Fast Algorithms for Brownian Matrices

Bernard Picinbono

Fast Algorithms for Brownian Matrices

Bernard Picinbono, *Senior Member, IEEE*

Laboratoire des Signaux et Systèmes (L2S), Division Signaux

Reference of this text : Reference of this text : IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-31, Num. 2 pp. 512 - 514,, 1983.

Abstract

Brownian motion is one of the most common models used to represent nonstationary signals. The covariance matrix of a discrete-time Brownian motion has a very particular structure, and is called a Brownian matrix. This note presents a number of results concerning linear problems appearing in digital signal processing with Brownian matrices. In particular, it is shown that fast algorithms used for Toeplitz matrices are simpler and faster for Brownian matrices. Examples are given to illustrate the different results presented in the note.

I. Introduction

In recent years, many papers have been published Toeplitz matrices (T matrices) and their applications to digital signal processing. This interest in Toeplitz matrices is mainly due to the fact that when they are

Manuscript received

B. P. is with the Laboratoire des Signaux et Systèmes (L2S), a joint laboratory of the C.N.R.S. and the École Supérieure d'Électricité, Plateau de Moulon, 3 rue Joliot–Curie 91192, Gif sur Yvette, France. The L2S is associated with the University of Paris–Sud, France. E-mail: bernard.picinbono@lss.supelec.fr, .

symmetric and positive definite, such matrices can be considered as the covariance matrices of discrete-time stationary random signals.

It is well known that many signals of practical interest are not stationary, in which case the Toeplitz property of the covariance matrix disappears.

As Brownian motion is a very common model for nonstationary signals, we shall consider the matrices introduced as covariance matrices of such signals. By analogy, such matrices are called Brownian matrices (B matrices). We shall show that most of the problems considered in the Toeplitz case have a simpler and faster solution in the Brownian case. This is particularly true for the solution of linear equations and for matrix inversion. It should be mentioned that similar results in a slightly different context were independently derived in [1], which appeared while this note was still under review.

II. Brownian Motion i Signal processing

Brownian motion is an important tool for modeling continuous- time random signals. It is the simplest and most rigorous way to introduce the classical white noise, which is very often used in signal theory [2] . But as we are most interested in discrete-time signals, we shall use discrete-time Brownian motion, sometimes called the random walk stochastic process. This signal can be written

$$b(n) = \sum_{k=0}^{n} u(k), \tag{1}$$

where the random variables $u(k)$, increments of $b(n)$, are zero mean, independent, and with variance $c(k)$. Then the covariance of $b(n)$ becomes

$$R(m,n) \triangleq E[b(m)b(n)] = \sum_{j=0}^{m\wedge n} c(j) \triangleq r(m \wedge n) \tag{2}$$

Moreover, the covariance of the signal $x(n)$ sum of a discrete-time Brownian motion and an independent white noise w(n) is

$$R(m,n) = r(m \wedge n) + \sigma_n^2 \delta_{mn}, \tag{3}$$

where σ_n^2 is the variance of $w(n)$. The extraction of a signal $s(n)$ from a noise modeled by $x(n)$ requires signal processing techniques such as detection and estimation. For these techniques, it is in general convenient to represent the signals in vectorial form [3], and then to solve the systems involving the covariance matrices associated with (3).

III. Brownian Matrices

Let us consider the random vector $\mathbf{x}$ with components $[x(0), x(1), \ldots, x(N-1)]$. Its covariance matrix $\mathbf{R}$ is an $N \times N$ matrix, and we assume that its elements are given by (3). In all of the following, we shall call this matrix a *Brownian matrix* (B matrix). Such a matrix is defined by the $2N$ coefficients r_i and σ_i, $0 \leq i \leq N-1$ where $r_i = r(i)$ is defined in (2). By introducing

$$v_i \triangleq r_i + \sigma_i^2, \tag{4}$$

where the v_i are the diagonal elements of the B matrix, we can write such a matrix in the form

$$B(\{v_i\},\{r_i\}) = \begin{pmatrix} v_0 & r_0 & r_0 & \dots & r_0 & r_0 \\ r_0 & v_1 & r_1 & \dots & r_1 & r_1 \\ r_0 & r_1 & v_2 & \dots & r_2 & r_2 \\ \vdots & \vdots & \vdots & & & \\ r_0 & r_1 & r_2 & \dots & r_{N-2} & v_{N-1} \end{pmatrix} \tag{5}$$

