

## Localização de Autovalores de Matrizes Laplacianas Perturbadas de Árvores

Rodrigo O. Braga<sup>1</sup>

Ciências Exatas e Tecnológicas, UNISINOS, São Leopoldo, RS

Virgínia M. Rodrigues<sup>2</sup>

Instituto de Matemática, UFRGS, Porto Alegre, RS

**Resumo.** Neste trabalho, apresentamos um algoritmo de tempo linear que permite localizar, num dado intervalo real, o número de autovalores de uma matriz laplaciana perturbada qualquer associada a uma árvore. Este algoritmo pode ser aplicado a árvores com ou sem pesos. Utilizando este procedimento, obtemos uma caracterização das árvores com até cinco autovalores distintos para uma família de matrizes laplacianas perturbadas, que inclui a matriz de adjacências e a matriz laplaciana normalizada como casos particulares.

**Palavras-chave.** Matriz Laplaciana Perturbada, Localização de Autovalores, Árvores.

### 1 Introdução

A *Teoria Espectral de Grafos* estuda a relação existente entre o espectro de matrizes associadas a grafos e propriedades estruturais dos grafos. A matriz mais comumente utilizada para representar um grafo é a *matriz de adjacências*. Se  $G$  é um grafo simples não orientado com vértices  $v_1, \dots, v_n$ , a matriz de adjacências  $A = (a_{ij})$  de  $G$  é a matriz real simétrica de ordem  $n$  com entradas 0 ou 1, definida por  $a_{ij} = 1$  se, e somente se, existe uma aresta conectando os vértices  $v_i$  e  $v_j$ , onde  $1 \leq i, j \leq n$ ,  $i \neq j$ .

Uma *árvore* é um grafo conexo e sem ciclos. Em [7], Jacobs e Trevisan apresentaram um algoritmo para calcular o número de autovalores de uma árvore em um dado intervalo real. Esse método tem como característica importante percorrer os vértices da árvore sem necessidade do uso da matriz explicitamente. Nesse artigo, os autores também observaram que o procedimento teria potencial para ser adaptado para outras matrizes, por exemplo para a *matriz laplaciana combinatória*, ou seja, a matriz  $L = D - A$ , onde  $D$  é a matriz diagonal cuja entrada  $ii$  da diagonal é o grau do vértice  $v_i$  de um grafo  $G$  e  $A$  é a matriz de adjacências de  $G$ .

O algoritmo de Jacobs e Trevisan e suas extensões que foram sendo desenvolvidas têm-se mostrado uma ferramenta prática e eficiente em Teoria Espectral de Grafos. Em particular, podemos destacar o trabalho de Fritscher *et al.* [5], em que foi apresentada pela primeira vez uma adaptação do algoritmo para a matriz laplaciana combinatória, utilizada

---

<sup>1</sup>rodrigob@unisinovs.br

<sup>2</sup>vrodrig@mat.ufrgs.br

na demonstração de que, dentre todas as árvores de  $n$  vértices, a árvore com energia laplaciana máxima é a estrela  $S_n$ , solucionando positivamente a conjectura de Radenković e Gutman em [8]. Já em [3], Braga *et al.* adaptaram o algoritmo de Jacobs e Trevisan para a *matriz laplaciana normalizada* de uma árvore, introduzida por Chung [4]. Os autores em [3] utilizaram o algoritmo para estudar a multiplicidade dos autovalores da matriz laplaciana normalizada de uma árvore com diâmetro menor ou igual a 4, e caracterizaram todas as árvores com até cinco autovalores laplacianos normalizados distintos.

A diversidade de resultados obtidos com a utilização do algoritmo para diferentes matrizes motivou-nos no desenvolvimento de um algoritmo para localizar os autovalores de uma *matriz laplaciana perturbada* qualquer associada a uma árvore, em um intervalo real dado, que apresentamos neste trabalho.

