# Como o Google sabe o que você está procurando

Hildebrando M. Rodrigues[1], Guilherme M. Alves[2], Matheus C. Nali[2], Victor T. B. Shime[2]

[1] Instituto de Ciências Matemáticas e de Computação
[2] Escola de Engenharia de São Carlos
Universidade de São Paulo
hmr@icmc.usp.br, guilherme.malacrida.alves@usp.br, matheus.nali@usp.br, victor.shime@usp.br

**Resumo**

*Este artigo é parte de um programa de iniciação cientifica, sob a orientação do professor Hildebrando Munhoz Rodrigues, no qual os alunos Guilherme Malacrida Alves, Matheus Carvalho Nali e Victor Takao Bernadino Shime estudaram o funcionamento de ferramentas de busca como o Google e seus algoritmos com destaque para o PageRank onde são explicados e ilustrados os fundamentos matemáticos utilizados em seu desenvolvimento.*

## 1. Introdução


**Figura 1:** *Ferramenta de busca*

Primeiramente, informamos que os resultados aqui encontrados não são novos e que o objetivo deste artigo é informativo, mostrando a importância da álgebra linear, teoria de matrizes e análise espectral no desenvolvimento do sistema de buscas do Google.

Agora, faremos um breve histórico sobre o Google [1]. Sergey Brin e Larry Page (Criadores do Google) conheceram-se na Universidade de Stanford enquanto estudavam para conseguir o título de doutores em ciência da computação. Naquela época, Page tinha um projeto louco de baixar toda a Internet no computador dele em apenas uma semana. Mas após um ano ele havia completado apenas uma porção dela. Bom, podemos dizer que Page estava bem otimista não?

Acontece que nesse projeto, Sergey se juntou a Page pois lhe interessava a mineração dos dados coletados e foi aí que eles começaram a trabalhar duro para desenvolver o Google. Em pouco tempo de desenvolvimento eram feitas por volta de 10 mil pesquisas por dia em Stanford. Dessa forma, eles perceberam que precisavam de mais computadores para dar suporte as pesquisas e, com isso, decidiram criar a empresa no Vale do Silício.

Porém, ainda que eles utilizassem um hardware de baixo custo, faltava dinheiro para comprar

os equipamentos. Com isso, um de seus professores (David Cheriton) entra em contato com Andy Bechtolsheim (Investidor e cofundador da Sun Microsystems) recomendando que conhecesse Sergey e Larry.

Andy conhece o projeto e se interessa bastante por ele acreditando no seu potencial de superar os buscadores que existiam na época (por exemplo o AltaVista). Além disso, ele admira a escolha de Sergey e Larry de investir o dinheiro em peças de baixo custo para construir os computadores ao invés de gastar com marketing ou equipamentos mais caros. Dessa forma, em 1998, Andy faz um cheque de $100,000 tornando-se o primeiro investidor do Google.

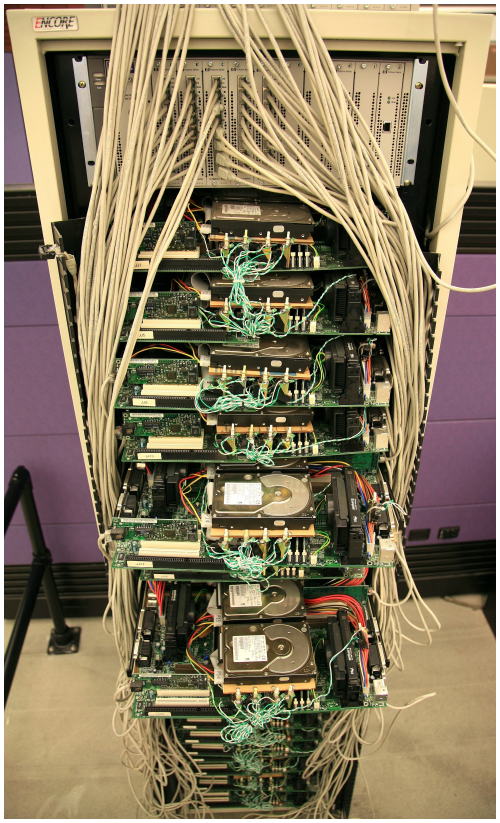


**Figura 2:** *Primeiro servidor de produção do Google - Construído com um hardware de baixo custo.*

Já notou que, em geral, sites de busca têm o que você procura nos primeiros resultados? Ou que uma certa enciclopédia virtual está sempre no topo dos resultados da sua busca? Isso não é nenhum tipo de magia negra, é simplesmente matemática (mais especificamente a álgebra linear) e computação.

