TOÁN HỌC TRONG TÌM KIẾM WEB
The Mathematics of Web Search
(Raluca Tanase - Remus Radu)

Bản dịch tiếng Việt của Hoàng Ngọc Thế
(Diễn đàn Toán học – *www.diendantoanhoc.net*)

Ngày 17 tháng 4 năm 2016

Mục lục

1 SƠ LƯỢC VỀ ĐẠI SỐ TUYẾN TÍNH 6

2 ĐỒ THỊ CÓ HƯỚNG - MA TRẬN CHUYỂN TIẾP 13

3 THUẬT TOÁN PageRank - TOÁN HỌC CỦA Google Search 18

4 THUẬT TOÁN HITS - Hub VÀ TRANG CÓ UY TÍN TRÊN INTERNET 26

5 TOÁN INTERNET 32

6 TÀI LIỆU THAM KHẢO 34

LỜI GIỚI THIỆU

Câu lạc bộ toán học Explorer (Hoa Kì) đã có một modun về Toán học trong tìm kiếm Web. Công trình này được tài trợ bởi *Quỹ Khoa học Quốc gia Hoa Kì*. Đây là một tài liệu hữu ích cho những người nghiên cứu về Lý thuyết đồ thị và Khoa học máy tính.

Với hi vọng phổ biến kiến thức toán học và khoa học máy tính, tôi đã cố gắng dịch tài liệu này sang tiếng Việt. Trong quá trình dịch, tôi đã trung thành với bản gốc, cả ví dụ cũng như các hành văn của tác giả. Tôi xin chân thành cảm ơn bạn Trần Trung Kiên (TP Hồ Chí Minh) đã đọc và cho tôi nhiều góp ý giá trị.

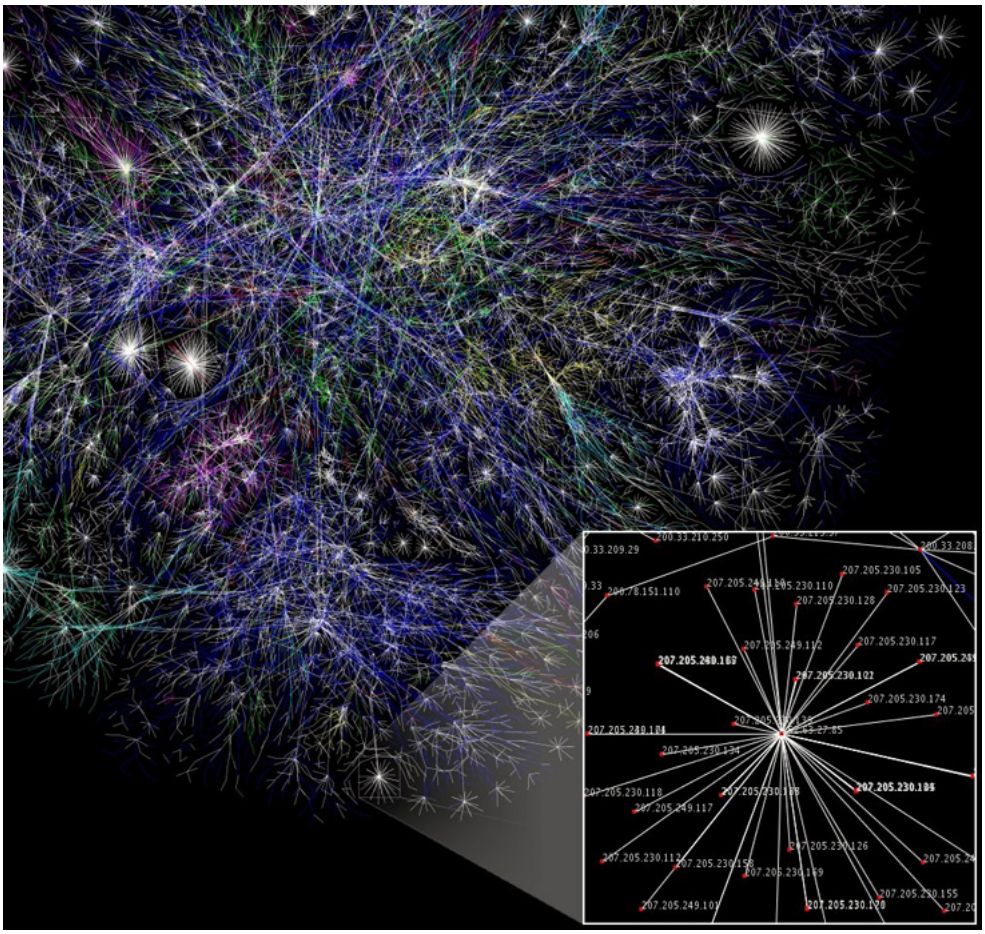
Vì bản thân không phải là người trong ngành nên quá trình dịch của tôi chắc chắn có nhiều sai sót. Rất mong được bạn đọc góp ý.
Mọi góp ý xin liên hệ qua email: *ngocthe.pk@gmail.com*.

Xin chân thành cảm ơn!

Hoàng Ngọc Thế

LỜI NÓI ĐẦU

Trong hai thập kỷ qua, công nghệ đã không ngừng phát triển. Từ 3 trang web đầu tiên (*Microsoft*, *Netscape*, và *Amazon*) đến hơn 800 triệu trang web trong năm 1999 và trên 50 tỷ trang ngày nay, Internet đã tăng trưởng theo cấp số nhân. Ngay cả một lần lướt web thông thường trên Internet cũng đủ để thuyết phục bạn rằng có một số lượng lớn thông tin và liên kết trực tuyến có sẵn. Tuy nhiên, tất cả các thông tin này đều là vô ích nếu như ta không có một cách tìm kiếm và phân loại nó. Từ công cụ tìm kiếm đầu tiên là Archie vào năm 1990 cho đến các công cụ tìm kiếm hiện đại mà ta sử dụng ngày nay, sự liên kết của các thông tin trực tuyến có sẵn là một vấn đề rất quan trọng.



Một ảnh chụp của đồ thị Internet (Wikipedia)

Trong mô-đun này, ta cố gắng phân tích các vấn đề toán học đằng sau một trong những công cụ tìm kiếm phổ biến nhất: Google. Hai phần đầu tiên là khép kín và có thể được đọc một cách độc lập. Tuy nhiên hai phần tiếp theo sử dụng kiến thức được nói đến ở phần đầu.

1. Sơ lược về Đại số tuyến tính.
2. Đồ thị định hướng - Ma trận chuyển tiếp
3. Thuật toán PageRank - Toán học của Google Search
4. Thuật toán HITS - Hub và Trang có uy tín trên Internet
5. Toán Internet

6. Tài liệu tham khảo.

Công trình được tài trợ bởi *Quỹ Khoa học Quốc gia Hoa Kì*.

1 SƠ LƯỢC VỀ ĐẠI SỐ TUYẾN TÍNH

Ta bắt đầu bằng cách nhắc lại một số kiến thức liên quan đến đại số tuyến tính. Lý thuyết ở đây, tất nhiên không được nêu lên một cách tổng quát, nhưng đơn giản và thích hợp với cuộc thảo luận này.

Ký hiệu tập hợp của tất cả các số thực là $\mathbb{R}$.
Một **vector n chiều** là một danh sách có n phần tử, còn được gọi là **vector cột**. Ký hiệu vector v có các phần tử $x_1, x_2, ..., x_n$ như sau:

$$v = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Ví dụ 1. *Các phần tử của một vectơ có thể là bất cứ thứ gì: từ ngữ, màu sắc, số, ... Sau đây là những ví dụ về vectơ:*

$$\begin{bmatrix} Tôi \\ là \\ một \\ vector \end{bmatrix}; \quad \begin{bmatrix} Đỏ \\ Xanh\ dương \\ Xanh\ lá\ cây \end{bmatrix}; \quad \begin{bmatrix} 1.1 \\ 0 \\ 13.5 \end{bmatrix}; \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \quad \begin{bmatrix} X \\ O \\ X \\ O \\ X \end{bmatrix}.$$

Tuy nhiên, ta sẽ chỉ quan tâm đến các vector có phần tử là các số thực (có nghĩa là $x_1, x_2, ..., x_n \in \mathbb{R}$), chúng được gọi là **vector thực**. Đối với các vector thực, ta xét thêm hai phép toán: phép cộng và phép nhân vô hướng.

Phép cộng:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

Phép nhân vô hướng:

$$a. \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a.x_1 \\ a.x_2 \\ \vdots \\ a.x_n \end{bmatrix}$$

Ví dụ 2. *Theo ký hiệu hai ở trên, ta có thể dễ dàng kiểm tra:*

$$\begin{bmatrix} 1.14 \\ 5 \\ \pi \end{bmatrix} + \begin{bmatrix} 2 \\ -1.86 \\ 0 \end{bmatrix} = \begin{bmatrix} 3.14 \\ 3.14 \\ \pi \end{bmatrix}; \quad \pi. \begin{bmatrix} 1 \\ 5 \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \pi \\ 5\pi \\ \frac{\pi}{2} \end{bmatrix}$$

Bài toán 1. *Hãy trả lời các câu hỏi sau:*

1. *Liệu có thể nhân hai vectơ? Làm thế nào để cộng 2 vectơ có số chiều khác nhau?*
2. *Có bao nhiêu vectơ 5 chiều với phần tử là số nguyên? Nếu ta bổ sung điều kiện các phần tử đó còn phải khác nhau nữa thì sao?*

Theo cách tương tự, ta định nghĩa khái niệm ma trận. Một **ma trận** $m \times n$ là một bảng hình chữ nhật có chiều cao m và chiều rộng n. Điều này cũng có thể được xem như là xếp m vectơ n chiều liên tiếp (các vectơ có thể khác nhau hoặc không). Một lần nữa, các phần tử ở đây là các số thực. Ta biểu thị A là ma trận $m \times n$ với các phần tử a_{ij} như sau:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

Điều này có nghĩa là trên hàng i và cột j, ta tìm thấy số thực a_{ij}. Ví dụ, phần tử nằm trên hàng đầu tiên, cột thứ hai là a_{12}. Chú ý rằng một vector m chiều chỉ là một ma trận $m \times 1$ và một số thực sự là một ma trận 1×1.

Nếu $m = n$ thì ta có một **ma trận vuông**, kích thước n. Đây là những gì ta sẽ nghiên cứu từ bây giờ. **Đường chéo của một ma trận vuông** bao gồm những phần tử mà chỉ số hàng bằng chỉ số cột (các phần tử, $a_{11}, a_{22}, \dots, a_{nn}$).

