Relatório do Modelo de Ordenamento de Páginas do Google

Amanda Costa amanda.cs@usp.br
João Pedro de Freitas jpfreitas2001@usp.br
Juan Nogueira juan.nog@usp.br
Maria Rita Xavier mrxavier74@usp.br
Octavio Augusto Potalej oapotalej@usp.br
Samuel Garcez samueltgarcez@usp.br

Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo

O relatório objetiva detalhar os conceitos matemáticos e computacionais que estão envolvidos no processo de organização de páginas que o sistema de pesquisa do Google utiliza para mostrar resultados.

1 Introdução

Desde seu desenvolvimento na década de 90, o Google destacou-se dentre os outros ambientes de pesquisa pela eficiência do seu sistema de ranqueamento de páginas. Esse processo sobressai quanto aos demais pela sua precisão em mostrar, primeiramente, as páginas que serão mais úteis para o usuário. Isso ocorre através de três etapas básicas que geram os resultados das páginas: o rastreamento, a indexação e a veiculação.

Primeiro, de acordo com a Central de Pesquisa do Google, o rastreamento consiste na procura constante do Google por URLs. Através disso, é possível catalogar os sites. Assim, o conteúdo textual e não textual e o layout dos sites são analisados para aparecerem nos resultados de pesquisa apropriados. Quanto mais o Google entender o site, melhor ele poderá qualificá-lo nas pesquisas.

Em seguida, acontece a indexação. Essa etapa consiste na análise que o Google faz nas páginas, a fim de identificar o conteúdo tratado e relacioná-lo com as pesquisas que tratam do mesmo assunto. Feita a análise, as informações são armazenadas em um grande banco de dados, denominado de Índice do Google.

Por fim, ocorre a veiculação (também chamada de exibição e classificação). Nessa etapa, o Google procura no Índice pelos resultados mais adequados e altamente qualificados para os assuntos pesquisados, de acordo com determinados aspectos.

Dessa forma, após todas as etapas serem realizadas, é preciso organizar as páginas das mais às menos importantes. Assim, o seguinte relatório detalha, com base em critérios matemáticos e computacionais, o modelo de funcionamento que conduz os aspectos de classificação da relevância das páginas apresentadas na pesquisa feita pelo usuário.

2 O Modelo

Tendo-se n páginas, enumeremo-as de 1 a n e representemos suas importâncias por um número real x_i, $i \in \{1, 2, \ldots, n-1, n\}$.

Inicialmente, façamos um sistema linear que representa a relação entre as páginas da seguinte forma:

$$\begin{cases} x_1 = p_{11}\dfrac{x_1}{q_1} + p_{12}\dfrac{x_2}{q_2} + \ldots + p_{1n}\dfrac{x_n}{q_n} \\ \\ x_2 = p_{21}\dfrac{x_1}{q_1} + p_{22}\dfrac{x_2}{q_2} + \ldots + p_{2n}\dfrac{x_n}{q_n} \\ \vdots \qquad \vdots \qquad\quad \vdots \qquad\quad \ldots \qquad\quad \vdots \\ x_n = p_{n1}\dfrac{x_1}{q_1} + p_{n2}\dfrac{x_2}{q_2} + \ldots + p_{nn}\dfrac{x_n}{q_n} \end{cases}$$

Onde definiremos as duas constantes, p_{ij} e q_j, como sendo:

$$p_{ij} = \begin{cases} 1, \text{ se há links de } j \text{ para } i \text{ e se } i \neq j. \\ 0, \text{ se não há links de } j \text{ para } i \text{ ou se } i = j. \end{cases}$$

e

q_j = quantidade de links de j que redirecionam para outras páginas.

Ou, em forma matricial:

$$\begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \frac{p_{11}}{q_1} & \frac{p_{12}}{q_2} & \cdots & \frac{p_{1n}}{q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{p_{n1}}{q_1} & \frac{p_{n2}}{q_2} & \cdots & \frac{p_{nn}}{q_n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$

Escrevendo, ainda, a equação matricial como $\mathbf{x} = \mathbf{Ax}$, chamaremos a matriz $\mathbf{A}$ de matriz de ligação.

Por construção, a soma de cada coluna da matriz de ligação será igual a 1, o que garante que o sistema sempre tenha solução. Queremos, ainda, garantir a existência de uma solução de componentes positivas e que qualquer outra solução seja múltipla dessa, para que possamos normalizar a solução para garantir a existência de importâncias x_i únicas.