Agora como isso funciona? Um dos fatores analisados que vamos abordar nesse artigo é a classificação em importância de cada página da web relacionada com a sua busca utilizando um método chamado de "PageRank", que basicamente atribui um valor a cada página representando o quão relevante ela é entre todas as páginas encontradas. Sabemos que a internet possui uma enorme quantidade de dados, por isso, para que fosse possível analisar todas essas informações de maneira eficiente e eficaz foi necessário encontrar um método que decidisse quais páginas são mais importantes sem que uma pessoa tivesse que ler todas elas e classificar, algo que além de ineficiente, depende fortemente das opiniões pessoais da pessoa que faz a classificação.

A solução para isso foi fazer com que as páginas decidissem qual delas é a mais importante através dos links entre elas, que são a maneira com que as páginas se relacionam. O algoritmo "PageRank" utiliza a lógica de que a importância de uma página é definida pelas páginas que fazem links para ela. Uma descrição mais detalhada desse método pode ser vista em [2].

Posteriormente várias modificações foram feitas no algoritmo do "PageRank"para aumentar a sua eficiência, como as propostas por Golub [5]. Ele notou que para as páginas de menor importância se obtinha o seu valor rapidamente, enquanto as de maior importância demoravam mais. Com isso ele propôs que em certo ponto se parasse de calcular as páginas menos importantes, já que seu valor estava próximo o bastante do real. Desta forma foi possível aumentar a velocidade do algoritmo em quase 30%. Por simplicidade não utilizaremos essa modificação aqui.

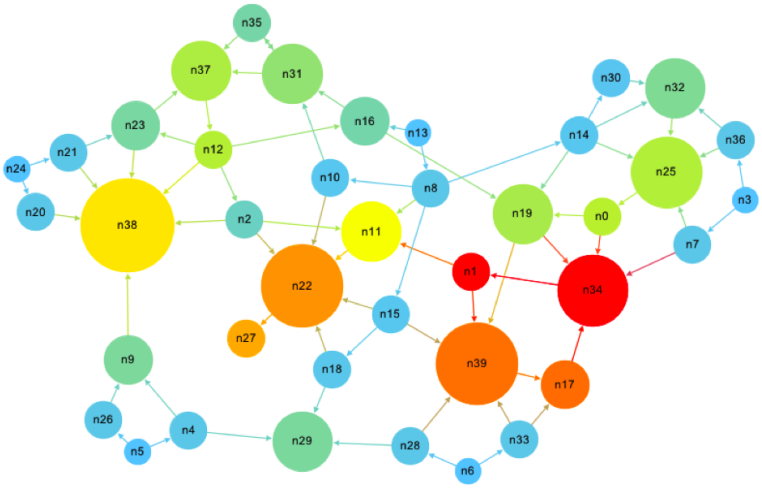


**Figura 3:** *Grafo representando os links entre páginas*

## 2. Equacionamento

Para ilustrar o equacionamento, começaremos com um exemplo de uma rede com quatro páginas (1,2,3,4) cujos links estão descritos na imagem abaixo. A importância de cada página serão denotadas por $x_1, x_2, x_3, x_4$ respectivamente.

