Estimation of a Digitised Gaussian ARMA model by Monte Carlo Expectation Maximisation

DOI:
10.1016/j.csda.2018.10.015

Document Version
Accepted author manuscript

Estimation of a Digitised Gaussian ARMA model by Monte Carlo Expectation Maximisation

Hannah Lennon*, Jingsong Yuan**

School of Mathematics, University of Manchester, Oxford Road, Manchester, M13 9PL, UK

Abstract

Dependence modelling of integer-valued stationary time series has gained considerable interest. A generalisation of the ARMA model has been previously provided using the binomial operator and its estimation carried out using Markov Chain Monte Carlo methods. There are also various models that make use of a latent process. The time series is considered now as a digitised version of a Gaussian ARMA process, which is equivalent to assuming a Gaussian copula with ARMA dependence. Naturally this becomes an incomplete data problem and an EM algorithm can be used for maximum likelihood estimation. Due to the complexity of the conditional distribution given the observed data, a Monte Carlo E-step is implemented. Details of the MCEM algorithm are provided and standard errors of the parameter estimates are considered. Examples with real and simulated data are provided.

Keywords: Maximum likelihood, EM algorithm, Monte Carlo EM, integer-valued time series, ARMA model, Gaussian copula

*Present address: International Agency for Research on Cancer, World Health Organization, 150 cours Albert Thomas, 69372 Lyon cedex 08, France

**Corresponding Author

Email addresses: lennonh@fellows.iarc.fr (Hannah Lennon), j.yuan@manchester.ac.uk (Jingsong Yuan)

1. Introduction

Dependence modelling of integer-valued stationary time series has gained considerable interest. There have been many attempts to generalise stationary time series models to the integer-valued/count case (Davis et al., 2016). One particular approach is to use the binomial thinning operator, beginning with the work of McKenzie (1985) and Al-Osh and Alzaid (1987) for the INAR(1) model. The INARMA(p, q) model has similarities to the ARMA model and its estimation has been considered by Neal and Subba Rao (2007) using MCMC.

Let $\{y_t\}$ be an integer-valued stationary time series that is observed at $t = 1, \ldots, n$. Such a time series $\{y_t\}$ can be obtained by simply taking the integer parts of a continuous-valued stationary time series $\{x_t\}$, or by more elaborate digitisation as follows. Let $F(y)$ be a discrete distribution function and $F^{-1}(p) = \inf\{y;\ F(y) \geq p\}$ its generalised inverse. Let $G(x)$ be the distribution function of x_t. It is well-known that $u_t = G(x_t) \sim U(0, 1)$ and $y_t = F^{-1}(u_t)$ has distribution given by F. The inverse CDF or quantile transform from u_t to y_t is a standard method of simulating from F (Ross, 2012).

It is much easier to specify a model for the latent process $\{x_t\}$ because it is continuous-valued. The distribution of the discrete-valued series $\{y_t\}$ then follows. The Gaussian ARMA(p, q) model has a rich covariance structure that is well understood (Priestley, 1981). Therefore it provides the basis of a fairly general model that can be written as follows:

$$x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \beta_j \varepsilon_{t-j}, \text{ and} \tag{1}$$

$$y_t = F^{-1}(\Phi(x_t)),\ t = 1, 2, \ldots, \tag{2}$$

where F is an unknown distribution function of the discrete type, and Φ is the CDF of the standard normal distribution $N(0, 1)$. It is assumed that the innovation $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$ for some $\sigma_\varepsilon > 0$ such that $x_t \sim N(0, 1)$. The ARMA coefficients $\{\alpha_i\}$ and $\{\beta_j\}$ satisfy the usual causality and invertibility conditions with no common factors between the AR and MA characteristic polynomials. For simplicity of notation, we assume that y_t only takes values in $\{0, 1, 2, ...\}$. This can be extended to any countable set. The orders p and q are assumed to be known.

This model has a system equation (1) for the latent ARMA process $\{x_t\}$ and an observation equation (2) for y_t in terms of x_t. It is known to give rise to a Gaussian copula for $y_1, \ldots, y_n$, which is a natural choice for its many desirable properties (Song, 2000). We present it as a digitised ARMA model because the concept is intuitively plausible and it leads naturally to the estimation method considered here. A more general model with time dependent marginals and covariates has been considered by Masarotto and Varin (2012). For models that utilise a latent process in different ways, we refer to Davis et al. (2016) and the references therein.

It is easy to see that the transformation from x_t to y_t is not invertible:

$$y_t = y \;\Leftrightarrow\; \Phi^{-1}(F(y-1)) < x_t \leq \Phi^{-1}(F(y)), \tag{3}$$

for any integer y. In other words, each probable value of y_t corresponds to an interval for the value of x_t. The stationarity of the observed series $\{y_t\}$ follows from that of $\{x_t\}$ which we assume.

