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

Sergiy Gogolenko

Ukrainian Catholic University, Lviv
2016, May 23

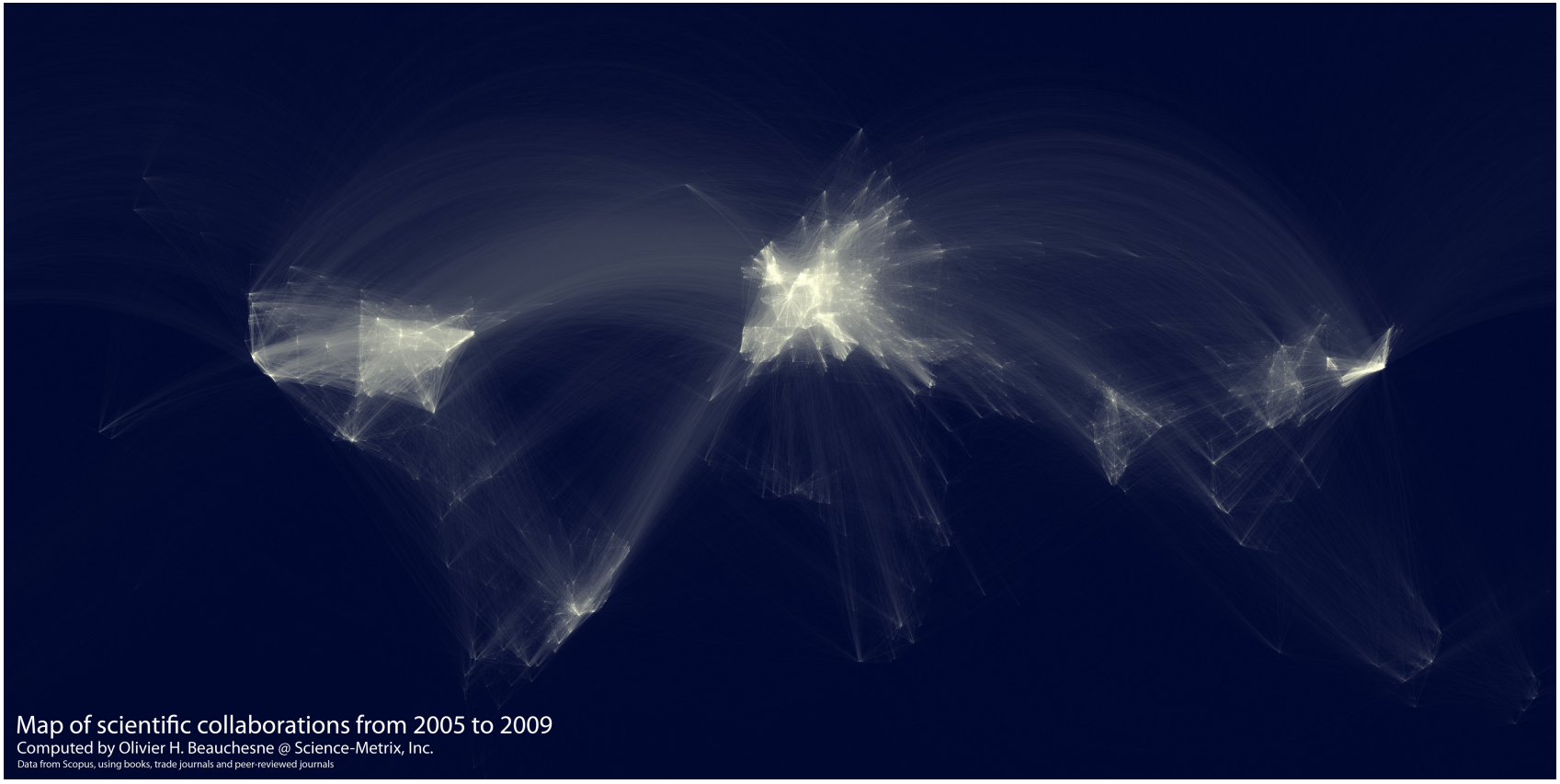
Outline

Introduction

Sources of big graphs

Introduction

Sources of big graphs



PageRank

Sergey Brin

Larry Page

The Anatomy of a Large-Scale Hypertextual Web Search Engine
Computer Science Department, Stanford University
1998

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.

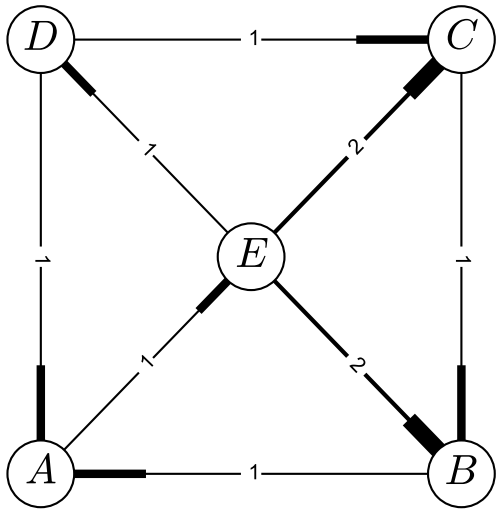
Probability of visiting page by idealized random Web surfer

