Algorithm AS 311

The Exact Likelihood Function of a Vector Autoregressive Moving Average Model

By José Alberto Mauricio

Universidad Complutense de Madrid

[Received July 1993. Final revision October 1996]

Keywords: Invertibility; Maximum likelihood; Stationarity; Vector autoregressive moving average model

Language

Fortran 77

Description and Purpose

This algorithm has the same purpose as algorithm AS 242 of Shea (1989), namely to compute the exact likelihood function of a vector autoregressive moving average (ARMA) model. It turns out to be faster in many relevant cases and not appreciably slower in any. In addition to the advantages offered by the algorithm of Shea (1989), including the calculation of an appropriate set of residuals, it also permits the automatic detection of non-invertible models as a by-product. The code presented here combines improved versions of the algorithms due to Ljung and Box (1979) and Hall and Nicholls (1980) with a corrected version of an algorithm due to Kohn and Ansley (1982). The resulting procedure puts together a set of useful features which can only be found separately in other existing methods.

Theory and Method

Let $\mathbf{w}_t$ be an m-dimensional vector-valued time series. It is assumed that $\mathbf{w}_t$ follows the vector ARMA(p, q) model

$$\mathbf{\Phi}(B)\tilde{\mathbf{w}}_t = \mathbf{\Theta}(B)\mathbf{a}_t, \tag{1}$$

where

$$\mathbf{\Phi}(B) = \mathbf{I} - \mathbf{\Phi}_1 B - \ldots - \mathbf{\Phi}_p B^p,$$

$$\mathbf{\Theta}(B) = \mathbf{I} - \mathbf{\Theta}_1 B - \ldots - \mathbf{\Theta}_q B^q,$$

B is the backshift operator, $\tilde{\mathbf{w}}_t = \mathbf{w}_t - \boldsymbol{\mu}$, $\mathbf{\Phi}_i$ $(i = 1, 2, \ldots, p)$, $\mathbf{\Theta}_i$ $(i = 1, 2, \ldots, q)$ and $\boldsymbol{\mu}$ are $m \times m$, $m \times m$ and $m \times 1$ parameter matrices respectively and the $\mathbf{a}_t$s are $m \times 1$ random vectors identically and independently distributed as $N(\mathbf{0}, \sigma^2\mathbf{Q})$, with $\sigma^2 > 0$ and $\mathbf{Q}$ $(m \times m)$ symmetric and positive definite. For stationarity, it is required that the zeros of $|\mathbf{\Phi}(B)|$ lie outside the unit circle. Likewise, the model will be invertible provided that the zeros of $|\mathbf{\Theta}(B)|$ lie outside the unit circle.

Consider a sample of size n and define $\widetilde{\mathbf{w}} = (\widetilde{\mathbf{w}}_1^{\mathrm{T}}, \ldots, \widetilde{\mathbf{w}}_n^{\mathrm{T}})^{\mathrm{T}}$ (mean-corrected observations), $\mathbf{a} = (\mathbf{a}_1^{\mathrm{T}}, \ldots, \mathbf{a}_n^{\mathrm{T}})^{\mathrm{T}}$ (white noise perturbations) and $\mathbf{u}_* = (\widetilde{\mathbf{w}}_{1-p}^{\mathrm{T}}, \ldots, \widetilde{\mathbf{w}}_0^{\mathrm{T}}, \mathbf{a}_{1-q}^{\mathrm{T}}, \ldots, \mathbf{a}_0^{\mathrm{T}})^{\mathrm{T}}$ (unknown presample values). Then equation (1) may be written as

$$\mathbf{D}_{\mathbf{\Phi},n}\widetilde{\mathbf{w}} = \mathbf{D}_{\mathbf{\Theta},n}\mathbf{a} + \mathbf{V}\mathbf{u}_*,$$

where $\mathbf{D}_{\mathbf{\Phi},n}$ and $\mathbf{D}_{\mathbf{\Theta},n}$ are $nm \times nm$ block matrices with identity matrices on the main diagonal and $-\mathbf{\Phi}_k$ and $-\mathbf{\Theta}_k$ respectively down the kth subdiagonal. Further, $\mathbf{V}$ is the $nm \times (p+q)m$ block matrix $\mathbf{V} = (\mathbf{G}_{\mathbf{\Phi},n}, \mathbf{G}_{\mathbf{\Theta},n})$, where $\mathbf{G}_{\mathbf{\Phi},n}$ and $\mathbf{G}_{\mathbf{\Theta},n}$ are the following $nm \times pm$ and $nm \times qm$ block matrices:

$$\mathbf{G}_{\mathbf{\Phi},n} = \begin{pmatrix} \mathbf{\Phi}_p & \mathbf{\Phi}_{p-1} & \ldots & \mathbf{\Phi}_1 \\ \mathbf{0} & \mathbf{\Phi}_p & \ldots & \mathbf{\Phi}_2 \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{\Phi}_p \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \end{pmatrix}, \qquad \mathbf{G}_{\mathbf{\Theta},n} = \begin{pmatrix} -\mathbf{\Theta}_q & -\mathbf{\Theta}_{q-1} & \ldots & -\mathbf{\Theta}_1 \\ \mathbf{0} & -\mathbf{\Theta}_q & \ldots & -\mathbf{\Theta}_2 \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & -\mathbf{\Theta}_q \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \end{pmatrix}.$$

