Network Science

#6 Insights on PageRank

Measuring closeness with PageRank

How can we use PageRank?

Want to know about a specific topic? TopicSpecific PageRank

Want to measure proximity/similarity to a node? Local PageRank

… appropriately select your teleport vector ***q*** !

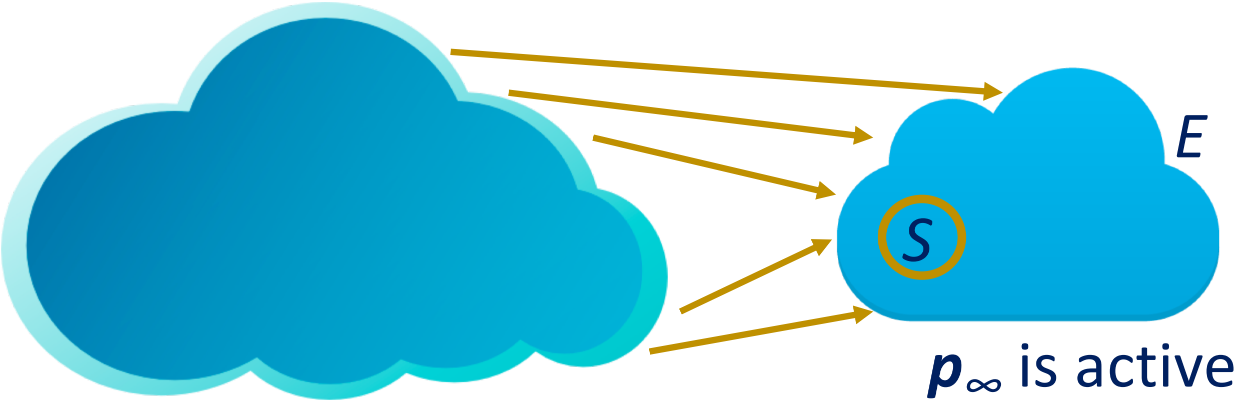
Topic specific PageRank

Idea

- Bias the random walk towards a topic specific teleport set *S* of nodes, i.e., make sure that $\boldsymbol{q}$ is active in *S* only
- *S* should contain only pages that are relevant to the topic

Result

- The random walk deterministically ends in a small set *E*, containing *S*, and being in some sense close to it



$\boldsymbol{p}_\infty$ is active in E only

Measuring closeness: Local PageRank

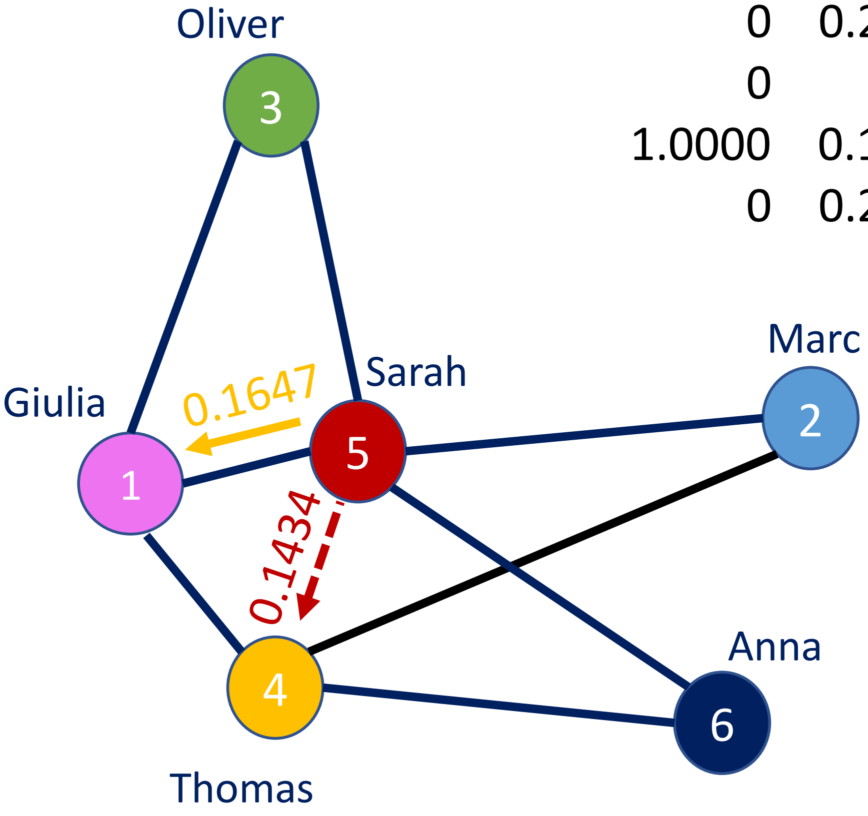
Idea

- Measure similarity / closeness to node i by applying TopicSpecific PageRank with teleport set $S=\{i\}$

Result

- Measures direct and indirect multiple connections, their quality, degree or weight

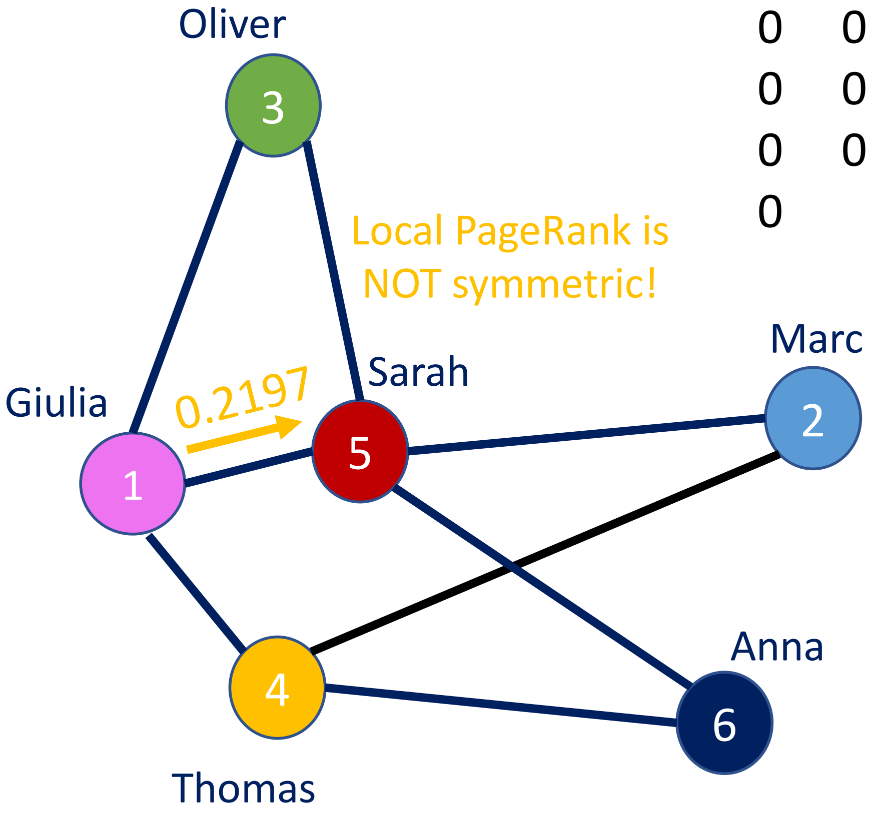
Example: who’s Sarah’s best friend?



t=1	2	3	4	5
0	0.2125	0.1222	0.2096	0.1290
0	0.2125	0.0319	0.1705	0.0708
0	0.2125	0.0921	0.1369	0.1127
0	0	0.2408	0.0617	0.2043
1.0000	0.1500	0.4811	0.2508	0.4125
0	0.2125	0.0319	0.1705	0.0708

10	20	50	75	100	
0.1743	0.1653	0.1647	0.1647	0.1647	Giulia
0.1238	0.1144	0.1138	0.1138	0.1138	Marc
0.1206	0.1199	0.1199	0.1199	0.1199	Oliver
0.1285	0.1426	0.1434	0.1434	0.1434	Thomas
0.3290	0.3435	0.3444	0.3444	0.3444	Sarah
0.1238	0.1144	0.1138	0.1138	0.1138	Anna

Example: who's Giulia's best friend?

