A Fast Algorithm for Estimating Parameters of a Multivariate Autoregressive Moving Average Processes

Emilie EPEKA MBAMBE1, Angèle YULE SOTAZO1, Jacques KISETA SABITI*1,2,3, & Roger AKUMOSO LIENDI1,3

1Faculty of Economics and Management, Bel Campus Technological University of Kinshasa
2Commercial and Financial Sciences Section of Higher Institute of Commerce (ISC-Kinshasa)
3Statistics Section of Higher Statistical Institute (ISS-Kinshasa)
Democratic Republic of Congo

*Corresponding author: jacquessabiti38@gmail.com

Doi: https://doi.org/10.47285/isr.v2i2.103


Research Article

Abstract

We propose in this paper a fast and iterative algorithm for estimating the parameters of a Gaussian vector autoregressive-moving average (VARMA) model. This algorithm is a multivariate generalization of that suggested by Sabiti (1996) for estimating the parameters of a univariate ARMA(p,q) process. It is proposed, mainly for providing initial estimators for the iterative maximization of a log-likelihood function. Comparisons about the number of computations in terms of multiplication operations are made with a method that uses gradients to locate a maximum of the likelihood function and the fast method suggested by Spliid (1983).

Keywords: Fast estimation, vector models, multivariate time series, vector auto regression, linear regression, ordinary least squares method

1. Introduction

This article proposes a new procedure for ordinary least squares (OLS) estimation of vector autoregressive moving average models. Consider a k-dimensional time series \( \{Y(t)\} \) where series \( Y(t) = \{Y_1(t), Y_2(t), ..., Y_k(t)\}' \) is generated by a mixed ARMA model of order \((p, q)\) of the form

\[
\varphi(B)Y(t) = \vartheta(B)e(t) \quad t = 1,2, ..., N
\]  

(1)

where \( N \) is the sample size, \( B \) is the backshift operator such that \( BY(t) = Y(t - 1) \) and

\[
\varphi(B) = I - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_p B^p
\]  

(2a)

\[
\vartheta(B) = I - \vartheta_1 B - \vartheta_2 B^2 - \cdots - \vartheta_q B^q
\]  

(2b)

are \( k \times k \) matrix polynomials and \( \varphi_j (j = 1,2, ..., p) \) and \( \vartheta_j (j = 1,2, ..., q) \) are respectively the autoregressive and moving average matrices of parameters, where we assume that the orders \((p, q)\) are known. The \( I \) is the \( k \times k \) identity matrix and the \( e(t) = \{e_1(t), e_2(t), ..., e_k(t)\}' \) are independent and identically distributed vector random variables with mean zero and matrix \( \Omega \).
We assume that the process (1) is stationary if the roots of determinantal equation \( |\varphi(B)| = 0 \) are all outside of the unit circle, and is invertible if the roots of the determinantal equation \( |\varphi(B)| = 0 \) are all outside circle. We also assume that the equation (1) satisfies the conditions derived by Hannan (1969) for the model to be identified. The unknown matrices of parameters are arranged in a single matrix \( \beta = (\phi_1, \phi_2, ..., \phi_p, \theta_1, \theta_2, ..., \theta_q)' \) of order \((kp + kq) \times k\).

A stationary and invertible VARMA \((p, q)\) process (1) can be expressed as an infinite autoregressive process

\[
\pi(B)Y(t) = \varepsilon(t), \quad \pi(B) = \varphi(B)/\psi(B), \quad \Pi(B) = I - \sum_{j=1}^{\infty} \Pi B^j \quad (3)
\]

or as an infinite moving average process

\[
Y(t) = \psi(B)\varepsilon(t), \quad \psi(B) = \phi(B)/\psi(B), \quad \Pi(B) = I - \sum_{j=1}^{\infty} \psi B^j . \quad (4)
\]

The estimation of the vector parameters \( \beta \) has been treated by several authors from a theoretical and a computational point of view. Some authors have proposed the OLS method as Spliid (1983) and Sabiti (1996, 1997). Since the maximum likelihood (ML) estimators are asymptotically efficient, many authors have preferred the use of the approximate maximum likelihood (AML) estimation and the exact maximum likelihood (EML) estimation for obtaining the ML estimators. Typically, such approximations are aimed at preserving the asymptotic properties of the estimators, namely consistency, efficiency and asymptotic normality while reducing the computational complexity. Another approach is the generalized least squares (GLS) method, which is often considered as a ML estimation method when suitable assumptions about the structure of the model are made as in Söderström (1974) and Poskitt and Salau (1995).


