

# The PageRank algorithm and application on searching of academic papers

Ping Yeh  
Google, Inc.

2009/12/9  
Department of Physics, NTU

# Disclaimer (legal)

The content of this talk is the speaker's personal opinion and is not the opinion or policy of his employer.

# Disclaimer (content)

- You will not hear physics.
- You will not see differential equations.
- You will:
  - get a review of PageRank, the algorithm used in Google's web search. It has been applied to evaluate journal status and influence of nodes in a graph by researchers,
  - see some linear algebra and Markov chains associated with it, and
  - see some results of applying it to journal status.

# Outline

- Introduction
- Google and Google search
- PageRank algorithm for ranking web pages
- Using MapReduce to calculate PageRank for billions of pages
- Impact factor of journals and PageRank
- Conclusion

# Google

- The name: homophone to the word “Googol” which means  $10^{100}$ .
- The company:
  - founded by Larry Page and Sergey Brin in 1998,
  - ~20,000 employees as of 2009,
  - spread in 68 offices around the world (23 in N. America, 3 in Latin America, 14 in Asia Pacific, 23 in Europe, 5 in Middle East and Africa).
- The mission: “to organize the world's information and make it universally accessible and useful.”

# Google Services

YouTube iGoogle talk web search Sky  
book search Chrome calendar scholar  
translate Android product search blogger.com maps  
news video picasaweb Gmail  
groups desktop reader Earth

# Google Search

- <http://www.google.com/> or <http://www.google.com.tw/>



# The abundance problem

Quote Langville and Meyer's nice book "Google's PageRank and beyond: the science of search engine rankings":

The men in Jorge Luis Borges' 1941 short story, "The Library of Babel", which describes an imaginary, infinite library.

- When it was proclaimed that the Library contained all books, the first impression was one of extravagant happiness. All men felt themselves to be the masters of an intact and secret treasure. There was no personal or world problem whose eloquent solution did not exist in some hexagon.

. . . As was natural, this inordinate hope was followed by an excessive depression. The certitude that some shelf in some hexagon held precious books and that these precious books were inaccessible seemed almost intolerable.

# The power of web search engines nowadays

Search



vs.

Typical desktop document application

Data size

$O(10^{14} \text{ B})$

$O(10^6 \text{ B})$

Web Proliferation

Speed

~ 250 ms

~ 1s

Cloud Computing

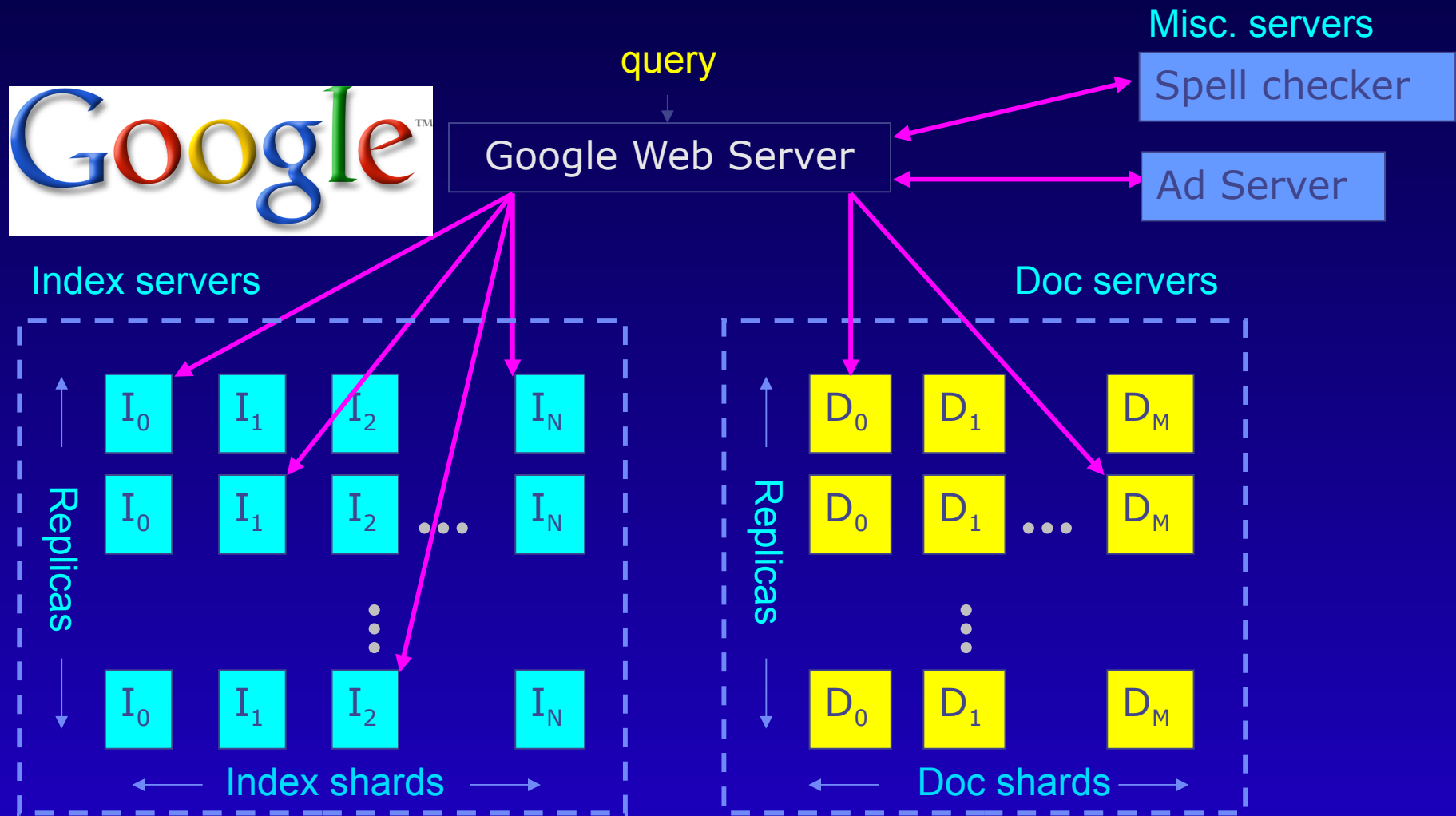
Relevance

high

low

Ranking

# Serving a query



IEEE Micro, 2003

0.25 seconds, **1000+** machines



NATIONAL ACADEMY  
OF ENGINEERING

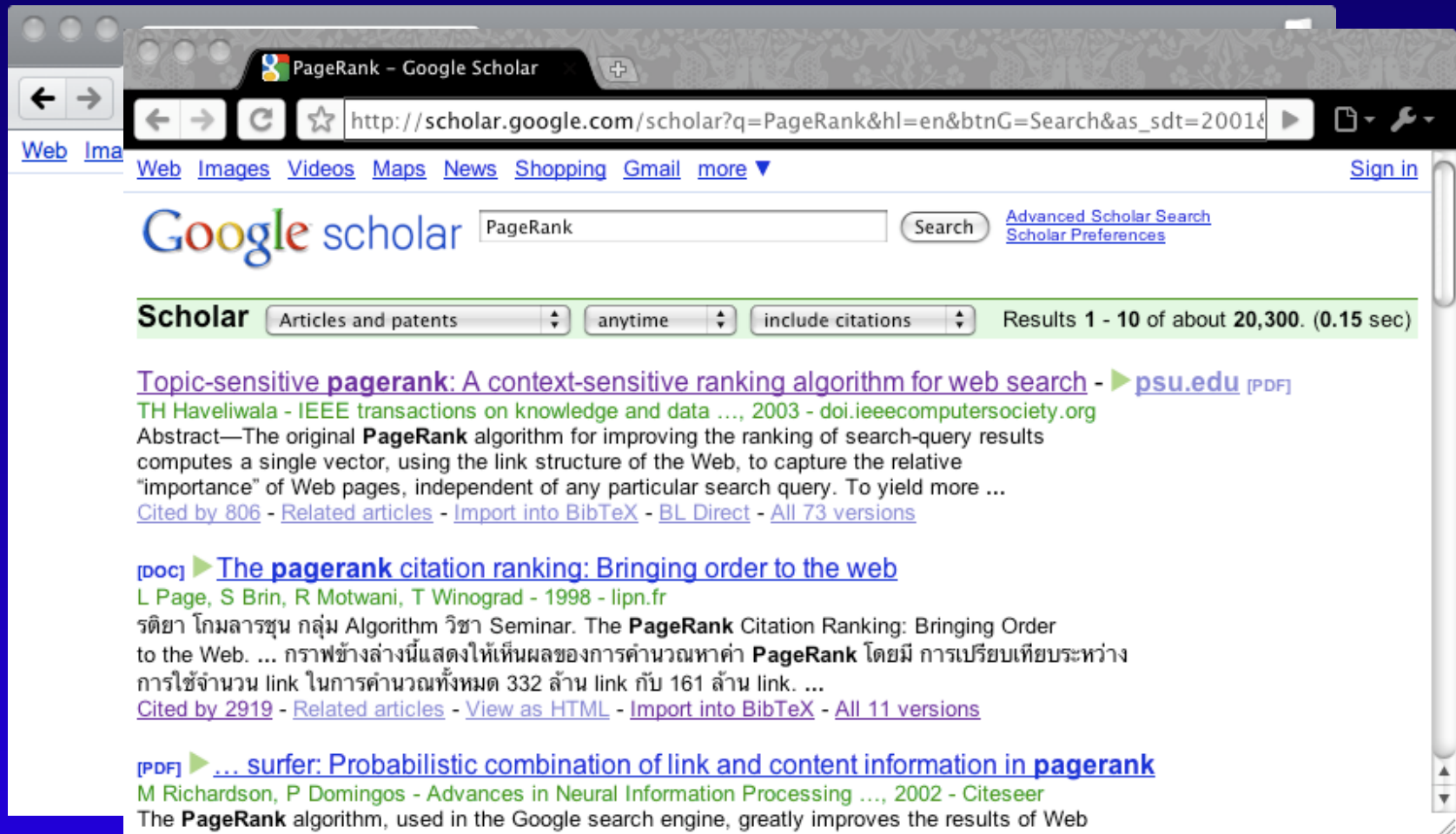
# Major contributors

- Ranking the results: Larry Page,  
“For the creation of the Google search engine.”  
(2004)
- Indexing the web: Sergey Brin,  
“For leadership in development of rapid indexing  
and retrieval of relevant information from the  
World Wide Web.” (2009)
- Dealing with scales: Jeff Dean and  
Sanjay Ghemawat,  
“for contributions to the science and  
engineering of large-scale distribute  
computer systems.” (2009)



# Google Scholar

- Search for scholarly literature from one place
  - Motto: “Stand on the shoulders of giants”
- <http://scholar.google.com/>



# What can be found with Google Scholar?

- Types: articles, theses, books and abstracts.
  - Added court opinions from U.S. federal and state district, appellate and supreme courts on Nov 17.
- Sources: academic publishers, professional societies, online repositories, universities and other web sites.

