Mục lục

1 Tổng quan về bài toán **2**

2 Cơ sở lý thuyết **3**
2.1 Các kiến thức về đại số tuyến tính 3
2.1.1 Trị riêng, vector riêng 3
2.1.2 Phương pháp luỹ thừa 3
2.2 Xích markov và phân bố dừng 3
2.2.1 Khái niệm xích Markov 3
2.2.2 Ma trận xác suất chuyển 3
2.2.3 Phân bố dừng 3
2.3 Thuật toán PageRank 4
2.3.1 Ý tưởng thuật toán 5
2.3.2 Thuật toán PageRank 7

3 Áp dụng vào bài toán phát hiện cuộc gọi spam **9**

Lời mở đầu

Chương 1

Tổng quan về bài toán

Chương 2

Cơ sở lý thuyết

2.1 Các kiến thức về đại số tuyến tính

2.1.1 Trị riêng, vector riêng

2.1.2 Phương pháp luỹ thừa

2.2 Xích markov và phân bố dừng

2.2.1 Khái niệm xích Markov

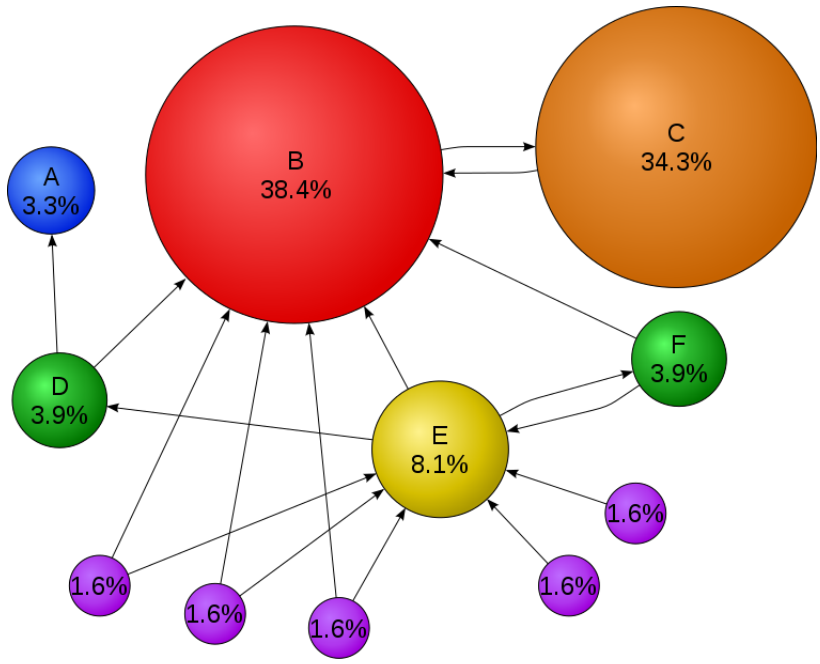
2.2.2 Ma trận xác suất chuyển

2.2.3 Phân bố dừng

2.3 Thuật toán PageRank

Kể từ khi xuất hiện mạng internet, đã có rất nhiều cách lưu trữ và tổ chức các trang web nhằm mục đích dễ dàng tìm kiếm. Trước đó, **Yahoo** đã lưu trữ các trang web trong các thư mục, người dùng muốn tìm kiếm trang web về thể loại gì thì có thể vào thư mục đó. Để cho kết quả tìm kiếm tốt hơn, khái niệm về **link popularity** ra đời. Theo đó, số trang liên kết tới một trang web sẽ đo độ quan trọng của trang web đó. Do đó, một trang web được coi là quan trọng hơn nếu có nhiều trang web liên kết tới nó. Trái ngược với link popularity, **PageRank** không chỉ dựa trên tổng số trang liên kết với trang web mà còn dựa trên *rank* của các trang liên kết tới nó.

PageRank là một thuật toán trước đây được **Google Search** sử dụng để xếp hạng trang web trên bộ máy tìm kiếm của họ. PageRank được sáng tạo bởi Brin và Page, là một thuật toán **học xếp hạng** dựa trên phân tích **đồ thị** liên kết giữa các trang web, mỗi trang web sẽ được xem như một đỉnh, mỗi liên kết sẽ được xem như một cạnh của đồ thị.



Hình 2.1: Mô phỏng thuật toán PageRank với 11 trang web

Hình 2.1 mô tả thuật toán pagerank với 11 trang web. Ta có 11 trang web. Trang web B có nhiều trang web liên kết với nó nhất, do đó rank của trang web B là cao nhất. Trang web C mặc dù số trang web liên kết với nó ít hơn trang E nhưng mà nó được trang B liên kết tới, do đó nó sẽ có rank cao hơn trang E. Như vậy kết quả rank cuối cùng của một trang web dựa trên cấu trúc liên kết của toàn bộ các trang web. Cách tiếp cận này nghe mặc dù rất rộng và phức tạp, nhưng Page và Brin đã có thể đưa nó vào thực tế bằng một thuật toán tương đối tầm thường.