It is known that the computation of the maximum likelihood estimations is complicated and requires extra computations and iterative procedures such as the Newton-Raphson algorithm or the Marquardt’s algorithm which is based on the calculation of the gradient vector and the Hessian matrix of the log likelihood function. A method that uses gradients to locate a maximum of the log likelihood function will generally need more iterations than parameters and at each iteration, gradients have to be computed in as many directions as parameters. More computations are needed in the estimation procedure where to obtain the optimum, the algorithm is: compute new values of the log likelihood function and update the Hessian matrix to obtain new estimators of the parameters.

To avoid these computations as well as their complexity, several authors have presented fast methods for estimating the vector \( \beta \). Several of these approximated methods have been proposed for providing initial estimators for iterative maximization of a log likelihood function. Spliid (1983) has presented a fast method based on the OLS principle for estimating parameters of multivariate ARMAX models. Sabiti (1996) has proposed a fast algorithm for estimating
parameters of univariate ARMA process that is a modified Mayne and Firoozan (1982) method in the univariate case which uses three least squares regressions in a version which only uses two least squares regressions.

For showing that the algorithm described in this paper is very comparable to the method of Spliiliid (1983) and requires less computations than those using the maximum likelihood principle, numerical comparisons will be presented to compare the proposed algorithm with these methods.

2. Estimation Procedure

The methodology allowing to the algorithm of Sabiti (1996) consists in observing that a multivariate ARMA process (1) can also be written under the form

\[ \phi(B)[\Pi(B)Y(t)] = \theta(B)[\Pi(B)e(t)] \]  

(5)

where the series \( \Pi(B) \) is defined in (3) and the \( \Pi_j \) are \( k \times k \) matrices of coefficients. The relation (5) can also be written as

\[ \phi(B)\tilde{Y}(t) = \theta(B)\tilde{e}(t) \]  

(6)

where \( \tilde{Y}(t) \) and \( \tilde{e}(t) \) are the filtered sequences given respectively by

\[ \tilde{Y}(t) = Y(t) - \sum_{j=1}^{\infty} \Pi_j Y(t-j) = \left[ \Phi(B) / \Theta(B) \right] Y(t) \]  

(7a)

\[ \tilde{e}(t) = e(t) - \sum_{j=1}^{\infty} \Pi_j e(t-j) = \left[ \Phi(B) / \Theta(B) \right] e(t) \]  

(7b)

In the sequel, we will consider the following approximations of (7a) and (7b) as

\[ \tilde{Y}(t) = Y(t) - \sum_{j=1}^{h} \Pi_j Y(t-j) \]  

(8a)

\[ \tilde{e}(t) = e(t) - \sum_{j=1}^{h} \Pi_j e(t-j) \]  

(8b)

where \( h \) is an approximated order of the autoregression (3) and the approximation error can be negligible by taking the order \( h \) sufficiently large. This order \( h \) can also be obtained as in Shittu and Asemota (2009) by using the Akaike information criterion or Schwarz information criterion.

To initiate the algorithm, we denote \( Y \) and \( \tilde{Y} \) the \( N \times k \) matrices of observed time series and the filtered sequences respectively by

\[ Y = \begin{bmatrix} Y_1(1) & Y_2(1) & \cdots & Y_k(1) \\ \vdots & \vdots & \ddots & \vdots \\ Y_1(N) & Y_2(N) & \cdots & Y_k(N) \end{bmatrix}, \quad \tilde{Y} = \begin{bmatrix} \tilde{Y}_1(1) & \tilde{Y}_2(1) & \cdots & \tilde{Y}_k(1) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{Y}_1(N) & \tilde{Y}_2(N) & \cdots & \tilde{Y}_k(N) \end{bmatrix}. \]  

(9)

Similarly, we define the matrix \( e \) and the matrix of (unknown) filtered sequences \( \tilde{e} \) respectively as

\[ e = \begin{bmatrix} e_1(1) & e_2(1) & \cdots & e_k(1) \\ \vdots & \vdots & \ddots & \vdots \\ e_1(N) & e_2(N) & \cdots & e_k(N) \end{bmatrix}, \quad \tilde{e} = \begin{bmatrix} \tilde{e}_1(1) & \tilde{e}_2(1) & \cdots & \tilde{e}_k(1) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{e}_1(N) & \tilde{e}_2(N) & \cdots & \tilde{e}_k(N) \end{bmatrix}. \]  