Live demo

# Ranking the web

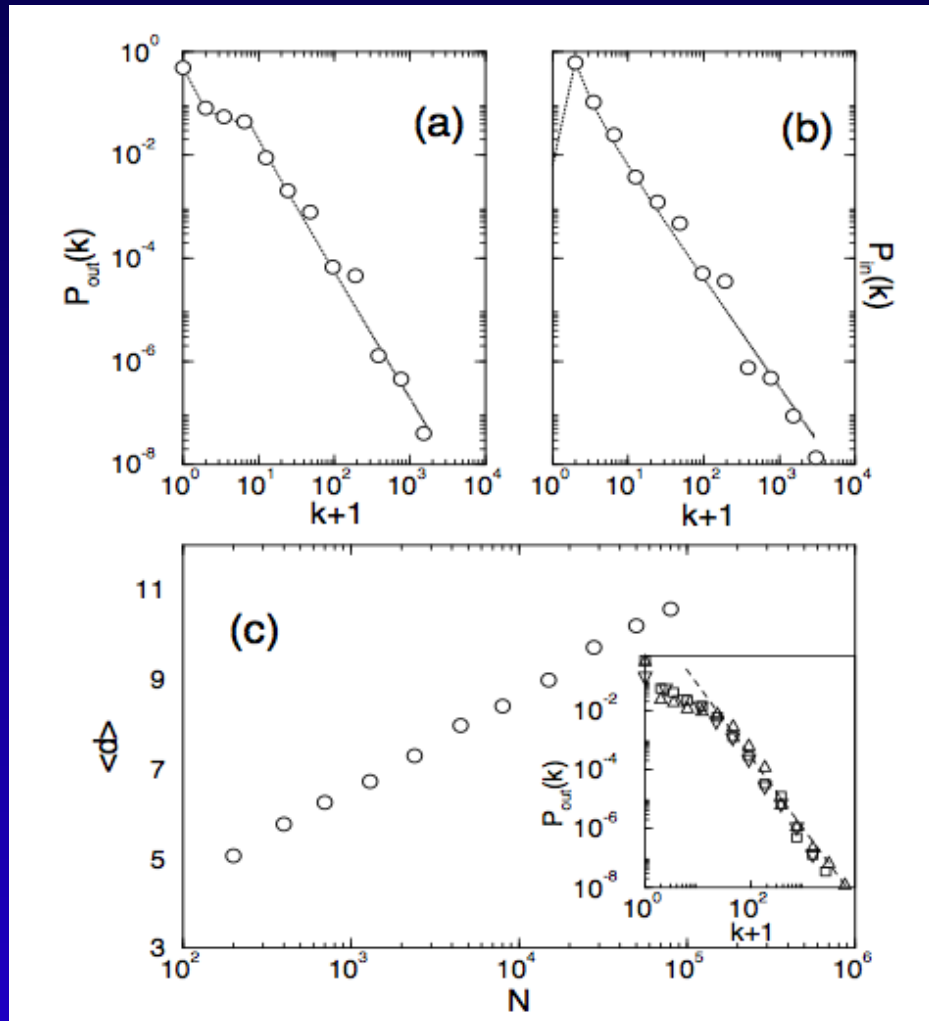
- Given a query, which are the most relevant / authoritative pages?
- Which pages should be in the search result?
  - Find pages containing the query?
    - Not enough. By searching “NTU physics” you expect to see <http://www.phys.ntu.edu.tw/> in results, but that page contains only one “NTU” and zero “physics” - because it's in Chinese!
  - Use links, anchor texts and other means to bring it in
- How to rank these pages?
  - Content analysis
  - Link analysis



# Link analysis of the web graph

- Crawl web pages and form a graph  $G = (V, E)$ 
  - $V$  = vertices = nodes in the graph. One node is one web page with a unique URL. Let  $N = |V|$ .
  - $E$  = edges = links in the graph. One link is one hyperlink.
- Calculate and attach score/indicators/ranks to nodes based on the links
  - in-links/out-links: links that enter/leave a node
  - in-degree/out-degree = number of in-links/out-links
- $N \times N$  Adjacency matrix  $A$ :  $A_{ij} = 1$  if  $(i,j) \in E$ , 0 otherwise.
  - $A$  is sparse: average out-degree  $\sim 10$ .

# Diameter of the web by Albert et. al. (1999)



- Power-law distribution of out-links (a) and in-links (b)

$$P(k) \propto k^{-\gamma}$$

with  $\gamma_{out} = 2.45$ ,  $\gamma_{in} = 2.1$ .

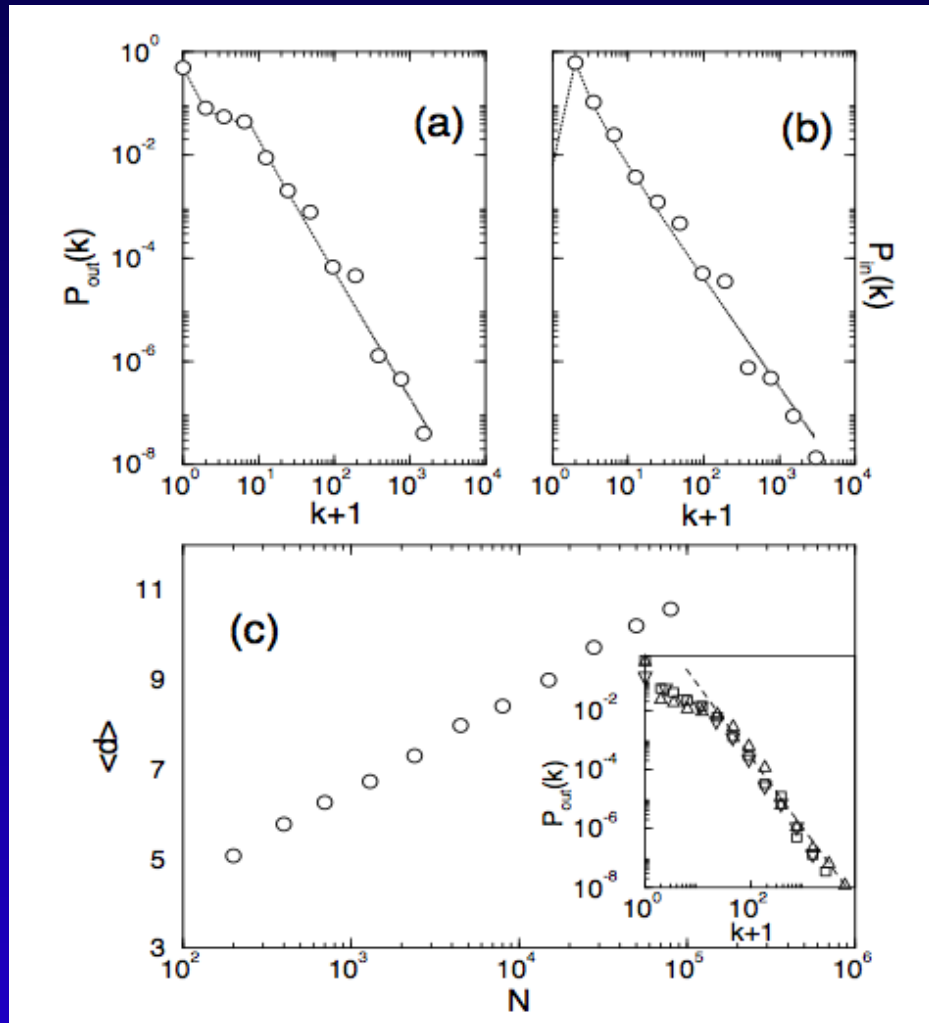
“Scale-free network”

- Mean distance vs. graph size.

$$\langle d \rangle = 0.30 + 2.01 \log(N)$$

“Small world”

# Diameter of the web by Albert et. al. (1999)



- Power-law distribution of out-links (a) and in-links (b)

$$P(k) \propto k^{-\gamma}$$

with  $\gamma_{out} = 2.45$ ,  $\gamma_{in} = 2.1$ .

“Scale-free network”

- Mean distance vs. graph size.

$$\langle d \rangle = 0.35 + 2.06 \log(N)$$

“Small world”

# The challenges of web search

- Web is huge.
  - $O(10^{10})$  pages,  $O(10^{11})$  links
  - the web graph most likely can't be stored in one computer's memory.
- Web is dynamic. Web pages appear and disappear everyday.
- Web is self-organized. Every author is free to create/delete pages and make links to which ever page s/he likes. There is no “formula” or central control

# A test graph

auth: much more in-links than out-links, like an “authoritative” paper.