It is interesting to compare this structure to that obtained for a symmetric Toeplitz matrix (T matrix), which is defined by its first row:

$$T_1(\{r_i\}) = [r_0, r_1, r_2, \dots, r_{N-2}, v_{N-1}]. \tag{6}$$

Let us examine some cases of particular interest. If there is no white noise component $w(n)$ in (3), then $\sigma_i = 0$ and $v_i = r_i$. In this case, we say that B is a pure B matrix, which is obviously determined only by the r_i, and may be written $B(\{ri\})$. Moreover, we can write any B matrix in the form

$$B(\{v_i\},\{r_i\}) = B(\{r_i\}) + \mathrm{diag}(\{v_i - r_i\}), \tag{7}$$

where $\mathrm{diag}(\{\lambda_i\})$ means a diagonal matrix with elements λ_i.

For the solution of linear problems, it is very important to know conditions on the matrix elements which secure that the covariance matrix is positive definite. In the case of a Toeplitz matrix defined by (6), the conditions on the r_i are very complex. That is not the case for B matrices.

Proposition 1: A pure B matrix $B(\{r_i\})$ is positive definite if and only if the sequence r_i is increasing ($0 < r_i < r_{i+1}$).

Proof: We consider first the if part. If the sequence is increasing, the numbers $c(i) = r_i - r_{i-1}$, $c(0) = r_0$, are positive. Then they can be considered as the variance of a sequence of N uncorrelated random variables $y(0), y(1), \dots, y(N-1)$ which are the components of a vector y. Its covariance matrix is, of course, $\mathrm{diag}(\{c_i\})$. Let us introduce the vector $\mathbf{b}$ with components

$$b(i) = \sum_{j=0}^{i} y(j), \quad 0 \leq i \leq N-1. \tag{8}$$

The covariance matrix of $\mathbf{b}$ is $B(\{r_i\})$, which shows that this matrix is positive definite.

For the "only if" part, we assume that $B(\{r_i\})$ is a pure B matrix which is also positive definite. Then it is the covariance matrix of a random vector $\mathbf{b}$ (see [2, p. 121) with components $b((i)$. From the B matrix structure, we have

$$E[b(i)^2] = r_i;\ E[b(i+1)^2] = r_{i+1};\ E[b(i)b(i+1)] = r_i. \tag{9}$$

Then from the Schwarz inequality, we obtain $r_i^2 < r_i r_{i+1}$, and as $r_i > 0$, we have $r_i < r_{i+1}$, which means that the sequence is increasing.

Proposition 2: If $r_i < r_{i+1}$ and $v_i > r_i$, then the B matrix $B(\{v_i\}, \{r_i\})$ is positive definite.

The proof follows immediately from (7) and Proposition 1. The converse is obviously not true because $B(\{v_i\}, \{r_i\})$ can be positive definite even if $B(\{r_i\})$ or $\mathrm{diag}(\{v_i - r_i\}$ do not have this property.

For the following discussion, it is also important to notice that the B and T matrices have a very similar structural property called the *accumulation property* which is at the root of fast algorithms. By this property, we mean that two successive columns of the triangular part of such matrices have only two different elements, After transposition, two such columns for a B matrix given by (7) are

$$\begin{array}{ccccccc} r_0, & r_1, & r_2, & \ldots, & r_{k-1}, & v_k & \\ r_0, & r_1, & r_2, & \ldots, & r_{k-1}, & r_k, & v_{k+1} \end{array}$$

and the new elements are r_k and v_{k+1}. For T matrices, such columns are

$$\begin{array}{cccccc} r_k, & r_{k-1}, & r_{k-2}, & \ldots, & r_0 & \\ r_{k+1}, & r_k, & r_{k-1}, & \ldots, & r_1 & r_0 \end{array}$$

and the accumulation property appears in the reversed order and with only one different element, r_{k+1}. This accumulation property, which is the origin of the fast algorithms presented in the next section, has been presented in another way in [1], and the B matrices studied here in connection with Brownian motion are called diagonal innovation matrices in this paper.

