Algorithms for Data Science
CSOR W4246

Eleni Drinea
Computer Science Department

Columbia University

Identifying important nodes on the Web:
Hubs and Authorities, and PageRank

Outline

Today

1. The structure of the WWW
2. Identifying important pages via link analysis
3. Hubs and authorities
4. PageRank

The World Wide Web

WWW enables sharing of information over the Internet by

- *creating* publicly available documents
- *accessing* publicly available documents

WWW can be modeled by a *directed graph*:

- nodes correspond to pages
- directed edges correspond to hyperlinks

The graphs in this lecture appear in "Networks, Crowds and Markets", by Easley and Kleinberg

The structure of the WWW

Classification of links on the Web

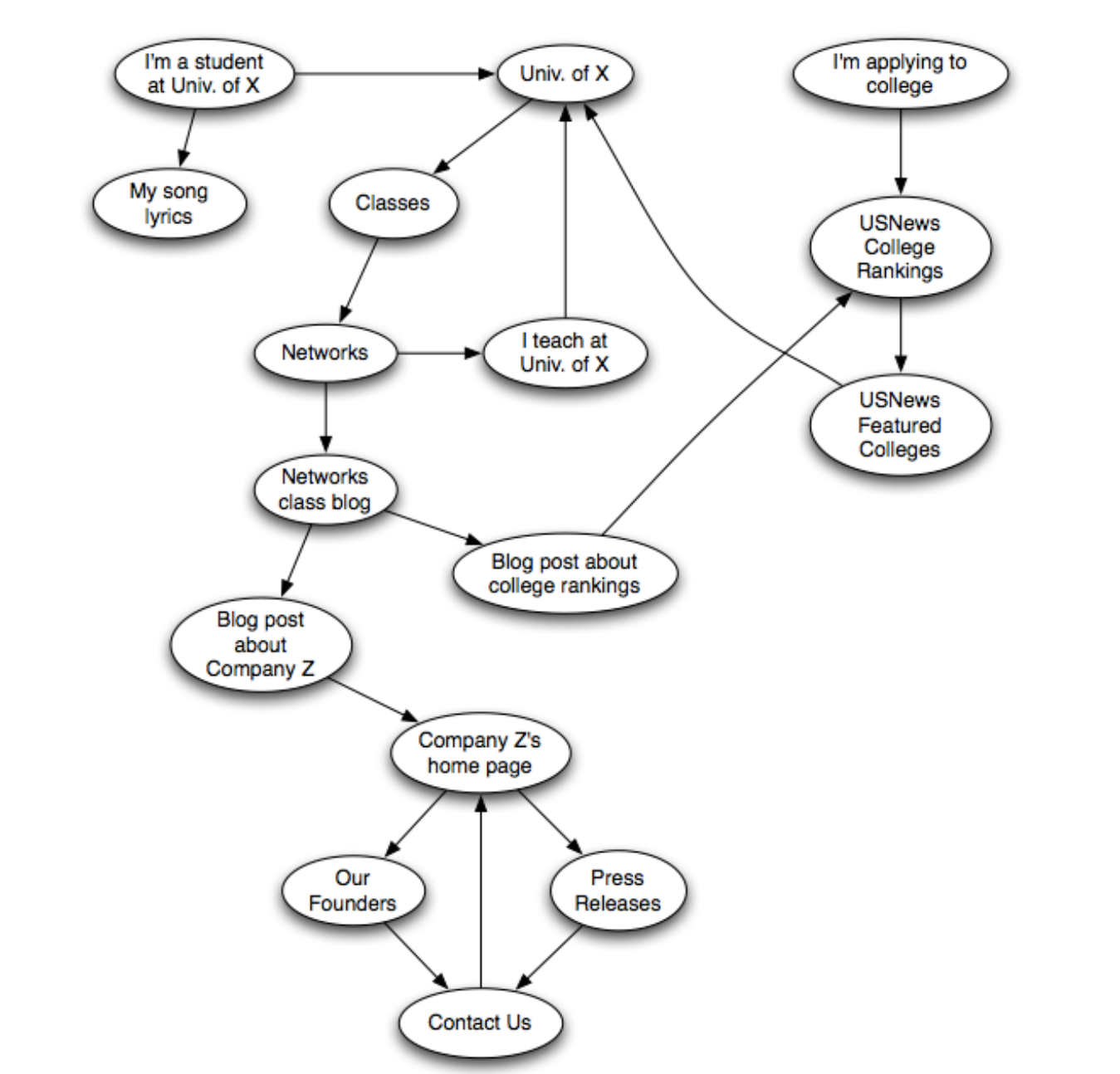
1. navigational links: transport the user between pages
   - Example: links between wikipedia pages
2. transactional links: perform computational transactions
   - Examples: "Add to shopping cart", "Buy now"

Search engines build their indexes by collecting content reachable via navigational links on the Web.

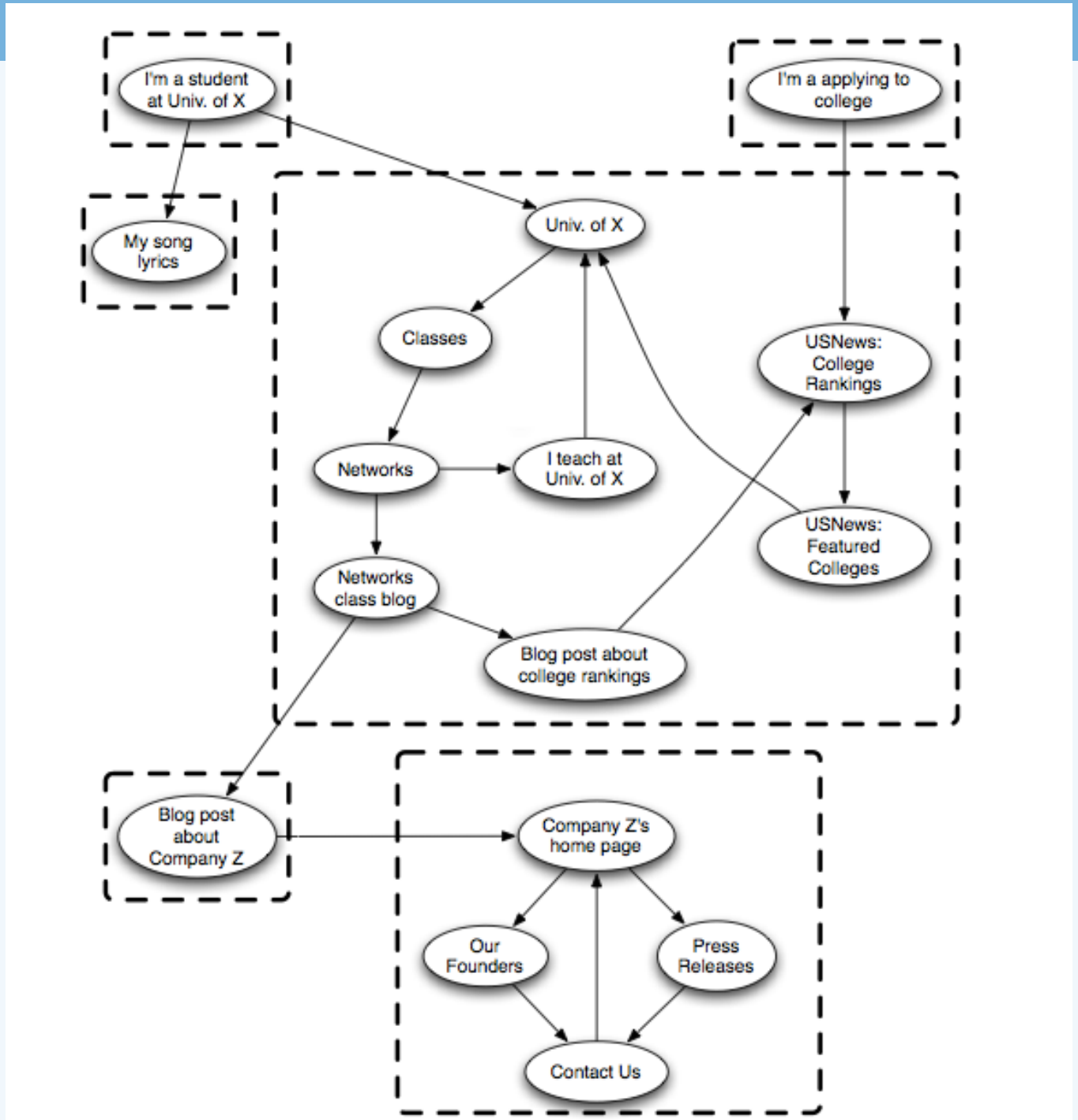
Goal: analyze the structure of its navigational "backbone"