sink: is a paper that cites nobody

source

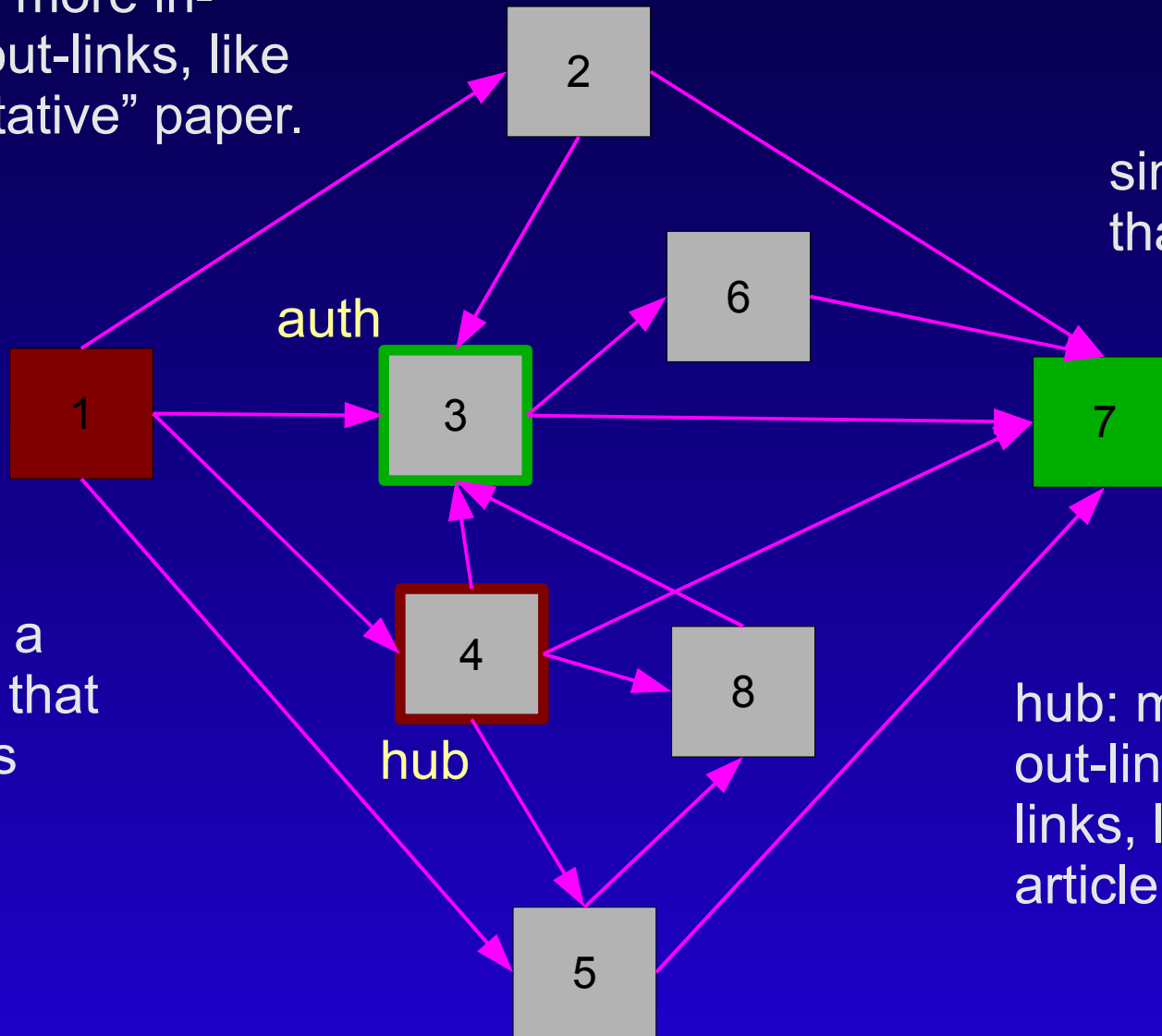
source: is like a master thesis that nobody knows about.

auth

hub

sink

hub: much more out-links than in-links, like a review article.



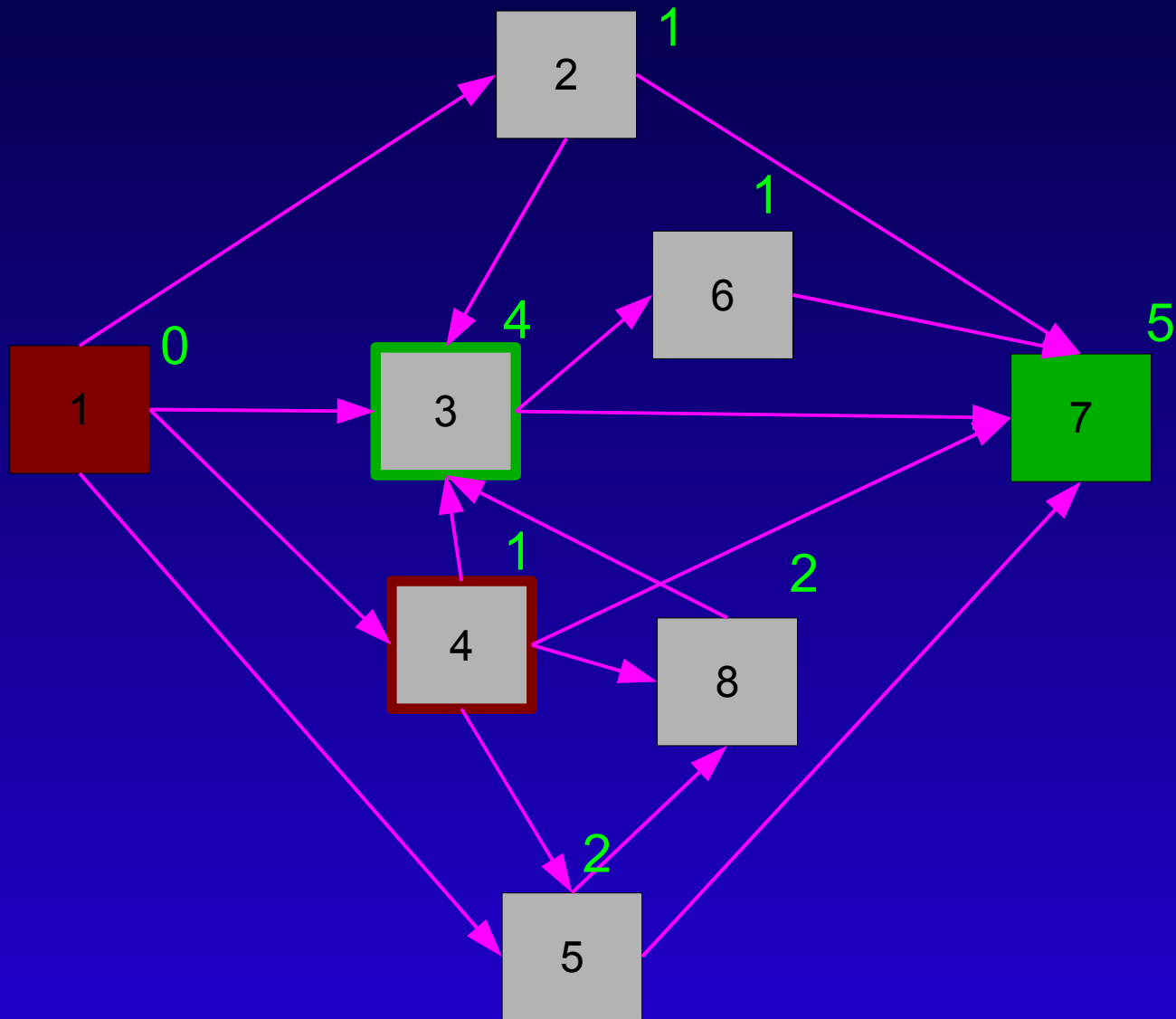
# Bibliometrics

- The study of written documents and their citation structure.
- A long and profound interest in use of citations to produce quantitative estimates of the importance and “*impact*” of individual scientific papers and journals.
  - Scientific value of a paper is only as much as recognized by the social network in the scientific community?
- The most well-known metric: Garfield’s **impact factor**.

# Garfield's Impact Factor (1972)

- Used to provide a numerical assessment of journals in Journal Citation Reports of the Institute for Scientific Information.
  - used in criteria for Ph.D. completion or as a reference point in proposal review in some places.
- IF of a journal  $X$  = average number of citations per paper received this year for articles published in  $X$  in last 2 years.

# Impact Factor



# Katz's Path-counting algorithm (1953)

- For evaluating the “status” of a person or a group by who “chooses” them.

- Number of paths from person  $i$  to person  $j$  of length  $k = C_{ij}^{(k)}$

Note that

$$C_{ij}^{(k)} = \sum_m C_{im}^{(k-1)} C_{mj}^{(1)} \Rightarrow C^{(k)} = C^{(k-1)} C^{(1)} = C^{(1)k} \equiv A^k$$

- Contribution from person  $i$  to person  $j$ 's status = a weighted path sum. The constant  $b$  is context and group dependent.

$$Q \equiv \sum_{k=1}^{\infty} b^k A^k = (I - b A)^{-1} - I$$

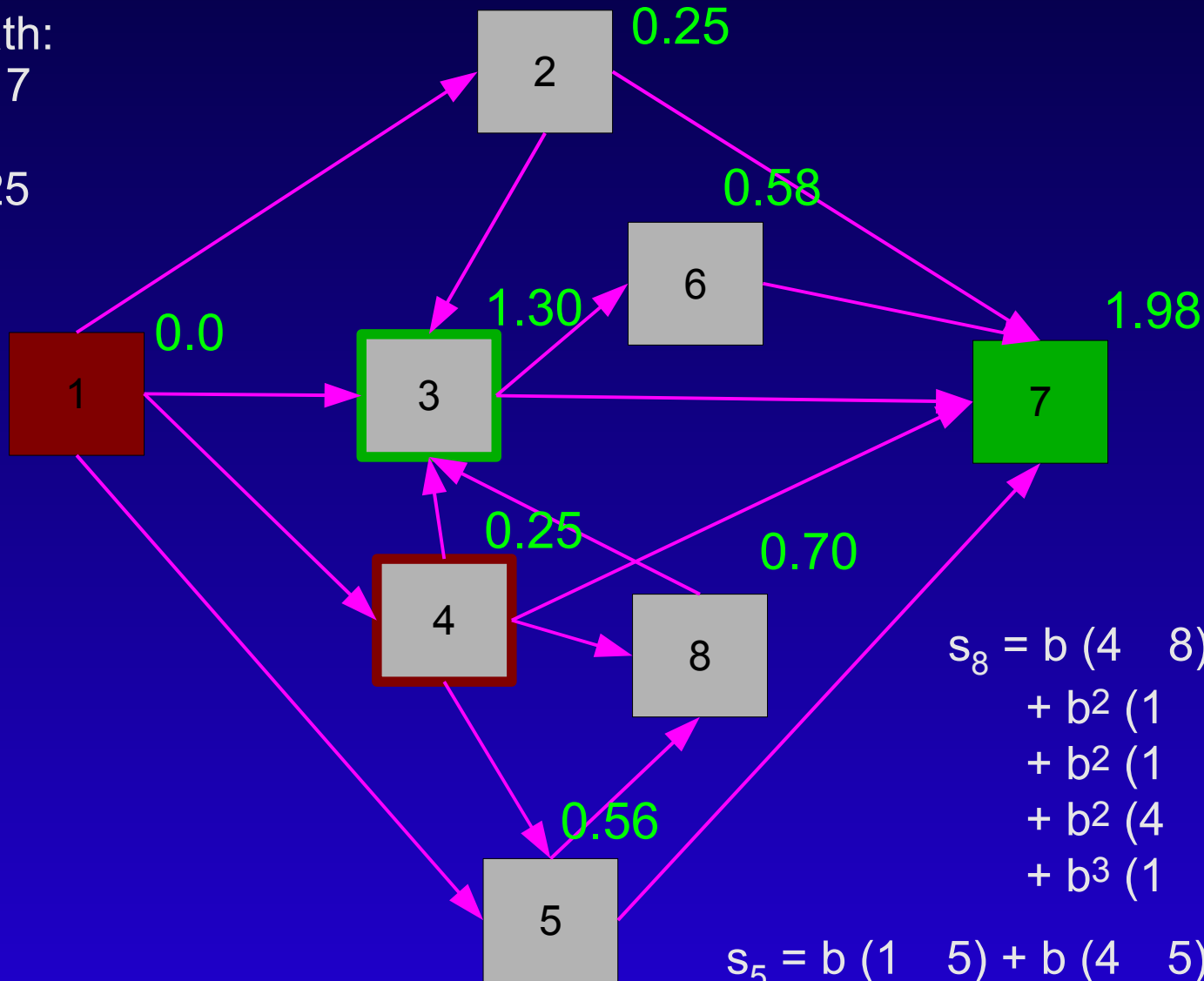
- The “status” of person  $j$  is then the sum of contributions from all other persons.