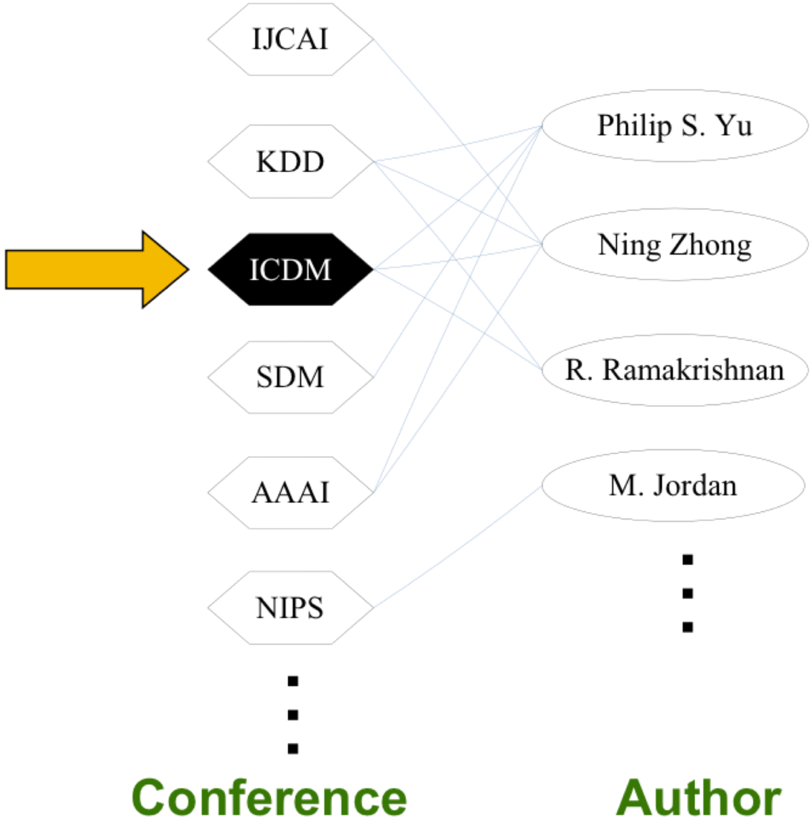


t=1	*2*	*3*	*4*	*5*
1.0000	0.1500	0.4109	0.2403	0.3404
0	0	0.1405	0.0467	0.1262
0	0.2833	0.1027	0.1510	0.1275
0	0.2833	0.0425	0.2358	0.1078
0	0.2833	0.1629	0.2795	0.1719
0	0	0.1405	0.0467	0.1262

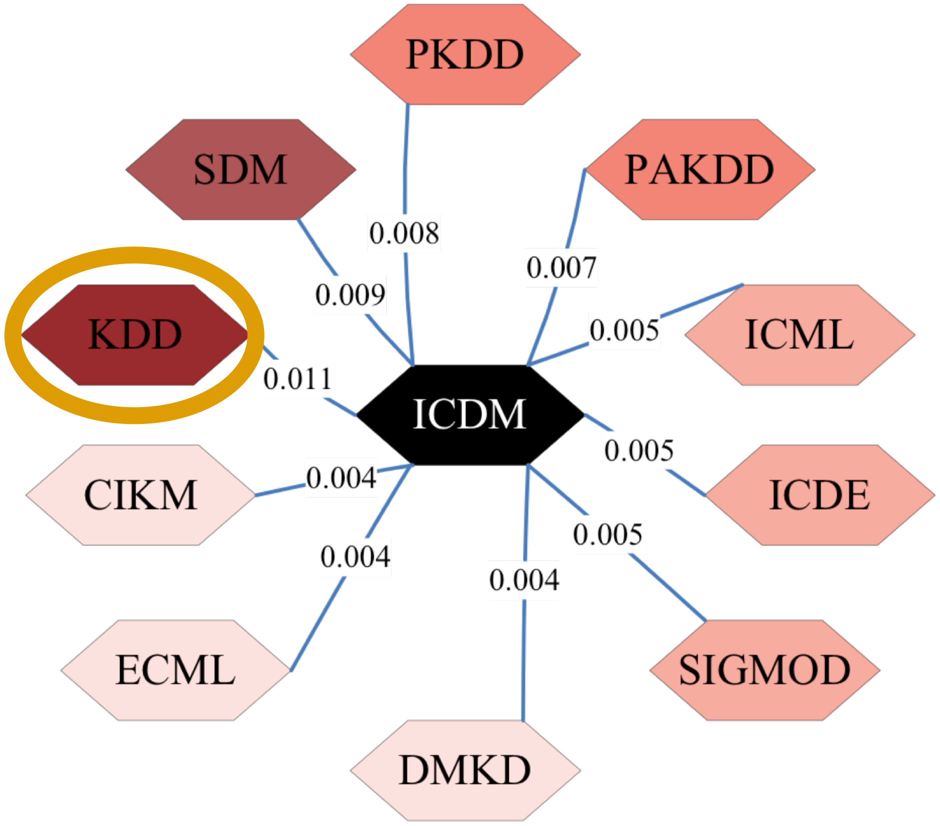
10	*20*	*50*	*75*	*100*	
0.2909	0.2985	0.2989	0.2989	0.2989	Giulia
0.0848	0.0926	0.0931	0.0931	0.0931	Marc
0.1309	0.1313	0.1314	0.1314	0.1314	Oliver
0.1763	0.1645	0.1638	0.1638	0.1638	Thomas
0.2324	0.2204	0.2197	0.2197	0.2197	Sarah
0.0848	0.0926	0.0931	0.0931	0.0931	Anna

Example

What is the most related conference to ICDM?



Top 10 ranking results

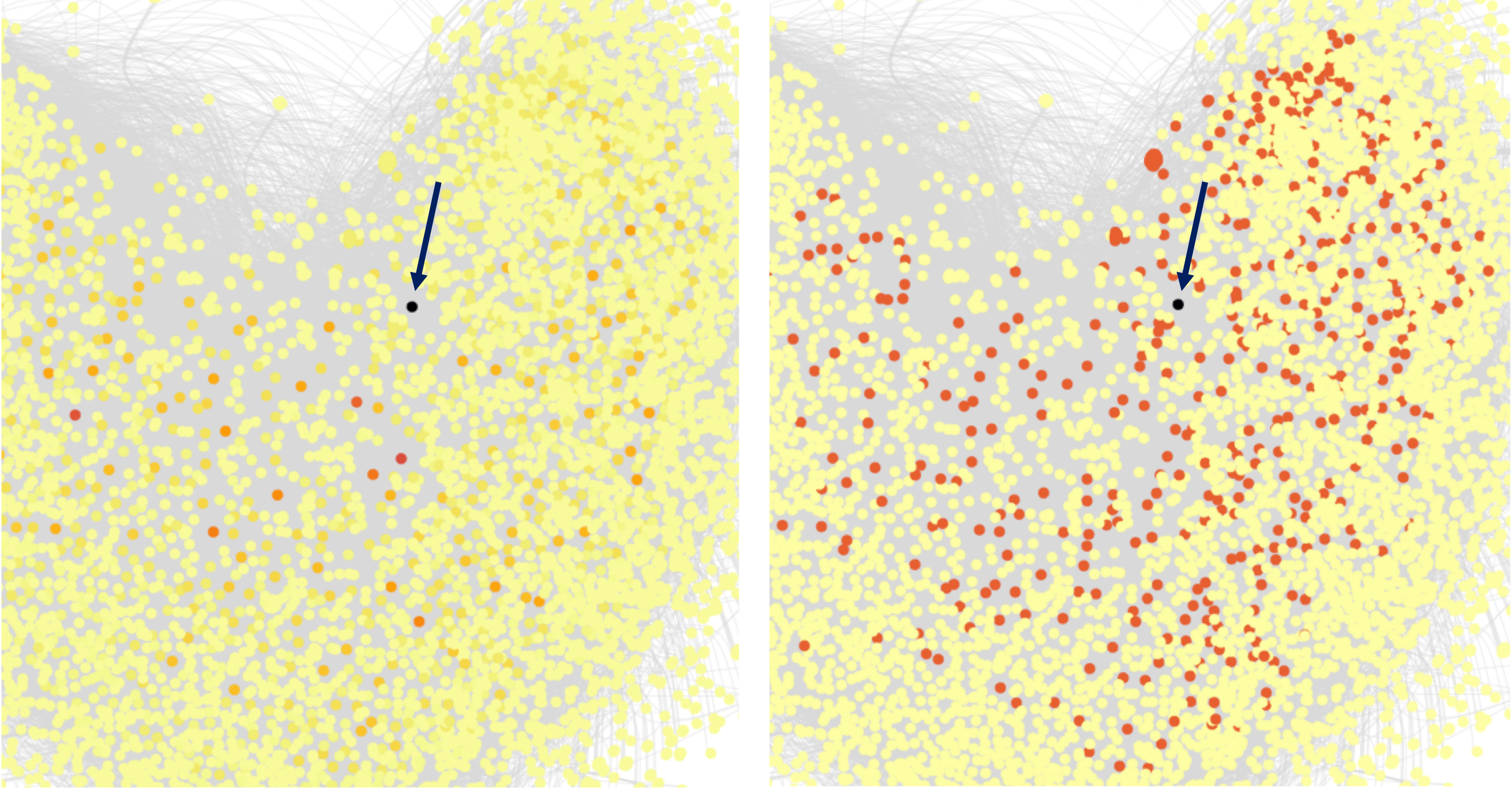


ICDM = international conf. on data mining
KDD = knowledge discovery and data mining

