Proceeding Series of the Brazilian Society of Applied and Computational Mathematics, Vol. 2, N. 1, 2014.
Trabalho apresentado no CMAC-Sul, Curitiba-PR, 2014.

Localizando Autovalores de Grafos Threshold

Fernando Tura,
Campus Alegrete, UNIPAMPA
97546–550 Alegrete, RS, Brazil
fernandotura@unipampa.edu.br

Resumo: *Seja G um grafo threshold de ordem n com matriz de adjacência A. Apresentamos um algoritmo de ordem $O(n)$ que constrói uma matriz diagonal congruente a $A + xI$, onde x é um número real. Como aplicação, podemos localizar os autovalores de um grafo threshold G em um intervalo real $(a, b]$.*

1 Introdução

O estudo do espectro de um grafo é um importante tópico da Teoria Algébrica de Grafos. Embora a teoria espectral de grafos tenha um considerável campo de contribuições [3], ainda existem problemas em aberto. Por exemplo, determinar grafos não isomorfos com o mesmo espectro.

Seja $G = (V, E)$ um grafo não direcionado com vértices $V = (v_1, \ldots, v_n)$ e o conjunto de arestas E, sua *matriz de adjacência* $A(G) = [a_{ij}]$ é uma matriz de ordem $n \times n$ de modo que $a_{ij} = 1$ se e somente se v_i é adjacente a v_j (isto é, se existe uma aresta entre v_i e v_j) e $a_{ij} = 0$, caso contrário. Um valor λ é dito um *autovalor* de G se $\det(\lambda I_n - A) = 0$, e como A é simétrica e real seus autovalores são reais.

Neste trabalho estamos interessados nos autovalores dos grafos *threshold*. Os grafos threshold foram introduzidos por Chvátal e Hammer [2] na década de 70. Eles são uma importante classe de grafos por causa das suas numerosas aplicações em diversas áreas, tais como ciência da computação e psicologia [6].

Os grafos threshold podem ser caracterizados de diversas maneiras. Uma dessas caracterizações diz que um grafo threshold pode ser obtido através de um processo recursivo que inicia com um vértice isolado, onde, a cada passo, ou um novo vértice *isolado* é adicionado, ou um vértice *dominante* é adicionado. Dessa forma podemos representar um grafo threshold de n vértices usando uma sequência binária $(b_1, \ldots, b_n)$, onde b_i é 0 se o vértice v_i foi adicionado como um vértice isolado, e b_i é 1 se v_i foi adicionado como um vértice dominate. Esta representação tem sido chamada de *sequência de criação* [5]. Em nossa representação b_1 é sempre zero.

Na construção da matriz de adjacência de um grafo threshold, ordenamos os vértices da mesma forma que eles dados em sua sequência de criação. Por exemplo, (1) mostra a matriz de adjacência A do grafo threshold com representação $(0, 1, 0, 1)$.

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \tag{1}$$

Dizemos que duas matrizes R e S são *congruentes* se existe uma matriz P não singular de modo que $R = P^T SP$. Nosso principal resultado é um algoritmo que constrói uma matriz *diagonal* congruente a $A + xI$, onde A é a matriz de adjacência de um grafo threshold e x é um número real. Como aplicação, mostraremos como obter os autovalores de um grafo threshold em um intervalo real $(a, b]$.

2 Diagonalizando $\mathbf{A} + xI$

Nessa seção apresentaremos o nosso algoritmo de diagonalização. Assumimos que A é de ordem n, o algoritmo executa $n-1$ passos e opera na matriz de baixo para cima e da direita para a esquerda. Em cada passo as linhas e colunas m e $m-1$ participam nas operações linha e coluna. A diagonalização é obtida pelo fato que, no fim de cada passo, todas as entradas da linha e coluna m serão nulas com exceção do elemento da diagonal. Por exemplo, para a matriz A dada em (1) e $x = 1$, o algoritmo procederia da seguinte forma:

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A propriedade de congruência é garantida pelo fato de que a cada estágio aplicamos o mesmo par de operações elementares nas linhas e colunas.

O que é notável é que na execução do nosso algoritmo não precisamos armazenar toda matriz, mas somente a diagonal e a representação $(b_1, \ldots, b_n)$ de G. Nosso algoritmo de ordem $O(n)$ em tempo e espaço é mostrado na Figura 1.

Para explicarmos o algoritmo, devemos considerar três casos principais. Aqui provaremos apenas dois subcasos, os demais podem ser verificados em [4].