$$s_j \equiv \sum_i Q_{ij}$$

# Applying Katz's Path-counting algorithm

Longest path:  
1 4 5 8 3 6 7

use  $b = 0.25$



$$\begin{aligned}
 s_8 &= b (4 \ 8) + b (5 \ 8) \\
 &+ b^2 (1 \ 4 \ 8) \\
 &+ b^2 (1 \ 5 \ 8) \\
 &+ b^2 (4 \ 5 \ 8) \\
 &+ b^3 (1 \ 4 \ 5 \ 8)
 \end{aligned}$$

$$\begin{aligned}
 s_5 &= b (1 \ 5) + b (4 \ 5) \\
 &+ b^2 (1 \ 4 \ 5)
 \end{aligned}$$

# Features of Katz's Path-Counting algorithm

- More than just direct link counting
  - Indirect links are taken into account
- The constant  $b$  is an effectiveness of a link. It is accumulated along a path.
- Problems when there are cross links or cycles in the graph
  - Cross links: every non-zero  $C_{ij}^{(k)}$  with the path going through one node in a cross link implies nonzero  $C_{ij}^{(k+2)}$ , generally an over-estimate.
  - Not an issue with journal publications.
  - Web pages can be updated, making cycles a common phenomena.

# The HITS algorithm by Kleinberg (1999)

- HITS = Hyperlink-Induced Topic Search, a.k.a. “Hubs and Authorities” algorithm.
- From a pre-selected graph of  $N$  pages, try to find hubs (out-link dominant) and authorities (in-link dominant).
  - assign page  $k$  a hub score  $h_k$  and an authority score  $a_k$ .
  - $a_j$  is the sum of  $h_i$  over all pages  $i$  that has  $(i \rightarrow j)$  link.  
 $h_i$  is the sum of  $a_j$  over all pages  $j$  that has  $(i \rightarrow j)$  link.

$$h_i = \sum_j A_{ij} a_j, \quad a_j = \sum_i A_{ij} h_i$$

where  $A$  is the adjacency matrix.

- normalized vector  $h = (h_1, h_2, \dots, h_N)$ ,  $a = (a_1, a_2, \dots, a_N)$

$$h = \frac{A \cdot a}{|A \cdot a|}, \quad a = \frac{A^T \cdot h}{|A^T \cdot h|}$$

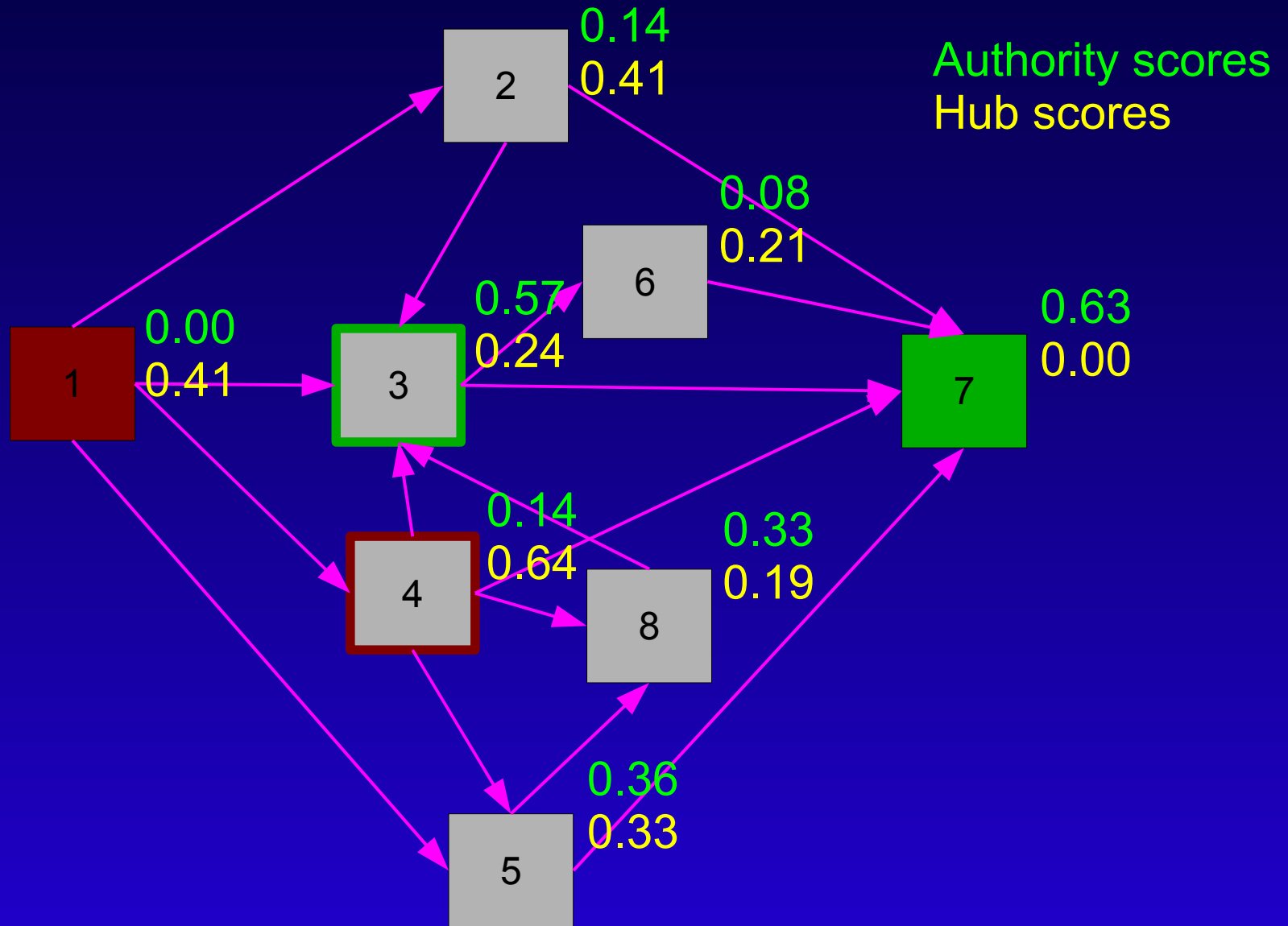
## HITS algorithm (cont.)

- Start with  $a_0 = h_0 = (1, 1, \dots, 1)$ , iterate until converge.

$$h_i = \frac{A \cdot a_{i-1}}{|A \cdot a_{i-1}|}, \quad a_i = \frac{A^T \cdot h_i}{|A^T \cdot h_i|}$$

- convergence to the vector with largest  $||$  is guaranteed by eigenvalue theory.
- Pick top  $X$  pages with highest authority scores to be the best search results.

# Applying HITS algorithm



# Pinski-Narin: Influence (1976)

- “Not all citations are created equal”
- Impact of journal  $j$  = influence weight of journal  $j$  =  $r_j$
- $W_{ij}$  = fraction of citations of journal  $i$  that goes to journal  $j$

$$L_i \equiv \sum_j A_{ij}, \quad W_{ij} \equiv \frac{A_{ij}}{L_i}$$

- Influence of journal  $j$  = weighted sums of influences of journals that cite journal  $j$

$$r_j = \sum_i W_{ij} r_i \Rightarrow W^T r = r$$

- More citations gives higher influences, citations from influential journals gives higher influences.
- $r$  is the eigenvector of matrix  $W^T$  with eigenvalue 1.
  - Question: Does such a solution always exist?

