연세경제연구 제 Ⅲ 권 제 1 호
Yonsei Economic Studies, Vol. Ⅲ, No. 1 (March 1996)

An Algorithm to Calculate the AR Parameters of a Mixed ARMA Model through the Extended Yule-Walker Equations

Chul Eung Kim and Byoung Seon Choi

A recursive algorithm is presented to solve the extended Yule-Walker equations of a mixed autoregressive moving-average process for the authoregressive parameters. The algorithm is based on the bordering technique for matrix inversion and is computational more efficient than any other existing method. Besides it is a generalization of the Levinson-Durbin algorithm of pure autoregressive processes to mixed autoregressive moving-average processes.

Ⅰ. Introduction

Consider the autoregressive moving-average (ARMA) model of orders p and q,

$$\phi(B)y_t = \theta(B)v_t \tag{1}$$

where $\phi(z) = -\phi_0 - \phi_1 z - \cdots - \phi_p z^p$, $\theta(z) = -\theta_0 - \theta_1 z - \cdots - \theta_q z^q$, $\phi_0 = \theta_0 = -1$, $\phi_p \neq 0$, $\theta_q \neq 0$, B is the backshift operator i.e., $B^k y_t = y_{t-k}$ and $\{v_t\}$ is a sequence of independent and identically distributed random variables with means 0 and variances σ^2. We assume that the model is stationary, i.e., the equation $\phi(z) = 0$ has all the roots outside the unit circle. We also assume that the equations $\phi(z) = 0$ and $\theta(z) = 0$ have no common roots. Let $\{\sigma(j)\}$ and $\{\rho_j\}$ be the autovariance function (ACVF) and the autocorrelation

Department of Applied Statistics Yonsei University Seoul, 120-749, Korea.

function (ACRF) of the process defined by

$$\rho(j) = \mathrm{Cov}\,(y_t, y_{t+j})$$

and

$$\rho_j = \sigma(j)/\sigma(0)$$

It is well-known that the ACRF of the stationary process (1) satisfies the following extended Yule-Walker equations (EYW) :

$$\rho_j = \phi_1\rho_{j-1} + \cdots + \phi_p\rho_{j-p} \qquad j = q + 1, q + 2, \cdots \tag{2}$$

When the ACRF is given, the AR parameters $\phi_1, \cdots, \phi_p$ can be obtained by solving the EYW with $j = q + 1, \cdots, q + p$. However, if we use other p EYW with $j = q + 1, \cdots, q + i + p - 1$ for any $i > 1$, then we obtain the same solution. Since the EYW are of linear forms, we may use the Gauss elimination method to solve them. Since the EYW constitute a Toeplitz system of equations, we may also use Trench and the Zohar algorithms about Toeplitz matrices [6], [9]~[11].

The purpose of this correspondence is to present a recursive algorithm to solve the EYW for AR parameters that is computationally more efficient than any other exsiting one. The algorithm is based on the bordering technique of matrix inversion [4, Section 15]. It will be shown that the algorithm is a generalized Levison-Durbin algorithm of pure AR processes to mixed ARMA processes.

Ⅱ. An Algorithm

For a positive integer k, we define a k by k Toeplitz matrix $B(k, i)$ by

$$B(k, i) = \begin{bmatrix} \rho_i & \rho_{i-1} & \cdots & \rho_{i-k+1} \\ \rho_{i+1} & \rho_i & \cdots & \rho_{i-k+2} \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{i+k-1} & \rho_{i+k-2} & \cdots & \rho_i \end{bmatrix}$$

Denote its determinant by $d(k, i)$, and define two vectors by

$$\underset{\sim}{\rho}(k, i) = (\rho_{i+1}, \cdots, \rho_{i+k})'$$

and

$$\underset{\sim}{r}(k, i) = (\rho_{i-k}, \cdots, \rho_{i-1})'$$

Then, our problem becomes to solve the Toeplitz system of the EYW:

$$B(p, q)\underset{\sim}{\phi}(p, q) = \underset{\sim}{\rho}(p, q) \tag{3}$$

Before proposing an algorithm to solve (3), we define some vectors and matrices. We denote $(x_n, \cdots, x_2, x_1)'$ by $\tilde{\underset{\sim}{x}}$ for any vector $\underset{\sim}{x} = (x_1, x_2, \cdots, x_n)'$. For $k = 1, 2, \cdots$, and $i = 0, 1, \cdots$, if $B(k, i)$ is nonsingular, then we define

