Matrizes e Google

Manuela da Silva Souza

IME - USP

13 de novembro de 2013

História

- Motores de busca do final dos anos 90: Altavista, Lycos, Yahoo.
- Por volta de 98/99 surge o Google e se torna o motor de busca mais importante da Web.
- Criadores do Google: Sergey Brin e Larry Page (na época estudantes de doutorado da Universidade de Stanford, EUA).


Google
Google

História

Funções de um motor de busca:

1. Percorrer toda a Web e localizar todas as páginas que podem ser acessadas;
2. Indexar as páginas encontradas $(p_1, p_2, \ldots, p_n)$
3. Classificar a importância de cada página de forma que, quando um utilizador realizar uma busca, as páginas mais importantes sejam apresentadas primeiro.

História

- (1) e (2) é comum a todos os motores de busca da web.
- Os motores de busca até 1997, para classificar a importância de uma página usavam comparações de conteúdos através de bases de dados (gigantescas). Nos dias atuais isso seria impraticável!

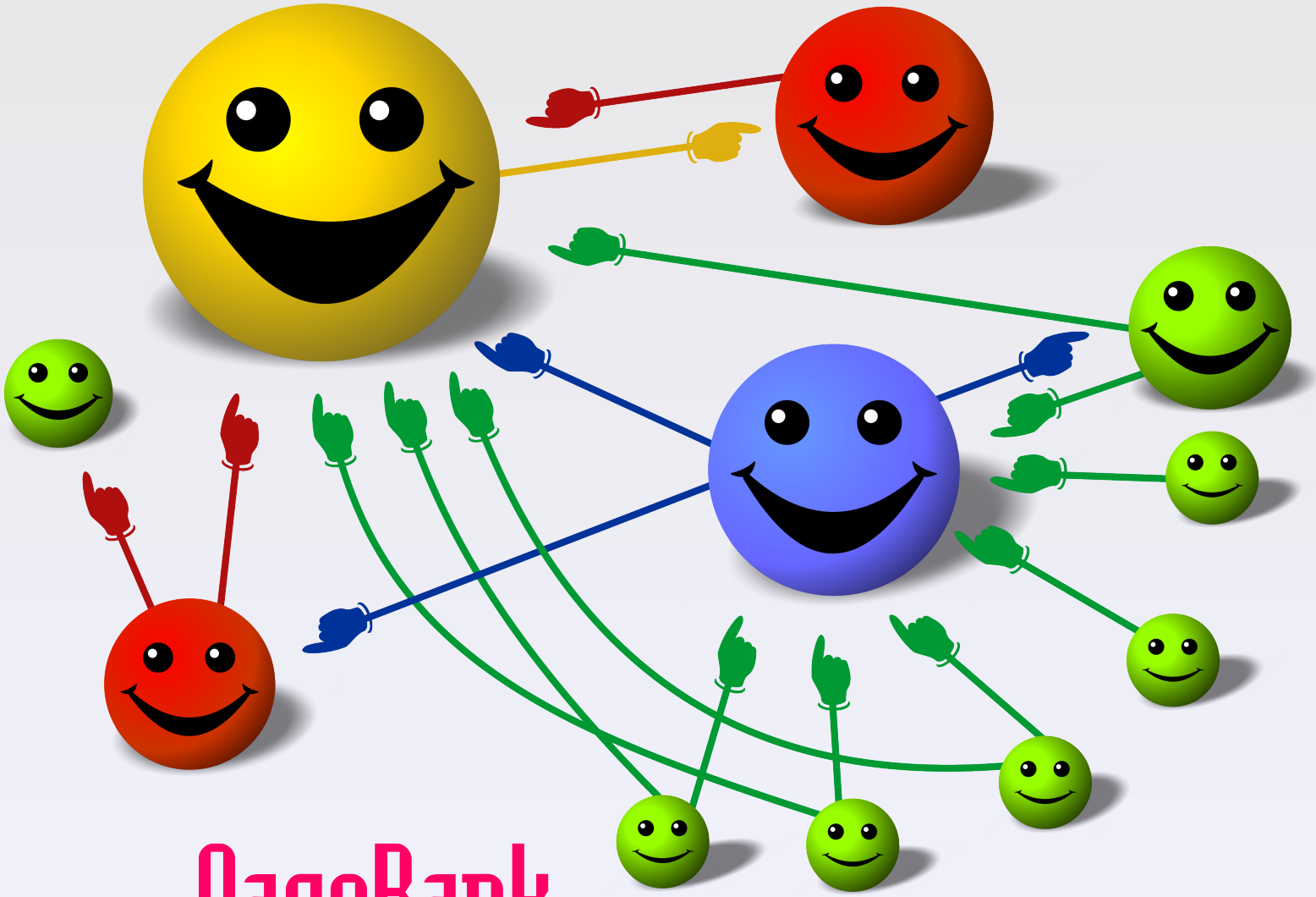
Revolução Google

- O Google criou um algoritmo eficiente para estabelecer o ranking das páginas da web.
- A ideia é, em vez de ser o motor de busca a julgar a importância das páginas com base em uma base de dados, deixar a decisão da importância da página a própria web com base em uma "votação democrática via links".

O peso númerico dado a cada página p da web (que mede o seu grau de importância) é chamado PageRank de p.
O valor de PageRank de uma página depende do número de links em qualquer lugar da web para aquela página. Links de páginas mais importantes, valem mais.

Uma página tem um valor mais alto de PageRank se:

- existem muitas páginas que a fazem referência via links (pdfs, imagens, referências, etc);
- Existem algumas páginas que a fazem referência mas se tratam de páginas importantes (com PageRank alto).


PageRank

No entanto, esse processo é vulnerável a manipulações.

Se eu criar uma página fictícia com 500 links para a página p estou aumentando artificialmente o PageRank de p. Para resolver isso, se a página p_i tem n_i links, então atribuímos a cada um desses links um valor proporcional ao número de links , ou seja, $\frac{1}{n_i}$.

