Tài liệu giảng dạy môn Khai thác dữ liệu Web

GIẢI THUẬT XẾP HẠNG TRANG WEB

TS. Nguyễn Ngọc Thảo – ThS. Lê Ngọc Thành

Bộ môn Khoa học Máy tính, FIT HCMUS, VNUHCM

Thành phố Hồ Chí Minh, 02/2019

Nội dung bài giảng

- Giải thuật PageRank
- Giải thuật HITS
- Khám phá cộng đồng

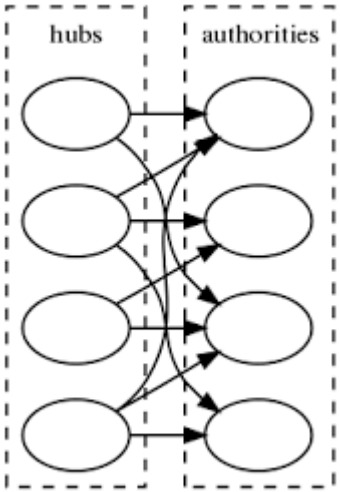
Giải thuật xếp hạng tìm kiếm

- Giải thuật PageRank và HITS được đề xuất vào năm 1998 → sự kiện quan trọng trong Phân tích liên kết Web và Tìm kiếm Web

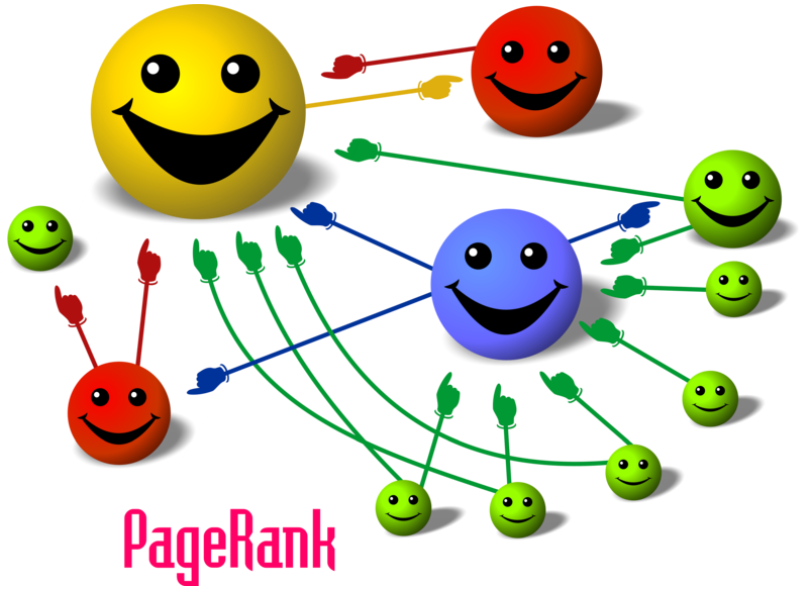
Google

PageRank

HITS



- **PageRank:** mô hình phân tích liên kết ưu việt, được sử dụng phổ biến trong Tìm kiếm Web
  - Đánh giá trang Web mà không phụ thuộc truy vấn, chống spam tốt
  - Một phần cũng nhờ vào thành công kinh doanh của Google



Giải thuật PageRank

Giải thuật PageRank

- S. Brin and L. Page, “The anatomy of a large-scale hypertextual Web search engine,” Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 107–117, 1998

Giải thuật PageRank

- Bản chất dân chủ của Web: Cấu trúc liên kết được xem là một chỉ thị cho chất lượng của mỗi trang đơn lẻ.
- Một siêu liên kết từ trang x đến trang y được hiểu như một bình chọn bởi trang x cho trang y.
  - Bình chọn từ những trang quan trọng có "trọng lượng" nhiều hơn và giúp cho trang nhận bình chọn trở nên "quan trọng" hơn.
  - Đây chính là ý tưởng của rank prestige trong phân tích mạng xã hội
- Xếp hạng tĩnh: Giá trị PageRank được tính toán ngoại tuyến (offline) cho mỗi trang và độc lập với các câu truy vấn.

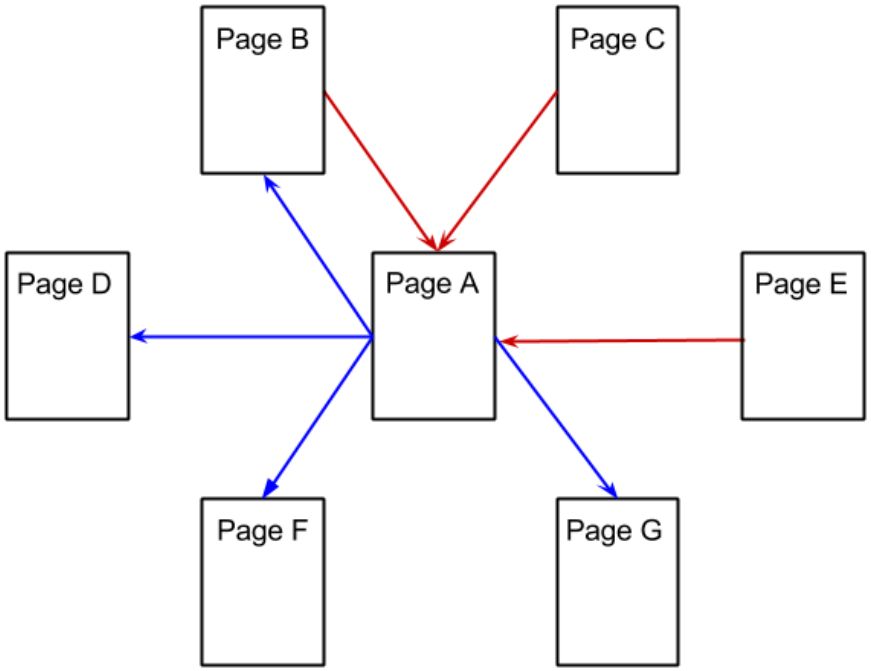
Khái niệm quan trọng

- Liên kết trong của trang i

  Siêu liên kết trỏ đến trang i từ những trang khác.

- Liên kết ngoài của trang i

  Siêu liên kết trỏ từ trang i đến những trang khác



- Siêu liên kết trong cùng một site thường không được xét đến trong ngữ cảnh này.

Giải thuật PageRank

- Trang i nhận càng nhiều liên kết trong thì trang i có càng nhiều uy tín.
  - Siêu liên kết từ một trang trỏ đến trang khác là sự truyền đạt ngầm độ uy tín từ trang nguồn đến trang đích.
- Một trang có điểm uy tín cao hơn trỏ đến trang i sẽ quan trọng hơn trang có điểm uy tín thấp hơn cũng trỏ đến i.
  - Các trang trỏ đến trang i cũng có điểm uy tín của riêng chúng.
- Nói cách khác, một trang được gọi là quan trọng nếu nó được trỏ đến bởi các trang quan trọng khác.

Điểm PageRank

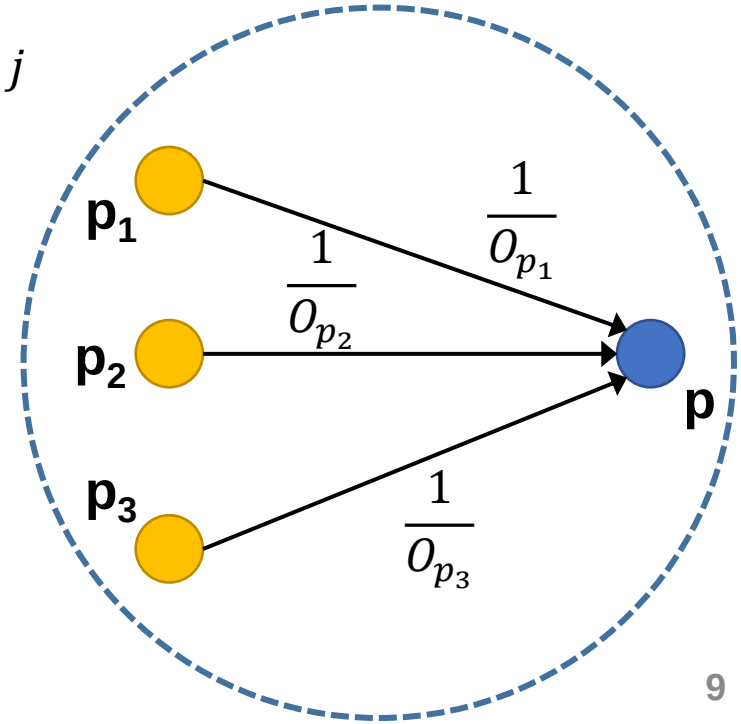
- Gọi Web là đồ thị có hướng $G = (V, E)$ gồm n trang.
- **Điểm PageRank (PR)** của trang i được định nghĩa là

$$\boldsymbol{P(i) = \sum_{(j,i)\in E} \frac{P(j)}{O_j}}$$

  - Trong đó O_j là số liên kết ngoài của trang j

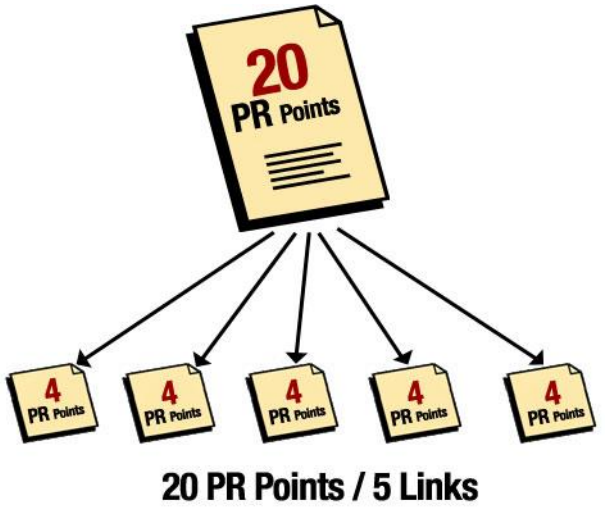
- Ví dụ,

$$P(p) = \frac{P(p_1)}{O_{p_1}} + \frac{P(p_2)}{O_{p_2}} + \frac{P(p_3)}{O_{p_3}}$$