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

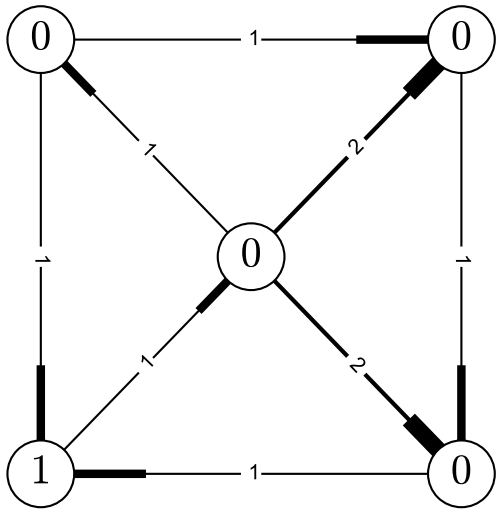
Probability evolution

$$\Pr'(x) = \sum_{y \to x} \frac{\Pr(y)}{\deg^+(y)}$$

- $\deg^+(y)$ – outdegree of y

PageRank: The model

Random surfing as Markov process



Probabilities

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

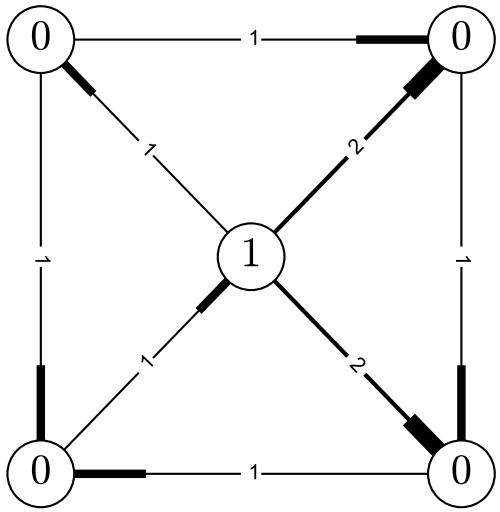
Probability evolution

$$\Pr'(x) = \sum_{y \to x} \frac{\Pr(y)}{\deg^+(y)}$$

- $\deg^+(y)$ – outdegree of y

PageRank: The model

Random surfing as Markov process



Probabilities

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

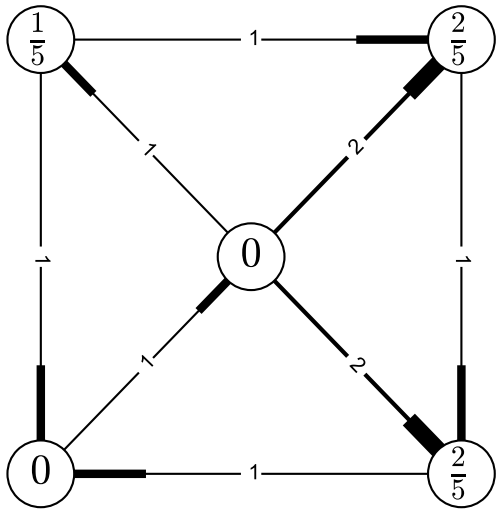
Probability evolution

$$\Pr'(x) = \sum_{y \to x} \frac{\Pr(y)}{\deg^+(y)}$$

- $\deg^+(y)$ – outdegree of y

PageRank: The model

Random surfing as Markov process



Probabilities

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

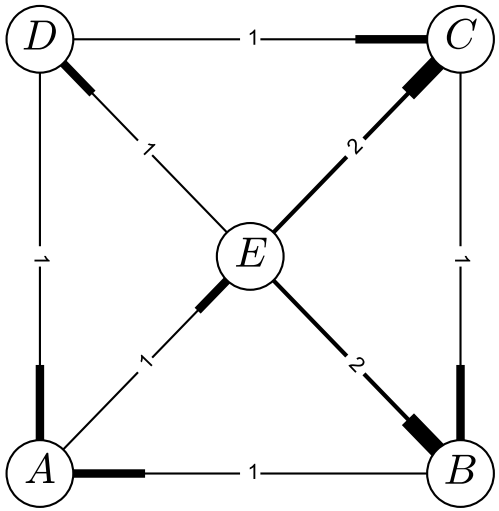
Probability evolution

$$\Pr'(x) = \sum_{y \to x} \frac{\Pr(y)}{\deg^+(y)}$$

- $\deg^+(y)$ – outdegree of y

PageRank: The model

Transition matrix of the Web



$$M = \begin{bmatrix} 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{2}{5} \\ 0 & 0 & 0 & 0 & \frac{1}{5} \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix-vector representation

$$\mathrm{Pr}' = M \cdot \mathrm{Pr}$$

- M:

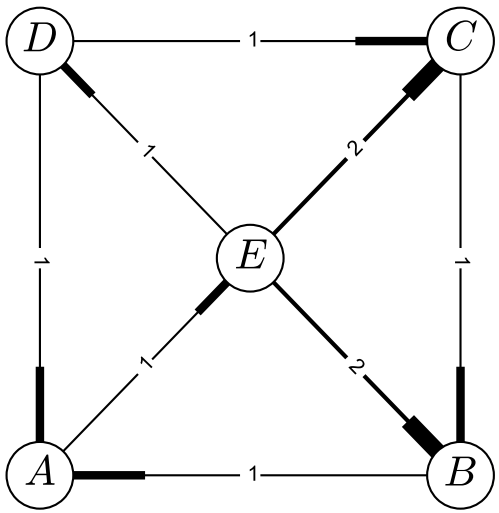
$$m_{xy} = \begin{cases} 1/\deg^+(y), & y \to x \\ 0, & y \not\to x \end{cases}$$

M is stochastic matrix

$$\forall y : \sum_x m_{xy} = 1$$

PageRank: The model

Simulation of random surfer



Simulation of random surfer

$\Pr^i$ probability distribution for the location of a random surfer at step i

$$\Pr^0 = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T \quad (1)$$

$$\Pr^{i+1} = M \cdot \Pr^i \quad (2)$$

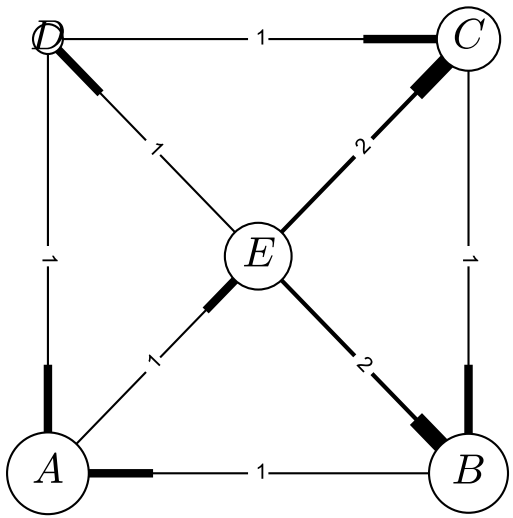
Restrictions

Strongly connected graph

- no dead-ends
- no spider traps

PageRank: The model

Simulation of random surfer



Simulation of random surfer

$$\begin{bmatrix} \frac{3}{10} \\ \frac{7}{25} \\ \frac{9}{50} \\ \frac{1}{25} \\ \frac{1}{5} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{2}{5} \\ 0 & 0 & 0 & 0 & \frac{1}{5} \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \end{bmatrix}$$

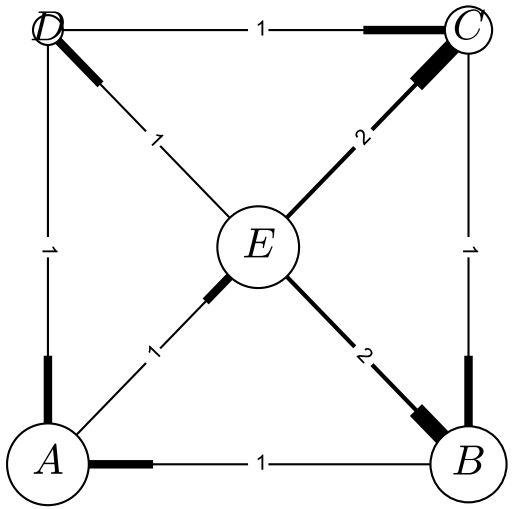
Restrictions

Strongly connected graph

- no dead-ends
- no spider traps

PageRank: The model

Simulation of random surfer



Simulation of random surfer

$$\begin{bmatrix} \frac{3}{10} \\ \frac{13}{50} \\ \frac{1}{10} \\ \frac{1}{25} \\ \frac{3}{10} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{2}{5} \\ 0 & 0 & 0 & 0 & \frac{1}{5} \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{3}{10} \\ \frac{7}{25} \\ \frac{9}{50} \\ \frac{1}{25} \\ \frac{1}{5} \end{bmatrix}$$

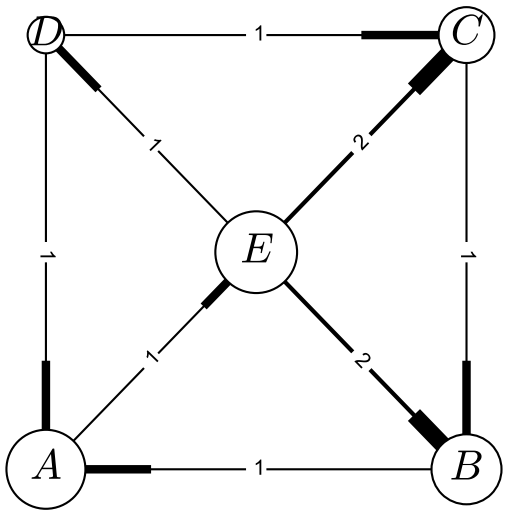
Restrictions

Strongly connected graph

- no dead-ends
- no spider traps

PageRank: The model

Simulation of random surfer



Simulation of random surfer

$$\begin{bmatrix} \frac{7}{25} \\ \frac{11}{50} \\ \frac{7}{50} \\ \frac{3}{50} \\ \frac{3}{10} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{2}{5} \\ 0 & 0 & 0 & 0 & \frac{1}{5} \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{3}{10} \\ \frac{13}{50} \\ \frac{1}{10} \\ \frac{1}{25} \\ \frac{3}{10} \end{bmatrix}$$

