

BOUNDS FOR THE DOMINANT EIGENVECTOR OF A GRAPH

Horst D. Simon

Department of Applied Mathematics and Statistics
State University of New York at Stony Brook
Stony Brook, NY 11794

Abstract. In this note we prove lower bounds on the components of the eigenvector associated with the dominant eigenvalue of a graph. These bounds depend only on the eccentricity of the corresponding node and on the eigenvalue. As corollary lower bounds on the dominant eigenvalue are derived, which depend on the diameter of the graph. These bounds were motivated by a heuristic algorithm for finding peripheral nodes in a graph. We present this algorithm and indicate its use in sparse matrix computations.

1. Introduction and Definitions

Several algorithms concerned with the solution of sparse, symmetric, positive definite systems of linear equations, for example envelope methods based on the reverse Cuthill-McKee ordering [2], or dissection algorithms (one-way dissection, nested dissection) [4], require the determination of a peripheral (or at least pseudo-peripheral) node in the associated graph. A widely used algorithm for this purpose is due to Gibbs, Pote, and Stockmeyer [5], and was improved by George and Liu [4], and by Lewis [6]. This algorithm does not guarantee to find a peripheral node. There exist examples, where this algorithm fares particularly badly. Here we introduce a quite different algorithm, which (unfortunately) does not guarantee to find a peripheral node either

However, while studying this algorithm, several new bounds on the dominant eigenvalue and associated eigenvector of the adjacency matrix of a graph were obtained, which are interesting in their own right. The eigenvectors of the adjacency matrix of a graph have so far received only marginal attention (c.f. [3]) There are, however, recent results [1], which indicate that certain algebraic properties of the eigenvectors of the adjacency matrix can be related to structural properties of the underlying graph. This area of research poses some challenging new questions and we will indicate some open problems in this note

Here we consider an undirected, connected graph $G = (X, E)$, where X is the set of nodes, and E is the set of edges. The elements a_{ij} of the adjacency matrix A of G are defined by

$$a_{ij} = \begin{cases} 1 & \text{if node } i \text{ and } j \text{ are adjacent, or if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (1.1)$$

This definition differs from the common definition of an adjacency matrix in as far as we also set $a_{ij} = 1$, whereas usually the diagonal elements are set to be zero. If G is the ordered graph of a symmetric positive definite matrix M , this definition proves to be more useful for our purposes. In this case the a_{ij} could be defined directly by

$$a_{ij} = \begin{cases} 1 & \text{if } m_{ij} \neq 0 \\ 0 & \text{if } m_{ij} = 0, \end{cases} \quad (1.2)$$

i.e. the adjacency matrix reflects directly the zero-nonzero structure of a given matrix and is therefore the appropriate tool for sparse matrix computations.

Since we assumed G to be connected, the matrix A is indecomposable. By the Perron-Frobenius theorem, A has a simple, positive eigenvalue λ . The corresponding eigenvector $v = (v_1, v_2, \dots, v_n)^T$ has all components $v_i > 0$, $i = 1, \dots, n$. Here $n = |X|$. Therefore v can be normalized such that $\sum_{i=1}^n v_i = 1$. In the following we will only deal with λ and v , such that

$$Av = \lambda v, \quad \sum_{i=1}^n v_i = 1, \quad v_i > 0 \text{ for } i = 1, \dots, n \quad (1.3)$$

No confusion with other eigenvalues and vectors is possible.

We are also going to use the notation $A_1 > A_2$, which means that all elements of the matrix A_1 are larger than the corresponding elements of A_2 . $A > \alpha$ for $\alpha \in \mathbb{R}$ means that all elements of A are larger than α . We will use the same notation for vectors

For a node $x_i \in X$ the degree x_i , i.e. the number of adjacent nodes is denoted by $\deg(x_i)$. $d(x_i, x_j)$ denotes the distance of two nodes x_i and x_j ,

i.e. the length of the shortest path connecting x_i and x_j . The eccentricity of a node x_i is the quantity

$$e(x_i) = \max_{j=1, \dots, n} d(x_i, x_j) \quad (1.4)$$

The diameter of G is then defined by

$$\delta(G) = \max_{i=1, \dots, n} e(x_i). \quad (1.5)$$

A node $x_i \in X$ is said to be peripheral if its eccentricity is equal to its diameter, i.e. if $\delta(G) = e(x_i)$.