1. decompose the Web into small cohesive units
2. identify important pages

A directed graph over a small subset of Web pages

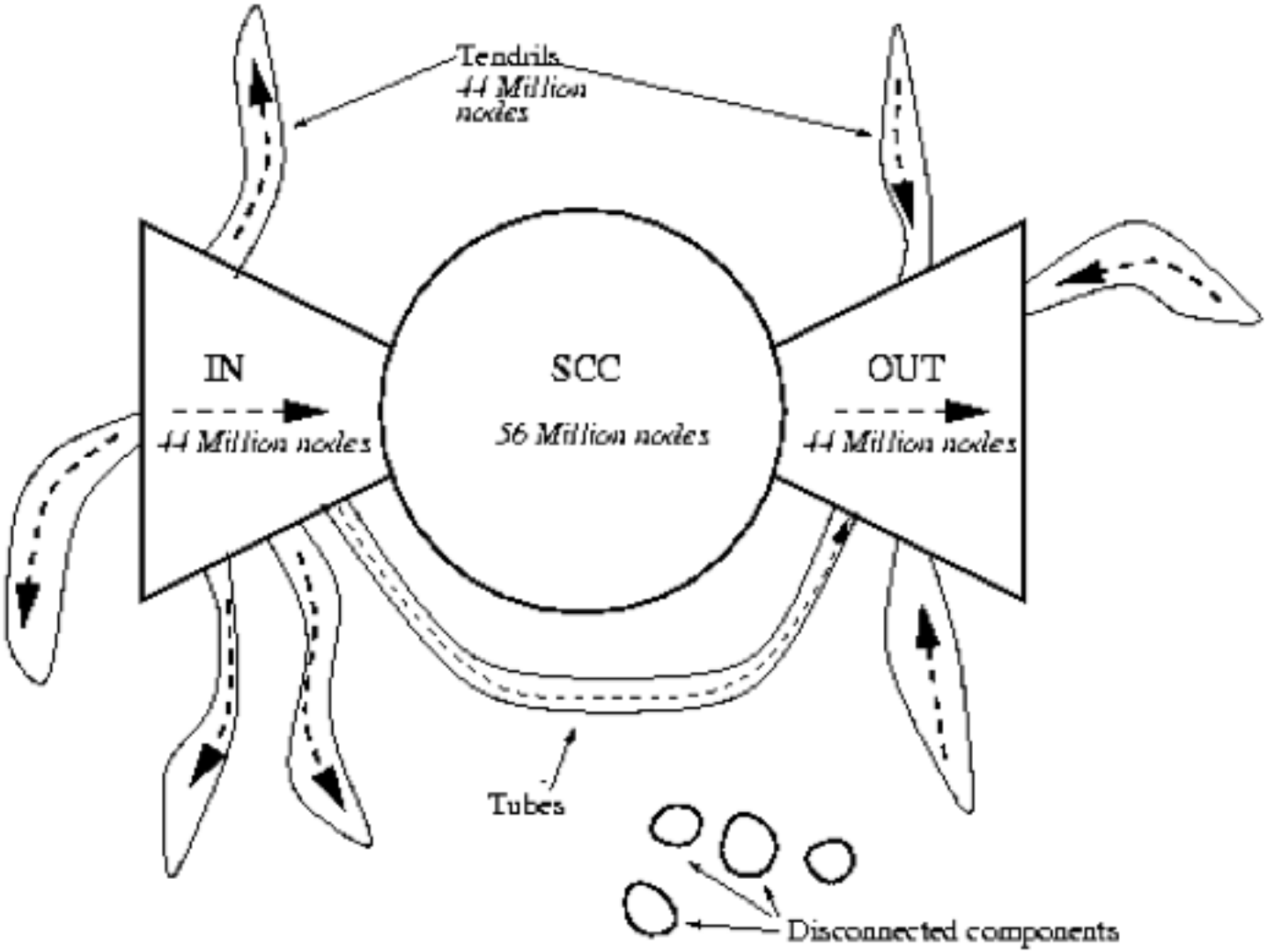


SCCs


I'm a student at Univ. of X
I'm a applying to college
My song lyrics
Univ. of X
Classes
USNews: College Rankings
Networks
I teach at Univ. of X
USNews: Featured Colleges
Networks class blog
Blog post about college rankings
Blog post about Company Z
Company Z's home page
Our Founders
Press Releases
Contact Us

The bow-tie structure of the Web

[Broder et.al., 1999]



The bow-tie structure of the Web in words

- one giant strongly connected component
- incoming nodes
- outgoing nodes
- *Other* nodes
  - Tendrils and tubes
  - Disconnected

Remark 1.

- *The bow-tie picture provides a high-level view of the Web.*
- *This picture is highly dynamic.*
- *Similar picture for* Wikipedia.
- *No information about important pages...*

Today

1 The structure of the WWW

2 Identifying important pages via link analysis

3 Hubs and authorities

4 PageRank

Challenges in...

Traditional information retrieval

- short list of keywords
- inexpressive keywords
  - synonymy: many ways to say the same thing (e.g., automobile and car)
  - polysemy: one thing has many meanings (e.g., jaguar)
- scarcity of relevant information

Web search: *which pages should the search engine recommend?*

- *everyone* can author and search on the Web!
- abundance of relevant information
- real-time awareness

Determine importance of webpages via link analysis

Suppose you search for "Columbia" in google.
Google returns `www.columbia.edu`.

Determine importance of webpages via link analysis

Suppose you search for "Columbia" in google.
Google returns `www.columbia.edu`.

How does it know?

- `www.columbia.edu` does not include the word "Columbia" more frequently or prominently than other pages.
- △ However, when a page is relevant to the query "Columbia", it very often links to `www.columbia.edu`.

Determine importance of webpages via link analysis

Suppose you search for "Columbia" in google.
Google returns `www.columbia.edu`.

How does it know?

- `www.columbia.edu` does not include the word "Columbia" more frequently or prominently than other pages.
- △ However, when a page is relevant to the query "Columbia", it very often links to `www.columbia.edu`.

Idea: if a page links to another, then essentially it endorses it.