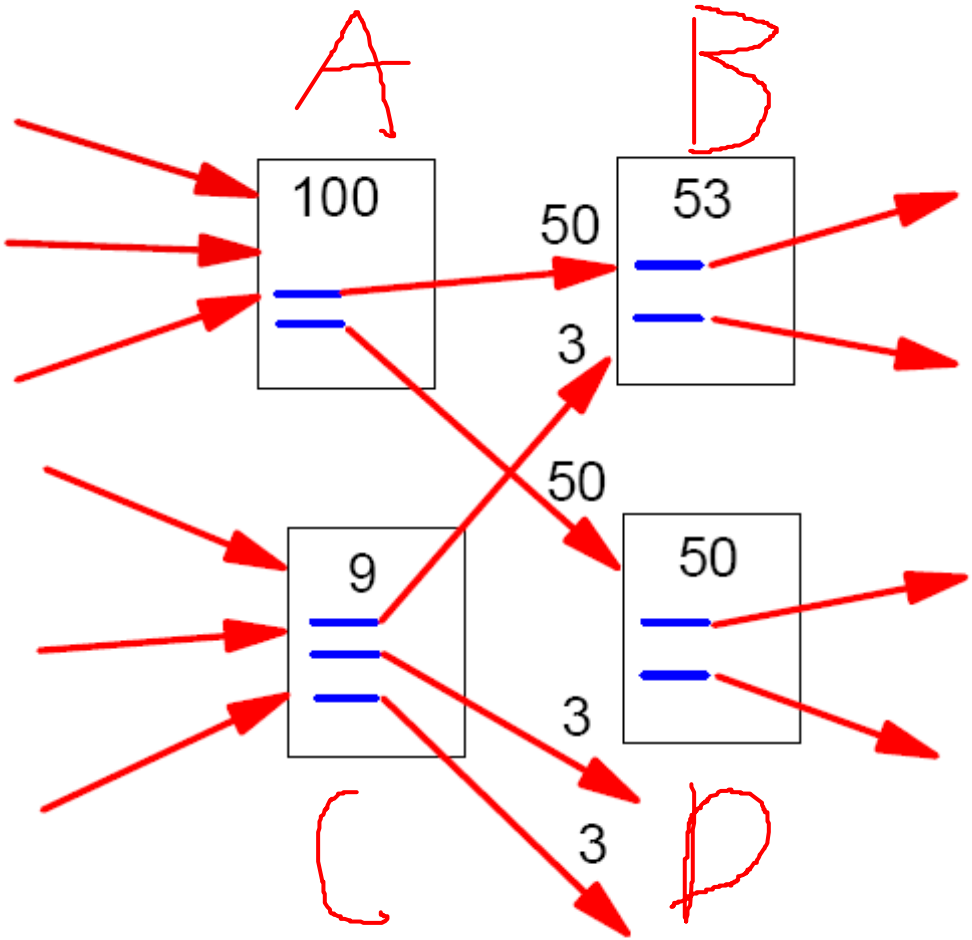


Điểm PageRank

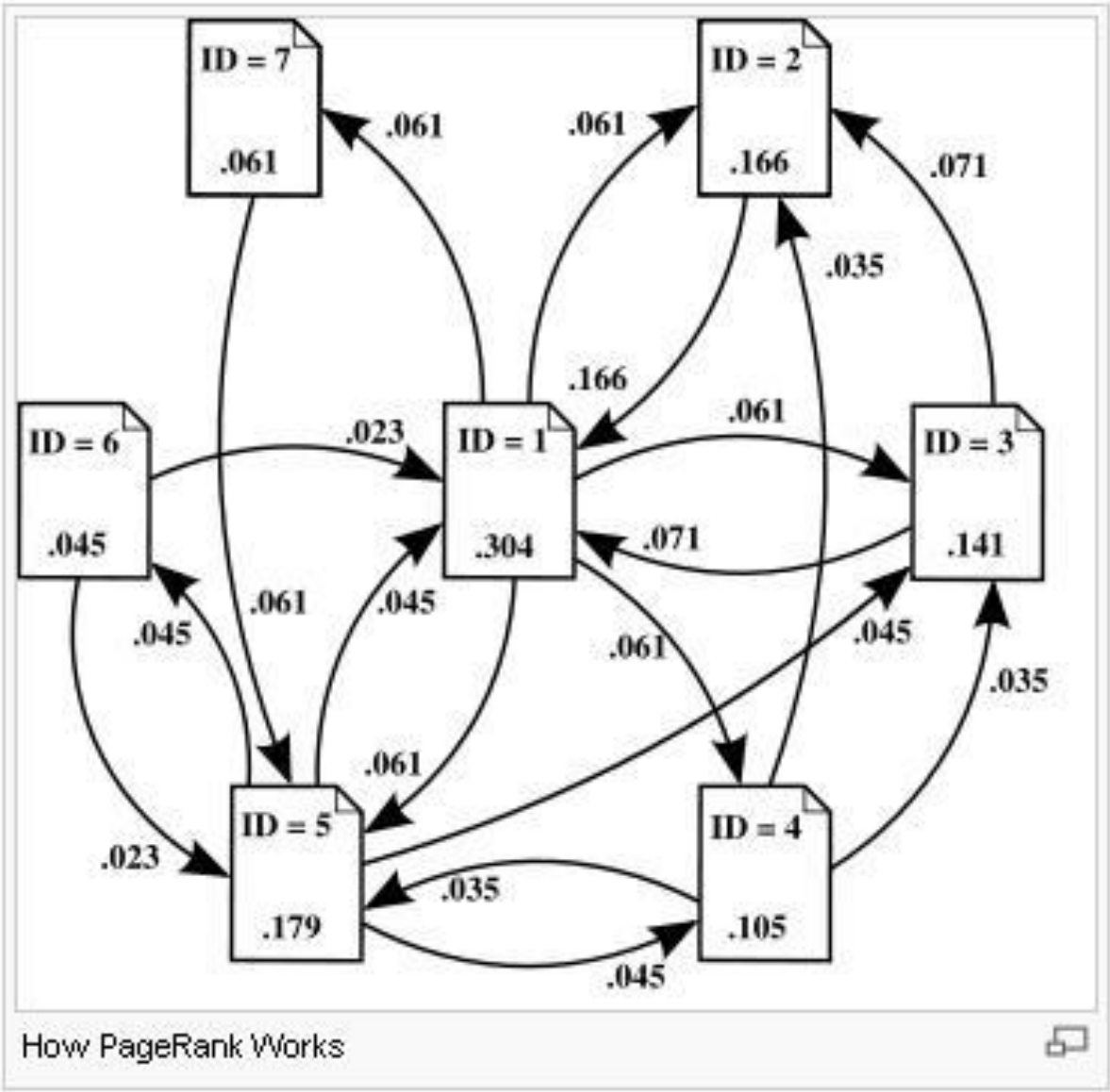
- Điểm PR của một trang i là tổng điểm PR của mọi trang có trỏ đến i và được chia sẻ đều cho mọi trang mà i trỏ đến.
- Trong khi đó, giá trị rank prestige không có tính chia sẻ.

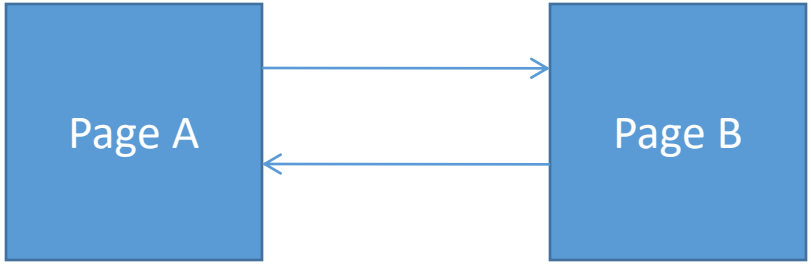


PageRank



Điểm PageRank: Ví dụ





PR(A) = PR(B)
PR(B) = PR(A)

Để tính PageRank cho trang A cần biết PageRank cho trang B, nhưng để tính PageRank cho trang B, cần tính PageRank cho trang A.

➔ Dùng vòng lặp

100 vòng lặp là đủ để có được một kết quả tương đối tốt cho toàn Bộ web

Điểm PageRank

- Gọi $\boldsymbol{P}$ là một vector n-chiều chứa các giá trị PR.

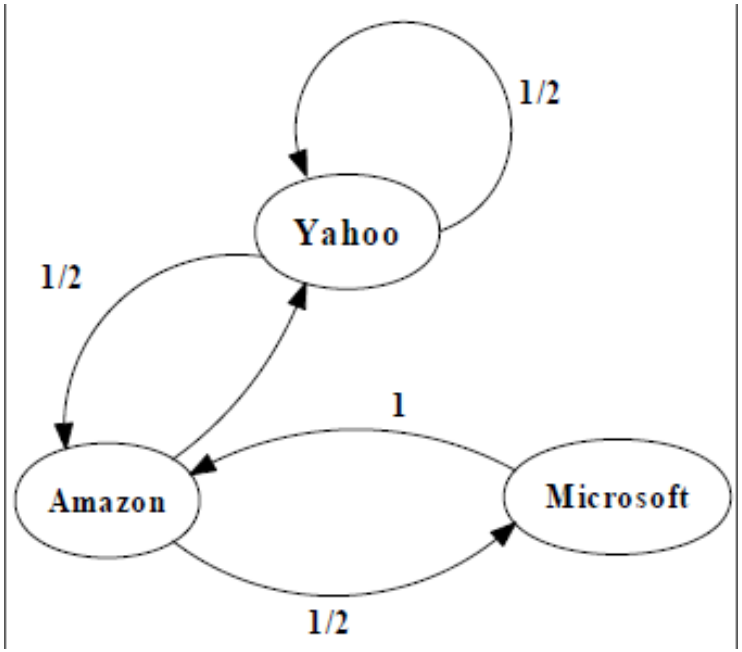
$$\boldsymbol{P} = (P(1), P(2), \ldots, P(n))^T$$

- Gọi $\boldsymbol{A}$ là ma trận kề của đồ thị G sao cho

$$A_{ij} = \begin{cases} \dfrac{1}{O_j} & nếu\ (i,j) \in E \\ 0 & ngược\ lại \end{cases}$$

- Hệ n phương trình có thể được biểu diễn thành $\boldsymbol{P = A^T P}$
  - Lời giải cho P là eigenvector với eigenvalue bằng 1.
  - Nếu một số điều kiện được thỏa, P là principal eigenvector và 1 là eigenvalue lớn nhất.

An example of Simplified PageRank



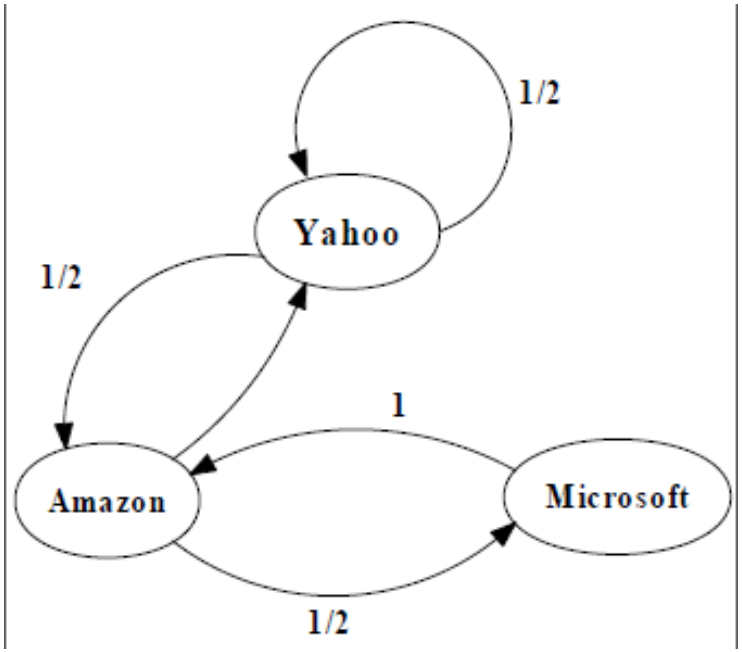
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

PageRank Calculation: first iteration

An example of Simplified PageRank



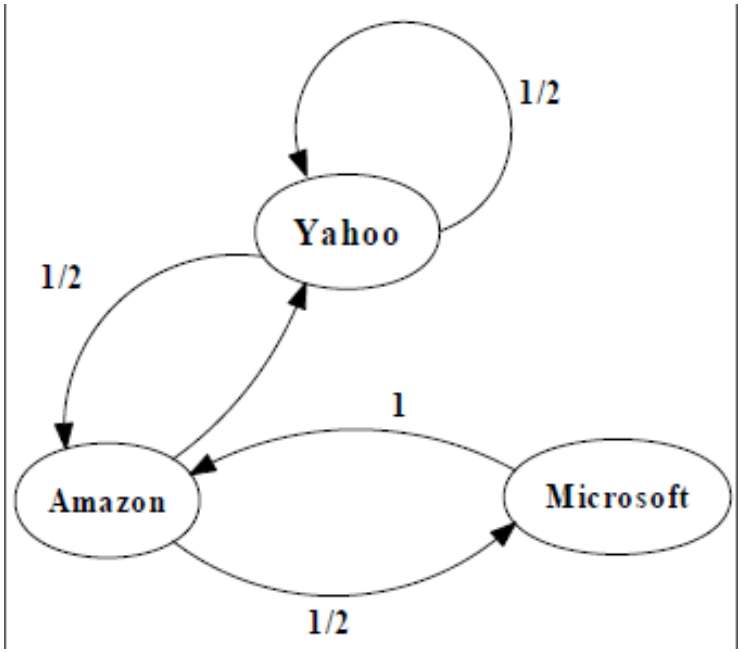
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

PageRank Calculation: second iteration

An example of Simplified PageRank



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

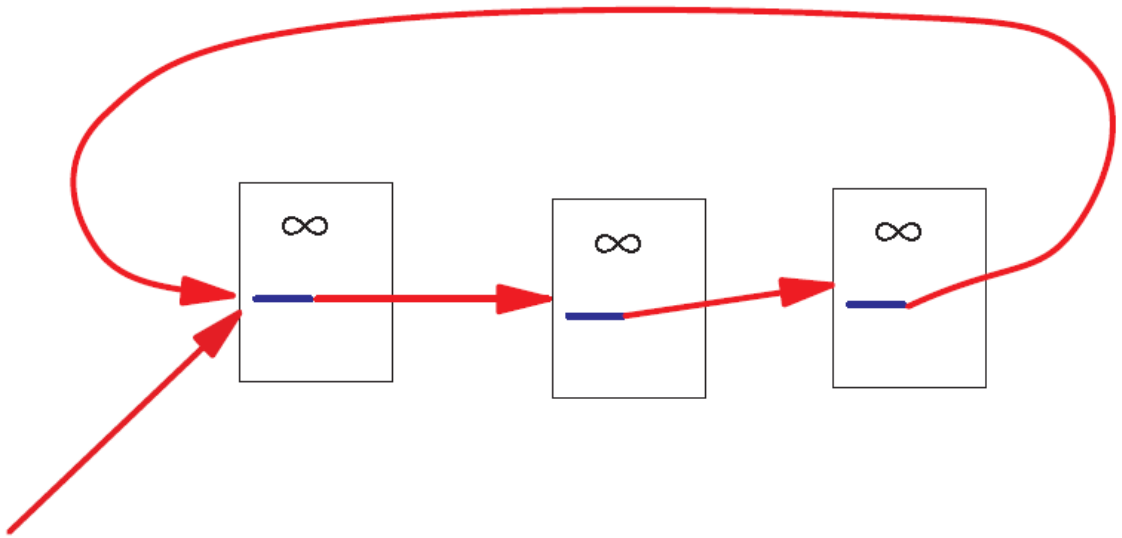
$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

