Topic-Sensitive PageRank

Motivation

- Different people have ***different interests***
  - Sometimes distinct interests are expressed using the same term in a query
  - (ex) the search query `jaguar`
    - The animal? the automobile? a version of Mac OS?

- If a search engine can deduce that the user is interested in automobiles, for example, then it can do a better job of returning relevant pages to the user

- A private PageRank vector for each user?
  - ***Not*** feasible because there are too many users

Topic-Sensitive PageRank

- Creates one vector for each of some small number of ***topics***
  - Biasing the PageRank to favor pages of that topic
- We then endeavor to classify users according to the degree of their interests in each of the selected topic
  - While we lose some accuracy, we store only a short vector for each user
- Example: Open Directory Project (DMOZ)
  - One useful topic set consisting of 16 top-level categories
  - We could create 16 PageRank vectors, one for each topic
  - We determine which of these topic the user is interested in
    - E.g., by the content of the pages they have recently viewed
  - We then use the PageRank vector for that topic to rank pages

Biased Random Walks

- To create a topic-sensitive PageRank for a certain topic, we arrange that the random surfers are introduced ***only to a random page of that topic***
  - (ex) sports, arts, science, business, health, etc.
- Consequence
  - Random surfers are likely to be at a page of that topic
- Our intuition
  - Pages linked to by pages of a certain topic are themselves likely to be about that topic

Mathematical Formulation

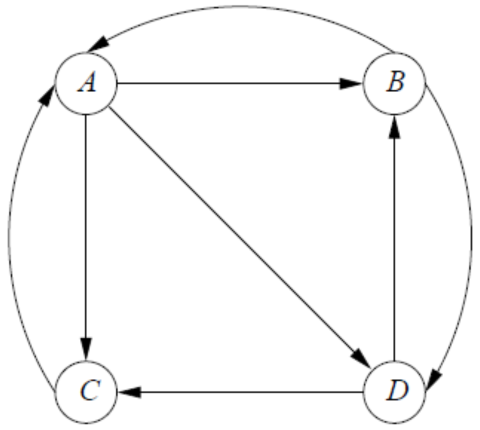
- Suppose S is a set of the pages we have identified as belonging to a certain topic (called the ***teleport set***)
- Let $\mathbf{e}_s$ be a vector that has 1 in the components in S and 0 in other components
- Then the topic-sensitive PageRank for S is the limit of the iteration

$$\mathbf{v}' = \beta M \mathbf{v} + (1 - \beta)\mathbf{e}_s / |S|$$

  – M : the transition matrix of the Web
  – $|S|$: the size of set S
  – Only difference is how we add the new surfers

Example (1/2)

- Consider the following Web graph and its transition matrix multiplied by $\beta = 0.8$



$$\beta M = \begin{bmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix}$$

- Suppose that the teleport set of our topic is $S = \{B, D\}$

- Then, $(1 - \beta)\mathbf{e}_S/|S| = (1 - 0.8)[0\ 1\ 0\ 1]^T/2 = [0\ 1/10\ 0\ 1/10]^T$

Example (2/2)

- Thus, the iteration equation is

$$\mathbf{v}' = \beta M\mathbf{v} + (1-\beta)\mathbf{e}_s/|S| = \begin{bmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ 1/10 \\ 0 \\ 1/10 \end{bmatrix}$$

- The first few iteration of the equation

$$\begin{bmatrix} 0/2 \\ 1/2 \\ 0/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 2/10 \\ 3/10 \\ 2/10 \\ 3/10 \end{bmatrix}, \begin{bmatrix} 42/150 \\ 41/150 \\ 26/150 \\ 41/150 \end{bmatrix}, \begin{bmatrix} 62/250 \\ 71/250 \\ 46/250 \\ 71/250 \end{bmatrix}, \ldots, \begin{bmatrix} 54/210 \\ \boxed{59/210} \\ 38/210 \\ \boxed{59/210} \end{bmatrix}$$

  - We start with the suffers only at the pages in the teleport set, i.e., B and D
  - Notice that because of the concentration of suffers at B and D, these nodes get a higher PageRank than they did in the example in Part 1

Using Topic-Sensitive PageRank (1/2)

1. Decide on the topics for which we shall create specialized PageRank vectors

2. Pick a teleport set for each of these topics, and use that set to compute the topic-sensitive PageRank vector for that topic

3. Find a way of determining the topic or set of topics that are most relevant for a particular search query

4. Use the PageRank vectors for that topic or topics in the ordering of the responses to the search query

Using Topic-Sensitive PageRank (2/2)

- About the first step
  - We can use the top-level topics of the Open Directory
  - Other approaches are possible, but there is probably a need for human classification of at least some pages

- About the third step
  - Probability the trickiest, and several methods have been proposed
    ① Allow the user to select a topic from a menu
    ② Infer the topic(s) by the words that appear in the Web pages recently searched by the user, or recent queries issued by the user → ***discussed soon***
    ③ Infer the topic(s) by information about the user (e.g., their bookmarks or their stated interests on Facebook)

Inferring Topics from Words (1/2)

- Classifying documents by topic
  - A subject that has been studied for decades
  - Topics are characterized by words that appear surprisingly often in documents on that topic
    - (ex) `fullback` and `measles` will appear far more than average in sport pages and medicine pages, respectively