2.3.1 Ý tưởng thuật toán

Tổng quát, giả sử chúng ta có n trang web được đánh số từ $1, \ldots, n$, PageRank của trang web i được tính dựa trên các liên kết trang web khác đến nó (trang web j liên kết trỏ đến i), nhưng không phải bất kỳ liên kết nào cũng được tính điểm như nhau. Thuật toán PageRank được xây dựng dựa trên hai ý tưởng như sau:

- Trang web A trỏ liên kết đến B, nếu A là một trang web xếp hạng cao thì phải giúp B xếp hạng cao hơn.
- Trang web A trỏ liên kết tới B, lượng trang web mà A trỏ tới nghịch biến với xếp hạng của B hay nói cách khác A trỏ đến càng nhiều trang thì giúp B tăng thứ hạng càng ít.

PageRank của một trang web P_i, ký hiệu là $r(P_i)$:

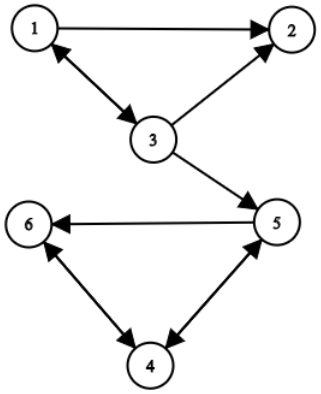
$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \tag{2.1}$$

trong đó B_{P_i} là tập các trang web trỏ đến P_i, $|P_j|$ là tổng số liên kết đi ra từ P_j. Công thức trên thoả mãn ý tưởng trên. Vấn đề với phương trình (2.1) đó là giá trị $r(P_j)$, PageRank của các trang P_j là chưa biết. Brin và Page giải quyết vấn đề này bằng phương pháp lặp. Họ giả sử rằng ban đầu tất cả các trang có PageRank bằng nhau (cụ thể hơn là $1/n$, trong đó n là số trang trong danh mục Web của Google). Sau đó các giá trị đó được dùng trong phương trình (2.1) để tính $r(P_i)$ mới cho từng trang P_i trong danh mục. Sau khi tính được các $r(P_i)$ mới, chúng lại được thay vào (2.1) ở vị trí của $r(P_j)$ trong vòng lặp tiếp theo. Ký hiệu $r_{k+1}(P_i)$ là PageRank của trang P_i tại vòng lặp thứ $k+1$, khi đó,

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|} \tag{2.2}$$

Trong đó $r_0(P_i) = 1/n$ với mọi trang và quá trình này được lặp lại liên tục với mong muốn rằng từ một thời điểm nào đó trở đi, các giá trị PageRank sẽ hội tụ đến các giá trị cố định.

Ví dụ 1. *Xét mô hình internet gồm 6 trang web được đánh số* $1, \ldots, 6$ *như hình 2.2. Tính toán thứ hạng cho các trang web sau một vài vòng lặp.*



Hình 2.2: Ví dụ mô hình liên kết của 6 trang web

Thực hiện việc tính toán theo thuật toán, ta có bảng sau:

Vòng lặp 0	Vòng lặp 1	Vòng lặp 2	Thứ hạng ở vòng lặp 2
$r_0(P_1) = 1/6$	$r_1(P_1) = 1/18$	$r_2(P_1) = 1/36$	5
$r_0(P_2) = 1/6$	$r_1(P_2) = 5/36$	$r_2(P_2) = 1/18$	4
$r_0(P_3) = 1/6$	$r_1(P_3) = 1/12$	$r_2(P_3) = 1/36$	5
$r_0(P_4) = 1/6$	$r_1(P_4) = 1/4$	$r_2(P_4) = 17/72$	1
$r_0(P_5) = 1/6$	$r_1(P_5) = 5/36$	$r_2(P_5) = 11/72$	3
$r_0(P_6) = 1/6$	$r_1(P_6) = 1/6$	$r_2(P_6) = 14/72$	2

Để đơn giản hơn, ta sẽ mô hình hoá ví dụ trên dưới dạng ma trận. Gọi H là ma trận cấp $n \times n$ thoả mãn: $H_{ij} = \frac{1}{|P_i|}$ nếu tồn tại đường dẫn từ trang i sang trang j, và bằng 0 nếu ngược lại. Trong ví dụ trên, ma trận H sẽ là:

$$H = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_4 \\ P_6 \end{array} \begin{array}{c} \begin{array}{cccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Dễ thấy các hàng của ma trận có tính chất xác suất. Các phần tử khác 0 ở dòng i tương ứng với các liên kết ra của trang thứ i, trong khi đó, các phần tử khác 0 của cột i tương ứng với liên kết vào của trang thứ i.