$$\underset{\sim}{\phi}(k, i) = B(k, i)^{-1}\underset{\sim}{\rho}(k, i)$$
$$\tilde{\underset{\sim}{\pi}}(k, i) = B(k, i)^{-1}\underset{\sim}{r}(k, i)$$

and

$$\lambda(k, i) = \rho_i - \tilde{\underset{\sim}{\rho}}(k, i)'B(k, i)^{-1}\underset{\sim}{r}(k, i)$$

for $i = 0, 1, \cdots$, we let $\lambda(0, i) = \rho_i$. We denote the j-th elements of $\underset{\sim}{\phi}(k, i)$ and $\underset{\sim}{\pi}(k, i)$ by $\phi_{k,j}^{(i)}$ and $\pi_{k,j}^{(i)}$, respectively. Also, we let $\phi_{k,0}^{(i)} = \pi_{k,0}^{(i)} = -1$.

Now we state a recursive algorithm to solve the generalized problem of (3):

$$B(k, i)\underset{\sim}{\phi}(k, i) = \underset{\sim}{\rho}(k, i) \quad \text{for } k = 1, 2, \cdots \text{ and } i = 0, 1, \cdots \tag{4}$$

Algorithm

[redacted] when $i = 0$, use the Levinson-Durbin algorithm:

For $k = 0$,

$$\lambda(0, 0) = 1$$
$$\underset{\sim}{\phi}(1, 0) = (\rho_1), \text{ i.e., } \phi_{1,1}^{(0)} = \rho_1$$

for $k = 1, 2, \cdots$,

$$\lambda(k, 0) = \{1 - (\phi_{k,k}^{(0)})^2\}\lambda(k-1, 0)$$
$$\phi_{k+1,k+1}^{(0)} = \{\rho_{k+1} - \tilde{\underset{\sim}{\phi}}(k, 0)'\underset{\sim}{\rho}(k, 0)\}/\lambda(k, 0)$$

For $j = 1, \cdots, k$,

$$\phi^{(0)}_{k+1,j} = \phi^{(0)}_{k,j} - \phi^{(0)}_{k+1,k+1}\phi^{(0)}_{k,k+1-j}$$

[redacted] when $i = 1, 2, \cdots$, use the following step:

For $k = 0$,

$$\lambda(0, i) = \rho_i$$

$$\underset{\sim}{\phi}(1, i) = (\rho_{i+1}/\rho_i), \text{ i.e., } \phi^{(i)}_{1,1} = \rho_{i+1}/\rho_i$$

for $k = 1, 2, \cdots$,

$$\lambda(k, i) = \{1 - \phi^{(i)}_{k,k}/\phi^{(i-1)}_{k,k}\}\lambda(k-1, i)$$

$$\phi^{(i)}_{k+1,k+1} = \{\rho_{i+k+1} - \tilde{\underset{\sim}{\phi}}(k, i)'\underset{\sim}{\rho}(k, i)\}/\lambda(k, i)$$

For $j = 1, \cdots, k$,

$$\phi^{(i)}_{k+1,j} = \phi^{(i)}_{k,j} + \phi^{(i)}_{k+1,k+1}\phi^{(i-1)}_{k,j-1}/\phi^{(i-1)}_{k,k}$$

Ⅲ. Derivation

Te derive the algorithm, we assume that $B(k, i)$ and $B(k+1, i)$ are nonsingular. The solution vector at orders $(k+1, i)$ is

$$\begin{aligned}\underset{\sim}{\phi}(k+1, i) &= B(k+1, i)^{-1}\underset{\sim}{\rho}(k+1, i)\\ &= \begin{bmatrix} B(k, i), & \underset{\sim}{r}(k, i) \\ \tilde{\underset{\sim}{\rho}}(k, i)', & \rho_i \end{bmatrix}^{-1}\begin{bmatrix} \underset{\sim}{\rho}(k, i) \\ \rho_{i+k+1}\end{bmatrix}\\ &= \begin{bmatrix} B^{1,1} & B^{1,2} \\ B^{2,1} & B^{2,2}\end{bmatrix}\begin{bmatrix} \underset{\sim}{\rho}(k, i) \\ \rho_{i+k+1}\end{bmatrix}\end{aligned}$$

