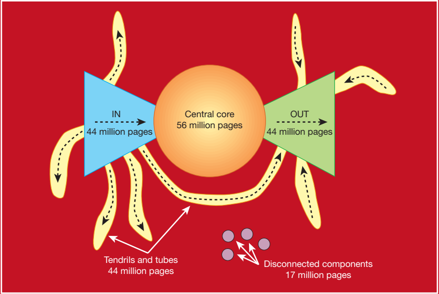
Structure of the Web

The web is a bow tie

A study of the web's structure, five times larger than any attempted previously, reveals that it isn't the fully interconnected network that we've been led to believe. The study suggests that the chance of being able to surf between two randomly chosen pages is less than one in four.

Researchers from three Californian groups — at IBM's Almaden Research Center in San Jose, the Altavista search engine in San Mateo and Compaq Systems Research Center in Palo Alto — have analysed 200 million web pages and 1.5 billion hyperlinks. Their results, which will be presented next week at the World Wide Web 9 Conference in Amsterdam, indicate that the web is made up of four distinct components.

A central core contains pages between which users can surf easily. Another large cluster, labelled 'in', contains pages that link to the core but cannot be reached from it. These are often new pages that have not yet been linked to. A separate 'out' cluster consists of pages that can be reached from the core but do not link to it, such as corporate websites containing only internal links. Other groups of pages, called 'tendrils' and 'tubes', connect to either the in or out clusters, or both, but not to the core, whereas some pages are completely unconnected. To illustrate this structure, the researchers picture the web as a plot shaped like a bow tie with finger-like projections.



©*Nature* 405, 113 (11 May 2000)

Web as "bowtie"

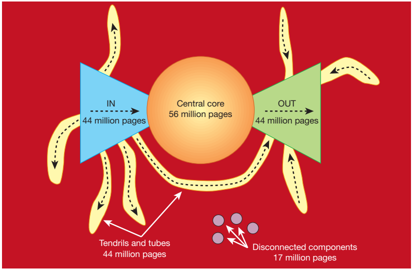
- in-component
- out-component
- tendrils
  - tubes
  - isolated components