Um grafo é dito *com pesos* se a cada aresta  $e_{ij}$  ligando o vértice  $v_i$  ao vértice  $v_j$  é associado um número real  $\omega(e_{ij}) = \omega_{ij}$ , denominado *peso* da aresta  $e_{ij}$ . Um grafo sem pesos pode ser entendido como um grafo com pesos onde todas as arestas têm peso 1. Em [1], Bapat *et al.* definiram a matriz laplaciana perturbada de um grafo  $G$  com pesos positivos, que tem como casos particulares a matriz de adjacências, a matriz laplaciana combinatória e a matriz laplaciana normalizada, entre outras, generalizando assim essas matrizes. Dada uma matriz diagonal real  $D$ , a matriz laplaciana perturbada de  $G$  com respeito a  $D$ , denotada por  $\mathcal{L}(G)$ , é a matriz  $\mathcal{L}(G) = D - A(G)$ , onde  $A(G)$  é a matriz de adjacências de  $G$ , isto é,  $A(G) = (a_{ij})$ , onde

$$a_{ij} = \begin{cases} \omega_{ij}, & \text{se } v_i v_j \text{ é uma aresta} \\ 0, & \text{caso contrário.} \end{cases}$$

Além disso, definimos o *peso* do vértice  $v_i$  como sendo a entrada  $ii$  da matriz diagonal  $D$ .

O algoritmo que desenvolvemos para a matriz laplaciana perturbada, além de preservar a praticidade do algoritmo de Jacobs e Trevisan e suas extensões, tem a vantagem de poder ser utilizado para obter-se simultaneamente resultados para diversas matrizes associadas a uma árvore, além de poder-se trabalhar com árvores com pesos, o que não era possível até então.

## 2 Algoritmo de Localização de Autovalores de Matrizes Laplacianas Perturbadas

Tendo como entradas uma árvore com pesos  $T$  de ordem  $n$ , um escalar  $\alpha \in \mathbb{R}$  e uma matriz diagonal real  $D$ , o algoritmo que apresentamos nesta seção faz a diagonalização da matriz  $\mathcal{L}(T) + \alpha I$ , onde  $\mathcal{L}(T)$  é a matriz laplaciana perturbada de  $T$  com respeito à matriz  $D$ . O algoritmo tem como saída uma matriz diagonal  $D_\alpha$  *congruente* à matriz  $\mathcal{L}(T) + \alpha I$ . Lembramos que duas matrizes quadradas de ordem  $n$ ,  $A$  e  $B$ , são congruentes se existe uma matriz invertível  $P$  tal que  $A = P^T B P$ .

Assim como no algoritmo de Jacobs e Trevisan para a matriz de adjacências, o algoritmo que descrevemos a seguir pode ser executado diretamente sobre a árvore  $T$ .

Para a execução do algoritmo, inicialmente deve-se escolher um vértice arbitrário como *raiz* da árvore  $T$ . Os vértices são numerados como  $v_1, v_2, \dots, v_n$ , de tal forma que se  $v_j$

é filho de  $v_k$ , então  $j > k$ . Neste caso, a raiz é  $v_1$ . Observamos que todo vértice  $v$  de  $T$ , exceto a raiz, possui um *pai*, que é o vértice adjacente a  $v$  que não é filho de  $v$ .

O algoritmo associa, a cada vértice  $v_i$  de  $T$ , um valor real  $a(v_i)$  que no final do algoritmo coincide com a entrada  $ii$  da matriz diagonal  $D_\alpha$ . Por esta razão, vamos denominar  $a(v_i)$  o *valor diagonal* de  $v_i$ . Inicialmente, cada vértice  $v_i$  recebe o valor diagonal  $a(v_i) = \delta_i + \alpha$ , onde  $\delta_i$  é o peso do vértice  $v_i$ , ou seja, a entrada  $ii$  da matriz diagonal  $D$ .

O algoritmo então é aplicado seguindo a numeração dos vértices, das folhas em direção à raiz, conforme apresentado na Figura 1. Para um vértice  $v_k$ , denotamos por  $F_k$  o conjunto de todos os filhos de  $v_k$ . Observamos que se  $v_k$  é uma folha, que não é a raiz, então  $F_k = \emptyset$ .