Simple voting scheme

- Form a large set S of pages (e.g., 200) that contain the text "Columbia".
- Expand S by including all of its out-links, and a fixed number of in-links (say, at most 50) per page in S.
  - The new set S' contains pages relevant to the query.
- Output the page on the Web that receives the largest number of in-links from S'.
  - So we can think of the out-links of S' as votes.

Remark 2.

This approach works well when there is a single page that most people agree should be ranked first.

Queries with no single best answer

- Suppose you search for “newspapers”.
- There is no **single** “best” answer: ideally, the search should return a list of few prominent newspapers.

Queries with no single best answer

- Suppose you search for "newspapers".
- There is no **single** "best" answer: ideally, the search should return a list of few prominent newspapers.

Suppose we try the same approach as before:

1. Form a large set S of pages that contain the text "newspapers"; expand it into S' as before.
2. Output the pages on the Web with highest number of in-links from S'.

Queries with no single best answer

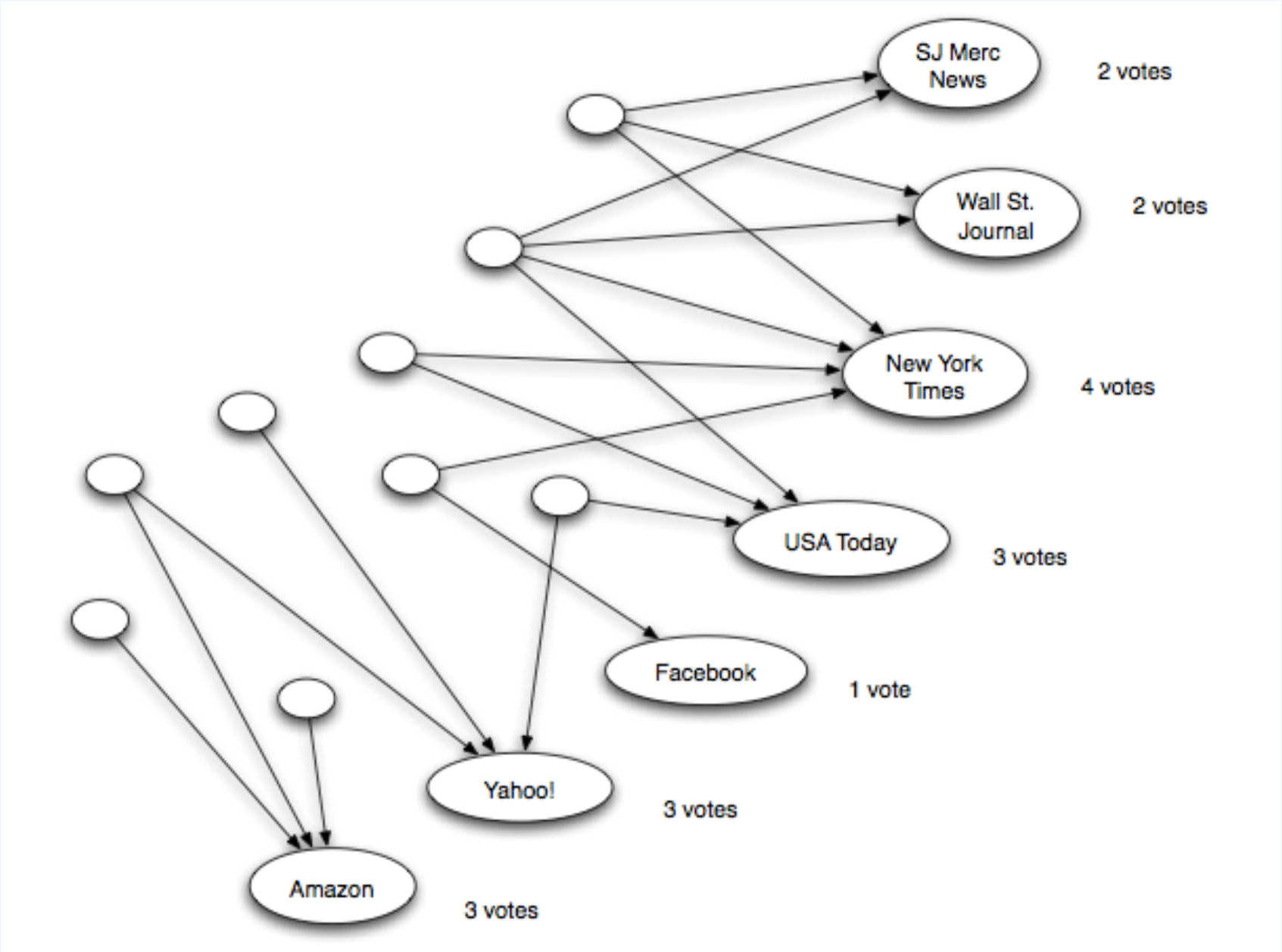
- Suppose you search for "newspapers".
- There is no **single** "best" answer: ideally, the search should return a list of few prominent newspapers.

Suppose we try the same approach as before:

1. Form a large set S of pages that contain the text "newspapers"; expand it into S' as before.
2. Output the pages on the Web with highest number of in-links from S'.

Typically, the output will consist of a mix of prominent newspapers and irrelevant pages with high numbers of in-links.

Output pages for the query "newspapers"



Today

Some votes are more important than others

- Suppose we identified a set of pages that are good lists of pages relevant to "newspapers".
  - We can think of such pages as good hubs.
- Intuitively, we should weigh the votes (out-links) of good hubs more in our search.

How can we filter for such hubs?

Some votes are more important than others

- Suppose we identified a set of pages that are good lists of pages relevant to "newspapers".
  - We can think of such pages as good hubs.
- Intuitively, we should weigh the votes (out-links) of good hubs more in our search.

How can we filter for such hubs?

Within our initial set of pages, few pages voted for many of the pages that received lots of votes.

- We can think of the latter pages as good authorities.

$\Rightarrow$ Pages who voted for many good authorities are good hubs.

The principle of repeated improvement

Intuition: votes from good hubs should receive more weight.

Define

- x_p = authority score of page p
- y_p = hub score of page p

Initially, $x_p = y_p = 1$ for every page p

The principle of repeated improvement

Intuition: votes from good hubs should receive more weight.