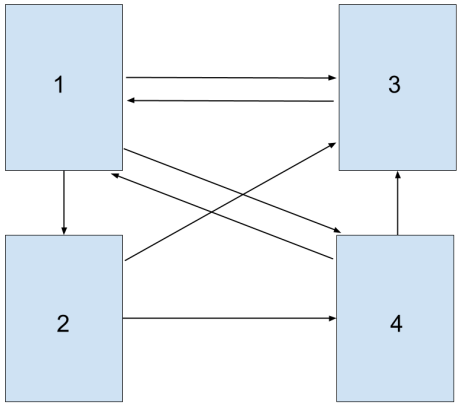


**Figura 4:** *Rede de páginas com setas indicando os links*

Uma maneira simples de interpretar o algoritmo é dizer que uma página tem uma quantia de votos igual a sua importância e que divide esses votos igualmente entre as páginas para as quais ela tem links. A importância de uma página é então a soma de todos os votos que ela recebe. Resumindo em uma equação, onde $n_j$ é o número de links da página $x_j$, temos:

$$x_i = \sum_{j \neq i} \frac{1}{n_j} x_j$$

Para o caso da figura anterior podemos escrever as importancias como sendo:

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

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Usaremos a notação com matrizes para simplificar na hora de escrever:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

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

Esse sistema é um tipo de problema chamado de problema de autovalor. Problemas de autovalor são equações do tipo $Mx = \lambda x$ onde M é uma matriz e $\lambda$ é uma constante, e ele pode ter uma única solução onde todas as componentes de x são zero ou infinitas soluções, a depender de M e

$\lambda$. Caso tenha infinitas soluções dizemos que $\lambda$ é um autovalor de M e os valores de x, diferentes de 0, que satisfazem são autovetores.

Sabendo disso, para que o sistema tenha solução, precisamos garantir que 1 seja autovalor da matriz **A** que construímos com os links da nossa rede, vamos usar o resultado abaixo e o teorema de Perron-Frobenius.

TEOREMA: Uma matriz A cujas entradas são todas não negativas e as colunas somam 1 (coincidentemente, ou não, como a que construímos) tem 1 como autovalor. Como é demonstrado em [2].

Com esse teorema sabemos que o nosso problema tem uma solução que não é zero. Mas ainda resta fazer com que obtenhamos uma solução única para ele, porque computadores não podem calcular infinitas soluções e não queremos ter resultados diferentes a cada vez que fazemos uma busca. O teorema de Peron-Frobenius nos dá uma solução para isso mas primeiro, vejamos algumas definições úteis.

## 3. Definições úteis

- Raio espectral: Para esse caso em específico, adotar que ele é o maior módulo entre os módulos dos autovetores da matriz, que é uma maneira simplificada de definir o raio espectral.
- Matriz não negativa: Uma matriz com todos os elementos maiores ou iguais a zero.
- Matriz positiva: Uma matriz com todos os elementos maiores do que zero.

## 4. Teorema de Perron-Frobenius

TEOREMA DE PERRON-FROBENIUS: Seja **M** uma matriz quadrada não negativa, o raio espectral de **M**, $\rho(\mathbf{M})$, é o maior autovalor em módulo e tem autovetor não negativo. Além de que, nenhum outro autovalor de **M** é maior em valor absoluto que $\rho(\mathbf{M})$ nem tem autovetor positivo. Se **M** for positiva teremos ainda que o autovalor é simples. [4]

Agora vamos às explicações do teorema. Qual é a utilidade desse teorema? Primeiro, ele mostra que existe um único autovalor com autovetor positivo, logo todos os autovetores positivos são múltiplos um dos outros, então, basta escolher qualquer um deles, o com a soma das suas entradas igual a 1 por exemplo (é importante ter um critério bem definido, para não gerar ambiguidades), e temos uma solução única a ser encontrada. O fato de ter uma solução única é essencial em aplicações computacionais para que elas sejam viáveis em grande escala com processamento rápido. Sem ele não seria possível lidar com todas as páginas da internet e seria muito difícil encontrar qualquer coisa, por mais simples que seja, sem saber o endereço da página.

Segundo, alguns problemas podem surgir no cálculo desse autovetor, como páginas que não recebem nenhum link (dangling nodes) e grupos de páginas que não tem ligações entre si. Essas situações geram problemas com a unicidade da solução que podem ser solucionados ao fazer algumas alterações na matriz **A** (Esses casos são mais detalhados em [2]). No caso dos grupos de páginas sem ligações entre si a solução é criar uma nova matriz a partir de **A** que tem todas as entradas positivas, mas que ainda tem soma 1 nas colunas, de forma que a segunda parte do teorema garante que a solução é única. O caso de páginas que não recebem links é mais

problemático de ser tratado, caso você não queria simplesmente excluí-la da sua lista.