Ví dụ 3. *Dưới đây là những ví dụ về ma trận, nhưng chỉ có ba ma trận vuông:*

$$\begin{bmatrix} 1 & 0 \\ 2 & -3 \\ 4 & 7 \end{bmatrix}; \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; \quad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \quad \begin{bmatrix} a & b \\ b & a \end{bmatrix}; \quad \begin{bmatrix} 5 & -1 & 11 \\ -1 & 0 & 3 \\ 11 & 3 & 9 \end{bmatrix}$$

Chú ý rằng hai ma trận cuối rất "đặc biệt": chúng là các **ma trận đối xứng**. Tức là phần tử trên hàng i và cột j bằng phần tử trên hàng j cột i (hay $a_{ij} = a_{ji}$). Ta dễ dàng chứng minh rằng bất kỳ **ma trận đường chéo** (các phần tử đều bằng 0, ngoại trừ trên đường chéo) là đối xứng. Ma trận đường chéo có đường chéo toàn phần tử 1 được gọi là **ma trận đơn vị** và được ký hiệu là I .

Phép nhân ma trận được thực hiện theo cách chung sau đây:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \dots & b_{ij} & \dots & b_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nj} & \dots & b_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1j} & \dots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & c_{i2} & \dots & \boxed{c_{ij}} & \dots & c_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nj} & \dots & c_{nn} \end{bmatrix}$$

Ở đó các phần tử c_{ij} được cho bởi công thức:

$$c_{ij} = \sum_{k=1}^{n} a_{ik}.b_{kj} = a_{i1}.b_{1j} + a_{i2}.b_{2j} + \dots + a_{in}.b_{nj}$$

Nhân một ma trận với một véc tơ được thực hiện như sau:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nj} & \dots & a_{nn} \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_j \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \boxed{y_i} \\ \vdots \\ y_n \end{bmatrix}$$

Ở đó mỗi phần tử y_i được cho bởi công thức

$$y_i = \sum_{k=1}^{n} a_{ik}.x_k = a_{i1}.x_1 + a_{i2}.x_2 + \dots + a_{in}.x_n$$

Hai phép toán ma trận khác là phép cộng và phép nhân vô hướng, được thực hiện như trong trường hợp của vectơ. Cộng hai ma trận A và B cho ta một ma trận C trong đó có các phần tử trên hàng i và cột j bằng tổng của các phần tử tương ứng của A và B. Nhân ma trận A với một số thực a là nhân mọi phần tử của A với a.

Ví dụ 4. *Tất cả các phép toán trên ma trận sẽ trở nên rõ ràng hơn thông qua các ví dụ sau đây:*

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix} = \begin{bmatrix} 15 \\ 12 \\ 22 \\ 16 \end{bmatrix} ; \qquad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} . \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 7 \\ 8 & 15 \end{bmatrix}$$

Bài toán 2. *Xem xét các ma trận sau đây:*

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} ; \quad B = \begin{bmatrix} \sqrt{3} & 1 & 0 \\ 0 & \sqrt{2} & 1 \\ 1 & 0 & 1 \end{bmatrix} ; \quad C = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \end{bmatrix}$$

1. *Tìm $A + B$, $B - C$ và $A - B$?*
2. *Chứng minh $(A + B) + C = A + (B + C)$ và $(A.B).C = A.(B.C)$.*
3. *Tìm tích của A với cột đầu tiên của C .* △

Định lý 1.
Với ba ma trận có cùng kích thước A , B , C bất kì, ta có:

$$(A+B).C = A.C + B.C; \qquad C.(A+B) = C.A + C.B$$

Dễ thấy trong bài toán 2 ở trên, phần tử trên hàng i và cột j trong ma trận A bằng phần tử trên hàng j và cột i trong ma trận C. Tất nhiên, các phần tử trên đường chéo của chúng bằng nhau. A được gọi là **ma trận chuyển vị** của C, hay C là ma trận chuyển vị của A. Đối với một ma trận A, ta kí hiệu chuyển vị của nó là A^t. Trực quan hơn, có thể tạo ra một ma trận chuyển vị của A bằng cách trao đổi các phần tử ở hàng i , cột j với phần tử tại hàng j, cột i. Nếu ta làm điều này hai lần ta nhận thấy rằng chuyển vị của ma trận chuyển vị là ma trận ban đầu, hay $(A^t)^t = A$.

Định lý 2.
Đối với 2 ma trận A , B bất kỳ, ta có:

$$(A.B)^t = B^t.A^t; \qquad (A+B)^t = A^t + B^t$$

Có hai khái niệm quan trọng trong lý thuyết ma trận. Đó là vectơ riêng và giá trị riêng.

Số thực z là được gọi là một **giá trị riêng** của A nếu tồn tại một vector thực n chiều v mà

$$A.v = z.v$$

Khi đó vector v được gọi là **vectơ riêng** tương ứng với giá trị riêng z. Đây không phải là định nghĩa tổng quát, nhưng đủ cho mục đích của tài liệu này. Nếu A là một ma trận thực đối xứng có kích thước n, khi đó nó có n giá trị riêng thực và tất cả các vectơ riêng tưng ứng cũng thực. Trong thực tế, một ma trận kích thước n có thể có nhiều nhất là n giá trị riêng thực.

Ví dụ 5. *Ma trận $A = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}$ có vectơ riêng $v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, giá trị riêng $z = 3$ tương ứng với vectơ riêng này.*
Thật vậy:

$$\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} = 3. \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Định lý 3.
Nếu z là 1 giá trị riêng của ma trận A thì nó cũng là giá trị riêng của A^t và ngược lại.□

Với cùng một giá trị riêng, nói chung, vector riêng của A và vector riêng của A^t là khác nhau. Đối với các ma trận trong ví dụ 5, $A^t = \begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}$ cũng có giá trị riêng $z = 3$

nhưng vectơ riêng tương ứng là $u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Điều này có được nhờ tính toán dưới đây

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

Trong nhiều trường hợp, một ma trận A có nhiều hơn một vectơ riêng tương ứng với 1 giá trị riêng. Các vectơ riêng có tương ứng với cùng một giá trị riêng có thể không có quan hệ với nhau. Tuy nhiên chúng có thể có liên quan, ví dụ như vectơ này là tích của một số với một vectơ khác. Chính xác hơn, trong ví dụ cuối cùng, vector có các phần tử là 0 và 1 là một vectơ riêng, mà vector có phần tử là 0 và 2 cũng là một vectơ riêng.

Ma trận A được gọi là **ma trận cột - ngẫu nhiên** nếu tất cả các phần tử của nó không âm và tổng của các phần tử trong mỗi cột bằng 1. Nếu tất cả các phần tử của một ma trận không âm, ta gọi nó là **ma trận không âm**. Hơn nữa, **ma trận dương** là ma trận mà tất cả các phần tử của nó đều là số thực dương.

Ví dụ 6. *Xét một ma trận A với chuyển vị A^t :*

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}, \qquad A^t = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

Dễ thấy A là ma trận cột - ngẫu nhiên, trong khi A^t thì không. Tuy nhiên, tổng của các phần tử trên mỗi hàng của A^t là bằng 1.

Ta thấy $z = 1$ là một giá trị riêng của A^t, tương ứng với vectơ riêng $v = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ vì $A^t.v = 1.v$.

Do đó, theo định lý 3, $z = 1$ cũng là một giá trị riêng của ma trận A, tương ứng với vector riêng $u = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$. Để tìm u, ta giải phương trình:

$$A.u = 1.u \quad \Leftrightarrow \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = 1. \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Thực hiện phép nhân, ta có:

$$\begin{cases} x_1 = 1x_3 + \frac{1}{2}x_4 \\ x_2 = \frac{1}{3}x_1 \\ x_3 = \frac{1}{3}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_4 \\ x_4 = \frac{1}{3}x_1 + \frac{1}{2}x_2 \end{cases} \Leftrightarrow \begin{cases} x_1 = x_3 + \frac{1}{2}x_4 \\ x_2 = \frac{1}{3}x_1 \\ x_3 = \frac{1}{2}x_1 + \frac{1}{2}x_4 \\ x_4 = \frac{1}{2}x_1 \end{cases} \quad (\textit{thế biến } x_2)$$

Lại thế $x_4 = \frac{1}{2}x_1$ trong phương trình thứ ba ta được $x_3 = \frac{3}{4}x_1$ và do đó:

$$u = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ \frac{1}{3}.x_1 \\ \frac{3}{4}.x_1 \\ \frac{1}{2}.x_1 \end{bmatrix} = x_1. \begin{bmatrix} 1 \\ \frac{1}{3} \\ \frac{3}{4} \\ \frac{1}{2} \end{bmatrix} = \frac{x_1}{12}. \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix}$$

Vì x_1 chỉ là một số thực (do đó là một vô hướng), ta có thể lấy $x_1 = 12$. Vậy vector $\begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix}$ là một vectơ riêng của A . △

Trong phần đầu tiên của ví dụ 6, ta thấy số 1 là một giá trị riêng cho trường hợp cụ thể. Tuy nhiên, điều này vẫn đúng cho ma trận cột - ngẫu nhiên bất kì, như định lý dưới đây.

Định lý 4.
Mỗi ma trận cột - ngẫu nhiên bất kì đều nhận $z = 1$ là giá trị riêng. □

Chú ý rằng vectơ riêng trong phần thứ hai của ví dụ 6 là khá "đặc biệt". Ta đã chọn $x_1 = 12$ và có được một vectơ riêng với phần tử dương. Tuy nhiên, nếu chọn $x_1 = -12$, ta có được một vectơ riêng với các số âm.

Định lý 5.
Với mọi ma trận cột - ngẫu nhiên dương, khi đó vectơ riêng tương ứng với giá trị riêng $z = 1$ hoặc chỉ có phần tử dương hoặc chỉ có phần tử âm. □

Bài toán 3. *Dễ thấy ma trận* $\begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{bmatrix}$ *là một ma trận - cột ngẫu nhiên và có $z = 1$ là giá trị riêng. Tìm một vectơ riêng tương ứng với giá trị riêng đó mà tất cả các phần tử của nó đều dương.*

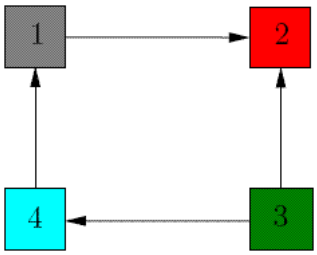
Khi làm việc với các ma trận cột - ngẫu nhiên dương, có thể tìm thấy một vectơ riêng v tương ứng với giá trị riêng $z = 1$ mà tất cả các phần tử của nó đều dương. Do đó tồn tại v sao cho $A.v = v$ và v có mọi phần tử đều dương.