De acordo com o teorema de Perron-Frobenius, se $\mathbf{M}$ é uma matriz $n \times n$ com todos os elementos positivos ($\mathbf{M}_{ij} > 0, \forall i, j \in 1, ..., n$) e colunas com soma 1 ($\sum_{i=1}^{n} \mathbf{M}_{ij} = 1, \forall$ j), então 1 é o autovalor de maior módulo de $\mathbf{M}$, seu auto-espaço é unidimensional e o autovetor normalizado só tem entradas estritamente positivas. Entretanto, a matriz de ligação $\mathbf{A}$, no geral, não tem a hipótese do teorema verdadeira. Definindo $m \in]0, 1[$, usualmente adotado como 0.15 e a matriz $\mathbf{S}$ com todas as entradas iguais a $\frac{1}{n}$, criemos, então, a matriz $\mathbf{M}$, que chamaremos de matriz do problema perturbado.

$$\mathbf{M} = (1 - m) \cdot \mathbf{A} + m \cdot \mathbf{S} = 0.85 \cdot \mathbf{A} + 0.15 \cdot \mathbf{S}$$

Essa correção feita na matriz $\mathbf{A}$ torna todas as entradas de $\mathbf{M}$ estritamente positivas e a soma dos elementos em cada coluna continua sendo igual a 1, satisfazendo as hipóteses do teorema de Perron-Frobenius.

A partir de $\mathbf{M}$, acharemos o vetor $\mathbf{x}$ computacionalmente com o seguinte algoritmo:

Primeiramente escolhemos um vetor $x^{(0)}$ qualquer, que seja normalizado e todo positivo ($x_i^{(0)} > 0, 1 \leq i \leq n$). Então, aplicando o método das potências para valores de $k = 1, 2, 3, ...$

$$x^{(k)} = Mx^{(k-1)}$$

Vale ressaltar que M deve ser uma matriz que satisfaça as hipóteses do Teorema de Perron-Frobenius, para que x convirja para o autovetor normalizado associado ao autovalor 1 com a taxa:

$$c = max_j |1 - min_i M_{ij}|,$$

sendo $1 \leq j \leq n$ e $1 \leq i \leq n$.

Agora, conseguimos calcular a estimativa do erro a posteriori pela fórmula:

$$||x - x^{(k)}|| \leq \frac{c}{1 - c} ||x^{(k)} - x^{(k-1)}||$$

Para definitivamente calcularmos x, precisamos encontrar um valor extremamente pequeno para a estimativa do erro, assim, chegando em uma aproximação do valor real de x.

A fim de realizar as operações do algoritmo descrito, será necessário garantir após as transformações do tipo $x^{(k)} = \mathrm{M}x^{(k-1)}$ o vetor continuará sendo normalizado e com todas as entradas positivas.

2.1 Propriedades da transformação por M

Mostremos que, sendo $\mathbf{M}$ uma matriz $n \times n$ que satisfaz o teorema de Perron-Frobenius, se $\mathbf{y}$ é um vetor com todas as entradas positivas e normalizado, então $\mathbf{z} = \mathbf{My}$ será também um vetor com todas as entradas positivas e normalizado.

Sabemos que $\mathbf{M}$ é dada por $\mathbf{M} = (1 - m)\mathbf{A} + m\mathbf{S}$, com $0 < m < 1$ e $\mathbf{S}$ a matriz $n \times n$ com todas as entradas iguais a $1/n$.

Seja $\mathbf{z} = \mathbf{My}$. Então,

$$z_i = m_{i1}y_1 + m_{i2}y_2 + m_{i3}y_3 + ... + m_{in}y_n$$

e o módulo $|z| = \sqrt{z_1^2 + z_2^2 + z_3^2 + ... + z_n^2}$.

Se todas as entradas de $\mathbf{M}$ são positivas, os sinais de $\mathbf{y}$ serão mantidos. Portanto a soma que gera a norma será composta de termos com os sinais dos elementos de $\mathbf{y}$, que são positivos. Por isso, todos os elementos de $\mathbf{z}$ serão somas de termos estritamente positivos.

$$|z| = \sqrt{\sum_{i=1}^{\mathrm{n}} \left(\sum_{j=1}^{\mathrm{n}} (m_{ij}y_j) \right)^2}$$

$$|z|^2 = \sum_{i=1}^{\mathrm{n}} \left(\sum_{j=1}^{\mathrm{n}} (m_{ij}y_j) \right)^2$$

Cada termo da soma acima é o produto entre cada elemento de y_i pela soma dos elementos da coluna j. Mas segundo o teorema, a soma dos elementos de uma coluna de $\mathbf{M}$ é igual a 1.