Local PageRank (authorities , *A*)

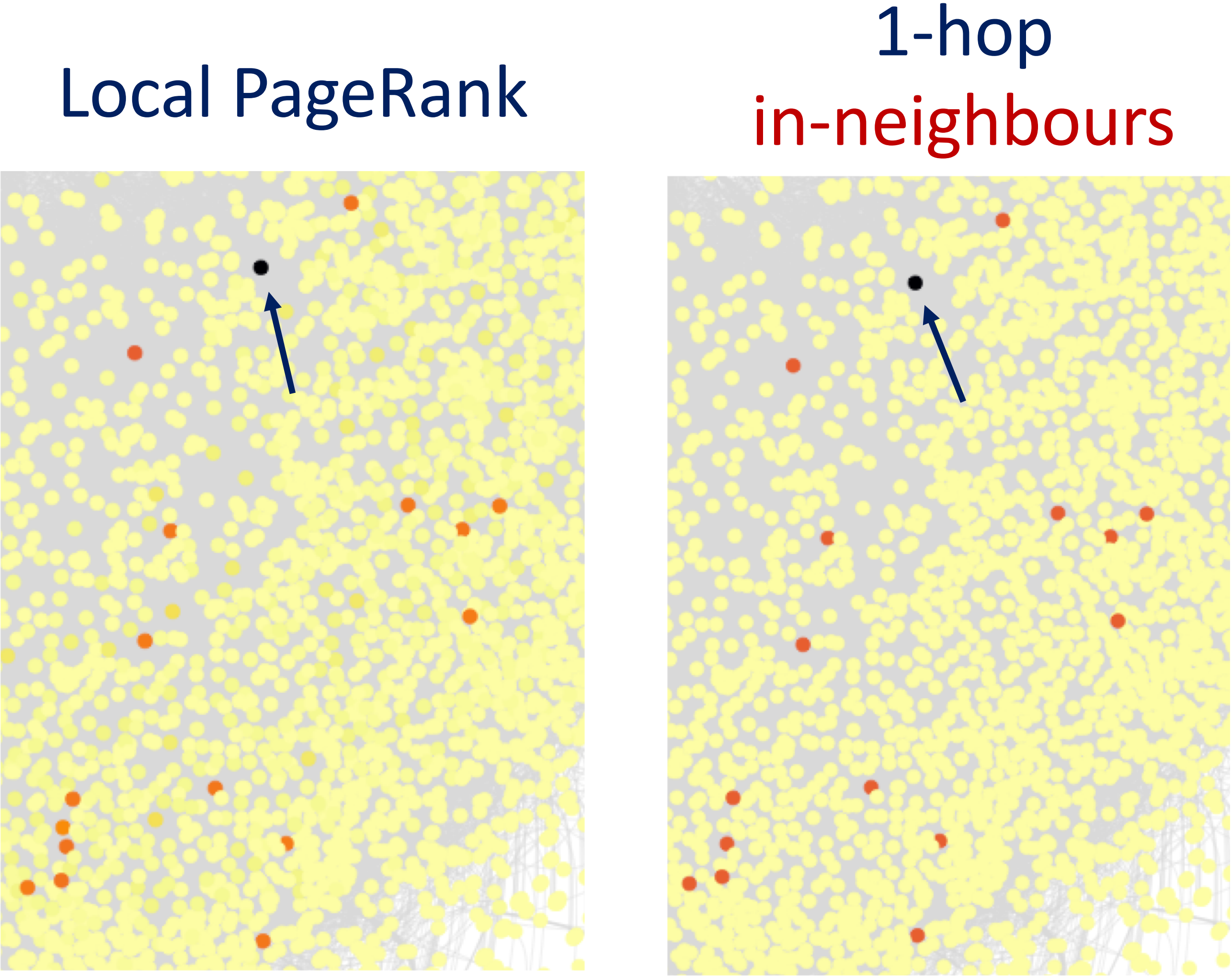
Local PageRank

1-hop
out-neighbours



neighbours authority score =
local node → neighbours

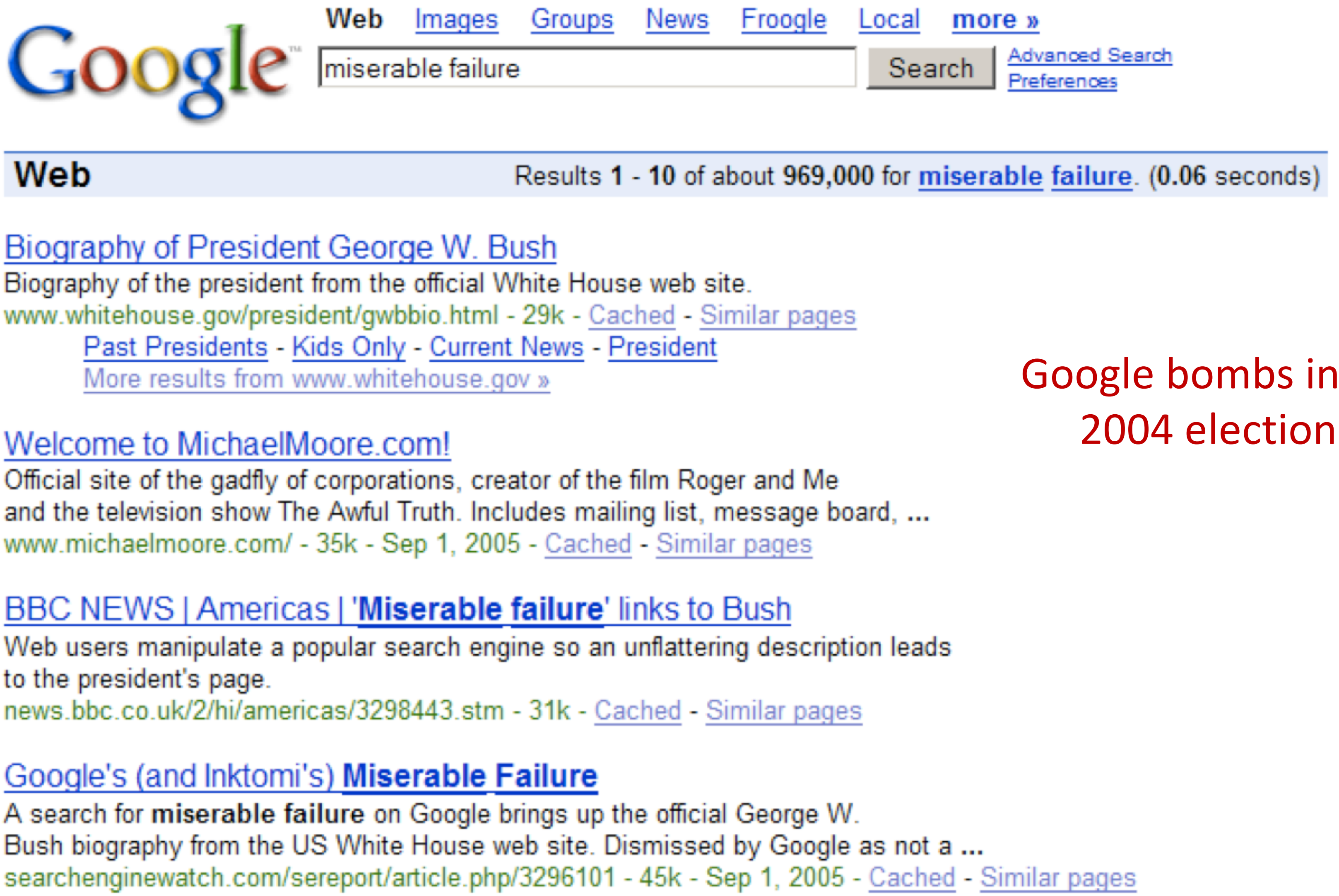
Local PageRank (hubs, A^T)



neighbours hub score =
neighbours → local node

Combating spam farms

Spam farms



Web Results **1** - **10** of about **969,000** for **miserable failure**. (**0.06** seconds)