Tiếp theo, gọi $\pi^{(k)T}$ là vector PageRank tại vòng lặp thứ k.
Khi đó, phương trình (2.2) có thể viết thành:

$$\pi^{(k+1)T} = \pi^{(k)T} H \tag{2.3}$$

Từ phương trình (2.3), ta đặt ra câu hỏi là liệu có tồn tại trạng thái π^T nào đó mà tại đó, nếu tiếp tục thực hiện (2.3) thì vector PageRank không thay đổi:

$$\pi^T = \pi^T H \tag{2.4}$$

hay

$$\pi = H\pi \tag{2.5}$$

Nếu nhìn kỹ, chúng ta cũng sẽ thấy đây là bài toán trị riêng, vector riêng của ma trận H^T với trị riêng $\lambda = 1$. Như chúng ta đã biết, với một trị riêng λ có thể có nhiều vector riêng và hơn nữa P^T chưa chắc đã có trị riêng $\lambda = 1$. Đây là vấn đề phát sinh với ý tưởng ban đầu của các tác giả, để từ đó đưa ra thuật toán PageRank hoàn thiện hơn.

2.3.2 Thuật toán PageRank

Ma trận H trông giống ma trận xác suất chuyển của xích Markov. Ta có định lý **Perron Frobenius** chỉ ra rằng nếu như ma trận ngẫu nhiên cột (tổng thành phần từng cột bằng 1) mà ta đang xét H là ma trận dương có từng thành phần $H_{i,j} > 0$ thì ma trận H chỉ có duy nhất một phân bố dừng (duy nhất một vector riêng tương ứng trị riêng 1). Tất cả trị riêng còn lại nhỏ hơn 1.
Dựa trên định lý này, Brin và Page đã chỉnh sửa H trở thành một ma trận xác suất của một chuỗi Markov để quá trình lặp có thể hội tụ bằng cách thêm một tham số d được gọi là **Damping Factor**.

Lý thuyết PageRank cho rằng, ngay cả một người dùng giả thiết click ngẫu nhiên vào các trang web cuối cùng cũng sẽ dừng lại. Xác suất người dùng tiếp tục click trong bất cứ bước nào được gọi là yếu tố damping. Có nhiều nghiên cứu đã thử các giá trị yếu tố damping, giá trị ước lượng bằng 0.85 là người dùng sẽ tiếp tục lướt web. Công thức tính Pagerank có tính đến yếu tố damping sử dụng mô hình khi người dùng bất kỳ sẽ cảm thấy chán sau một vài lần click và chuyển đến vài trang web khác một cách ngẫu nhiên. Như vậy:

$$\pi = \left(\frac{1-d}{n}E + dH\right)\pi \tag{2.6}$$

Với E là ma trận vuông $n \times n$ mà tất cả phần tử bằng 1. Ma trận $G = \frac{1-d}{n}E + dH$ được gọi là **ma trận Google**.
Ta cũng có thể biểu diễn công thức dưới dạng sau:

$$r(P_i) = \frac{1-d}{N} + d\sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|} \tag{2.7}$$

Công thức trên đã làm "mất" các số 0 ban đầu, sử dụng mô hình khi người dùng ngẫu nhiên cảm thấy chán sau khi click và được chuyển đến một số trang ngẫu nhiên. Giá

trị Pagerank thể hiện những cơ hội mà người dùng ngẫu nhiên sẽ được chuyển đến trang đó bằng cách click vào các đường link. Mô hình này có thể được hiểu tương tự như Markov chain, trong đó các tỉnh là các trang web, quá trình di chuyển có xác suất ngang nhau được coi như các link giữa các trang web. Nếu như trang web không có đường link đến các trang khác, nó sẽ thành ngõ cụt và việc truy cập ngẫu nhiên sẽ dừng lại. Nhưng nếu người dùng đến trang không có các link khác, thì người dùng sẽ chọn ngẫu nhiên một trang khác để tiếp tục truy cập. Khi tính Pagerank, những trang không có link trỏ đi các trang khác sẽ được giả định có link trỏ đến tất cả các trang trong tập văn bản. Và như vậy giá trị Pagerank sẽ được chia đều cho các trang khác.

Quay trở lại với ví dụ 1, nếu với $d = 0.9$, ta sẽ có ma trận:

$$G = \begin{pmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{pmatrix}$$

Và vector PageRank là vector phân phối dừng của ma trận $\mathbf{G}$ và bằng

$$\pi^T = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ (0.03721 & 0.05396 & 0.04151 & 0.3751 & 0.206 & 0.2862) \end{matrix}$$

Điều này tức là, với $\pi_1 = 0.03721$ có nghĩa là 3.721% thời gian người lướt web sẽ ở trang 1. Vì vậy, các trang trong mạng Hyperlink trên có thể xếp hạng theo thứ tự $(4 \quad 6 \quad 5 \quad 2 \quad 3 \quad 1)$, tức là trang thứ 4 là trang quan trọng nhất và trang 1 là trang ít quan trọng nhất.

Chương 3

Áp dụng vào bài toán phát hiện cuộc gọi spam

Tài liệu tham khảo