对称 Toeplitz 矩阵相乘的快速算法[*1)]

沈 光 星

(杭州师范学院计算机系)

THE FAST ALGORITHM FOR MULTIPLICATION OF THE SYMMETRIC TOEPLITZ MATRICES

Shen Guangxing

(*Department of Computer, Hangzhou Teachers' College*)

Abstract

In this paper, we give the fast algorithm for multiplication for two n-order symmetric Toeplitz matrices, proving that the time complexity of this algorithm are n^2 (multiplication) and $2n^2-4n+3$ (addition), and that the space complexity of this algorithm is n^2+4.

§1. 引　　言

在数字信号处理的领域中，经常会遇到一种特殊形状的 Toeplitz 矩阵

$$\begin{bmatrix} t_1 & t_2 & t_3 & \cdots & t_n \\ t_2 & t_1 & t_2 & \cdots & t_{n-1} \\ t_3 & t_2 & t_1 & \cdots & t_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ t_n & t_{n-1} & t_{n-2} & \cdots & t_1 \end{bmatrix}. \tag{1}$$

它除了具有一般 T 型矩阵的特点(主对角线上的各元素彼此相等，平行于主对角线上的元素也彼此相等，矩阵中的元素关于次对角线对称)外，还是一个对称矩阵，即形如 (1) 的矩阵是对称 T 型矩阵. 因它可由矩阵第一行的元素唯一确定，故可简记为 $\mathrm{ST}(t1,t2,\cdots,tn)\in$ STM.

关于对称 T 型系统的快速算法，已有不少研究成果，如求逆的 Trench 算法，解线性方程组的 Levinson 算法等[1,2,3]. 本文研究两个 n 阶对称 T 型矩阵相乘的快速算法.

两个 n 阶对称 T 型矩阵的乘积，一般不再是 T 型矩阵，甚至既不是对称矩阵，又不是次对称矩阵. 按照通常的矩阵相乘算法硬乘，若利用 n^2+2n+3 个存贮单元，需 n^3 次乘法和 $n^2(n-1)$ 次加法. 若利用 $2n^2+3$ 个存贮单元，也需 n^2 次乘法和 $n^2(n-1)$ 次加法. 而本文给出快速相乘算法，只需 n^2+4 个存贮单元，n^2 次乘法和 $2n^2-4n+3$ 次加法，可见计算量和存贮量都大为减少，而且容易编制程序在计算机上实现.

* 1995 年 6 月 12 日收到.

1) 国家和浙江省自然科学基金资助课题.

§2. 快速相乘算法

为讨论两个 n 阶对称 T 型矩阵相乘的快速算法，先给出乘积矩阵的一个重要性质. 设

$$A = \mathrm{ST}(a_1, a_2, \cdots, a_n) \in \mathrm{STM},$$

$$B = \mathrm{ST}(b_1, b_2, \cdots, b_n) \in \mathrm{STM},$$

则 $C = AB$ 的元素 $C_{i,j}$ 满足以下关系式：

$$C_{i,j} = C_{n+1-i,n+1-j}, \quad i,j = \overline{1,n}. \tag{2}$$

(2) 式的验证是很方便的，因为 $C_{i,j}$ 与 $C_{n+1-i,n+1-j}$ 都是对相同的 n 项求和，仅求和的次序相反，其值当然相等.

(2) 式表明，计算 $C = AB$, 只要算出矩阵 C 的上三角部分元素即可. 对角线以下第 i 行元素刚好是对角线以上第 $n+1-i$ 行元素按相反次序的排列. 同样，对角线以下第 j 列的元素也刚好是对角线以上第 $n+1-j$ 列元素按相反次序的排列，对角线上后半部的元素刚好是前半部元素按相反次序的排列. 因此，先将 A, B 分别分裂为上下两个三角形矩阵，并分别计算

$$R = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ 0 & a_1 & a_2 & \cdots & a_{n-1} \\ 0 & 0 & a_1 & \cdots & a_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_1 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_n \\ 0 & b_1 & b_2 & \cdots & b_{n-1} \\ 0 & 0 & b_1 & \cdots & b_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & b_1 \end{bmatrix} = (r_{i,j})_{n*n},$$