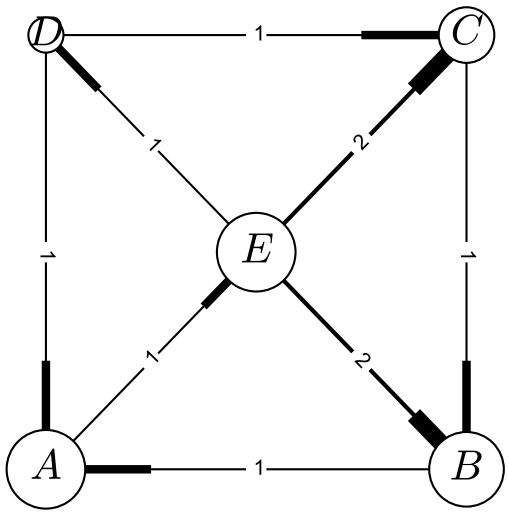
Restrictions

Strongly connected graph

- no dead-ends
- no spider traps

PageRank: The model

Simulation of random surfer



Simulation of random surfer

$$\underbrace{\begin{bmatrix} 0.28 \\ 0.25 \\ 0.14 \\ 0.06 \\ 0.28 \end{bmatrix}}_{\mathrm{Pr}^{\infty}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 & 0 & \frac{2}{5} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{2}{5} \\ 0 & 0 & 0 & 0 & \frac{1}{5} \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{M} \times \underbrace{\begin{bmatrix} 0.28 \\ 0.25 \\ 0.14 \\ 0.06 \\ 0.28 \end{bmatrix}}_{\mathrm{Pr}^{\infty}}$$

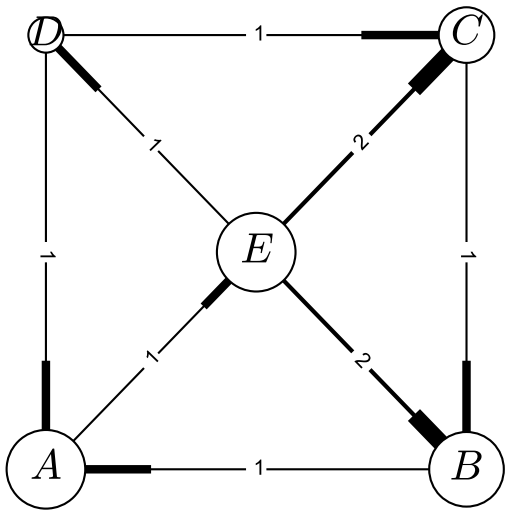
Restrictions

Strongly connected graph

- no dead-ends
- no spider traps

PageRank: The model

Simulation of random surfer



Simulation of random surfer

$$M \cdot \mathrm{Pr}^{\infty} = 1 \cdot \mathrm{Pr}^{\infty}$$

Pr^{∞} is eigenvector of M with eigenvalue $\lambda = 1$

Restrictions

Strongly connected graph

- no dead-ends
- no spider traps

PageRank: The model

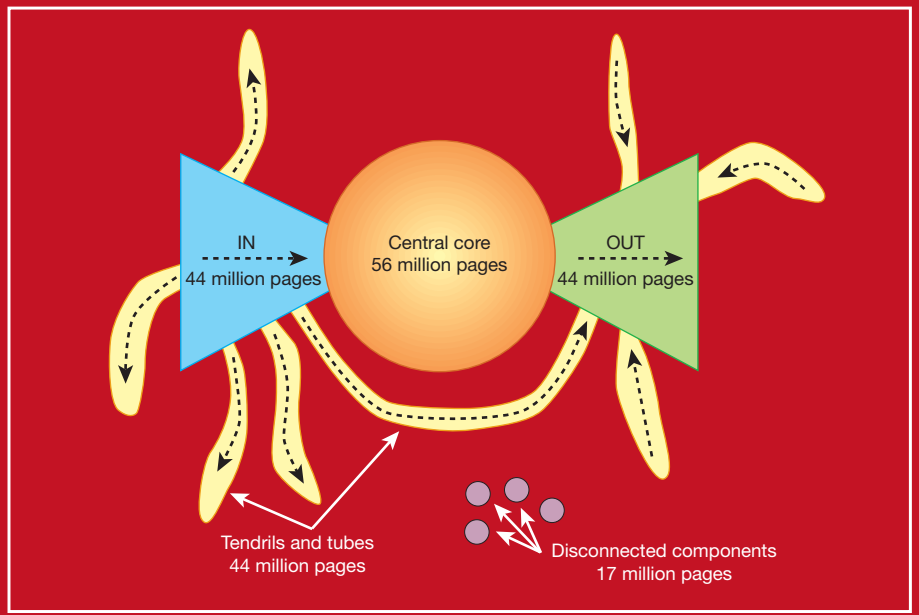
Structure of the Web

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

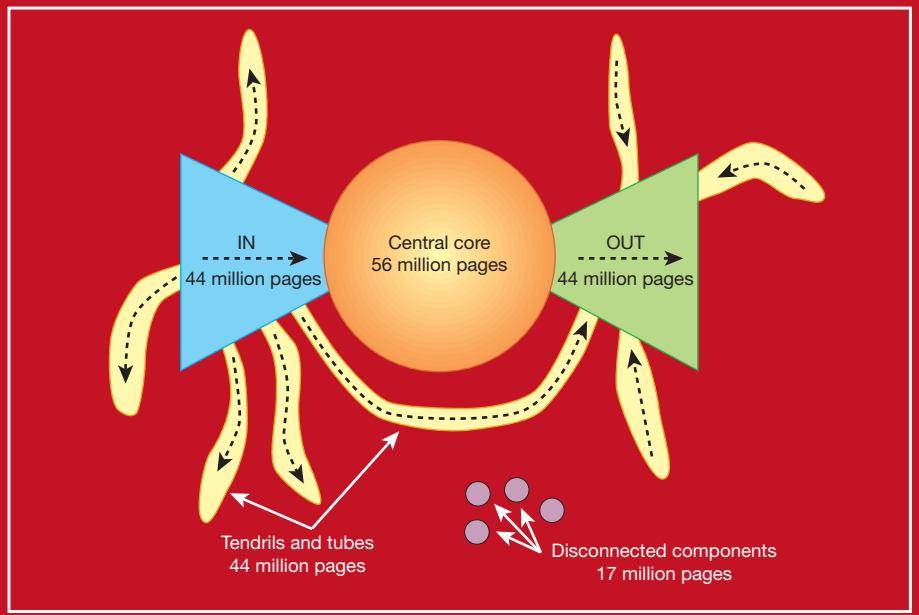
Structure of the Web

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"

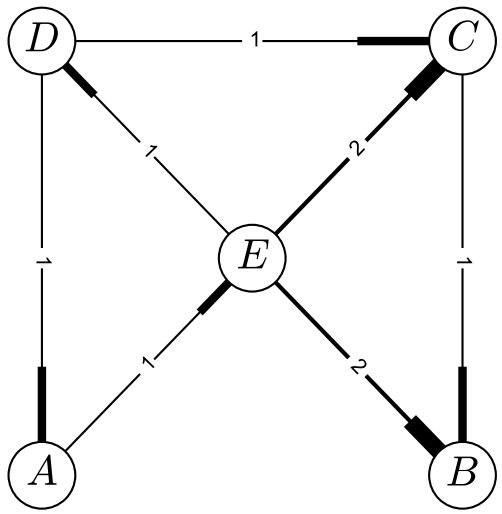
- in-component
- out-component
- tendrils
  - tubes
  - isolated components