Convergence after some iterations

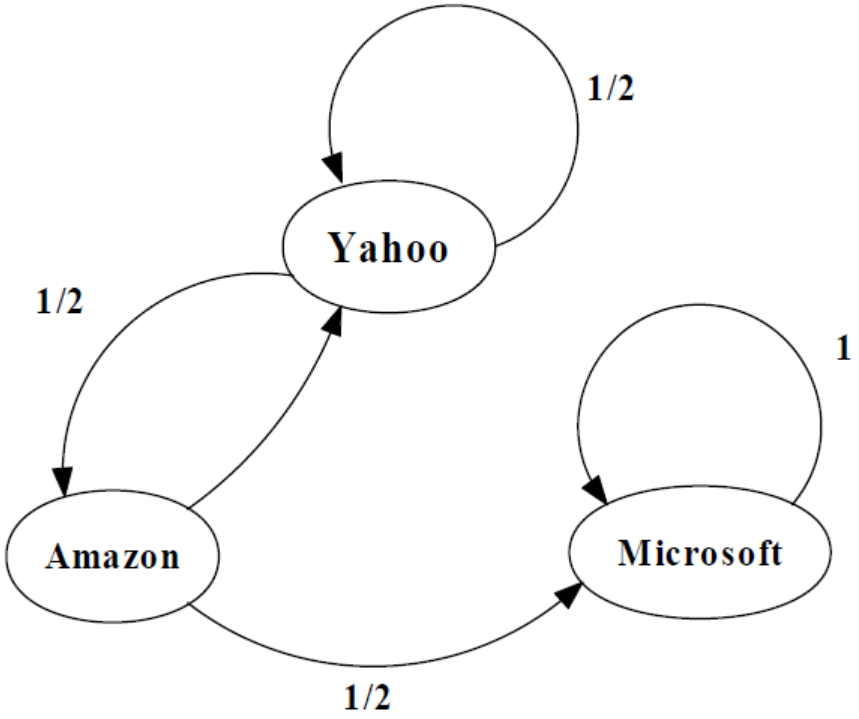
A Problem with Simplified PageRank

A loop:



During each iteration, the loop accumulates rank but never distributes rank to other pages!

An example of the Problem

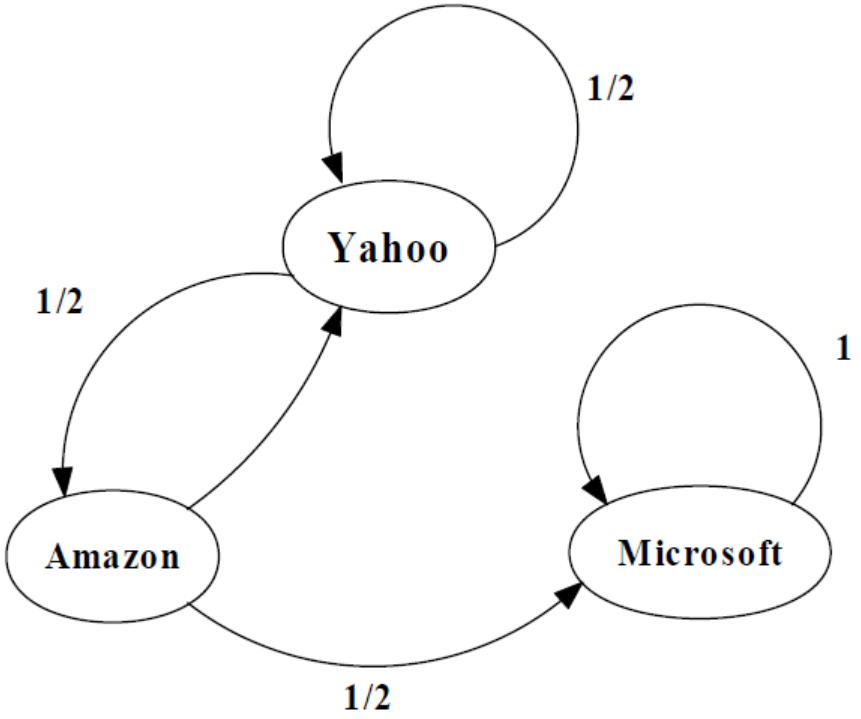


$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

An example of the Problem

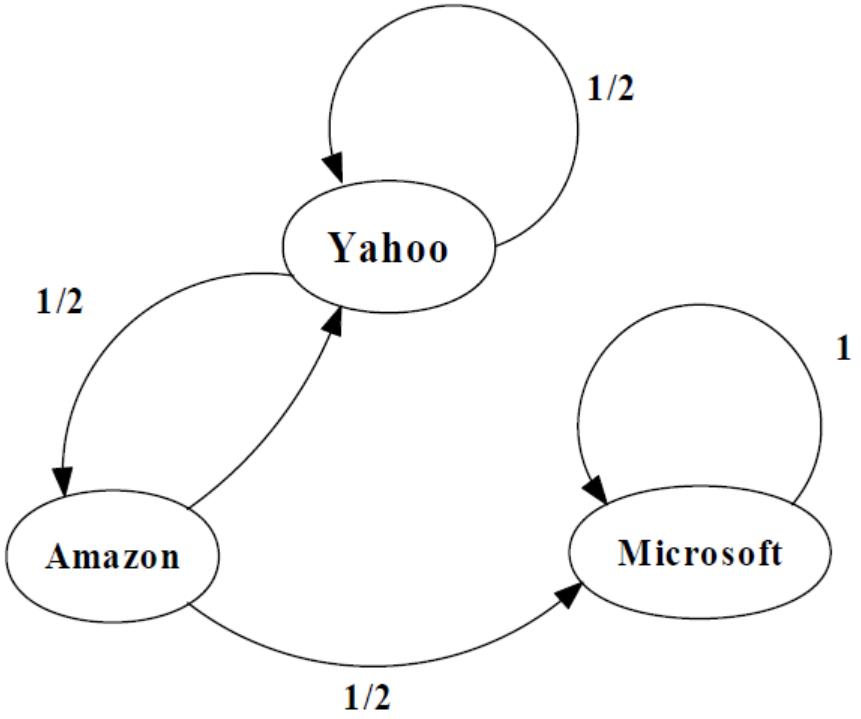


$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix}$$

An example of the Problem



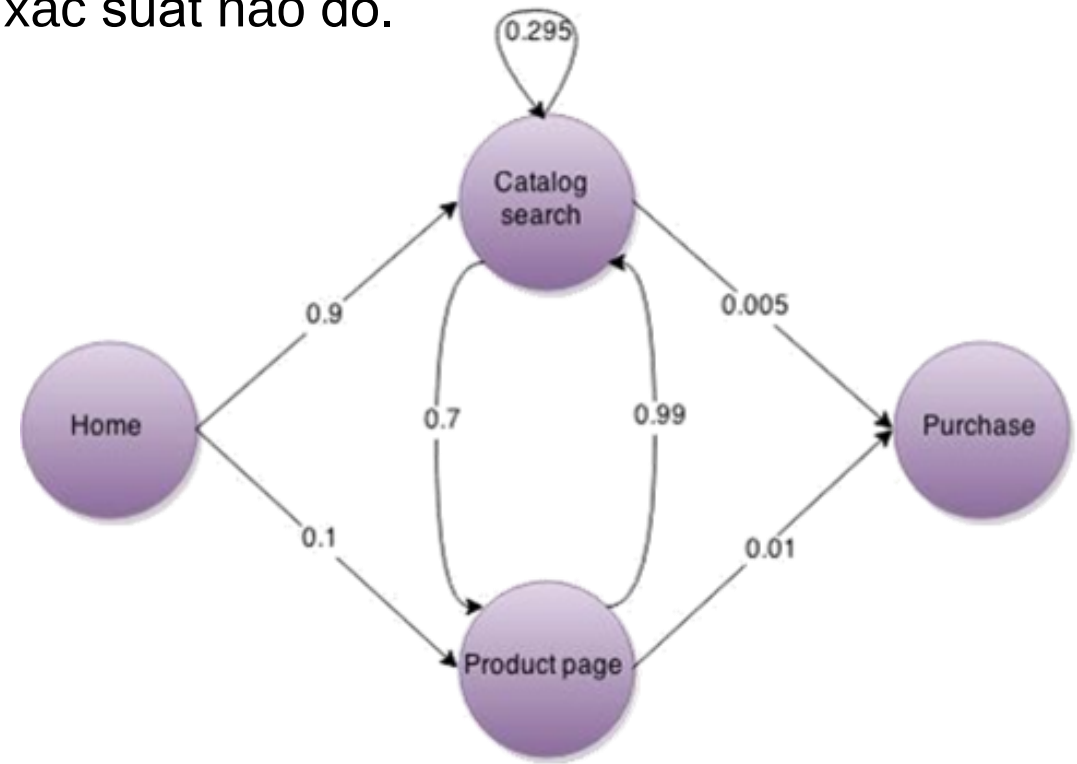
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \ldots \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

PageRank và Chuỗi Markov

- Chuỗi Markov mô hình hóa việc một người dùng Web duyệt ngẫu nhiên trên Web như sự chuyển trạng thái trong chuỗi.
  - Mỗi trang Web, hay đỉnh của đồ thị Web, được xem là một trạng thái.
  - Mỗi siêu liên kết là sự chuyển từ trạng thái này sang trạng thái khác với một xác suất nào đó.



PageRank và Chuỗi Markov

- Giả sử rằng
  - Người duyệt Web sẽ nhấn chọn các siêu liên kết trong trang i một cách ngẫu nhiên phân phối đều.
  - Không sử dụng nút "back" trên trình duyệt
  - Người duyệt Web không nhập trực tiếp URL vào thanh địa chỉ
- Gọi $\boldsymbol{A}$ là ma trận xác suất chuyển có kích thước $n \times n$
  - A_{ij} biểu diễn xác suất chuyển mà người dùng ở trạng thái state i (trang i) sẽ di chuyển đến trạng thái j (trang j).

$$\boldsymbol{A} = \begin{pmatrix} A_{11} & A_{12} & \cdot & \cdot & \cdot & A_{1n} \\ A_{21} & A_{22} & \cdot & \cdot & \cdot & A_{2n} \\ \cdot & \cdot & & & & \cdot \\ \cdot & & & & & \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ A_{n1} & A_{n2} & \cdot & \cdot & \cdot & A_{nn} \end{pmatrix}$$

PageRank và Chuỗi Markov

- Cho trước một vector phân phối xác suất khởi tạo, biểu diễn khả năng người duyệt Web ở một trạng thái (trang) lúc đầu.

$$\boldsymbol{p_0} = \left(p_0(1), p_0(2), \ldots, p_0(n)\right)^{\boldsymbol{T}}$$

- Và cho trước ma trận xác suất chuyển $\boldsymbol{A}$ kích thước $n \times n$

- Như vậy, $\sum_{i=1}^{n} p_0(i) = 1$ và $\sum_{j=1}^{n} A_{ij} = 1$

Mỗi trang Web phải có ít nhất một liên kết ra

- Phương trình thứ hai không hoàn toàn đúng đối với một số trang Web không có liên kết ngoài.