IV. Recursive Solution of the Linear Equations

In this section we shall study the solution of the linear equation $\mathbf{Ra} = \mathbf{b}$, where $\mathbf{R}$ is a positive definite (p.d.) B matrix and $\mathbf{b}$ is a given vector. This problem is fundamental in many aspects of digital signal processing. Indeed, as $\mathbf{R}$ is p.d., it can be considered to be the covariance matrix of a random vector, and $\mathbf{Ra} = \mathbf{b}$ is a basic equation in detection and estimation problems. Our aim is to calculate the vector $\mathbf{a}$ by a method recursive on the order, as in the Levinson algorithm. Let us assume that $\mathbf{R}$ is given by (5) and that the components of $\mathbf{b}$ are $b_1, b_2, \ldots, b_N$.

From the accumulation property, it is clear that the matrix obtained from $\mathbf{R}$ by taking only the $n+1$ first rows and column is still a positive definite B matrix $\mathbf{R}_{n+1} = B[v_0, v_1, \ldots, v_n; r_0, r_1, \ldots, r_{n+1}]$. Thus, it can be partitioned in the form

$$\mathbf{R}_{n+1} = \begin{bmatrix} \mathbf{R}_n & \mathbf{u}_n \\ \mathbf{u}_n^T & v_n \end{bmatrix}; \quad \mathbf{u}_n^T = [r_0, r_1, \ldots, r_{n+1}]. \tag{10}$$

Let us call $\mathbf{a}_{n+1}$ the solution of the linear equation

$$\mathbf{R}_{n+1}\mathbf{a}_{n+1} = \mathbf{b}_{n+1} \tag{11}$$

where $\mathbf{b}_{n+1}$ is the vector with components $(b_1, b_2, \ldots, b_{n+1})$.

For the following discussion, it is convenient to partition the vectors $\mathbf{a}_{n+1}$ and $\mathbf{b}_{n+1}$ in a form similar to (10), i.e.,

$$\mathbf{a}_{n+1} = \begin{bmatrix} \mathbf{a}_n^{n+1} \\ a_{n+1}^{n+1} \end{bmatrix}; \quad \mathbf{b}_{n+1} = \begin{bmatrix} \mathbf{b}_n \\ b_{n+1} \end{bmatrix}. \tag{12}$$

Note that while the vector $\mathbf{b}_n$ has, by definition, the accumulation property, this is not true of the vector $\mathbf{a}_n$ that is, $\mathbf{a}_n^{n+1} \neq \mathbf{a}_n$, as we shall verify shortly.

By using (10) and (12), (11) can easily be written in the form

$$\mathbf{R}_n\mathbf{a}_n^{n+1} + a_{n+1}^{n+1}\mathbf{u}_n = \mathbf{b}_n \tag{13}$$

$$\mathbf{u}_n^T\mathbf{a}_n^{n+1} + a_{n+1}^{n+1}v_n = b_{n+1}. \tag{14}$$

As $\mathbf{R}_n$ is positive definite, we can calculate $\mathbf{R}_n^{-1}$ and introduce the vector

$$\mathbf{w}_n \stackrel{\triangle}{=} \mathbf{R}_n^{-1}\mathbf{u}_n. \tag{15}$$

Since from (11) $\mathbf{a}_n = \mathbf{R}_n^{-1}\mathbf{b}\mathbf{v}_n$, we can write (13) and (14) in the form

$$\mathbf{a}_n^{n+1} = \mathbf{a}_n - a_{n+1}^{n+1}\mathbf{w}_n \tag{16}$$

$$a_{n+1}^{n+1} = \beta_n(a)/\alpha_n \tag{17}$$

$$\beta_n(a) = b_{n+1} - \mathbf{u}_n^T\mathbf{a}_n\ ;\ \alpha_n = v_n - \mathbf{u}_n^T\mathbf{w}_n. \tag{18}$$