**Entrada:** árvore com pesos  $T$  de vértices  $v_1, v_2, \dots, v_n$ , escalar  $\alpha$  e matriz diagonal  $D$ .

**Saída:** entradas da diagonal de uma matriz diagonal  $D_\alpha$  congruente a  $\tilde{L}(T) + \alpha I$ .

Inicialize  $a(v_i) := \delta_i + \alpha$ , para todo vértice  $v_i$  de  $T$ .  
 Para  $k = n$  até 1  
   se  $v_k$  não é uma folha, então  
     1. se  $a(v_i) \neq 0$ , para todo  $v_i \in F_k$ , então  
         $a(v_k) \leftarrow a(v_k) - \sum_{v_i \in F_k} \frac{(\omega_{ik})^2}{a(v_i)}$ .  
     2. se  $a(v_i) = 0$  para algum  $v_i \in F_k$ , então  
        escolha um vértice  $v_j$  em  $F_k$  tal que  $a(v_j) = 0$ ;  
         $a(v_k) \leftarrow -\frac{(\omega_{jk})^2}{2}$ ;  $a(v_j) \leftarrow 2$ ;  
        se  $v_k$  tem um pai  $v_\ell$ , remova a aresta  $v_k v_\ell$ .  
 Imprima  $a(v_1), a(v_2), \dots, a(v_n)$ .

Figura 1: Algoritmo *DiagonalizeW(T, α)*.

Para mostrar que o procedimento acima realmente calcula os valores de uma matriz diagonal congruente à matriz  $\tilde{L}(T) + \alpha I$ , consideremos um vértice  $v_k$  de  $T$  com um filho  $v_j$ , o que corresponde às entradas abaixo:

$$\begin{matrix}
 & & & \vdots & & \vdots & & \\
 k & \left[ \begin{array}{cccccc}
 \cdots & a(v_k) & \cdots & \omega_{kj} & \cdots & \\
 & \vdots & \ddots & \vdots & & \\
 j & \cdots & \omega_{jk} & \cdots & a(v_j) & \cdots \\
 & & \vdots & & \vdots & 
 \end{array} \right] & & \\
 & & & \vdots & & \vdots & & 
 \end{matrix}$$

Se  $a(v_j) \neq 0$ , então, as seguintes operações elementares nas linhas e colunas da matriz anulariam as entradas  $kj$  e  $jk$ :

$$L_k \leftarrow L_k - \frac{\omega_{jk}}{a(v_j)} L_j \quad \text{e} \quad C_k \leftarrow C_k - \frac{\omega_{jk}}{a(v_j)} C_j.$$



A seguir, as operações abaixo

$$L_k \leftarrow L_k - \frac{(\delta_k + \alpha)}{2\omega_{jk}}L_j \quad \text{e} \quad C_k \leftarrow C_k - \frac{(\delta_k + \alpha)}{2\omega_{kj}}C_j$$

anulariam a entrada  $kk$  e as novas entradas seriam:

$$\begin{matrix} \ell \\ k \\ j \\ i \end{matrix} \begin{bmatrix} \delta_\ell + \alpha & 0 & & \\ & 0 & \omega_{kj} & 0 \\ & & \omega_{jk} & 0 \\ & & 0 & a(v_i) \end{bmatrix}.$$

Por fim, as operações abaixo

$$\begin{matrix} L_j & \leftarrow & L_j + \frac{1}{\omega_{kj}}L_k, & C_j & \leftarrow & C_j + \frac{1}{\omega_{jk}}C_k, \\ L_k & \leftarrow & L_k - \frac{\omega_{kj}}{2}L_j, & C_k & \leftarrow & C_k - \frac{\omega_{jk}}{2}C_j \end{matrix}$$

resultariam na seguinte forma diagonalizada:

$$\begin{matrix} \ell \\ k \\ j \\ i \end{matrix} \begin{bmatrix} \delta_\ell + \alpha & 0 & & \\ & 0 & -\frac{(\omega_{kj})^2}{2} & 0 \\ & & 0 & 2 \\ & & 0 & a(v_i) \end{bmatrix}.$$

Note que os valores diagonais de  $v_j$  e  $v_k$  obtidos nas operações anteriores são exatamente os valores diretamente atribuídos pelo algoritmo no passo 2.

Uma vez que os valores diagonais calculados diretamente pelo algoritmo foram obtidos através de operações elementares realizadas nas linhas e colunas de tal forma que cada operação realizada para uma linha foi também realizada para a coluna correspondente, a matriz diagonal  $D_\alpha$  cujas entradas são os  $n$  valores diagonais calculados pelo algoritmo é tal que  $D_\alpha = S(\bar{\mathcal{L}}(T) + \alpha I)S^T$ , onde  $S$  é a matriz invertível composta pelas operações elementares realizadas nas linhas de  $\bar{\mathcal{L}}(T) + \alpha I$ . Desta forma, segue que  $D_\alpha$  e  $\bar{\mathcal{L}}(T) + \alpha I$  são congruentes. Consequentemente, pela Lei da Inércia de Sylvester (veja [6, Teorema 4.5.8]), obtemos o resultado abaixo:

**Teorema 2.1.** *Fixada uma matriz diagonal real  $D$ , sejam  $T$  uma árvore com pesos,  $\bar{\mathcal{L}}(T)$  a matriz laplaciana perturbada de  $T$  com respeito a  $D$  e  $\alpha$  um número real. Seja  $D_\alpha$  a matriz diagonal cujas entradas na diagonal são produzidas pelo algoritmo  $DiagonalizeW(T, -\alpha)$ . Então, o número de entradas positivas, negativas e nulas em  $D_\alpha$  é igual ao número de autovalores de  $\bar{\mathcal{L}}(T)$  que são maiores, menores e iguais a  $\alpha$ , respectivamente.*

**Exemplo 2.1.** *Seja  $T$  a árvore do lado esquerdo da Figura 2, cujos pesos estão representados sobre as arestas. Consideremos o caso em que os pesos dos vértices (entradas diagonais da matriz  $D$ )  $v_1, v_2, v_3, v_4$  e  $v_5$  de  $T$  são dados, respectivamente, por  $\delta_1 = 1, \delta_2 = 2, \delta_3 = 1, \delta_4 = 1$  e  $\delta_5 = 1$ .*

*Aplicando o algoritmo  $DiagonalizeW(T, \alpha)$ , com  $\alpha = -2$ , cada vértice  $v_i$  de  $T$  será inicializado com o valor diagonal  $a(v_i) = \delta_i - 2$ , para  $i = 1, \dots, 5$ . Estes valores estão representados sobre os vértices de  $T$  no lado esquerdo da Figura 2.*

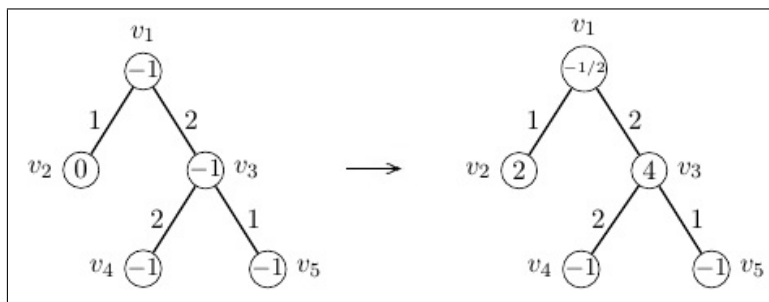


Figura 2: Aplicação do algoritmo  $DiagonalizeW(T, \alpha)$  com  $\alpha = -2$

Como  $v_4$  e  $v_5$  são os filhos de  $v_3$  e têm valores não nulos, o algoritmo atribui  $a(v_3) = -1 - \frac{(\omega_{34})^2}{a(v_4)} - \frac{(\omega_{35})^2}{a(v_5)} = -1 - \frac{2^2}{-1} - \frac{1^2}{-1} = 4$ .

Ao processar o vértice  $v_1$ , como ele tem um filho com valor nulo (o vértice  $v_2$ ), o algoritmo atribui o valor diagonal 2 para  $v_2$ , enquanto que o valor diagonal de  $v_1$  torna-se  $a(v_1) = -\frac{(\omega_{12})^2}{2} = -\frac{1^2}{2} = -\frac{1}{2}$ .

Assim, como o algoritmo produziu dois valores diagonais positivos e três negativos, pelo Teorema 2.1, podemos concluir que  $\mathcal{L}(T)$  tem dois autovalores maiores do que 2 e três autovalores menores do que 2.

Ao aplicarmos o algoritmo com  $\alpha = 0$ , obtemos que  $\mathcal{L}(T)$  tem quatro autovalores positivos e um negativo, enquanto que, utilizando  $\alpha = -1$ , obtemos que  $\mathcal{L}(T)$  tem um autovalor igual a 1, dois autovalores menores do que 1 e dois autovalores maiores do que 1. Podemos, desta forma, concluir que  $\mathcal{L}(T)$  tem um autovalor negativo, um autovalor em  $(0, 1)$ , um autovalor igual a 1 e dois autovalores maiores do que 2.