where

$$B^{1,1} = B(k, i)^{-1} + B(k, i)^{-1}\underset{\sim}{r}(k, i)\tilde{\underset{\sim}{\rho}}(k, i)'B(k, i)^{-1}/\lambda(k, i)$$

$$B^{1,2} = -B(k, i)^{-1}\underset{\sim}{r}(k, i)/\lambda(k, i)$$

$$B^{2,1} = -\tilde{\underset{\sim}{\rho}}(k, i)'B(k, i)^{-1}/\lambda(k, i)$$

and

$$B^{2,2} = 1/\lambda(k, i)$$

Here, the inverse of the partitioned matrix is obtained by the bordering technique [4, pp. 107~111]. Then, we know that

$$\begin{aligned}\phi^{(i)}_{k+1,k+1} &= B^{2,1}\underset{\sim}{\rho}(k, i) + B^{2,2}\rho_{i+k+1} \\ &= \{\rho_{i+k+1} - \tilde{\underset{\sim}{\rho}}(k, i)^t B(k, i)^{-1}\underset{\sim}{\rho}(k, i)\}/\lambda(k, i) \\ &= \{\rho_{i+k+1} - \tilde{\underset{\sim}{\rho}}(k, i)^t \underset{\sim}{\phi}(k, i)\}/\lambda(k, i)\end{aligned}$$

and

$$\begin{aligned}&B^{1,1}\underset{\sim}{\rho}(k, i) + B^{1,2}\rho_{i+k+1} \\ &= \underset{\sim}{\phi}(k, i) - \tilde{\underset{\sim}{\pi}}(k, i)\{\rho_{i+k+1} - \tilde{\underset{\sim}{\rho}}(k, i)^t B(k, i)^{-1}\underset{\sim}{\rho}(k, i)\}/\lambda(k, i) \\ &= \underset{\sim}{\phi}(k, i) - \tilde{\underset{\sim}{\pi}}(k, i)\phi^{(i)}_{k+1,k+1}\end{aligned}$$

Therefore, the solution vector is

$$\underset{\sim}{\phi}(k+1, i) = \begin{bmatrix} \underset{\sim}{\phi}(k, i) - \tilde{\underset{\sim}{\pi}}(k, i)\phi^{(i)}_{k+1,k+1} \\ \phi^{(i)}_{k+1,k+1} \end{bmatrix} \tag{5}$$

We are going to represent (5) without using the π vector. For $i = 0$, it can be easily shown that $\underset{\sim}{\pi}(k, 0) = \underset{\sim}{\phi}(k, 0)$. Thus, the solution vector in (5) implies the Levinson-Durbin algorithm for pure AR cases. For $i = 1, 2, \cdots$, we consider the solution vector $\underset{\sim}{\phi}(k, i-1)$.

$$\begin{aligned}&B(k, i-1)\underset{\sim}{\phi}(k, i-1) - \rho(k, i-1) = 0 \\ (\Leftrightarrow)\quad &\sum_{j=0}^{k}\phi^{(i-1)}_{k,j}\underset{\sim}{\rho}(k, i-1-j) = 0 \\ (\Leftrightarrow)\quad &\sum_{m=1}^{k}\{-\phi^{(i-1)}_{k,k-m}/\phi^{(i-1)}_{k,k}\}\underset{\sim}{\rho}(k, i-1-k+m) = \underset{\sim}{\rho}(k, i-1-k)\end{aligned}$$

If we let $\pi^{(i)}_{k,m} = -\phi^{(i-1)}_{k,k-m}/\phi^{(i-1)}_{k,k}$, then the last equation implies

$$\sum_{m=1}^{k}\pi^{(i)}_{k,m}\underset{\sim}{\rho}(k, i-1-k+m) = \underset{\sim}{\rho}(k, i-1-k)$$

which is equivalent to

$$B(k, i)\tilde{\underset{\sim}{\pi}}(k, i) = \underset{\sim}{r}(k, i)$$

Thus, the nonsingularity of $B(k, i)$ implies that

$$\pi_{k,j}^{(i)} = -\phi_{k,k-j}^{(i-1)} / \phi_{k,k}^{(i-1)}$$
$$j = 1, \cdots, k$$