- Finding characteristic words of a topic
  - Examine the entire Web (or a large sample of the Web) to get the background frequency of each word
  - Go to a large sample of pages known to be about a certain topic
  - Example the frequencies of words in the sample
  - Identify the words that appear significantly more frequently in the sample than in the background

Inferring Topics from Words (2/2)

- Classifying a page by topic
  - Let S_1, S_2, …, S_k be the sets of the characteristic words of each topic
  - Compute the Jaccard similarity between a given page and each of the S_i's
  - Classify the page as that topic with the highest Jaccard similarity

- Classifying a user by topic
  - Classify the pages the users has most recently retrieved (or the pages the user currently has bookmarked)
  - We could say the user is interested in the topic into which the largest number of these pages fall
  - Or we could ***blend*** the topic-sensitive PageRank vectors in proportion to the fraction of these pages that fall into each topic
    - i.e., construct a single PageRank vector that reflects the user's current blend of interests

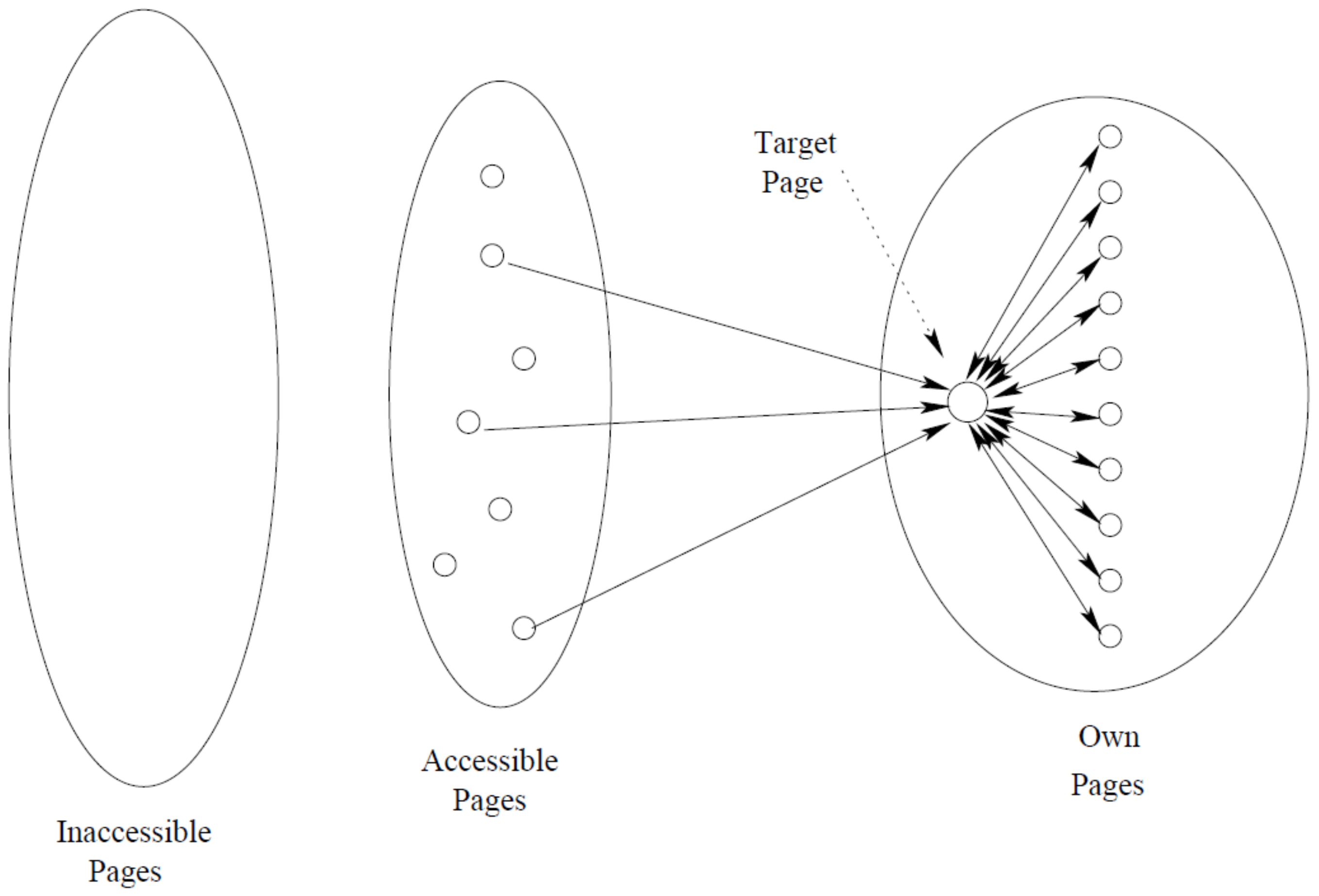
Link Spam

Link Spam

- It became apparent that PageRank and other techniques used by Google made ***term spam ineffective***
- As a result, spammers turned to methods designed to fool the PageRank algorithm into overevaluating certain pages
- ***Link spam***
  - The techniques for artificially increasing the PageRank of a page
- Methods for preventing link spam
  - TrustRank
  - Measurement of spam mass