Problems

- dead-ends
- spider traps

PageRank: The model

Avoiding dead ends by dropping



Handling dead ends

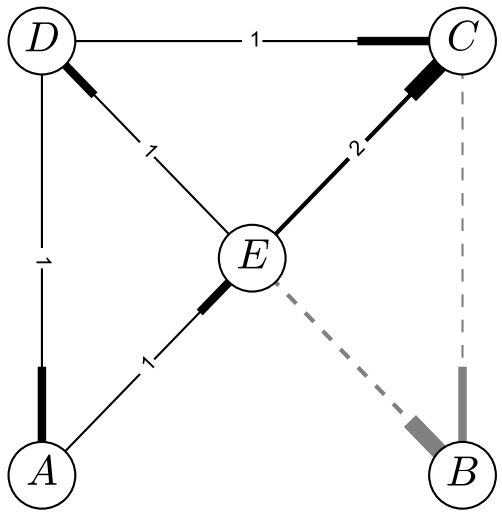
- $\Pr = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} \frac{1}{4} & * & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & * & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & 0.42 & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$

Algorithm: dropping dead ends (for substochastic M)

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



Handling dead ends

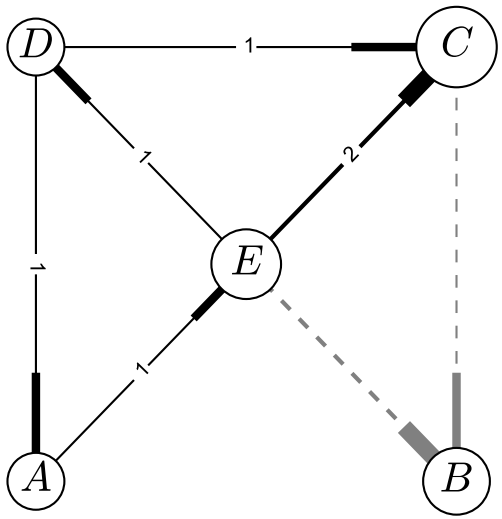
- $\Pr = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} \frac{1}{4} & * & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & * & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & 0.42 & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$

Algorithm: dropping dead ends (for substochastic M)

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



Handling dead ends

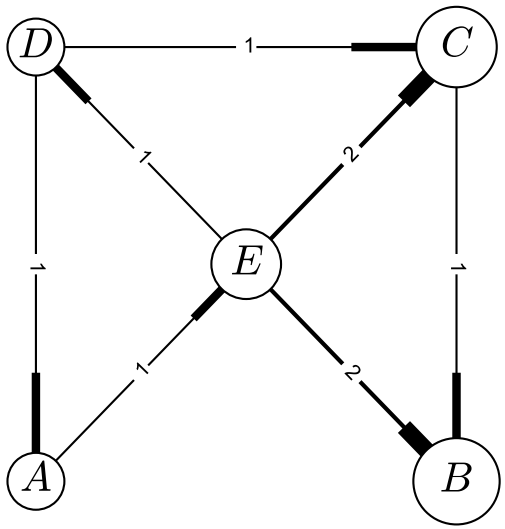
- $\Pr = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} \frac{1}{4} & * & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & * & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & 0.42 & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$

Algorithm: dropping dead ends (for substochastic M)

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



Handling dead ends

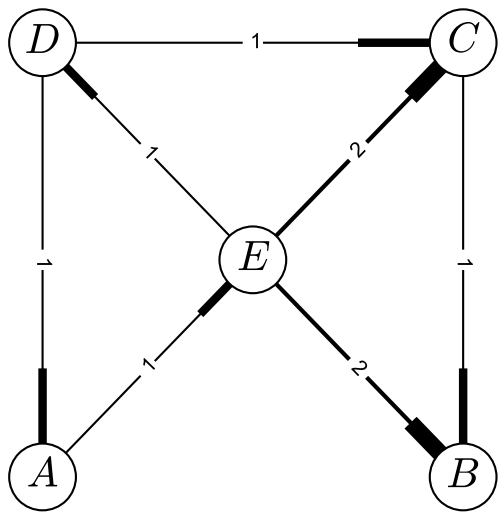
- $\Pr = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} \frac{1}{4} & * & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & * & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$
- $\Pr = \begin{bmatrix} 0.18 & 0.42 & 0.36 & 0.18 & 0.27 \end{bmatrix}^T$

Algorithm: dropping dead ends (for substochastic M)

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

PageRank: The model

Teleporting



PageRanks with $\beta = 0.8$

- $\mathrm{Pr}^0 = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T$
- $\mathrm{Pr}^\infty = \begin{bmatrix} 0.14 & 0.32 & 0.21 & 0.12 & 0.2 \end{bmatrix}^T$

Idea

Introduce small probability $1 - \beta$ of teleporting to a random page (usually $0 < 1 - \beta \leq 0.2$)

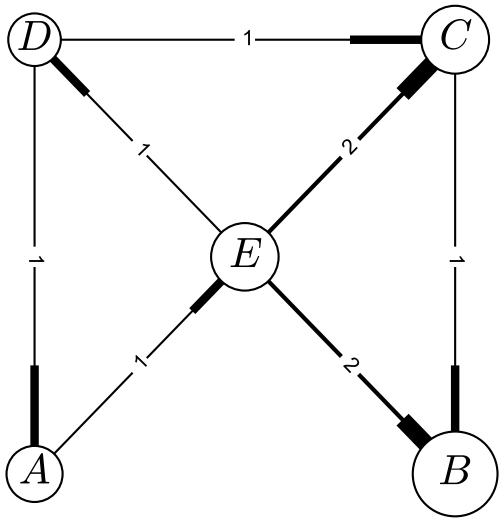
$$\mathrm{Pr}' = \beta \cdot M \cdot \mathrm{Pr} + (1 - \beta)\frac{\mathbb{1}}{n} \quad (1)$$

$$\mathrm{Pr}'(x) = \frac{1 - \beta}{n} + \beta \sum_{y \to x} \frac{\mathrm{Pr}(y)}{\deg^+(y)} \quad (2)$$

- Brin and Page: 50-100 iterations to converge

PageRank: The model

Teleporting



PageRanks with $\beta = 0.8$

- $\Pr^0 = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}^T$
- $\Pr^\infty = \begin{bmatrix} 0.14 & 0.32 & 0.21 & 0.12 & 0.2 \end{bmatrix}^T$

Idea

Introduce small probability $1 - \beta$ of teleporting to a random page (usually $0 < 1 - \beta \leq 0.2$)