Problems

- dead-ends
- spider traps

PageRank: The model

Avoiding dead ends by dropping

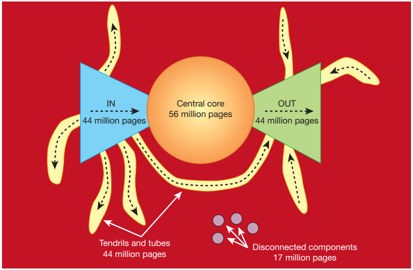


Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model

Avoiding dead ends by dropping

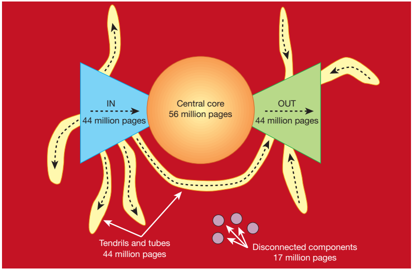


Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model

Avoiding dead ends by dropping

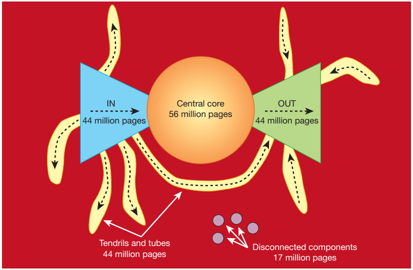


Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model

Avoiding dead ends by dropping

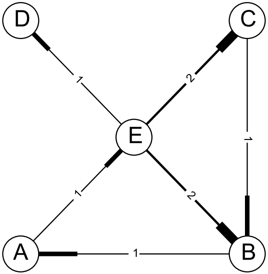


Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model

Teleporting

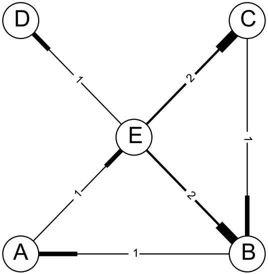


Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model

Teleporting

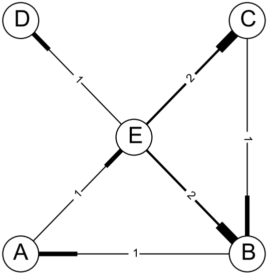


Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model

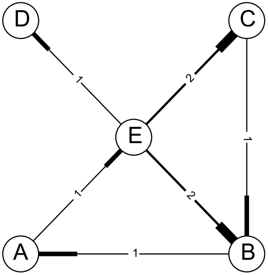
Teleporting



Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

PageRank: The model
Teleporting



Algorithm

1. Backward graph reduction: remove dead ends iteratively
2. Compute PageRanks of reduced graph
3. Forward PageRank computing

Applications

Major application areas

Key

Yet the war between those who want to make the Web useful and those who would exploit it for their own purposes is never over.

Techniques for preventing link spammers (search-engine optimization)

- TrustRank
- topic-sensitive `PageRank`

Key

Yet the war between those who want to make the Web useful and those who would exploit it for their own purposes is never over.

Techniques for preventing link spammers (search-engine optimization)

- TrustRank
- topic-sensitive `PageRank`

Basic algorithms

Framework

Factorization-based approaches to matrix inversion

- based on Gaussian elimination
  - row partial pivoting is fast but unstable for p-cyclic matrices (S. Wright, 1993)
- based on structured orthogonal factorization
  - factorization is stable and has identical complexity to the best known Gaussian elimination algorithms

Orthogonal inversion approach

$$\boxed{H^{-1} = R^{-1} Q^T}$$

1. Compute QR decomposition $H = QR$
2. Invert the factor R
3. Apply the factor Q to R^{-1}

Factorization-based approaches to matrix inversion

- based on Gaussian elimination
  - row partial pivoting is fast but unstable for p-cyclic matrices (S. Wright, 1993)
- based on structured orthogonal factorization
  - factorization is stable and has identical complexity to the best known Gaussian elimination algorithms

Orthogonal inversion approach

$$\boxed{H^{-1} = R^{-1}Q^T}$$

1. Compute QR decomposition $H = QR$
2. Invert the factor R
3. Apply the factor Q to R^{-1}

Basic algorithms

Structured orthogonal factorization

BSOF – Wright's serial version of SOF algorithm

Data: H, n, p

Result: R, $\{Q^{(k)} | 1 \leq k < p-1\}$

$R \leftarrow O$; $\tilde{A}_1 \leftarrow A_1$; $\tilde{B}_1 \leftarrow B_p$;

for $\underline{k \in \{1, 2, \ldots, p-2\}}$ **do**

  Compute regular QR: $Q^{(k)} \begin{bmatrix} R_{kk} \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{A}_k \\ B_k \end{bmatrix}$;

  $\begin{bmatrix} R_{k,k+1} & R_{k,p} \\ \tilde{A}_{k+1} & \tilde{B}_{k+1} \end{bmatrix} \leftarrow \left(Q^{(k)}\right)^T \begin{bmatrix} 0 & \tilde{B}_k \\ A_{k+1} & 0 \end{bmatrix}$;

Compute the QR: $Q^{(p-1)} \begin{bmatrix} R_{p-1,p-1} & R_{p-1,p} \\ 0 & R_{p,p} \end{bmatrix} = \begin{bmatrix} \tilde{A}_{p-1} & \tilde{B}_{p-1} \\ B_{p-1} & A_{p,p} \end{bmatrix}$;

Basic algorithms

Inversion via block back substitution

BSTRI_RV – Row Version of the BBS

Data: R, n, p

Result: X

$X \leftarrow O$;

$X_{p-2:p,p-2:p} \leftarrow R^{-1}_{p-2:p,p-2:p}$;

$\texttt{Batched}\ _{i=1:p-3}\ \{X_{ii} \leftarrow R_{ii}^{-1}\}$;

$X_{1:p-3,p} \leftarrow R_{1:p-3,p} X_{p,p}$;