On the basis of the previous definitions, Nicholls and Hall (1979) have shown that the exact log-likelihood of the parameters $\mathbf{\Phi} = (\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_p)$, $\mathbf{\Theta} = (\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_q)$, $\boldsymbol{\mu}$, σ^2 and $\mathbf{Q}$ has the form

$$l(\mathbf{\Phi}, \mathbf{\Theta}, \boldsymbol{\mu}, \sigma^2, \mathbf{Q}|\mathbf{w}) = -\frac{1}{2}\left\{nm \log(2\pi\sigma^2) + n \log|\mathbf{Q}| + \log|\mathbf{\Lambda}^{\mathrm{T}}\mathbf{\Lambda}| + \frac{1}{\sigma^2}S(\mathbf{\Phi}, \mathbf{\Theta}, \boldsymbol{\mu}, \mathbf{Q}|\mathbf{w})\right\}, \tag{2}$$

where, by means of extending and improving the procedures of Ljung and Box (1979) and Hall and Nicholls (1980), it has been shown in Mauricio (1995a) that

$$S(\mathbf{\Phi}, \mathbf{\Theta}, \boldsymbol{\mu}, \mathbf{Q}|\mathbf{w}) = \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\eta} - (\mathbf{M}^{\mathrm{T}}\widetilde{\mathbf{h}})^{\mathrm{T}}(\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M})^{-1}(\mathbf{M}^{\mathrm{T}}\widetilde{\mathbf{h}}) \tag{3}$$

and

$$|\mathbf{\Lambda}^{\mathrm{T}}\mathbf{\Lambda}| = |\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M}|. \tag{4}$$

Subroutine ELF2 computes the expression on the right-hand side of equation (2) using equations (3) and (4). Also, it optionally returns the residual vector, computed from

$$\widehat{\mathbf{a}} = \widehat{\mathbf{a}}_0 - \mathbf{D}_{\mathbf{\Theta},n}^{-1}\begin{pmatrix} \mathbf{M}(\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M})^{-1}\mathbf{M}^{\mathrm{T}}\widetilde{\mathbf{h}} \\ \mathbf{0} \end{pmatrix}. \tag{5}$$

To evaluate equations (3), (4) and (optionally) (5), the following steps (Mauricio, 1995a) are performed within subroutine ELF2.

(a) Compute the Cholesky factor $\mathbf{Q}_1$ of matrix $\mathbf{Q}$ (i.e. $\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_1^{\mathrm{T}}$), its determinant $|\mathbf{Q}| = |\mathbf{Q}_1|^2$ and a matrix $\mathbf{R}$ such that $\mathbf{R}\mathbf{Q}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$ (i.e. $\mathbf{R} = \mathbf{Q}_1^{-1}$).

(b) Evaluate the theoretical autocovariances $\mathbf{\Gamma}(k) = \sigma^{-2}E[\widetilde{\mathbf{w}}_t\widetilde{\mathbf{w}}_{t+k}^{\mathrm{T}}]$ ($k = 0, 1, \ldots, p-1$) and the theoretical cross-covariances $\mathbf{\Gamma}_{\mathbf{wa}}(k) = \sigma^{-2}E[\widetilde{\mathbf{w}}_t\mathbf{a}_{t+k}^{\mathrm{T}}]$ ($k = 0, -1, \ldots, -q+1$).

(c) Compute the symmetric $gm \times gm$ matrix $\mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1^{\mathrm{T}}$, with $g = \max(p, q)$, where $\mathbf{V}_1$ consists of the first gm rows of $\mathbf{V}$ and $\mathbf{\Omega} = \sigma^{-2}E[\mathbf{u}_*\mathbf{u}_*^{\mathrm{T}}]$. Then, calculate its Cholesky factor $\mathbf{M}$ (i.e. $\mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1^{\mathrm{T}} = \mathbf{M}\mathbf{M}^{\mathrm{T}}$). The (i, j)th $m \times m$ block of $\mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1^{\mathrm{T}}$ ($i = 1, 2, \ldots, g$; $j = 1, 2, \ldots, i$) is given by

$$(\mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1^{\mathrm{T}})_{ij} = \sum_{k=0}^{p-i} \mathbf{\Phi}_{p-k}\mathbf{E}_{k+i,j} - \sum_{k=0}^{q-i} \mathbf{\Theta}_{q-k}\mathbf{E}_{k+p+i,j},$$

where, for $j = 1, 2, \ldots, g$, the required $m \times m$ $\mathbf{E}_{ij}$-matrices are given by

$$\mathbf{E}_{ij} = \sum_{k=j-i}^{p-i} \mathbf{\Gamma}(k)\mathbf{\Phi}_{p-k-i+j}^{\mathrm{T}} - \sum_{k=j-i}^{q-i} \mathbf{\Gamma}_{\mathbf{wa}}(-q+p+k)\mathbf{\Theta}_{q-k-i+j}^{\mathrm{T}} \qquad (i = 1, 2, \ldots, p),$$

$$\mathbf{E}_{ij} = \sum_{k=p+j-i}^{2p-i} \mathbf{\Gamma}_{\mathbf{wa}}(-q+p-k)^{\mathrm{T}}\mathbf{\Phi}_{2p-k-i+j}^{\mathrm{T}} - \mathbf{Q}\mathbf{\Theta}_{q+p-i+j}^{\mathrm{T}} \qquad (i = p+1, p+2, \ldots, p+q),$$

with $\mathbf{\Gamma}(k) = \mathbf{\Gamma}(-k)^{\mathrm{T}}$ for $k < 0$, $\mathbf{\Gamma}_{\mathbf{wa}}(k) = \mathbf{0}$ for $k > 0$ and $\mathbf{\Theta}_i = \mathbf{0}$ for $i > q$.

(d) Evaluate the $m \times m$ matrices $\mathbf{\Xi}_k$ ($k = 0, 1, \ldots, n-1$) as follows:

$$\mathbf{\Xi}_k = \sum_{j=1}^{q} \mathbf{\Theta}_j\mathbf{\Xi}_{k-j} \qquad (k = 1, 2, \ldots, n-1),$$

with $\mathbf{\Xi}_0 = \mathbf{I}$ and $\mathbf{\Xi}_k = \mathbf{0}$ for $k < 0$. Then, premultiply every $\mathbf{\Xi}_k$ by (lower triangular) $\mathbf{R}$.