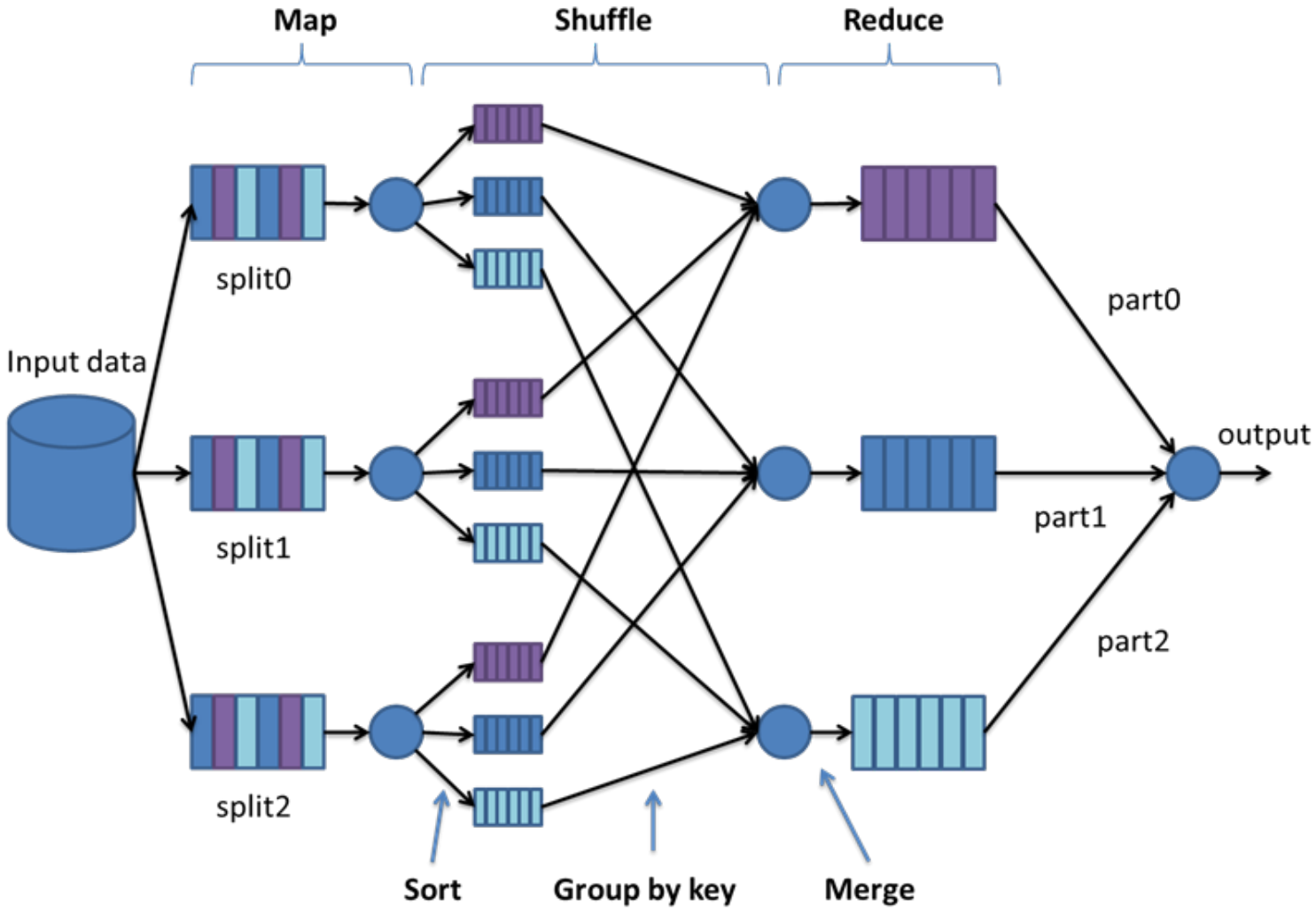
$$\Pr' = \beta \cdot M \cdot \Pr + (1 - \beta)\frac{\mathbb{1}}{n} \quad (1)$$

$$\Pr'(x) = \frac{1 - \beta}{n} + \beta \sum_{y \to x} \frac{\Pr(y)}{\deg^+(y)} \quad (2)$$