(10)

Finally we define the following matrices of lagged data:

\[ Z = (B^1Y, B^2Y, \ldots, B^pY), \quad \tilde{Z} = (B^1\tilde{Y}, B^2\tilde{Y}, \ldots, B^p\tilde{Y}) \]  

(11a)

\[ \tilde{e} = (B^1e, B^2e, \ldots, B^pe), \quad \tilde{X} = (Z, \tilde{e}) \]  

(11b)

where for example \( B^s = \{ Y_i(t-s) \} \) for \( i = 1, \ldots, k \) and \( t = 1, \ldots, N \). The matrix of autoregressive coefficients are also arranged to one single matrix of order \( h \times k \) as

\[ \Pi = (\pi_{11}, \pi_{12}, \ldots, \pi_{1p}, \ldots, \pi_{k1}, \ldots, \pi_{kp}) \]  

(12)

The algorithm proposed by Sabiti (1996) is based on the ordinary least squares (OLS) principle and only uses two regressions. No initial estimators of parameters are needed as for the Mayne and Firoozan (1982) method. The proposed algorithm can now be described by the following stages.
where the stage 0 is an initialization. Afterwards the algorithm cycles through stages 2-3 until convergence.

**Stage 0:** Determine the autoregressive coefficients vector $\hat{\Pi}$ from

$$\hat{\Pi} = (Z'Z)^{-1}Z'Y$$  \hspace{1cm} (13)

which is obtained by fitting the autoregression $y = Z\Pi + \epsilon$. The order $h$ of this autoregression may be selected by minimizing the information criterion (Quinn, 1980 and Schwarz, 1977)

$$\text{BIC}(h) = \ln(\hat{\Omega}) + \left(\frac{1}{2}k^2 \log N\right)/N$$  \hspace{1cm} (14)

**Stage 1:** Determine the residual sequence $\hat{\zeta}(t)$ and the filtered sequences $\bar{Y}(t)$ and $\bar{\zeta}(t)$ respectively from

$$\hat{\zeta}(t) = Y(t) - \sum_{j=1}^{h} \hat{\Pi}_j Y(t - j)$$  \hspace{1cm} (15a)

$$\bar{Y}(t)^{(0)} = Y(t) - \sum_{j=1}^{h} \hat{\Pi}_j Y(t - j)$$  \hspace{1cm} (15b)

$$\bar{\zeta}(t)^{(0)} = \hat{\zeta}(t) - \sum_{j=1}^{h} \hat{\Pi}_j \hat{\zeta}(t - j)$$  \hspace{1cm} (15c)

Put $\beta^{(0)} = 0$ and $J = 0$ where $J$ is the iteration counter and proceed with stage 3.

**Stage 2:** Generate the new filtered sequences $\bar{Y}(t)^{(J)}$ and $\bar{\zeta}(t)^{(J)}$ from

$$\bar{Y}(t)^{(J)} = \left(\hat{\Phi}^{(J)}(B)\right)^{-1}\left(\hat{\Phi}^{(J)}(B)\right)\bar{Y}(t)^{(J-1)} \bar{\zeta}(t)^{(J)}$$  \hspace{1cm} (16b)

$$\bar{\zeta}(t)^{(J)} = \left(\hat{\Phi}^{(J)}(B)\right)^{-1}\left(\hat{\Phi}^{(J)}(B)\right)\bar{\zeta}(t)^{(J-1)}$$  \hspace{1cm} (16b)

**Stage 3:** Create the matrices $\bar{Z}$ and $\bar{\varepsilon}$ and construct the matrix $\bar{X}^{(J)}$. Obtain the new estimator of $\beta$ from

$$\beta^{(J+1)} = (\bar{X}^{(J)}\bar{X}^{(J)})^{-1}\bar{X}^{(J)}\bar{Y}^{(J)}$$  \hspace{1cm} (17)

If the maximum of the following convergence criterion

$$\frac{|\beta^{(J+1)} - \hat{\beta}^{(J)}|}{\tau + |\hat{\beta}^{(J)}|} > \lambda \quad \text{for} \quad i = 1, \ldots, (pk + qk)k$$  \hspace{1cm} (18)

with $0 < \tau$ and $\lambda << 1$ is not satisfied where $\tau$ and $\lambda$ are two positive constants sufficiently small, then increase $J$ by 1 and repeat stages 2 through 3. Else $\hat{\beta} = \beta^{(J+1)}$ and stop the iterative procedure.