For a subset $Y \subseteq X$, the adjacency set of Y , denoted by $\text{Adj}(Y)$ is

$$\text{Adj}(Y) = \{x_i \in X - Y \mid \{x_i, x_j\} \in E \text{ for some } x_j \in Y\} \quad (1.6)$$

For a node $x \in X$, the level structure rooted at x is the partitioning $L(x)$ of X satisfying

$$L(x) = \{L_0(x), L_1(x), \dots, L_{e(x)}(x)\}$$

where $L_0(x) = \{x\}, \quad L_1(x) = \text{Adj}(L_0(x))$

and $L_i(x) = \text{Adj}(L_{i-1}(x)) - L_{i-2}(x) \quad i = 2, 3, \dots, e(x).$

2. A Heuristic Algorithm for Finding Peripheral Nodes

We are trying to find a peripheral node of the graph G , i.e. a node with maximal eccentricity. Such a node is, on average, further away from all other nodes.

Consider now the matrix A^k . Its (i, j) -th entry denotes the number of different paths of length k leading from x_i to x_j , where also paths are included, which "stay for a while" at a node, because of $a_{ii} = 1$. Now

let $u = (1, 1, \dots, 1)^T$, then the i -th component of $A^k u$ is equal to the number of paths of length k , beginning at an arbitrary node and ending in x_i . If a node x_i is "peripheral", this number will be smaller and if a node x_i lies in the "center" of the graph, this number will be larger. So for $k \rightarrow \infty$ one should obtain some average number, which indicates how many paths go "on average" through a node. But with some suitable normalization, $A^k u$ converges to the largest eigenvector v of A , unless u were orthogonal to this eigenvector. But this cannot happen, since $u = (1, \dots, 1)^T$ we have

$$u^T v = \sum_{i=1}^n v_i = 1 \neq 0$$

These arguments suggest the following very simple algorithm for finding "peripheral" nodes of a graph:

- 1) Find v , the dominant eigenvector of the adjacency matrix A
- 2) The node corresponding to the smallest component in v is a "peripheral" node

Unfortunately this algorithm will not always produce a peripheral node as defined in Section 1. The following graph is a typical counterexample

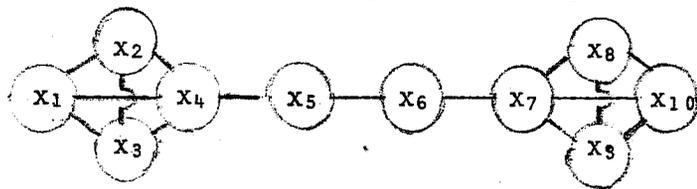


Figure 1. Counterexample.

Clearly all the nodes in the two cliques at the end (x_1, x_2, x_3 and x_8, x_9, x_{10}) are peripheral. The vector v however is given by

$$v \approx (0.1073, 0.1073, 0.1073, 0.1211, 0.569, 0.569, 0.1211, 0.1073, 0.1073, 0.1073)^T$$

The smallest components of v are just corresponding to the "interior" nodes x_5 and x_6 .

3. Bounds for the Dominant Eigenvector.

Although the counterexample above shows that the heuristic algorithm from section 2 will not always produce peripheral nodes, we were able to obtain lower bounds on the components of the dominant eigenvector. These bounds indicate that there is a certain inverse relationship between the components of the eigenvector and the eccentricity of the corresponding node.

Proposition 1. The components v_i of the dominant eigenvector v satisfy

$$v_i \geq \frac{1}{e(x_i)(\lambda-1) + 1}$$

for $i = 1, 2, \dots, n$.

Proof. Let $Av = \lambda v$ and let $L(x_i) = \{L_0(x_i), L_1(x_i), \dots, L_{e(x_i)}(x_i)\}$ be the levelstructure rooted at x_i . Furthermore, let for brevity

$$\sum_{\text{Adj}(x_i)} v_j$$

denote the sum of all v_j over all indices j , such that $x_j \in \text{Adj}(x_i)$, and similar $\sum_{L_k(x_i)} v_j$, etc.