Architecture of a Spam Farm (1/2)

- Spam farm
  - A collection of pages whose purpose is to increase the PageRank of a certain page or pages



Architecture of a Spam Farm (2/2)

- Inaccessible pages
  - The pages that the spammer cannot affect (most of the Web)
- Accessible pages
  - Those page that, while they are not controlled by the spammer, can be affected by the spammer
  - (ex) blogs or newspapers that invite others to post their comments
- Own pages
  - The pages that the spammer owns and controls
  - **Target page t**: The page at which the spammer attempts to place as much PageRank as possible
  - **Supporting pages**: accumulate the portion of the PageRank and prevent the PageRank of t from being lost
  - t has a link to every supporting page, every supporting page links only to t

Analysis of a Spam Farm (1/4)

- Notations
  - β : a taxation parameter (typically around 0.85)
    - The fraction of a page's PageRank that gets distributed to its successors at the next round
  - n : the number pages on the Web
  - t : the target page
  - m : the number of supporting pages
  - x : the amount of PageRank contributed by the accessible pages
    - That is, x is the sum, over all accessible pages p with a link to t, of the PageRank of p times β, divided by the number of successors of p
  - y : the PageRank of t

Analysis of a Spam Farm (2/4)

- First, the PageRank of each supporting page is

$$\beta y/m + (1-\beta)/n$$

  - The first term
    - The contribution from t
    - The PageRank y of t is taxed, so only by is distributed to t's successors, which are the m supporting pages
  - The second term
    - The supporting page's share of the fraction $1-\beta$ of the PageRank that is divided equally among all pages on the Web

Analysis of a Spam Farm (3/4)

- The PageRank y of target page t comes from three sources:
  1. Contribution x from outside, as we have assumed
  2. β times the PageRank of every supporting page, which is

$$\beta\big(\beta y/m + (1-\beta)/n\big)$$

  3. $(1-\beta)/n$, the share of the fraction $1-\beta$ of the PageRank that belongs to t
     - This amount is negligible and will be dropped to simplify the analysis

- Therefore,

$$y = x + \beta m\Big(\frac{\beta y}{m} + \frac{1-\beta}{n}\Big) = x + \beta^2 y + \beta(1-\beta)\frac{m}{n}$$

Analysis of a Spam Farm (4/4)

- We may solve the previous equation for y, yielding

$$y = \frac{x}{1-\beta^2} + c\frac{m}{n}$$

  - Where $c = \beta(1-\beta)/(1-\beta^2) = \beta/(1+\beta)$

- Example
  - If $\beta = 0.85$, then $1/(1-\beta^2) = 3.6$ and $c = \beta/(1+\beta) = 0.46$
  - That is, the structure has amplified the external PageRank contribution by 360%
  - Also obtained an amount of PageRank that is 46% of the fraction of the Web, m/n, that is in the spam farm

Combating Link Spam

- It has become essential for search engines to detect and eliminate ***link spam***
  - Just as it was necessary to eliminate ***term spam***
- This war between the spammers and the search engines will likely go on for a long time
- There are two approaches to link spam
  - **TrustRank**
    - A variation of topic-sensitive PageRank designed to lower the score of spam pages
  - **Spam mass**
    - A calculation that identifies the pages that are likely to be spam and allows the search engine to eliminate those pages or to lower their PageRank strongly

TrustRank

- Topic-sensitive PageRank, where the "***topic***" is a set of pages believed to be trustworthy (not spam)

- Theory
  – While a spam page might easily be made to link to a trustworthy page, it is unlikely that a trustworthy page would link to a spam page

- Borderline area
  – A site with blogs or other opportunities for spammers to create links
  – These pages ***cannot*** be considered trustworthy, even if their own content is highly reliable

Implementation of TrustRank

- Two approaches to develop a teleport set of ***trustworthy*** pages

① Let ***humans*** examine a set of pages and decide which of them are trustworthy
  - (ex) Pick the pages of highest PageRank to examine

② Pick a ***domain*** whose membership is controlled, on the assumption that it is hard for a spammer to get their pages into these domains
  - (ex) .edu, .mil, or .gov domain, since they are unlikely to be spam farms

✓ PageRank is really thought as a form of TrustRank

Spam Mass (1/2)

- Idea
  - For each page, measure the fraction of its PageRank that comes from ***spam***
- How?
  - Compute both the ordinary PageRank and the TrustRank based on some teleport set of trustworthy pages
  - Suppose page p has PageRank r and TrustRank t
  - Then the spam mass of p is $(r - t)/r$
- Meaning
  - **A negative small positive value**: p is probably ***not*** a spam page
  - **A value close to 1**: p is probably spam

Spam Mass (2/2)

- We can eliminate pages with a high spam mass from the index of Web pages used by a search engine
  - Without having to identify particular structures that spam farmers use

- Example
  - Consider the following information for nodes A, B, C, and D

Node	PageRank	TrustRank	Spam Mass
A	$3/9$	$54/210$	0.229
B	$2/9$	$59/210$	-0.264
C	$2/9$	$38/210$	0.186
D	$2/9$	$59/210$	-0.264