O valor do PageRank de uma página p_i denotado por x_{p_i} é:

$$x_{p_i} = \sum_{p_j \in L_i} \frac{x_{p_j}}{n_j}$$

em que L_i é o conjunto de todas as páginas na web que possuem links para p_i.

Note que

$$x_{p_i} = \sum_{p_j \in L_i} \frac{x_{p_j}}{n_j} \Leftrightarrow x_{p_i} - \sum_{p_j \in L_i} \frac{x_{p_j}}{n_j} = 0$$

para toda página p_i da web.
Trata-se de um sistema linear homogêneo com bilhões de variáveis (e cresce a cada dia).

Construindo uma matriz M de ordem n em que n é igual ao número total de páginas da web tal que cada elemento m_{ij} de M é dado pela função

$$I(p_i, p_j) = \begin{cases} 0, & \text{se não existe referência de } p_i \text{ para } p_j \\ \dfrac{1}{L_j}, & \text{se existe referência de } p_i \text{ para } p_j \end{cases}$$

em que L_j é igual ao número de referências existentes em p_j,

obtemos:

$$X = \underbrace{\begin{pmatrix} l(p_1,p_1) & l(p_1,p_2) & \cdots & l(p_1,p_n) \\ l(p_2,p_1) & \ddots & & \vdots \\ \vdots & & l(p_i,p_j) & \\ l(p_n,p_1) & \cdots & & l(p_n,p_n) \end{pmatrix}}_{M=\text{matriz de transição}} X$$

em que $X = \begin{pmatrix} x_{p_1} \\ x_{p_2} \\ \vdots \\ x_{p_n} \end{pmatrix}$ é o vetor que contém o valor do PageRank de todas as páginas.

Note que

- $l(p_i, p_j)$ é a probabilidade de um usário sair da página p_i para a página p_j através de links.
- $l(p_1, p_j) + \cdots + l(p_n, p_j) = 1$.
- A matriz M supõe que as transições entre páginas se dão aleatoriamente através de links (matriz de Markov).

O problema não se encontra muito bem formulado do ponto de vista da "engenharia matemática", uma vez que o PageRank de uma página p de acordo com a nossa formulação significa a probabilidade de um usuário chegar a página p via links saindo de uma página aleatória da web.

O que pode acontecer com um usuário depois de algum tempo fazendo uma busca na web?

- Se "aborrecer" e não seguir mais os links, abrir uma outra página da web aleatoriamente.
- Chegar a uma página que não tem links.
- Ficar preso em um ciclo de páginas p_1 que cita p_2 que cita p_3 que cita p_1.

Para resolver esses problemas é introduzido uma probabilidade $0 < d < 1$ do usuário continuar a seguir os links, chamado fator de amortecimento.