(e) Calculate the $m \times 1$ vectors $\boldsymbol{\eta}_i = \mathbf{R}\widehat{\mathbf{a}}_{0i}$ ($i = 1, 2, \ldots, n$), where

$$\widehat{\mathbf{a}}_{0i} = \widetilde{\mathbf{w}}_i - \sum_{j=1}^{p} \mathbf{\Phi}_j\widetilde{\mathbf{w}}_{i-j} + \sum_{j=1}^{q} \mathbf{\Theta}_j\widehat{\mathbf{a}}_{0,\,i-j} \qquad (i = 1, 2, \ldots, n),$$

where $\widetilde{\mathbf{w}}_i = \mathbf{0}$ for $i < 1$ and $\widehat{\mathbf{a}}_{0i} = \mathbf{0}$ for $i < 1$.

(f) Compute the $m \times 1$ vectors $\widetilde{\mathbf{h}}_i$ ($i = 1, 2, \ldots, g$) as follows:

$$\widetilde{\mathbf{h}}_i = \sum_{j=0}^{n-i} \mathbf{\Xi}_j^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\boldsymbol{\eta}_{i+j} \qquad (i = 1, 2, \ldots, g).$$

Then evaluate the $gm \times 1$ vector $\mathbf{M}^{\mathrm{T}}\widetilde{\mathbf{h}}$, where $\widetilde{\mathbf{h}} = (\widetilde{\mathbf{h}}_1^{\mathrm{T}}, \ldots, \widetilde{\mathbf{h}}_g^{\mathrm{T}})^{\mathrm{T}}$.

(g) Evaluate the symmetric $gm \times gm$ matrix $\mathbf{H}^{\mathrm{T}}\mathbf{H}$. The first block column of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is given by

$$(\mathbf{H}^{\mathrm{T}}\mathbf{H})_{i1} = \sum_{k=0}^{n-i} \mathbf{\Xi}_k^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{\Xi}_{k+i-1} \qquad (i = 1, 2, \ldots, g),$$

and the remaining diagonal and subdiagonal $m \times m$ blocks of matrix $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ are given by

$$(\mathbf{H}^{\mathrm{T}}\mathbf{H})_{ij} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})_{i-1,j-1} - \mathbf{\Xi}_{n-i+1}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{\Xi}_{n-j+1},$$

with $i = 2, 3, \ldots, g$ and $j = 2, 3, \ldots, i$.

(h) Evaluate the symmetric $gm \times gm$ matrix $\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M}$, its Cholesky factor $\mathbf{L}$ (i.e. $\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$) and its determinant $|\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M}| = |\mathbf{L}|^2$, which is, in turn, the determinant (4).

(i) Use forward substitution to solve for $\boldsymbol{\lambda}$ in the triangular system $\mathbf{L}\boldsymbol{\lambda} = \mathbf{M}^{\mathrm{T}}\widetilde{\mathbf{h}}$.

(j) Compute the quadratic form (3) as $S(\mathbf{\Phi}, \mathbf{\Theta}, \boldsymbol{\mu}, \mathbf{Q}|\mathbf{w}) = \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\eta} - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\lambda}$, where $\boldsymbol{\eta} = (\boldsymbol{\eta}_1^{\mathrm{T}}, \ldots, \boldsymbol{\eta}_n^{\mathrm{T}})^{\mathrm{T}}$.

(k) Use backward substitution to solve for $\mathbf{c}$ in the triangular system $\mathbf{L}^{\mathrm{T}}\mathbf{c} = \boldsymbol{\lambda}$, calculate the $gm \times 1$ vector $\mathbf{d} = \mathbf{M}\mathbf{c}$ and compute $\widehat{\mathbf{a}}_i = \widehat{\mathbf{a}}_{0i} - \mathbf{r}_i$ $(i = 1, 2, \ldots, n)$, with the $m \times 1$ vector $\mathbf{r}_i$ given by

$$\mathbf{r}_i = \sum_{j=1}^{i} \mathbf{\Xi}_{i-j}\mathbf{d}_j \qquad (i = 1, 2, \ldots, n).$$

where $\mathbf{d}_j$ is the jth $m \times 1$ block of $\mathbf{d}$ $(j = 1, 2, \ldots, g)$ and $\mathbf{d}_j = \mathbf{0}$ for $j > g$.

Subroutines CGAMMA, CXI and CRES are called from within subroutine ELF2 to carry out steps (b), (d) and (k) (optional) respectively. Some matrix computations are performed by subroutines CHOLDC (which is called from within ELF2 to carry out steps (a), (c) and (h)), CHOLFR (which is called from within ELF2 to carry out steps (a) and (i)) and CHOLBK (which is called from within CRES to carry out step (k)). Subroutine CGAMMA implements a corrected version (Mauricio, 1995b) of the algorithm of Kohn and Ansley (1982), to evaluate the matrices $\mathbf{\Gamma}(k)$ $(k = 0, 1, \ldots, p-1)$ and $\mathbf{\Gamma}_{\mathbf{wa}}(k)$ $(k = 0, -1, \ldots, -q+1)$ in a computationally more efficient manner than the routine coded in Shea (1989). The procedure in subroutine CGAMMA solves the system of linear equations

$$\mathbf{\Gamma}(0) - \sum_{i=1}^{p} \mathbf{\Phi}_i\mathbf{\Gamma}(0)\mathbf{\Phi}_i^{\mathrm{T}} - \sum_{i=1}^{p-1}\sum_{j=1}^{p-i} \left\{\mathbf{\Phi}_{i+j}\mathbf{\Gamma}(i)\mathbf{\Phi}_j^{\mathrm{T}} + \mathbf{\Phi}_j\mathbf{\Gamma}(i)^{\mathrm{T}}\mathbf{\Phi}_{i+j}^{\mathrm{T}}\right\} = \mathbf{W}_0, \tag{6a}$$