  - Conclusion
    - B and D: a ***negative*** spam mass → they are not spam
    - A and C: a ***small positive*** spam mass → they also are probably not spam

Hubs and Authorities

Hubs and Authorities

- Proposed shortly after PageRank was first implemented

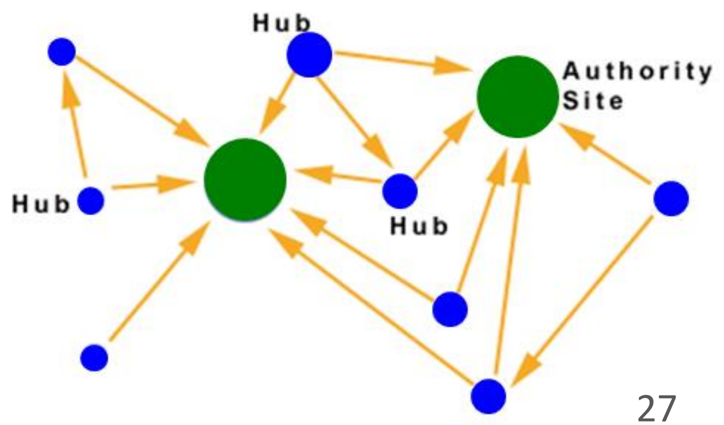
- The algorithm resembles the computation of PageRank
  - The iterative computation of a fixed-point involving repeated matrix-vector multiplication

- However, there are significant differences between the two ideas
  - Neither can substitute for the other

- ***HITS (hyperlink-induced topic search) algorithm***
  - A hubs-and-authorities algorithm to rank the responses to a search query
  - Believed to be used by the Ask search engine

The Intuition Behind HITS

- HITS assumes that there are ***two*** types of important pages
  - ***Authorities***
    - Those pages providing information about a topic
    - Examples
      - Newspaper homepages
      - Course homepages
      - Auto-manufacturers homepages
  - ***Hubs***
    - Those pages telling you where to go to find out about that topic
    - Examples
      - List of newspapers
      - Course bulletin
      - List of auto-manufactures



Comparison with PageRank

- PageRank
  - *"a page is important if important pages link to it"*
  - A recursive definition of importance

- HITS
  - *"a page is a good hub if it links to good authorities, and a page is a good authority if it is linked to by good hubs"*
  - A mutually recursive definition of two concepts