Nếu A là ma trận ngẫu nhiên thống kê

- $\boldsymbol{A}$ là ma trận ngẫu nhiên thống kê của một chuỗi Markov nếu thỏa $\sum_{j=1}^{n} A_{ij} = 1$.
- Xác suất để người duyệt Web ở trạng thái j sau 1 bước chuyển trạng thái là

$$p_1(j) = \sum_{i=1}^{n} A_{ij}(1)\, p_o(i)$$

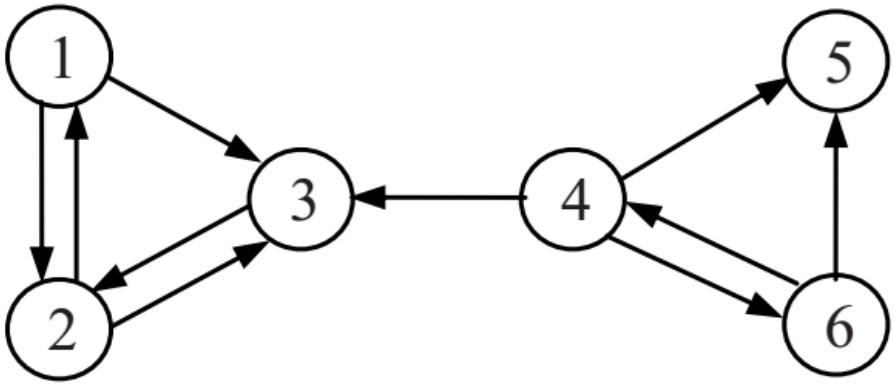
  - Trong đó $A_{ij}(1) = A_{ij}$ là xác suất đi từ i đến j trong 1 bước chuyển
- Ta có thể viết lại dưới dạng ma trận, $\boldsymbol{p_1} = A^T \boldsymbol{p_0}$
- Phân phối xác suất sau k bước chuyển trạng thái là

$$\boldsymbol{p_k} = A^T \boldsymbol{p_{k-1}}$$

Nếu A là ma trận ngẫu nhiên thống kê

- **Định lý Ergodic:** Chuỗi Markov hữu hạn được định nghĩa bởi ma trận $\boldsymbol{A}$ có phân phối xác suất không đổi đơn nhất nếu $\boldsymbol{A}$ là bất khả quy (irreducible) và phi tuần hoàn (aperiodic).
- Stationary probability distribution: $\lim_{k\to\infty} \boldsymbol{p_k} = \boldsymbol{\pi}$
  - $\boldsymbol{p_k}$ sẽ hội tụ về vector xác suất trạng thái ổn định $\boldsymbol{\pi}$ sau một choỗi chuyển trạng thái, bất chất lựa chọn về $\boldsymbol{p_0}$
  - Tại trạng thái ổn định: $\boldsymbol{p_k} = \boldsymbol{p_{k+1}} = \boldsymbol{\pi} \rightarrow \boldsymbol{\pi} = \boldsymbol{A^T\pi} \rightarrow$ PageRank $\boldsymbol{P}$
  - Xác suất đường dài mà một người duyệt Web ngẫu nhiên sẽ thăm viếng các trang
- Trang có độ uy tín cao nếu như nó có xác suất được thăm viếng cao.

Ví dụ về đồ thị chứa siêu liên kết



$$A = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

- $A_{12} = A_{13} = 1/2$: đỉnh 1 có hai liên kết ngoài
- A không phải là ma trận ngẫu nhiên thống kê vì đỉnh 5 không có liên kết ngoài (**dangling page**).

A không là ma trận ngẫu nhiên thống kê

- Có nhiều cách để chuyển A về ma trận chuyển ngẫu nhiên thống kê.
- Loại bỏ các trang không có liên kết ngoài ra khỏi hệ thống trong quá trình tính toán điểm PR
  - Trang bị bỏ không ảnh hưởng trực tiếp đến thứ hạng của trang khác.
  - Liên kết ngoài từ những trang khác trỏ đến trang bị bỏ cũng được gỡ bỏ $\rightarrow$ xác suất chuyển bị ảnh hưởng nhưng không đáng kể
  - Sau quá trình tính toán, những trang bị bỏ và siêu liên kết trỏ đến chúng được phục hồi lại và điểm PR của chúng được tính theo phương trình $P = A^T P$.

A không là ma trận ngẫu nhiên thống kê

- Từ mỗi trang i như thế, thêm liên kết ngoài đến mọi trang có trong Web $\rightarrow$ xác suất đồng dạng $1/n$
  - Thay mỗi dòng chứa toàn số 0 trong A bằng $\boldsymbol{e}/n$, trong đó $\boldsymbol{e}$ là vector n-chiều chứ toàn số 1

$$\overline{\boldsymbol{A}} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}.$$

Từ trang 5 thêm một liên kết ngoài đến mỗi trang

A bất khả quy

- Bất khả quy nghĩa là đồ thị Web G liên thông mạnh.
- Đồ thị có hướng $G = (V, E)$ liên thông mạnh nếu và chỉ nếu với mỗi cặp đỉnh, $u, v \in V$, đều có đường đi từ u đến v.
- Đồ thị Web tổng quát biểu diễn bởi A có thể không bất khả quy vì có vài cặp đỉnh (u, v) không tồn tại đường đi $u \to v$.
  - Ví dụ, trong $\bar{A}$, không có đường đi có hướng từ đỉnh 3 đến đỉnh 4

A phi tuần hoàn

- Trạng thái i trong chuỗi Markov được gọi là tuần hoàn khi tồn tại một chu trình có hướng mà chuỗi phải duyệt qua
- Trạng thái i là tuần hoàn với chu kỳ $k > 1$ nếu k là giá trị nhỏ nhất mà mọi đường đi từ i trỏ về i có độ dài là bội số của k.
- Trang thái i được gọi là phi tuần hoàn khi $k = 1$.
- Chuỗi Markov phi tuần hoàn có mọi trạng thái phi tuần hoàn

Ví dụ: tuần hoàn

- Hình 5 cho thấy một chuỗi Markov với k = 3. Vd, nếu chúng ta bắt đầu từ trạng thái 1, để trở về trạng thái 1 đường đi duy nhất là 1-2-3-1 với một vài lần, gọi là h. Do đó, bất kì sự trở về trạng thái một sẽ mất 3 dịch chuyển.

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

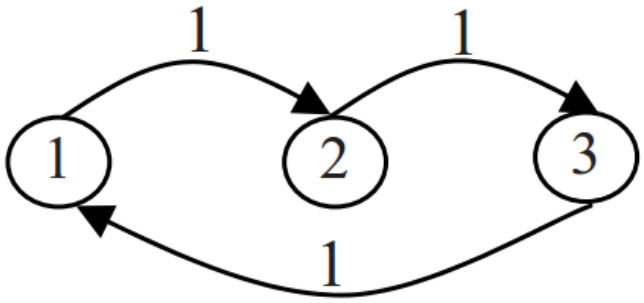


Fig. 7.6. A periodic Markov chain with $k = 3$.

Giải pháp cho cả hai vấn đề

- Thêm một liên kết từ mỗi trang đến mọi trang khác và gán cho mỗi liên kết một xác suất chuyển nhỏ được quản lý bằng tham số d.

$$(1-d)\frac{\boldsymbol{E}}{n}+d\boldsymbol{A}^{T}=\begin{pmatrix} 1/60 & 7/15 & 1/60 & 1/60 & 1/6 & 1/60 \\ 7/15 & 1/60 & 11/12 & 1/60 & 1/6 & 1/60 \\ 7/15 & 7/15 & 1/60 & 19/60 & 1/6 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 1/60 \end{pmatrix}$$

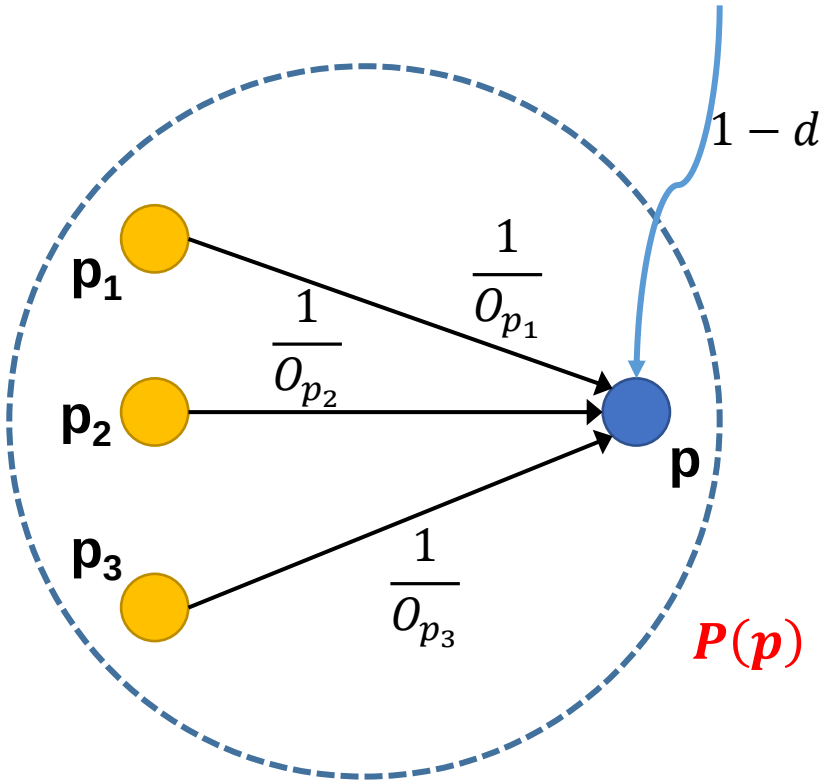
Mô hình PageRank cải tiến

- Tại một trang, người duyệt Web ngẫu nhiên có hai lựa chọn
  - Với xác suất d, chọn ngẫu nhiên một liên kết ngoài để đi theo đó.
  - Với xác suất $1 - d$, nhảy đến một trang ngẫu nhiên mà không cần đến liên kết.

- Mô hình cải tiến là $$\boldsymbol{P} = \left((1-d)\frac{\boldsymbol{E}}{n} + d\boldsymbol{A}^{\boldsymbol{T}} \right) \boldsymbol{P}$$
  - Trong đó $\boldsymbol{E} = \boldsymbol{e}\boldsymbol{e}^{\boldsymbol{T}}$ là ma trận vuông kích thước $n \times n$ chứa số 1, và $1/n$ là xác suất nhảy đến một trang nào đó trong đồ thị Web n trang
  - A được giả sử đã là ma trận ngẫu nhiên thống kê
  - $d \in [0,1]$: **damping factor**, $d = 0.85$

Mô hình PageRank cải tiến

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji}P(j) \qquad P(i) = (1-d) + d\sum_{(j,i)\in E} \frac{P(j)}{O_j}$$



$$P(p) = (1-d) + d\left(\frac{P(p_1)}{O_{p_1}} + \frac{P(p_2)}{O_{p_2}} + \frac{P(p_3)}{O_{p_3}}\right)$$

Giải thuật PageRank

PageRank-Iterate(G)

$\boldsymbol{P}_0 \leftarrow \boldsymbol{e}/n$

$k \leftarrow 1$

repeat

$\boldsymbol{P}_k \leftarrow (1-d)\boldsymbol{e} + d\boldsymbol{A}^T \boldsymbol{P}_{k-1}$;

$k \leftarrow k+1$;

until $||\boldsymbol{P}_k - \boldsymbol{P}_{k-1}||_1 < \varepsilon$

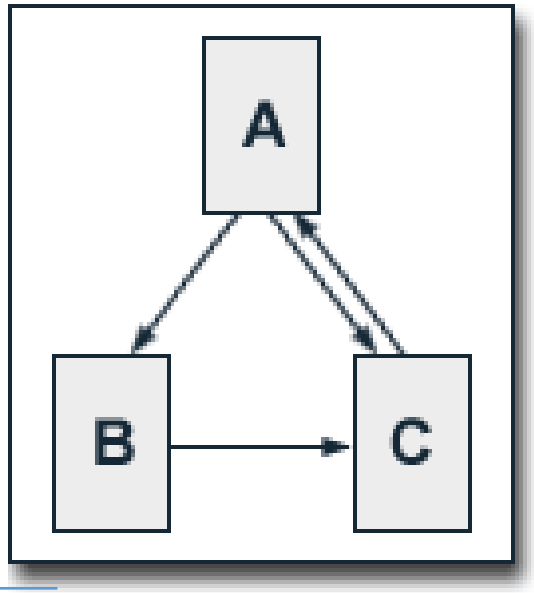
return $\boldsymbol{P}_k$

Phương pháp power iteration để tính điểm PageRank

- Trạng thái hội tụ thật sự đôi khi không cần thiết vì ta chỉ qua tâm đến thứ hạng giữa các trang.
- Cơ sở dữ liệu có 322 triệu liên kết đạt hội tụ sau khoảng 52 vòng lặp

Giải thuật PageRank: Ví dụ

- Sử dụng hệ số $d = 0.5$.
- P(A) = 0.5 + 0.5 * P(C)
- P(B) = 0.5 + 0.5 * (P(A) / 2)
- P(C) = 0.5 + 0.5 * (P(A) / 2 + P(B))