Định lý 6.
Nếu A là một ma trận cột - ngẫu nhiên dương, khi đó có duy nhất một vectơ riêng tương ứng với giá trị riêng $z = 1$ sao cho vectơ đó có các phần tử đều dương và tổng các phần tử của nó bằng 1. □

2 ĐỒ THỊ CÓ HƯỚNG - MA TRẬN CHUYỂN TIẾP

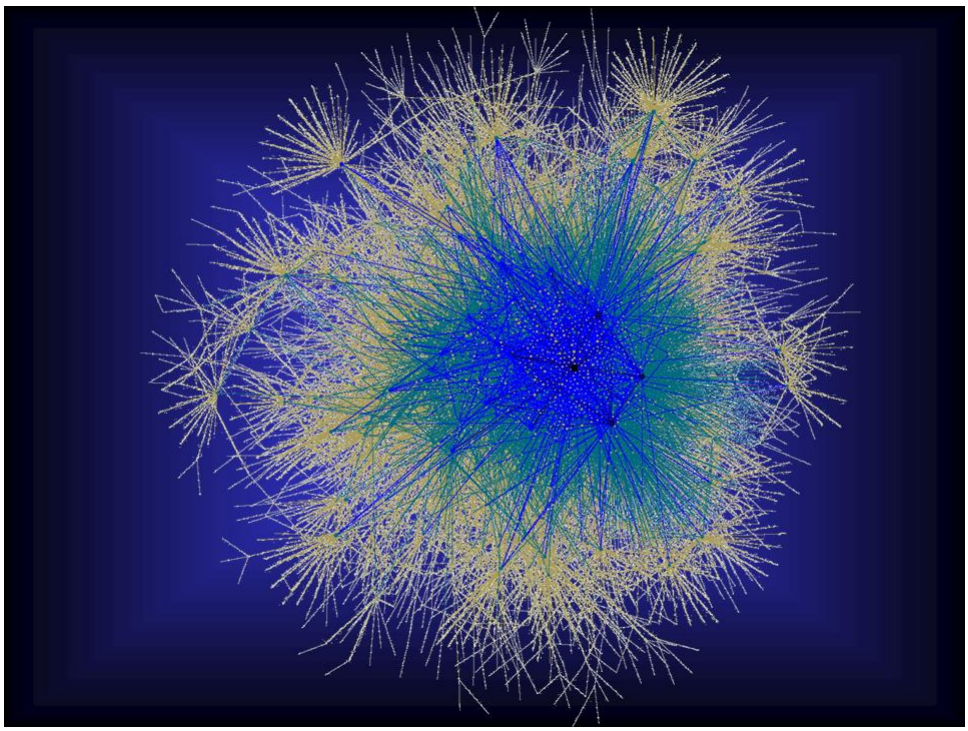
Một **đồ thị** là đối tượng bao gồm một tập hợp không rỗng các đỉnh và tập hợp các cạnh. Các đỉnh thường được gọi là các **nút** hoặc các **điểm**, còn cạnh được gọi là các **liên kết** (links) hoặc **đường** (lines). Tập hợp các cạnh có thể rỗng, khi đó đồ thị là một tập hợp các đỉnh.

Ví dụ 7. *Xét đồ thị với các đỉnh* $1, 2, 3, 4$ *và các cạnh từ đỉnh* 1 *đến đỉnh* 2*; đỉnh* 3 *đến đỉnh* $2, 4$*; và đỉnh* 4 *đến đỉnh* 1*.*



△

Trong phần này, ta sẽ chỉ làm việc với **đồ thị có hướng** và các ví dụ thực tế của chúng (đồ thị Internet), còn đối với các kiểu đồ thị khác, bạn hãy tham khảo trên trang web của Câu lạc bộ Toán Explorer. Ví dụ trung tâm trong mô-đun này là **đồ thị web**, trong đó các trang web được biểu diễn bằng các đỉnh và các liên kết giữa chúng được biểu diễn bằng các cạnh. Chẳng hạn một đồ thị con của đồ thị BGP (Gateway Protocol), bao gồm các bộ định tuyến Internet lớn, có khoảng 6.400 đỉnh và 13.000 cạnh. Nó được nêu ra bởi Ross Richardson và nhắc lại bởi Graham Fan Chung.



Mặc dù đồ thị Internet rất lớn, có số đỉnh khoảng 30 tỉ (hoặc nhiều hơn), nhưng tất cả các đồ thị trong mô-đun này được coi là hữu hạn (hữu hạn số lượng đỉnh và cạnh).

Ta nói rằng hai đỉnh i và j của một đồ thị có hướng là **kề nhau** nếu có một cạnh

từ i tới j hoặc từ j và i. Nếu một cạnh như vậy tồn tại thì i và j được gọi là các **điểm đầu cuối** của cạnh đó. Nếu có một cạnh từ i đến j thì j thường được gọi là **điểm cuối**, trong khi i được gọi là **điểm đầu**.

Trong Ví dụ 7, đỉnh 1 và 2 là kề nhau bởi vì có một cạnh đi từ 1 đến 2, trong khi đỉnh 1 và 3 không kề nhau. Tuy nhiên, không có cạnh từ đỉnh 2 tới đỉnh 1. Chú ý rằng có thể có không nhiều hơn hai cạnh giữa hai đỉnh bất kỳ.

Có một mối liên hệ giữa đồ thị và ma trận. Xét một đồ thị có hướng n đỉnh. Khi đó, ta xây dựng một ma trận A cấp $n \times n$ gọi là **ma trận kề** của đồ thị như sau: nếu có một cạnh từ đỉnh i đến đỉnh j, thì ta đặt số 1 vào vị trí hàng i, cột j của ma trận.

Ví dụ 8. *Ma trận liên kết của đồ thị trong Ví dụ 7:*

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

Nếu có thể đi từ đỉnh i đến đỉnh j dọc theo các cạnh của đồ thị thì ta nói rằng có một **đường đi** từ i tới j. Nếu ta đi trên k cạnh, thì đường đi có **chiều dài** k.

Đối với mỗi ma trận, ta kí hiệu A^k là ma trận thu được bằng cách nhân A với chính nó k lần. Phần tử trên hàng i, cột j của $A^2 = A.A$ tương ứng với số lượng các đường đi có độ dài 2 từ đỉnh i đến đỉnh j trong đồ thị.

Trong Ví dụ 8, ta có bình phương ma trận liên kết.

$$A^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Điều này có nghĩa là có một đường đi độ dài 2 từ đỉnh 4 đến đỉnh 2, bởi vì phần tử ở vào hàng thứ tư, cột thứ hai là 1. Tương tự như vậy, có một đường đi từ đỉnh 3 đến đỉnh 1 độ dài 2, ta có thể dễ dàng nhìn thấy từ Ví dụ 7.

Định lý 7.
Cho một đồ thị có hướng và k là số nguyên dương. Khi đó, số đường đi từ đỉnh i đến đỉnh j có chiều dài k là phần tử trên hàng i và cột j của ma trận A^k, với A là ma trận kề. □

Một ma trận được gọi là **ma trận nguyên thủy** nếu có một số nguyên dương k sao cho A^k là một ma trận dương. Một đồ thị được gọi là **liên thông** nếu với hai đỉnh bất kì i và j, có một đường đi hoặc từ i tới j hoặc từ j tới i. Mặt khác, một đồ thị được gọi là **liên thông mạnh**, nếu bắt đầu từ đỉnh i bất kỳ, ta có thể đến được đỉnh j bất kỳ khác i bằng cách đi trên các cạnh của nó.

Trong ngôn ngữ ma trận, điều này có nghĩa là nếu tồn tại một số nguyên dương k

sao cho ma trận

$$B = I + A + A^2 + A^3 + ... + A^k$$

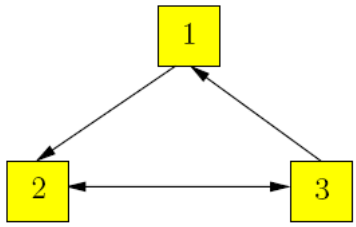
dương thì đồ thị là đồ thị liên thông mạnh. Ta thêm ma trận đơn vị I để chỉ các cạnh từ một đỉnh đến chính nó. Nói cách khác, nếu tồn tại ít nhất một đường đi từ đỉnh i đến đỉnh j có độ dài k, thì ta có thể đi từ đỉnh i đến đỉnh j. Do đó, nếu ma trận B có một phần tử dương trên hàng i và cột j, thì có thể đi đến đỉnh j bắt đầu từ đỉnh i. Nếu điều này xảy ra cho tất cả các đỉnh của đồ thị thì đồ thị là đồ thị liên thông mạnh.

Dễ thấy rằng đồ thị trong Ví dụ 7 được liên thông, nhưng không liên thông mạnh vì không có cạnh từ đỉnh 1 đến đỉnh 3. Đối với các ma trận trong Ví dụ 8, ta nhận thấy rằng A^4 là một ma trận chỉ có số không, và vì thế với mọi $k \geq 4$, A^k sẽ là một ma trận toàn số không. Do đó cho bất kỳ $k \geq 4$, ma trận $B = I + A + A^2 + A^3 + ... + A^k$ là:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 \end{bmatrix}$$

Do ma trận B không dương nên đồ thị ở Ví dụ 7 không liên kết mạnh như ta đã thấy.

Bài toán 4. *Xây dựng ma trận kề của đồ thị dưới đây. Dồ thị này có liên thông không? Có liên thông mạnh không? Có bao nhiêu đường đi độ dài 3 từ đỉnh 3 đến đỉnh 2?*

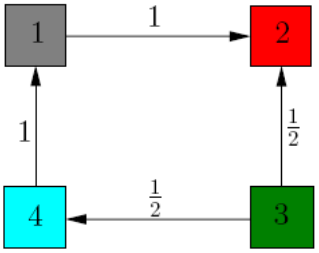


△

Trong ví dụ trên, ta nhận thấy rằng đối với mỗi đỉnh i có một số cạnh nối tới(tức là i là một điểm cuối) và một số cạnh xuất phát từ đỉnh này (i là điểm đầu). Vì vậy, ta định nghĩa **mật độ nhập** (indegree) đỉnh i là số cạnh mà i là một điểm cuối. Tương tự như vậy, **mật độ xuất** (outdegree) đỉnh i là số cạnh mà i là một điểm đầu. Chẳng hạn, đồ thị trong Bài tập 4, mật độ nhập của đỉnh 2 là 2 và các mật độ xuất đỉnh 1 là 1.

Ma trận chuyển tiếp A của một đồ thị có hướng được định nghĩa như sau. Nếu có một cạnh từ i đến j và mật độ xuất của đỉnh i là d_i thì trên cột i và hàng j ta đặt số $\frac{1}{d_i}$. Nếu không, ta đặt ở cột i, hàng j với số không. Chú ý rằng đầu tiên là với cột, sau đó với hàng. Ta thường viết $\frac{1}{d_i}$ trên các cạnh đi từ đỉnh i đến đỉnh j liền kề, do đó có được một **đồ thị có trọng số**. Điều này sẽ trở nên rõ ràng thông qua ví dụ sau đây.

Ví dụ 9. *Xem xét đồ thị từ Ví dụ 7 với trọng số trên các cạnh của nó như mô tả ở trên.*



△

Khi đó, ma trận chuyển tiếp của nó là:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

Chú ý rằng tổng của các phần tử trên cột đầu tiên là 1. Tương tự cho cột thứ ba và thứ tư. Tổng quát, ta có định lý.