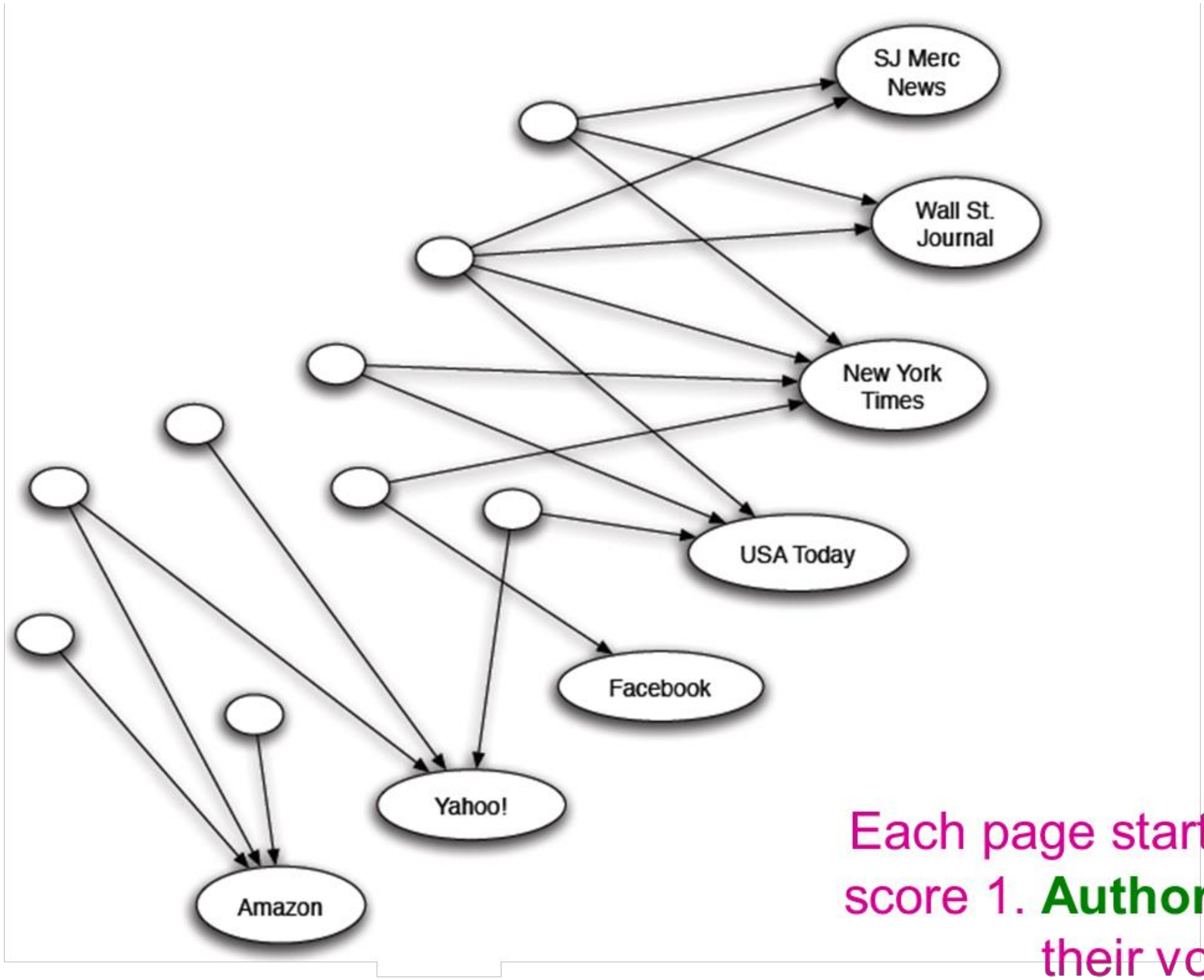
Formalizing Hubbiness and Authority

- We assign two scores to each Web page
  - Hubbiness: the degree to which it is a good bug
  - Authority: the degree to which it is a good authority

- We represent these scores by vectors $\mathbf{h}$ and $\mathbf{a}$
  - The ith component of $\mathbf{h}$: the hubbiness of the ith page
  - The ith component of $\mathbf{a}$: the authority of the ith page

- Computing $\mathbf{h}$ and $\mathbf{a}$
  - Add the authority of successors to estimate hubbiness
  - Add the hubbiness of predecessors to estimate authority
  - Scale the values of $\mathbf{h}$ and $\mathbf{a}$ so that the largest component is 1
    - Otherwise, the hubbiness and authority values would grow beyond bounds

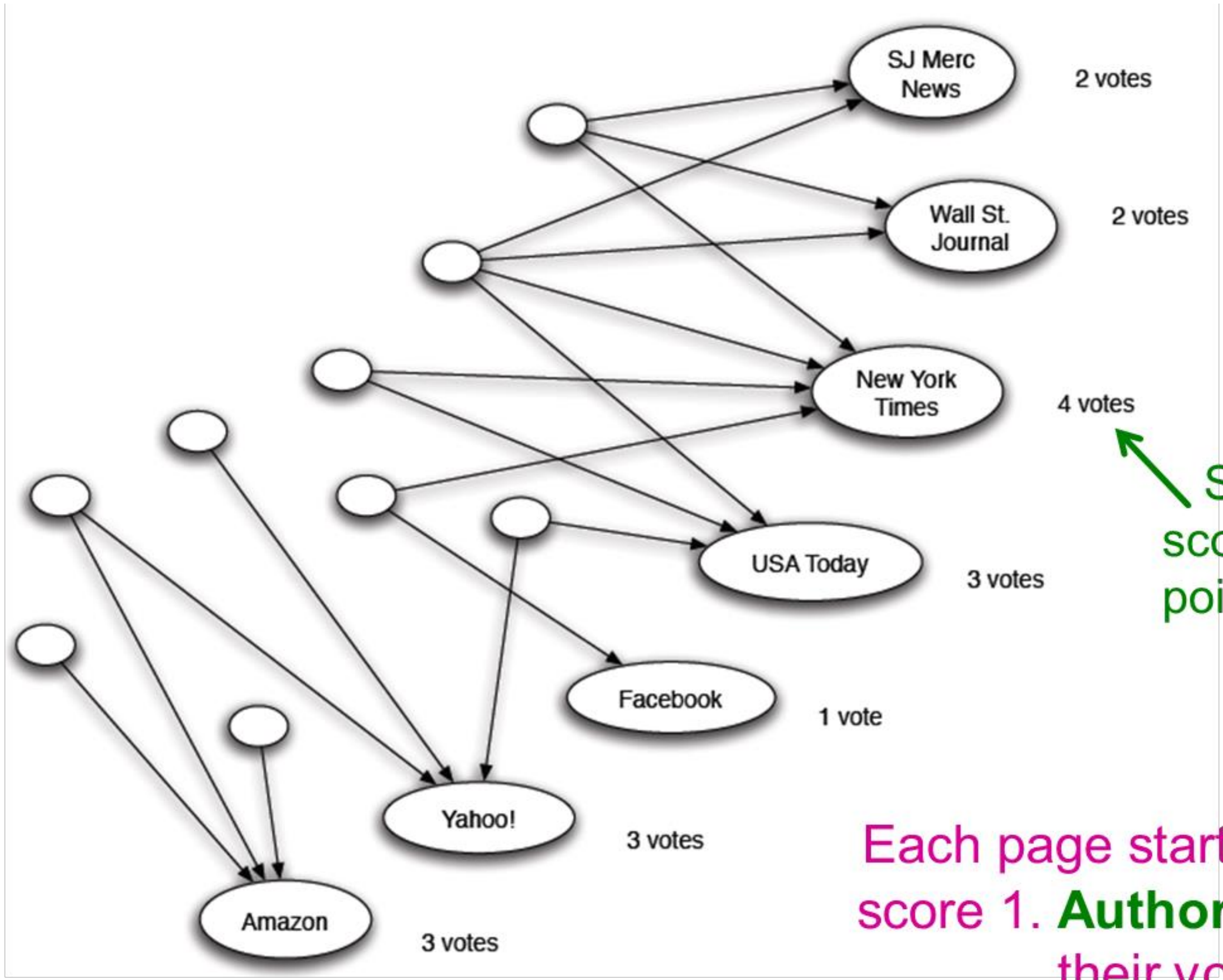
Counting in-links: Authority



Each page starts with **hub** score 1. **Authorities** collect their votes

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Counting in-links: Authority