$$\mathbf{\Gamma}(k) - \sum_{i=1}^{k-1} \mathbf{\Gamma}(i)\mathbf{\Phi}_{k-i}^{\mathrm{T}} - \sum_{i=0}^{p-k} \mathbf{\Gamma}(i)^{\mathrm{T}}\mathbf{\Phi}_{k+i}^{\mathrm{T}} = \mathbf{W}_k \qquad (k = 1, 2, \ldots, p-1), \tag{6b}$$

for $\widetilde{\mathbf{\Gamma}}(0), \mathbf{\Gamma}(1), \ldots, \mathbf{\Gamma}(p-1)$, where $\widetilde{\mathbf{\Gamma}}(0)$ is the diagonal and upper triangle of $\mathbf{\Gamma}(0)$,

$$\mathbf{W}_0 = \mathbf{Q} - (\mathbf{A} + \mathbf{A}^{\mathrm{T}}) + \sum_{j=1}^{q} \mathbf{\Theta}_j \mathbf{Q} \mathbf{\Theta}_j^{\mathrm{T}},$$

$$\mathbf{A} = \sum_{i=1}^{p} \sum_{j=i}^{q} \mathbf{\Phi}_i \mathbf{\Gamma}_{\mathbf{wa}}(i-j) \mathbf{\Theta}_j^{\mathrm{T}},$$

$$\mathbf{W}_k = -\sum_{j=k}^{q} \mathbf{\Gamma}_{\mathbf{wa}}(k-j) \mathbf{\Theta}_j^{\mathrm{T}} \qquad (k = 1, 2, \ldots, p-1),$$

$$\mathbf{\Gamma}_{\mathbf{wa}}(-k) = -\mathbf{\Theta}_k \mathbf{Q} + \sum_{i=1}^{k} \mathbf{\Phi}_i \mathbf{\Gamma}_{\mathbf{wa}}(i-k) \qquad (k = 1, 2, \ldots, q-1),$$

with $\mathbf{\Phi}_i = \mathbf{0}$ if $i > p$ and $\mathbf{\Gamma}_{\mathbf{wa}}(0) = \mathbf{Q}$ (note that $\mathbf{\Gamma}_{\mathbf{wa}}(k) = \sigma^{-2} E[\tilde{\mathbf{w}}_t \mathbf{a}_{t+k}^{\mathrm{T}}] = \mathbf{0}$ for $k > 0$). The resulting system contains $m(m+1)/2 + m^2(p-1)$ unknowns. Thus, subroutine CGAMMA solves for $m(m-1)/2 + m^2$ fewer unknowns than subroutine COVARS of Shea (1989), which unnecessarily solves for $\mathbf{\Gamma}(0)$ through $\mathbf{\Gamma}(p)$ instead of $\tilde{\mathbf{\Gamma}}(0)$ and $\mathbf{\Gamma}(1)$ through $\mathbf{\Gamma}(p-1)$.

Structure

SUBROUTINE ELF1(M, IM, P, Q, N, W, PHI, THETA, QQ, ISMU, MU, ATF, A, SIGMA2, XITOL, LOGELF, F1, F2, WS, NWS, IWS, NIWS, IFAULT)

Formal parameters

M	Integer	input:	the number of time series, m ($\geqslant 1$)
IM	Integer	input:	on entry, IM ($\geqslant$ M) must specify the leading dimension of the arrays W, PHI, THETA, QQ and A as declared in the user's calling (sub)program
P	Integer	input:	the value of p ($\geqslant 0$)
Q	Integer	input:	the value of q ($\geqslant 0$), but $p = q = 0$ is not allowed
N	Integer	input:	the length of each series, n ($\geqslant 1$)
W	Real array of dimension (IM, N)	input:	on entry, W(I, J) must contain the Ith component of $\mathbf{w}_{\mathrm{J}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., N
PHI	Real array of dimension (IM, P × M + 1)	input:	on entry, PHI(I, (K − 1) × M + J) must contain the (I, J)th element of $\mathbf{\Phi}_{\mathrm{K}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., M, K = 1, 2, . . ., P
THETA	Real array of dimension (IM, Q × M + 1)	input:	on entry, THETA(I, (K − 1) × M + J) must contain the (I, J)th element of $\mathbf{\Theta}_{\mathrm{K}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., M, K = 1, 2, . . ., Q

QQ	Real array of dimension (IM, M)	input/output:	on entry, QQ(I, J) must contain the (I, J)th element of $\mathbf{Q}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., I (i.e. the lower triangle of $\mathbf{Q}$); on exit, if IFAULT = 0 or IFAULT $\geqslant$ 9 then the strict upper triangle of QQ is set equal to its lower triangle
ISMU	Logical	input:	set equal to .TRUE. if $\boldsymbol{\mu} \neq \mathbf{0}$ and .FALSE. if $\boldsymbol{\mu} = \mathbf{0}$
MU	Real array of dimension M	input:	if ISMU is set equal to .TRUE. then MU(I) must contain the Ith component of $\boldsymbol{\mu}$ for I = 1, 2, . . ., M; if ISMU is set equal to .FALSE. then MU is not used
ATF	Logical	input:	set equal to .TRUE. if computation of the residual vector is required and set to .FALSE. otherwise
A	Real array of dimension (IM, N)	output:	if ATF is set equal to .TRUE. then, on successful exit, A(I, J) contains the Ith component of $\hat{\mathbf{a}}_{\mathrm{J}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., N; if ATF is set equal to .FALSE. then, if IFAULT = 0 or IFAULT = 13, A(I, J) contains the Ith component of $\boldsymbol{\eta}_{\mathrm{J}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., N
SIGMA2	Real	input:	the value of σ^2
XITOL	Real	input:	convergence tolerance for the Ξ_ks; on entry, it must be set to any negative number if an exact evaluation of the log-likelihood is desired; if an approximate evaluation is desired or if $q = 0$ then it should be set to a small positive number
LOGELF	Real	output:	on successful exit, contains the value of the log-likelihood function (2)
F1	Real	output:	on successful exit, contains the value of the quadratic form (3)
F2	Real	output:	on successful exit, contains the value of $\lvert\mathbf{Q}\rvert^n$ times the determinant (4)
WS	Real array of dimension at least NWS	workspace:	
NWS	Integer	input:	the dimension of the array WS as declared in the user's calling (sub)program: NWS $\geqslant$ M $\times$ M $\times$ (3 + 3 $\times$ G $\times$ G + (P + Q) $\times$ G + B3) + B1 $\times$ B1 + B2 + M, where G = max(P, Q), B1 = M $\times$ (M + 1)/2 + M $\times$ M $\times$ (P − 1) if P > 0 and B1 = 1 otherwise, B2 = max(B1, G $\times$ M) and B3 = max(N, Q)