Define

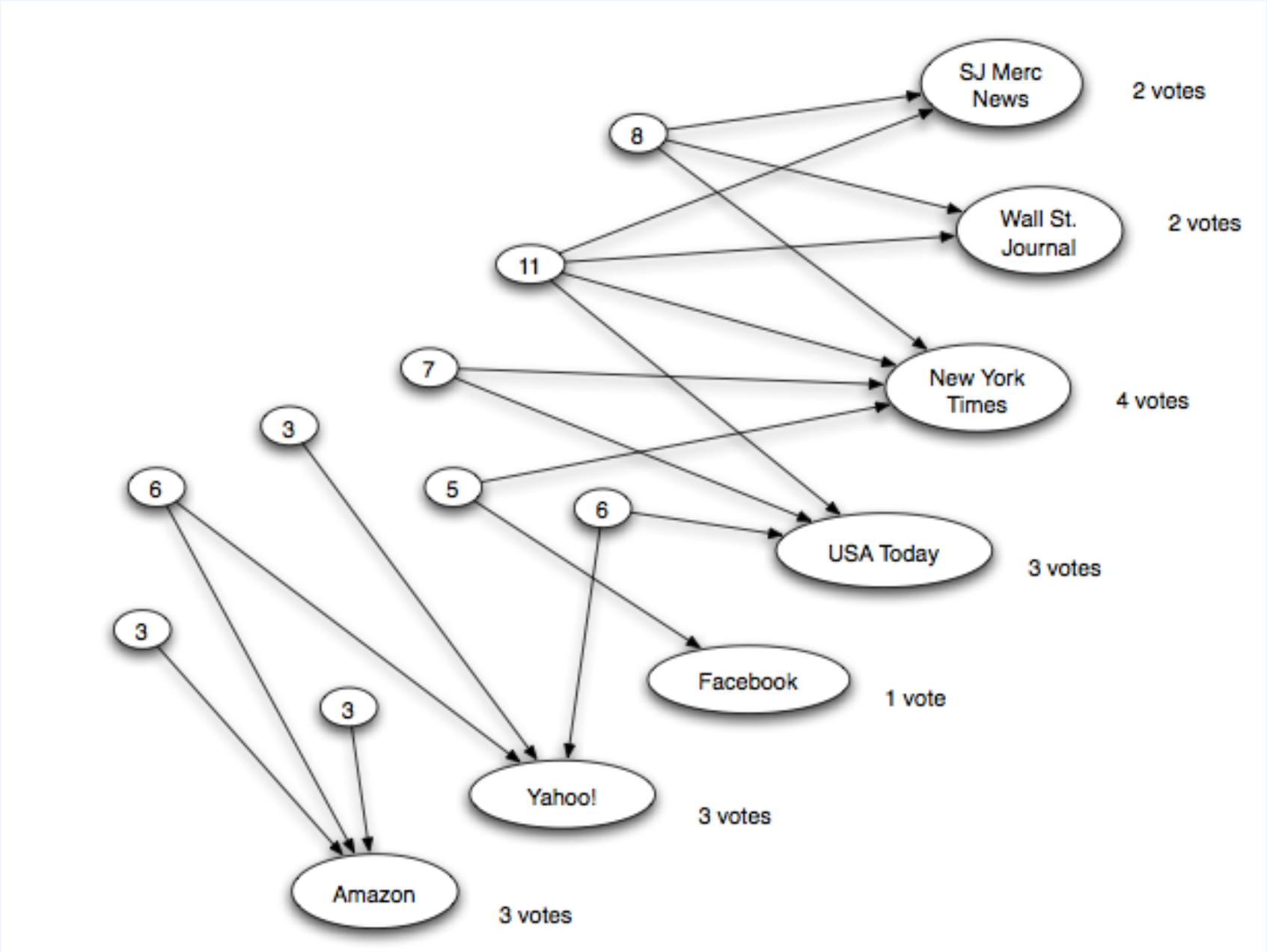
- x_p = authority score of page p
- y_p = hub score of page p

Initially, $x_p = y_p = 1$ for every page p

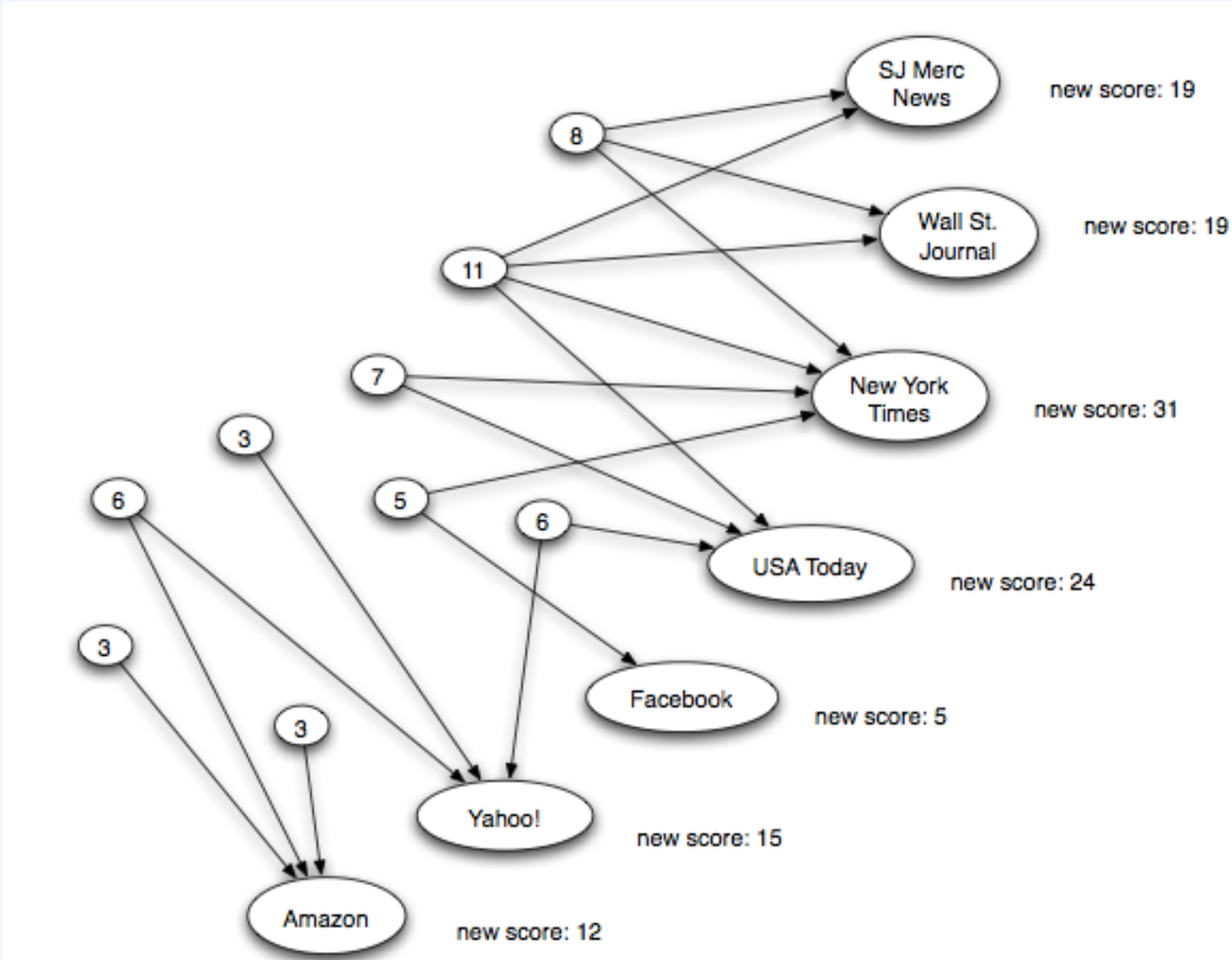
Principle of repeated improvement: keep refining estimates for hub and authority scores

- Authority Update rule: for each page p, update its authority score x_p as $\sum_{q:(q,p)\in E} y_q$
- Hub Update rule: for each page p, update its hub score y_p as $\sum_{q:(p,q)\in E} x_q$

1st hub score update for the query "newspapers"



2nd authority score update for the query "newspapers"



Short description of the algorithm

Short description: the algorithm chooses a number of steps k and performs a sequence of k authority-hub updates.