# Calculation of influence

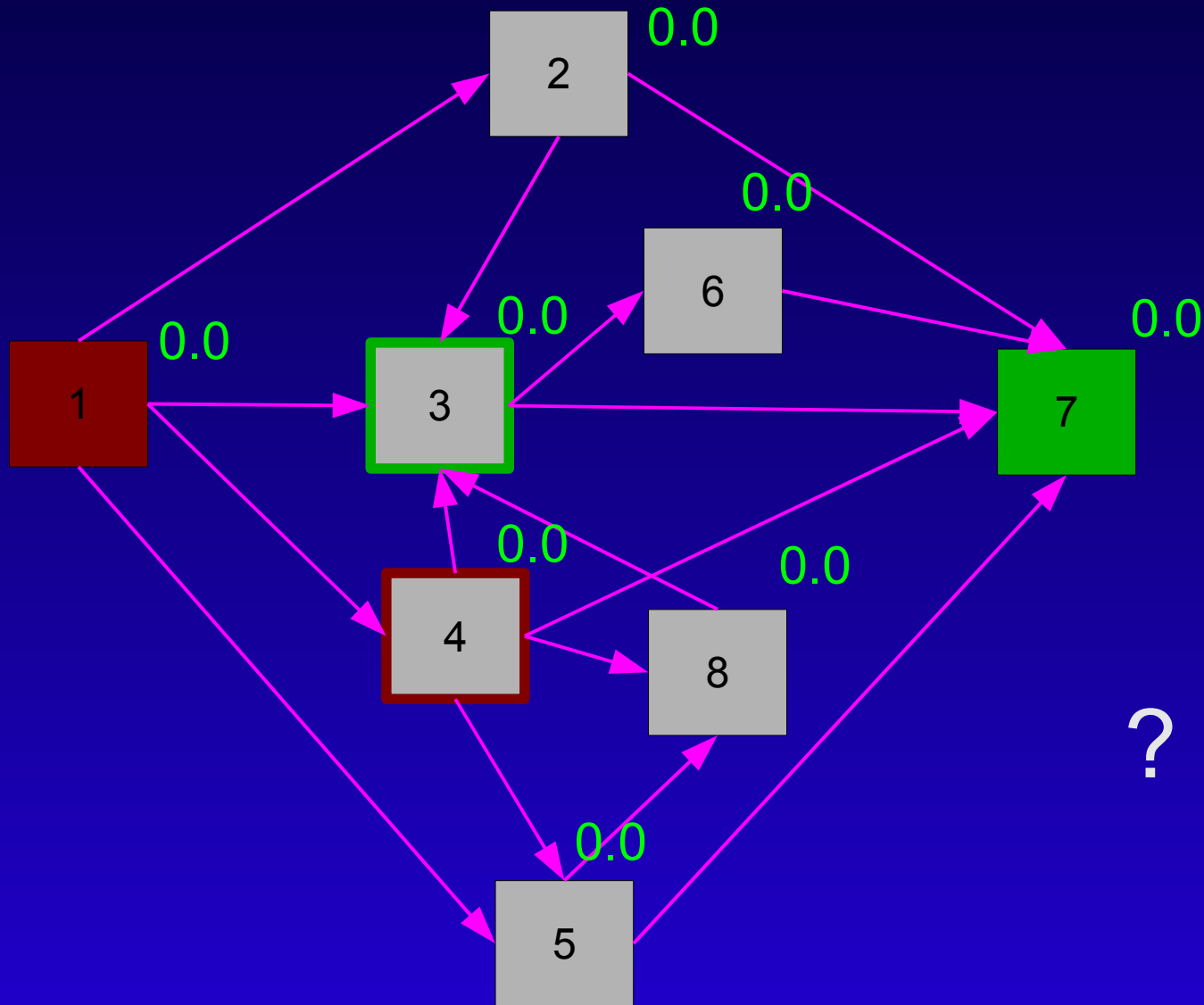
- From the eigenvalue equation

$$W^T r = r$$

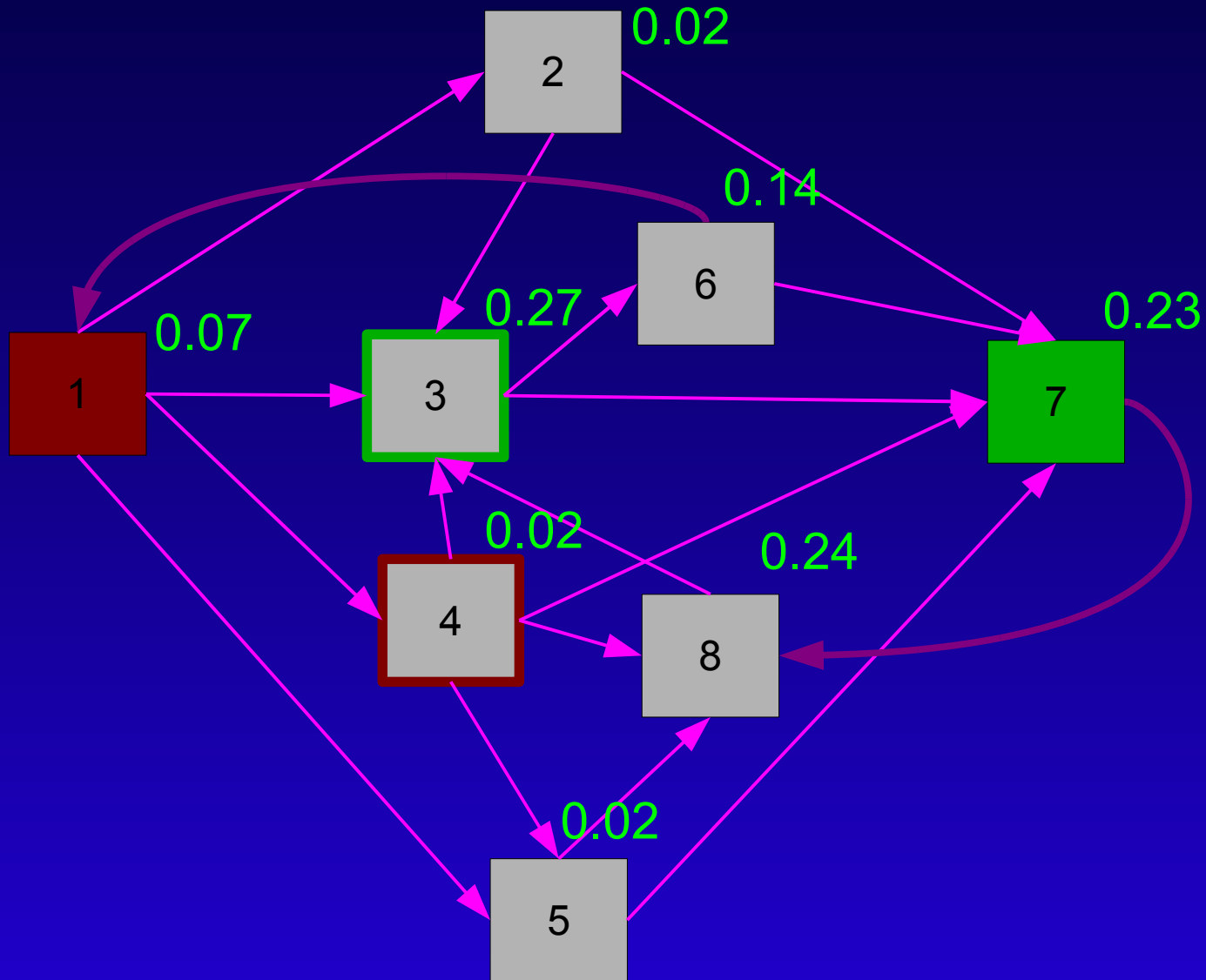
we can use the Power Method to find the eigenvector of the largest eigenvalue.

- Start with  $r_0 = (1, 1, \dots, 1)^T$ .
  - Calculate  $r_k = W^T r_{k-1}$ .
  - Repeat until convergence is reached:  $\|r_k - r_{k-1}\| < \epsilon$ .
- Questions:
    - choice of  $r_0$ ?
    - convergence?

# Applying Pinski and Narin's influence



Eliminate both the source and the sink



# Features of Pinski and Narin's Influence

- Propagation of influences
- Geller introduced Markov chain interpretation to  $W^T r = r$  in 1978.
  - $W$  is a *stochastic matrix*.
$$W_{ij} \geq 0 \quad \forall i, j$$
$$\sum_j W_{ij} = 1 \quad \forall i$$
  - Markov process: reader randomly start with a journal, then choose another journal randomly from the citations.
  - $r_j$  = the stationary probability that journal  $j$  is read. It's natural to use the L1 norm for the  $r$  vector, i.e.,  $\sum_j r_j = 1$ .
  - Perron–Frobenius theorem: principal eigenvalue = 1 for stochastic matrices.

# What's wrong?

$$W^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/2 & 0 & 1/4 & 0 & 0 & 0 & 1 \\ 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 1/2 & 1/4 & 1/2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 & 0 \end{pmatrix}$$

← The source

↑ The sink

It's no longer a transition matrix!

The sink

No solution is guaranteed if there are sinks or sources. A group of nodes linking to each other but nowhere else is also a sink. The Pinski-Narin influence algorithm is fragile.

# The PageRank algorithm

- Improve over Pinski-Narin's fragility on sinks by introducing a *random surfer model*.
  - A random surfer reads pages, randomly following links or jump to a random page:

$$r_j = \alpha \sum_i W_{ij} r_i + (1 - \alpha) v_j$$

where  $v$  is a “preference vector” as a probability distribution:

$$\|v\|_1 \equiv \sum |v_j| = 1$$

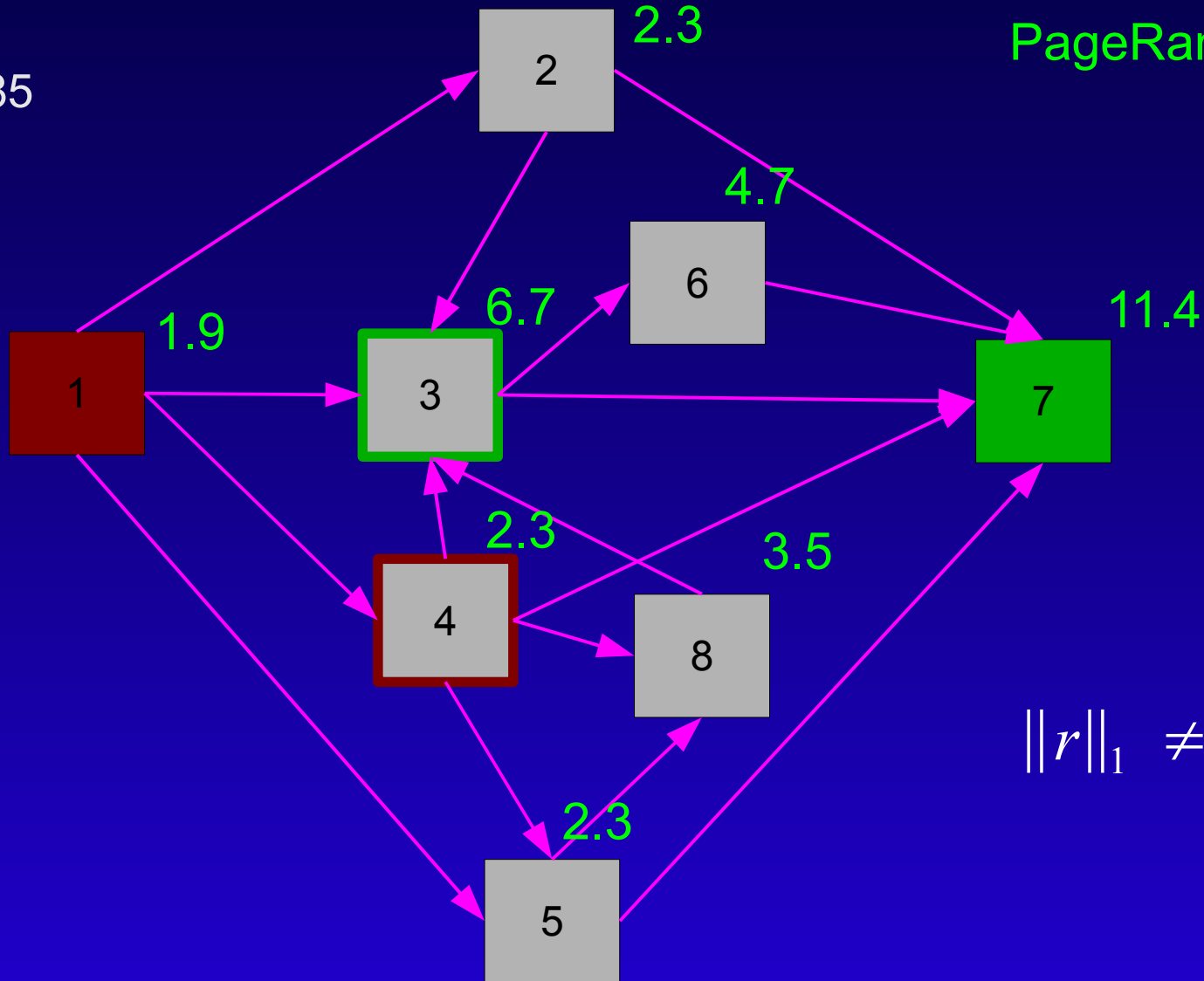
- $r_j$  = the probability that page  $j$  is read.  $\|r\|_1 = 1$
- For the case of  $v_j = \text{constant}$ , it becomes