The previous relations show that it is now necessary to calculate the vector $\mathbf{w}_n$ to obtain the vector $\mathbf{a}_n$. In general, this calculation requires the solution of (15), which gives a new sequence of linear equations. But in our case, as the $\mathbf{u}_n$ are deduced from a B matrix [see (10)] , they possess the accumulation property as do vectors $\mathbf{b}$. Therefore, we can apply the same method to solve (15) and (11) as well, and we then obtain the following recursion for $\mathbf{w}_{n+1}$ partitioned as $\mathbf{a}_{n+1}$ in (12):

$$\mathbf{w}_n^{n+1} = \mathbf{w}_n - w_{n+1}^{n+1}\mathbf{w}_n = (1 - w_{n+1}^{n+1})\mathbf{w}_n \tag{19}$$

$$w_{n+1}^{n+1} = \beta_n(w)/\alpha_n \tag{20}$$

$$\beta_n(w) = r_n - \mathbf{u}_n^T \mathbf{w}_n. \tag{21}$$

This yields a reccusrion from $(\mathbf{a}_n, \mathbf{w}_n)$ to $(\mathbf{a}_{n+1}, \mathbf{w}_{n+1})$. In this recursion, the only calculations necessary are those for the coefficients α_n, β_n which require the computation of the scalar products $\mathbf{u}_n^T \mathbf{a}_n$ and $\mathbf{u}_n^T \mathbf{w}_n$. We thus require $2n$ operations (multiplication-addition) to calculate such products; however, we shall now show that this number can be reduced. (This is not possible for T matrices.) For this purpose, we shall show that the coefficients α_n and β_n can be obtained recursively without any scalar product. To simplify the discussion, let us define

$$\beta_n = \beta_n(w).\ ;\ \delta_n = \beta_n(a). \tag{22}$$

From (18) and (20), we obtain $\beta_n = \alpha_n + r_n - v_n$. Moreover, we can write

$$\alpha_{n+1} = v_{n+1} - (u_{n+1} w_{n+1}^{n+1} + \mathbf{u}_n^T \mathbf{w}_n^{n+1}), \tag{23}$$

and by using (19) and (20l), we obtaine asily

$$\alpha_{n+1} = \alpha_n + v_{n+1} - v_n - \beta_n^2/\alpha_n. \tag{24}$$

The same procedure used for δ_n yields

$$\delta_{n+1} = \delta_n(1 - \beta_n/\alpha_n) + b_{n+2} - b_{n+1}. \tag{25}$$

Then we can calculate $(\alpha_{n+1}, \beta_{n+1}, \delta_{n+1})$ from $(\alpha_n, \beta_n, \delta_n)$ without any scalar product. It is easy to find that a complete recursive step needs $n + 9$ additions and $2n + 6$ multiplications, while the same operation needs in the Toeplitz case $4n+1$ additions and $4n+6$ multiplications. Consequently, we see that for situations that warrant its use, a Brownian model is less "expensive" than a corresponding Toeplitz model. Moreover, it is well known that the speed of an algorithm is not only a function of the number of operations, but also of the structure of operations, e.g., whether the computations are done in sequence (as scalar operation) or in parallel (as in vector processor). For a sequential operation, it will usually be less complex to calculate $a_{n+}^{n+1} \mathbf{w}_n$ as in (16), than $\mathbf{u}_n^T \mathbf{a}_n$ as in (18), even if the number of multiplications is the same because of the reduced number of memory access. We have seen that for B matrices, there is no scalar product to calculate, and this advantage does not appear for T matrices.

Finally, we can notice that the same procedure can be applied for the inversion of B matrices, and some examples and extensions will be presented in another paper.

REFERENCES

[1] G. Carayannis, N. Kalouptsidis, and D. B. Manolakis, "Fast recursive algorithms for a class of linear equation ", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 221-239, Apr. 1982.

[2] A. Blanc-Lapierre and B. Picinbono, *Fonctions Aléatoires*. Paris, France: Masson, 1981, pp. 109, 166.

[3] B. Picinbono, "A geometrical interpretation of signal detection and estimation", *IEEE Trans. Inform. Theory,*, vol. IT-26, pp. 493-497, 1980.