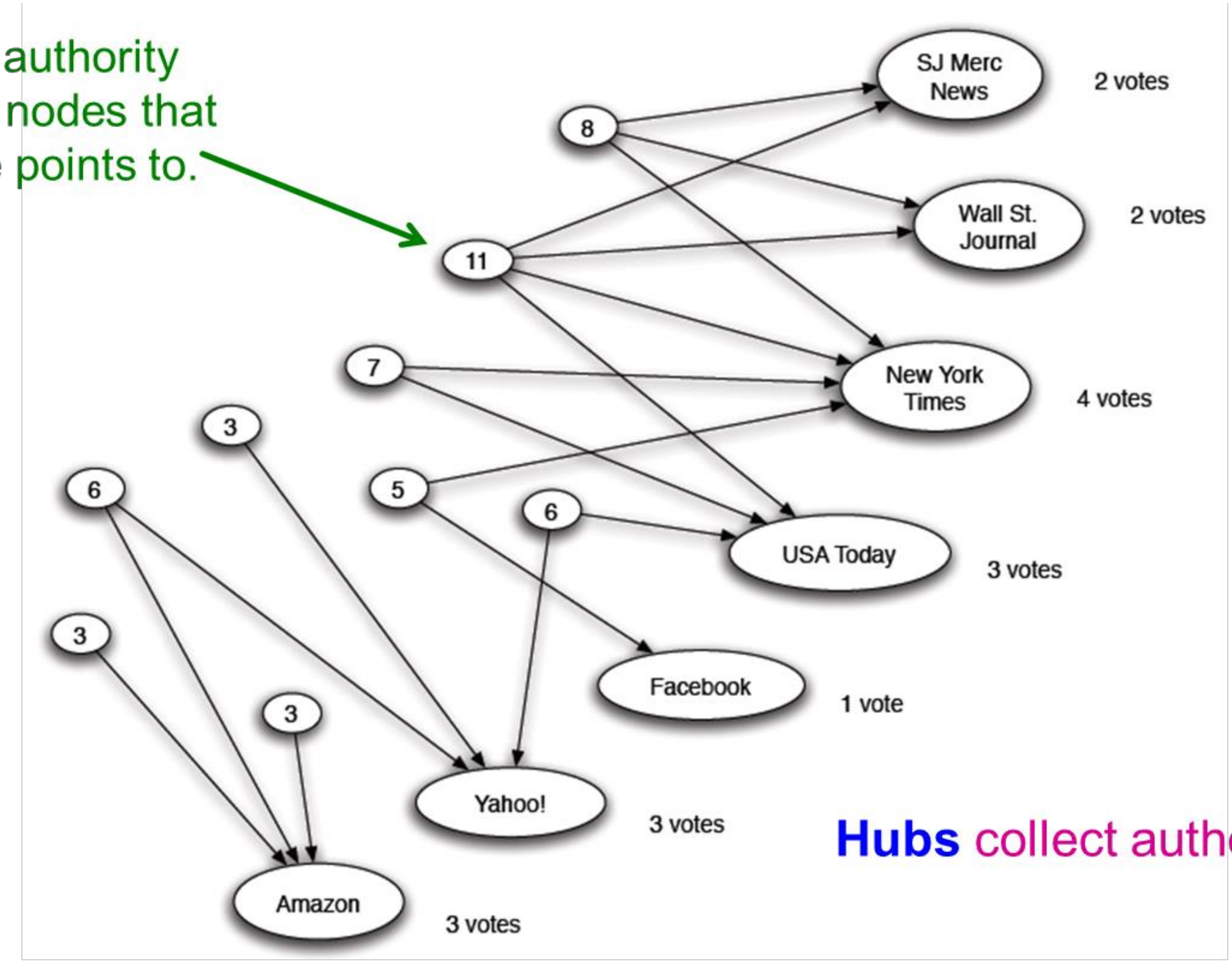
Sum of **hub** scores of nodes pointing to NYT.