Note that the convergence can be checked by computing this relative change in the estimator of each parameter in successive iterations. If each of the relative changes is less than a preselected threshold value, convergence may be assumed.

**3. Computational Speed**

We indicate the computational advantages of the proposed fast algorithm by comparing it with a method that uses gradients to locate a maximum of the log likelihood function. We also compare our algorithm with the fast method suggested by Spliid (1983). As a measure, we take the number of multiplications used by each method. We will only consider the number of multiplication operations required in the iterative procedure for these methods. Numerical comparisons between these methods are also proposed. We first give the computation operations required by the maximum likelihood method following an indication of Spliid (1983). After we will give the number of multiplication operations required by our algorithm. The number of operations for the Spliid (1983) method will be evaluated as for the proposed fast algorithm.
As we said, a method that uses gradients to locate a maximum of the likelihood functions needs more iterations than parameters and at each iteration gradients have to be computed in as many directions as parameters. One evaluation of the residuals
\[ \hat{\epsilon}^{(j)} = Y(t) - \sum_{j=1}^{p} \hat{a}_j^{(j)} Y(t-j) + \sum_{j=1}^{q} \hat{b}_j^{(j)} \hat{\epsilon}^{(j)}(t-j) \] (19)
requires \( N(p + q)k^2 \) multiplications. The number of multiplications needed to estimate the second derivative of the log likelihood with respect to the parameter estimators, that is the Hessian matrix \( \partial^2 L(\hat{\beta}) / (\partial \hat{\beta} \partial \hat{\beta}) \) where \( L(.) \) is that log-likelihood is \( N(p + q)k^2 \times ((p + q)k^2)^2 \) multiplications.

Note that we have neglected certain operations used for example in the evaluation of the Hessian matrix at several points where each value of \( L(.) \) requires computation of new residuals. Since extra computations are needed for the ML method, it is difficult to obtain exactly the total number of multiplications required by this method and we will give approximately this number. We assume that all the methods will converge after a same number of iterations.

For the present algorithm, the evaluation of the filtered sequences \( \hat{Y}(t) \) and \( \hat{\epsilon}(t) \) in (15a–c) requires \( 2Nh k^2 \) multiplications. In stage 3, the determination of \( \hat{\beta} = (X'X)^{-1}X'\hat{Y} \), the product \( X'X \) and its inversion require respectively \( (p + q)(p + q + 1)k^2N/2 \) and \( ((p + q)k)^2/2 \) multiplications. The products \( X'Y \) and \( (X'X)^{-1}X'Y \) require respectively \( (p + q)k^2N \) and \( ((p + q)k)^2 \) multiplications. If \( IT \) denotes the number of iterations and \( M \) symbolizes multiplications, one concludes that the present algorithm and the ML method require approximately
\[ M_{\text{Alg}or} = IT \left[ \frac{(p+q)(p+q+1)k^2N}{2} + \frac{((p+q)k)^3}{2} + (p + q)k^2N + ((p + q)k)^2 + 2(p + q)Nk^2 \right] \]
and
\[ M_{\text{ML}} = IT \left[ N(p + q)k^2 \times ((p + q)k^2)^2 + (p + q)k^2N + Q \right] \]
multiplications respectively with \( N \gg (p + q)^2 \) and \( Q \) represents the first four terms of \( M_{\text{Alg}or} \) which are needed to obtain an increase to be added to the initial estimate. By the same procedure of the evaluation of the number of operations for the present algorithm, the Spliid (1983) method will require
\[ M_{\text{SPL}} = IT \left[ \frac{(p+q)(p+q+1)k^2N}{2} + \frac{((p+q)k)^3}{2} + (p + q)k^2N + ((p + q)k)^2 + N(p + q)k^2 \right] \]
multiplications.