### 3 Árvores com até Cinco Autovalores Distintos com Relação a uma Família de Matrizes Laplacianas Perturbadas

O *diâmetro* de um grafo é a máxima distância entre seus vértices. Sabe-se que se  $G$  é um grafo conexo com pesos positivos e com diâmetro  $d$ , então toda matriz laplaciana perturbada de  $G$  tem pelo menos  $d + 1$  autovalores distintos (veja [2, Teorema 4.1.7]). Se além disso  $G$  for bipartido, então qualquer matriz laplaciana perturbada de  $G$  da forma  $\mu I - A(G)$ , onde  $\mu \in \mathbb{R}$ , tem espectro simétrico em relação ao número real  $\mu$  ([2, Teorema 4.3.2]).

Com o objetivo de caracterizar todas as árvores com até cinco autovalores distintos para matrizes laplacianas perturbadas da forma  $\mu I - A$ , para algum  $\mu \in \mathbb{R}$ , basta considerar as árvores com diâmetro  $d$  menor ou igual a 4, visto que uma árvore é um grafo conexo bipartido.

Seja  $T$  uma árvore de  $n$  vértices e diâmetro  $d$  menor ou igual a 4. Se  $d = 1$ ,  $T$  é o grafo completo com dois vértices e tem 2 autovalores distintos. No caso em que  $d = 2$ ,  $T$  é a estrela  $S_n$  e, utilizando o algoritmo  $DiagonalizeW(T, \alpha)$ , obtemos que  $T$  tem exatamente 3 autovalores distintos, simétricos em relação ao autovalor  $\mu$ , cuja multiplicidade é igual

a  $n - 2$ . Segue também do algoritmo que o caminho  $P_4$  com quatro vértices é a única árvore com exatamente 4 autovalores distintos e qualquer outra árvore de diâmetro  $d = 3$  tem exatamente 5 autovalores distintos ([2, Teorema 4.3.7]). No caso em que  $d = 4$ ,  $T$  é da forma apresentada na Figura 3: contém um vértice  $v$  adjacente a  $k \geq 2$  vértices  $v_i$ ,  $i = 1, \dots, k$ , de grau  $p_i + 1$ , onde  $p_i \geq 1$ , e cada  $v_i$  é adjacente a  $p_i$  folhas, e  $v$  também é, possivelmente, adjacente a  $m \geq 0$  folhas. Neste caso, segue-se do algoritmo que  $T$  tem exatamente 5 autovalores distintos se, e somente se,  $m = 0$  e  $k = 2$ , ou  $m = 0$ ,  $k \geq 3$  e todos os vértices  $v_i$ ,  $i = 1, \dots, k$ , possuem a mesma soma dos quadrados dos pesos das arestas que ligam  $v_i$  aos seus pendentes ([2, Teorema 4.4.5]).

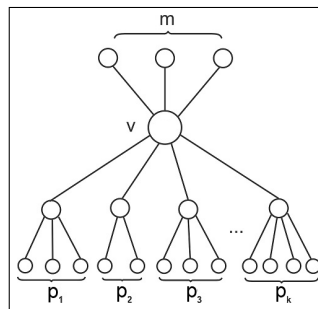


Figura 3: Árvore de diâmetro 4.

## Referências

- [1] R. B. Bapat, S. J. Kirkland, and S. Pati. The perturbed Laplacian matrix of a graph, *Linear and Multilinear Algebra*, 49:219–242, 2001.
- [2] R. O. Braga. Localização de autovalores de árvores e de grafos unicyclicos, Tese de Doutorado, UFRGS, 2015.
- [3] R. O. Braga, R. R. Del-Vecchio, V. M. Rodrigues, and V. Trevisan. Trees with 4 or 5 distinct normalized Laplacian eigenvalues, *Linear Algebra and its Applications*, 471:615–635, 2015.
- [4] F.R.K. Chung. *Spectral Graph Theory*. American Math. Soc., Providence, 1997.
- [5] E. Fritscher, C. Hoppen, I. Rocha, and V. Trevisan. On the sum of the Laplacian eigenvalues of a tree, *Linear Algebra and its Applications*, 435:371–399, 2011
- [6] R. Horn, and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [7] D. P. Jacobs, and V. Trevisan. Locating the eigenvalues of trees, *Linear Algebra and its Applications*, 434:81–88, 2011.
- [8] S. Radenković, and I. Gutman. Total  $\pi$ -electron energy and Laplacian energy: how far the analogy goes?, *Journal of the Serbian Chemical Society*, 73:1343–1350, 2007.