Now $Av = \lambda v$ implies that (for $n > 1$),

$$v_i = \frac{1}{\lambda-1} \sum_{\text{Adj}(x_i)} v_j = \frac{1}{\lambda-1} \sum_{L_1(x_i)} v_j, \quad i = 1, \dots, n \quad (3.3)$$

Substituting (3.3) into itself and taking into account that $x_i \in \text{Adj}(x_i)$ for $x_j \in L_1(x_i)$, we obtain

$$v_i \geq \frac{1}{(\lambda-1)^2} \sum_{L_2(x_i)} v_j, \quad i = 1, \dots, n \quad (3.4)$$

This process can be repeated $e(x_i)$ times so that we obtain

$$v_i \geq \frac{1}{(\lambda-1)^k} \sum_{L_k(x_i)} v_j \quad \text{for } i = 1, 2, \dots, n \quad (3.5)$$

$$k = 1, 2, \dots, e(x_i)$$

Summing up the $e(x_i)$ inequalities (3.5), it follows that

$$e(x_i)v_i \geq \sum_{k=1}^{e(x_i)} \frac{1}{(\lambda-1)^k} \sum_{L_k(x_i)} v_j \geq \frac{1}{(\lambda-1)^{e(x_i)}} \sum_{\substack{j=1 \\ j \neq i}}^n v_j = \frac{1-v_i}{(\lambda-1)^{e(x_i)}}.$$

Therefore

$$v_i \geq \frac{1}{e(x_i)(\lambda-1)^{e(x_i)+1}} \quad \text{for } i = 1, 2, \dots, n$$

and this is also correct for $n = 1$. \square

Proposition 2.

$$\lambda \geq 1 + \delta \sqrt{\frac{n-1}{\delta}} \quad (3.6)$$

Proof. From (3.1) it follows

$$v_i \geq \frac{1}{\delta(\lambda-1)^\delta + 1} \quad \text{for } i = 1, \dots, n.$$

Summing up for $i = 1, \dots, n$ and rearranging yields the result. \square

In order to prove Proposition 3, the following Lemma is needed

Lemma. Let $a_{ij}^{(k)}$ be the (i,j) -th entry of the matrix A^k , $k = 1, 2, 3, \dots$

Then it holds that

$$a_{ij}^{(k)} \geq 1 \quad \text{for all } i, j \text{ with } d(x_i, x_j) = k \quad (3.7)$$

$$a_{ij}^{(k)} \geq k \quad \text{for all } i, j \text{ with } d(x_i, x_j) < k \quad (3.8)$$

not true for
 $n \geq$

Proof. If $d(x_i, x_j) = k$ then there exists at least one path of length k leading from x_i to x_j , and (3.7) follows

We show (3.8) by induction. For $k = 1$ we have $a_{ij}^{(1)} \geq 1$ for all i, j with $d(x_i, x_j) < 1$, i.e. for all diagonal elements. Suppose now for some $k > 1$ it holds that $a_{ij}^{(k-1)} \geq k-1$ for all i, j with $d(x_i, x_j) < k-1$.

$a_{ij}^{(k)}$ is given by

$$a_{ij}^{(k)} = \sum_{\ell=1}^n a_{i\ell}^{(k-1)} a_{\ell j}^{(1)}$$

Now $a_{jj}^{(1)} = 1$ and since the graph is connected, there exists at least one m , $1 \leq m \leq n$, $m \neq j$, such that $a_{mj}^{(1)} = 1$. Therefore

$$\begin{aligned} &\geq a_{ij}^{(k-1)} a_{jj}^{(1)} + a_{im}^{(k-1)} a_{mj}^{(1)} \\ &= a_{ij}^{(k-1)} + a_{mj}^{(k-1)} \geq k-1 + a_{im}^{(k-1)} \end{aligned}$$

It remains to be shown that $a_{im}^{(k-1)} \geq 1$. We have $d(x_i, x_j) < k-1$ and $d(x_j, x_m) = 1$. Therefore

$$d(x_i, x_m) \leq d(x_i, x_j) + d(x_j, x_m) < k-1+1 = k.$$

Now if $d(x_i, x_m) = k-1$, we can apply (3.7) and if $d(x_i, x_m) < k-1$, the result follows by induction. \square

Proposition 3.

$$\frac{1}{\lambda e(x_i) + 1} \quad \text{for } i = 1, \dots, n \quad (3.9)$$

Proof. Let $a_{ij}^{(k)}$ be the (i, j) -th entry of A^k as before, and let D be the distance matrix of the graph, i.e. $D = (d_{ij})$, where

$$= d(x_i, x_j) \quad \begin{array}{l} i = 1, \dots, n \\ j = 1, \dots, n \end{array}$$