Biography of President George W. Bush
Biography of the president from the official White House web site.
www.whitehouse.gov/president/gwbbio.html - 29k - Cached - Similar pages
Past Presidents - Kids Only - Current News - President
More results from www.whitehouse.gov »

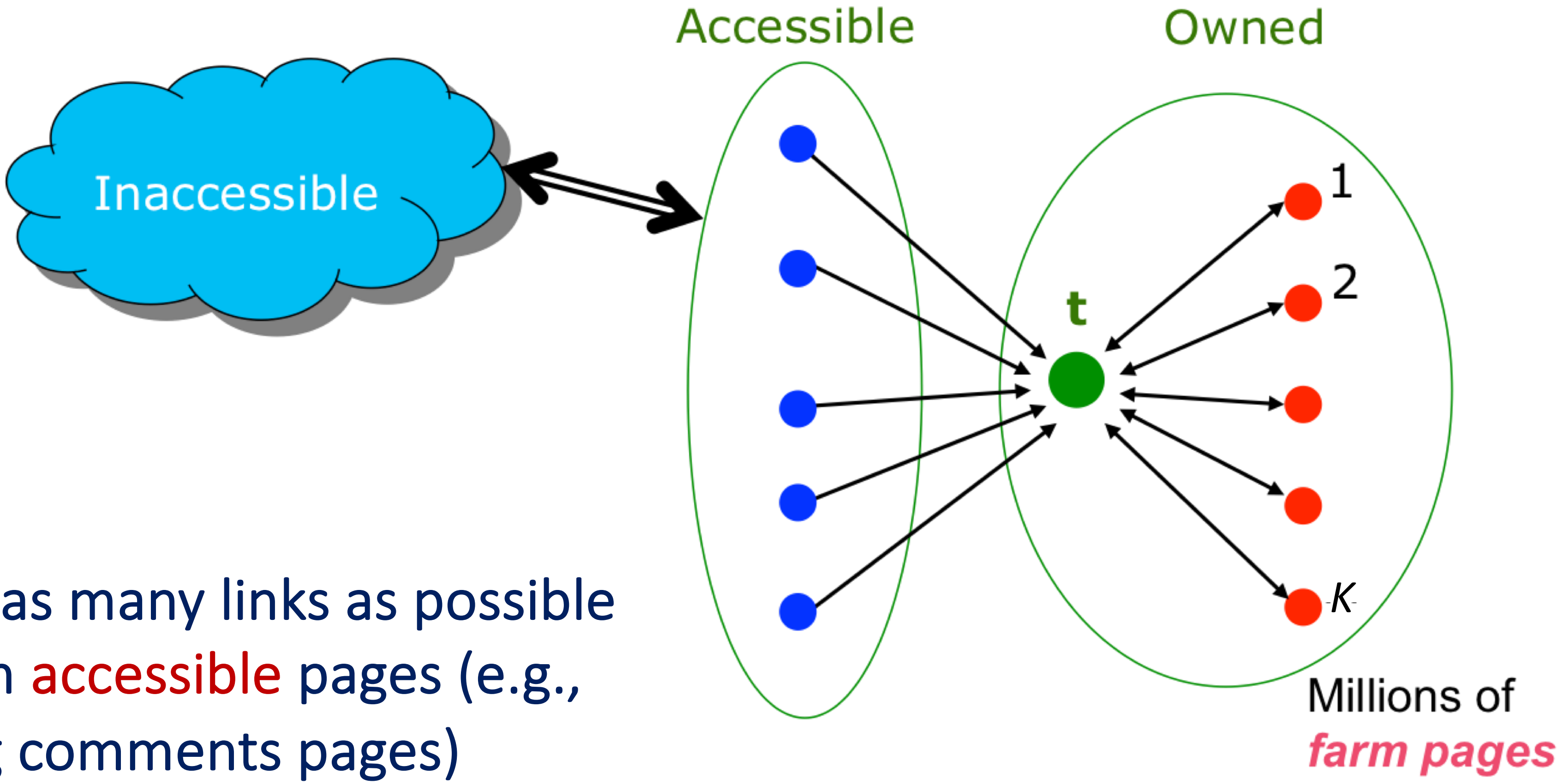
Welcome to MichaelMoore.com!
Official site of the gadfly of corporations, creator of the film Roger and Me and the television show The Awful Truth. Includes mailing list, message board, ...
www.michaelmoore.com/ - 35k - Sep 1, 2005 - Cached - Similar pages

BBC NEWS | Americas | '**Miserable failure**' links to Bush
Web users manipulate a popular search engine so an unflattering description leads to the president's page.
news.bbc.co.uk/2/hi/americas/3298443.stm - 31k - Cached - Similar pages

Google's (and Inktomi's) **Miserable Failure**
A search for **miserable failure** on Google brings up the official George W. Bush biography from the US White House web site. Dismissed by Google as not a ...
searchenginewatch.com/sereport/article.php/3296101 - 45k - Sep 1, 2005 - Cached - Similar pages

Google bombs in the 2004 elections

Spamming method

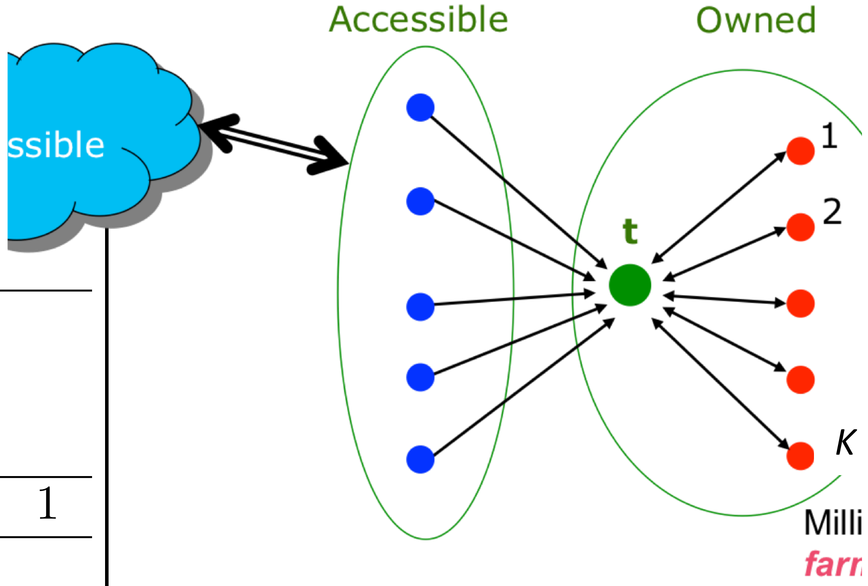


1. Get as many links as possible from accessible pages (e.g., blog comments pages)

2. Construct link farm to get a PageRank multiplier effect

PageRank analysis

$$
\begin{array}{r|c|c|c|c|}
 & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{inacc.} & ? & ? & \begin{matrix}0\\ \vdots \\ 0\end{matrix} & \mathbf{0} \\
\hline
\text{acc.} \quad M = & ? & ? & \begin{matrix}0\\ \vdots \\ 0\end{matrix} & \mathbf{0} \\
\hline
\text{t} & 0 \ \cdots \ 0 & ? \ \cdots \ ? & 0 & 1 \ \cdots \ 1 \\
\hline
\text{owned} & \mathbf{0} & \mathbf{0} & \begin{matrix}\frac{1}{K}\\ \vdots \\ \frac{1}{K}\end{matrix} & \mathbf{0} \\
 & \text{inaccessible} & \text{accessible} & \text{t} & \text{owned}
\end{array}
$$