Desta forma, o valor do PageRank de uma página p passa a ter uma componente correspondente à contribuição das páginas que citam p, ponderada pela probabilidade d do utilizador seguir os links das páginas:

$$d\left(\sum_{p_j \in L} \frac{x_{p_j}}{n_j}\right)$$

e uma componente correspondente ao usuário ter selecionado a página aleatoriamente (sem seguir links) ponderada pela probabilidade (1-d):

$$(1-d)\frac{1}{n}.$$

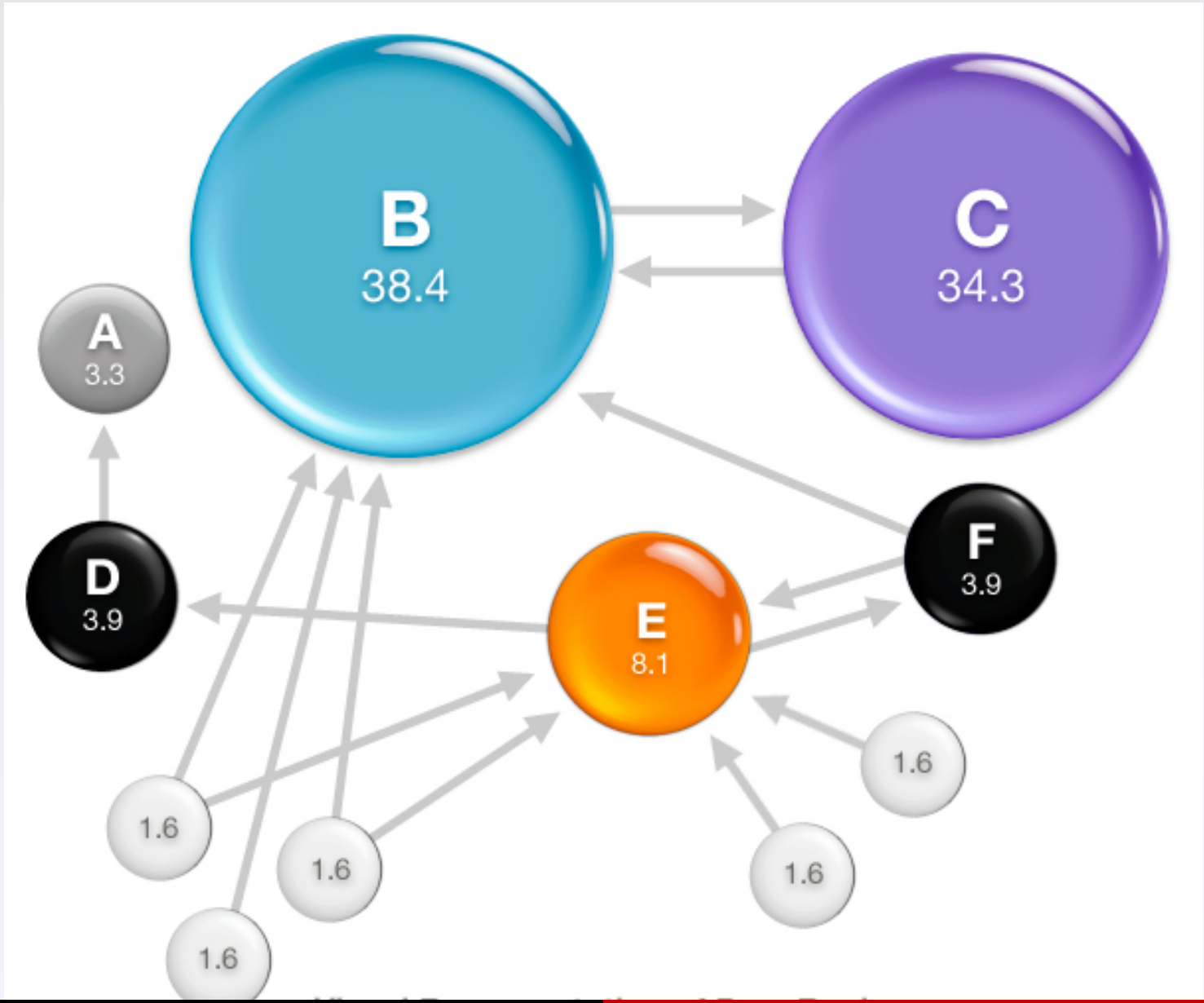
Disso, o valor do PageRank de uma página p passa a ser:

$$x_p = d\left(\sum_{p_j \in L} \frac{x_{p_j}}{n_j}\right) + (1-d)\frac{1}{n}.$$

Experimentalmente, chegou-se a conclusão que o valor adequado para d é $0,85$.