1. Let $G = (V, E)$ be the directed graph induced by the pages in S, and the pages pointed to by the out-edges of S.
2. Let $\mathbf{x^{(i)}}, \mathbf{y^{(i)}}$ be vectors in R^n such that
   - $\mathbf{x^{(i)}} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})$ is the vector of authority scores at the end of iteration i
     - hence the p-th coordinate of $\mathbf{x}^{(i)}$, denoted by $x_p^{(i)}$, is the authority score of page p
   - $\mathbf{y^{(i)}} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_n^{(i)})$ is the vector of hub scores at the end of iteration i
     - hence $y_p^{(i)}$ is the hub score of page p

Hubs and authorities algorithm

`Iterate`$(G = (V, E), k)$

Let $\mathbf{z}$ be the vector $(1, 1, \ldots, 1) \in R^n$.
Set $\mathbf{x^{(0)}} = \mathbf{z}$
Set $\mathbf{y^{(0)}} = \mathbf{z}$
for $i = 1, 2, \ldots, k$ **do**
  for $p \in V$ **do** set $x_p^{(i)} = \sum_{q:(q,p)\in E} y_q^{(i-1)}$
  for $p \in V$ **do** set $y_p^{(i)} = \sum_{q:(p,q)\in E} x_q^{(i)}$
  Normalize the vector $\mathbf{x^{(i)}}$
  Normalize the vector $\mathbf{y^{(i)}}$
end for
return $(\mathbf{x^{(k)}}, \mathbf{y^{(k)}})$

`Filter`(G, k, c)

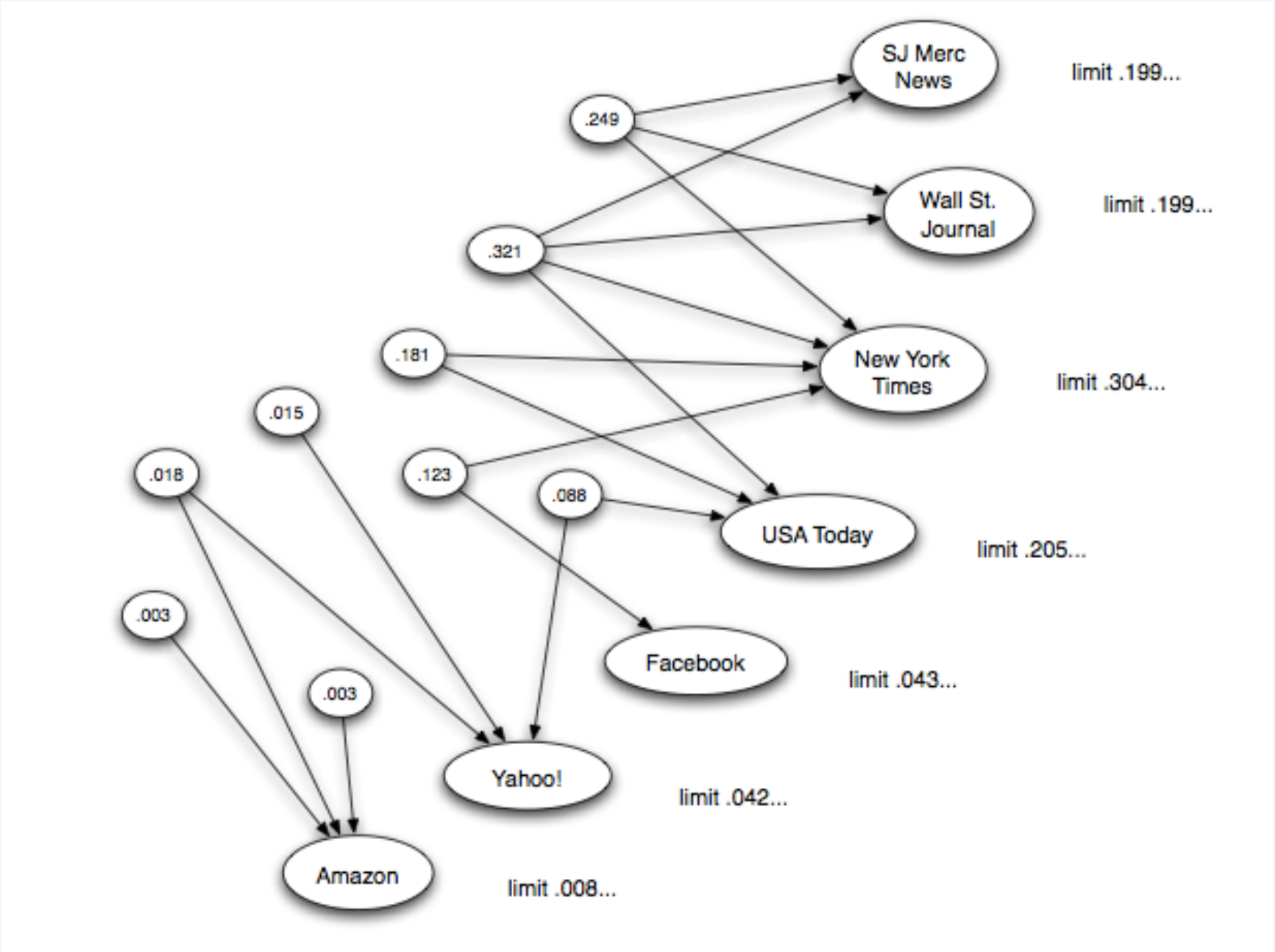
$(\mathbf{x^{(k)}}, \mathbf{y^{(k)}}) =$ `Iterate`(G, k)
Set as authorities the pages with the c largest coordinates in $\mathbf{x^{(k)}}$
Set as hubs the pages with the c largest coordinates in $\mathbf{y^{(k)}}$

Limiting hubs and authorities scores

- The normalized values converge to limiting values as $k \to \infty$.
- The same limiting values are reached regardless of the initial hub and authority scores (provided they are all positive).

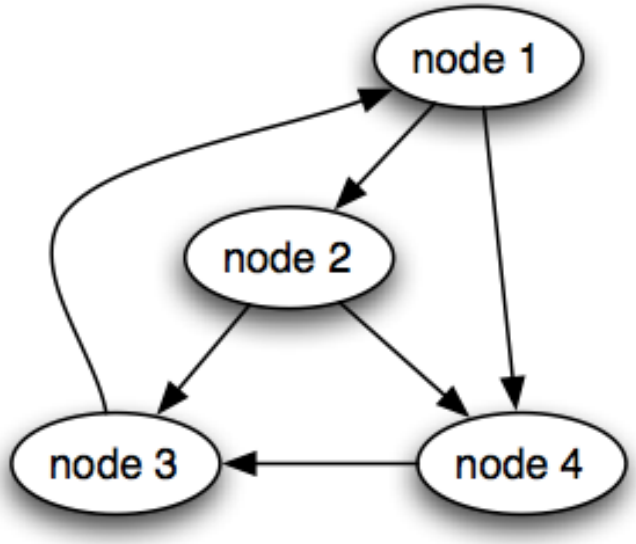
$\Rightarrow$ The limiting hub and authority scores are a property of the link structure.