$$r_j = \alpha \sum_i W_{ij} r_i + \frac{1 - \alpha}{N}$$

- Use power method to solve  $r$ .

# Applying PageRank algorithm

use = 0.85

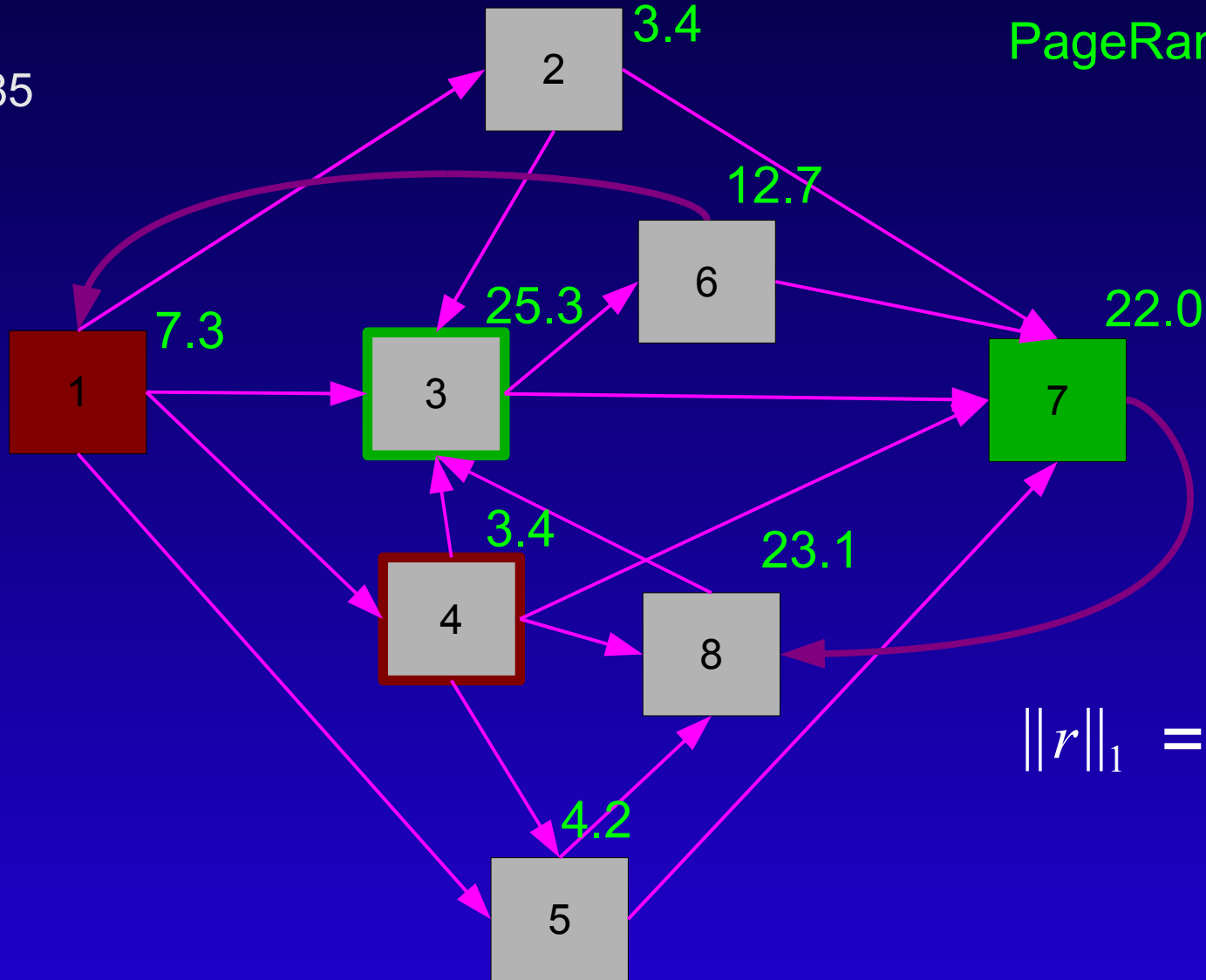


$\|r\|_1 \neq 1?$

# Without sinks and sources

use = 0.85

PageRank \* 100



$$\|r\|_1 = 1$$

# The rank sink problem

- The sink is still a problem, although convergence is reached and non-zero values are obtained. Why?
- Re-write the PageRank equation to eigenvalue form:

$$r_j = \alpha \sum_i W_{ij} r_i + (1-\alpha)v_j = \sum_i P_{ji} r_i$$

where

$$P_{ji} = \alpha W_{ij} + (1-\alpha)v_j$$

- Is P a column-stochastic matrix?

$$\sum_j P_{ji} = \alpha \sum_j W_{ij} + (1-\alpha) \sum_j v_j = \alpha + (1-\alpha) = 1 \quad \checkmark$$

- Wait... first term = 0 if node i is a sink! 

# The Google matrix

- Define  $G_{ji} \equiv \alpha(W_{ij} + a_j) + (1 - \alpha)v_j$

where  $a_j = 1/N$  if node  $j$  is a sink node, 0 otherwise.

- Now  $G$  is a column-stochastic matrix.

$$\begin{aligned} G &= \alpha(W^T + a \times e^T) + (1 - \alpha)v \times e^T \\ &= \alpha S + (1 - \alpha)E \end{aligned}$$

$e$  is a column vector with all 1's.

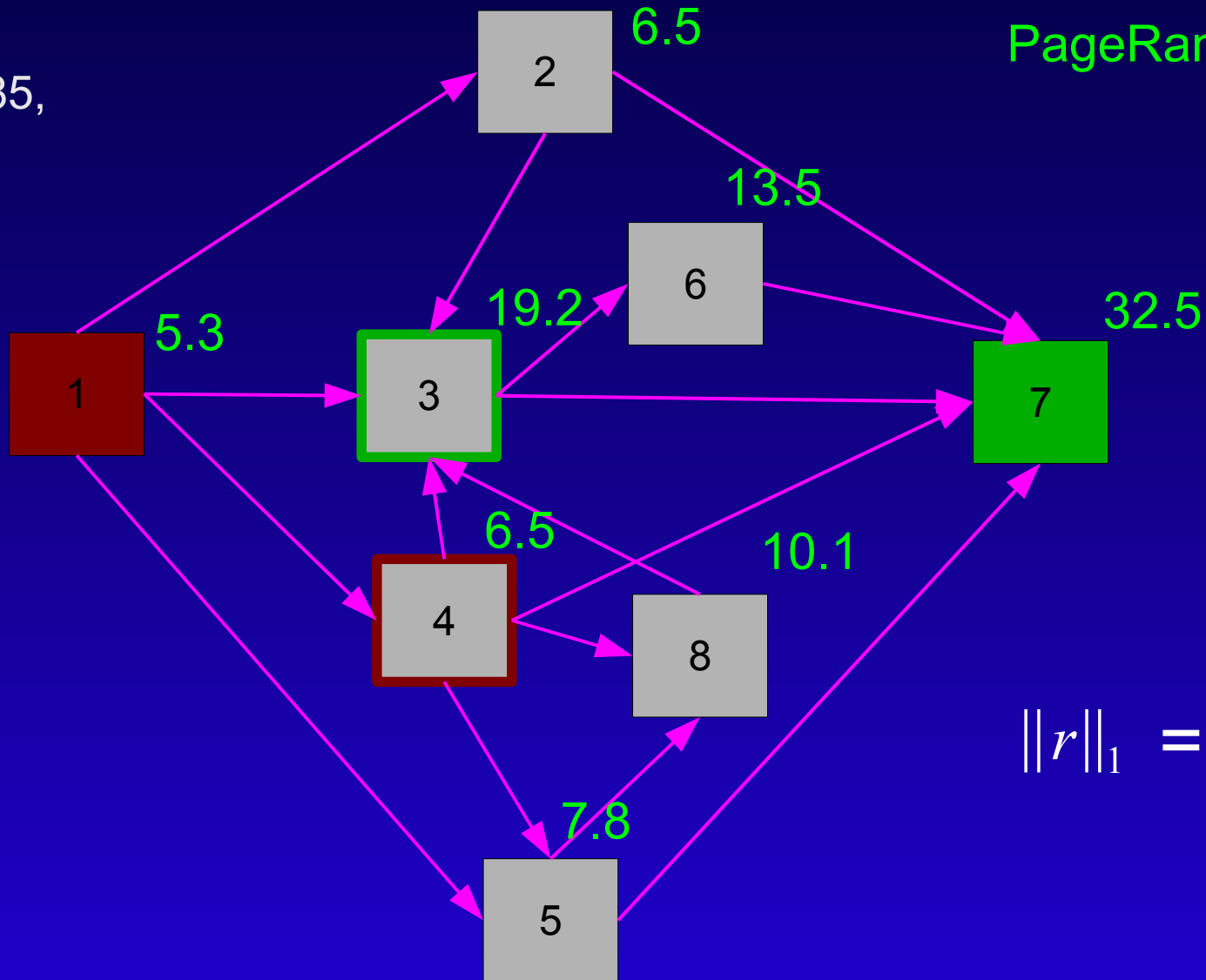
- The final PageRank equation is

$$\{\alpha S + (1 - \alpha)E\} r = r$$

- PageRank Pinski-Narin influence as  $\alpha \rightarrow 1$ .
- PageRank uniform constant as  $\alpha \rightarrow 0$ .