The likelihood of the observations $y_1, \ldots, y_n$ is

$$\begin{aligned} & P(\Phi^{-1}(F(y_t-1)) < x_t \leq \Phi^{-1}(F(y_t)),\ t = 1, \ldots, n) \\ = & \sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} (-1)^{i_1+\cdots+i_n} \Phi_\Sigma(\Phi^{-1}(F(y_1-i_1)), \ldots, \Phi^{-1}(F(y_n-i_n))), \end{aligned} \tag{4}$$

where Φ_Σ is the joint distribution function of $x_1, \ldots, x_n$ (Song, 2000). This is difficult to compute when n is moderately large due to the 2^n number of n-dimensional normal distribution function evaluations involved.

The aim of this work is to find a computationally feasible method for maximum likelihood estimation of the underlying ARMA parameters. We consider it as an incomplete data problem because the latent process $\{x_t\}$ is not observed. The EM (Expectation Maximisation) algorithm is designed for such a situation (Dempster et al., 1977). It has proved very popular and useful, see McLachlan and Krishnan (2007) for a summary of applications among 1700 publications.

The complete data $\mathbf{x} = (x_1, \ldots, x_n)^\top$ has the Gaussian ARMA likelihood and established algorithms can be used for its computation. Let the complete data log-likelihood be $\ell(\boldsymbol{\theta}; \mathbf{x})$ and the incomplete data log-likelihood $\ell(\boldsymbol{\theta}; \mathbf{y})$ where $\mathbf{y} = (y_1, \ldots, y_n)^\top$. The EM algorithm works on an estimate of the

former, yet it can be shown that the latter log-likelihood improves through the iterations, i.e.,

$$\ell(\boldsymbol{\theta}^{(i)}; \mathbf{y}) \;\geq\; \ell(\boldsymbol{\theta}^{(i-1)}; \mathbf{y}).$$

Furthermore, if $\boldsymbol{\theta}^{(i)}$ converges then it converges to a local stationary point of $\ell(\boldsymbol{\theta}; \mathbf{y})$ (Wu, 1983).

The EM algorithm alternates between two stages, the E-step (Expectation) and the M-step (Maximisation), where the parameter values are updated repeatedly until a convergence criterion is met. More specifically, the E-step at iteration i evaluates the objective function

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = \mathrm{E}_{\boldsymbol{\theta}^{(i-1)}} \left[\ell \left(\boldsymbol{\theta}; \mathbf{x} \right) \mid \mathbf{y} \right], \tag{5}$$

as an estimate of $\ell \left(\boldsymbol{\theta}; \mathbf{x} \right)$ using $\boldsymbol{\theta}^{(i-1)}$ in the conditional expectation and the M-step maximises the objective function with respect to $\boldsymbol{\theta}$ to obtain an updated set of parameter values $\boldsymbol{\theta}^{(i)}$.

For the digitised Gaussian ARMA model, the second moments of the latent process $\{x_t\}$ given the observed count data $\mathbf{y}$ in the E-step are not easy to evaluate, even though we know the form of the conditional distribution to be truncated multivariate normal. Therefore a Monte Carlo E-step is used resulting in a Monte Carlo EM algorithm (Wei and Tanner, 1990), where a Monte Carlo estimate of the objective function (5) is used in the M-step. We implement a Geweke-Hajivassilou-Keane (GHK) simulator (Geweke, 1989; Hajivassiliou and McFadden, 1998; Keane, 1994) to generate samples recursively.

This paper is organised as follows. We derive the E-step of the EM algorithm for model (1)-(2) in Subsection 2.1. A Monte Carlo E-step is described in 2.2 with further details of simulation in 2.3. A brief mention of the M-step is in 2.4. We consider standard errors of the parameter estimates in 2.5. This is followed by practical considerations in 2.6 and a summary of the estimation method in 2.7. Examples using real and simulated data are provided in Section 3. We conclude in Section 4 with a discussion of the strengths and limitations of this MCEM approach to maximum likelihood estimation of the dependence parameters.

2. The MCEM Algorithm

We provide details of the algorithm that is developed specifically for the model given by (1)-(2).