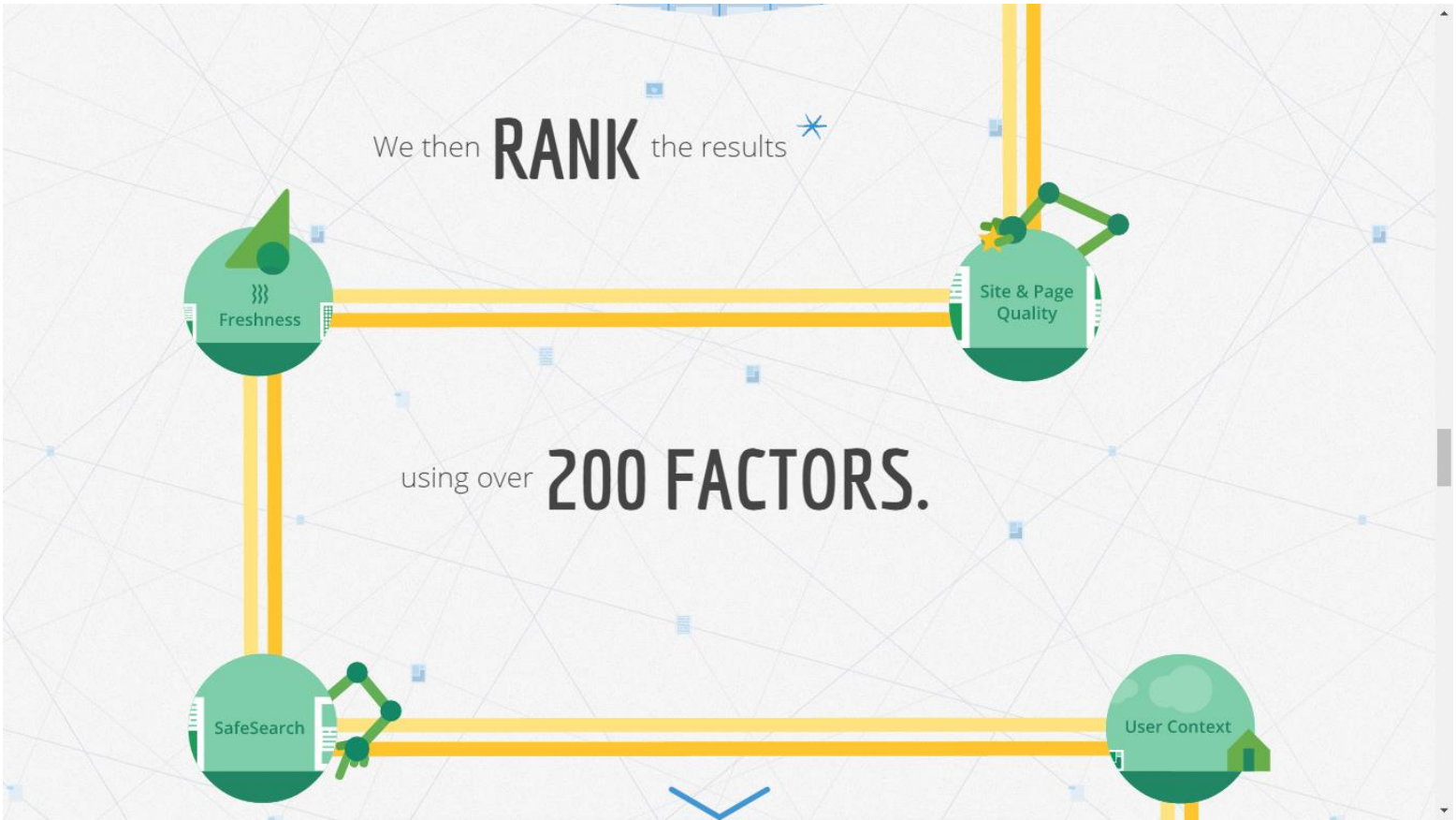
Vòng lặp	P(A)	P(B)	P(C)
0	0.333333	0.333333	0.333333
1	0.666667	0.583333	0.75
2	0.875	0.666667	0.958333
3	0.979167	0.71875	1.052083
4	1.026042	0.744792	1.104167
5	1.052083	0.75651	1.128906

Đánh giá giải thuật PageRank

- Ưu điểm chính: khả năng chống spam
  - Một trang là quan trọng nếu các trang trỏ đến nó quan trọng.
  - Người chủ của một trang Web khó có thể xâm nhập vào những trang khác để tạo liên kết trong đến trang của mình.
- Độ đo toàn cục, không phụ thuộc câu truy vấn
  - Điểm PR của mọi trang được tính toán và lưu trữ ngoại tuyến.
  - Đạt hiệu suất tốt tại thời điểm truy vấn: chỉ cần tra giá trị để tích hợp với các chiến lược khác khi xếp hạng trang

Đánh giá giải thuật PageRank

- Không thể phân biệt giữa trang có uy tín chung chung và trang có uy tín đặc thù theo chủ đề truy vấn.
- Giải thuật gốc chưa xét tính cập nhật theo thời gian.

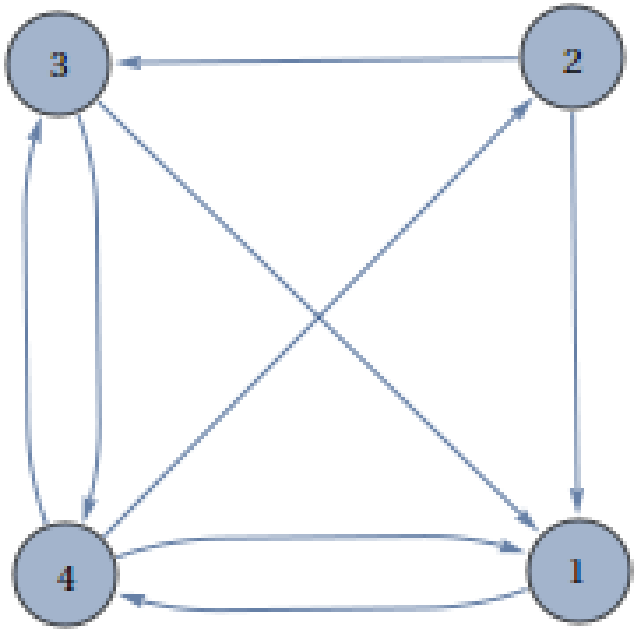


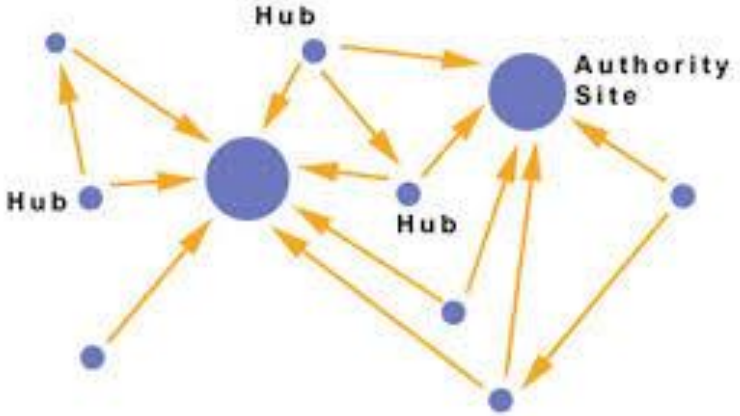
Xếp hạng dựa trên liên kết

- Không phải là chiến lược duy nhất được sử dụng trong cỗ máy tìm kiếm
  - Các phương pháp như truy vấn thông tin, heuristic và tham số thực nghiệm, v.v. cũng được sử dụng.
- PageRank cũng không phải là giải thuật xếp hạng toàn cục và dựa trên liên kết tĩnh duy nhất.
  - Mọi cỗ máy tìm kiếm lớn (ví dụ như Bing and Yahoo!) đều có giải thuật riêng.
- Các nhà nghiên cứu cũng đề xuất nhiều giải thuật xếp hạng khác mà không dựa trên liên kết.
  - Ví dụ, BrowseRank – tìm kiếm trên đồ thị dựng từ user search log

Bài tập 1: Giải thuật PageRank

- Cho tập hợp gồm 4 trang Web liên kết với nhau như hình bên dưới.
- Tính giá trị PR qua các vòng lặp. Biết rằng d = 0.85 và giá trị khởi tạo của mỗi trang bằng 1/n.





Giải thuật HITS

Giải thuật HITS

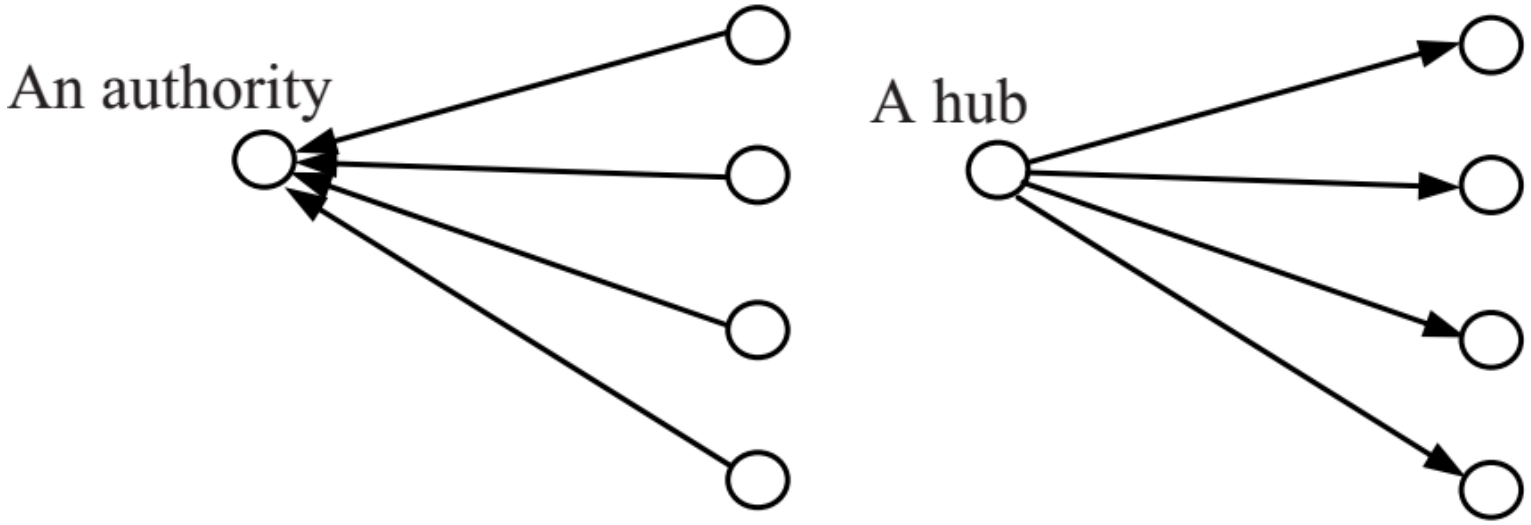
- Kleinberg, J. Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM), 1999, 46(5): pp. 604-632.
- Jon Kleinberg (1971), American computer scientist.

Giải thuật HITS

- **H**ypertext **I**nduced **T**opic **S**earch (**HITS**)
- Phụ thuộc vào câu truy vấn
- Người dùng đưa ra một câu truy vấn tìm kiếm.
- HITS trước tiên mở rộng danh sách trang liên quan được trả về từ một cỗ máy tìm kiếm cơ bản
- Tiếp đó, HITS tạo hai danh sách xếp hạng trên tập trang mở rộng, gọi là **authority ranking** và **hub ranking**

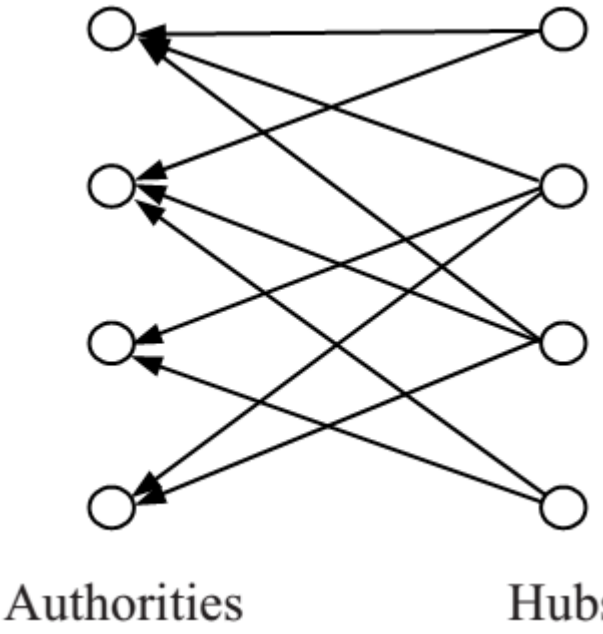
Trang authority và trang hub

- Trang authority là trang có nhiều liên kết trong.
  - Trang có nội dung đáng tin cậy về một chủ đề nào đó → nhiều người tin tưởng và đặt liên kết trỏ đến.
- Trang hub là trang có nhiều liên kết ngoài.
  - Trang đóng vai trò người tổ chức thông tin về một chủ đề cụ thể và do đó trỏ đến nhiều trang authority tốt trong chủ đề này.