teleportation value to pages owned by the spammer

ranking due to accessible pages

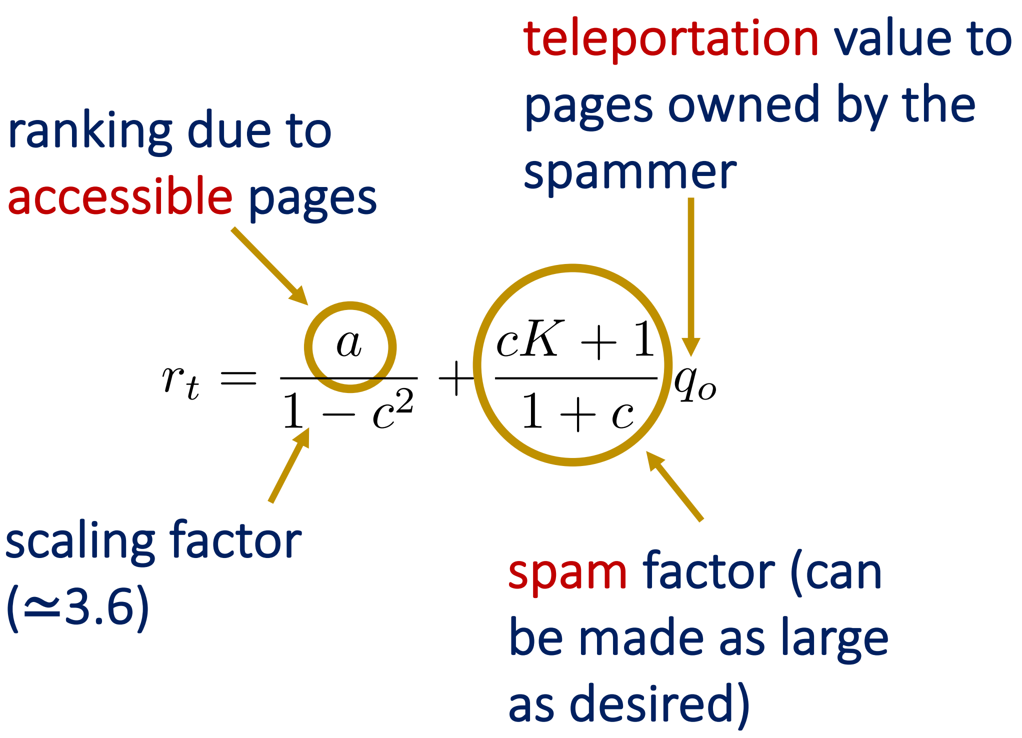
$$\boldsymbol{r} = c\,\boldsymbol{M}\,\boldsymbol{r} + (1 - c)\,\boldsymbol{q}$$

$$r_t = \bigcirc\!\!\!\!\!a + cKr_o + (1-c)q_o$$

$$r_o = c\frac{1}{K}r_t + (1-c)q_o$$

Solution

analysis recap



teleportation value to pages owned by the spammer

ranking due to accessible pages

$$r_t = \frac{a}{1-c^2} + \frac{cK+1}{1+c} q_o$$

scaling factor (≃3.6)

spam factor (can be made as large as desired)

solution

teleport only to trusted pages (i.e., set $q_o = 0$)

can also be used as a method to identify spam farms

PageRank in signed networks

Jung, Jim, Sael, Kang, “Personalized ranking in signed networks using signed random walk with restart,” 2016

https://ieeexplore.ieee.org/iel7/7837023/7837813/07837935.pdf

use institutional Sign In with your unipd credentials

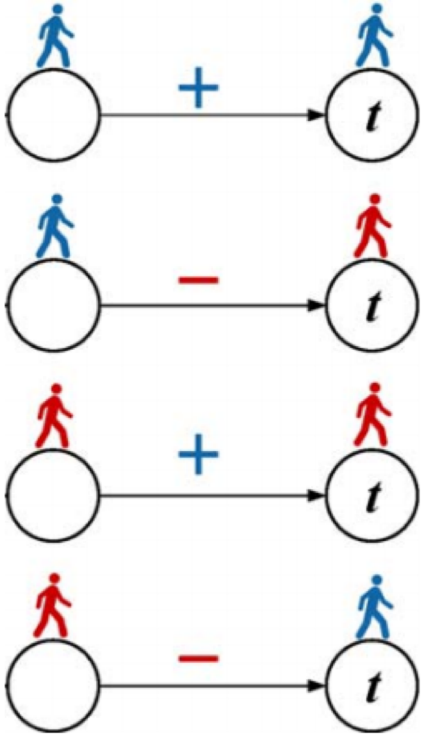
PageRank in signed networks

Idea:

- ☐ Identify + (favourable) and – (adversarial) paths, i.e., ranking values r_+ and r_- for positive and negative surfers
- ☐ Extract positive A_+ and negative A_- contributions to $A = A_+ - A_-$
- ☐ Normalize the absolute value, to get M_+ and M_- (with normalized $M_+ + M_-$)
- ☐ Run a signed random walk

$$r_+ = c\, M_+\, r_+ + c\, M_-\, r_- + (1-c)\, q$$

$$r_- = c\, M_-\, r_+ + c\, M_+\, r_-$$



PageRank in signed networks

signed random
walk equation

$$M = \begin{bmatrix} M_+ & M_- \\ M_- & M_+ \end{bmatrix}$$

damping factor

(column) normalized
adjacency matrix

$$r = c\,M\,r + (1-c)\,q_0$$

teleportation vector

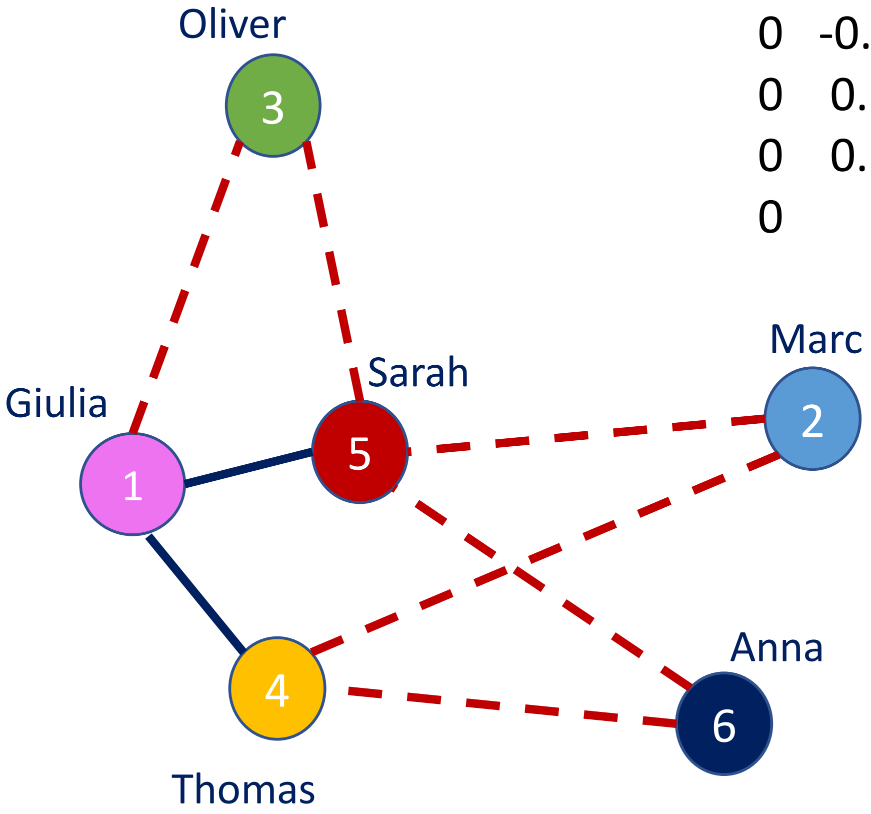
PageRank vector
(centrality)

$$r = \begin{bmatrix} r_+ \\ r_- \end{bmatrix}$$

$$q_0 = \begin{bmatrix} q \\ 0 \end{bmatrix}$$

signed centrality
outcome $\boxed{r_+ - r_-}$

Example



t=1	*2*	*3*	*4*	*5*
1.0000	0.1500	0.4109	0.2403	0.3404
0	0	-0.1405	-0.0467	-0.1262
0	-0.2833	-0.1027	-0.1510	-0.1275
0	0.2833	0.0425	0.2358	0.1078
0	0.2833	0.1629	0.2795	0.1719
0	0	-0.1405	-0.0467	-0.1262

10	*20*	*50*	*75*	*100*	
0.2909	0.2985	0.2989	0.2989	0.2989	Giulia
-0.0848	-0.0926	-0.0931	-0.0931	-0.0931	Marc
-0.1309	-0.1313	-0.1314	-0.1314	-0.1314	Oliver
0.1763	0.1645	0.1638	0.1638	0.1638	Thomas
0.2324	0.2204	0.2197	0.2197	0.2197	Sarah
-0.0848	-0.0926	-0.0931	-0.0931	-0.0931	Anna