2.1. The E-Step

The latent process $\{x_t\}$ has a Gaussian structure and completely determines the observed series $\{y_t\}$. Therefore the complete data log-likelihood

$$\ell(\boldsymbol{\theta}; \mathbf{x}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}\mathbf{x}^\top \Sigma^{-1}\mathbf{x} \tag{6}$$

only involves $\mathbf{x}$. The variance-covariance matrix Σ can be difficult to invert when the sample size is large, so we adopt the time series approach (Brockwell and Davis, 1987). Let

$$\begin{aligned} e_1 &= x_1, \\ e_2 &= x_2 - \phi_{1,1}x_1, \\ e_3 &= x_3 - (\phi_{2,1}x_2 + \phi_{2,2}x_1), \\ &\vdots \\ e_n &= x_n - (\phi_{n-1,1}x_{n-1} + \cdots + \phi_{n-1,n-1}x_1), \end{aligned} \tag{7}$$

where $\phi_{t1}, \ldots, \phi_{tt}$ are coefficients of the best linear predictor of x_{t+1} using $x_t, \ldots, x_1$ with minimum mean square error. The linear transformation from $\mathbf{x}$ to $\mathbf{e}$, with determinant equal to one, de-correlates the data and the log-likelihood can be written as

$$\begin{aligned} \ell(\boldsymbol{\theta}; \mathbf{x}) &= -\frac{1}{2}\sum_{t=1}^{n}\log(2\pi\tau_t^2) \; - \frac{1}{2}\sum_{t=1}^{n}\left(x_t - \sum_{\ell=1}^{t-1}\phi_{t-1,\ell}x_{t-\ell}\right)^2/\tau_t^2 \\ &= -\frac{1}{2}\sum_{t=1}^{n}\log(2\pi\tau_t^2) \; - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_i x_j, \end{aligned} \tag{8}$$

where τ_t^2 is the variance of $e_t = x_t - \sum_{\ell=1}^{t-1}\phi_{t-1,\ell}x_{t-\ell}$. The elements c_{ij} of Σ^{-1} can be obtained easily from $\{\phi_{t-1,\ell}\}$ and $\{\tau_t^2\}$, which in turn can be calculated recursively using the Levinson-Durbin algorithm (Brockwell and Davis, 1987).

It follows from (3) that the distribution of $\mathbf{x}$ given $\mathbf{y}$ is truncated multivariate normal with truncation interval $(\mathbf{a}, \mathbf{b}]$ consisting of

$$(a_t, b_t] = (\Phi^{-1}\ (F(y_t - 1)),\ \Phi^{-1}\ (F(y_t))]$$

in each dimension $t = 1, \ldots, n$. Thus the E-step at iteration i is to evaluate the objective function

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = -\frac{1}{2}\sum_{t=1}^{n} \log(2\pi\tau_t^2)\ - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[x_i x_j \middle| \mathbf{a} < \mathbf{x} \leq \mathbf{b}\right], \quad (9)$$

where the τ_t^2 and c_{ij} are functions of $\boldsymbol{\theta}$ and the values $\boldsymbol{\theta}^{(i-1)}$ are used in the evaluation of the conditional expectation.

2.2. The Monte Carlo E-step

Direct evaluation of the $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$ function is difficult because the second moments of the truncated multivariate normal

$$\mathrm{E}[x_i x_j | \mathbf{a} < \mathbf{x} \leq \mathbf{b}] = \frac{1}{P(\mathbf{a} < \mathbf{x} \leq \mathbf{b})} \int_{\mathbf{a}}^{\mathbf{b}} x_i x_j \phi_{\Sigma}(\mathbf{x}) \mathrm{d}\mathbf{x} \quad (10)$$

involve not only an n-dimensional integral but also the incomplete data likelihood $P(\mathbf{a} < \mathbf{x} \leq \mathbf{b})$ as given by (4).

Wei and Tanner (1990) first proposed a Monte Carlo (MC) method to estimate the E-step resulting in the MCEM algorithm. The idea is to use simulated samples from the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ to estimate the Q-function. Chan and Ledolter (1995) showed that with suitable initial parameter values, an MCEM sequence will converge with high probability to the set of parameter values at which the maximum of the likelihood occurs.

To implement the MC E-step, we make use of the Geweke-Hajivassilou-Keane (GHK) simulator. This is described in the next sub-section. Once the samples are obtained, the conditional moments (10) are estimated by simple averaging.

2.3. The GHK simulator

It is possible to sample directly from the truncated multivariate normal distribution, $TN_n(\mathbf{0}, \Sigma, \mathbf{a}, \mathbf{b})$, which arises from restricting the multivariate

normal $N_n(\mathbf{0}, \Sigma)$ distribution to the interval $(\mathbf{a}, \mathbf{b}]$. To obtain exact draws, rejection sampling is the simplest method however it can be highly inefficient in high dimensions. Approximate methods such as Markov Chain Monte Carlo have been proposed to overcome the inefficiency (Geweke, 1991), but they can suffer from poor mixing, convergence problems and correlation among samples (Wilhelm and Manjunath, 2010). The standard GHK simulator uses a Cholesky decomposition of Σ and samples recursively from univariate truncated normal distributions (Shum, 2008). In the time series case, we can write

$$\begin{aligned} x_1 &= e_1, \\ x_2 &= e_2 + \theta_{11} e_1, \\ x_3 &= e_3 + \theta_{21} e_2 + \theta_{2,2} e_1, \\ &\vdots \\ x_n &= e_n + \theta_{n-1,1} e_{n-1} + \cdots + \theta_{n-1,n-1} e_1, \end{aligned} \tag{11}$$

where the coefficients θ_{11} to $\theta_{n-1,n-1}$ can be calculated recursively together with e_t and τ_t^2, $t = 1, \ldots, n$ using the innovations algorithm (Brockwell and Davis, 1987). The autocovariances of x_t up to lag $n-1$ are required, which are also calculated recursively, from the ARMA model parameters.