$$G = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ a_2 & 0 & 0 & \cdots & 0 \\ a_3 & a_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_n & a_{n-1} & a_{n-2} & \cdots a_2 & 0 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 & \cdots & b_n \\ 0 & b_1 & b_2 & \cdots & b_{n-1} \\ 0 & 0 & b_1 & \cdots & b_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & b_1 \end{bmatrix} = (g_{i,j})_{n*n},$$

$$H = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ 0 & a_1 & a_2 & \cdots & a_{n-1} \\ 0 & 0 & a_1 & \cdots & a_{n-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ b_2 & 0 & 0 & \cdots & 0 \\ b_3 & b_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ b_n & b_{n-1} & b_{n-2} & \cdots b_2 & 0 \end{bmatrix} = (h_{i,j})_{n*n}$$

的上三角部分元素，然后相加得到矩阵 C 的上三角部分元素，即

$$R + G + H = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \cdots & c_{1,n} \\ * & c_{2,2} & c_{2,3} & \cdots & c_{2,n} \\ * & * & c_{3,3} & \cdots & c_{3,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ * & * & * & \cdots & c_{n,n} \end{bmatrix}.$$

由 [4] 知， R 是上三角形 T 型矩阵，其元素满足以下的递推关系式：

$$\begin{cases} r_{1,j} = \sum_{k=1}^{j} a_k b_{j+1-k}, & j = \overline{1,n}, \\ r_{i,j} = r_{1,j+1-i}, & i = \overline{2,n},\ j = \overline{i,n}. \end{cases} \tag{3}$$

不难验证 G, H 矩阵的元素满足下面的递推关系式 (只要取上三角形部分):

$$\begin{cases} g_{i,j} = 0, & j = \overline{1,n}, \\ g_{i,j} = g_{i-1,j-1} + a_i b_j, & i = \overline{2,n},\quad j = \overline{i,n}, \end{cases} \tag{4}$$

$$\begin{cases} h_{i,n} = 0, & i = \overline{1,n}, \\ h_{i,j} = h_{i+1,j+1} + a_{n+1-i,n+1-j}, & i = \overline{n-1,1},\quad j = \overline{i,n-1}. \end{cases} \tag{5}$$

若令 $D = (d_{i,j})_{n*n}$, 其中

$$d_{i,j} = a_i b_j, \quad i,j = \overline{1,n}, \tag{6}$$

则 (3)—(5) 式可分别改写为

$$\begin{cases} r_{1,j} = \sum_{k=1}^{j} d_{k,j+1-k}, & j = \overline{1,n}, \\ r_{i,j} = r_{1,j+1-i}, & i = \overline{2,n},\quad j = \overline{i,n}, \end{cases} \tag{3'}$$

$$\begin{cases} g_{1,j} = 0, & j = \overline{1,n}, \\ g_{i,j} = g_{i-1,j-1} + d_{i,j}, & i = \overline{2,n},\quad j = \overline{i,n}, \end{cases} \tag{4'}$$

$$\begin{cases} h_{i,n} = 0, & i = \overline{1,n}, \\ h_{i,j} = h_{i+1,j+1} + d_{n+1-i,n+1-j}, & i = \overline{n-1,1},\quad j = \overline{i,n-1}. \end{cases} \tag{5'}$$

综上所述，两个 ST 型矩阵相乘的快速算法如下 (均在矩阵 D 内进行):

第一步. 按公式 (6) 计算矩阵 D, 即执行

$$d_{i,j} := a_i * b_j, \quad i,j = \overline{1,n}.$$

第二步. 按公式 (3′) 计算矩阵 R 的第一行元素，并把结果放在矩阵 D 的第一行上，即执行

$$d_{1,j} := \sum_{i=1}^{j} d_{i,j+1-i}, \quad j = \overline{2,n}.$$

第三步. 分别按公式 (4′) 和 (3′) 计算矩阵 G, R 的对角线以上部分元素，并把对应元素相加后放在矩阵 D 的相应位置上 (注意 G 的第一行元素均为 O, 不必计算)，合并起来只要执行

$$d_{i,j} := d_{i,j} + d_{i-1,j-1}, \quad i = \overline{2,n}, \quad j = \overline{i+1,n}.$$