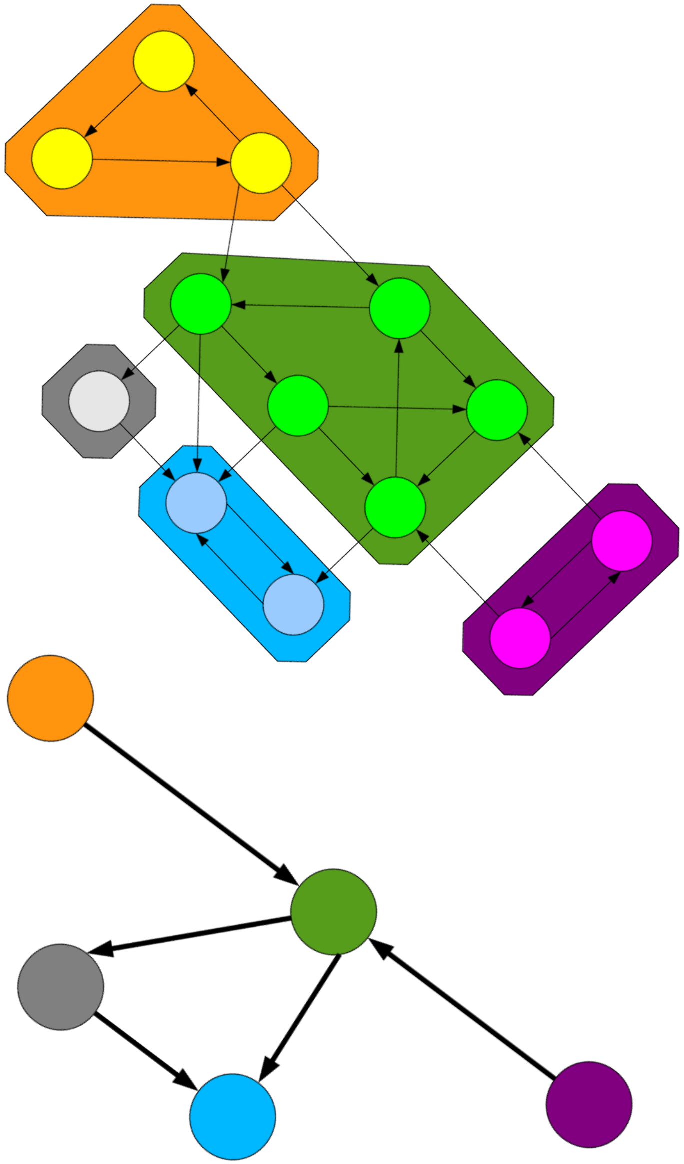
PageRank eigenstructure

Haveliwala and Kamvar, “The second eigenvalue of the Google matrix,” 2003

http://ilpubs.stanford.edu:8090/582/1/2003-20.pdf

Condensation graph

- ☐ Strong connectivity induces a partition in disjoint strongly connected sets $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_K$
- ☐ By reinterpreting the sets as nodes we obtain a condensation graph $\mathcal{G}^*$ where $i \rightarrow j$ is an edge if a connection exists between sets $\mathcal{V}_i \rightarrow \mathcal{V}_j$



Properties of $\mathcal{G}^*$

- ☐ $\mathcal{G}^*$ does not contain cycles

  otherwise the sets in the cycle would be strongly connected

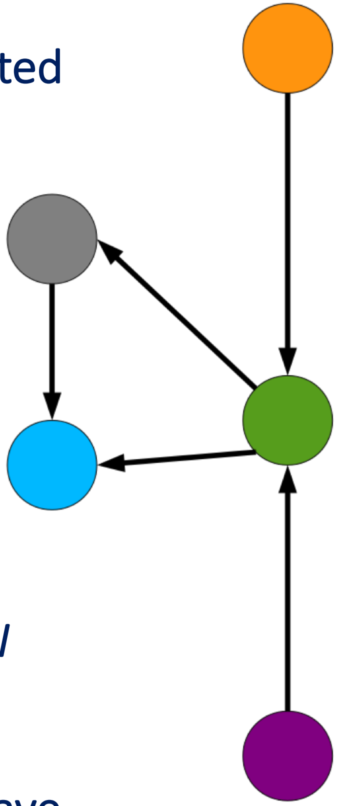
- ☐ $\mathcal{G}^*$ has at least one root and one leaf

  and every node in the graph can be reached from one of the roots

- ☐ $\mathcal{G}^*$ allows a particular reordering

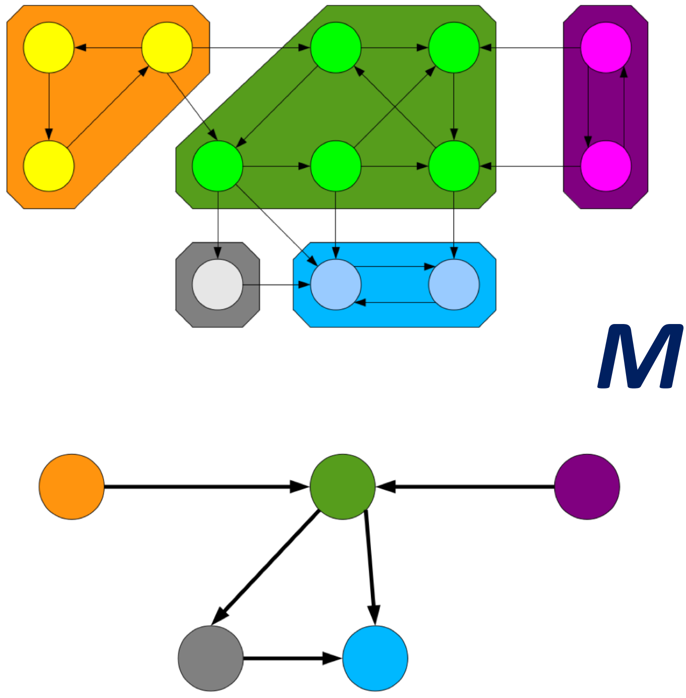
  where node n_i does not reach any of the nodes n_j with $j<I$

  procedure: identify a root n_1 and remove it from the network, then identify a new root; cycle until all nodes have been selected



Condensation graph

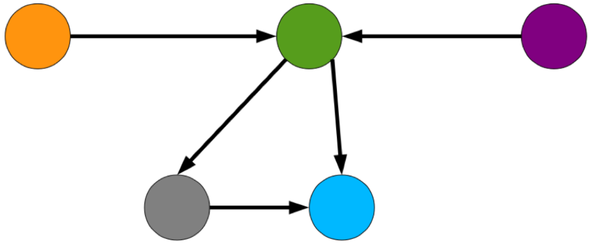
- ❑ The condensation graph ordering induces a block-lower-triangular matrix structure on the adjacency matrix



$$M=\left[\begin{array}{ccc|cc|ccccc|c|cc}
 & & 3 & & 2 & & & & & 5 & 1 & & 2 \\
\hline
 & \frac{1}{3} & & & & & & & & & & & \\
 & & 1 & & & & & & & & & & \\
1 & & & & & & & & & & & & \\
\hline
 & & & & \frac{1}{2} & & & & & & & & \\
 & & & \frac{1}{2} & & & & & & & & & \\
\hline
 & \frac{1}{3} & & & & & & & & \frac{1}{2} & & & \\
 & & & \frac{1}{2} & & \frac{1}{2} & & & \frac{1}{3} & & & & \\
 & \frac{1}{3} & & & & \frac{1}{2} & & & & & & & \\
 & & & & & & & \frac{1}{3} & & & & & \\
 & & & & \frac{1}{2} & & 1 & & \frac{1}{3} & & & & \\
\hline
 & & & & & & & \frac{1}{3} & & & 0 & & \\
\hline
 & & & & & & & \frac{1}{3} & \frac{1}{3} & & 1 & & 1 \\
 & & & & & & & & & \frac{1}{2} & & 1 &
\end{array}\right]$$

blocks in the diagonal are irreducible = no block-diagonal form !