Định lý 8.
Ma trận chuyển tiếp của đồ thị liên thông mạnh là ma trận cột - ngẫu nhiên. □

Ta sử dụng các ma trận chuyển tiếp mô tả hành vi của một người lướt web ngẫu nhiên trên một đồ thị web. Lướt chọn một trang ngẫu nhiên, sau đó theo liên kết đến các trang web khác miễn là anh ta muốn. Ở mỗi bước, xác suất di chuyển từ đỉnh i đến đỉnh j là số không nếu không có liên kết từ i tới j và là $\frac{1}{d_i}$ nếu có liên kết. Nhớ lại rằng d_i là mật độ xuất đỉnh i. Ban đầu, xác suất của mỗi trang được lựa chọn như điểm khởi đầu là

$$v = \frac{1}{n}. \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Tại bước 1, xác suất của mỗi đỉnh được truy cập sau một cú nhấp chuột là Av. Tại bước 2, xác suất của mỗi đỉnh được truy cập sau khi hai lần nhấp chuột là A^2v. Xác suất của một trang được truy cập tại bước k là A^kv. Nếu ma trận A là ma trận nguyên thủy, cột - ngẫu nhiên, thì quá trình này hội tụ về một phân phối dừng duy nhất: vector p ,

$$p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}$$

Ý nghĩa của phần tử thứ i của p là người lướt thăm trang i tại bất kỳ thời gian nào với xác suất p_i.

Bài toán 5. *Cho A là ma trận chuyển tiếp của một đồ thị, B là một ma trận kích thước n chứa toàn $\frac{1}{n}$ và x là một số thực dương nhỏ hơn 1. Khi đó ma trận $C = x.A+(1-x).B$ là ma trận cột - ngẫu nhiên.*
Chứng minh rằng điều này là đúng trong trường hợp đặc biệt cụ thể: bài tập 4.

3 THUẬT TOÁN PageRank - TOÁN HỌC CỦA Google Search

Chúng ta đang sống trong kỷ nguyên máy tính. Internet trở thành một phần trong cuộc sống hàng ngày của chúng ta và việc tìm kiếm thông tin chỉ thông qua một cú nhấp chuột. Chỉ cần mở công cụ tìm kiếm ưa thích của bạn, như *Google*, *Altavista*, *Yahoo*, ... gõ từ khóa, các công cụ tìm kiếm sẽ hiển thị các trang có liên quan cho tìm kiếm của bạn. Nhưng thực sự một công cụ tìm kiếm làm việc như thế nào?

Thoạt nhìn, có vẻ như hợp lý khi tưởng tượng rằng những gì một công cụ tìm kiếm làm là giữ một chỉ số của tất cả các trang web, khi một người dùng trong một truy vấn tìm kiếm, công cụ duyệt thông qua chỉ số của nó và đếm số lần xuất hiện của các từ khóa mỗi tập tin web. Chiến thắng dành cho các trang có số lượng xuất hiện các từ khóa cao nhất. Những kết qủa nhận được hiển thị lại cho người dùng.

Điều này chỉ đúng cho những năm 90 của thế kỉ XX, khi các công cụ tìm kiếm đầu tiên được sử dụng **Hệ thống xếp hạng theo văn bản** để quyết định những trang nào là phù hợp nhất với một truy vấn nhất định. Tuy nhiên có một số vấn đề với cách tiếp cận này. Chẳng hạn, một tìm kiếm về một thuật ngữ phổ biến như "*Internet*". Trang đầu tiên được hiển thị bởi một trong những công cụ tìm kiếm ban đầu được viết bằng tiếng Trung, với sự xuất hiện lặp đi lặp lại từ "*Internet*" và không chứa các thông tin khác về Internet. Hơn nữa, giả sử ta muốn tìm thấy một số thông tin về Cornell. Ta gõ vào từ "*Cornell*" và hy vọng rằng "*www.cornell.edu*" sẽ là trang web có liên quan đến truy vấn này. Tuy nhiên có thể là hàng triệu trang web khác bằng cách sử dụng nhiều từ Cornell, còn *www.cornell.edu* có thể không được sử dụng từ đó thường xuyên. Giả sử ta quyết định viết một trang web có chứa từ "*Cornell*" một tỷ lần mà không có gì khác. Thì liệu trang web đó có là một trong những trang đầu tiên được hiển thị bằng một công cụ tìm kiếm hay không? Câu trả lời rõ ràng là không. Tuy nhiên, nếu tất cả các công cụ tìm kiếm không đếm số lần xuất hiện của các từ trong truy vấn thì chúng tìm kiếm như thế nào?

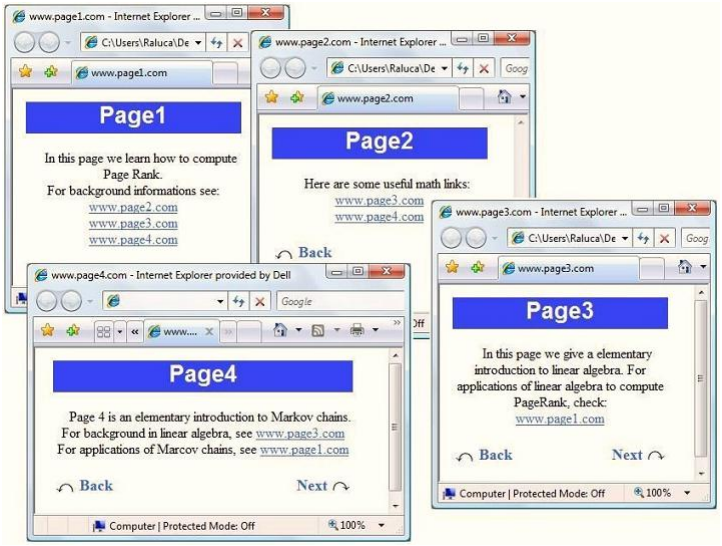
Tính hữu ích của một công cụ tìm kiếm phụ thuộc vào sự thích hợp của tập hợp các kết quả mà nó mang lại. Tất nhiên có thể có là hàng triệu các trang web bao gồm một từ hoặc cụm từ cụ thể, tuy nhiên một số trong chúng sẽ có liên quan, phổ biến, hoặc chuyên sâu hơn, uy tín hơn những trang khác. Một người sử dụng không có khả năng kiên nhẫn để quét qua tất cả các trang có chứa các từ truy vấn. Ta luôn hy vọng các trang có liên quan được hiển thị trong danh sách 20-30 trang đầu được trả lại bởi các công cụ tìm kiếm.

Các công cụ tìm kiếm hiện đại sử dụng các phương pháp xếp hạng các kết quả để cung cấp kết quả "tốt nhất" đầu tiên. Nó phức tạp hơn so với phương pháp xếp hạng theo văn bản đơn giản. Một trong những thuật toán nổi tiếng và có ảnh hưởng nhất cho việc tính toán sự phù hợp của trang web là Thuật toán xếp hạng trang (**PageRank algorithm**) được sử dụng bởi công cụ tìm kiếm *Google*. Nó được phát minh bởi *Larry Page* và *Sergey Brin* trong khi họ còn là học viên sau đại học tại Stanford, và nó đã trở thành một thương hiệu của Google trong năm 1998. Ý tưởng *PageRank* được đưa ra là: tầm quan trọng của bất kỳ trang web có thể được đánh giá bằng cách nhìn vào các trang liên kết đến nó. Nếu ta tạo ra một trang web i và bao gồm một siêu liên kết đến trang web j, điều này có nghĩa là ta xem j quan trọng và có liên quan cho chủ đề của ta. Nếu có rất nhiều các trang liên kết đến j, điều này có nghĩa là có niềm tin phổ biến rằng

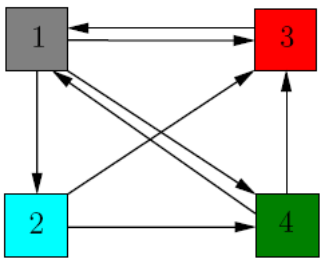
trang j là quan trọng. Nếu mặt khác, j có chỉ có một liên kết đến, nhưng mà đến từ một trang web k có uy tín (như *www.google.com*, *www.cnn.com*, *www.cornell.edu*, ...) chúng ta nói rằng k chuyển giao uy tín của mình cho j , nói cách khác, k khẳng định rằng j là quan trọng. Cho dù chúng ta nói về trang phổ biến hay trang có uy tín, ta nhắc lại rằng có thể chỉ định một cấp bậc cho mỗi trang web, dựa trên cấp bậc của các trang trỏ đến nó.

Để thực hiện mục tiêu này, ta bắt đầu bằng hình dung mạng Web như là một đồ thị có hướng, với các đỉnh biểu diễn các trang web và các cạnh biểu diễn liên kết giữa chúng.

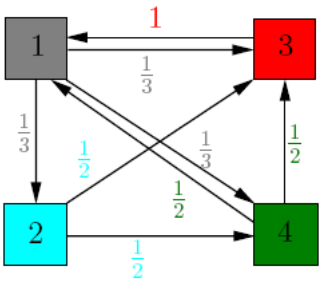
Giả sử có một mạng Internet nhỏ chỉ gồm 4 trang web *www.page1.com*, *www.page2.com*, *www.page3.com*, *www.page4.com*, sự liên kết giữa chúng được thể hệ như dưới đây:



Chúng ta "chuyển" hình ảnh vào một đồ thị định hướng với 4 đỉnh, mỗi đỉnh cho một trang web. Khi trang web i liên kết đến trang j, ta thêm 1 cạnh định hướng giữa đỉnh i và đỉnh j trong đồ thị. Với mục đích là tính toán thứ hạng trang của chúng, ta bỏ qua bất kỳ liên kết điều hướng như Back, Next, chỉ quan tâm đến các kết nối giữa các trang web khác nhau. Chẳng hạn, *Page1* liên kết đến tất cả các trang khác, do vậy, đỉnh 1 trong đồ thị sẽ có cạnh đi cho tất cả các đỉnh khác. *Page3* có chỉ có một liên kết đến *Page1*, do đó đỉnh 3 sẽ có một cạnh đi vào đỉnh 1. Sau khi phân tích mỗi trang web, ta nhận được đồ thị dưới đây:



Trong mô hình này, mỗi trang cần chuyển đều **giá trị** của nó đến các trang mà nó liên kết đến. Đỉnh 1 có 3 cạnh đi, do đó, nó sẽ chuyển giá trị của nó với 3 đỉnh khác. Đỉnh 3 đã chỉ có một đi cạnh, do đó, nó sẽ chuyển tất cả giá trị tới đỉnh 1. Nói chung, nếu một đỉnh có k cạnh đi, nó sẽ chuyển giá trị của nó cho mỗi đỉnh mà nó liên kết đến. Hãy hình dung rõ hơn quá trình bằng cách chỉ định trọng số trên mỗi cạnh.