- Brin and Page: 50-100 iterations to converge

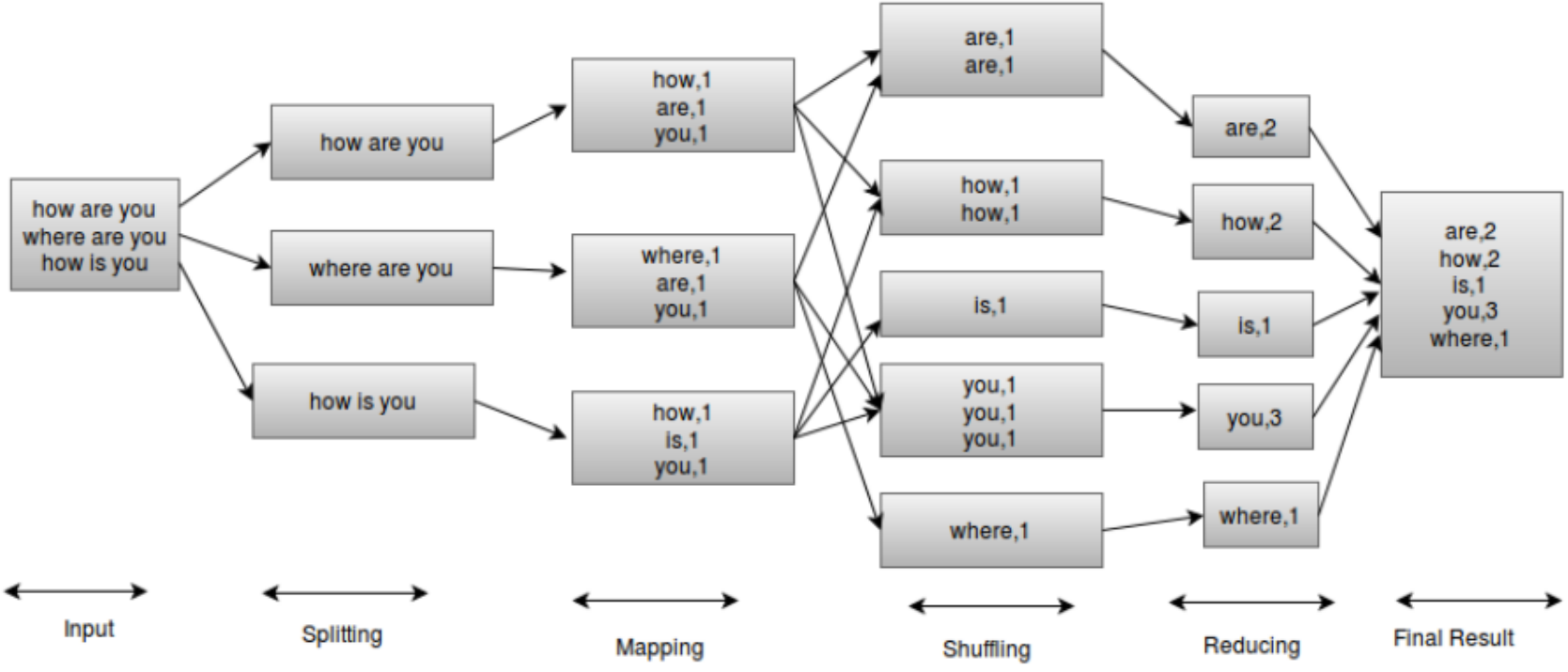
PageRank: MapReduce

MapReduce workflow



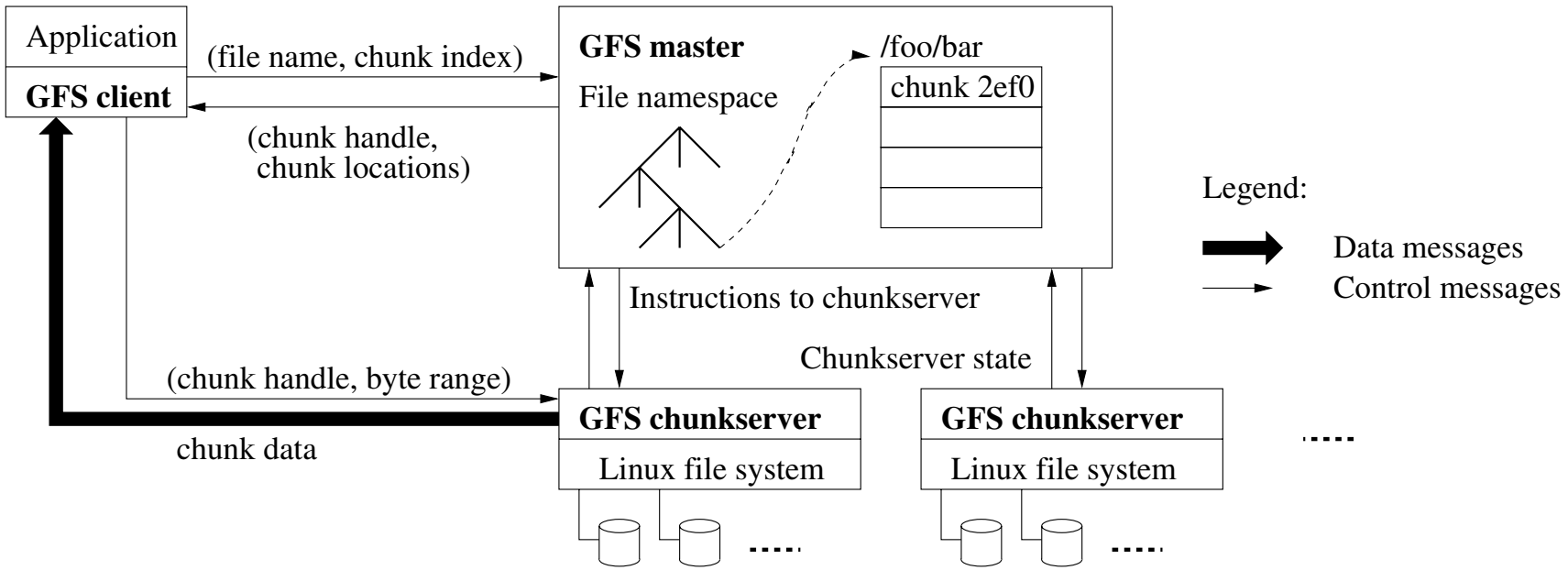
PageRank: MapReduce

Word count with MapReduce



PageRank: MapReduce

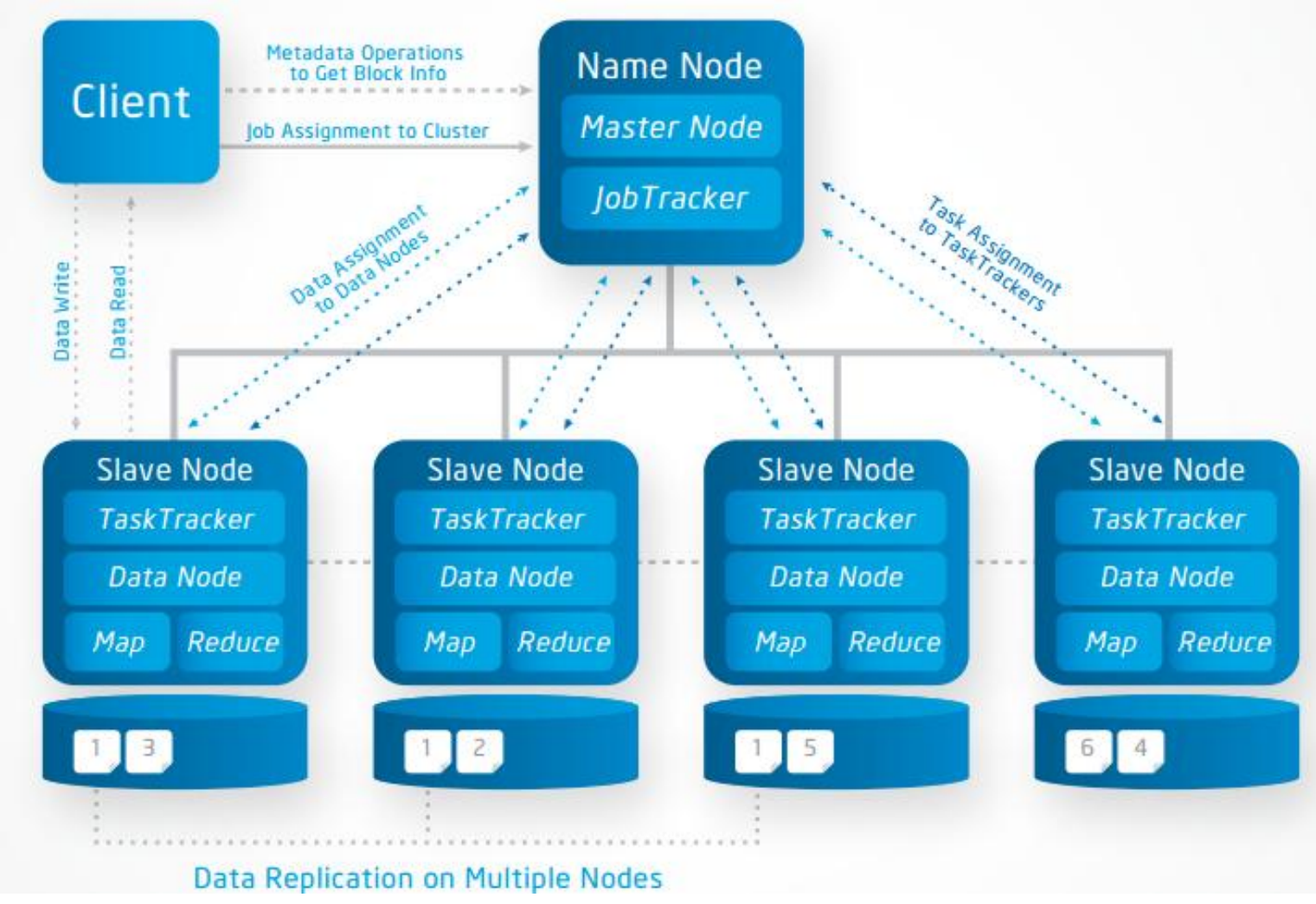
GFS



©Ghemawat, S.; Gobioff, H.; Leung, S.-T. *The Google File System*, Google

PageRank: MapReduce

GFS



PageRank: MapReduce

PageRank using MapReduce

Mapper (node y)

Data: $\langle y | \{x_1, ..., x_k\}, \Pr(y) \rangle$ (node – outlinks)

1 **for** $j \in \{1, ..., k\}$ **do**

2 emit $\left\langle x_j \middle| \frac{\Pr(y)}{\deg^+(y)} \right\rangle$;

3 emit $\langle y | \{x_1, ..., x_k\} \rangle$;

Reducer (node x)

Data: $\left\langle x \middle| \{\frac{\Pr(y_1)}{\deg^+(y_1)}, ..., \frac{\Pr(y_l)}{\deg^+(y_l)}, \{x_1, ..., x_k\}\} \right\rangle$ (node – $\Delta \Pr$)

1 $\Pr(x) \leftarrow \frac{1-\beta}{n} + \beta \sum_{i=1}^{l} \frac{\Pr(y_i)}{\deg^+(y_i)}$

PageRank: MapReduce

PageRank using MapReduce

Mapper (node y)

Data: $\langle y | \{x_1, ..., x_k\}, \Pr(y) \rangle$ (node – outlinks)

1 **for** $j \in \{1, ..., k\}$ **do**

2 emit $\left\langle x_j \middle| \frac{\Pr(y)}{\deg^+(y)} \right\rangle$;

3 emit $\langle y | \{x_1, ..., x_k\} \rangle$;

Reducer (node x)

Data: $\left\langle x \middle| \{\frac{\Pr(y_1)}{\deg^+(y_1)}, ..., \frac{\Pr(y_l)}{\deg^+(y_l)}, \{x_1, ..., x_k\}\} \right\rangle$ (node – $\Delta \Pr$)

1 $\Pr(x) \leftarrow \frac{1-\beta}{n} + \beta \sum_{i=1}^{l} \frac{\Pr(y_i)}{\deg^+(y_i)}$

PageRank: Modifications

Topic-sensitive PageRank

How to organize private PageRank for each user?