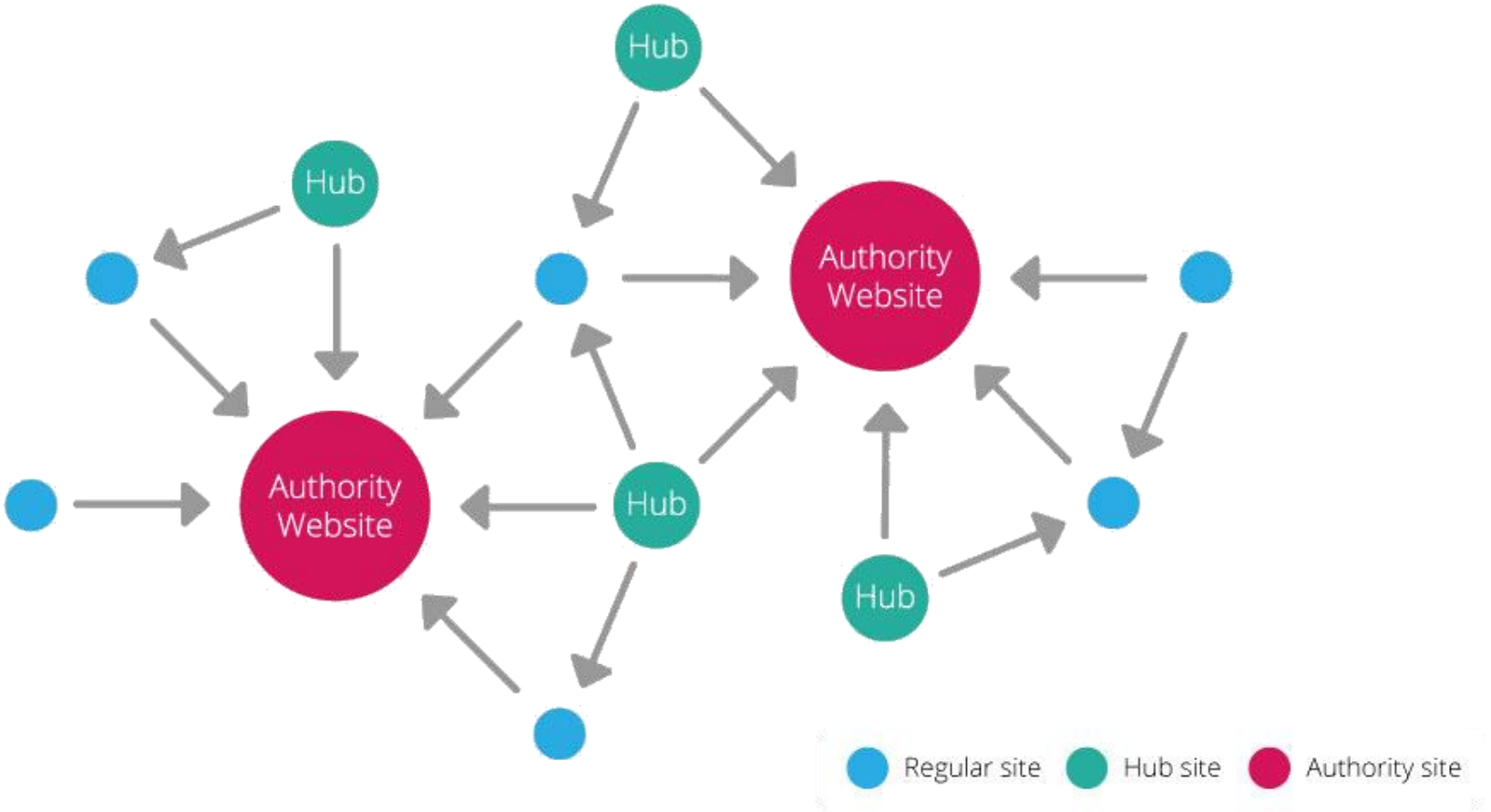
Trang authority và trang hub

- Tăng cường qua lại (Mutual reinforcement): Một trang hub tốt trỏ đến nhiều trang authority tốt, và ngược lại một trang authority tốt được trỏ đến bởi nhiều trang hub tốt.

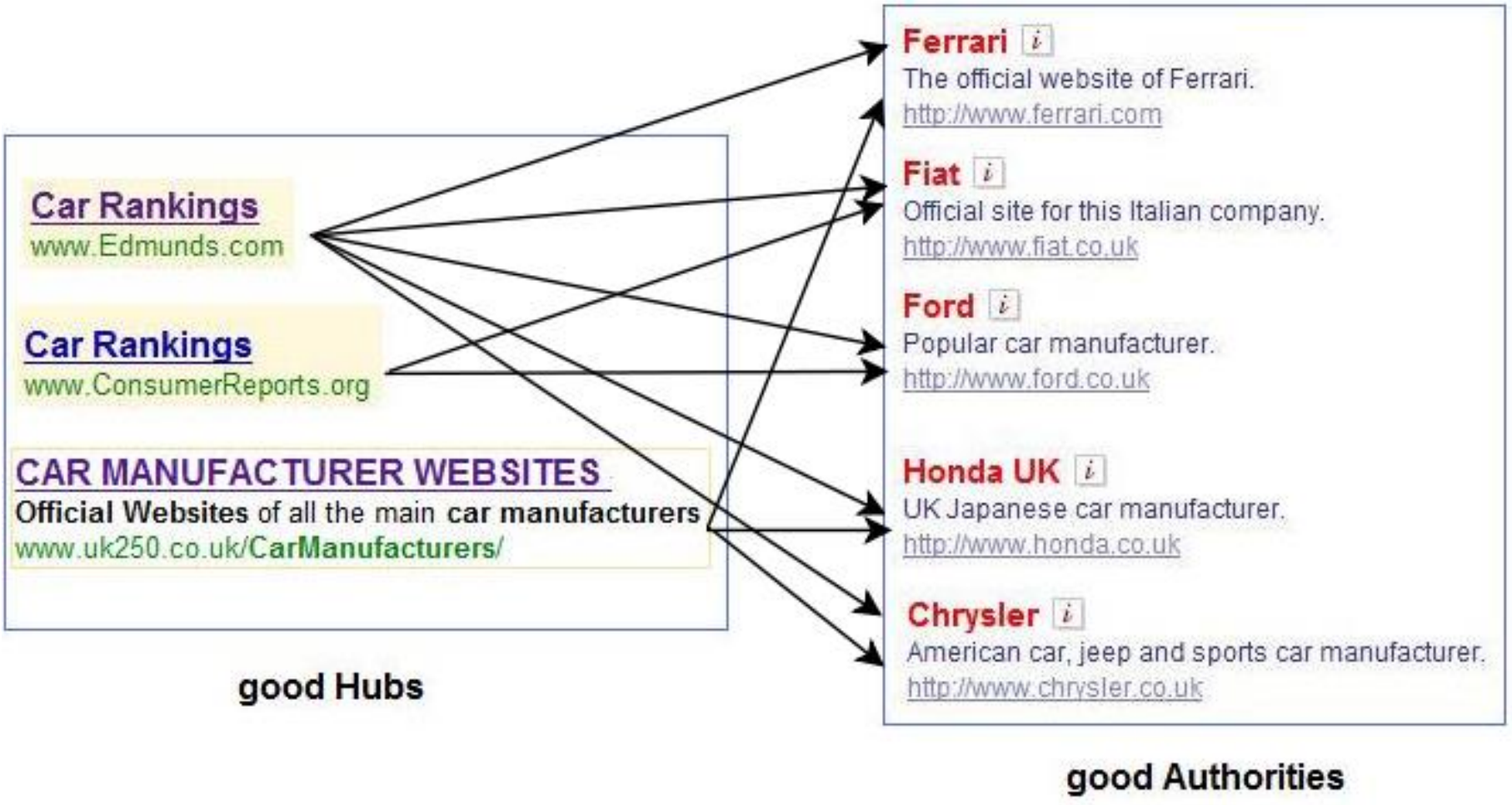


Một tập hợp các trang hub và authority liên kết dày đặc lẫn nhau, tạo thành cấu trúc đồ thị con lưỡng phân.

Trang authority và trang hub: Ví dụ



Trang authority và trang hub: Ví dụ

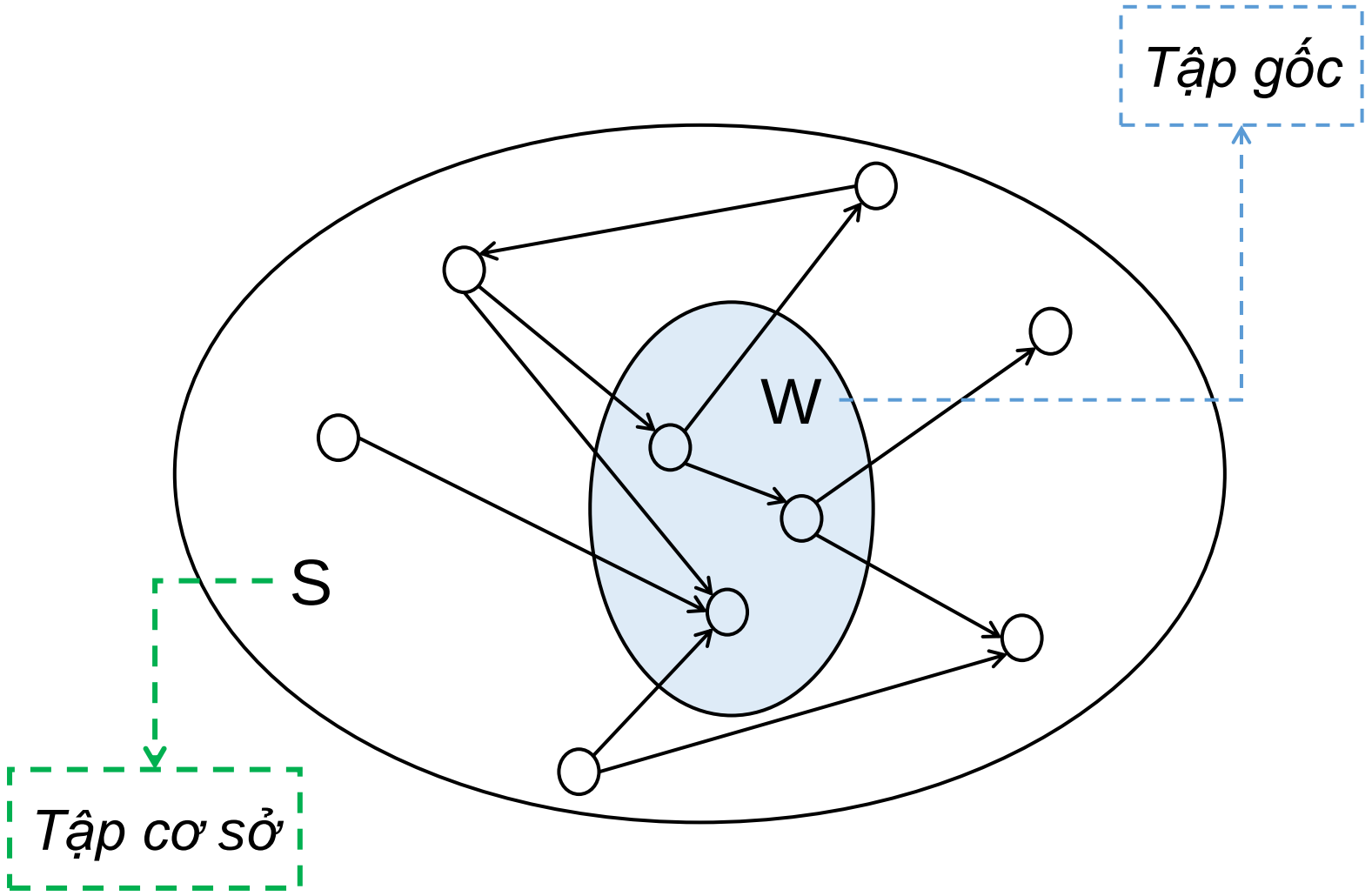


Query: Top automobile makers

Thu thập trang để xếp hạng

- Cho trước câu truy vấn q có chủ đề tìm kiếm rộng.
- HITS thu thập các trang cần thiết trong hai bước như sau.
- **Bước 1:** gửi q đến cỗ máy tìm kiếm và thu thập **tập gốc** (root set) $\boldsymbol{W}$ gồm t trang được xếp hạng cao nhất (đề xuất t = 200)
- **Bước 2:** xây dựng **tập cơ sở** (base set) $\boldsymbol{S}$ bằng cách bổ sung thêm vào $\boldsymbol{W}$ những trang được trỏ đến bởi một trang trong $\boldsymbol{W}$ và những trang trỏ đến một trang trong $\boldsymbol{W}$.
  - S có thể rất lớn $\rightarrow$ Giới hạn: Mỗi trang trong W được phép đưa vào S tối đa k pages (đề xuất k = 50) trỏ đến nó.