Each page starts with **hub** score 1. **Authorities** collect their votes

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Expert Quality: Hub

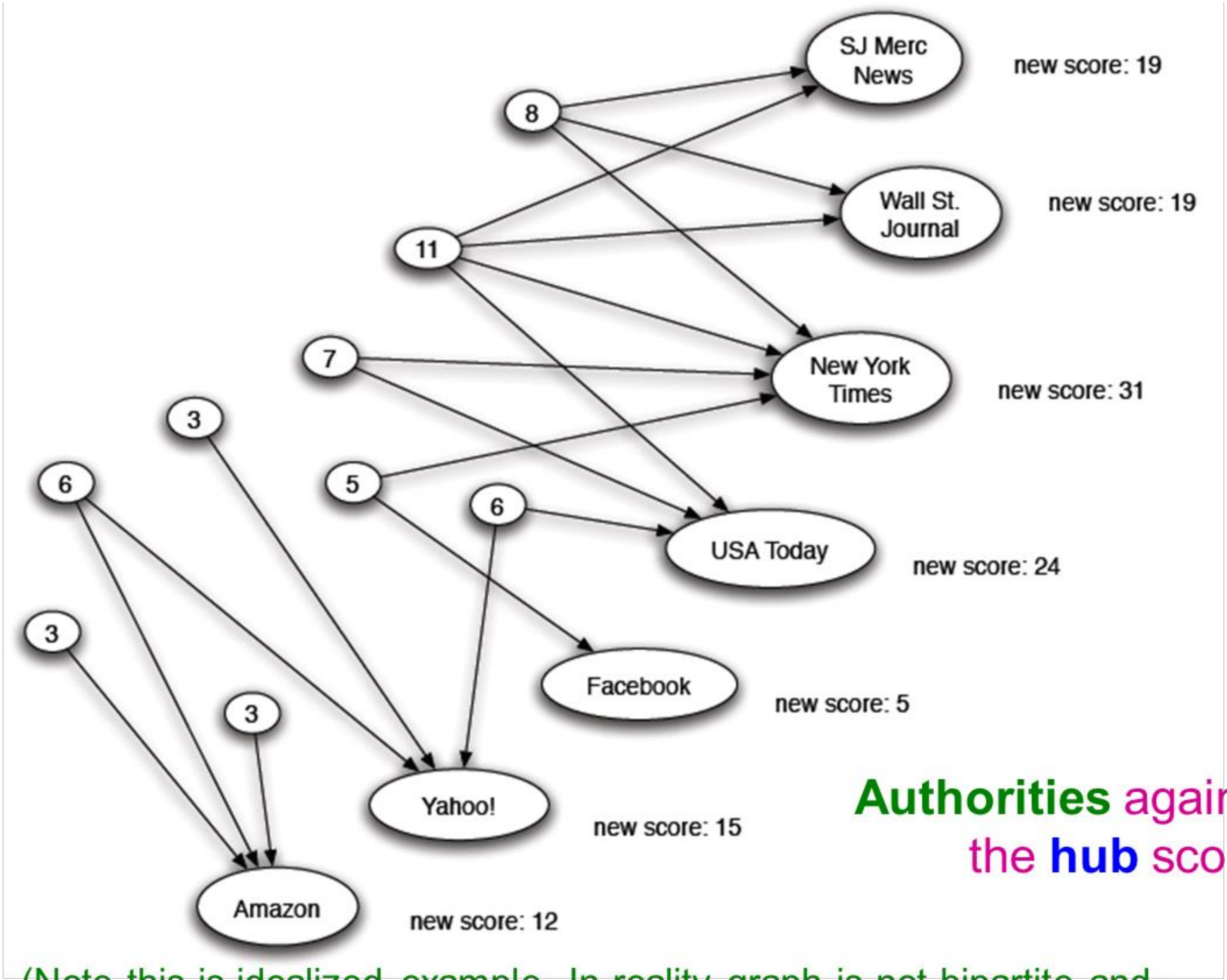
Sum of authority scores of nodes that the node points to.



Hubs collect authority scores

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Reweighting

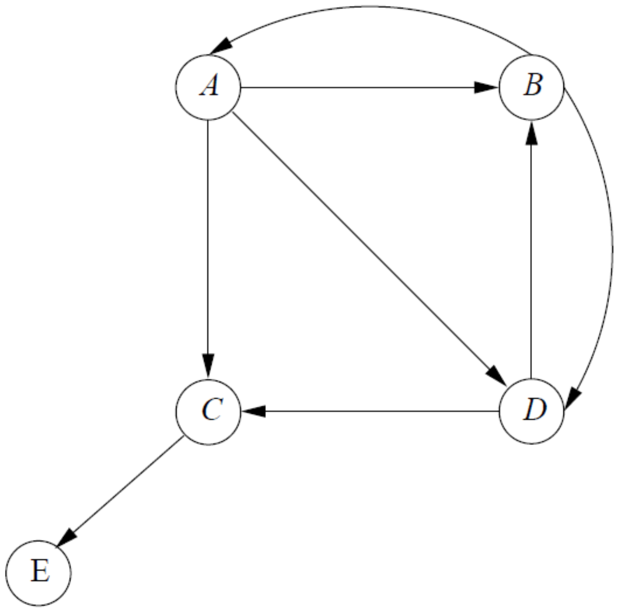


Authorities again collect the **hub** scores

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Iterative Computation of $\mathbf{h}$ and $\mathbf{a}$ (1/3)

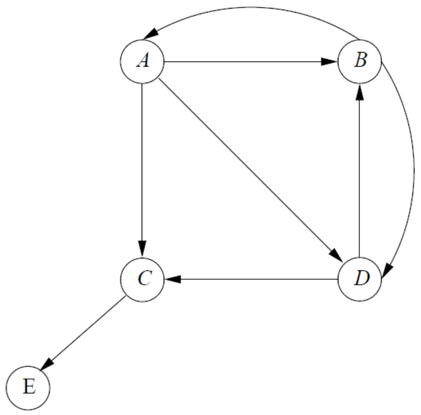
- L : the link matrix of the Web
  - If we have n pages, then L is an $n \times n$ matrix
  - $L_{ij} = 1$ if there is a link from page i to page j, and $L_{ij} = 0$ if not
- L^T : the transpose of L
  - $L^T_{ij} = 1$ is there is a link from page j to page i, and $L^T_{ij} = 0$ otherwise



$$L = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad L^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Iterative Computation of $\mathbf{h}$ and $\mathbf{a}$ (2/3)