Ta có ma trận chuyển tiếp của đồ thị:

$$A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Xét theo hệ động lực

Giả sử rằng ban đầu giá trị được phân bố đồng đều giữa 4 đỉnh, mỗi đỉnh nhận được $\frac{1}{4}$. Ký hiệu v là vector xếp hạng ban đầu, có tất cả các phần tử bằng $\frac{1}{4}$. Mỗi liên kết đến làm tăng giá trị của một trang web, do đó, tại bước 1, chúng ta cập nhật thứ hạng của mỗi trang bằng cách thêm giá trị các liên kết đến vào các giá trị hiện tại. Điều này giống như nhân ma trận A với v. Sau bước 1, các vector giá trị mới là $v_1 = A.v$. Chúng ta có thể lặp lại quá trình, do đó sau bước 2, vector giá trị là $v_2 = A.(Av) = A^2.v$. Ta có:

$$v = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}; \quad A.v = \begin{bmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{bmatrix}; \quad A^2v = \begin{bmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{bmatrix}; \quad A^3v = \begin{bmatrix} 0.35 \\ 0.14 \\ 0.29 \\ 0.20 \end{bmatrix}$$

$$A^4v = \begin{bmatrix} 0.39 \\ 0.11 \\ 0.29 \\ 0.19 \end{bmatrix}; \quad A^5v = \begin{bmatrix} 0.39 \\ 0.13 \\ 0.28 \\ 0.19 \end{bmatrix}; \quad A^6v = \begin{bmatrix} 0.38 \\ 0.13 \\ 0.29 \\ 0.19 \end{bmatrix}; \quad A^7V = \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix}; \quad A^8v = \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix}$$

Dễ thấy rằng dãy $v, Av, ..., A^kv$ hội tụ đến $v^* = \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix}$. Vector $v*$ được gọi là **vector PageRank** của đồ thị web ở trên.

Xét theo đại số tuyến tính

Gọi x_1, x_2, x_3, x_4 tương ứng là giá trị của bốn trang. Phân tích tình hình tại mỗi đỉnh, chúng ta có được hệ:

$$\begin{cases} x_1 = & & 1.x_3 & +\frac{1}{2}x_4 \\ x_2 = & \frac{1}{3}x_1 & & \\ x_3 = & \frac{1}{3}x_1 & +\frac{1}{2}x_2 & +\frac{1}{2}x_4 \\ x_4 = & \frac{1}{3}x_1 & +\frac{1}{2}x_2 & \end{cases}$$

Điều này tương đương với yêu cầu giải hệ phương trình.

$$A.\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Từ Ví dụ 6, ta biết rằng các vector riêng tương ứng các giá trị riêng 1 có dạng.

$$c.\begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix}$$

Do PageRank chỉ phản ánh giá trị tương đối của các đỉnh, và khi các vector riêng là bội vô hướng của nhau, ta có thể chọn vector bất kỳ trong số chúng là vector PageRank. Chọn v^* là vector riêng duy nhất với tổng của tất cả các phần tử bằng 1. (Gọi là vector riêng xác suất tương ứng với giá trị riêng 1). Vector riêng

$$\frac{1}{31}.\begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix} \sim \begin{bmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{bmatrix}$$

là véc tơ PageRank cần tìm.

Xét theo quan điểm Xác suất

Vì giá trị của một trang web được đo bằng tính phổ biến của nó (nó có bao nhiêu liên kết đến), ta có thể xem giá trị của trang i là xác suất mà một người lướt ngẫu nhiên trên Internet mở ra trang bất kì và sau các siêu liên kết, sẽ ghé thăm i. Có thể giải thích trọng số được gán cho các cạnh của đồ thị một cách xác suất: Một người lướt ngẫu nhiên mà hiện tại đang xem *Page2*, có $\frac{1}{2}$ xác suất để đi đến *Page3*, và $\frac{1}{2}$ xác suất để đi đến *Page4*. Ta có thể mô hình hóa các quá trình như là một bước đi ngẫu nhiên trên đồ thị. Mỗi trang có xác suất bằng $\frac{1}{4}$ được chọn là điểm khởi đầu. Vì vậy, phân bố xác suất ban đầu được đưa ra bởi các vector cột

$$x = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}.$$

Xác suất trang i sẽ được truy cập sau một bước bằng Ax, và cứ như vậy. Xác suất trang i sẽ được truy cập sau khi bước k bằng A^kx. Dãy $Ax, A^2x, A^3x, ..., A^kx, ...$ hội tụ về một vector xác suất duy nhất v^*. Khi đó v^* được gọi là *phân phối dừng* và nó sẽ là vector RankPage của i. Hơn nữa, phần tử thứ i trong v^* chỉ đơn giản là xác suất tại từng thời điểm mỗi một người lướt ngẫu nhiên thăm trang i. Các tính toán là giống hệt nhau với gì ta đã làm trong việc giải thích các hệ động lực, chỉ có ý nghĩa chúng ta gán cho mỗi bước là hơi khác nhau.

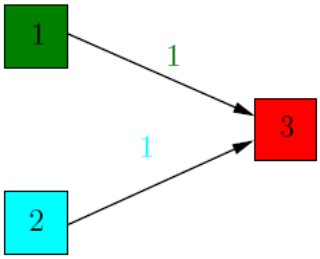
Quay lại vấn đề

Tính toán vector PageRank v^* bằng các phương pháp khác nhau, ta đã chỉ ra rằng *Page1* là trang có nhiều mối liên hệ nhất. Đây là kết quả đáng ngạc nhiên khi *Page1* có 2 liên

kết đến, trong khi *Page3* có 3 liên kết đến. Nếu nhìn vào đồ thị, ta thấy rằng đỉnh 3 chỉ có một cạnh đi tới đỉnh 1, do đó, nó chuyển tất cả giá trị của nó đến đỉnh 1. Điều này nghĩa là một người lướt web bất kì, sau khi thăm *Page3*, anh ta chỉ có thể đi đến *Page1*. Kết quả xếp hạng của mỗi trang không phụ thuộc vào tổng trọng số của các cạnh đến đỉnh tương ứng. Theo trực giác, tại bước 1, một đỉnh nhận được một giá trị từ các đỉnh kề trực tiếp của nó, ở bước 2, từ những đỉnh kề của đỉnh kề trực tiếp của nó, và cứ như vậy.

Thay đổi sơ đồ web có thể dẫn đến một số vấn đề nhất định.

Các đỉnh với không có đường đi đi (đỉnh treo)



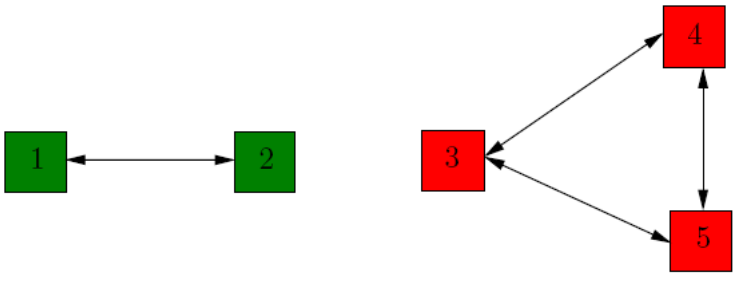
Ta lặp lại tính toán thứ hạng của 3 trang:

$$v_0 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}; \quad v_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix}; \quad v_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Vì vậy, trong trường hợp này, thứ hạng của mỗi trang đều là 0. Điều này vô lý, *Page3* có 2 liên kết đến, vì vậy nó phải có giá trị khác 0.

Dễ dàng khắc phục vấn đề này bằng cách thay thế các cột tương ứng với đỉnh 3 bằng một vector cột có toàn phần tử $\frac{1}{3}$. Bằng cách này, giá trị của đỉnh 3 sẽ được chia đều giữa cho các đỉnh khác của đồ thị, thay vì bị mất.

Các thành phần không liên thông



Một người lướt ngẫu nhiên không có cách nào đến được đỉnh 5 từ các đỉnh 1 và 2 (vì không có liên kết tới đỉnh 5). Đại số tuyến tính không giúp gì tốt hơn. Ma trận chuyển tiếp cho đồ thị này là

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

Chú ý rằng cả hai vector riêng

$$u = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; \quad v = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

tương ứng với các giá trị riêng $z = 1$, và chúng không **phụ thuộc tuyến tính** (không phải là kết quả của nhau qua một phép nhân với đại lượng vô hướng). Vì vậy, cả về lý thuyết và trong thực tế, kết quả xếp hạng các trang thuộc nhóm màu xanh mà dựa vào những yếu tố từ nhóm màu đỏ là không có giá trị.

Hiển nhiên, các trang web không đồng nhất về bản chất và rất lớn về số lượng, vì vậy ta không mong đợi đồ thị của tất cả chúng đều liên thông. Tương tự như trên, sẽ có các trang không chứa các liên kết đi. Cần làm gì trong trường hợp này? Ta cần 1 cách xếp hạng trang web cho bất kỳ đồ thị định hướng với n đỉnh.

Phương pháp của Page và Brin:
Để khắc phục những vấn đề trên, lấy một số dương p nằm giữa 0 và 1, gọi là *hệ số tắt dần* (một giá trị tiêu biểu cho p là 0,15). Ma trận

$$M = (1 - p).A + p.B$$

được gọi là **ma trận PageRank** (hay **ma trận Google**) của đồ thị. Trong đó:

$$B = \frac{1}{n} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

Bài toán 6. *Chứng minh rằng:*

1. *M là một ma trận cột - ngẫu nhiên.*
2. *M chỉ có một số phần tử dương.* △

Ma trận M thể hiện mô hình lướt mạng ngẫu nhiên như sau: ở hầu hết thời gian, một người lướt mạng sẽ đi theo các liên kết. Từ trang i sẽ theo các liên kết đi và chuyển đến một trong những trang lân cận của i. Một tỷ lệ nhỏ hơn, nhưng chiếm tương đối thời gian, anh ta sẽ bỏ trang hiện tại và chọn ngẫu nhiên một trang khác và chuyển đến đó. Hệ số tắt dần p phản ánh khả năng mà người lướt bỏ trang hiện tại và chuyển trang mới. Do khi anh ta có thể dịch chuyển đến bất kỳ trang web nào nên mỗi trang có xác suất được chọn là $\frac{1}{n}$. Điều này giải thích cho cấu trúc của ma trận B.

Bài toán 7. *Bằng cách thay ma trận chuyển tiếp A bằng ma trận M, hãy Làm lại các tính toán PageRank cho đồ thị có đỉnh treo và đồ thị có các thành phần không liên thông. Khi đó những vấn đề đã được đề cập trong đó còn xảy ra nữa không?* △

Theo trực giác, ma trận M "kết nối được" các đồ thị và "giải thoát" các đỉnh treo.