# PageRank from the Google matrix

use  $\alpha = 0.85$ ,  
 $v = e / N$



# Discussion on PageRank

- Is PageRank guaranteed to converge?
- What's the reasonable value of  $\alpha$ ? How sensitive is the ranks with respect to it?
- How to calculate it for the web graph?
- Independent of query – need to add query-dependent rankings for real world applications.

# Convergence of PageRank Calculation

- The speed of convergence of the power method is  $|\lambda_2|/|\lambda_1|$ 
  - It can be proven that  $|\lambda_2|/|\lambda_1| \approx \alpha$  for the Google matrix.
  - Brin and Page used 50 to 100 iterations to converge,  $0.85^{50} = 3 \times 10^{-4}$ .
- Choosing  $\alpha$  close to 1 emphasizes the link structure of the web, but the convergence will be much slower.
- Choosing  $\alpha$  close to 0 converges quickly, but the result is close to uniform ranks and useless.

# Stability of PageRank

- Evaluate the change of principal eigenvector when the Google matrix is perturbed by changing  $\alpha$  or the link structure.
- The condition number of  $G$  is shown by Kamvar and Haveliwala to be

$$\kappa = \frac{1 + \alpha}{1 - \alpha}$$

- Let  $\tilde{G} = G + \epsilon B$

$$G r = r, \quad \tilde{G} \tilde{r} = \tilde{r}$$

- Then  $\|\tilde{r} - r\|_1 \leq \kappa \epsilon \|B\| = \epsilon \frac{1 + \alpha}{1 - \alpha} \|B\|$

- Implications:

- PageRank is not stable for  $\alpha$  close to 1.
- PageRank is reasonably stable for  $\alpha < 0.9$ .

# More meanings of the $\alpha$ parameter

- Revisit the PageRank equation:

$$r = \alpha S r + (1 - \alpha) v \Rightarrow r = (I - \alpha S)^{-1} (1 - \alpha) v$$
$$= \sum_{k=0}^{\infty} \alpha^k S^k (1 - \alpha) v$$

- With uniform  $v$ , it becomes  $r_j = \frac{1 - \alpha}{N} \sum_{k=0}^{\infty} \alpha^k \sum_i S_{ji}^k$

Compare with Katz's path-counting algorithm:

$$s_j = \sum_i Q_{ij} = \sum_{k=1}^{\infty} b^k \sum_i A_{ij}^k$$

The  $\alpha$  constant plays a role of damping of influence along a path. That's why it is called “damping factor” in later literatures.

- S vs. A: S has branching factor  $1/L$ , a penalty for cycles.

# Calculating PageRank for the web

- Again, web is huge.



[Home](#) | [Select Country](#)

[Products & Services](#)

[Solutions](#)

[Academia](#)

[Support](#)

[Newsletters Main Page](#)

**News & Notes**

[2009](#)

[2008](#)

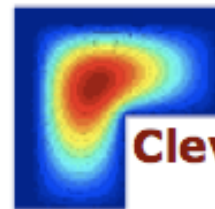
[Cleve's Corner](#)

[Past Issues](#)

**MATLAB Digest**

[November 2009](#)

## MATLAB News & Notes - October 2002



**Cleve's Corner**

**The World's Largest Matrix Computation**

**Google's PageRank is an eigenvector of a matrix of order 2.7 billion**

by [Cleve Moler](#)

# Practical considerations

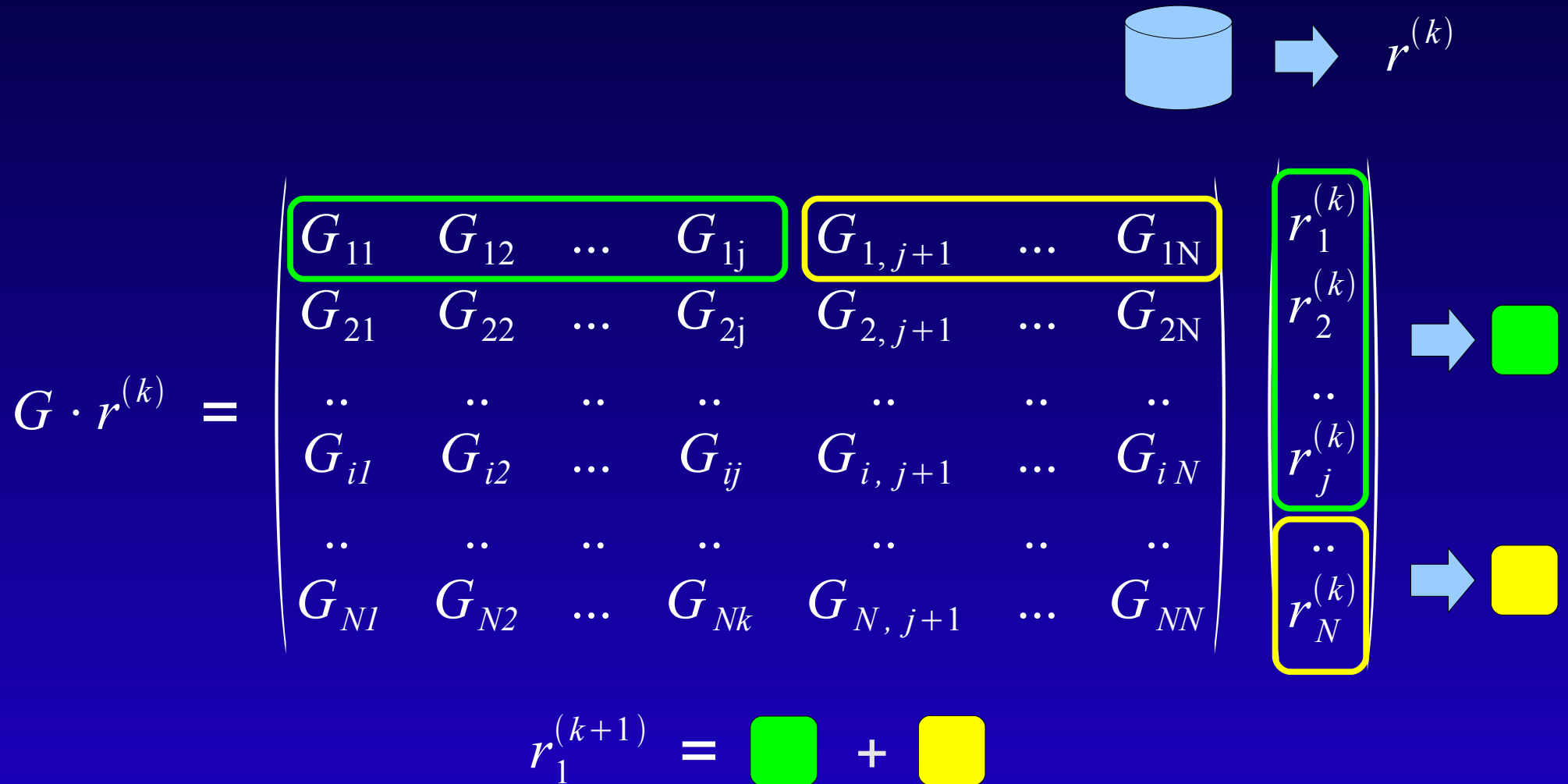
## ■ Storage:

- The  $G$  matrix is dense due to the sink nodes – make use of the sparse  $W$  matrix and rank-one updates  $a_j$  and  $v_j$ .
- Store  $W_{ij}$ ,  $a_j$ ,  $v_j$  and  $r_j^{(k)}$ :  $O(10N) + O(N) + 2N = O(N) \sim O(\text{TB})$

## ■ Calculation:

- The power method can easily be parallelized (it better is!)
- Decomposition of  $G$  matrix is one way
- Rank forwarding is another
- Use your creativity!

# The matrix decomposition method



Do you see the problem of this approach?