第四步. 按公式 (3′) 计算矩阵 R 对角线上的元素，然后除 $d_{1,1}$ 外，每个对角线元素均减去 $d_{1,1}$, 并把结果分别放在矩阵 D 的对角线上及第一列上，合并起来只要依次执行

$$d_{i,i} := d_{i,i} + d_{i-1,i-1}, \quad i = \overline{3,n},$$

$$d_{i,1} := d_{i,i}, \quad i = \overline{2,n}.$$

第五步. 按公式 (5′) 计算矩阵 H 的上三角形部分元素 (注意, H 矩阵的最后一列全为 0, 不必计算和存贮, 矩阵 H 对角线上的元素, 刚好是存放在矩阵 D 第一列上元素的倒序, 因而也不必计算和存贮), 但为了节省存贮单元, 可先将矩阵 H 对角线以上第 $n+1-j$ 列 (第 n 列除外) 的元素, 按相反次序放到矩阵 D 的对角线以下第 j 列 (第一列除外) 中去, 而 $d_{i,2}=h_{n+1-i,n-1}$, $i=\overline{2,n}$, 合并起来只要执行

$$d_{i,j}:=d_{i,j}+d_{i-1,j-1},\quad i=\overline{3,n},\quad j=\overline{3,i-1}.$$

第六步. 先把矩阵 D 第一列的前 $[(n+1)/2]$ 个元素 (除 $d_{1,1}$ 外) 均加上 $d_{1,1}$, 然后按相反次序加到矩阵 D 的对角线后 $[(n+1)/2]$ 个元素中去, 最后把矩阵 D 的对角线上后 $[(n+1)/2]$ 个元素, 按相反次序放到前 $[(n+1)/2]$ 个元素的位置上去, 即依次执行

$$\begin{aligned}
&d_{i,1}:=d_{i,1}+d_{1,1}, && i=\overline{2,[(n+1)/2]},\\
&d_{n+1-i,n+1-i}:=d_{n+1-i,n+1-i}+d_{i,1}, && i=\overline{1,[(n+1)/2]},\\
&d_{i,i}:=d_{n+1-i,n+1-i}, && i=\overline{1,[(n+1)/2]}.
\end{aligned}$$

第七步. 把矩阵 D 对角线以下第 j 列 (第一列除外) 的元素, 按相反次序加到矩阵 D 对角线以上第 $n+1-j$ 列 (第 n 列除外) 的元素中去, 即执行

$$d_{i,j}:=d_{i,j}+d_{n+1-i,n+1-j},\quad i=\overline{1,n-2},\ j=\overline{i+1,n-1}.$$

第八步. 把矩阵 D 对角线以上第 i 行的元素, 按相反次序放到矩阵 D 对角线以下第 $n+1-i$ 行的位置上去, 即执行

$$d_{i,j}:=d_{n+1-i,n+1-j},\quad i=\overline{2,n},\ j=\overline{1,i-1}.$$

至此, 矩阵 D 即为要求的乘积矩阵 C.

§3. 复杂性分析

先分析时间复杂性. 显然, 第一步只需 n^2 次乘法, 第二步只需 $n(n-1)/2$ 次加法, 第三步只需 $(n-1)(n-2)/2$ 次加法, 第四步只需 $n-2$ 次加法, 第五步只需 $(n-2)(n-3)/2$ 次加法, 第六步只需 n 次加法, 第七步只需 $(n-1)(n-2)/2$ 次加法, 第八步不需要四则运算, 因而总的时间复杂性为: n^2 次乘法和 $2n^2-4n+3$ 次加法.

再分析空间复杂性. 若先将 A 的第一列放到矩阵 D 的第 n 列, B 的第一行前 $n-1$ 个元素放到矩阵 D 的第 n 行 (除 $d_{n,n}$ 外), 把 b_n 放到单元 b, 因快速相乘算法的八个步骤都在矩阵 D 中进行, 所以该算法除另需 4 个单元 (存放 n,i,j,b_n) 外, 只要 n^2 个存贮单元, 可见空间复杂性为 n^2+4.

