Journal of Computational and Applied Mathematics 255 (2014) 353–361

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A fast elementary algorithm for computing the determinant of Toeplitz matrices

Zubeyir Cinkir

Department of Mathematics, Zirve University, 27260, Gaziantep, Turkey

ARTICLE INFO

Article history:
Received 29 May 2012
Received in revised form 10 March 2013

Keywords:
Toeplitz matrix
Determinant
Fast algorithm
Logarithmic time

ABSTRACT

In recent years, a number of fast algorithms for computing the determinant of a Toeplitz matrix were developed. The fastest algorithm we know so far is of order $k^2 \log n + k^3$, where n is the number of rows of the Toeplitz matrix and k is the bandwidth size. This is possible because such a determinant can be expressed as the determinant of certain parts of the n-th power of a related $k \times k$ companion matrix. In this paper, we give a new elementary proof of this fact, and provide various examples. We give symbolic formulas for the determinants of Toeplitz matrices in terms of the eigenvalues of the corresponding companion matrices when k is small.

1. Introduction

In this paper, we consider an $n \times n$ Toeplitz band matrix T_n with r and s superdiagonals as shown below:

$$T_n = \begin{pmatrix} a_0 & a_1 & \cdots & a_s & & & \mathbf{0} \\ a_{s+1} & a_0 & \cdots & & a_s & & \\ \vdots & \ddots & \ddots & & & \ddots & \\ a_{s+r} & & & & & & a_s \\ & & & & & & \vdots \\ & a_{s+r} & & & & & \\ & & \ddots & & & & a_1 \\ \mathbf{0} & & & a_{s+r} & \cdots & a_{s+1} & a_0 \end{pmatrix}_{n\times n},$$

where $r + s = k$, $a_s \neq 0$, $a_{s+r} \neq 0$, $a_i \in K$ for $i = 0, \ldots, k$ and K is a field. Without loss of generality we assume that $s \leq k$, as the determinant remains the same under taking the transpose, and our main concern is the determinant of T_n.

Finding fast algorithms to compute $\det(T_n)$ is of interest for various applications. A number of fast algorithms computing $\det(T_n)$ have been developed recently (for the tridiagonal and pentadiagonal cases, see [1–7]). Whenever $r = s = 2$, the author [1] gave an elementary algorithm computing $\det(T_n)$ in $82\sqrt{n} + O(\log n)$ operations. In this paper, we both improved and generalized the algorithm given in [1]. Namely, the algorithm we give here works for any $r \geq 1$ and $s \geq 1$, i.e., for any $k \geq 2$. Moreover, it takes $O(\frac{3}{2}k^2 \log_2 \frac{n}{k} + s^3)$ operations to compute $\det(T_n)$. The key part in this improvement is that the

E-mail addresses: zubeyirc@gmail.com, zubeyir.cinkir@zirve.edu.tr.

0377-0427/$ – see front matter
http://dx.doi.org/10.1016/j.cam.2013.05.014

computation of $\det(T_n)$ can be related to the powers of the following $k \times k$ companion matrix C associated to T_n:

$$C = \begin{pmatrix} \frac{-a_{s-1}}{a_s} & & & \\ \vdots & & & \\ \frac{-a_1}{a_s} & & & \\ \frac{-a_0}{a_s} & & I_{(k-1)\times(k-1)} & \\ \frac{-a_{s+1}}{a_s} & & & \\ \vdots & & & \\ \frac{-a_{s+r}}{a_s} & 0 & \cdots & 0 \end{pmatrix}_{k\times k}, \tag{1.1}$$

where $I_{(k-1)\times(k-1)}$ is the identity matrix of size $(k-1) \times (k-1)$. The characteristic polynomial $\mathrm{ch}_C(x)$ of C is given by

$$\det(xI - C) = x^k + \frac{a_{s-1}}{a_s}x^{k-1} + \cdots + \frac{a_0}{a_s}x^{k-s} + \frac{a_{s+1}}{a_s}x^{r-1} + \cdots + \frac{a_{s+r-1}}{a_s}x + \frac{a_{s+r}}{a_s}.$$

The fastest known algorithm to compute $\det(T_n)$ is given by D. Bini and V. Pan in 1988 (see [8]):

Theorem 1.1 (Theorem 2.7 *in Section* 2)**.** *Let M be the upper left $s \times s$ submatrix of C^n. For every integer $n \geq k$,* $\det(T_n) = (-1)^{ns}a_s^n \cdot \det(M)$.