Một đỉnh không có cạnh đi vẫn có xác suất để di chuyển đến bất kỳ đỉnh khác. Chặt chẽ hơn, ta có định lý sau:

Định lý 9 (Định lý Perron-Frobenius).
Nếu M là một ma trận cột - ngẫu nhiên dương, khi đó:

- 1 *là giá trị riêng của nhiều vector riêng nhất.*
- 1 *là giá trị riêng lớn nhất: tất cả các giá trị riêng khác nhỏ hơn 1.*
- *Vector riêng tương ứng với giá trị riêng* 1 *có tất cả các phần tử đều dương. Đặc biệt, với giá trị riêng* 1*, tồn tại một vector riêng duy nhất có tổng của các phần tử của nó bằng 1.* □

Định lý 10 (Định lý Phương pháp hội tụ lũy thừa).
Cho M là ma trận cột - ngẫu nhiên dương cấp $n \times n$.
Ký hiệu v^* là vector riêng xác suất của nó tương ứng với giá trị riêng 1, z là các vector cột với tất cả các phần tử bằng $\frac{1}{n}$. Khi đó, dãy

$$z, Mz, ..., M^k z$$

hội tụ tới v^*. □

Từ các lập luận ở trên, ta kết luận rằng:

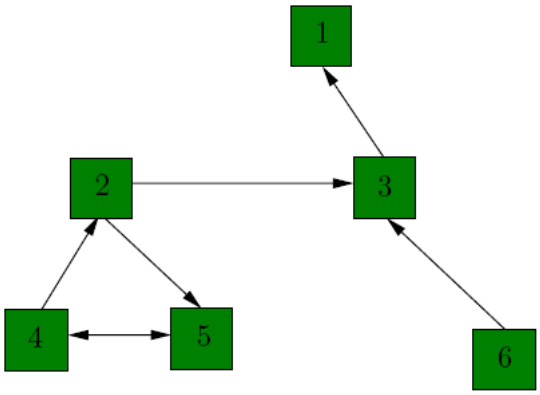
Định lý 11.
Vector PageRank của đồ thị web với ma trận chuyển tiếp A, và hệ số tắt dần p, là vector riêng xác suất duy nhất của ma trận M, ứng với giá trị riêng 1. □

Theo quan điểm toán học, khi có M, việc tính toán các vector riêng tương ứng với giá trị riêng 1, theo lý thuyết, là một nhiệm vụ dễ dàng. Theo phần 1, ta chỉ cần giải hệ $Ax = x$. Tuy nhiên, khi M ma trận có kích thước 30.000.000.000 (vì nó đại diện cho hệ thống Web thực sự) thì thậm chí cả các phần mềm toán học như Matlab hoặc Mathematica rõ ràng cũng quá tải.

Một cách thay thế tính toán vector riêng xác suất tương ứng với giá trị riêng 1 là **Phương pháp lũy thừa**. Định lý 10 chỉ ra rằng phương pháp này áp dụng cho ma trận cột - ngẫu nhiên dương. Bắt đầu từ vector với tất cả các phần tử 1, tính $x, Mx, ..., M^n x$ cho đến khi hội tụ sau đó tính toán các vector riêng của M. Thực tế, trong trường hợp này, chỉ cần tính toán một số phép tính đầu để có được một xấp xỉ tốt của vector PageRank. Đối với một ma trận ngẫu nhiên, phương pháp lũy thừa giúp trì hoãn hội tụ. Điều đó khiến phương pháp này khá hiệu quả vì thực tế đồ thị web khá thưa thớt. Một đỉnh chỉ có một số lượng nhỏ các liên kết đi (nhiều là khoảng một vài trăm, mà như vậy thì cực kỳ nhỏ so với 30 tỷ đỉnh có thể liên kết theo lý thuyết). Do đó ma trận chuyển tiếp có rất nhiều phần tử bằng 0.

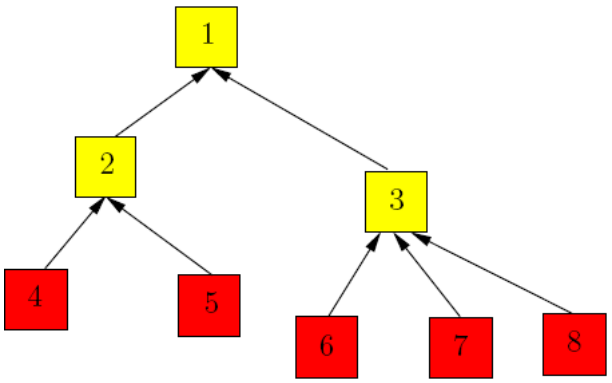
Kết thúc phần này, bạn hãy thử sức với các bài tập sau:

Bài toán 8. *Tính toán vector PageRank của đồ thị dưới đây, lấy hệ số tắt dần p lần lượt là $p = 0; p = 0,15; p = 0,5; p = 1$.*



△

Bài toán 9. *Tính toán vector PageRank của mô tả dưới đây, xem xét hệ số tắt dần $p = 0,15$. Giải thích kết quả của bạn về mối quan hệ giữa số lượng các liên kết mà mỗi đỉnh có và thứ hạng của nó.*



△

4 THUẬT TOÁN HITS - Hub VÀ TRANG CÓ UY TÍN TRÊN INTERNET

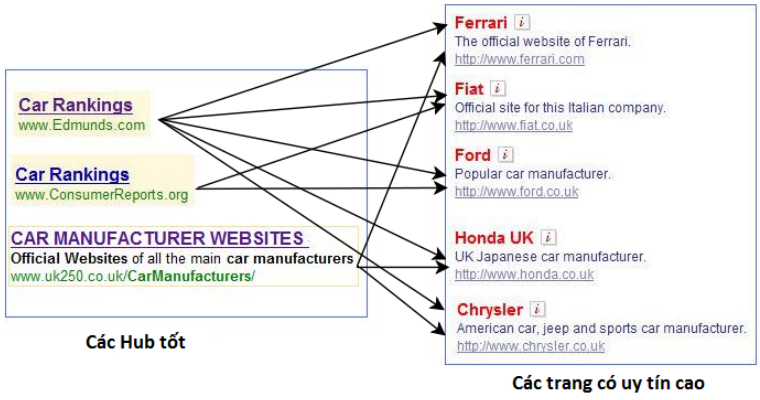
Cùng thời gian PageRank được phát triển, *Jon Kleinberg* - giáo sư tại khoa Khoa học Máy tính (Cornell) đã đưa ra phương pháp của riêng mình đối với bài toán tìm kiếm Web. Ông đã phát triển một thuật toán sử dụng các cấu trúc liên kết của trang web để xếp thứ hạng các trang liên quan cho mỗi chủ đề cụ thể. **HITS** (*hyperlink-induced topic search*) hiện là một phần của công cụ tìm kiếm **Ask** (*www.Ask.com*).

Trong công trình của *Jon Kleinberg* có một điểm khá thú vị. Ông đã phát hiện ra rằng quan điểm của con người về quá trình tìm kiếm phức tạp hơn việc so sánh một danh sách các từ truy vấn với danh sách trang được trả lại bởi máy tính. Giả sử bạn muốn mua một chiếc xe hơi và bạn gõ một cụm từ truy vấn chung chung như "*các nhà sản xuất ô tô tốt nhất trong 4 năm qua*". Ý định của bạn là tìm một danh sách các thương hiệu xe hơi hàng đầu cùng với website chính thức của họ. Khi bạn đặt câu hỏi này với bạn bè của bạn, họ sẽ hiểu ngay rằng ý bạn là xe hơi. Nhưng khi bạn đặt câu hỏi này cho một máy tính đang chạy hệ thống **xếp hạng trang theo văn bản** thì kết quả sẽ khiến bạn thất vọng. Nó sẽ đếm tất cả số lần xuất hiện của các từ trong các trang web và đưa cho bạn một câu trả lời không thông minh chút nào. Danh sách các trang web hàng đầu do máy tính trả về có thể là rất khác so với những gì bạn mong đợi. Một vấn đề là hầu hết các trang web chính thức thường không tự mô tả mình. Họ có thể không quảng cáo bản thân giống như cách công chúng nhìn nhận họ. Các công ty hàng đầu như Huyndai, Toyota, ... thậm chí có thể không sử dụng thuật ngữ "nhà sản xuất ô tô" trên các trang web của họ. Họ có thể chỉ sử dụng thuật ngữ "nhà sản xuất xe" hay chỉ là mô tả sản phẩm và dịch vụ của họ.

Cần làm gì trong trường hợp này? Thật tuyệt vời nếu máy tính có một từ điển hay bản thể luận. Khi đó, với bất kỳ cụm từ truy vấn nào, nó cũng có thể tìm ra các từ đồng nghĩa, gần nghĩa. Điều này có thể cải thiện chất lượng tìm kiếm. Tuy nhiên, cuối cùng, máy tính vẫn sử dụng hệ thống xếp hạng trang web theo văn bản. Kết quả thu được vẫn có số lượng lớn các trang không liên quan đến vấn đề truy vấn. Giống như đã nói ở đầu mục 3, xét một trang mà cụm từ "*nhà sản xuất ô tô*" = "*sản xuất xe hơi*" = "*nhà thiết kế xe*" lặp đi lặp lại một tỷ lần. Trang web này sẽ là trang đầu đầu tiên được công cụ tìm kiếm tìm thấy. Tuy nhiên, trang này có thực tế không phải là trang cần tìm.

Vậy là khi tìm kiếm web bằng một cụm từ truy vấn, ta cần một hệ thống xếp hạng trang khác để tìm ra những trang có uy tín cho một truy vấn cụ thể. Trang i được gọi là **trang có uy tín** đối với truy vấn "*nhà sản xuất ô tô*" nếu nó có chứa thông tin có giá trị về đề tài này. Trang web chính thức của các nhà sản xuất xe hơi như *www.bmw.com*, *HyundaiUSA.com*, *www.mercedes-benz.com* sẽ là có uy tín cho tìm kiếm này. Các trang web thương mại về bán xe ô tô cũng là những trang có uy tín. Đây là những trang thực sự có liên quan đến truy vấn của ta và cũng là thứ mà người sử dụng mong muốn thu được từ các công cụ tìm kiếm. Tuy nhiên, có một số trang liên quan đến quá trình tìm kiếm các trang có uy tín, được gọi là **Hub**. Vai trò của chúng là quảng cáo cho các trang có uy tín. Chúng chứa các liên kết hữu ích đến các trang có uy tín. Nói cách khác, các Hub giúp các công cụ tìm kiếm đi đúng hướng. Trong cuộc sống thực, khi mua một chiếc xe, bạn thường ưu tiên nhiều hơn cho đại lý được bạn bè của bạn giới thiệu. Trang có uy tín trong trường hợp này sẽ giống như đại lý xe hơi, còn các Hub chính là bạn bè của bạn. Bạn tin tưởng bạn bè mình, do đó bạn tin tưởng những gì họ giới thiệu. Trong thế

giới web, Hub về xe ô tô có thể là các trang chứa bảng xếp hạng xe, blog (nơi mọi người thảo luận về những chiếc xe mà họ mua) và các trang tương tự.