Perron-Frobenius theorem

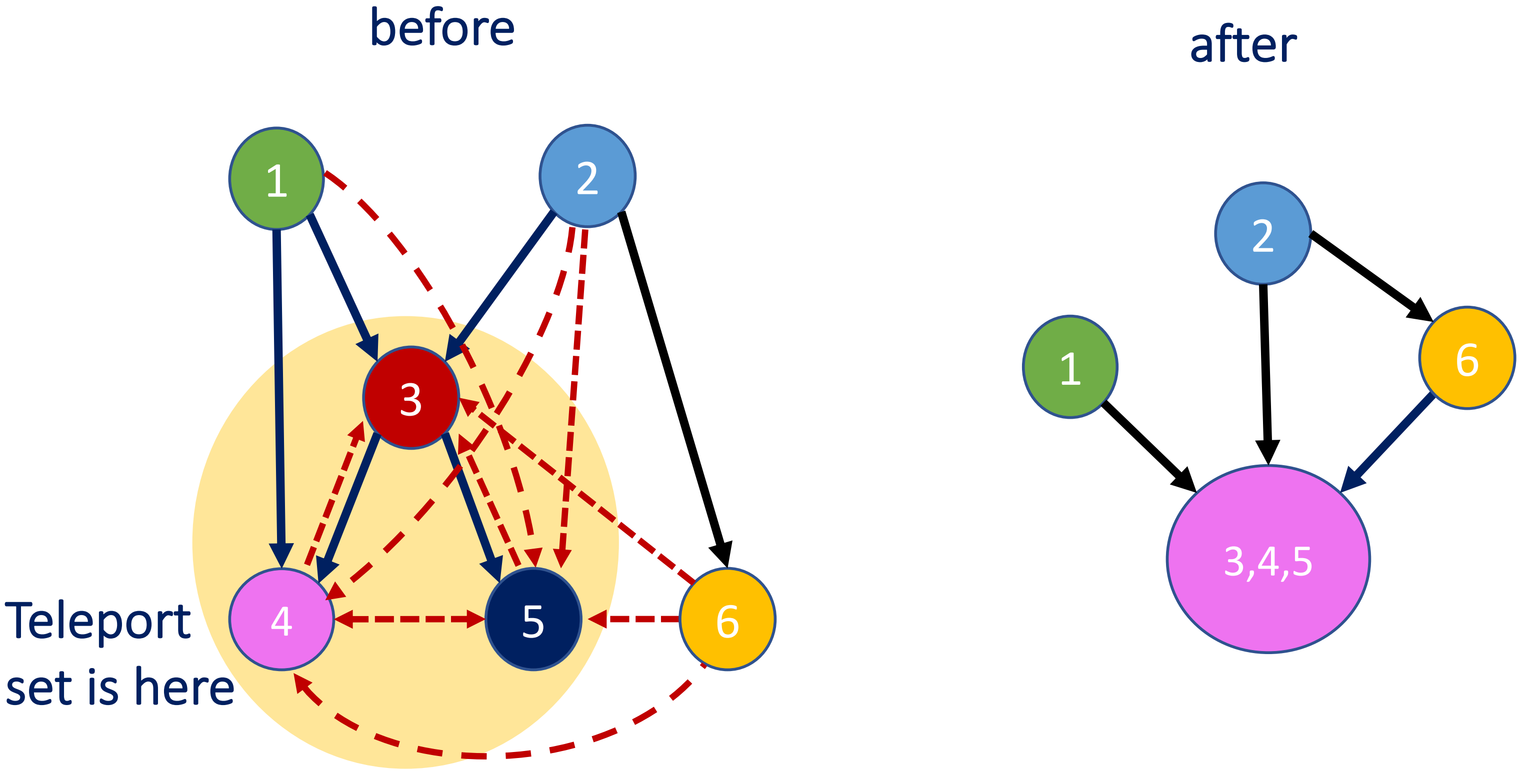


the eigenvalues of the diagonal blocks, except for the leaves, lie inside the unit circle, i.e., $|\lambda|<1$

$$M=\begin{array}{ccc|cc|ccccc|c|cc}
\multicolumn{3}{r}{3} & \multicolumn{2}{r}{2} & \multicolumn{5}{r}{5} & \multicolumn{1}{r}{1} & \multicolumn{2}{r}{2} \\
 & \frac{1}{3} & & & & & & & & & & & \\
 & & 1 & & & & & & & & & & \\
1 & & & & & & & & & & & & \\
\hline
 & & & & \frac{1}{2} & & & & & & & & \\
 & & & \frac{1}{2} & & & & & & & & & \\
\hline
 & \frac{1}{3} & & & & & & & & \frac{1}{2} & & & \\
 & & & \frac{1}{2} & & \frac{1}{2} & & & \frac{1}{3} & & & & \\
 & \frac{1}{3} & & & & \frac{1}{2} & & & & & & & \\
 & & & & & & & \frac{1}{3} & & & & & \\
 & & & & \frac{1}{2} & & 1 & & \frac{1}{3} & & & & \\
\hline
 & & & & & & & \frac{1}{3} & & & 0 & & \\
\hline
 & & & & & & & \frac{1}{3} & \frac{1}{3} & & 1 & & 1 \\
 & & & & & & & & & \frac{1}{2} & & 1 &
\end{array}$$

each leaf-block has at least one eigenvalue in the unit circle, $\lambda=1$

Teleportation ensures one-leaf only



- ❑ Hence M_1 carries only one eigenvector associated with the eigenvalue $\lambda=1$

MIME.

Lemma

- PageRank matrix $\boldsymbol{M_1} = c\,\boldsymbol{M} + (1-c)\,\boldsymbol{q}\,\boldsymbol{1}^{\mathrm{T}}$
- Normalization property $\boldsymbol{1}^{\mathrm{T}}\,\boldsymbol{M_1} = \boldsymbol{1}^{\mathrm{T}}$
- Jordan form $\boldsymbol{M_1} = \boldsymbol{V}\,\boldsymbol{J}\,\boldsymbol{V}^{-1}$

carries the right (generalized) eigenvectors $\boldsymbol{e}_i$ of M_1

carries the eigenvalues of M_1

$$J = \begin{bmatrix} \boxed{\begin{matrix} \lambda_1 & 1 & \\ & \lambda_1 & 1 \\ & & \lambda_1 \end{matrix}} & & & & \\ & \boxed{\begin{matrix} \lambda_2 & 1 \\ & \lambda_2 \end{matrix}} & & & \\ & & \boxed{\lambda_3} & & \\ & & & \ddots & \\ & & & & \boxed{\begin{matrix} \lambda_n & 1 \\ & \lambda_n \end{matrix}} \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{1}^{\mathrm{T}}\,\boldsymbol{M_1}\,\boldsymbol{V} &= \boldsymbol{1}^{\mathrm{T}}\,\boldsymbol{V} \\ &= \boldsymbol{1}^{\mathrm{T}}\,\boldsymbol{V}\,\boldsymbol{J} \end{aligned} \quad \Rightarrow \quad \underbrace{\boldsymbol{1}^{\mathrm{T}}\,\boldsymbol{V}}_{\boldsymbol{\rho}}\;\underbrace{(\boldsymbol{J} - \boldsymbol{I})}_{\text{only one eigenvalue is 0}} = \boldsymbol{0}$$

- Hence $\boldsymbol{1}^{\mathrm{T}}\,\boldsymbol{e}_i = 0$ for $i>1$, i.e., except for the eigenvector associated with eigenvalue 1

Main result

$$\underbrace{\boldsymbol{M}_1\, \boldsymbol{e}_i = c\, \boldsymbol{M}\, \boldsymbol{e}_i}_{\text{same eigenvalues of M, but multiplied by c !!!}} + (1\text{-}c)\, \boldsymbol{q}\, \xcancel{\boldsymbol{1}^T}\, \boldsymbol{e}_i \qquad \text{for } i>1$$

☐ $\boldsymbol{M}_1$ has one eigenvalue equal to 1

☐ The remaining eigenvalues satisfy $|\lambda| \leq c$

Approximate PageRank

Andersen, Chung, Lang, “Local graph partitioning using PageRank vectors,” 2006

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4031383

use institutional Sign In with your unipd credentials

Approximate algorithm

- Start from $\boldsymbol{u} = \boldsymbol{0}$ and $\boldsymbol{v} = \boldsymbol{q}$

precision

degree of node i

sum of the degrees

- To all the nodes $\underline{i}$ satisfying $\underline{v_i > \varepsilon\ d_i/D}$ apply the push operation