2.1.1 Desigualdade de Cauchy-Schwarz

Sejam $a_1, a_2, \ldots, a_n$ e $b_1, b_2, \ldots, b_n$ números reais. Então,

$$\left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right) \geq \left(\sum_{i=1}^{n} a_i b_i\right)^2$$

Demonstração:
Consideremos o trinômio quadrado

$$\sum_{i=1}^{n}(a_i - b_i x)^2 = \sum_{i=1}^{n}(a_i^2 - 2a_i b_i x + b_i^2 x^2) =$$

$$= \sum_{i=1}^{n} a_i^2 - 2x\sum_{i=1}^{n} a_i b_i + x^2 \sum_{i=1}^{n} b_i^2$$

Como

$$\sum_{i=1}^{n}(a_i - b_i x)^2 \geq 0$$

Então o delta da Equação de segundo grau deve ser não positivo:

$$\left(-2\sum_{i=1}^{n} a_i b_i\right)^2 - 4\left(\sum_{i=1}^{n} b_i^2\right)\left(\sum_{i=1}^{n} a_i^2\right) \leq 0$$

$$\cancel{4}\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leq \cancel{4}\left(\sum_{i=1}^{n} b_i^2\right)\left(\sum_{i=1}^{n} a_i^2\right)$$

$$\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{n} b_i^2\right)\left(\sum_{i=1}^{n} a_i^2\right)$$

A igualdade ocorre se, e somente se, $a_i - b_i x = 0$, para todo $i = 1, 2, \ldots, n$, isto é, se, e somente se, $x = \dfrac{a_i}{b_i}$, para todo $i = 1, 2, \ldots, n$, o que é equivalente a

$$x = \frac{a_1}{b_1} = \frac{a_2}{b_2} = \ldots = \frac{a_n}{b_n}$$

Portanto, voltando ao problema de z:
Uma vez que $|\mathbf{y}| = 1$ e $|\mathbf{M}_j| = 1$, e tomando em conta que a desigualdade de Cauchy Schwarz garante que dados dois vetores, u e v quaisquer, o produto interno entre u e v será:

$$u \cdot v \leq |u||v|$$

No entanto, se u v são vetores Linearmente Dependentes (um pode ser escrito como o outro multiplicado por um escalar).

$$u \cdot v = |u||v|$$

E, portanto o resultado da somatória e a norma de z é igual a 1.

$$z = \sqrt{\sum_{i=1}^{n}(m_{ij} y_i)^2} = \sqrt{1} = 1$$

E isto garante que **z** seja um vetor normalizado.

2.2 A forma do vetor transformado

Para chegar a uma fórumla para o vetor $\mathbf{My}$, resultante da transformação de $\mathbf{y}$ por $\mathbf{M}$, será conveniente utilizar a propriedade distributiva pela direita do produto de matrizes. Mostremos, portanto, que os elementos correspondentes das matrizes $(\mathbf{A}+\mathbf{B})\mathbf{C}$ e $\mathbf{AC}+\mathbf{BC}$ são iguais, como se segue.

Notação: $[\mathbf{M}]_{ij}$ denota o elemento da linha i e coluna j da matriz $\mathbf{M}$.

$$[(\mathbf{A}+\mathbf{B})\mathbf{C}]_{ij} = \sum_k [\mathbf{A}+\mathbf{B}]_{ik}[\mathbf{C}]_{kj}$$

$$= \sum_k ([\mathbf{A}]_{ik} + [\mathbf{B}]_{ik})[\mathbf{C}]_{kj} = \sum_k ([\mathbf{A}]_{ik}[\mathbf{C}]_{kj} + [\mathbf{B}]_{ik}[\mathbf{C}]_{kj})$$

$$= \sum_k [\mathbf{A}]_{ik}[\mathbf{C}]_{kj} + \sum_k [\mathbf{B}]_{ik}[\mathbf{C}]_{kj} = [\mathbf{AC}]_{ij} + [\mathbf{BC}]_{ij}$$

$$= [\mathbf{AC}+\mathbf{BC}]_{ij}$$

Assim, podemos multiplicar $\mathbf{M}$ por um vetor $\mathbf{y}$:

$$\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S}$$
$$\Rightarrow \mathbf{My} = ((1-m)\mathbf{A} + m\mathbf{S}) \cdot \mathbf{y}$$
$$\Rightarrow \mathbf{My} = (1-m)\mathbf{Ay} + m\mathbf{Sy}$$