- classify users by interest in each of the selected topics
- one Pr vector for each of some small number of topics
- biase the PageRank to favor pages of that topic

$$\mathrm{Pr}' = \beta \cdot M \cdot \mathrm{Pr} + (1 - \beta)\frac{\mathbb{1}_S}{|S|}$$

- S – set of pages belonging to a certain topic (teleport set)

PageRank: Modifications

Topic-sensitive PageRank

How to organize private PageRank for each user?

- classify users by interest in each of the selected topics
- one Pr vector for each of some small number of topics
- biase the PageRank to favor pages of that topic

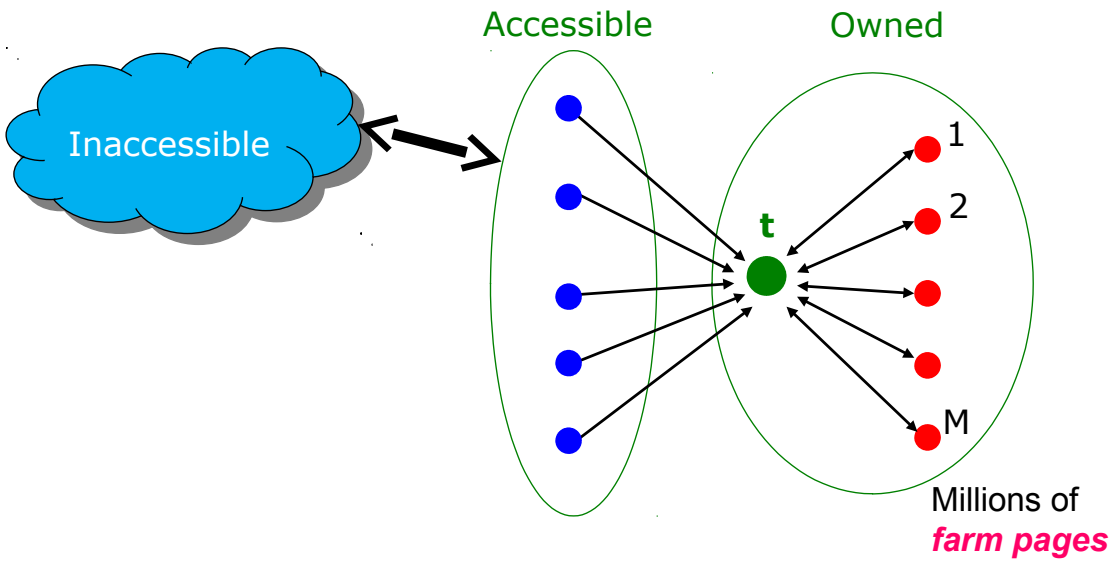
$$\mathrm{Pr}' = \beta \cdot M \cdot \mathrm{Pr} + (1 - \beta)\frac{\mathbb{1}_S}{|S|}$$

  - S – set of pages belonging to a certain topic (teleport set)

PageRank: Modifications

Link spam and TrustRank

> 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.
>
> ©Jeffrey D. Ullman



- n pages in total
- m supporting pages
- 1 target page t

$$\Pr(t) = \Pr(\text{access.}) + \beta m \underbrace{\left(\frac{\beta \Pr(t)}{m} + \frac{1-\beta}{n} \right)}_{\Pr(\text{supporting page})} \approx \frac{\Pr(\text{access.})}{1-\beta^2} + \frac{\beta}{1+\beta} \frac{m}{n}$$

PageRank: Modifications

Link spam and TrustRank

TrustRank

Topic-sensitive PageRank, where the "topic" is a set of pages believed to be trustworthy

- suitable teleport set:
  domain whose membership is controlled (e.g., .edu)
- spam mass

$$\text{SpamMass}(y) = \frac{\text{PageRank}(y) - \text{TrustRank}(y)}{\text{PageRank}(y)}$$

negative (small positive) $\text{SpamMass}(y) \implies y$ is probably not a spam

PageRank: Modifications

Link spam and TrustRank

TrustRank

Topic-sensitive PageRank, where the "topic" is a set of pages believed to be trustworthy

- suitable teleport set:
  domain whose membership is controlled (e.g., .edu)
- spam mass

$$\text{SpamMass}(y) = \frac{\text{PageRank}(y) - \text{TrustRank}(y)}{\text{PageRank}(y)}$$

negative (small positive) $\text{SpamMass}(y) \implies y$ is probably not a spam

PageRank as an orthogonal iteration

Eigenvectors & convergence of OI

Definition

Eigenvalues $(\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n)$ and eigenvectors:

$$Av_i = \lambda_i v_i$$

Convergence of orthogonal iteration

Let $\lambda_i \neq \lambda_j$, $v_i^* \cdot v_j = 0$, $\|v_i\| = 1$.
Take arbitrary $u = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n$

$$\begin{aligned} A^k u &= c_1 A^k v_1 + c_2 A^k v_2 + \cdots + c_n A^k v_n \\ &= c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \cdots + c_n \lambda_n^k v_n \\ &= c_1 \lambda_1^k \left(v_1 + \frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1} \right)^k v_2 + \cdots + \frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1} \right)^k v_n \right) \end{aligned}$$

$$v_1 \approx \frac{1}{c_1 \lambda_1^k} A^k u + \mathcal{O}\left(\frac{\lambda_2}{\lambda_1} \right)^k$$

PageRank as an orthogonal iteration

PageRank convergence

Find v_i: $Av_i = \lambda_i v_i$

Find Pr: $M \cdot \mathrm{Pr} = 1 \cdot \mathrm{Pr}$

Orthogonal iterations

Data: A, k
Result: Q

1 Choose initial guess for Q (random orthonormal $n \times k$ matrix) ;
2 **while** $\underline{\text{error}(Q) > \epsilon}$ **do**
3 $\quad V \leftarrow AQ$;
4 $\quad Q \leftarrow \texttt{Orthonormalize}(V)$

PageRank iterations

Data: M, 1
Result: Pr

1 Choose initial guess for Pr ($\mathrm{Pr} = \mathbb{1}/n$) ;
2 **while** $\underline{\text{error}(\mathrm{Pr}) > \epsilon}$ **do**
3 $\quad \mathrm{Pr} \leftarrow M \cdot \mathrm{Pr}$;
4 $\quad$ ~~Orthonormalize(Pr)~~

PageRank convergence estimates

- $\lambda_1 = 1, v_1 = \mathrm{Pr} \implies$ convergence depends on λ_2: $\epsilon = \mathcal{O}(\lambda_2^k)$
- link structure of the Web $\lambda_2 \approx 0.9$: $\lambda_2^k = 0.9^{328} \approx 10^{-15} = \epsilon$