IWS	Integer array of dimension at least NIWS	workspace:	
NIWS	Integer	input:	the dimension of the array IWS as declared in the user's calling (sub)program: NIWS $\geqslant$ B1, where B1 is as for NWS
IFAULT	Integer	output:	a fault indicator: $= 1$ if M < 1; $= 2$ if N < 1; $= 3$ if P < 0; $= 4$ if Q < 0; $= 5$ if P $= 0$ and Q $= 0$; $= 6$ if IM $<$ M; $= 7$ if NWS is too small; $= 8$ if NIWS is too small; $= 9$ if QQ is not positive definite; $= 10$ if equations (6) for calculating the $\mathbf{\Gamma}(k)$s could not be solved (this indicates that the autoregressive parameters are very close to the boundary of the stationarity region); $= 11$ if the matrix $\mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1^{\mathrm{T}}$ is not positive definite (this indicates that the model is not stationary); $= 12$ if the $\mathbf{\Xi}_k$s turn out to be explosive (i.e. if any given norm of $\mathbf{\Xi}_k$ increases with k; this indicates that the model is not invertible); $= 13$ if the matrix $\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M}$ is not positive definite; $= 0$ otherwise (on a successful exit)

Subroutine ELF1 checks for errors in the input parameters, sets up workspace arrays, calls subroutine MACHEP to compute the *machine epsilon* (used by subroutine CHOLDC) and then calls subroutine ELF2 to evaluate the exact log-likelihood function. A description of the formal parameters of ELF2 is given next.

SUBROUTINE ELF2(M, IM, P, Q, N, G, W, PHI, THETA, QQ, ISMU, MU, ATF, A, SIGMA2, XITOL, LOGELF, F1, F2, EPS, BIG1, BIG2, BIG3, Q1, Q1INV, MTMP4, VTMP1, VTMP2, MATPHI, MTMP0, MTMP2, MTMP3, MTMP1, GAMXI, INDX, IFAULT)

Formal parameters

M, IM, P, Q, N		input: as for ELF1
G	Integer	input: max(P, Q), set by ELF1
W, PHI, THETA		input: as for ELF1

QQ		input/output:	as for ELF1
ISMU, MU, ATF		input:	as for ELF1
A		output:	as for ELF1; A is given values in ELF2
SIGMA2, XITOL		input:	as for ELF1
LOGELF, F1, F2		output:	as for ELF1; LOGELF, F1 and F2 are given values in ELF2
EPS	Real	input:	the machine epsilon (set within ELF1)
BIG1	Integer	input:	BIG1 $= \mathrm{M} \times (\mathrm{M}+1)/2 + \mathrm{M} \times \mathrm{M} \times (\mathrm{P}-1)$ if $\mathrm{P} > 0$ and BIG1 = 1 otherwise; this value is set by ELF1
BIG2	Integer	input:	max(BIG1, G $\times$ M) (set by ELF1)
BIG3	Integer	input:	max(N, Q) (set by ELF1)
Q1, Q1INV, MTMP4, VTMP1, VTMP2, MATPHI, MTMP0, MTMP2, MTMP3, MTMP1, GAMXI	Real workspace arrays		
INDX	Integer workspace array		
IFAULT	Integer	output:	a fault indicator; on exit from ELF2, IFAULT will have either the value 9, 10, 11, 12, 13 or 0

Subroutine ELF2 implements the steps described in the previous section to calculate the log-likelihood function. One of these steps is the computation of the covariance matrices $\boldsymbol{\Gamma}(k)$ $(k = 0, 1, \ldots, p-1)$ and $\boldsymbol{\Gamma}_{\mathbf{wa}}(k)$ $(k = 0, -1, \ldots, -q+1)$. This is done by subroutine CGAMMA, which implements a corrected version (Mauricio, 1995b) of the method of Kohn and Ansley (1982).

SUBROUTINE CGAMMA(M, IM, P, Q, PHI, THETA, QQ, BIG1, MAT, WZERO, MZERO, GAMWA, RHS, INDX, IFAULT)

Formal parameters

M, IM, P, Q, PHI, THETA, QQ, BIG1 MAT, WZERO, MZERO, INDX	Workspace arrays	input:	as for ELF2
GAMWA	Real workspace array	output:	on successful exit, GAMWA(I, K $\times$ M+ J) contains the (I, J)th element of $\boldsymbol{\Gamma}_{\mathbf{wa}}(-\mathrm{K})$ for I = 1, 2, . . ., M, J = 1, 2, . . ., M, K = 0, 1, . . ., Q $-$ 1

RHS	Real workspace array	output:	on successful exit, RHS(J × (J − 1)/2 + I) contains the (I, J)th element of $\boldsymbol{\Gamma}(0)$ for I = 1, 2, . . ., M, J = I, I + 1, . . ., M and RHS(M × (M + 1)/2 + M × M × (K − 1) + M × (J − 1) + I) contains the (I, J)th element of $\boldsymbol{\Gamma}$(K) for I = 1, 2, . . ., M, J = 1, 2, . . ., M, K = 1, 2, . . ., P − 1
IFAULT	Integer	output:	a fault indicator: = 1 if equations (6) could not be solved (in which case IFAULT is returned from ELF2 as 10); = 0 otherwise (on successful exit)

Subroutine ELF2 also calls subroutine CXI, which computes recursively the matrix sequence Ξ_k ($k = 0, 1, \ldots, n-1$). This is a key step to check for invertibility of the model.