Sampling can be done recursively as follows:

1. Sample $e_1 = \tau_1 z_1$ from $N(0, \tau_1^2)$ so that $x_1 = e_1$ falls within $(a_1, b_1]$;
2. Sample $e_2 = \tau_2 z_2$ from $N(0, \tau_2^2)$ so that $x_2 = e_2 + \theta_{11} e_1$ falls within $(a_2, b_2]$;
3. Sample $e_3 = \tau_3 z_3$ from $N(0, \tau_3^2)$ so that $x_3 = e_3 + \theta_{21} e_2 + \theta_{22} e_1$ falls within $(a_3, b_3]$;

   $\vdots$

n. Sample $e_n = \tau_n z_n$ from $N(0, \tau_n^2)$ so that $x_n = e_n + \theta_{n-1,1} e_{n-1} + \cdots + \theta_{n-1,n-1} e_1$ falls within $(a_n, b_n]$.

Each step above uses the fact that for any $c_t < d_t$,

$$z_t \sim TN(0, 1, c_t, d_t) \;\Leftrightarrow\; u_t = \Phi(z_t) \sim U(\Phi(c_t), \Phi(d_t)]. \tag{12}$$

It is a matter of setting

$$\begin{aligned} c_t &= (a_t - \theta_{t-1,1} e_{t-1} - \cdots - \theta_{t-1,t-1} e_1)/\tau_t, \text{ and} \\ d_t &= (b_t - \theta_{t-1,1} e_{t-1} - \cdots - \theta_{t-1,t-1} e_1)/\tau_t, \end{aligned} \tag{13}$$

simulating u_t from the uniform distribution in (12), then calculating x_t using (11) with $e_t = \tau_t \Phi^{-1}(u_t)$, $t = 1, \ldots, n$. An equivalent formulation of the GHK simulator has been given by Masarotto and Varin (2012) who suggested the use of the Kalman filter to speed up the computation.

The joint density of a GHK sample is $f(\mathbf{x}) = \phi_\Sigma(\mathbf{x})/w(\mathbf{x})$, where $w(\mathbf{x}) = \prod_{t=1}^{n}(\Phi(d_t) - \Phi(c_t))$ is the importance sampling weight (Shum, 2008). This is not the same as the truncated normal density $\phi_\Sigma(\mathbf{x})/P(\mathbf{a} < \mathbf{x} \leq \mathbf{b})$ in (10). However, each $w(\mathbf{x})$ is an unbiased estimator of $P(\mathbf{a} < \mathbf{x} \leq \mathbf{b})$ (Gourieroux and Monfort, 1996). Having obtained m samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, we use the simple average of $\mathbf{x}^{(i)}\mathbf{x}^{(i)\top}$ to estimate $\mathrm{E}[\mathbf{x}\mathbf{x}^\top | \mathbf{a} < \mathbf{x} \leq \mathbf{b}]$. The weighted average

$$\sum_{i=1}^{m} \mathbf{x}^{(i)}\mathbf{x}^{(i)\top} w(\mathbf{x}^{(i)}) / \sum_{i=1}^{m} w(\mathbf{x}^{(i)})$$

provides a better estimate (Shum, 2008) at the expense of computation time.

2.4. The M-step

There is no closed form solution to the M-step so numerical optimisation is used. For consistent results, we use the R package `dfoptim` which implements the Nelder-Mead-Kelley approach (Kelley, 1999).

2.5. Standard errors of estimates

The second derivatives of the observed log-likelihood $\ell(\boldsymbol{\theta};\ \mathbf{y})$ can be calculated through $\ell(\boldsymbol{\theta};\ \mathbf{x})$ using (Louis, 1982)

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta};\ \mathbf{y}) &= \mathrm{E}\left[\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta};\ \mathbf{x}) \mid \mathbf{y} \right], \text{ and} \\
-\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \ell(\boldsymbol{\theta};\ \mathbf{y}) &= \mathrm{E}\left[-\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} \ell(\boldsymbol{\theta}; \mathbf{x}) \mid \mathbf{y} \right] - \mathrm{E}\left[\frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta};\ \mathbf{x})\ \frac{\partial}{\partial \boldsymbol{\theta}^\top} \ell(\boldsymbol{\theta};\ \mathbf{x}) \mid \mathbf{y} \right] \\
&\quad + \frac{\partial}{\partial \boldsymbol{\theta}} \ell(\boldsymbol{\theta};\ \mathbf{y}) \frac{\partial}{\partial \boldsymbol{\theta}^\top} \ell(\boldsymbol{\theta};\ \mathbf{y}). \qquad (14)
\end{aligned}$$