PageRank as an orthogonal iteration

PageRank convergence

Find v_i: $Av_i = \lambda_i v_i$

Find Pr: $M \cdot \mathrm{Pr} = 1 \cdot \mathrm{Pr}$

Orthogonal iterations

Data: A, k
Result: Q

1 Choose initial guess for Q (random orthonormal $n \times k$ matrix) ;
2 **while** $\underline{\text{error}(Q) > \epsilon}$ **do**
3 $\quad V \leftarrow AQ$;
4 $\quad Q \leftarrow \texttt{Orthonormalize}(V)$

PageRank iterations

Data: M, 1
Result: Pr

1 Choose initial guess for Pr ($\mathrm{Pr} = \mathbb{1}/n$) ;
2 **while** $\underline{\text{error}(\mathrm{Pr}) > \epsilon}$ **do**
3 $\quad \mathrm{Pr} \leftarrow M \cdot \mathrm{Pr}$;
4 $\quad$ ~~Orthonormalize(Pr)~~

PageRank convergence estimates

- $\lambda_1 = 1, v_1 = \mathrm{Pr} \implies$ convergence depends on λ_2: $\boxed{\epsilon = \mathcal{O}(\lambda_2^k)}$
- link structure of the Web $\lambda_2 \approx 0.9$: $\lambda_2^k = 0.9^{328} \approx 10^{-15} = \epsilon$

Decentralized OI

David Kempe

Frank McSherry

A Decentralized Algorithm for Spectral Analysis

Department of Computer Science and Engineering, University of Washington

2006

Decentralized OI

Motivation

Can we...?

- Can we omit masted nodes? (master fault is critical)
- Can we compute more eigenvectors and handle more general graph matrices?

Min-cut approximation

$$v_{n-1} = \arg\min_{\|x\|=1} x^T \mathcal{L} x$$

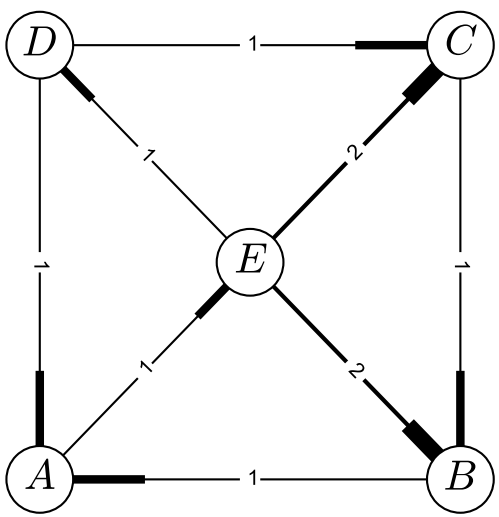
$$\mathcal{L} = D - A = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & -2 \\ 0 & -1 & 4 & -1 & -2 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & -2 & -2 & -1 & 6 \end{bmatrix}$$

Decentralized OI

Motivation

Can we...?

- Can we omit masted nodes? (master fault is critical)
- Can we compute more eigenvectors and handle more general graph matrices?



Min-cut approximation

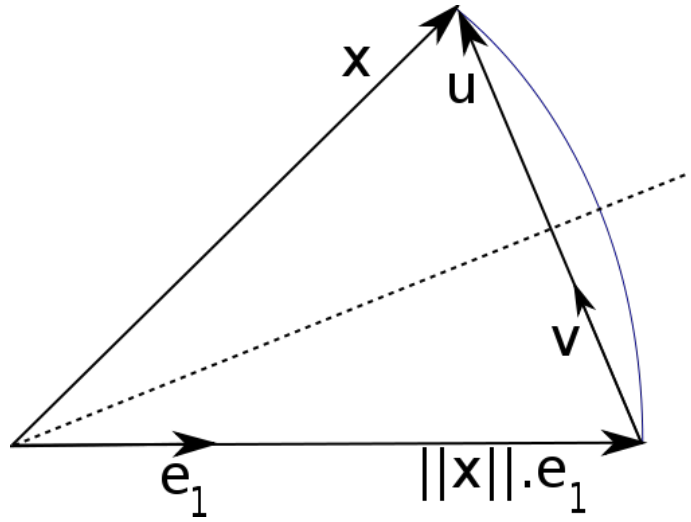
$$v_{n-1} = \arg\min_{\|x\|=1} \mathbf{x}^T \mathcal{L} \mathbf{x}$$

$$\mathcal{L} = D - A = \begin{bmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & -2 \\ 0 & -1 & 4 & -1 & -2 \\ -1 & 0 & -1 & 3 & -1 \\ -1 & -2 & -2 & -1 & 6 \end{bmatrix}$$

Decentralized OI

Orthonormalization in OI

QR-factorization: $QR = V$



Orthogonal iterations

Data: A, k
Result: Q

1 Choose initial guess for Q (random orthonormal $n \times k$ matrix) ;
2 **while** $\underline{\text{error}(Q) > \epsilon}$ **do**
3 $\quad V \leftarrow AQ$;
4 $\quad Q \leftarrow \texttt{Orthonormalize}(V)$

Householder reflection

$$Q_1 = I - 2\mathbf{v}\mathbf{v}^T$$

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|}, \mathbf{u} = \mathbf{x} - \alpha_1 \mathbf{e}_1$$

$$Q_1 \mathbf{x}_1 = (\alpha_1, 0, \cdots, 0)^T$$

Decentralized OI

Sketch of the Kempe-McSherry algorithm

QR-factorization: $QR = V$

$$K = V^T V = (QR)^T (QR)$$
$$= R^T Q^T QR = R^T R$$

Orthogonal iterations

Data: A, k
Result: Q

1 Choose initial guess for Q (random orthonormal $n \times k$ matrix) ;
2 **while** $\underline{\text{error}(Q) > \epsilon}$ **do**
3 $\quad V \leftarrow AQ$;
4 $\quad Q \leftarrow \texttt{Orthonormalize}(V)$

Decentralized OI (on node x)

Data: A_x, k
Result: Q_x

1 Choose initial guess for Q_x (k-dimensional vector) ;
2 **while** $\underline{\max(\text{error}(Q_x)) > \epsilon}$ **do**
3 $\quad V_x \leftarrow \sum_{x \to y} a_{xy} Q_y$;
4 $\quad K^{(x)} \leftarrow V_x^T V_x$;
5 $\quad K \leftarrow \texttt{PushSum}(K^{(x)})$;
6 $\quad$ Cholesky fact.: $R^T R = K$;
7 $\quad Q_x \leftarrow V_x R^{-1}$

Decentralized OI

Convergence

B stochastic matrix with ergodic and reversible Markov Chain:

$$b_{xy} = \begin{cases} 1/\deg(x), & x \to y \\ 0, & x \not\to y \end{cases}$$

PushSum (on node x)

Data: $K^{(x)}$, B

Result: $K = S_x/w_x$

1 $w_x = \begin{cases} 1, & x = \hat{x} \\ 0, & x \neq \hat{x} \end{cases}$;

2 $S_x \leftarrow K^{(x)}$;

3 **while** large error **do**

4 $\quad S_x \leftarrow \sum_{y \to x} b_{yx} S_y$;

5 $\quad w_x \leftarrow \sum_{y \to x} b_{yx} w_y$;

Decentralized OI

PushSum protocol

Convergence of decentralized OI

It converges essentially in $\mathcal{O}(\tau_{\text{mix}} \log^2 n)$ rounds of communication and computation, where τ_{mix} is the mixing time of a random walk on the network.

Thank you for your attantion!