Thu thập trang để xếp hạng



Điểm authority và điểm hub

- Tiếp đó, HITS làm việc trên các trang thuộc S, gán cho mỗi trang trong S một điểm authority và một điểm hub.
- Gọi số trang cần xét là n.
- Đồ thị liên kết có hướng của S được ký hiệu là $G = (V, E)$.
- Ma trận kề tương ứng của đồ thị là $L = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & otherwise \end{cases}$

Điểm authority và điểm hub

- Gọi trang i có điểm authority $a(i)$ và điểm hub $h(i)$.
- Mối quan hệ tăng cường lẫn nhau giữa hai điểm số là

$$\mathbf{a}(\boldsymbol{i}) = \sum_{(\boldsymbol{j},\boldsymbol{i})\in \boldsymbol{E}} \boldsymbol{h}(\boldsymbol{j}) \qquad \boldsymbol{h}(\boldsymbol{i}) = \sum_{(\boldsymbol{i},\boldsymbol{j})\in \boldsymbol{E}} \boldsymbol{a}(\boldsymbol{j})$$

- Biểu diễn theo dạng ma trận là

$$\boldsymbol{a} = \boldsymbol{L}^{\boldsymbol{T}}\boldsymbol{h} \qquad \boldsymbol{h} = \boldsymbol{L}\boldsymbol{a}$$

  - Trong đó $a = (a(1), a(2), \dots, a(n))^T$ và $h = (h(1), h(2), \dots, h(n))^T$ là các vector cột.

Giải thuật HITS

- HITS tìm các principal eigenvector tại "trạng thái cân bằng".

HITS-Iterate(G)

$\boldsymbol{a}_0 \leftarrow \boldsymbol{h}_0 \leftarrow (1, 1, \ldots, 1)$;

$k \leftarrow 1$

Repeat

$\boldsymbol{a}_k \leftarrow \boldsymbol{L}^T \boldsymbol{L}\boldsymbol{a}_{k-1}$;

$\boldsymbol{h}_k \leftarrow \boldsymbol{L}\boldsymbol{L}^T \boldsymbol{h}_{k-1}$;

$\boldsymbol{a}_k \leftarrow \boldsymbol{a}_k / ||\boldsymbol{a}_k||_1$; // normalization

$\boldsymbol{h}_k \leftarrow \boldsymbol{h}_k / ||\boldsymbol{h}_k||_1$; // normalization

$k \leftarrow k + 1$;

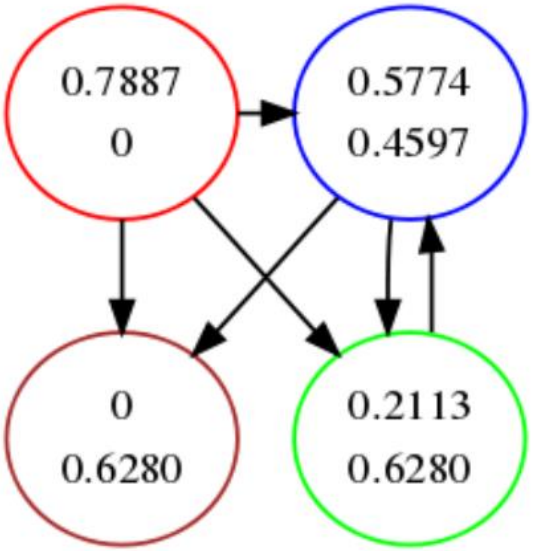
until $||\boldsymbol{a}_k - \boldsymbol{a}_{k-1}||_1 < \varepsilon_a$ and $||\boldsymbol{h}_k - \boldsymbol{h}_{k-1}||_1 < \varepsilon_h$;

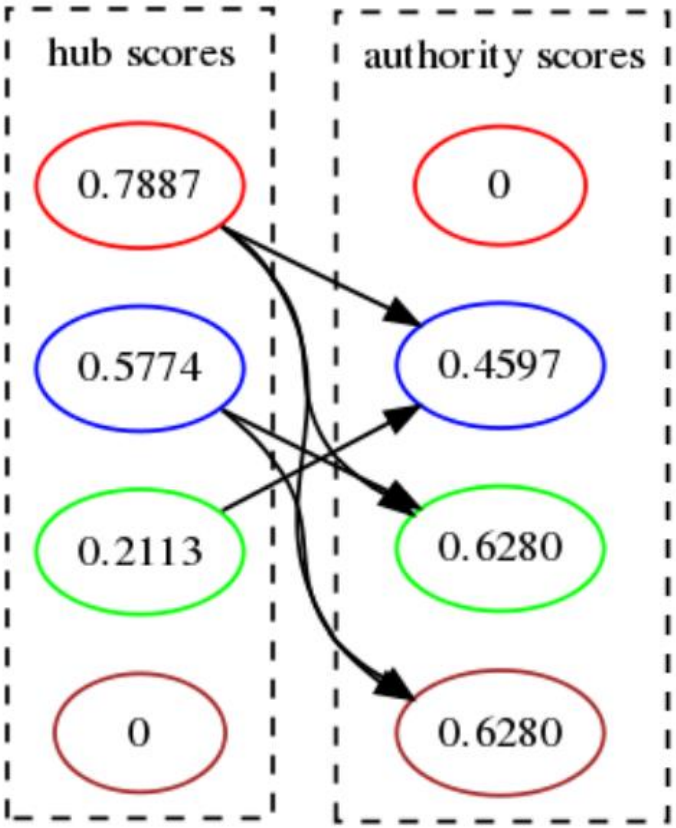
return $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$

Giải thuật HITS dựa trên phương pháp power iteration

Giải thuật HITS

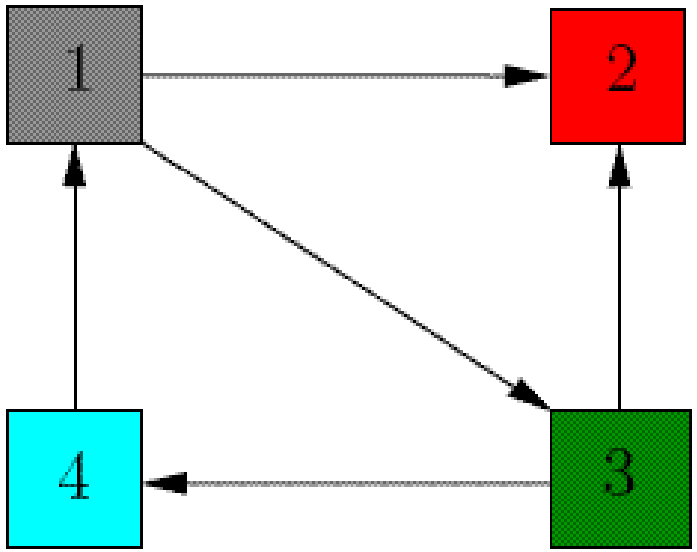
- HITS chọn ra một vài trang xếp hạng đầu làm các trang authority và trang hub, rồi đưa chúng cho người dùng.
  - Trang có điểm authority lớn là trang authority tốt, và trang có điểm hub lớn là trang hub tốt.





Giải thuật HITS: Ví dụ

Vòng lặp	Điểm số	Trang			
		1	2	3	4
0	Hub	1	1	1	1
	Authority	1	1	1	1
1	Hub	2	0	2	1
	Authority	1	2	1	1
2	Hub	3	0	3	1
	Authority	1	4	2	2
3	Hub	6	0	6	1
	Authority	1	6	3	3



Vấn đề hội tụ đơn nhất

- HITS luôn luôn hội tụ nhưng gặp vấn đề về tính đơn nhất của sự hội tụ của các vector điểm authority và hub.
  - Đối với một số loại đồ thị nhất định, các lượt khởi tạo khác nhau trong phương pháp power iteration có thể tạo ra vector điểm authority và hub cuối cùng khác nhau.
  - Một số kết quả có thể thiếu nhất quán hoặc sai lầm.
- $L^T L$ (và tương tự LL^T) là khả quy (reducible) $\rightarrow$ các dominant (principal) eigenvalue lặp lại

Mối quan hệ với ma trận C và B

- Trang authority giống như một bài báo có sức ảnh hưởng, được trích dẫn bởi nhiều bài báo sau đó.

$$C_{ij} = \sum_{k=1}^{n} L_{ki} L_{kj} = \left(L^T L\right)_{ij}$$

- Trang hub giống như một bài báo khảo sát, trích dẫn nhiều bài báo khác (kể cả những bài báo có sức ảnh hưởng).

$$B_{ij} = \sum_{k=1}^{n} L_{ki} L_{kj} = \left(L L^T\right)_{ij}$$

Đánh giá giải thuật HITS

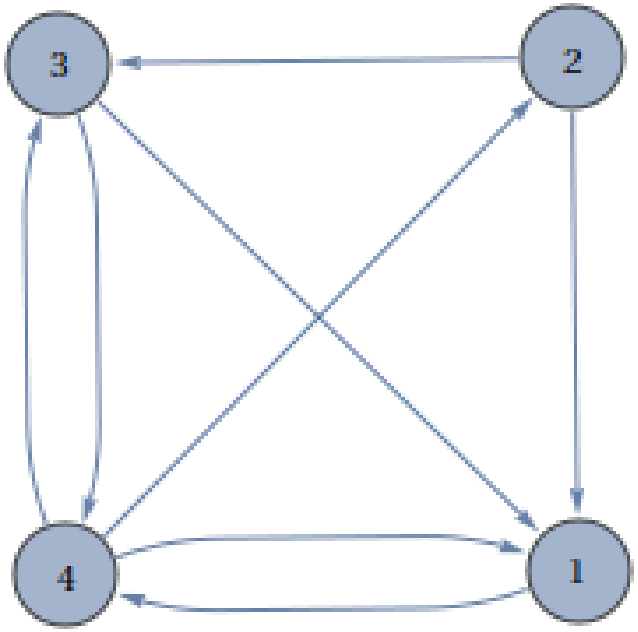
- Xếp hạng các trang theo chủ đề truy vấn → cung cấp trang hub và trang authority liên quan hơn
- Kết quả xếp hạng này cũng có thể kết hợp với các phương pháp xếp hạng dựa trên truy vấn thông tin
- Không có khả năng chống spam mạnh như PageRank
  - Dễ dàng tác động đến HITS bằng cách thêm liên kết ngoài từ một trang đến các trang authority tốt → điểm hub của trang này được khuếch đại → điểm authority cũng được khuếch đại.

Đánh giá giải thuật HITS

- Lạc chủ đề
  - Giải thuật trong quá trình mở rộng tập gốc dễ dàng thu thập phải nhiều trang không liên quan đến chủ đề tìm kiếm (kể cả khi chúng là trang hub và trang authority)
  - Liên kết ngoài của một trang không nhất thiết trỏ đến trang cùng chủ đề. Liên kết trong đến trang ở tập gốc cũng có thể không liên quan
  - Người ta đặt siêu liên kết với mọi lý do khác nhau đa dạng, không nhất thiết cùng chủ đề, kể cả spamming
- Thời gian đánh giá truy vấn cũng là một hạn chế lớn.
  - Thu thập và mở rộng tập gốc, tính toán eigenvector, v.v. đều là những thao tác tốn thời gian.

Bài tập 2: Giải thuật HITS

- Cho tập hợp gồm 6 trang Web liên kết với nhau như hình bên dưới.
- Tính giá trị hub và authority qua các vòng lặp. Các giá trị khởi tạo đều là 1.



Khám phá cộng đồng

Cộng đồng (community)

- Nhóm thực thể chia sẻ mối quan tâm chung hoặc cùng tham dự vào một hoạt động/sự kiện nào đó
  - Không chỉ ở Web, cộng đồng còn tồn tại trong ngữ cạnh quan hệ giữa các email và tài liệu văn bản
- Được biểu diễn bằng đồ thị con lưỡng phân liên kết dày đặc

Khám phá cộng đồng

- Có nhiều lý do để nghiên cứu bài toán khám phá cộng đồng
- Cộng đồng cung cấp nguồn thông tin có giá trị, đáng tin cậy và có tính cập nhật cao cho các nhà phân tích dữ liệu
- Biểu diễn khía cạnh xã hội học của Web $\rightarrow$ sự thấu hiểu về quá trình tiến hóa của Web
- Quảng cáo hướng đối tượng có thể thực hiện rất chính xác

Phát biểu bài toán

- Cho trước tập hợp hữu hạn $S = \{s_1, s_2, \dots, s_n\}$ gồm các thực thể cùng loại.
- Mỗi **cộng đồng** (community) là một cặp $C = (T, G)$.
  - Trong đó T là chủ đề của cộng đồng (community theme) và $G \subseteq S$ là tập hợp mọi thực thể thuộc S cùng chia sẻ chủ đề T.
- Nếu $s_i \in G$ thì s_i được gọi là thành viên của cộng đồng C.

Phát biểu bài toán

- Một cộng đồng (T, G) có thể có nhiều cộng đồng con (sub-community) $\{(T_1, G_1), \ldots, (T_m, G_m)\}$
  - T_i được gọi là một chủ đề con của T và $G_i \subseteq G$.
- Ngược lại, (T, G) được gọi là cộng đồng cha (super-community) của (T_i, G_i).
- Mỗi cộng đồng con lại còn có thể được phân rã nhỏ hơn, tạo thành cấu trúc phân cấp cộng đồng.