O fator de amortecimento introduz as seguintes características ao cálculo do PageRank:

- Uma página, simplesmente por existir, tem uma probabilidade igual a todas as outras de ser escolhida por um usuário na web
- Uma página que não tem ligações está ligada a todas as outras da web
- Resolução dos problemas de páginas sem ligações e ciclos (p_1 que cita p_2 que cita p_3 que cita p_1)


B
38.4
C
34.3
A
3.3
D
3.9
E
8.1
F
3.9
1.6
1.6
1.6
1.6
1.6

Logo,

$$x_{p_i} = (1-d)\frac{1}{n} + d\left(\sum_{P_j \in L_i} \frac{x_{P_j}}{n_j}\right),$$

para toda página p_i da web. Ou equivalentemente, na linguagem matricial

$$X = \begin{pmatrix} \frac{1-d}{n} \\ \frac{1-d}{n} \\ \vdots \\ \frac{1-d}{n} \end{pmatrix} + d \begin{pmatrix} I(p_1,p_1) & I(p_1,p_2) & \cdots & I(p_1,p_n) \\ I(p_2,p_1) & \ddots & & \vdots \\ \vdots & & I(p_i,p_j) & \\ I(p_n,p_1) & \cdots & & I(p_n,p_n) \end{pmatrix} X$$

Como X é um vetor de probabilidade (a soma de suas coordenadas é 1) temos:

$$\begin{pmatrix} \frac{1-d}{n} \\ \frac{1-d}{n} \\ \vdots \\ \frac{1-d}{n} \end{pmatrix} = \frac{1-d}{n} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \ddots & & \vdots \\ \vdots & & 1 & \\ 1 & \ldots & & 1 \end{pmatrix} X.$$

Matriz do Google

Portanto,

$$X = \underbrace{\left(\frac{1-d}{n} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \ddots & & \vdots \\ \vdots & & 1 & \\ 1 & \cdots & & 1 \end{pmatrix} + d \begin{pmatrix} I(p_1,p_1) & I(p_1,p_2) & \cdots & I(p_1,p_n) \\ I(p_2,p_1) & \ddots & & \vdots \\ \vdots & & I(p_i,p_j) & \\ I(p_n,p_1) & \cdots & & I(p_n,p_n) \end{pmatrix} \right)}_{G = \text{ matriz do google}} X$$

Em outras palavras, procuramos um vetor de probabilidade X que é fixo pela matriz G, ou seja,

$$X = GX.$$

Uma matriz de transição é regular se uma potência positiva da matriz tem todas as entradas positivas. Em particular, a matriz do Google G é regular.

Teorema

Se G é uma matriz de transição regular e Y é um vetor de probabilidade qualquer, então

$$G^n Y \to \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = X$$

quando $n \to \infty$, em que X é um vetor de probabilidade fixo, ou seja, $GX = X$, independente de n, cujas entradas são todas positivas.

Teorema

O vetor de estado estacionário q de uma matriz de transição G é o único vetor de probabilidade que satisfaz a equação

$$GX = X.$$

A técnica mais eficiente para calcular o vetor de estado estacionário X é simplesmente calcular $G^n X$ para n bem grande (neste caso, aproximadamente 52).

Mais ou menos uma vez por mês o Google determina o vetor fixo X da matriz G (a matriz G muda com o tempo) e através dos valores de x_{p_j} faz-se a ordenação das páginas da web por ordem de importância absoluta. Esse ranking é guardado em base de dados e é usado para ordenar as páginas selecionadas, quando fazemos uma pesquisa.

Referências

 http://pt.wikipedia.org/wiki/PageRank

 http://www.pgarrao.uac.pt/Matematical/googlecronica.pdf (crónica do professor Jorge Buescu da Universidade de Lisboa)

 http://zoo.cs.yale.edu/classes/cs426/2012/bib/brin98theanatomy.pdf

 J. Kemeny and J. Snell; *Finite Markov Chains*; 1970

Obrigada pela atenção!