At the MLE, the last term in (14) is zero. Therefore using simulated samples (Chan and Ledolter, 1995) the Fisher information is estimated by

$$-\frac{1}{m} \sum_{j=1}^{m} \frac{\partial^2 \ell(\boldsymbol{\theta}; \mathbf{x}^{(j)})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^\top} - \frac{1}{m} \sum_{j=1}^{m} \left(\frac{\partial \ell(\boldsymbol{\theta}; \mathbf{x}^{(j)})}{\partial \boldsymbol{\theta}} \right) \left(\frac{\partial \ell(\boldsymbol{\theta}; \mathbf{x}^{(j)})}{\partial \boldsymbol{\theta}^\top} \right). \qquad (15)$$

The partial derivatives of the Gaussian ARMA log-likelihood $\ell(\boldsymbol{\theta}; \mathbf{x})$ as given in (6) require additional steps because the innovation variance σ_ε^2 of model (1)- (2) depends on the ARMA parameters. For example, if the dependence structure is AR(2) then

$$\sigma_\varepsilon^2 = (1+\alpha_2)((1-\alpha_2)^2 - \alpha_1^2)(1-\alpha_2)^{-1}.$$

2.6. Practical considerations

Wei and Tanner (1990) suggested the use of a small number m of Monte Carlo samples to begin with, increasing it in later runs to allow the estimates to explore the parameter space, and that it should be large in the final few iterations to reduce the Monte Carlo error in the E-step. The approximate value of a local maximum is often easy to discern and further analysis with larger m can then be carried out, if necessary, to refine the approximation. Every limit of an EM sequence $\boldsymbol{\theta}^{(i)}$ is a stationary point of $\ell(\boldsymbol{\theta}; \mathbf{y})$ and so the choice of the initial parameter values may lead to a local maximum rather than a global maximum, i.e. the maximum likelihood estimate.

The goal is to find the parameter values that globally maximise the likelihood function. In practice, we can run the algorithm several times with different starting points. Increased confidence in an MCEM procedure can be achieved by repeating the procedure with different starting values representative of the parameter space. This allows for easy searching for multiple modes and helps discover the landscape of the likelihood function. The starting values $\boldsymbol{\theta}^{(0)}$ do not need to be close to the true $\boldsymbol{\theta}$; but the EM algorithm will converge very slowly if a poor choice of $\boldsymbol{\theta}^{(0)}$ is used.

Due to the variability introduced in the E-step by simulation, the parameter updates can still fluctuate after 'convergence' so it can be difficult to certify convergence of $\{\boldsymbol{\theta}^{(i)}\}$ (Mollerberg 2003). The issue is expanded when the number of parameters is impractical to monitor. Chan and Ledolter (1995) suggest monitoring the change in log-likelihood $\Delta\ell(\boldsymbol{\theta}; \mathbf{y}) = \ell(\boldsymbol{\theta}^{(i)}; \mathbf{y}) - \ell(\boldsymbol{\theta}^{(i-1)}; \mathbf{y})$ going from $\boldsymbol{\theta}^{(i-1)}$ to $\boldsymbol{\theta}^{(i)}$, estimating it by

$$-\log\left(\frac{1}{m}\sum_{k=1}^{m}\exp\{\ell(\boldsymbol{\theta}^{(i-1)}; \mathbf{x}^{(k)}) - \ell(\boldsymbol{\theta}^{(i)}; \mathbf{x}^{(k)})\}\right), \tag{16}$$

where $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ are samples from the conditional distribution of $\mathbf{x} \mid \mathbf{y}$ using $\boldsymbol{\theta}^{(i)}$. Plot the estimated change in log-likelihood value (16) against the

number of iterations. Convergence can be claimed when the plot appears to fluctuate randomly about the abscissa, for example when the absolute value of (16) is less than a precision value for the last 5 iterations.

2.7. Summary of the estimation method

The estimation method takes the following steps.