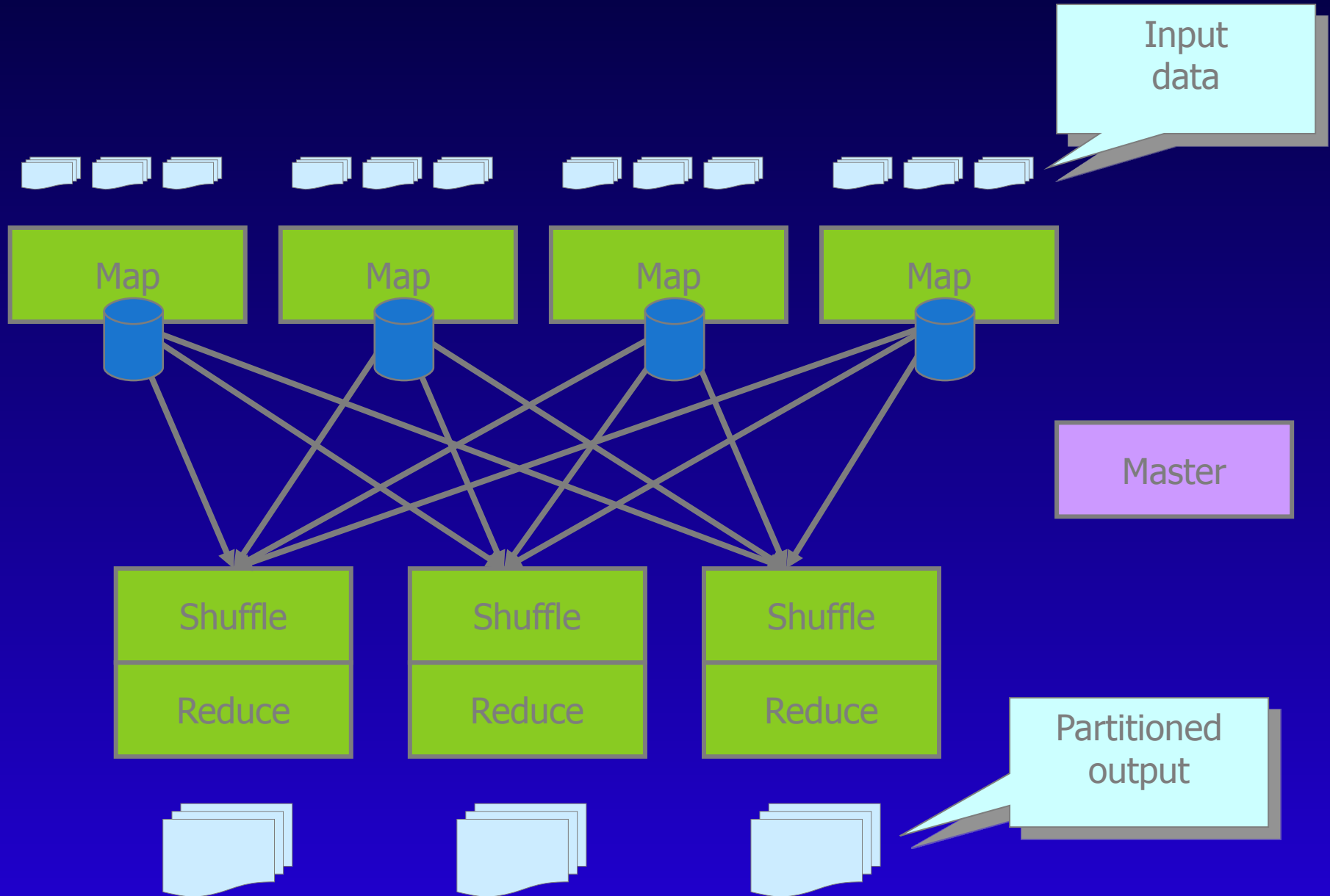
# Rank forwarding approach

- Use the equation  $r_j^{(k+1)} = \alpha \sum_i W_{ij} r_i^k + \alpha a_j + (1-\alpha)v_j$
- Phase 1: outgoing links
  - A master process reads data, and send data of one node to one process.  
input =  $i$ , out-links  $j_1, j_2, \dots, j_L$ ,  $v_i$ , PageRank  $r_i^{(k)}$   
output = key-value pairs  $(j_1, r_i^{(k)}/L), \dots, (j_L, r_i^{(k)}/L),$   
 $(i, a_i + (1-\alpha)v_i)$  {key = node id}
- Phase 2: shuffle:
  - Send (key, value) pairs of the same key to the same process
- Phase 3: incoming links
  - Each process calculates one node at a time  
input:  $(j, v_1), (j, v_2), \dots$   
output =  $(j, \sum v) = (j, r_j^{(k)})$ , and write to disk

# MapReduce

- The 3-phase calculation is applicable to various massive data processing tasks
  - Read a lot of data
  - **Map**: extract something you care about from each record
  - Shuffle and Sort
  - **Reduce**: aggregate, summarize, filter, or transform
  - Write the results
- Google developed a framework around this paradigm of parallel computing, called MapReduce.
  - Data analyst write a mapper function and a reducer function
  - Specify the input data, and the framework takes care of the rest

# Parallel MapReduce



# MapReduce scheduling

- One master, many workers
  - Input data split into  $M$  map tasks (typically 64 MB in size)
  - Reduce phase partitioned into  $R$  reduce tasks
  - Tasks are assigned to workers dynamically
  - Often:  $M=200,000$ ;  $R=4,000$ ; workers=2,000
- Master assigns each map task to a free worker
  - Considers locality of data to worker when assigning task
  - Worker reads task input (often from local disk!) to produce  $R$  local files containing intermediate k/v pairs
- Master assigns each reduce task to a free worker
  - Worker reads intermediate k/v pairs from map workers
  - Worker sorts & applies user's Reduce function to produce the output

# Some features of MapReduce

- Backup tasks can be spawned near end of completion – don't get stuck with hang tasks.
- Mapper often reads from local disk – schedule process to data, reduce network load.
- Robust fault tolerance:
  - Detect worker failure via periodic heartbeats
  - Re-execute map tasks and/or reduce tasks as necessary
  - Master state is checkpointed to Google File System: new master recovers & continues
  - Lost 1600 of 1800 machines once, but finished fine.

# Some published usage statistics

	Aug, '04	Mar, '05	Mar, '06
Number of jobs	29,423	72,229	171,834
Average completion time (secs)	634	934	874
Machine years used	217	981	2,002
Input data read (TB)	3,288	12,571	52,254
Intermediate data (TB)	758	2,756	6,743
Output data written (TB)	193	941	2,970
Average worker machines	157	232	268
Average worker deaths per job	1.2	1.9	5.0
Average map tasks per job	3,351	3,097	3,836
Average reduce tasks per job	55	144	147
Unique map/reduce combinations	426	411	2345

# Ranking Journals

# IF vs. PR by Bollen et. al. (2006)

- Formed a journal citation graph with 2003 ISI Journal Citation Reports data set. 5710 journals have non-zero citations from 2002 and 2001.
- Calculated Impact Factor and PageRank, and multiply them as the “Y-factor”.

rank	ISI IF		PR <sub>w</sub>		Y-factor	
	value	Journal	value (x 10 <sup>3</sup> )	Journal	value(x 10 <sup>2</sup> )	Journal
1	52.28	ANNU REV IMMUNOL	16.78	NATURE	51.97	NATURE
2	37.65	ANNU REV BIOCHEM	16.39	J BIOL CHEM	48.78	SCIENCE
3	36.83	PHYSIOL REV	16.38	SCIENCE	19.84	NEW ENGL J MED
4	35.04	NAT REV MOL CELL BIO	14.49	PNAS	15.34	CELL
5	34.83	NEW ENGL J MED	8.41	PHYS REV LETT	14.88	PNAS
6	30.98	NATURE	5.76	CELL	10.62	J BIOL CHEM
7	30.55	NAT MED	5.70	NEW ENGL J MED	8.49	JAMA
8	29.78	SCIENCE	4.67	J AM CHEM SOC	7.78	LANCET
9	28.18	NAT IMMUNOL	4.46	J IMMUNOL	7.56	NAT GENET
10	28.17	REV MOD PHYS	4.28	APPL PHYS LETT	6.53	NAT MED

Table 1: The highest ranking journals according to ISI IF, Weighted PageRank and Y-factor

# IF vs. PR in Physics

rank	IF	Title	$PR_w \times 10^3$	Title	$Y \times 10^2$	Title
1	28.17	REV MOD PHYS	8.41	PHYS REV LETT	5.91	PHYS REV LETT
2	13.09	ADV PHYS	4.28	APPL PHYS LETT	1.73	APPL PHYS LETT
3	11.98	PHYS REP	2.59	J APPL PHYS	1.50	REV MOD PHYS
4	10.03	MAT SCI ENG R	2.38	PHYS REV D	1.09	PHYS REV D
5	8.67	ANNU REV NUCL PART S	2.34	PHYS REV E	0.69	J CHEM PHYS
6	8.41	REP PROG PHYS	2.32	J CHEM PHYS	0.66	J HIGH ENERGY PHYS
7	7.04	PHYS REV LETT	1.56	PHYS LETT B	0.63	PHYS LETT B
8	7.00	SOLID STATE PHYS	1.55	PHYS REV A	0.57	NUCL PHYS B
9	6.06	J HIGH ENERGY PHYS	1.22	CHEM PHYS LETT	0.56	J APPL PHYS
10	5.97	PROG NUCL MAG RES SP	1.09	J HIGH ENERGY PHYS	0.56	PHYS REP

Table 5: The highest ranking Physics journals according to ISI IF, Weighted PageRank (PR<sub>w</sub>) and Y-factor.

Does Impact Factor result make sense to you?

# Issues of citation counting

- Some of the most cited journals are review articles or data tables. They are popular and of utility values, but counting such citations toward impact doesn't make much sense.
- Journal editors/referees may encourage authors to cite their journals to increase impact factor (Smith 1997).
  - Spamming the citation count “at no cost”.
- Length bias: “communications” and “letters” type of journals may cite less due to length limit.
- Many other arguments can be found on the literature.

# Conclusions

- PageRank goes beyond link counting and forwards standing/influence/impact of each node to the nodes it links to.
- PageRank vector as the stationary probability vector of the Markov chain specified by the Google matrix has many nice properties and has been extensively studied.
- It has been applied to journal status in literatures, but didn't catch on – probably the calculation is not simple enough compared to impact factor and h-index.
  - But it is actually very easy to implement for small networks!
- Maybe you can give it a try on your network

# References

- Page, Brin, Motwani, and Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” Technical Report. Stanford InfoLab (1998).
- L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika*, 18(1953), pp. 39–43.
- E. Garfield, “Citation analysis as a tool in journal evaluation,” *Science*, 178(1972), pp. 471–479.
- G. Pinski, F. Narin, “Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics,” *Inf. Proc. and Management*, 12(1976), pp. 297–312.
- N. Geller, “On the citation influence methodology of Pinski and Narin,” *Inf. Proc. and Management*, 14(1978), pp. 93–95.

## References (II)

- Amy N. Langville and Carl D. Meyer, “Google's PageRank and Beyond: The Science of Search Engine Rankings”, Princeton Press (2006).
- Jon Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” J. ACM Vol. 46, 5 (1999): 604–632.
- Ng, Zheng and Jordan, “Link analysis, eigenvectors and stability”, IJCAI 17, 1: 903-910 (2001).
- Barabási and Albert, "Emergence of scaling in random networks", Science, 286:509-512, October 15, 1999.
- Albert, Jeong and Barabási, "The Diameter of the WWW", Nature 401: 130-131 (1999).
- Bollen, Rodriguez and Van de Sompel, “Journal status”, Scientometrics 69, 3: 669-687 (2006).

## References (III)

- Seglen, “Why the impact factor of journals should not be used for evaluating research”, BMJ 1997;314:497.
- Smith, R., “Journal accused of manipulating impact factor”, BMJ 1997;314:7079.