Limiting hub and authority scores



Using the adjacency matrix

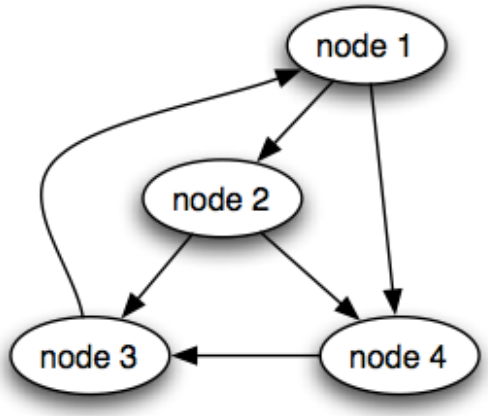
A graph G and its associated adjacency matrix A



$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Example

A graph G, its associated adjacency matrix A, a vector of authority scores $\mathbf{x^{(i)}} = [2\ 6\ 4\ 3]^T$ and the derived hubs scores $\mathbf{y^{(i)}} = [9\ 7\ 2\ 4]^T$



$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 6 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 9 \\ 7 \\ 2 \\ 4 \end{bmatrix}$$

Matrix notation for hubs and authorities algorithm

`Iterate`$(G = (V, E), k)$

Let $\mathbf{z} = [1\ 1\ \ldots\ 1]^T$

Set $\mathbf{x_0} = \mathbf{y_0} = \mathbf{z}$

for $i = 1, 2, \ldots, k$ **do**

Set $\mathbf{x^{(i)}} = A^T \mathbf{y^{(i-1)}}$

Set $\mathbf{y^{(i)}} = A\mathbf{x^{(i)}}$

Normalize $\mathbf{x^{(i)}}$ to have unit length

Normalize $\mathbf{y^{(i)}}$ to have unit length

end for

return $(\mathbf{x^{(k)}}, \mathbf{y^{(k)}})$

Remark 3.

- $\mathbf{y^{(k)}}$ *is the unit vector in the direction of* $(AA^T)^k \mathbf{z}$
- $\mathbf{x^{(k)}}$ *is the unit vector in the direction of* $(A^T A)^{k-1} A^T \mathbf{z}$.

Today

1. The structure of the WWW
2. Identifying important pages via link analysis
3. Hubs and authorities
4. PageRank

How pages may endorse each other

- Queries with a commercial aspect: competing firms will not link to each other.
  - So endorsement does not pass from one authority to the other directly.
  - Thus hubs are necessary to pull authorities together.

How pages may endorse each other

- Queries with a commercial aspect: competing firms will not link to each other.
  - So endorsement does not pass from one authority to the other directly.
  - Thus hubs are necessary to pull authorities together.

- Academic, governmental or personal pages: endorsement passes directly from one web page to the other.
  - PageRank algorithm: applies the Principle of Repeated Improvement by a node-to-node endorsement-propagation scheme.

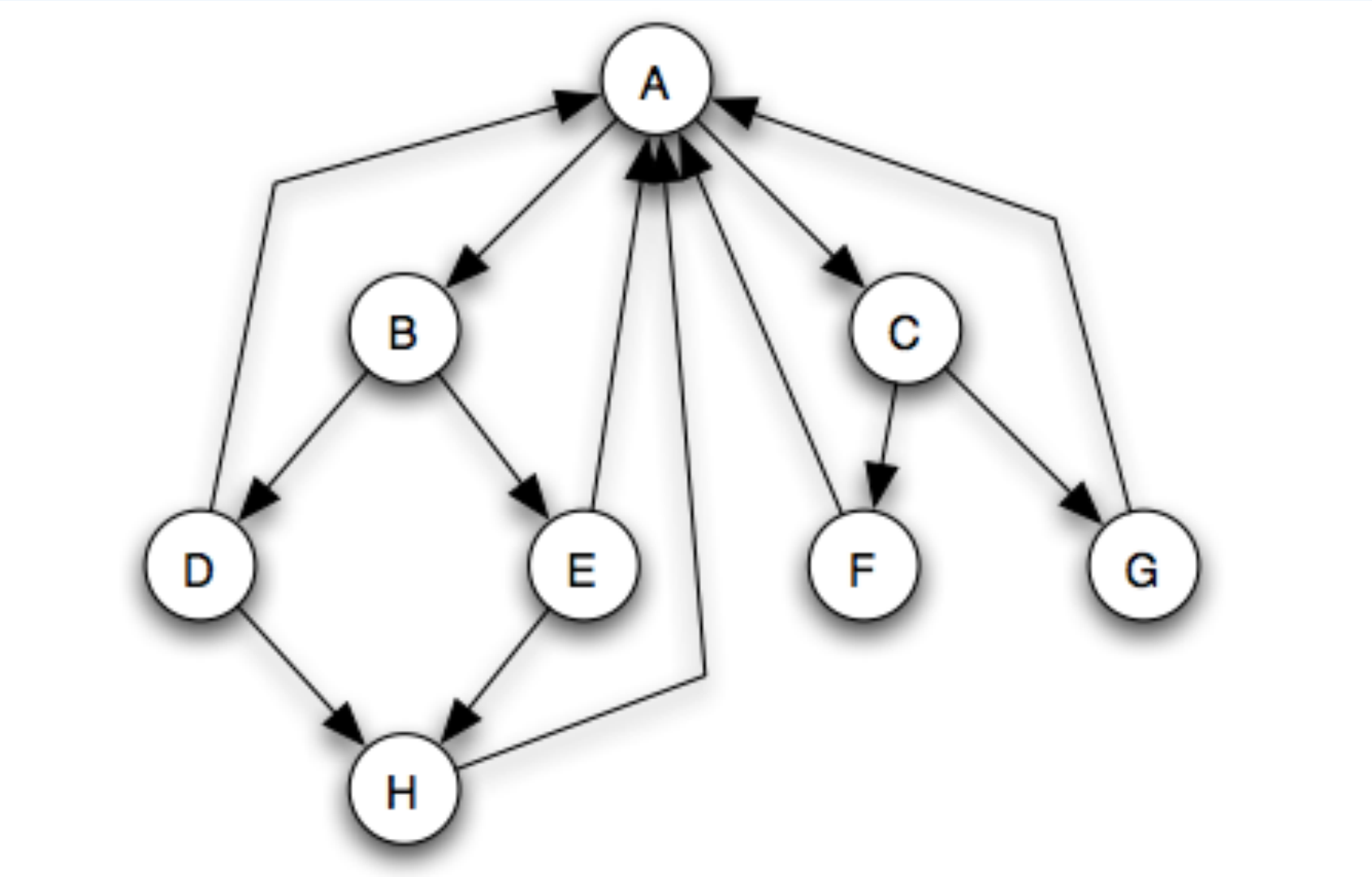
The basic PageRank Update rule

Short description: the algorithm chooses a number of steps k and performs a sequence of k basic PageRank updates.