Caso 1: $b_{m-1} = b_m = 1$. Então a submatriz de ordem $m \times m$ tem a seguinte forma:

$$\begin{array}{c} \\ \\ \\ m-1 \\ m \end{array} \begin{bmatrix} & & & 1 & 1 \\ & & & \vdots & \vdots \\ & & & 1 & 1 \\ 1 & \ldots & 1 & x & 1 \\ 1 & \ldots & 1 & 1 & \alpha \end{bmatrix}.$$

Neste caso, fazemos as operações abaixo nas linhas e colunas

$$l_m \leftarrow l_m - l_{m-1}$$
$$c_m \leftarrow c_m - c_{m-1},$$

obtendo-se a matriz

$$\begin{array}{c} \\ \\ \\ m-1 \\ m \end{array} \begin{bmatrix} & & & 1 & 0 \\ & & & \vdots & \vdots \\ & & & 1 & 0 \\ 1 & \ldots & 1 & x & 1-x \\ 0 & \ldots & 0 & 1-x & \alpha+x-2 \end{bmatrix}.$$

DOI: 105540/03.2014.002.01.0061

```
Algorithm Diagonalize(G, x)
   initialize d_i ← x, for all i
   for m = n to 2
      α ← d_m
      if b_{m-1} = 1 and b_m = 1
         if α + x ≠ 2                 //subcase 1a
            d_{m-1} ← (αx-1)/(α+x-2)
            d_m ← α + x - 2
         else if x = 1                //subcase 1b
            d_{m-1} ← 1
            d_m ← 0
         else                         //subcase 1c
            d_{m-1} ← 1
            d_m ← -(1-x)^2
            b_{m-1} ← 0
      else if b_{m-1} = 0 and b_m = 1
         if x = 0                     //subcase 2a
            d_{m-1} ← 1
            d_m ← -1
         else                         //subcase 2b
            d_{m-1} ← α - 1/x
            d_m ← x
            b_{m-1} ← 1
   end loop
```

Figura 1: Algoritmo de diagonalização.

A maior parte dos elementos não nulos da linha e coluna m foram removidos. Agora devemos remover as duas entradas iguais a $1-x$. Assim, consideraremos três subcasos.
Subcaso 1a: $\alpha + x - 2 \neq 0$. Então procedendo com as seguintes operações :

$$l_{m-1} \leftarrow l_{m-1} - \frac{1-x}{\alpha+x-2} l_m$$

$$c_{m-1} \leftarrow c_{m-1} - \frac{1-x}{\alpha+x-2} c_m$$

obtemos

$$\begin{array}{c} \\ \\ \\ m-1 \\ m \end{array} \begin{bmatrix} & & & 1 & 0 \\ & & & \vdots & \vdots \\ & & & 1 & 0 \\ 1 & \dots & 1 & \gamma & 0 \\ 0 & \dots & 0 & 0 & \alpha+x-2 \end{bmatrix},$$

onde $\gamma = x - \frac{(1-x)^2}{\alpha+x-2} = \frac{\alpha x - 1}{\alpha+x-2}$.

Caso 2: $b_{m-1} = 0$ e $b_m = 1$. Então a submatriz de ordem $m \times m$ tem a seguinte forma:

$$\begin{bmatrix} & & & 0 & 1 \\ & & & \vdots & \vdots \\ & & & 0 & 1 \\ 0 & \dots & 0 & x & 1 \\ 1 & \dots & 1 & 1 & \alpha \end{bmatrix}.$$

DOI: 105540/03.2014.002.01.0061

Neste caso trocamos as linhas m e $m-1$ e as colunas m e $m-1$, obtendo a matriz:

$$\begin{bmatrix} & & & 1 & 0 \\ & & & \vdots & \vdots \\ & & & 1 & 0 \\ 1 & \dots & 1 & \alpha & 1 \\ 0 & \dots & 0 & 1 & x \end{bmatrix}.$$

Consideraremos dois subcasos, dependendo se x é zero ou não.

Subcaso 2a: $x = 0$. Então para cada i, $1 \le i \le m-2$ fazemos as operações:

$$l_i \leftarrow l_i - l_m$$

$$c_i \leftarrow c_i - c_m$$

que anulam os $1's$ da linha e coluna $m-1$, à esquerda e acima de α. Com as operações

$$l_{m-1} \leftarrow l_{m-1} + \frac{1-\alpha}{2} l_m$$

$$c_{m-1} \leftarrow c_{m-1} + \frac{1-\alpha}{2} c_m$$

$$l_m \leftarrow l_m - l_{m-1}$$

$$c_m \leftarrow c_m - c_{m-1}$$

obtemos a matriz

$$\begin{bmatrix} & & & 0 & 0 \\ & & & \vdots & \vdots \\ & & & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & -1 \end{bmatrix}.$$

Assim após a verificação de todos os subcasos, temos provado o seguinte resultado:

Teorema 2.1 *Dados G e x, onde G é um grafo threshold com matriz de adjacência A, o algoritmo de diagonalização calcula uma matriz diagonal D, que é congruente a $A + xI$.*

3 Exemplo

Para ilustrarmos o algoritmo de diagonalização nessa seção apresentaremos um exemplo. Assumimos que G é um grafo threshold dado por $(0,1,0,1)$ e $x = 1$.