Our main result in this paper is to give a new proof of Theorem 1.1 (see Theorem 2.7 in Section 2). Our proof is elementary. It generalizes the improved version of the method given in [1], which deals with the determinants of pentadiagonal Toeplitz matrices. This is interesting because our method basically improves an algorithm of polynomial time to obtain an algorithm of logarithmic time.

Since the result given in Theorem 1.1 requires the computation of C^n, we briefly describe various methods of computing the powers of C in Section 3.

In Section 4, we consider tridiagonal Toeplitz matrices and we relate computation of C^n with Lucas sequences as an application of what is done in Section 3. In this way, we recover the well-known closed form formulas for tridiagonal Toeplitz matrices.

Closed form formulas for $\det(T_n)$ can be given in terms of the roots of $\mathrm{ch}_C(x)$ as previously described in [9]. In Section 5, we illustrate how this is possible for pentadiagonal Toeplitz matrices, and give explicit formulas.

The algorithm given in this paper is effective as long as the number of nonzero diagonals of T is not close to the number of rows of T, i.e., when k is not close to n.

2. A fast algorithm for computing $\det(T_n)$

In this section, we give an elementary algorithm for computing $\det(T_n)$, which is the fastest known algorithm so far.

First, we describe the outline of the algorithm as follows. We move the first s column vectors of T_n and make them the last vectors, successively. This gives a matrix P. If the column vectors of T_n are $\{C_1, C_2, \ldots, C_n\}$, then the column vectors of P are $\{C_{s+1}, C_{s+2}, \ldots, C_n, C_1, \ldots, C_s\}$. Note that $\det(T_n) = (-1)^{(n-1)s}\det(P)$. Then we multiply the first column by suitable terms and add to the last s columns of P so that the only nonzero entry in the first row will be a_s. If the resulting matrix is P', $\det(P)$ is nothing but a_s times the determinant of the cofactor $P'_{1,1}$ of P'. We note that $P'_{1,1}$ is a matrix of size $(n-1) \times (n-1)$, and it is of similar form as P. Following the same procedure applied to P, we relate $\det(P'_{1,1})$ to the determinant of a matrix of size $(n-2) \times (n-2)$. Continuing in this way, the problem of computing $\det(T_n)$ can be reduced to the computation of the determinant of a $k \times k$ matrix, and this matrix can be computed easily as it is n-th power of a $k \times k$ companion matrix. Next, we describe this algorithm in detail. To be precise, we introduce some notations and deduce some results.

Let $f : \mathbb{R}^k \longrightarrow \mathbb{R}^k$ be a linear transformation given by $f([x_1, x_2, \ldots, x_k]^t) = [x_2 - x_1\frac{a_{s-1}}{a_s}, x_3 - x_1\frac{a_{s-2}}{a_s}, \ldots, x_{s+1} - x_1\frac{a_0}{a_s}, x_{s+2} - x_1\frac{a_{s+1}}{a_s}, \ldots, x_k - x_1\frac{a_{k-1}}{a_s}, -x_1\frac{a_k}{a_s}]^t$, where v^t is the transpose of a vector v. Let F be a map sending a $k \times s$ matrix A to another $k \times s$ matrix $F(A)$ by applying f to every column of A. That is, if the columns of A are $\{C_1, C_2, \ldots, C_s\}$, then the columns of $F(A)$ are $\{f(C_1), f(C_2), \ldots, f(C_s)\}$.