Sobre o vetor $\mathbf{y}$, impomos a condição de suas entradas somarem 1, isto é, $\mathbf{y} = (y_1, y_2, \ldots, y_n)^t$ tal que

$$y_1 + y_2 + \ldots + y_n = 1$$

Neste caso, relembrando que a matriz $\mathbf{S}$ é definida pelos elementos iguais a $\frac{1}{n}$, temos

$$\mathbf{Sy} = \begin{bmatrix} \frac{1}{n} & \ldots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \ldots & \frac{1}{n} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_1\frac{1}{n} + y_2\frac{1}{n} + \ldots + y_n\frac{1}{n} \\ \vdots \\ y_1\frac{1}{n} + y_2\frac{1}{n} + \ldots + y_n\frac{1}{n} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{1}{n}(y_1 + y_2 + \ldots + y_n) \\ \vdots \\ \frac{1}{n}\underbrace{(y_1 + y_2 + \ldots + y_n)}_{=1} \end{bmatrix} = \begin{bmatrix} \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{bmatrix}$$

Logo,

$$\mathbf{My} = (1-m)\mathbf{Ay} + m\mathbf{Sy} = (1-m)\mathbf{Ay} + m \begin{pmatrix} 1/n \\ \vdots \\ 1/n \end{pmatrix}$$

Em particular, se $\mathbf{y}$ tem cada entrada igual a $\frac{1}{n}$, então a condição de suas entradas somarem 1 é respeitada e vale a expressão acima.

3 O Programa

O programa desenvolvido requer apenas uma entrada, que é dada pelas relações entre as páginas, constituindo a matriz $\mathbf{A}$. É possível escolher entre usar uma das situações de exemplo ou inserir sua própria:

```
Vamos definir a matriz de ligação A de acordo com o
problema que se deseja explorar:
1 - Exemplo do enunciado, com 4 páginas.
2 - Exemplo do Exercício 3, com 8 páginas.
3 - Criar novo exemplo.
Digite a opção correspondente:
```

Nesse último caso, o processo de constituição de $\mathbf{A}$ se dá através da função `insercaoDoUsuario`, que requisita a entrada de quais páginas mencionam determinada páginas através de um loop **`for`**, e inseridos os valores, a função calcula cada elemento correspondente de $\mathbf{A}$:

```
def insercaoDoUsuario():
    # entrada do usuário
    ...

    # A matriz "A" é instanciada como uma matriz n x
    ↪  n nula
    for linha in range(len(A)):
        ...
        # R é uma lista com os valores inseridos
        for item in R:
            coluna = int(float(item)) - 1
            if coluna == linha: A[linha][coluna] = 0
            else: A[linha][coluna] = 1

    # os elementos de uma coluna são divididos pela
    ↪  quantidade de elementos não nulos dela.
    for coluna in range(len(A[0])):
        contador = 0
        for linha in range(len(A)):
            if A[linha][coluna] != 0:
                contador += 1
        if contador != 0:
            for linha_aux in range(len(A)):
                A[linha_aux][coluna] =
                ↪  A[linha_aux][coluna]/contador

    return A
```

Tendo $\mathbf{A}$, há três padrões estabelecidos:

1) A variável m, adotada por padrão como $m = 0.15$;
2) Uma margem aceitável de erro (para que o programa seja encerrado em determinado momento), estabelecida como 10^{-5};
3) O vetor inicial $\mathbf{x^{(0)}}$, estabelecido como uma matriz de apenas uma coluna com valores $\frac{1}{n}$

Há uma classe central denominada `Paginas` que recebe em seu instanciamento $\mathbf{A}$ e m. Esta possui um método denominado `X` que recebe a margem considerada e retorna, em lista, a lista de iterações para x, a lista de erro para cada iteração e a constante c.