Último, esses resultados não são tão imediatos quanto está escrito aqui, ou os criadores do Google não teriam ficado ricos com ele. Exemplificando a importância disso temos o fato de que o PageRank de 26 milhões de páginas pode ser calculado em algumas horas por uma workstation mediana como citado pelos próprios criadores do Google em [3], considerando o poder computacional da época em que o resultado foi publicado.

## 5. De volta ao exemplo

Agora, para resolver o problema do autovalor para o exemplo dado, calculamos os autovetores da matriz **A** correspondentes ao autovalor 1. Voltando ao sistema construído e deixando a resposta em função de $x_1$ .

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

O primeiro valor é obtido imediatamente de $x_2 = \frac{1}{3}x_1$, em seguida temos:

$$x_4 = \frac{1}{3}x_1 + \frac{1}{2}x_2 = \frac{1}{3}x_1 + \frac{1}{6}x_1 = \frac{1}{2}x_1$$

E por fim:

$$x_3 = \frac{1}{3}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_4 = \frac{1}{3}x_1 + \frac{1}{6}x_1 + \frac{1}{4}x_1 = \frac{3}{4}x_1$$

O que resulta em:

$$\implies \begin{cases} x_1 = x_1 \\ x_2 = \frac{1}{3}x_1 \\ x_3 = \frac{3}{4}x_1 \\ x_4 = \frac{1}{2}x_1 \end{cases}$$

Se convenientemente adotarmos $x_1 = 12$, obtemos: [12 4 9 6]. Normalizando esses valores para que a soma seja 1, obtemos o novo autovetor (lembrando que qualquer múltiplo não nulo de um autovetor também é um autovetor).

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \approx \begin{bmatrix} 0.39 \\ 0.13 \\ 0.29 \\ 0.19 \end{bmatrix}$$

Com isso, a página mais importante é a 1! Mas por que não a 3? a página 3 recebe links de todas as outras 3 páginas como podemos ver pela imagem. O que acontece é que embora a página 3 receba links das outras, ela faz link apenas com a página 1 o que impacta no seu resultado de importância. É importante lembrar que a importância está associada com a relação entre as páginas, ou seja, com os links que saem e chegam nela e, neste caso, toda a importância da página 3 é repassada à página 1, que é a única que recebe link de 3. Sendo assim, a 1 se mostra a mais importante, pois tem que ter ao menos a mesma importância da 3 como comprovam os cálculos.

## 6. Conclusão

Com isso, concluímos que utilizando teoremas, modelos convenientes e adaptações necessárias para determinados casos é possível analisar e classificar as páginas de uma busca. Além disso, é fundamental a otimização dos algoritmos para melhorar o uso dos recursos computacionais já que estamos falando de uma enorme quantidade de páginas analisadas.

Por fim, é importante compreender que o PageRank classifica as páginas de acordo com a **importância** delas para determinada busca e que muitos outros algoritmos também são utilizados no processo de busca de forma a tentar apresentar o resultado que deseja na primeira página mostrada.

Outros algoritmos importantes na classificação de páginas:

- Hilltop Algorithm
- Topic-Sensitive PageRank

## Referências

[1] Vise, D. A., Inside the Hottest Bussiness, Media and Technology Sucess of Our Time, PAN, 2005.

[2] Bryan, K., & Leise, T., The $25,000,000,000 Eigenvector:The Linear Algebra Behind Google, Society for Industrial and Applied Mathematics, 2006.

[3] BRIN, S., & PAGE, L.,The anatomy of a large-scale Hypertextual search Engine. http://www-db.stanford.edu/ backrub/google.html

[4] KATO, T., A short introduction to linear perturbation theory, Springer-Verlag, 1982.

[5] KAMVAR, S., HAVELIWALA, T., GOLUB, G., Adaptive methods for the computation of PageRank, Linear Algebra and its Applications 386 (2004) 51–65