SUBROUTINE CXI(M, IM, N, Q, THETA, XITOL, R, NLIM, XI, MTMP, IFAULT)

Formal parameters

M, IM, N, Q, THETA, XITOL		input:	as for ELF2
R	Real workspace array	input:	on entry, R(I, J) must contain the (I, J)th element of **R** for I = 1, 2, . . ., M, J = 1, 2, . . ., I, as set by ELF2
NLIM	Integer	output:	on successful exit, NLIM is such that $\Xi_k = \mathbf{0}$ for $k >$ NLIM
XI	Real workspace array	output:	on successful exit, XI(I, M × K + J) contains the (I, J)th element of $\mathbf{R}\Xi_k$ for I = 1, 2, . . . , M, J = 1, 2, . . ., M, K = 0, 1 . . ., NLIM
MTMP	Real workspace array		
IFAULT	Integer	output:	a fault indicator: = 1 if the computation of the sequence Ξ_k turns out to be explosive (in which case IFAULT is returned from ELF2 as 12); = 0 otherwise (on successful exit)

Optionally (if ATF was set equal to .TRUE. on entry to ELF1), subroutine ELF2 ends with a call to subroutine CRES, to calculate the residual vector by using some of the computations carried out so far to evaluate the log-likelihood.

SUBROUTINE CRES(M, IM, N, G, NLIM, XI, Q1, MATM, MATL, LAMBDA, RES)

Formal parameters

M, IM, N, G		input:	as for ELF2
NLIM	Integer	input:	on entry, NLIM must be such that $\Xi_k = \mathbf{0}$ for $k >$ NLIM, as set by CXI from within ELF2
XI	Real workspace array	input:	on entry, XI(I, M $\times$ K + J) must contain the (I, J)th element of $\mathbf{R}\Xi_k$ for I = 1, 2, . . ., M, J = 1, 2, . . ., M, K = 0, 1, . . ., NLIM, as set by CXI from within ELF2
Q1	Real workspace array	input:	on entry, Q1(I, J) must contain the (I, J)th element of the Cholesky factor $\mathbf{Q}_1$ of $\mathbf{Q}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., I, as set by ELF2
MATM	Real workspace array	input:	on entry, MATM(I, J) must contain the (I, J)th element of the Cholesky factor $\mathbf{M}$ of $\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1^{\mathrm{T}}$ for I = 1, 2, . . ., G $\times$ M, J = 1, 2, . . ., I, as set by ELF2
MATL	Real workspace array	input:	on entry, MATL(I, J) must contain the (I, J)th element of the Cholesky factor $\mathbf{L}$ of $\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M}$ for I = 1, 2, . . ., G $\times$ M, J = 1, 2, . . ., I, as set by ELF2
LAMBDA	Real workspace array	input:	on entry, LAMBDA(I) must contain the Ith element of $\boldsymbol{\lambda}$ (the solution of the triangular system $\mathbf{L}\boldsymbol{\lambda} = \mathbf{M}^{\mathrm{T}}\widetilde{\mathbf{h}}$) for I = 1, 2, . . ., G $\times$ M, as computed within ELF2
RES	Real array of dimension (IM, N)	input/output:	on entry, RES(I, J) contains the Ith component of $\boldsymbol{\eta}_{\mathrm{J}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., N, as set by ELF2; on exit, RES(I, J) contains the Ith component of $\widehat{\mathbf{a}}_{\mathrm{J}}$ for I = 1, 2, . . ., M, J = 1, 2, . . ., *N*

The following subroutines are also called from within ELF2. Subroutine CHOLDC returns the Cholesky factor of a symmetric real matrix **M**, and its determinant in the form $a2^b$. Subroutine CHOLFR solves for **x** in the system $\mathbf{Lx} = \mathbf{b}$, with **L** lower triangular, using forward substitution.

SUBROUTINE CHOLDC(M, NDIM, N, D1, D2, EPS, IFAULT)

Formal parameters

M	Real array of dimension (NDIM, NDIM)	input/output:	on entry, contains the matrix **M** (only the upper triangle is needed); on successful exit, contains the required Cholesky factor with the strict upper triangle set to 0

NDIM	Integer	input:	the declared dimension of **M**
N	Integer	input:	the order of **M**
D1	Real	output:	on successful exit, D1 is set equal to a in the expression $\|\mathbf{M}\| = 2^b a$
D2	Real	output:	on successful exit, D2 is set equal to b in the expression $\|\mathbf{M}\| = 2^b a$
EPS	Real	input:	the machine epsilon, as set by MACHEP with a call from within ELF1
IFAULT	Integer	output:	a fault indicator: $= 1$ if **M** is not positive definite; $= 0$ otherwise (on a successful exit)

SUBROUTINE CHOLFR(MATL, NDIM, N, RHSOL)

Formal parameters

MATL	Real array of dimension (NDIM, NDIM)	input:	on entry, contains the lower triangular matrix **L**
NDIM	Integer	input:	the declared dimension of **L**
N	Integer	input:	the order of **L**
RHSOL	Real array of dimension *NDIM*	input/output:	on entry, contains the right-hand side vector **b**; on exit, contains the solution vector **x**

Finally, subroutine CHOLBK is called from within subroutine CRES to solve for **c** in the system $\mathbf{L}^{\mathrm{T}}\mathbf{c} = \boldsymbol{\lambda}$, with **L** lower triangular, using backward substitution. Subroutine MACHEP is called from within subroutine ELF1 to calculate the machine epsilon.

