PageRank: A Álgebra Linear e o algoritmo do *Google Search*

Elyabe Alves, *Ciência da Computação, UFES, Gabriel Moura, Ciência da Computação, UFES*

Resumo—Este trabalho sintetiza a teoria e apresenta a experiência prática de estudo e compreensão do algoritmo de escalonamento de página da *Google*, caracterizado como uma investigação dos conceitos correlacionados da matemática Aplicada como ferramenta para a computação.

Palavras-chave—PageRank; Google algoritmo; Álgebra Linear.

1 INTRODUÇÃO

ATUALMENTE, o *Google* é um das maiores companhias do setor de tecnológico. A empresa está presente no cotidiano de milhões de usuários por meio de seus produtos que vão desde softwares, sejam de uso livre ou comercializado, até hardware como *smartphones*, aparelhos vestíveis, *notebooks* e muitos outros acessórios avançados. No entanto, é inegável que todo este sucesso teve origem no serviço tido como seu *core*: O motor de busca.

Criado por Larry Page e Sergey Brim e dos motores mais utilizados no mundo, o *Google Search* inovou na década de 90 pelo seu modo, prático, rápido e, até então, impensado de construir um *ranking* das páginas *web* recém-nascida à época**.**

Nesta publicação, tratar-se-á o algoritmo utilizado diariamente pela empresa em suas fases, correções e aprimoramentos. Por fim, uma sugestão de implementação é descrita na seção 5.

2 CONCEITOS RELACIONADOS

Um *hiperlink* é uma ligação entre páginas ou arquivos da web. Desse modo, por simplificação, dizemos que há um *link* entre as páginas 1 e 2, se é possível navegar com destino à página 2, tendo como ponto de partida a página 1. Matematicamente, um conjunto de páginas da *web* – os *sites* serão aqui representadas por meio da estrutura especial: um grafo, mais especificamente, direcionado.

Um grafo $G = (V, E)$ é um par ordenado composto pelos conjuntos de vértices (V) e, segmentos curvos que conectam dois vértices, representado por (a, b), denominadas arestas (E). Neste contexto, em particular, os vértices rotulados de 1 a n, representam cada um, uma única página dentre um conjunto de n sites. A Figura 1 exibe um caso no qual $n = 4$ páginas.

- *A. Elyabe é graduando em Ciência da Computação pela Universidade Federal do Espírito Santo, Ufes, São Mateus – ES, Brasil.*
  E-mail: elyabe@outlook.com Github: https://github.com/Elyabe

- *M. Gabriel é graduando em Ciência da Computação pela Universidade Federal do Espírito Santo, Ufes, São Mateus – ES, Brasil.*
  E-mail: gab.mv@hotmail.com Github: https://github.com/GMVargass

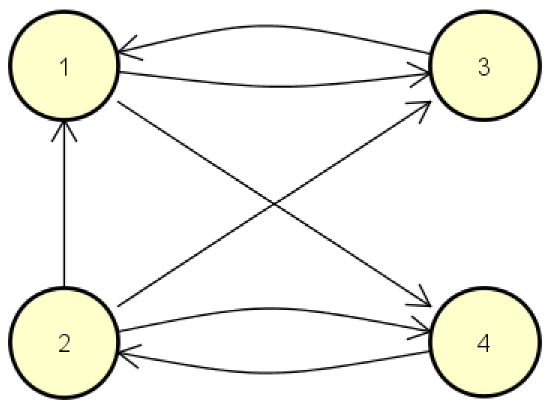

Fig. 1. Grafo representando 4 páginas da web. Note que, partindo da página 1, podemos chegar às páginas 4 e 3. Existem, também, *links* entre as páginas 2, 3 e 4.

2.1 A matriz de *hiperlinks*

Dado um conjunto formado por n páginas, uma matriz de

$$w_{ij} = \begin{cases} 1, & \text{se existe } link \text{ na página } j \text{ para a página } i. \\ 0, & \text{caso contrário.} \end{cases} \quad (1)$$

hiperlinks é uma matriz $W = [w_{ij},]_{nxn}$, onde

$$W = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \quad (2)$$

Desse modo, para o exemplo da Figura 1, tem-se

2.2 O *score importance*

A ideia básica por trás do algoritmo é simples, e consiste em mapear a importância de uma página baseada nos chamados *backlinks*, *links* que se originam em outras páginas e possuem como destino a página em questão. Existem 2 *backlinks* para a página 3 e apenas 1 para a página 2.