In the following table, we present the number of operations required by each method by varying the dimension of \( Y(t) \) and the orders of the polynomials \( \Phi(B) \) and \( \Theta(B) \). The maximum likelihood, the Spliid (1983) method and the proposed algorithm are referred to as ML, SPL and Algol respectively.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>( p )</th>
<th>( q )</th>
<th>ML</th>
<th>SPL</th>
<th>Algol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.520</td>
<td>0.076</td>
<td>0.084</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>8.500</td>
<td>6.200</td>
<td>6.200</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>1906.700</td>
<td>125.200</td>
<td>125.400</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>8489.900</td>
<td>546.700</td>
<td>546.900</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>64.000</td>
<td>4.700</td>
<td>4.700</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>4096.200</td>
<td>266.000</td>
<td>266.200</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>64000.400</td>
<td>4061.000</td>
<td>4061.400</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>64.000</td>
<td>4.700</td>
<td>4.700</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>8000.200</td>
<td>515.500</td>
<td>515.700</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>32.800</td>
<td>0.400</td>
<td>0.500</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>5452.000</td>
<td>45.900</td>
<td>46.100</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>54339.300</td>
<td>4309.300</td>
<td>4310.100</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>32768.300</td>
<td>266.400</td>
<td>266.700</td>
<td></td>
</tr>
</tbody>
</table>
From this table, we can observe that the proposed fast algorithm and the method of Spliid (1983) are faster than maximum likelihood procedures. For different models and different dimensions, these approximated methods are very fast. When we compare the present algorithm with the Spliid (1983) method, we find that these two methods are very comparable.

For an ARMA(1,1) with \( k = 2 \), the approximated methods are more than 6 times faster than the ML method. For an ARMA(50,1) with \( k = 2 \), the approximated methods are more than 15 times faster than the ML method. The interesting comparison between these approximated methods and the ML methods appears when we increase the dimensions. Thus, for an ARMA(1,1) with \( k = 4 \), the approximated methods become more than 75 times faster than the ML method and more than 123 times faster for an ARMA(10,10) with \( k = 4 \).

4. Some comments about the present fast algorithm

The algorithm proposed by Sabiti (1996) uses two least squares regressions instead of the three regressions procedures of Mayne and Firoozan (1982) described in the Sabiti (1996) article. For that, we have suggested to write the ARMA model (1) in the form (5) where \( Y(t) \) and \( \epsilon(t) \) are multiplied by the series \( x(B) \). When the autoregression (3) of order \( h \) is fitted at the first stage, we can obtain directly the filtered sequences \( \bar{Y}(t) \) and \( \bar{\epsilon}(t) \) as in (15a-c) and to continue with the final regression at the third stage. By this way, we have avoided to fit another regression in the second stage as for the method of Mayne and Firoozan (1982).

For the Mayne and Firoozan (1982) method, the estimators \( \hat{\beta}_k \) of the autoregressive coefficients are used to obtain the residuals \( \hat{\epsilon}(t) \) as in (15a). The lagged values of these residuals are used together with the lagged values of \( Y(t) \) to form the regression of the form

\[
Y(t) = \sum_{j=1}^{p} \phi_j Y(t-j) - \sum_{j=1}^{q} \phi_j \hat{\epsilon}(t-j) + e(t) \tag{20}
\]

which will be fitted by the ordinary least squares (OLS) method. Thus the estimators of \( \phi_j \) will be used to filter the sequences \( \bar{Y}(t) \) and \( \bar{\epsilon}(t) \) and to continue with a third regression. Although our filtered sequences (16a-b) contain a ratio of two polynomials compared to those of Mayne and Firoozan (1982), the form (5) have allowed us to avoid the fitting of the regression (20) and to obtain the filtered sequences \( \bar{Y}(t) \) and \( \bar{\epsilon}(t) \) directly after fitting an autoregression at the first stage. Except the number of regressions fitted by our fast algorithm and that of Mayne and Firoozan (1982), another difference comes from the filtered sequences used by each procedure. For the method of Mayne and Firoozan (1982), the sequences \( \bar{Y}(t) \) and \( \bar{\epsilon}(t) \) are generated by a moving average process. Whereas for our algorithm, these filtered sequences are generated by an ARMA process.

5. Conclusion

We have described in this paper a fast and iterative algorithm in order to compute the least square estimators of multivariate ARMA processes. The present algorithm is a multivariate generalization of a method proposed by Sabiti (1996) to estimate the parameters of a univariate ARMA process. This algorithm has been suggested mainly for providing initial estimators for iterative maximization of a log likelihood function. The performance of the present algorithm has been compared in terms of number multiplications with the Spliid (1983) and the maximum likelihood methods. We conclude that the present algorithm is faster than the methods that use gradients to locate a maximum of the likelihood function.

Conflict of Interest: The authors declare no conflict of interest.

REFERENCES


© 2021 by the authors. Licensee Scienetific Research Initiative, Michigan, USA. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).