O método `X` é definido como:

```
def X(self, margem):
    # chute inicial
    self.x0 = self.chuteInicial()
    # matriz perturbada
    self.M = self.Mat.calculoM(self.m, self.A)

    # constante de erro
    self.c_ = self.c()

    [ autovetor_iteracoes, listaErro ] =
    ↪  self.autovetorDominante(margem)

    return [autovetor_iteracoes, listaErro, self.c_]
```

O método `chuteInicial()` gera o vetor $\mathbf{x^{(0)}}$, enquanto o método `calculoM`, da classe `Matrizes` (definida no contexto como `Mat`), calcula a matriz perturbada de $\mathbf{A}$, $\mathbf{M}$, a partir da definição apresentada anteriormente.

```
def calculoM(self, m, A):
    S = []
    for i in range(len(A)):
        linha = []
        for j in range(len(A[i])):
            linha.append(1/len(A))
        S.append(linha)

    m1 = self.produtoNumeroReal(A, 1-m) # (1-m)A
    m2 = self.produtoNumeroReal(S, m) # mS
    M = self.somaMatrizes(m1, m2)
    return M
```

Tendo calculado $\mathbf{M}$, o programa também encontra a constante c a partir da definição apresentada:

```
def c(self):
    c = -1
    for coluna in range(len(self.M[0])):
        min_ = self.M[0][coluna]

    for linha in range(len(self.M)):
        if min_ > self.M[linha][coluna]:
            min_ = self.M[linha][coluna]
```

```
    mod = 1 - 2*min_
    if mod < 0: mod = (-1)*mod

    if c < mod: c = mod

    return c
```

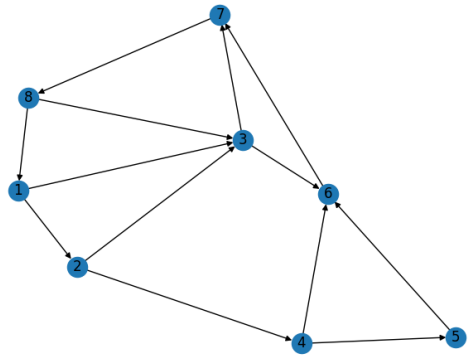
Assim, tendo a margem, $\mathbf{M}$ e c, é chamado o método `autovetorDominante`, que retorna as iterações de $\mathbf{x}$ e o erro para cada iteração:

```
def autovetorDominante(self, margem):
    autovetor_iteracoes = [self.x0] # vetor X
    erro = 10**8 # erro absurdo inicial
    listaErro = [] # lista onde serão armazenados os
    ↪  erros

    # calcula até o erro ficar menor que a margem
    ↪  oferecida
    while erro > margem:
        # calcula o X e o adiciona na lista de X
        xi = self.Mat.produtoMatriz(self.M,
        ↪  autovetor_iteracoes[-1])
        autovetor_iteracoes.append(xi)
        # calcula o erro e o adiciona na lista de
        ↪  erros
        erro =
        ↪  self.calculoErro(autovetor_iteracoes[-1],
        ↪  autovetor_iteracoes[-2])
        listaErro.append(erro)

    return [autovetor_iteracoes, listaErro]
```

Por fim, é exibido o calculado até então:

```
Para a matriz informada, o autovetor dominante calculado é:

[0.10564275444167291]
[0.06364814360535095]
[0.17759130945825363]
[0.04580041141122971]
[0.038215212250709084]
[0.14617444325866877]
[0.2184744662728349]
[0.20445325930127956]


Com erro estimado em:
 9.055351235832498e-06

E a constante c para a matriz informada é:
 0.9625

* Rankeamento das páginas *

 - Posição 1: Página 7
 - Posição 2: Página 8
 - Posição 3: Página 3
 - Posição 4: Página 6
 - Posição 5: Página 1
 - Posição 6: Página 2
 - Posição 7: Página 4
 - Posição 8: Página 5
```

É gerado, ainda, um grafo com a situação apresentada. Neste caso, foi considerada a seguinte situação de páginas:



O código completo pode ser encontrado *neste repositório*.

Referências

[1] Google. Adicionar, excluir e organizar páginas. [Online]. Available: https://support.google.com/sites/answer/98216?hl=pt-BR

[2] K. Bryan and T. Leise, "The $25,000,000,000 eigenvector: The linear algebra behind google," vol. 48, no. 3, pp. 569–581, 2006.

[3] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.