§4. 算　　例

利用本文给出的快速相乘算法, 我们编制了 PASCAL 程序, 在 IBM-PC 机上验证了算法的正确性 (程序附后). 下面举例说明计算步骤:

算例. 设 $n=5$, $A=\mathrm{ST}(1,2,3,4,5)$, $B=\mathrm{ST}(2,3,4,5,6)$.

$$\begin{bmatrix} & & & & 1\\ & & & & 2\\ & & & & 3\\ & & & & 4\\ 2 & 3 & 4 & 5 & 5 \end{bmatrix}_{6\Rightarrow b} \xrightarrow{\text{第一步}} D=\begin{bmatrix} 2 & 3 & 4 & 5 & 6\\ 4 & 6 & 8 & 10 & 12\\ 6 & 9 & 12 & 15 & 18\\ 8 & 12 & 16 & 20 & 24\\ 10 & 15 & 20 & 25 & 30 \end{bmatrix} \xrightarrow{\text{第二步}} \begin{bmatrix} 2 & 7 & 16 & 30 & 50\\ 4 & 6 & 8 & 10 & 12\\ 6 & 9 & 12 & 15 & 18\\ 8 & 12 & 16 & 20 & 24\\ 10 & 15 & 20 & 25 & 30 \end{bmatrix} \xrightarrow{\text{第三步}}$$

$$\begin{bmatrix} 2 & 7 & 16 & 30 & 50\\ 4 & 6 & 15 & 26 & 42\\ 6 & 9 & 12 & 30 & 44\\ 8 & 12 & 16 & 20 & 54\\ 10 & 15 & 20 & 25 & 30 \end{bmatrix} \xrightarrow{\text{第四步}} \begin{bmatrix} 2 & 7 & 16 & 30 & 50\\ 6 & 6 & 15 & 26 & 42\\ 18 & 9 & 18 & 30 & 44\\ 38 & 12 & 16 & 38 & 54\\ 68 & 15 & 20 & 25 & 68 \end{bmatrix} \xrightarrow{\text{第五步}} \begin{bmatrix} 2 & 7 & 16 & 30 & 50\\ 6 & 6 & 15 & 26 & 42\\ 18 & 9 & 18 & 30 & 44\\ 38 & 12 & 25 & 38 & 54\\ 68 & 15 & 32 & 50 & 68 \end{bmatrix} \xrightarrow{\text{第六步}}$$

$$\begin{bmatrix} 70 & 7 & 16 & 30 & 50\\ 8 & 46 & 15 & 26 & 42\\ 20 & 9 & 38 & 30 & 44\\ 38 & 12 & 25 & 46 & 54\\ 68 & 15 & 32 & 50 & 70 \end{bmatrix} \xrightarrow{\text{第七步}} \begin{bmatrix} 70 & 57 & 48 & 45 & 50\\ 8 & 46 & 40 & 38 & 42\\ 20 & 9 & 38 & 39 & 44\\ 38 & 12 & 25 & 46 & 54\\ 68 & 15 & 32 & 50 & 70 \end{bmatrix} \xrightarrow{\text{第八步}} \begin{bmatrix} 70 & 57 & 48 & 45 & 50\\ 54 & 46 & 40 & 38 & 42\\ 44 & 39 & 38 & 39 & 44\\ 42 & 38 & 40 & 46 & 54\\ 50 & 45 & 48 & 57 & 70 \end{bmatrix} = C = AB.$$

参 考 文 献

[1] N. Levinson, The weiner RMS error griterion in filter design and prediction, *J. Math. Phys.*, **25** (1947), 261—278.

[2] W.F. Trench, An algorithm for the inversion of finite toeplitz matrices, *J. SIAM.*, **12** (1964), 515—522.

[3] 徐士良编，计算机常用算法，清华大学出版社， 1994.

[4] 游兆永、李磊，关于三角形 Toeplitz 系统的复杂性，计算数学. **9**:3 (1987), 262—265.