$\texttt{Batched}\ _{i=1:p-3}\ \{X_{i,p} \leftarrow -X_{ii} R_{i,p},\ X_{i,i+1} \leftarrow -X_{ii} R_{i,i+1}\ \}$;

for $\underline{i \in \{p-3, p-4, \ldots, 1\}}$ **do**

  $X_{i,i+2:p} \leftarrow X_{i,i+2:p} + X_{i,i+1} X_{i+1,i+2:p}$;

  $X_{i,i+1} \leftarrow X_{i,i+1} X_{i+1,i+1}$;

BSTRI_CV – Column Version of the BBS

Similar to BSTRI_RV

Basic algorithms

Complexity

Operation counts

Phase	Routine	Additions	Multiplications	Total Flops
I	BSOF	$\Theta\left(\frac{23}{3}n^3p\right)$	$\Theta\left(\frac{23}{3}n^3p\right)$	$\Theta\left(\frac{46}{3}n^3p\right)$
II	BSTRI_RV	$\Theta\left(\frac{1}{2}n^3p^2\right)$	$\Theta\left(\frac{1}{2}n^3p^2\right)$	$\Theta\left(n^3p^2\right)$
	BSTRI_CV	$\Theta\left(n^3p^2\right)$	$\Theta\left(n^3p^2\right)$	$\Theta\left(2n^3p^2\right)$
III	BSOI	$\Theta\left(3n^3p^2\right)$	$\Theta\left(3n^3p^2\right)$	$\Theta\left(6n^3p^2\right)$

To sum up,

$$\boxed{\text{Total complexity} = \Theta\left(\frac{7}{2}nN^2\right)}$$

Operation counts

Phase	Routine	Additions	Multiplications	Total Flops
I	BSOF	$\Theta\left(\frac{23}{3}n^3p\right)$	$\Theta\left(\frac{23}{3}n^3p\right)$	$\Theta\left(\frac{46}{3}n^3p\right)$
II	BSTRI_RV	$\Theta\left(\frac{1}{2}n^3p^2\right)$	$\Theta\left(\frac{1}{2}n^3p^2\right)$	$\Theta\left(n^3p^2\right)$
	BSTRI_CV	$\Theta\left(n^3p^2\right)$	$\Theta\left(n^3p^2\right)$	$\Theta\left(2n^3p^2\right)$
III	BSOI	$\Theta\left(3n^3p^2\right)$	$\Theta\left(3n^3p^2\right)$	$\Theta\left(6n^3p^2\right)$

To sum up,

$$\boxed{\text{Total complexity} = \Theta\left(\frac{7}{2}nN^2\right)}$$

- Hardware

Single-GPU . . *Hybrid node at UCD*

- 2 × Intel Xeon X5670
  - 6-cores, 2.9GHz
- 1 × NVIDIA GTX480
  - 15 SMs × 32 cores

Multi-GPU *Dirac cluster at NERSC*

- 2 × Intel Xeon E5530
  - 4-cores, 2.4 GHz
- 4 × NVIDIA Tesla C1060
  - 240 cores

- Software
  - POSIX threads for threading in step 2 of `BSOFTRI`
  - CuBlas, Magma, and Intel's MKL for LAPACK interface
- Codes
  - stand-alone CPU, stand-alone GPU, and hybrid CPU+GPU implementations
  - publicly available at `https://github.com/SGo-Go/BSOFI`

Performance tuning approach

benchmark LAPACK kernels $\rightarrow$ approximate parameters of PM $\rightarrow$ embed approximates into the code

CPU/GPU performance ratio parameters (κ_R, κ_C, and κ_Q)

- 1st order rational functions of n
- Gauss-Markov estimator

Switching (l_F) and correction parameters (c_i, c_j, c'_k, and c''_k)

- step-wise functions of n
- 1D filtering + rounding

Performance tuning approach

benchmark LAPACK kernels $\rightarrow$ approximate parameters of PM $\rightarrow$ embed approximates into the code

CPU/GPU performance ratio parameters (κ_R, κ_C, and κ_Q)

- 1st order rational functions of n
- Gauss-Markov estimator

Switching (l_F) and correction parameters (c_i, c_j, c'_k, and c''_k)

- step-wise functions of n
- 1D filtering + rounding