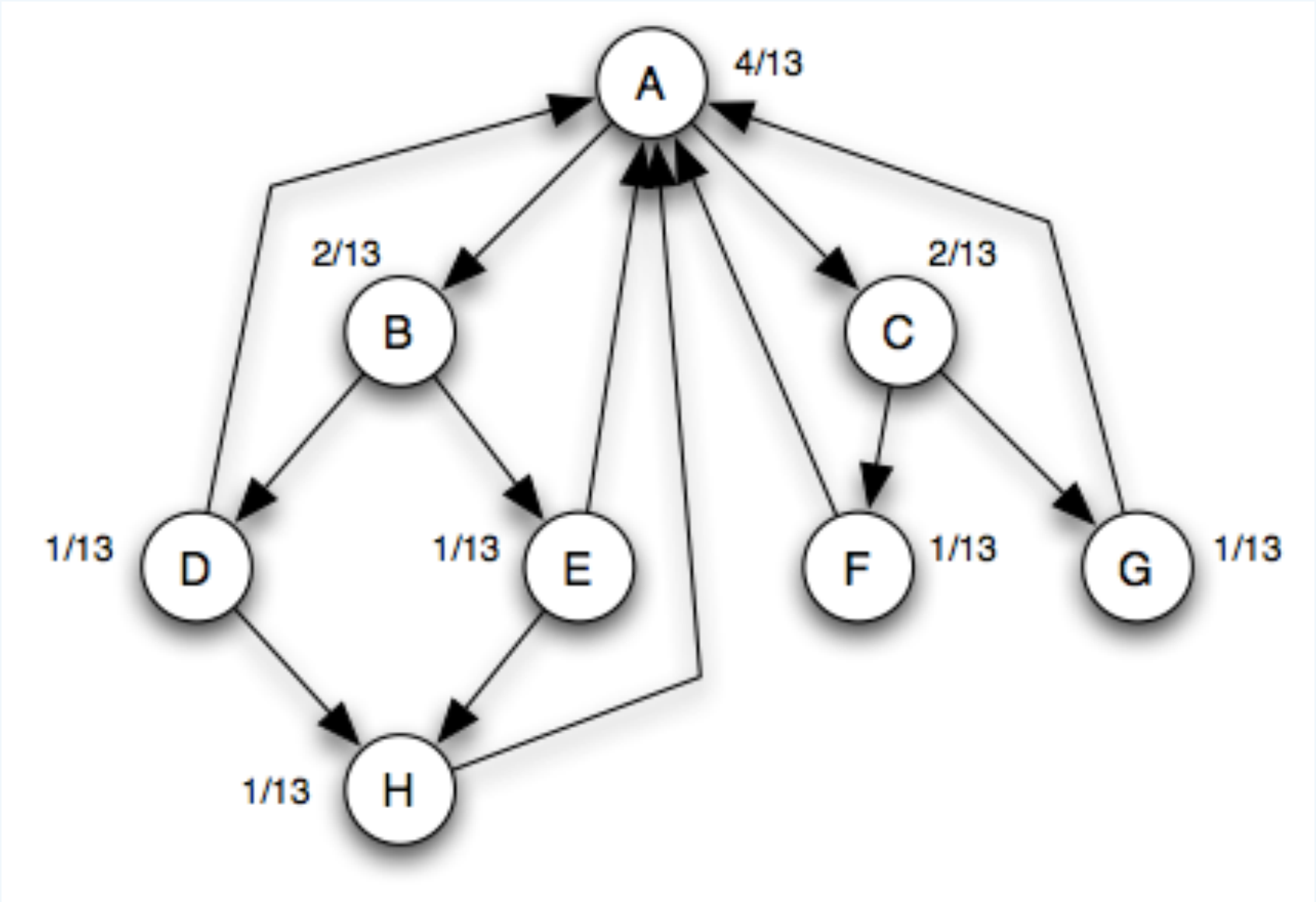
Basic PageRank Update rule:

1. Each page divides its current PageRank equally across its out-going links.
2. Each page passes these equal shares to the pages it points to.
3. Each page updates its new PageRank to be the sum of the shares it receives.

Example: PageRank computation



PageRank equilibrium



Basic PageRank algorithm

Notation

1. $G = (V, E)$ is the directed graph of an index of the Web
2. $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ is the vector of PageRank scores
3. $d_p =$ out-degree(p)

`Basic Pagerank`(G, k)

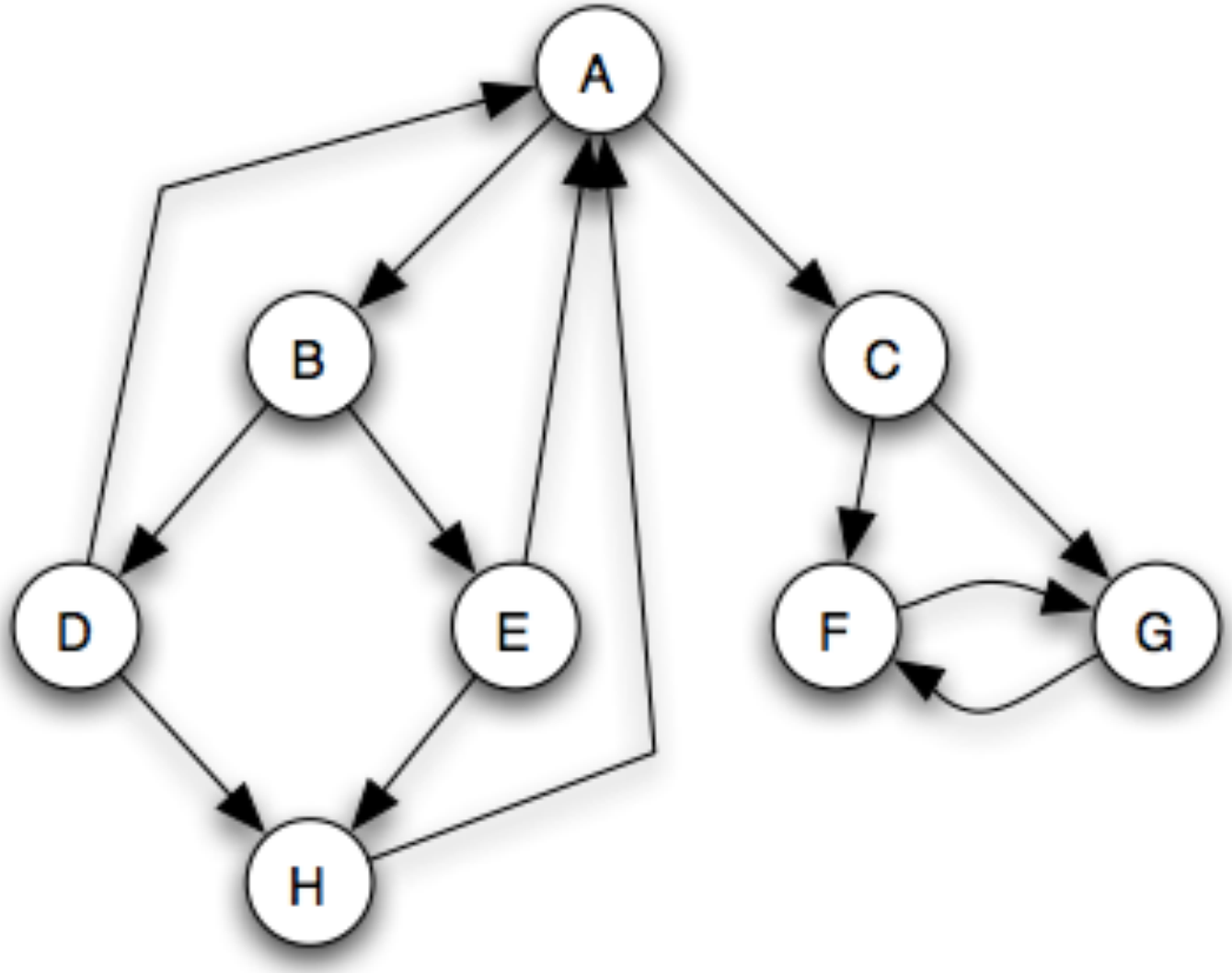
Initialize $\mathbf{r}$ to $(1/n, 1/n, \ldots, 1/n)$
for $i = 1, 2, \ldots, k$ **do**
  for $p \in V$ **do**
    $$r'_p = \sum_{q:(q,p)\in E} r_q/d_q$$
  end for
  $\mathbf{r} = \mathbf{r}'$
end for
return $\mathbf{r}$

Remarks on Basic PageRank algorithm

Remark 4.

1. *Total PageRank remains* 1 *throughout the execution of the algorithm; it just gets redistributed.*
2. *PageRank scores converge to limiting values as* $k \to \infty$.