Seja x_k o número de *backlinks* da k-ésima página, para algum $1 \le k \le n$. No entanto, esta abordagem falha no sentido de que atribui o mesmo peso tanto a um *backlink* de uma página importante quanto a um advindo de uma página não importante. Desse modo, uma forma de garantir

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j} \ . \quad (3)$$

esta proporcionalidade é tornar

Onde $L_k \subset \{1, 2, \dots, n\}$ e n_j é quantidade de *links* saindo da j-ésima página. Desse modo, para o exemplo

$$(ii) \quad : \begin{cases} \frac{x_2}{3} + \frac{x_3}{1} = x_1 \\ \frac{x_4}{1} = x_2 \\ \frac{x_1}{2} + \frac{x_2}{3} = x_3 \\ \frac{x_1}{2} + \frac{x_2}{3} = x_4 \end{cases} \quad (4)$$

apresentado na Figura 1, forma-se o sistema linear

$$\begin{bmatrix} 0 & 1/3 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1/2 & 1/3 & 0 & 0 \\ 1/2 & 1/3 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \quad (5)$$

que representado na forma matricial fica

É fácil ver que o sistema possui o formato $Av = \lambda v$, em que A é a matriz de um operador linear de um espaço V e $v \in V$, $v \neq o$ e $\lambda \in \mathbb{R} - \{0\}$ são, respectivamente, autovetor e autovalor. Dessa forma, encontrar os *scores* das páginas reduz-se à encontrar o autovetor associado ao autovalor $\lambda = 1$. Analiticamente, encontrar v significa resolver o sistema

$$Av = \lambda v \Leftrightarrow Av - \lambda v I_n = 0_n \Leftrightarrow (A - \lambda I_n)v = 0_n \ , \quad (6)$$

em que λ é as raízes do polinômio chamado característico

$$p(\lambda) = \det(A - \lambda I_n). \qquad (7)$$

Além do mais, os autovetores associados a cada λ encontrado, é relativamente simples.
Computacionalmente, um método para encontrar o autovetor associado é apresentado na seção a seguir.

3 O Método das Potências

O *Método das Potências* é um algoritmo matemático aplicado à busca de autovalor e autovetor de uma dada matriz A. Caracteriza-se como um método iterativo que determina numericamente o autovalor de módulo máximo de A. É considerado vantajoso pois não calcula a decomposição matricial, podendo ser utilizado em uma matriz esparsa, e apenas uma singela modificação permite que outros autovalores sejam encontrados além de encontrar de forma paralela o par autovalor-autovetor associado.

3.1 Funcionamento do algoritmo

Dados uma matriz A e $\boldsymbol{q}^{(0)} \neq \boldsymbol{0}$ um vetor inicial com $\|v\|_2 = 1$, um limiar de tolerância ϵ e um número máximo $iter\_max$ de iterações, o procedimento é mostrado no Quadro 1.

$k \leftarrow 0$
Enquanto $erro \geq \epsilon$ e $iter\_\max > k$ **faça**:
$\boldsymbol{z}^{(k+1)} = \boldsymbol{A} \cdot \boldsymbol{q}^{(k)}$
$\boldsymbol{q}^{(k+1)} = \boldsymbol{z}^{(k+1)} / \|\boldsymbol{z}^{(k+1)}\|_2$
$\lambda^{(k+1)} = \boldsymbol{q}^{(k+1)} \cdot \boldsymbol{A} \cdot \boldsymbol{q}^{(k+1)}$
$k \leftarrow k + 1$
Fimenquanto

QUADRO 1 : Algoritmo do método das potências

Ao final das iterações, $\lambda^{(k+1)} \approx \lambda_1$ e $q^{(k+1)} \approx v_1$.

3.2 Fundamentos teóricos

A base teórica que fundamenta este método como demonstrações, convergência e melhorias pode ser encontrada em materiais como [1] e [2].