SUBROUTINE CHOLBK(MATL, NDIM, N, RHSOL)

Formal parameters

MATL	Real array of dimension (NDIM, NDIM	input:	on entry, contains the lower triangular matrix **L**
NDIM	Integer	input:	the declared dimension of **L**
N	Integer	input:	the order of **L**
RHSOL	Real array of dimension NDIM	input/output:	on entry, contains the right-hand side vector $\boldsymbol{\lambda}$; on exit, contains the solution vector **c**

SUBROUTINE MACHEP(EPSIL)

Formal parameter

EPSIL	Real	output:	the machine epsilon

Auxiliary Algorithms

Subroutines DECOMP and SOLVE in Moler (1972) are used to solve the system of linear equations (6) within subroutine CGAMMA.

Restrictions

Subroutine ELF2 will terminate abnormally if the model turns out to be either non-stationary or non-invertible. The model will be non-stationary if the matrix $\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1^{\mathrm{T}}$ is not positive definite. This is detected by subroutine CHOLDC, which in turn will make ELF2 return IFAULT = 11. The model will be non-invertible if the computation of the matrix sequence Ξ_k is explosive, i.e. when

$$\sum_{i=1}^{m}\sum_{j=1}^{m}|\Xi_h(i,j)| > \sum_{k=1}^{\min(h,q)}\left\{\sum_{i=1}^{m}\sum_{j=1}^{m}|\Xi_k(i,j)|\right\}$$

for at least one $h < n-1$. This check for non-invertibility takes place within subroutine CXI and, if it holds, ELF2 will return IFAULT = 12. However, the exact likelihood can still be evaluated when any root of the moving average operator lies on the unit circle, provided that the other roots have moduli larger than 1.

To calculate the log-likelihood function for the model

$$\mathbf{w}_t = \boldsymbol{\mu} + \mathbf{a}_t$$

(i.e. to take $p = q = 0$), q should be set to 1 and the first m rows and columns of THETA to 0 (note that $p = q = 0$ is not allowed by subroutine ELF1 since some of the workspace arrays in ELF2 would have zero length).

Although subroutine ELF1 will not flag an error if N is set to 1, it is left to the user to ensure that $N \times M$ is greater than the number of parameters in the model.

Precision

When using the present algorithm on machines with small word length, all the real variables should be replaced by double-precision variables. This amounts to replacing all declarations of type REAL by declarations of type DOUBLE PRECISION and all remaining occurrences of REAL by DBLE. To make an accurate and machine-independent decision on whether a given symmetric real matrix is not positive definite, subroutine CHOLDC makes use of the quantity machine epsilon as discussed in Dennis and Schnabel (1983), pages 318–319. Overflow or underflow will not occur in the calculation of $|\mathbf{Q}|$ and $|\mathbf{I} + \mathbf{M}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{M}|$ since these determinants are stored in the form $2^b a$ (Martin and Wilkinson, 1965).

Accuracy

When the model considered is invertible, the matrix sequence Ξ_k converges to $\mathbf{0}$, the more quickly the larger the moduli of the zeros of $|\boldsymbol{\Theta}(B)|$ are (when $q = 0$, it is clear that $\Xi_k = \mathbf{0}$ for $k \geqslant 1$). This fact may be exploited in the subsequent computations involved in steps (f), (g) and (k) within subroutine ELF2, since if $\Xi_k = \mathbf{0}$ for, say, $k > r^*$ then not all of these operations need to be carried out. The sequence Ξ_k may be considered to have converged when

$$\sum_{i=1}^{m} \sum_{j=1}^{m} |\Xi_{r^*+1}(i, j)| < \delta,$$

where the parameter $\delta > 0$ can be used to control the desired degree of approximation to the exact computation of the whole sequence. However, to avoid complications due to any $\mathbf{\Theta}_i = \mathbf{0}$ for $i < q$ (i.e. in the case of seasonal or gapped models), once the above condition has been met, subroutine CXI calculates the next q Ξ_ks following Ξ_{r^*} to make sure that convergence has effectively occurred. The convergence criterion can be made sufficiently rigid (i.e. δ sufficiently small) that the error implied by considering $\Xi_k = \mathbf{0}$ for $k > r^*$ becomes negligible.

This property, which may save much computing time, is analogous to the 'quick recursions' property offered by the Chandrasekhar equations that form the basis of the method of Shea (1989), pages 169–170. Thus, the comparisons between the exact (calculated with Ξ_k from $k = 1$ to $k = n - 1$) and the 'approximate' (calculated with $\Xi_k = \mathbf{0}$ for $k > r^*$) log-likelihood functions for any given model display results which are very similar to those reported by Shea (1989).

Timing and Related Algorithm

The algorithm implemented in subroutine ELF2 requires fewer time-consuming

TABLE 1
Ratios between the number of multiplications and divisions required by algorithm AS 242 of Shea (1989) and those required by subroutine ELF2 for several vector ARMA models