Basic PageRank update rule issue: "slow leaks"



Scaled PageRank update rule

Fix: use a scaling factor d between 0 and 1 to redistribute a fraction of the total PageRank to **every** node.

Scaled PageRank update rule:

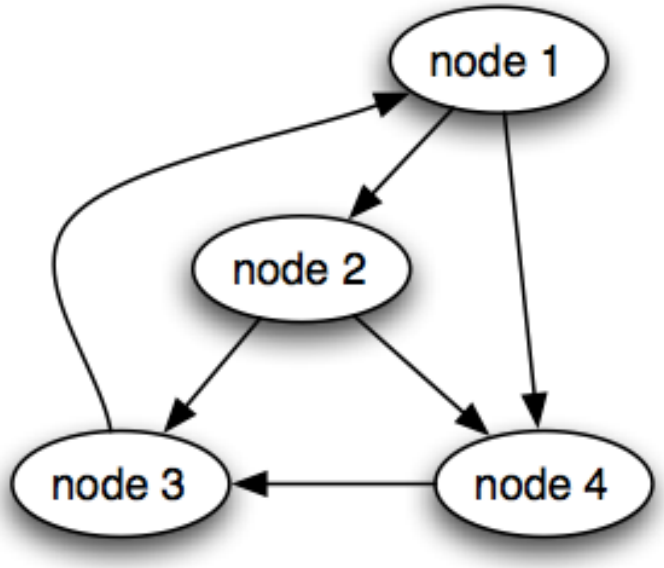
1. Apply the Basic PageRank update rule.
2. Scale down all PageRank scores by a factor of d; thus the total PageRank in the network has shrunk to d.
3. Divide the residual $1 - d$ units of PageRank equally among all nodes, giving $(1 - d)/n$ to each.

Remark 5.

- *Limiting values for Scaled PageRank exist and are unique, given scaling factor d.*
- *Typical values for d are between* 0.8 *and* 0.9.

Example for Basic PageRank rule

A graph G and the associated normalized by out-degrees adjacency matrix N'



$$\begin{bmatrix} 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Rewriting the Basic PageRank algorithm

Let N' be the normalized by out-degrees adjacency matrix of the Web where $N'_{ij} = A_{ij}/\text{out-deg}(i)$.

Basic Pagerank(N', k)

$\mathbf{r^{(0)}} = (1/n, 1/n, \ldots, 1/n)^T$

for $i = 1, 2, \ldots, k$ **do**

$\mathbf{r^{(i)}} = (N')^T \mathbf{r^{(i-1)}}$

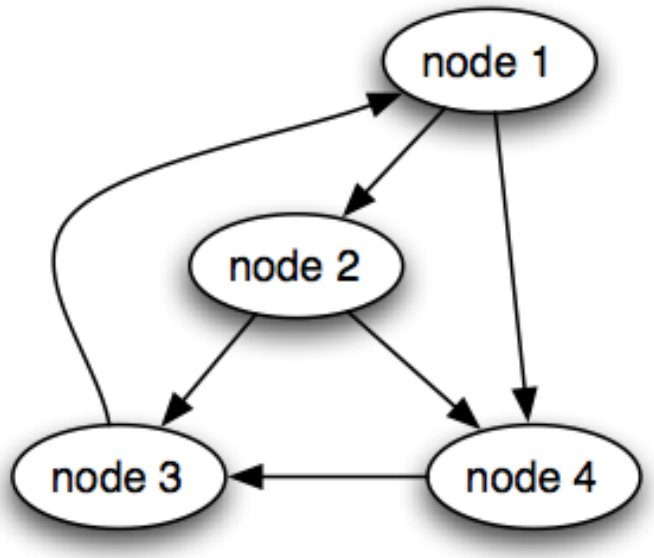
end for

return the vector $\mathbf{r^{(k)}}$

We can compute Scaled PageRank by replacing N' with N, where $N_{ij} = (1-d)/n + dN'_{ij}$.

Example for Scaled PageRank rule

A graph G and the associated scaled and normalized by out-degrees adjacency matrix N (scaling factor $d = .8$)



$$\begin{bmatrix} .05 & .45 & .05 & .45 \\ .05 & .05 & .45 & .45 \\ .85 & .05 & .05 & .05 \\ .05 & .05 & .85 & .05 \end{bmatrix}$$

The PageRank vector in the limit

- At steady state, we expect $\mathbf{r}^* = N^T \mathbf{r}^*$
- So $\mathbf{r}^*$ is an eigenvector of N^T with corresponding eigenvalue 1
- It can be shown (by Perron's theorem for matrices with positive entries and largest eigenvalue 1) that for any initial vector $\mathbf{x}$ with non-negative entries, as $k \to \infty$, the sequence of vectors $(N^T)^k \mathbf{x}$ converges to a vector in the direction of the leading eigenvector $\mathbf{r}^*$ ($\mathbf{r}^*$ corresponds to $\lambda = 1$ and has positive real coordinates); in particular, this holds for $\mathbf{x} = \mathbf{r}^{(}\mathbf{0})$.

The random surfer model (basic PageRank)

Consider a user randomly browsing the Web as follows

- First, the user picks a page at random with equal probability.
- Then he follows links for k steps: in each step he follows a random out-going link from his current page (if the page has no outgoing links, then he stays there).

This process defines a random walk on the network.

The random surfer model (basic PageRank)

Consider a user randomly browsing the Web as follows

- First, the user picks a page at random with equal probability.
- Then he follows links for k steps: in each step he follows a random out-going link from his current page (if the page has no outgoing links, then he stays there).

This process defines a random walk on the network.

Claim 1.

The probability of being at page p after k steps of this random walk is precisely the PageRank of p after k applications of the Basic PageRank Update rule.

The random surfer model (basic PageRank)

Consider a user randomly browsing the Web as follows

- First, the user picks a page at random with equal probability.
- Then he follows links for k steps: in each step he follows a random out-going link from his current page (if the page has no outgoing links, then he stays there).

This process defines a random walk on the network.

Claim 1.

The probability of being at page p after k steps of this random walk is precisely the PageRank of p after k applications of the Basic PageRank Update rule.

Thus the limiting probability that the random walk will end up at p is precisely the limiting value of the PageRank of p.

The random surfer model (scaled PageRank)

Consider a user randomly browsing the Web as follows

- First, the user picks a page at random with equal probability
- Then he follows links for k steps: in each step
  - with probability d, he follows a random out-going link from his current page (if the page has no outgoing links, then he stays there)
  - with probability $1-d$, he jumps to a random node anywhere in the network, choosing each node with equal probability

The random surfer model (scaled PageRank)

Consider a user randomly browsing the Web as follows

- First, the user picks a page at random with equal probability
- Then he follows links for k steps: in each step
  - with probability d, he follows a random out-going link from his current page (if the page has no outgoing links, then he stays there)
  - with probability $1-d$, he jumps to a random node anywhere in the network, choosing each node with equal probability

The limiting probability that this random walk will end up at p is the limiting value of the scaled PageRank of p.

Other remarks on PageRank

- PageRank was proposed by Brin and Page in 1999 for ranking webpages; still used by Google (*together with numerous other factors that are kept secret*) to determine popularity of a webpage
- Manipulating PageRank
  - Google bombs: a web page ranks highly in search engine results for unrelated or off-topic search terms
  - Link farming (spamdexing): a group of websites that all hyperlink to each other