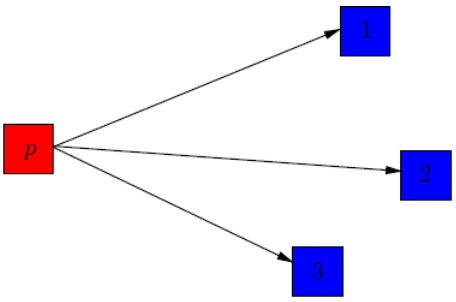
Truy vấn: Top automobile makers

Thuật toán HITS xác định *trang có uy tín* cao và *Hub* tốt cho một chủ đề tìm kiếm bằng cách gán cho mỗi trang 2 chỉ số, ta gọi là **trọng số uy tín** và **trọng số Hub**. Các trọng số được định nghĩa đệ quy với nhau. Trang web nào được trỏ đến bởi Hub có trọng số Hub cao thì cũng có trọng số uy tín cao. Hub nào trỏ đến nhiều trang có trọng số uy tín cao thì Hub đó có trọng số Hub cao.

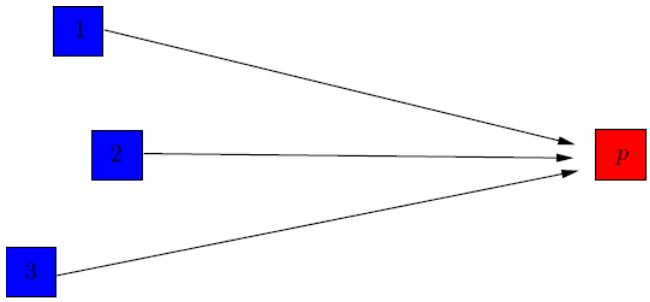
Để có được một tập hợp phong phú cả các Hub và các trang có uy tín cho một truy vấn Q, trước tiên cần thu thập top 200 trang có số lần xuất hiện các cụm từ tìm kiếm Q nhiều nhất. Những trang này, như đã chỉ ra ở trên, khả năng không liên quan đến điều cần tìm là rất lớn, nhưng ta phải bắt đầu từ những nơi như vậy. *Kleinberg* gọi tập này là **tập gốc**, kí hiệu R_Q. Các trang trong R_Q về bản chất là rất khác nhau và nói chung chỉ chứa rất ít (nếu có) liên kết với nhau. Vì vậy, đồ thị con được xác định bằng các trang này gần như hoàn toàn không liên thông. Ta cũng không thể thực hiện kỹ thuật PageRank trên R_Q.

Các trang có uy tín cho truy vấn Q chưa chắc đã có ở trong tập R_Q. Tuy nhiên, chúng có thể được chỉ ra bởi ít nhất một trang trong R_Q. Vì vậy, ta cần mở rộng đồ thị con bằng cách thêm vào tất cả các cạnh đến hoặc đi từ các đỉnh của R_Q. Kết quả của sự mở rộng đó cho ta đồ thị con S_Q, gọi là **hạt giống tìm kiếm**. Chú ý rằng S_Q mà ta vừa xây dựng vẫn là một đồ thị con nhỏ (nó chắc chắn là nhỏ hơn nhiều so với đồ thị web 30 tỷ đỉnh). Nó cũng có khả năng chứa rất nhiều nguồn tin cậy cho Q. Câu hỏi còn lại là làm thế nào để nhận biết và đánh giá chúng? Các trang có uy tín cho cùng một chủ đề sẽ có rất nhiều các trang từ S_Q trỏ vào chúng. Như cách nói ở đầu phần này, có sự trùng lặp lớn trong tập hợp các Hub chỉ đến chúng.

Bây giờ ta sử dụng ngôn ngữ toán học. Ta gán cho mỗi trang i hai số: Trọng số uy tín a_i, và trọng số hub h_i. Trang nào có a_i cao hơn thì trang đó tốt hơn, hub nào có h_i cao hơn thì hub đó tốt hơn. Kí hiệu $\{a_i\}$ và $\{h_i\}$ là tập hợp các a_i và h_i của tất cả các đỉnh trong S_Q. Ta cập nhật các trọng số như sau:



a_p là các h_i của tất cả các đỉnh i mà p trỏ đến.



h_p là các a_i của tất cả các đỉnh i mà trỏ đến p.

Một *hub* tốt làm tăng trọng số uy tín cho các trang mà nó trỏ đến. Một trang có uy tín cao làm tăng trọng số hub cho các trang trỏ đến nó. Ý tưởng bây giờ là áp dụng hai phép toán trên luân phiên cho đến khi đạt được giá trị cân bằng của 2 trọng số.

Cho A là ma trận kề của đồ thị S_Q. Gọi v, u lần lượt là vector biểu thị trọng số uy tín và trọng số hub.

$$v = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}; \quad u = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

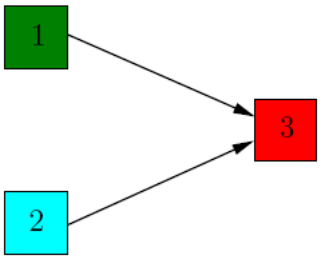
Thực hiện luân phiên hai phép toán với các trọng số theo sơ đồ trên, ta có:

$$\begin{cases} v = A^t.u \\ u = A.v \end{cases}$$

Gọi trọng số ban đầu tại các đỉnh là $u_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ và $v_0 = A^t.\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$. Khi đó, sau k bước, ta có hệ:

$$\begin{cases} v_k = (A^t.A).v_{k-1} \\ u_k = (A.A^t)u_{k-1} \end{cases}$$

Ví dụ 10. *Xét một đồ thị đơn giản:*



Ma trận kề của đồ thị là: $A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ *và ma trận chuyển vị* $A^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$.

Giả sử trọng số hub ban đầu được cho bởi vector $u = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. *Ta tính vector trọng số uy tín:*

$$v = A^t.u = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} . \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Khi đó, trọng số hub mới được cập nhật:

$$u = A.v = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

Điều này rất sát với trực giác của ta rằng đỉnh 3 là uy tín nhất, vì nó là đỉnh duy nhất có các liên kết đến, và các đỉnh 1, 2 là hai hub tốt như nhau. Nếu lặp lại các thao tác trên nhiều lần nữa, ta sẽ chỉ thu được bội vô hướng của vectơ v và u đã tính ở bước 1. Vì vậy, trọng số tương đối của các đỉnh vẫn như cũ. △

Với những đồ thị phức tạp hơn, ta không hi vọng có sự hội tụ cho bài toán, và trạng thái cân bằng (nếu có) cũng khó tìm hơn.

Định lý 12.
Giả sử rằng AA^t *và* A^tA *là các ma trận nguyên thủy. Gọi* $(v_k), (u_k)$ *lần lượt là dãy trọng số uy tín và dãy trọng số hub. Còn* $(V_k), (U_k)$ *lần lượt là dãy đã được chuẩn hóa từ* $(v_k), (u_k)$ *(mỗi vector có tổng các phần tử bằng 1). Tức là* $V_k = \frac{1}{c}v_k$, *với c là tổng các phần tử của* v_k *và* $U_k = \frac{1}{d}u_k$, *với d là tổng các phần tử của* u_k. *Khi đó:*

1. *Dãy* (V_k) *hội tụ tới vector xác suất duy nhất tương ứng với giá trị riêng của ma trận* A^tA.
2. *Dãy* (U_k) *hội tụ tới vector xác suất duy nhất tương ứng với giá trị riêng của ma trận* AA^t. □

Vì vậy, vector trọng số uy tín là vector riêng xác suất tương ứng với giá trị riêng lớn nhất của A^tA, và vector trọng số hub là các vector riêng xác suất ứng với giá trị riêng lớn nhất của AA^t.

Ta nhắc lại 3 định lý toán học dưới đây:

Định lý 13.

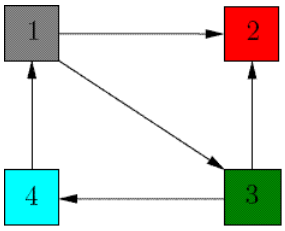
1. *Các ma trận AA^t và A^tA là các ma trận thực và đối xứng, do đó, chúng chỉ có giá trị riêng thực.*
2. ***(Perron Frobenius)** Nếu M là một ma trận nguyên thủy thì*
   (a) *Giá trị riêng lớn nhất λ của M là dương và bội 1.*
   (b) *Mọi giá trị riêng khác của M đều có môđun nhỏ hơn λ*
   (c) *Giá trị riêng lớn nhất λ có một vector riêng tương ứng với tất cả các phần tử đều dương.*
3. *Gọi M là một ma trận đối xứng, nguyên thủy không âm và v là vector riêng lớn nhất của M, với tổng của phần tử bằng 1. Gọi z là vector cột với tất cả các phần tử không âm, khi đó nếu ta chuẩn hóa các vectơ $z, Mz, ..., M^kz$ thì dãy này hội tụ đến v.* □

Khái niệm "**hội tụ**" ở đây được sử dụng một cách "lỏng lẻo". Một dãy vectơ (z_k) được gọi là hội tụ tới một vector v nếu như với k đủ lớn, các phần tử trong vector cột z_k rất gần với các phần tử tương ứng của vector cột v. Không đi sâu vào kỹ thuật chứng minh, phương pháp lũy thừa được sử dụng bởi vì ta chỉ có một vector riêng lớn nhất.

Thuật toán **HITS** theo tinh thần giống như **PageRank**. Cả hai đều sử dụng các cấu trúc liên kết của đồ thị web để quyết định thứ hạng của trang. Sự khác biệt là ở chỗ: không giống như *thuật toán PageRank*, *thuật toán HITS* chỉ xem xét hoạt động trên một đồ thị con nhỏ (hạt giống tìm kiếm S_Q) từ đồ thị web. Đồ thị con này phụ thuộc vào truy vấn ban đầu. Bất cứ khi nào ta thay đổi cụm từ truy vấn, các hạt giống tìm kiếm cũng thay đổi theo. HITS xếp hạng các nút hạt giống tìm kiếm theo trọng số uy tín và trọng số hub của chúng. Các trang xếp hạng cao nhất sẽ được hiển thị cho người dùng.