1. Set initial values of ARMA parameters, number of MC samples, stopping criterion etc;
2. Estimate marginal distribution nonparametrically using emirical CDF or parametrically by maximising the product of marginal probabilities;
3. Calculate truncation parameters $\mathbf{a}$ and $\mathbf{b}$ using marginal CDF;
4. While convergence is not achieved
   - MC E-step (estimating the Q function):
     - Calculate autocovariance matrix Σ using latest values of ARMA parameters;
     - Generate samples $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}$ using GHK method;
     - Calculate Σ using new ARMA parameter values, run Levinson-Durbin algorithm to obtain τ_t^2 and $\phi_{t-1,\ell}$, and use them to calculate elements c_{ij} of Σ^{-1};
     - Evaluate the Q-function (9) replacing conditional moments $\mathrm{E}[\mathbf{x}\mathbf{x}^\top | a < \mathbf{x} \leq \mathbf{b}]$ by $\frac{1}{m}\sum_{i=1}^{m} \mathbf{x}^{(i)}\mathbf{x}^{(i)\top}$;
   - M-step: Maximise Q function to obtain optimal new values of ARMA parameters;
   - Check convergence and exit loop if criterion is met,
5. Calculate standard errors of ARMA parameter estimates.

For a fully parametric approach including maximum likelihood estimation of marginal parameters, put steps 3 and 4 inside a function to maximise the Q-function with respect to all the parameter values.

As shown in Section 3, the MCEM algorithm can be adapted to accommodate time dependent marginal distributions F_t with shape and regression parameters.

3. Examples

3.1. A simulation study

To assess the performance of the MCEM algorithm, a simulation study was conducted using samples of sizes $n = 250$, 500 and 1000. For comparison we also obtained estimates using the R `gcmr` package (Masarotto and Varin, 2017) which implements the simulated likelihood method of Masarotto and Varin (2012).

We simulated $S = 1000$ samples of each size n from a digitised Gaussian ARMA(1,1) model with $(\alpha_1, \beta_1) = (0.7, -0.5)$ as typical values of the ARMA parameters. The marginal distribution is negative binomial NB(s, π) with parameters $s = 5$ and $\pi = 0.5$, which were estimated by maximising the product of marginal probabilities. The initial values were $(\alpha_1 = 0, \beta_1 = 0)$ for the MCEM algorithm. To aid convergence, the number of Monte Carlo samples began with $m = 100$ in the first five iterations then increased to $m = 500$. The estimated change in log-likelihood was computed at each step to monitor convergence. The sample mean and variance of the parameter estimates were calculated together with the bias, mean squared error and root mean squared error.

Table 1 gives results of estimation using both methods. The mean values of parameter estimates became closer to the true values as n increases. The variances of the estimates decreased as n went from n=250 to n=1000 and the bias values were close to zero. The mean square errors and root MSE's (RMSE's) also decreased as n increased.

The results of MCEM method are similar to those of the simulated likelihood method with smaller biases but larger mean square errors. Both methods worked remarkably well compared with maximum likelihood estimation using Gaussian ARMA(1,1) data with only a small increase in mean square error. A typical simulated series of length 1000 from the same ARMA(1,1) model gave estimates 0.6478 and -0.4521 with standard errors 0.1033 and 0.1226 respectively.

The downside of the MCEM method is computation time. For $n = 1000$ it can take more than 4 hours on a desktop PC.

Table 1: Simulation results using $S = 1000$ samples.

Parameter		MCEM			Simulated Likelihood		
		n=250	n=500	n=1000	n=250	n=500	n=1000
α_1=0.7	mean	0.676	0.677	0.685	0.632	0.660	0.677
	var	0.022	0.017	0.010	0.053	0.013	0.004
	bias	-0.024	-0.023	-0.015	-0.068	-0.040	-0.023
	MSE	0.022	0.018	0.011	0.058	0.014	0.005
	RMSE	0.149	0.133	0.103	0.241	0.119	0.071
β_1=-0.5	mean	-0.481	-0.482	-0.489	-0.428	-0.458	-0.476
	var	0.033	0.023	0.013	0.056	0.017	0.007
	bias	0.019	0.018	0.011	0.072	0.042	0.024
	MSE	0.033	0.023	0.013	0.062	0.018	0.008
	RMSE	0.183	0.152	0.115	0.248	0.136	0.087

3.2. Real Data Example

We use the polio data (Zeger, 1988) which often appear in the literature as an example of real world data to illustrate the MCEM algorithm. The values are monthly counts of reported polio cases in the US from January 1970 to December 1983 with $n = 168$ observations. Following existing work on these data (Masarotto and Varin, 2012), we assume an ARMA(2,1) model for the underlying time series and a negative binomial distribution $NB(s, \pi_t)$ for y_t, with mean $\mu_t = s(1 - \pi_t)/\pi_t$ satisfying

$$\begin{aligned}\log\left(\mu_t\right) = \gamma_0 + \gamma_1(t-73)/1000 + \gamma_2\cos\left(\tfrac{2\pi(t-1)}{12}\right) + \gamma_3\sin\left(\tfrac{2\pi(t-1)}{12}\right) \\ + \gamma_4\cos\left(\tfrac{2\pi(t-1)}{6}\right) + \gamma_5\sin\left(\tfrac{2\pi(t-1)}{6}\right).\end{aligned} \tag{17}$$

The reason for using covariates is due to perceived seasonality and a trend in the data. We refer to Enciso-Mora et al. (2009) for an INAR(1) model with the same explanatory variables.