- The hubbiness of a page
  - Proportional to the sum of the authority of its successors
  - Expressed by $\mathbf{h} = \lambda L \mathbf{a}$
    - λ : an unknown constant representing the scaling factor needed
- The authority of a page
  - Proportional to the sum of the hubbinesses of its predecessors
  - Expressed by $\mathbf{a} = \mu L^T \mathbf{h}$
    - μ : another scaling constant



A's successors

$$L = \begin{bmatrix} 0 & \boxed{\begin{matrix} 1 & 1 & 1 \end{matrix}} & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A's predecessors

$$L^{\mathrm{T}} = \begin{bmatrix} 0 & \boxed{1} & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Iterative Computation of $\mathbf{h}$ and $\mathbf{a}$ (3/3)

- We can compute $\mathbf{h}$ and $\mathbf{a}$ independently, by substituting one equation in the other, as:

$$\mathbf{h} = \lambda\mu L L^{\mathrm{T}}\mathbf{h}$$

$$\mathbf{a} = \lambda\mu L^{\mathrm{T}} L\mathbf{a}$$

- However, LL^T and L^TL are not sparse as L and L^T

- Thus, we usually compute $\mathbf{h}$ and $\mathbf{a}$ in a true mutual recursion as:
  ① Compute $\mathbf{a} = L^T\mathbf{h}$ and then scale so the largest component is 1
  ② Next, compute $\mathbf{h} = L\mathbf{a}$ and scale again
  ③ Repeat steps ① and ② until the changes to h and a are sufficiently small

Example (1/2)

- The first two iterations of HITS on the previous example

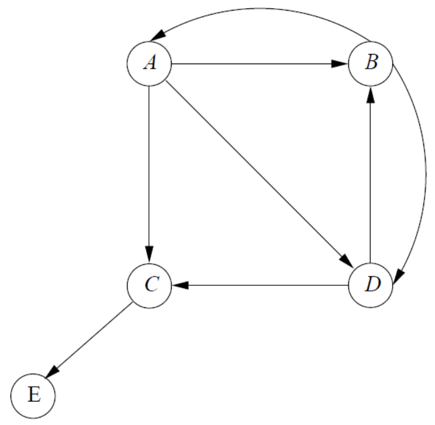
$$
\begin{array}{ccccc}
\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} &
\begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} &
\begin{bmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{bmatrix} &
\begin{bmatrix} 3 \\ 3/2 \\ 1/2 \\ 2 \\ 0 \end{bmatrix} &
\begin{bmatrix} 1 \\ 1/2 \\ 1/6 \\ 2/3 \\ 0 \end{bmatrix} \\
\mathbf{h} & L^{\mathrm{T}}\mathbf{h} & \mathbf{a} & L\mathbf{a} & \mathbf{h} \\
\text{(initial)} & & \text{(scaled)} & & \text{(scaled)}
\end{array}
$$

$$
\begin{array}{cccc}
\begin{bmatrix} 1/2 \\ 5/3 \\ 5/3 \\ 3/2 \\ 1/6 \end{bmatrix} &
\begin{bmatrix} 3/10 \\ 1 \\ 1 \\ 9/10 \\ 1/10 \end{bmatrix} &
\begin{bmatrix} 29/10 \\ 6/5 \\ 1/10 \\ 2 \\ 0 \end{bmatrix} &
\begin{bmatrix} 1 \\ 12/29 \\ 1/29 \\ 20/29 \\ 0 \end{bmatrix} \\
L^{\mathrm{T}}\mathbf{h} & \mathbf{a} & L\mathbf{a} & \mathbf{h} \\
 & \text{(scaled)} & & \text{(scaled)}
\end{array}
\Rightarrow
\mathbf{h} = \begin{bmatrix} 1 \\ 0.3583 \\ 0 \\ 0.7165 \\ 0 \end{bmatrix}
\quad
\mathbf{a} = \begin{bmatrix} 0.2087 \\ 1 \\ 1 \\ 0.7913 \\ 0 \end{bmatrix}
$$

The limits

Example (2/2)

- Interpretation of the result



$$\mathbf{h} = \begin{bmatrix} 1 \\ 0.3583 \\ 0 \\ 0.7165 \\ 0 \end{bmatrix} \qquad \mathbf{a} = \begin{bmatrix} 0.2087 \\ 1 \\ 1 \\ 0.7913 \\ 0 \end{bmatrix}$$

  - The hubbiness of E is 0
    - Since it leads nowhere
  - The hubbiness of C is also 0
    - Since it depends only on the authority of E and vice versa
  - A is the greatest hub
    - Since it links to the three biggest authorities, B, C, and D
  - B and C are the greatest authorities
    - Since they are linked to by the two biggest hubs, A and D