Bài toán 10. *Chứng minh rằng với mọi ma trận vuông A, các ma trận A^tA và AA^t đối xứng.*

Bài toán 11. *Tính trọng số uy tín, trọng số hub của các đỉnh trong đồ thị sau:*



Gợi ý: Tính ma trận A và chỉ ra rằng các giá trị riêng của AA^t là 0,1 và 3. Vector riêng đã được chuẩn hóa ứng với giá trị riêng lớn nhất $\lambda = 3$ là vector trọng số hub. Trọng số uy tín của đỉnh 4 là bao nhiêu? Điều này có đúng với trực giác của bạn không? △

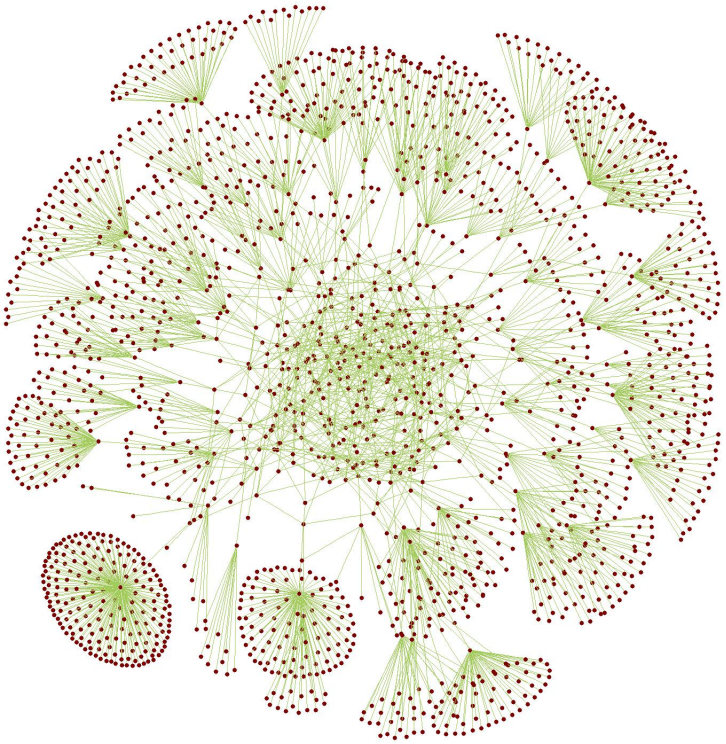
5 TOÁN INTERNET

Sự phát triển của World Wide Web tạo ra một lĩnh vực hoàn toàn mới của toán học được gọi là Toán Internet. Nó là sự kết hợp của xác suất, đại số tuyến tính, lý thuyết đồ thị và hệ động lực, được xây dựng để giải quyết các vấn đề phức tạp về cách lan truyền thông tin trên mạng.

Trong các phần trước, bạn đã biết cách Google sử dụng PageRank để xác định thứ hạng các trang web, do đó làm cho tìm kiếm Internet trở nên đơn giản và hiệu quả. Từ góc nhìn của một nhà thiết kế web, điều quan trọng là không chỉ tạo ra một trang web đẹp, có hiệu ứng đồ họa thú vị, mà điều quan trọng là những gì các trang khác liên kết đến trang web của bạn. Một PageRank tốt có thể khiến doanh nghiệp của bạn thu được một lợi nhuận rất cao. Tuy nhiên, từ quan điểm của Google, điều quan trọng là giữ cho các tính toán PageRank thật chính xác, và ngăn chặn những nỗ lực gian lận. **Link spam** được định nghĩa là những nỗ lực có chủ ý để cải thiện thứ hạng của một trang trên các công cụ tìm kiếm, bằng cách tăng số lượng các trang liên kết đến nó. **Link farms** liên quan đến việc tạo ra cộng đồng liên thông mạnh của các trang web tham chiếu lẫn nhau, thường được gọi một cách hài hước là "*Hội ngưỡng mộ lẫn nhau*". Mặc dù một số Link farms có thể được tạo ra bằng tay, nhưng hầu hết được tạo ra thông qua các chương trình tự động. Do đó dễ nhận ra đồ thị con dày đặc các web được thiết kế để làm việc đó, và loại bỏ chúng ra khỏi các tính toán xếp hạng trang. Mặc dù phần cốt lõi là PageRank như mô tả trong phần trước, nhưng trong thực tế, Google sử dụng thuật toán tìm kiếm đã được cải tiến để đối phó với những vấn đề này. Một trong số các tính năng bổ sung là thuật toán mới có thể gán trọng số khác nhau cho liên kết đi, hoặc quyết định rằng một số liên kết nhất định không nên sử dụng để xếp hạng trang. Như tuyên bố của Google, "Chúng tôi có hàng chục kỹ sư làm việc liên tục để cải thiện mọi khía cạnh của Google hàng ngày, PageRank vẫn tiếp tục cung cấp cơ sở cho tất cả các công cụ tìm kiếm web của chúng tôi".

Một hướng nghiên cứu là phân tích và so sánh các thuật toán xếp hạng khác nhau. Một thuộc tính quan trọng để đánh giá một thuật toán tốt là sự ổn định. World Wide Web đang thay đổi liên tục, các trang web được tạo ra và xóa đi ở mọi thời điểm. Cần có giải pháp ổn định bảng xếp hạng trang trong hoàn cảnh đồ thị Web nhiễu loạn nhẹ. Lý tưởng nhất là các thuật toán xếp hạng tự làm điều đó, khi đó ta không phải liên tục khởi động lại các tính toán.

Sử dụng cả toán học và khoa học máy tính, hiểu rõ cấu trúc của đồ thị Internet là một vấn đề nghiên cứu quan trọng. Tuy nhiên phát triển mô hình cho đồ thị Internet có hàng tỷ của đỉnh là rất xa so với các ví dụ đơn giản. Các web thường được mô tả là có **cấu trúc bow-tie**. Các đỉnh của bow-tie được đại diện bởi một thành phần liên thông mạnh của đồ thị, gọi là **lõi**. Một phần của bow-tie bao gồm các trang liên kết đến lõi, trong phần khác của bow-tie chứa các trang có chứa ít nhất một liên kết từ lõi. Phần còn lại của các trang web hoặc là trong các thành phần không liên thông, hoặc là các **tua**(trang chỉ có liên kết đến các trang bên ngoài lõi). Trong mô hình này, các **mạng xã hội** (cộng đồng web được định nghĩa là đồ thị con với nhiều liên kết nội bộ) là những trang bên ngoài. **Phương pháp quang phổ** được sử dụng để phân tích sự phân bố của các cộng đồng này.



Cấu trúc liên kết Internet bán ngẫu nhiên.
Các đỉnh là các chấm đỏ còn liên kết là những đường màu xanh lá cây.

Hiểu đồ thị Internet có thể giúp trả lời các câu hỏi về cách thông tin lan truyền trên mạng. Đặc biệt là sự lây truyền virus máy tính. Hiểu cơ chế lan truyền của virus có thể giúp các nhà sản xuất phần mềm chống virus dự đoán và kiểm dịch khu vực nhiễm.

Mặc dù trong các phần trước, ta chỉ xét các đồ thị hữu hạn, nhưng cũng có thể xem các trang web như một đồ thị vô hạn. Bởi vì các trang web có hai loại: tĩnh và động. Các trang tĩnh chứa thông tin không thay đổi. Trang động là những trang được dựng "tại chỗ", là kết quả của một truy vấn. Ví dụ, nếu ta tìm kiếm trên Google với cụm từ "toán học và khoa học máy tính", Google trả về một trang với các liên kết có liên quan cùng những mô tả của chúng. Tất nhiên, Google không có trang nào được chuẩn bị từ trước cho mỗi truy vấn ngẫu nhiên từ người dùng như vậy. Đối với mỗi truy vấn nhận được, nó tìm kiếm thông qua các chỉ số (dùng để theo dõi các tài liệu có chứa các từ tìm kiếm). Sau đó Google sẽ chạy thuật toán Page Rank để xếp hạng các truy cập có liên quan. Cuối cùng, họ dùng nó xây dựng lên câu trả lời cho người dùng. Một trang như vậy là một trang động. Trong khi số lượng các trang web tĩnh là hữu hạn (trên 30 tỷ trang, nhưng vẫn hữu hạn), thì số lượng các trang động là vô hạn. Xét một cách toàn diện cả trang tĩnh và trang động thì đồ thị web có thể được nghiên cứu bởi lý thuyết đồ thị vô hạn. Đằng sau đó là cả một ngành toán học thú vị, đầy thử thách, và vẫn đang được nghiên cứu liên tục.

Cuốn sách "*A course on web graphs*" của *Anthony Bonato* và tạp chí trực tuyến "Internet Mathematics" thành lập năm 2003, cung cấp các nguồn thông tin tuyệt vời về cách toán học cải thiện những trải nghiệm Internet của chúng ta. Tất cả các chủ đề thảo luận trên được trình bày chi tiết, với rất nhiều ví dụ thú vị.

6 TÀI LIỆU THAM KHẢO

Chúng tôi tìm thấy nhiều thông tin hữu ích trong việc chuẩn bị bài giảng này trong "*A course on web graphs*" của *Anthony Bonato* và "*Authoritative sources in a hyperlinked environment*" của *Jon Kleinberg*. Cả 2 đều được viết bằng một phong cách rất rõ ràng và hấp dẫn. Chúng tôi cũng tham khảo cuốn thứ hai khi trình bày quan điểm về tìm kiếm web và phương pháp giải quyết bài toán đó bằng thuật toán HITS. Cuốn "*The 25,000,000,000 Eigenvector: The Linear Algebra behind Google*" đã cố gắng cung cấp cái nhìn toàn diện về thuật toán PageRank dành cho người mới tiếp cận. Đối với những độc giả chuyên ngành khoa học máy tính, đặc biệt là nghiên cứu công cụ tìm kiếm, tài liệu "*The anatomy of a large-Scale Hypertextual Web Search Engine*" (của những người sáng lập Google, *Larry Page* và *Sergey Brin*) sẽ rất hữu ích. Cuốn sách "*A course on web graphs*" dành cho cả nhà toán học và khoa học máy tính. Nó cung cấp một cái nhìn toàn diện, hiện đại cho nhiều ứng dụng của lý thuyết đồ thị vào nghiên cứu đồ thị Web. Tài liệu tham khảo cuối cùng là các tạp chí trực tuyến Toán Internet.

1. **Anthony Bonato**, *A Course on The Web Graph*, AMS-AARMS, Graduate Studies in Mathematics v. 89, 2008.
   http://www.math.ryerson.ca/ abonato/webgraph.html

2. **Kurt Bryan, Tanya Leise**, *The 25,000,000,000 Eigenvector: The Linear Algebra behind Google*, SIAM Review, Vol. 48, No. 3. (2006).
   http://www.siam.org/journals/sirev/48-3/62328.html

3. **Sergey Brin, Lawrence Page,** *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Seventh International World-Wide Web Conference (WWW 1998).
   http://infolab.stanford.edu/pub/papers/google.pdf

4. **M. Brin, G. Stuck**, *Introduction to Dynamical Systems* , Cambridge University Press, 2002.

5. **David Austin**, *How Google Finds Your Needle in the Web's Haystack.*
   http://www.ams.org/featurecolumn/archive/pagerank.html

6. **Jon Kleinberg**, *Authoritative sources in a hyperlinked environment*, Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.
   http://www.cs.cornell.edu/home/kleinber/auth.pdf

7. *Internet Mathematics Journal.*