Then the following statements about $a_{ij}^{(k)}$ and d_{ij} can be made for $k = 1, 2, \dots$ using (3.7) and (3.8):

$$\begin{array}{l} k \\ d_{ij} \geq 1 \end{array} \quad \begin{array}{l} \text{for all } i, j \text{ with } d(x_i, x_j) < \text{except} \\ \text{for the diagonal elements where } d_{ii} = 0 \end{array} \quad (3.10)$$

$$\begin{array}{l} a_{ij}^{(k)} \geq 1 \\ d_{ij} = k \end{array} \quad \begin{array}{l} \text{for all } i, j \text{ with } d(x_i, x_j) = k \end{array} \quad (3.11)$$

$$\left. \begin{array}{l} a_{ij}^{(k)} = 0 \\ d_{ij} \geq k+1 \end{array} \right\} \quad \begin{array}{l} \text{for all } i, j \text{ with } d(x_i, x_j) \geq k+1 \end{array} \quad (3.12)$$

Taking (3.10)-(12) together in matrix form it holds that

$$I + A^k + D > (k+1)U \quad \text{for } k = 1, 2, 3, \dots, \quad (3.13)$$

where I is the $n \times n$ identity matrix. Let U be the $n \times n$ matrix with all entries equal to one, then (3.13) can be written

$$I + A^k + D \geq (k+1)U$$

Therefore $v + A^k v + Dv \geq (k+1)Uv = (k+1)u,$ (3.14)

where $u = (1, 1, \dots, 1)^T$. The i -th component of Dv can be bounded as follows

$$(Dv)_i = \sum_{\ell=1}^n d_{i\ell} v_\ell = \sum_{\ell=1}^n d(x_i, x_\ell) v_\ell \leq e(x_i) \sum_{\ell=1}^n v_\ell = e(x_i) \quad (3.15)$$

Using (3.15) and $A^k v = \lambda^k v$, one obtains for the components in (3.14)

$$v_i + \lambda^k v_i + e(x_i) \geq k+1 \quad \text{for } i=1,2,\dots,n \text{ and } k=1,2,3,\dots \quad (3.16)$$

If k is chosen to be $e(x_i)$, then (3.9) follows. \square

Note that the choice $k = e(x_i)$ in (3.16) makes the bounds the best possible, since $k < e(x_i)$ yields trivial bounds and $k > e(x_i)$ yields in general some worse bounds because of the rapidly growing denominator.

Proposition 4.

$$\geq \delta / \sqrt{n-1} \quad (3.17)$$

Proof: Set $k = \delta$ in (3.16). Then

$$v_i \geq \frac{\delta+1-e(x_i)}{\lambda^\delta+1} \geq \frac{1}{\lambda^\delta+1} \quad \text{for } i = 1, \dots, n$$

Summing over i and rearranging yields (3.17). \square

All the bounds in the propositions above are rather weak. But this is to be expected, since they were proven for general graphs without any further assumptions. The bounds of Proposition 1 are better for some smaller graphs, whereas the bounds of Proposition 3 are better for larger graphs (for larger $e(x_i)$). It should be also noted that the bounds of Proposition 3 are almost sharp, if the graph is a clique. Therefore, there is not much hope to improve these bounds in all generality.

However, all bounds on the components of the eigenvector show that there is an inverse relationship between eccentricity $e(x_i)$ and the corresponding v_i .

Example: (c.f. Figure 4.32 in [4]).

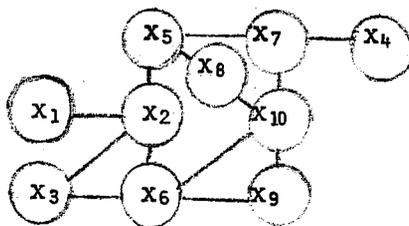


Figure 2. Example

For this graph the following table is obtained

i	v_i	$e(x_i)$	lower bound (3.1)	lower bound (3.9)
1	0.0364	4	0.00235	0.00317
2	0.1111	3	0.00998	0.01322
3	0.0756	4	0.00235	0.00317
4	0.0414	4	0.00235	0.01322
5	0.1144	3	0.00998	0.01322
6	0.1315	3	0.00998	0.01322
7	0.1330	3	0.00998	0.01322
8	0.1232	3	0.00998	0.01322
9	0.0871	3	0.00998	0.01322
10	0.1480	3	0.00998	0.01322