Thể hiện cộng đồng trong dữ liệu

Hyperlinks	Emails exchange	Co-occurrence of entities
Content words	Content words	Content words

- Các thành viên trong cộng đồng liên kết theo cách nào đó.
- Văn bản kết hợp đi kèm thường chứa từ ngữ biểu thị chủ đề của cộng đồng

Bài toán khám phá cộng đồng

- Cho trước tập dữ liệu gồm các thực thể
- Khám phá từ cấu trúc liên kết những cộng đồng tiềm ẩn
- Với mỗi cộng đồng, tìm chủ đề và các thành viên tương ứng
  - Chủ đề thường được biểu diễn bằng tập hợp các từ khóa.

Cộng đồng lõi lưỡng phân

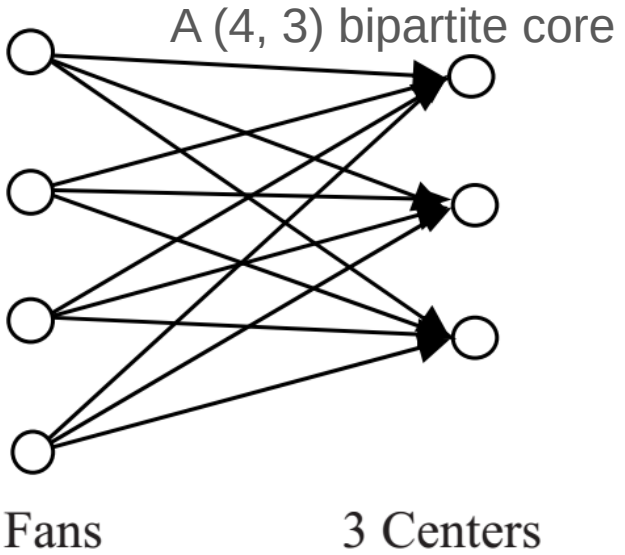
- HITS tìm các cộng đồng dạng đồ thị lưỡng phân dày đặc dựa trên các câu truy vấn chủ đề rộng.
- Tuy nhiên, tính toán eigenvector có hiệu suất tương đối thấp

Làm thế nào để tìm mọi cộng đồng như thế một cách hữu hiệu từ một lượt crawling trên toàn bộ Web?

Cộng đồng theo lõi lưỡng phân

- Lõi lưỡng phân (Bipartite core): đồ thị con lưỡng phân đầy đủ với ít nhất i đỉnh trong F và ít nhất j đỉnh trong C.
  - Trong đó F và C là hai tập con của đồ thị lưỡng phân.
  - Các cạnh trong F (hoặc C) có ý nghĩa phù hợp với ngữ cảnh Web.
- Một core hầu như chắc chắn biểu diễn một cộng đồng, nhưng một cộng đồng có thể có nhiều lõi lưỡng phân.

Các fan giống như những trang *hub* chuyên biệt cho một chủ đề, và các centers giống như những trang *authority*.



Tìm các bipartite core: Tỉa nhánh

- Loại bỏ những trang không đủ tiêu chuẩn làm fan hoặc center
- **Tỉa nhánh bằng bậc trong:** xóa mọi trang được tham chiếu (liên kết) cực nhiều trên Web
  - Ví dụ, trang chủ của các Web portals (Yahoo!, AOL, v.v.)
  - Những trang này được tham chiều vì nhiều lý do khác nhau, nhưng ít liên quan đến sự phát sinh của một cộng đồng $\rightarrow$ an toàn để xóa
  - Xóa những trang có số liên kết trong lớn hơn k, giá trị k thường được xác định theo thực nghiệm ($k = 50$).
- **Vòng lặp tỉa nhánh fan và center:** Nếu cần tìm các (i, j)-core, lặp nhiều lần việc xóa bất kỳ fan nào có bậc ngoài nhỏ hơn j cùng với các cạnh có liên kết với nó.
  - Thực hiện tương tự cho bất kỳ center nào có bậc trong nhỏ hơn i

Tìm các bipartite core: Phát sinh

- Trang còn lại sau tỉa nhánh được dùng để phát hiện core.
- Cố định j, bắt đầu với mọi $(1, j)$-core, là tập hợp mọi đỉnh có bậc ngoài ít nhất là j.
- Xay dựng mọi $(2, j)$-core bằng cách kiểm tra mọi fan trỏ đến một center bất kỳ trong một $(1, j)$-core
- Tìm mọi $(3, j)$-core theo cách tương tự, kiểm tra mọi fan trỏ đến một center bất kỳ trong một $(2, j)$-core
- Cứ như thế tiếp diễn.

Cộng đồng theo bipartite core

- Từ một lượt crawling trên Web có thể tìm được nhiều core gắn kết chặt chẽ về mặt chủ đề
- Chỉ tìm được những trang cốt lõi của cộng đồng, không phải toàn bộ trang
- Không thể xác định chủ đề của cộng đồng cũng như tổ chức phân cấp của cộng đồng

Cộng đồng luồng tối đại

- Các bipartite core thường rất nhỏ và do đó không thể biểu diễn toàn bộ cộng đồng.
- Người dùng cung cấp tập S chứa trang hạt giống
  - Những trang mẫu của cộng đồng đang được tìm kiếm
- Hệ thống dựa trên S để duyệt Web tìm nhiều trang hơn.
- Áp dụng giải thuật luồng tối đại để phân tách cộng đồng C liên quan đến các trang hạt giống và những trang khác.
- Lặp lại các bước trên đến khi tìm ra cộng đồng mong muốn.

- Không thể xác định chủ đề của cộng đồng cũng như tổ chức phân cấp của cộng đồng

Cộng đồng luồng tối đại

Algorithm Find-Community (S)
  while number of iteration is less than desired **do**
    build $G = (V, E)$ by doing a fixed depth crawl starting from S;
    $k = |S|$;
    C = Max-Flow-Community(G, S, k);
    rank all $v \in C$ by the number of edges in C;
    add the highest ranked non-seed vertices to S
  end-while
  return all $v \in V$ still connected to the source s

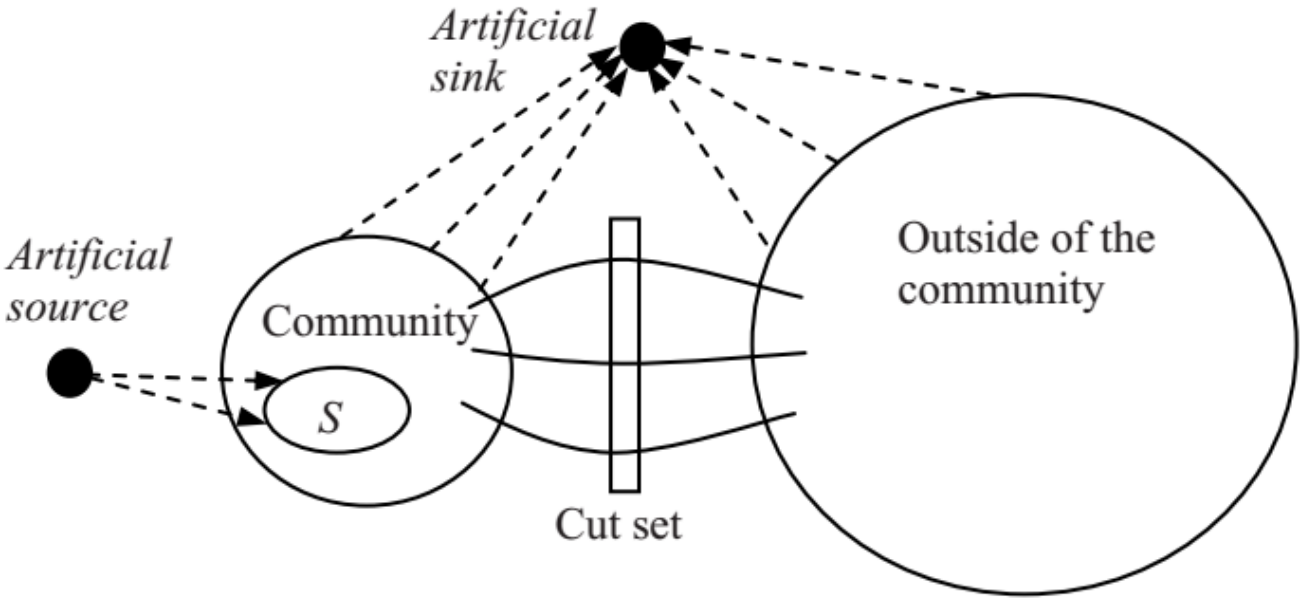
Giải thuật tiến hành crawling đến một độ sâu cố định, xét các liên kết trong cũng như liên kết ngoài (với liên kết trong được phát hiện bằng cách truy vấn trên một cỗ máy tìm kiếm)

Cộng đồng luồng tối đại

Procedure Max-Flow-Community(G, S, k)
create artificial vertices, s and t and add to V; // V is the vertex set of G.
for all $v \in S$ **do**
add (s, v) to E with $c(s, v) = \infty$ // E is the edge set of G.
endfor
for all $(u, v) \in E, u \neq s$ **do**
$c(u, v) = k$;
if $(v, u) \notin E$ **then**
add (v, u) to E with $c(v, u) = k$
endif
endfor
for all $v \in V, v \notin S \cup \{s, t\}$ **do**
add (v, t) to E with $c(v, t) = 1$
endfor
Max-Flow(G, s, t);
return all $v \in V$ still connected to s.

Nguồn và đích nhân tạo

- Đồ thị Web không có nguồn (source) và đích (sink) $\rightarrow$ trước tiên tăng cường G bằng nguồn và đích nhân tạo, s và t.
  - s có cạnh đi đến mọi đỉnh hạt giống trong S, với dung lượng vô hạn
  - Mọi đỉnh, ngoại trừ nguồn, đích và các đỉnh hạt giống, được hướng đến t với dung lượng đơn vị



Cộng đồng luồng tối đại

- Mỗi cạnh đã tồn tại được chuyển đổi thành 2 chiều và được gán một dung lượng hằng số k.
  - k được chọn theo kinh nghiệm, thường bằng kích thước của tập $S \rightarrow$ tạo ra cùng những lát cắt trên đồ thị gốc và đồ thị tăng cường
- Áp dụng thủ tục luồn tối đại, Max-Flow(), để tạo đồ thị residual flow graph.

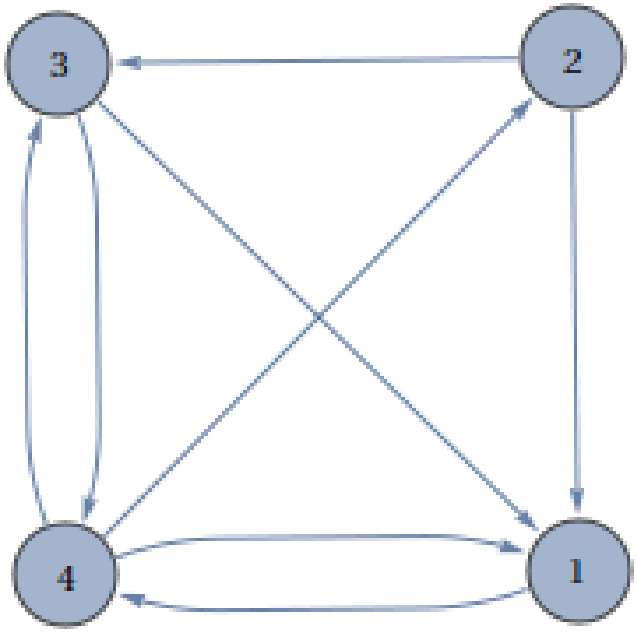
Tài liệu tham khảo

- Bing Liu. 2007. *Web Data Mining-Exploring Hyperlinks, Contents, and Usage Data.* Springer Series on Data-Centric Systems and Applications. **Chapter 7**.

Bài tập 1: Giải thuật PageRank

- PR(1) = 0.15 + 0.85*[PR(2)/2 + PR(3)/2)+PR(4)/3]
- PR(2) = 0.15 + 0.85*[PR(4)/3]
- PR(3) = 0.15 + 0.85*[PR(2)/2 + PR(4)/3]
- PR(4) = 0.15 + 0.85*[PR(1) + PR(3)/2)]

Page	1	2	3	4
Loop 0	0.25	0.25	0.25	0.25
Loop 1	0.433	0.221	0.327	0.469
Loop 2	0.516	0.283	0.377	0.657



Bài tập 2: Giải thuật HITS

- H(1) = A(4)
- H(2) = A(1) + A(3)
- H(3) = A(1) + A(4)
- H(4) = A(1) + A(2) + A(3)

A(1) = H(2) + H(3) + H(4)

A(2) = H(4)

A(3) = H(2) + H(4)

A(4) = H(1) + H(3)

Page		1	2	3	4
Loop 0	H	1	1	1	1
	A	1	1	1	1
Loop 1	H	1	2	2	3
	A	3	1	2	2
Loop 2	H	2	5	5	6
	A	7	3	5	3