Finally we are going to derive the relation between the λ's. The (1, 1) element of $B(k+1, i)^{-1}$ is

$$\begin{aligned}\{B(k+1, i)^{-1}\}_{1,1} &= (B^{1,1})_{1,1} \\ &= \{B(k, i)^{-1}\}_{1,1} + \{\tilde{\underline{\pi}}(k, i)\tilde{\underline{\phi}}(k, i)' / \lambda(k, i)\}_{1,1} \\ &= \{B(k, i)^{-1}\}_{1,1} + \pi_{k,k}^{(1)}\phi_{k,k}^{(1)} / \lambda(k, i) \\ &= 1 / \lambda(k, i)\end{aligned}$$

where the last equality holds because the inverse of a persymmetric matrix is persymmetric [10]. If we replace k with $k+1$, we obtain that

$$\{B(k+1, i)^{-1}\}_{1,1} + \pi_{k+1,k+1}^{(i)}\phi_{k+1,k+1}^{(i)} / \lambda(k+1, i) = 1 / \lambda(k+1, i)$$

Thus, we know that

$$\begin{aligned}\lambda(k+1, i) &= \{1 - \pi_{k+1,k+1}^{(i)}\phi_{k+1,k+1}^{(i)}\}\lambda(k, i) \\ &= \{1 - \phi_{k+1,k+1}^{(i)} / \phi_{k+1,k+1}^{(i-1)}\}\lambda(k, i)\end{aligned}$$

It completes the derivation.

Ⅳ. Comments

The algorithm needs less computational time than any other known method such as the Gauss elimination, Trench's inversion technique, and Zohar's algorithm. Actually it is an abridged version of the Trench and the Zohar algorithms obtained by applying them to the mixed ARMA problem. Recently a lot of methods to determine orders of ARMA processes have been proposed by utilizing the EYW [1], [3], [5], [8], in which the new algorithm will be particullarly useful. When a realization of an ARMA process is given, we may estimate the AR parameters by replacing the ACRF with its estimate. The estimates are called the EYW estimates. They are known to be consistent and asymptotically normal. To obtain the MA parameters $\theta_1, \cdots, \theta_q$ and the white noise variance σ^2 when the first $q+1$ autovariances $\sigma(0), \cdots, \sigma(q)$ are given, it is necessary to find some equations connecting the parameters and the ACVF. Such equations and an algorithm to solve them for the MA parameters and the variance using a special property of Toeplitsz matrices are given in [2]. The equations may be solved by the Newton-Rahpson method, which has been already applied to pure MA cases [7].

◈ *REFRENCES* ◈

1. Beguin, J. M., Gourieroux, C. and Monfort, A., "Identification of a Mixed Autoregressive Moving-Average Process: The Corner Method," in *Times Series*, O. D. Anderson, Ed., Amsterdam: North-Holland, 1980, pp. 423~436.
2. Choi, B. S., "An Algorithm for Solving the Extended Yule-Walker Equations of an Autoregressive Moving-Average Time Series," *IEEE Transactions on Information Theory*, IT-32, 1986, pp. 417~419.
3. Chow, J. C., "On Estimating the Orders of an Autoregressive Moving-Average Process with Uncertain Observations," *IEEE Transactions on Automatic Control*, AC-17, 1972, pp. 707~709.
4. Faddeeva, V. N., *Computational Methods of Linear Algebra*, New York: Dover, 1959.
5. Glasbey, C. A., "A Generalization of Partial Autocorrelations Useful in Identifying ARMA Models," *Technometrics*, 24, 1982, pp. 223~228.
6. Trench, W. F., "An Algorithm for the Inversion of Finite Hankel Matrices," *SIAM Journal of Numerical Analysis*, 13, 1965, pp. 1102~1107.
7. Wilson G., "Factorization of the Covariance Generating Function of a Pure Moving Average Process," *SIAM Journal of Numerical Analysis*, 6, 1974, pp. 272~276.
8. Woodward, W. A. and Gray, H. L., "On the Relationship between the S Array and the Box-Jenkins Method of ARMA Model Identification," *Journal of the American Statistical Society*, 76, 1981, pp. 579~587.
9. Zohar, S., "Toeplitz Matrix Inversion: The Algorithm of W. F. Trench," *Journal of Association for Computing Machinery*, Mach, 16, 1969, pp. 592~601.
10. ________, "The Solution of a Toeplitz Set of Linear Equations," *Journal of Association for Computing Machinery*, Mach, 21, 1974, pp. 272~276.
11. ________, "Fortran Subroutines for the Solution of Toeplitz Sets of Linear Equations," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-27, 1979, pp. 656~658.