We implemented the MCEM algorithm using estimated regression parameters from fitting the marginal model using `glm.nb` from the MASS library. The initial values for the ARMA parameters were all zeros and the number of Monte Carlo samples begins small with $m = 10$ and increases at every tenth iteration to 50, 10^2, 5×10^2, 10^3, 5×10^3, 10^4, 5×10^4 and 10^5.

Table 2: Estimated parameters for polio data using MCEM and simulated likelihood method.

		Estimation Method			
		MCEM		Simulated likelihood	
		estimate	se	estimate	se
ARMA	α_1	-0.5664	(0.260)	-0.5229	(0.220)
	α_2	0.2701	(0.099)	0.3046	(0.090)
	β_1	0.7214	(0.272)	0.6959	(0.229)
Marginal	$1/s$	0.5671	(0.484)	0.5700	(0.170)
	γ_0	0.2093	(0.096)	0.2095	(0.121)
	γ_1	-4.3318	(1.895)	-4.3151	(2.284)
	γ_2	-0.1430	(0.129)	0.1215	(0.147)
	γ_3	-0.5025	(0.138)	-0.4967	(0.157)
	γ_4	0.1682	(0.131)	0.1903	(0.129)
	γ_5	-0.4214	(0.132)	-0.4030	(0.128)

The MCEM parameter values at iterations 1 to 90 are plotted in Fig. 1. We see that the sequences begin to stabilise after 50 iterations and convergence was confirmed by the change in log-likelihood (Fig. 2). The parameters and the estimated change in log-likelihood continues to fluctuate randomly around zero. The parameter estimates and standard errors are given in Table 2. The standard errors of ARMA parameter estimates were calculated using the estimated information matrix (15), while those of marginal parameters were taken from negative binomial regression.

Figure 1: MCEM iterations with parameters α_1 (solid), α_2 (dashed) and β_1 (dotted).

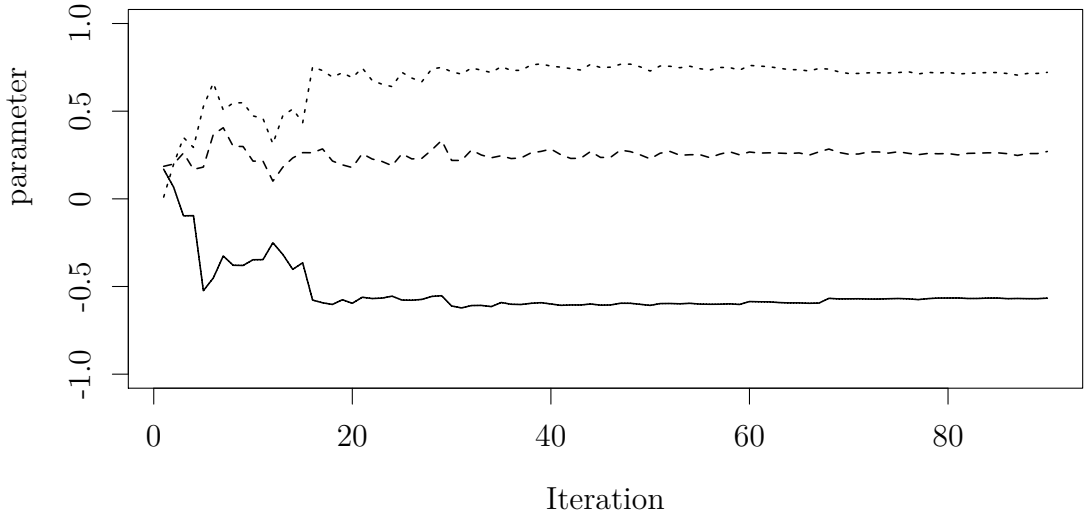
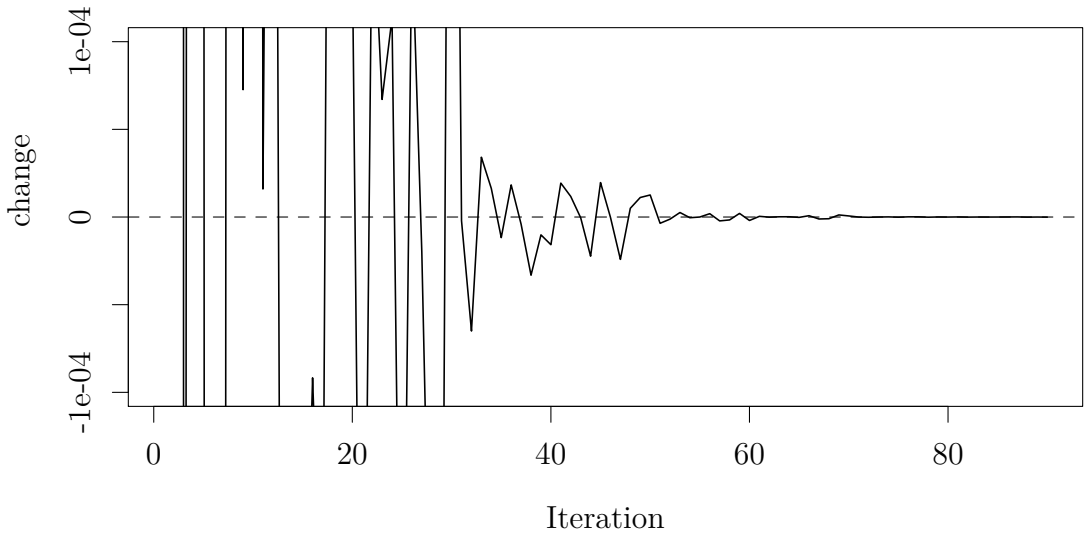