Entre os tópicos importantes a serem destacados no que tange a isso, refere-se ao fato de que A é uma matriz coluna estocástica, isso é, suas entradas são todas não negativas e somatório das entradas pertencente a uma coluna é igual a 1 para toda coluna de A. Isso tem importantes implicações para o algoritmo. Uma delas garante que um autovalor associado a um autovetor de A não será maior do que 1.

4 Considerações e aprimoramento

Até o momento, algumas hipóteses foram consideradas sobre o conjunto de páginas *web,* as quais, certamente não são possíveis no mundo real. É óbvio, por exemplo, que nem todas as páginas são conectadas entre si. Um exemplo disso, seria imaginar o grafo da Figura 1 sem a aresta que conecta 4 e 3 partindo da página 4.
Isso implicaria em uma coluna de zeros em W o que prejudicaria o resultado final e, consequentemente, o *ranking*.

Outro fator complicador surge do fato da não unicidade dos *scores* gerados. O sistema ficaria indeciso em caso de empate de dois ou mais sites. Desse modo, uma pequena modifição faz-se necessária.

Sejam S uma matriz quadrada de ordem n cujas entradas são todas iguais a $1/n$ e $m \in [0,1)$. Então,

$$G = (1 - m)A + mS \quad (8)$$

É a matriz Google que gera um *ranking* satisfatório onde os problemas discutidos são sanados. Se $m = 0$, então $G = A$, e temos a matriz original. Por outro lado, se m se aproxima de 1, então G se aproxima de S, e todas as páginas tem probabilidade próximas em S.

É sabido que o valor do parâmetro m influênciará proporcionalmente no *rankeamento*, no entanto, tende a manter alta a importância de sites que já a possuem e não zerar valores para nós isolados.

Para o grafo da Figura 1, e $m = 0.15$ tem-se

$$G = \begin{bmatrix} 0.0375 & 0.3208 & 0.8875 & 0.0375 \\ 0.0375 & 0.0375 & 0.0375 & 0.8875 \\ 0.4625 & 0.3208 & 0.0375 & 0.0375 \\ 0.4625 & 0.3208 & 0.0375 & 0.0375 \end{bmatrix}. \quad (9)$$

5 IMPLEMENTAÇÃO

O *PageRank* foi implementado utilizado a linguagem *Matrix Laboratory – MatLab*. O programa foi modularizado da seguinte forma:

Módulo principal: Entrada da matriz de adjacência do grafo que representa o conjunto de páginas da *web* e suas conexões e parâmetros como ϵ e m;

Módulo 2: Geração da matriz de *hiperlinks* e associadas além do vetor inicial unitário $\boldsymbol{q}^{(\mathbf{0})}$;

Módulo 3: Método das Potências e geração dos *scores* requeridos.

6 RESULTADOS

O algoritmo foi executado sobre o exemplo com os parâmetros $m = 0.15$, $\epsilon = 10^{-6}$, $iter\_\max = 100$ e $\boldsymbol{q}^{(\mathbf{0})} = [1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$. Assim, foram retornados ao final das $num\_iter =$ 20 iterações, os seguintes resultados:

$$\lambda = 1 \text{ e}$$

$$\mathbf{v} = [0.5984 \quad 0.4665 \quad 0.4606 \quad 0.4606]^{\mathrm{T}} \quad (10)$$

Determinando que, para este conjunto de páginas, a página com rótulo igual a 1 é a mais importante, seguida da 2 e assim por diante.

7 CONCLUSÃO

Durante os testes realizados, pode-se notar que, a depender do vetor inicial, a convergência pode ser alcançada em uma quantidade maior ou menor de iterações. Isto é extremamente importante para este caso em particular, já que a empresa precisa garantir que o método convergirá e em tempo hábil para que uma resposta seja dada ao cliente como retorno à requisição de busca no motor da *Google*.

REFERÊNCIAS

[1] M. Andretta e F. Toledo, "Determinação numérica de autovalores e autovetores: Método das Potências," ICMC-USP.

[2] R. L. Burden, D. J. Faires e A. M. Burden, Análise Numérica, Cengage , 2016.

[3] T. Guizzani, "O Algoritmo do Google de PageRank e seu Cálculo através do Método das Potências," 2017.