Model	*Ratios of operations for the following values of m and n:*			
	m = 2		*m = 4*	
	n = 100	*n = 200*	*n = 100*	*n = 200*
AR(1)	1.19	1.10	3.88	2.65
AR(2)	1.27	1.14	3.41	2.79
MA(1)	2.46	2.49	2.81	2.83
MA(2)	1.88	1.91	2.09	2.11
ARMA(1, 1)	2.52	2.54	3.39	3.21
$\mathrm{AR}(1)_4$	1.18	1.13	2.16	2.07
$\mathrm{MA}(1)_4$	1.47	1.53	1.60	1.67
$\mathrm{ARMA}(1, 1)_4$	1.56	1.66	2.04	2.03
$\mathrm{AR}(1)_{12}$	0.84†	0.85†	1.27	1.27
$\mathrm{MA}(1)_{12}$	0.81†	1.00	0.84†	1.06
$\mathrm{ARMA}(1, 1)_{12}$	0.91†	1.05	1.25	1.29
$\mathrm{AR}(1) \times \mathrm{MA}(1)_4$	1.50	1.57	1.77	1.79
$\mathrm{AR}(1) \times \mathrm{MA}(1)_{12}$	0.81†	1.00	0.86†	1.08
$\mathrm{MA}(1) \times \mathrm{AR}(1)_4$	1.57	1.64	2.08	2.02
$\mathrm{MA}(1) \times \mathrm{AR}(1)_{12}$	0.90†	0.98†	1.27	1.27
$\mathrm{ARMA}(1, 1) \times \mathrm{AR}(1)_4$	1.40	1.49	1.85	1.82
$\mathrm{ARMA}(1, 1) \times \mathrm{MA}(1)_4$	1.37	1.46	1.59	1.65
$\mathrm{ARMA}(1, 1) \times \mathrm{AR}(1)_{12}$	0.87†	0.94†	1.23	1.24
$\mathrm{ARMA}(1, 1) \times \mathrm{MA}(1)_{12}$	0.75†	0.96†	0.80†	1.02

†Algorithm AS 242 requires fewer time-consuming operations than the new algorithm.

operations than algorithm AS 242 of Shea (1989) does in many cases. To see this, the exact log-likelihood function has been evaluated for a variety of vector ARMA models. In Table 1, the ratio between the number of multiplications and divisions required by the algorithm of Shea (1989) and those required by subroutine ELF2 is presented for each of the models considered.

It can be seen that algorithm AS 242 requires fewer time-consuming operations than the new algorithm only for low dimension ($m = 2$) high order models, whereas the new algorithm performs faster than algorithm AS 242 (by a factor of more than 2 in many cases) for higher dimension ($m = 4$) models. In fact, the relative efficiency of the new algorithm increases with the dimension m of the model (and, in almost all cases, with the series length n also). Thus, the ratios for $m = 4$ and $n = 200$ are all advantageous to the new algorithm, irrespective of the orders p and q.

A similar table can be constructed using timings rather than operation counts. In that case, the timing ratios are all advantageous to the new algorithm for the models considered in Table 1. This fact shows not only that the new algorithm requires fewer time-consuming operations than algorithm AS 242 of Shea (1989) does but also that it is more efficiently coded (by means of, for example, carrying out within a single loop two or more tasks which might be included within two or more different loops).

Additional Comments

The main purpose of subroutine ELF2 is to serve as an integral part of an exact maximum likelihood estimation program for vector ARMA models. It can be shown (Mauricio, 1995a) that maximizing the exact likelihood function is equivalent to minimizing

$$S(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\mu}, \mathbf{Q}|\mathbf{w})^m\, |\mathbf{Q}|\, |\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda}|^{1/n}.$$

Since, on successful exit, subroutine ELF2 returns $S(\boldsymbol{\Phi}, \boldsymbol{\Theta}, \boldsymbol{\mu}, \mathbf{Q}|\mathbf{w})$ (the quadratic form in the exact likelihood) and $|\mathbf{Q}|^n\, |\boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{\Lambda}|$ (the determinant in the exact likelihood) as F1 and F2 respectively, it is straightforward to use that output in the computation of the objective function to be minimized. Further, since subroutine ELF2 automatically detects parameter values that imply non-stationarity, non-invertibility and/or non-positive definiteness of $\mathbf{Q}$, an objective function that penalizes these situations can be constructed following the guidelines in Shea (1984) and Mauricio (1995a). The residual vector should be evaluated (using subroutine CXI) only after the minimization routine has converged, since it is not used during the estimation process. Finally, note that the computations involved in this process can be speeded up by using the approximation to the exact likelihood function (based on the convergence of the $\boldsymbol{\Xi}_k$s to $\mathbf{0}$) discussed previously.

Acknowledgements

I am grateful to Arthur B. Treadway, under whose supervision this work was carried out, to the Algorithm Editor and the referee for some helpful suggestions, and to Sonia Sotoca for some useful hints on the Fortran compiler.

References

Dennis, J. E. and Schnabel, R. B. (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs: Prentice Hall.

Hall, A. D. and Nicholls, D. F. (1980) The evaluation of exact maximum likelihood estimates for VARMA models. *J. Statist. Computn Simuln*, **10**, 251–262.

Kohn, R. and Ansley, C. F. (1982) A note on obtaining the theoretical autocovariances of an ARMA process. *J. Statist. Computn Simuln*, **15**, 273–283.

Ljung, G. M. and Box, G. E. P. (1979) The likelihood function of stationary autoregressive-moving average models. *Biometrika*, **66**, 265–270.

Martin, R. S. and Wilkinson, J. H. (1965) Symmetric decomposition of positive definite band matrices. *Numer. Math.*, **7**, 355–361.

Mauricio, J. A. (1995a) Exact maximum likelihood estimation of stationary vector ARMA models. *J. Am. Statist. Ass.*, **90**, 282–291.

——— (1995b) A corrected algorithm for computing the theoretical autocovariance matrices of a vector ARMA model. *Working Paper 9502*. Instituto Complutense de Análisis Económico, Madrid.

Moler, C. B. (1972) Algorithm 423: Linear equation solver. *Communs Ass. Comput. Mach.*, **15**, 274.

Nicholls, D. F. and Hall, A. D. (1979) The exact likelihood function of multivariate autoregressive-moving average models. *Biometrika*, **66**, 259–264.

Shea, B. L. (1984) Maximum likelihood estimation of multivariate ARMA processes via the Kalman filter. In *Time Series Analysis: Theory and Practice* (ed. O. D. Anderson), vol. 5, pp. 91–101. Amsterdam: North-Holland.

——— (1989) Algorithm AS 242: The exact likelihood of a vector autoregressive moving average model. *Appl. Statist.*, **38**, 161–184.