COMPUTING THE CHARACTERISTIC POLYNOMIAL OF A TREE*

Bojan MOHAR

Department of Mathematics, University of Ljubljana, Jadranska 19, 61111 Ljubljana, Yugoslavia

Received 3 November 1988
(received by the Publisher 20 September 1989)

Abstract

An algorithm is given for computing the values of the characteristic polynomial of a tree. Its time complexity is linear; hence, the polynomial is readily accessible from the tree and no computation is necessary to get the polynomial ready for applications. If necessary, the coefficients can be determined in time $O(n^2)$. This improves the complexity $O(n^3)$, reached by Tinhofer and Schreck, to $O(1)$.

1. Introduction

Tinhofer and Schreck [5] developed an algorithm with time complexity $O(n^3)$ for computing the characteristic polynomial of a tree. This is not a very surprising result, since the characteristic polynomial of any (symmetric) $n \times n$ matrix can be easily determined in time proportional to n^3 using the reduction to the tri-diagonal form. In this note, we show that *no* computation is needed to determine the characteristic polynomial $\varphi(T;x)$ of a tree T. More precisely, for any value of x we can determine the value $\varphi(T;x)$ in linear time, which means that the polynomial is readily accessible from the given tree. If we need the explicit coefficients of $\varphi(T;x)$, the application of our algorithm results in an $O(n^2)$ algorithm which calculates all the $n + 1$ coefficients of $\varphi(T;x)$.

We assume the basic knowledge of graph theory. Graphs are finite, undirected and simple. The characteristic polynomial $\varphi(T;x)$ of a graph G is just the characteristic polynomial of the adjacency matrix of G (cf. [2]).

*This work was supported in part by the Research Council of Slovenia, Yugoslavia.

2. Computing $\varphi(T;x)$

The following two results are well known (see, for example, [2,3]):

LEMMA 1

If $G_1, G_2, \ldots, G_k$ are components of the graph G, then

$$\varphi(G;x) = \varphi(G_1;x)\ldots\varphi(G_k;x).$$

LEMMA 2

Let T be a forest and $v \in V(T)$. If $v_1, v_2, \ldots, v_d$ are the neighbours of v, then

$$\varphi(T;x) = x\varphi(T-v;x) - \sum_{i=1}^{d} \varphi(T-v-v_i;x).$$

Suppose that a tree T is given and choose a vertex $v \in V(T)$. We shall assume henceforth that T is a *rooted tree with root* v. (Otherwise, it can be transformed to such a form in linear time.) This means that each vertex $w \in V(T)$, except the root v, has a unique *predecessor*, the vertex $u \in V(T)$ which is the neighbour of w and is closer to the root than w. The other neighbours of w are its *successors*, and their number $\mathrm{dout}(w) = \deg(w) - 1$ is called the *out-degree* of w. The root has $\mathrm{dout}(v) = \deg(v)$.

If $w = v$, then let $T_w := T$. If $w \in V(T)$, $w \neq v$, has the predecessor w', denote by T_w the component of $T - \{ww'\}$ which contains w. Let T'_w be the forest $T_w - w$. Notice that $T'_w = T_{w_1} \cup T_{w_2} \cup \ldots \cup T_{w_d}$, where $w_1, \ldots, w_d$ are the successors of w.

THEOREM 1

Let $w \in V(T)$.

(a) If w is a leaf, i.e. $\mathrm{dout}(w) = 0$, then

$$\varphi(T_w;x) = x \quad \text{and} \quad \varphi(T'_w;x) = 1.$$

(b) If $w_1, \ldots, w_d$ are the successors of w, then

$$\varphi(T'_w;x) = \prod_{i=1}^{d} \varphi(T_{w_i};x)$$

and

$$\varphi(T_w;x) = \varphi(T'_w;x)\,(x - \sum_{i=1}^{d} \varphi(T'_{w_i};x)/\varphi(T_{w_i};x)).$$

Proof

Part (a) is trivial, and so is the first formula of (b), by lemma 1. The last result follows from lemmas 1 and 2:

$$\begin{aligned}\varphi(T_w) &= x\,\varphi(T'_w) - \sum \varphi(T'_w - w_i)\\ &= x\,\varphi(T'_w) - \sum \varphi(T_{w_1})\ldots\varphi(T_{w_d})\,\varphi(T'_{w_i})/\varphi(T_{w_i})\\ &= \varphi(T'_w)\,[x - \sum \varphi(T'_{w_i})/\varphi(T_{w_i})]\,.\end{aligned}$$

COROLLARY 1

Given a (rooted) tree T and a number x, the value $\varphi(T;x)$ can be computed in linear time by using the formulas of theorem 1.

Proof

We use the recursion of theorem 1(a) and (b). Starting at the lowest level (vertices at the maximum distance from the root v) determine, for each vertex w, the values $\varphi(T'_w;x)$ and $\varphi(T_w;x)$. To compute these two numbers, one needs a constant number, say C, plus $D.\mathrm{dout}(w)$ time units, where D is also a constant. The total time is thus at most

$$\sum [C + D.\mathrm{dout}(w)] = Cn + D\sum \mathrm{dout}(w) = Cn + D(n-1),$$

i.e. linear.

The explicit coefficients of the polynomial are usually not needed in applications. However, the coefficients of the characteristic polynomial contain nontrivial combinatorial information about the graph, so one might want to compute them. In order to do this, we choose numbers $x_0, x_1, \ldots, x_n$ and compute the values $y_i = \varphi(T;x_i)$. By taking $x_i = \omega^i$, where ω is the primitive nth root of unity, one can compute the coefficients of the polynomial from the values $y_i = \varphi(T;\omega^i)$ in time $O(n \log n)$ by using the inverse fast Fourier transform (see, for example, [1]).

Since we have already spent $O(n^2)$ time for computing y's, one can choose some simpler algorithm for determining the coefficients of the polynomial. For example, in time $O(n^2)$, the polynomial $\varphi(T;x)$ can be transformed to the Newton divided difference interpolation form (see, for example, [4]). From this, one can compute the coefficients by the respective use of the modified Horner algorithm. This takes another $O(n^2)$ operations.

It is worth mentioning that the total space complexity is linear.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, Mass., 1974).

[2] D. Cvetković, M. Doob and H. Sachs, *Spectra of Graphs* (Academic Press, New York, 1979).

[3] C.D. Godsil and I. Gutman, On the theory of the matching polynomial, J. Graph Theory 5 (1981)137 – 144.

[4] E. Isaacson and H.B. Keller, *Analysis of Numerical Methods* (Wiley, New York, 1966).

[5] G. Tinhofer and H. Schreck, Computing the characteristic polynomial of a tree, Computing 35(1985)113 – 125.