u constantly increases

$$\boldsymbol{u}^+ = \boldsymbol{u} + (1\text{-}c)\ \boldsymbol{\delta}$$

only one active element in position i with value v_i

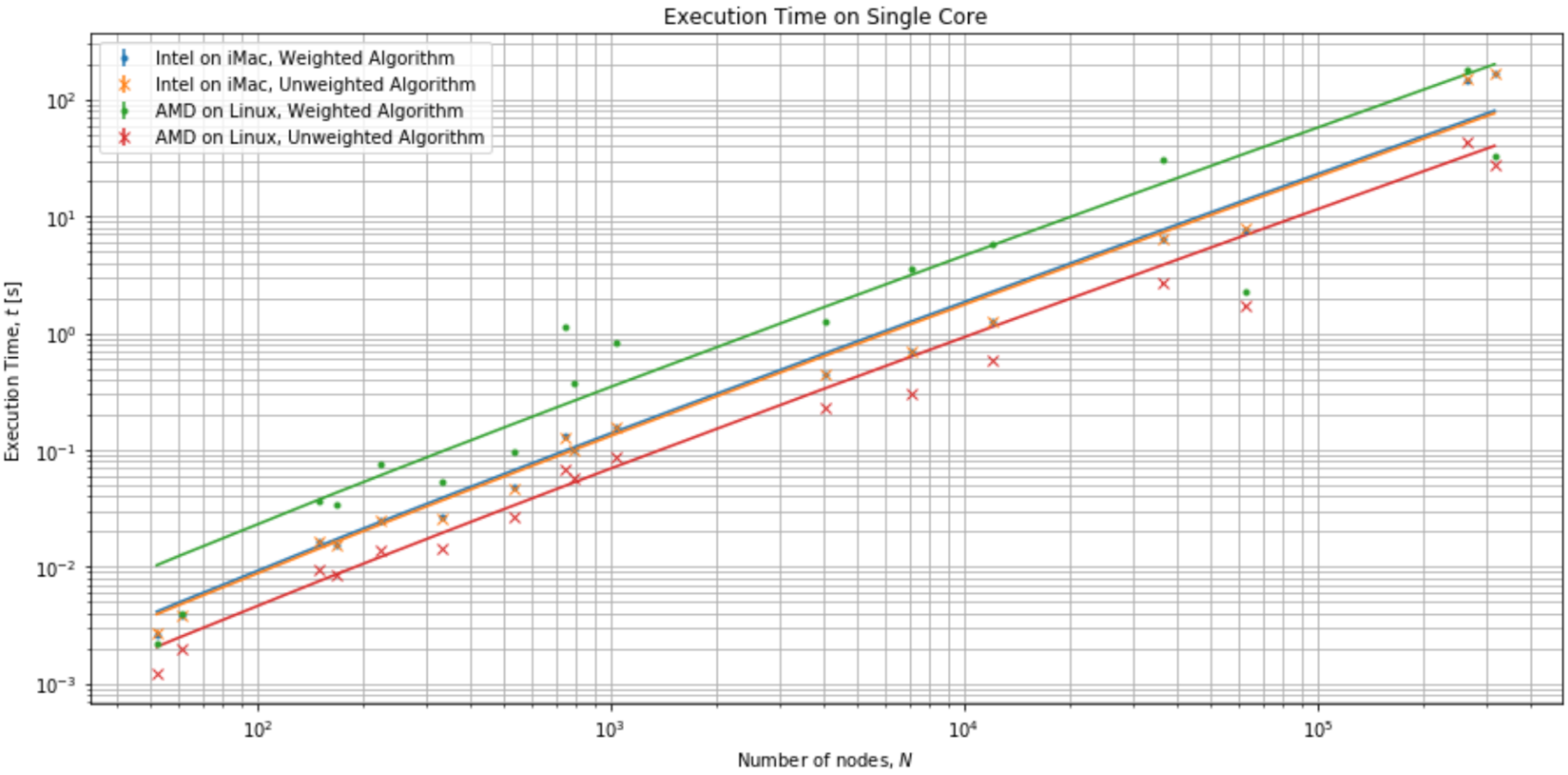
v always positive

$$\boldsymbol{v}^+ = \boldsymbol{v} - \boldsymbol{\delta} + c\ \boldsymbol{M}\ \boldsymbol{\delta}$$

- Returns $\boldsymbol{r} \simeq \boldsymbol{u}$ with precision $|\boldsymbol{r} - \boldsymbol{u}|_1 < \varepsilon$
- It is simple

Scalability properties

(Francesco Barbato & Tommaso Boccato, 2020)



PageRank linearity

column stochastic matrix $\mathbf{1}^{\mathrm{T}} \boldsymbol{M} = \mathbf{1}^{\mathrm{T}}$

- PageRank equation $\boldsymbol{r}_q = c\, \boldsymbol{M}\, \boldsymbol{r}_q + (1\text{-}c)\, \boldsymbol{q}$

stochastic ranking vector
$\mathbf{1}^{\mathrm{T}} \boldsymbol{r}_q = 1,\ \boldsymbol{r}_q \geq 0$

stochastic Teleport vector
$\mathbf{1}^{\mathrm{T}} \boldsymbol{q} = 1$

- Alternative equation $\boldsymbol{r}_q = (\boldsymbol{I} - c\, \boldsymbol{M})^{-1}\, (1\text{-}c)\, \boldsymbol{q}$

$\Downarrow$ linear in q

$$\boldsymbol{r}_{au+bv} = a\, \boldsymbol{r}_u + b\, \boldsymbol{r}_v$$

Alternative equation

one-step random walk

- PageRank equation $\boldsymbol{r}_q = c\, \boldsymbol{r}_{Mq} + (1-c)\, \boldsymbol{q}$

  - $\boldsymbol{r}_q = (\boldsymbol{I} - c\, \boldsymbol{M})^{-1} (1-c)\, \boldsymbol{q}$
  - $\boldsymbol{r}_q = (1-c) \Sigma (c\, \boldsymbol{M})^k \boldsymbol{q}$
  - $\boldsymbol{M}\, \boldsymbol{r}_q = (1-c) \Sigma (c\, \boldsymbol{M})^k \boldsymbol{M}\, \boldsymbol{q}$
  - $\boldsymbol{M}\, \boldsymbol{r}_q = \boldsymbol{r}_{Mq}$

Main property of push: $r_q = u + r_v$

- At starting point $\boldsymbol{u} = \boldsymbol{0}$ and $\boldsymbol{v} = \boldsymbol{q}$ imply $\boldsymbol{r}_q = \boldsymbol{0} + \boldsymbol{r}_q$

- The following steps are proved by induction

$$\boldsymbol{u}^+ = \boldsymbol{u} + (1\text{-}c)\,\delta$$
$$\boldsymbol{v}^+ = \boldsymbol{v} - \delta + c\,\boldsymbol{M}\,\delta$$

$$\Downarrow$$

$$\boldsymbol{u}^+ + \boldsymbol{r}_{v+} = \boldsymbol{u} + (1\text{-}c)\,\delta + \overbrace{\boldsymbol{r}_v - \boldsymbol{r}_\delta + \underbrace{c\,\boldsymbol{r}_{M\delta}}_{\boldsymbol{r}_\delta - (1\text{-}c)\,\delta}}^{\text{by linearity}}$$

$$\Downarrow$$

$$\boldsymbol{u}^+ + \boldsymbol{r}_{v+} = \boldsymbol{u} + \boldsymbol{r}_v = \boldsymbol{r}_q$$

Precision guarantee $|r_q - u|_1 < \varepsilon$

- The push property implies $\boldsymbol{r}_q = \boldsymbol{u} + \boldsymbol{r}_v$
- Hence $|\boldsymbol{r}_q - \boldsymbol{u}|_1 = |\boldsymbol{r}_v|_1 = \boldsymbol{1}^T \boldsymbol{r}_v$

- The PageRank equation is $\boldsymbol{r}_v = c\, \boldsymbol{M}\, \boldsymbol{r}_v + (1-c)\, \boldsymbol{v}$
- Hence $\boldsymbol{1}^T \boldsymbol{r}_v = c\, \underbrace{\boldsymbol{1}^T \boldsymbol{M}}_{\boldsymbol{1}^T}\, \boldsymbol{r}_v + (1-c)\, \boldsymbol{1}^T \boldsymbol{v}$ so that $\boldsymbol{1}^T \boldsymbol{r}_v = \boldsymbol{1}^T \boldsymbol{v}$

- As a result $|\boldsymbol{r}_q - \boldsymbol{u}|_1 = \boldsymbol{1}^T \boldsymbol{v} < \Sigma\, \varepsilon\, d_i/D = \varepsilon$

Lazy PageRank

- Lazy PageRank $\boldsymbol{r} = \underbrace{a\,\boldsymbol{M}_2}_{\boldsymbol{M}_2 = b\,\boldsymbol{I} + (1-b)\,\boldsymbol{M}}\,\boldsymbol{r} + (1-a)\,\boldsymbol{q}$

- Lazy because a fraction b of the times the surfer stays where she/he is

- Equivalent to $\underbrace{\boldsymbol{r} = c}_{c = a(1-b)/(1-ab) \underbrace{< a}_{\text{slower algorithm}}}\,\boldsymbol{M}\,\boldsymbol{r} + (1-c)\,\boldsymbol{q}$

Lessons learned

- ☐ Importance of Teleport vector
- ☐ PageRank can measure similarity
- ☐ PageRank can be extended to signed networks (with a trick)
- ☐ Reliable and scalable implementations exist