O primeiro passo do algoritmo para $m = 4$, executa o subcaso 2b, já que $b_3 = 0, b_4 = 1$ e $\alpha = x = 1$, e as seguintes atribuições são feitas:

$$d_3 \leftarrow 1 - \frac{1}{1} = 0$$

$$d_4 \leftarrow 1$$

$$b_3 \leftarrow 1.$$

Para $m = 3$, o subcaso 1a é executado, já que $b_2 = b_3 = 1$, $x = 1$, e $\alpha = 0$, logo

$$d_2 \leftarrow \frac{\alpha x - 1}{\alpha + x - 2} = \frac{-1}{-1} = 1$$

$$d_3 \leftarrow \alpha + x - 2 = -1.$$

Para $m = 2$, é executado o subcaso 2b, já que $b_1 = 0, b_2 = 1$, e $x = 1$. Assim,

$$d_1 \leftarrow 0$$

$$d_2 \leftarrow 1.$$

A seguinte tabela fornece os passos do algoritmo em cada passo:

b_i	d_i	b_i	d_i	b_i	d_i	b_i	d_i
0	1	0	1	0	1	0	0
1	1	1	1	1	1	1	1
0	1	1	0	1	-1	1	-1
1	1	1	1	1	1	1	1
inicial		após 2b		após 1a		após 2b	

4 Encontrando os Autovalores

Para determinarmos os autovalores de um grafo threshold G, enunciaremos o seguinte resultado conhecido como a lei da Inércia de Sylvester . A prova pode ser encontrada em [1].

Teorema 4.1 *Duas matrizes reais, simétricas, de ordem $n \times n$ são congruentes se e somente se elas têm o mesmo número de autovalores positivos e o mesmo número de autovalores negativos.*

O seguinte resultado estabelece uma relação entre os autovalores de um grafo threshold G com a matriz diagonal D obtida pelo algoritmo de diagonalização. A prova pode ser encontrada em [4].

Teorema 4.2 *Seja D a matriz diagonal resultante da aplicação do algoritmo de diagonalização para $(G, -a)$, onde G é um grafo threshold. Então:*

i. O número de autovalores de G maiores do que a é o número de entradas positivas em D.

ii. O número de autovalores de G menores do que a é o número de entradas negativas em D.

iii. A multiplicidade de um autovalor a é o número de entradas nulas na diagonal de D.

Corolário 4.1 *O número de autovalores de um grafo threshold G com matriz de adjacência A em um intervalo $(a, b]$, incluindo multiplicidades, é o número de entradas positivas na diagonalização de $A - aI$, menos o número de entradas positivas na diagonalização de $A - bI$.*

Segue do Corolário 4.1 que podemos determinar o número de autovalores em um intervalo chamando o algoritmo de diagonalização duas vezes. Por exemplo, consideremos o grafo threshold G representado por $(0, 1, 1, 1)$. Quando aplicamos o algoritmo para $x = 1$, obtemos a sequência final igual a $d = (0, 1, 0, 0)$. Pelo Teorema 4.2, significa que $x = -1$ é um autovalor de G com multiplicidade 3. Além disso, existe 1 autovalor maior do que -1. Quando aplicamos o algoritmo ao mesmo grafo para $x = -4$, a sequência final é igual a $d = (-5/12, -4, -15/2, -10)$. Isto implica que nenhum autovalor é maior do que 4. Concluímos que existe um único autovalor $\lambda \in (-1, 4]$. Usando-se o método de bisseção podemos localizar esse autovalor com aproximação eficiente.

5 Conclusões

Neste trabalho apresentamos um método eficiente para determinar o espectro de grafos threshold, a partir do algoritmo de diagona-lização. Embora nosso algoritmo seja baseado em operações elementares de matrizes, na prática sua implementação depende apenas da sequência binária do grafo threshold de ordem n, e pode assim ser implementado usando dois vetores de comprimento n, tal que sua complexidade é de ordem $O(n)$.

Referências

[1] G. L. Bradley, A Primer of Linear Algebra, Prentice Hall, Englewood Cliffs, 1975.

[2] V. Chvátal, P. L. Hammer, Aggregation of inequalities in integer programming, in Studies in Integer Programming, Annals of Discrete Math., 1, 145–162, North-Holland, Amsterdam, 1977.

[3] D. Cvetković, M. Doob, H. Sachs, Spectra of Graphs: Theory and Application, third edition, Wiley-VHC, 1998.

[4] D. P. Jacobs, V. Trevisan and F. Tura, Eigenvalue location in threshold graphs, Linear Algebra and its applications 439, (2013) 2762-2773 .

[5] A. Hagberg, P. J. Swart, D. A. Schult. Designing threshold networks with given structural and dynamical properties, Physical Review E 74 (2006) doi: 10.1103/PhysRevE.74.056116.

[6] N. V. R. Mahadev and U. N. Peled, Threshold graphs and related topics, Elsevier, 1995.