Here $\delta = 4$ $\lambda \approx 4.21$ and the bound of Proposition 2 and 4 yield

$$\lambda \geq 1 + \sqrt[4]{\frac{9}{4}} \approx 2.225$$

$$\lambda > \sqrt[4]{9} \approx 1.732$$

Obviously none of the bounds is very good. However, the components of v

reflect very well what by intuition one would call a "peripheral" node. The components v_1 and v_4 are smallest, and these correspond to nodes of maximal eccentricity and minimal degree.

4. Computational Aspects.

Suppose the graph is given in the form of an adjacency list, i.e. as a pair of integer arrays ADJNCY, XADJ, where XADJ(J) points to the beginning of the list of neighbors of x_j stored in the array ADJNCY. This data structure is common in sparse matrix computations [4]. With these arrays a matrix-vector multiplication of the form $v \leftarrow Au$ can be accomplished easily by the following lines of FORTRAN code:

```

      DO 10 I=1,N
      V(I)=U(I)
      DO 5J =XADJ(I),XADJ(I+1)-1
5          V(I)=V(I)+U(ADJNCY(J))
10     CONTINUE

```

Using this code for the matrix-vector multiplication, the dominant eigenvector can be determined by the power method of the Lanczos algorithm [7]. In this context the power method seems preferable, since then the whole computation can be performed in integer arithmetic without any multiplications. Since the eigenvector by itself is not of interest, but only the relative size of its components, it seems to be sufficient to do δ steps of the power method and then compare the components of v , without ever normalizing the vector. (Under some circumstances this may cause integer overflow, and the program has to be modified to handle this case).

In order to evaluate the practical usefulness of the algorithm proposed here, in connection with reordering schemes for sparse Gaussian elimination, extensive empirical tests comparable to [6] will have to be performed. But even without these tests, we can state the advantages of the algorithm:

it is very easy and economical to program,
 no starting guess has to be made, which could influence the results of the algorithm considerably,
 the bounds derived here provide at least some indication for the success of algorithm

5. An Open Problem and Conclusion.

In connection with the results from this note, an interesting graph theoretical problem arises. It is known [3] that in general the spectrum of the adjacency matrix alone does not determine the graph uniquely, i.e. there exist examples of isospectral, but not isomorphic graphs. On the other hand, it is clear that the spectrum and all eigenvectors do determine the graph uniquely. The interesting question is then: can we do with less, since the adjacency matrix is very special? More precisely one could ask: is a graph uniquely (up to isomorphisms) determined by its spectrum and its first $k, k < n$ eigenvectors? Is it possible that $k = 1$, or are there nonisomorphic, isospectral graphs, which also have the dominant eigenvector in common?

If it turns out that the dominant eigenvector, together with the spectrum determines the graph uniquely, all structural properties of the graph would be determined by some algebraic relations between $2n$ numbers. This would open up a fruitful new direction of research in graph theory and its applications

REFERENCES

- B. Aspvall and J.R. Gilbert, Graph Coloring using Eigenvalue Decomposition, Talk presented at SIAM Conf. on Applied Linear Algebra, Raleigh, 1982.
- [2] E. Cuthill and J. McKee, Reducing the Bandwidth of Sparse Symmetric Matrices, Proc. 24th Nat. Conf. Assoc. Comp. Mach., ACM Publ. P-69, 1969.
- [3] D. Cvetkoviv, M. Doob, H. Sachs, Spectra of Graphs, Academic Press New York, 1980.
- [4] A. George and J. Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice Hall, Englewood Cliffs, 1981.
- N Gibbs, W. Poole, and P. Stockmeyer, An Algorithm for Reducing the Band Width and Profile of a Sparse Matrix, SIAM J. Num. An. 13, 1976, 236-249.
- J.G. Lewis, Implementation of the Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms, ACM Transactions on Math. Software, 8, 1982, 180-189.
- [7] B. Parlett, H. Simon, and L. Stringer, On Estimating the Largest Eigenvalue with the Lanczos Algorithm, Math. Comp. 38, 1982, 153-165.