Figure 2: Estimated change in log-likelihood over iterations 1 to 90 where m increases every 10 iterations.



4. Concluding Discussion

We have derived an MCEM algorithm for maximum likelihood estimation of the digitised Gaussian ARMA model. The marginal distributions can have parameters and regression coefficients to be estimated together with the ARMA parameters.

In the Monte Carlo E-step, we utilised a GHK simulator as an alternative to approximate samplers such as the commonly used Gibbs sampler (Tan et al., 2007) that can suffer from poor mixing and convergence issues (Christen et al., 2017).

We found the quality of the MCEM estimates to be similar to the maximum simulated likelihood method of Masarotto and Varin (2012) with smaller biases but larger mean square errors. Both methods compared well with maximum likelihood estimation using the original ARMA data.

We have demonstrated the robustness of the MCEM method and considered time series of different lengths. A common criticism of the EM algorithm is that convergence can be relatively slow. We note that computation time increases significantly for longer time series and with larger values of m. For $n = 1000$, the timing can come close to 18 hours. This is already using chol which is a wrapper for LAPACK routines. The minimum time for convergence when $n = 1000$ was 4.2 hours and for shorter time series, the timings are generally much faster. Work is under way to speed up computation so that importance sampling weights can be applied.

The R code to fit the digitised Gaussian ARMA to integer-valued time series using the MCEM algorithm is available from the first author as an R package named `copulaIVTS`.

All computation was carried out in the R (R Core Team, 2013) environment.

5. Acknowledgements

This work was supported by the University of Manchester under an Engineering and Physical Sciences Research Council (EPSRC) DTA Award.

The polio data are available from the cited source.

6. References

M. A. Al-Osh and A. A. Alzaid. First-order integer-valued autoregressive (INAR(1)) process. *Journal of Time Series Analysis*, 8(3):261–275, 1987.

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer, 2 edition, 1987.

K. S. Chan and J. Ledolter. Monte Carlo EM Estimation for Time Series Models Involving Counts. *Journal of the American Statistical Association*, 90(429):242–252, 1995.

J. A. Christen, C. Fox, and M. Santana-Cibrian. Optimal direction Gibbs sampler for truncated multivariate normal distributions. *Communications in Statistics - Simulation and Computation*, 46(4):2587–2600, 2017. doi: 10.1080/03610918.2015.1053926. URL `http://dx.doi.org/10.1080/03610918.2015.1053926`.

R. A. Davis, S. H. Holan, R. Lund, and N. Ravishanker. *Handbook of discrete-valued time series*. CRC Press, 2016.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

V. Enciso-Mora, P. Neal, and T. Subba Rao. Integer valued ar processes with explanatory variables. *Sankhya: The Indian Journal of Statistics*, 71-B: 248–263, 2009.

J. Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society*, page 1317–1339, 1989.

J. Geweke. Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints and the Evaluation of Constraint Probabilities. In *Computing science and statistics: Proceedings of the 23rd symposium on the interface*, page 571–578. Citeseer, 1991.

C. Gourieroux and A. Monfort. *Simulation-Based Econometric Methods*. Oxford University Press, 1996.

V. A. Hajivassiliou and D. L. McFadden. The method of simulated scores for the estimation of LDV models. *Econometrica*, page 863–896, 1998.

M. P. Keane. A computationally practical simulation estimator for panel data. *Econometrica: Journal of the Econometric Society*, page 95–116, 1994.

C. T. Kelley. *Iterative methods for optimization*, volume 18. Siam, 1999.

T. A. Louis. Finding the Observed Iinformation matrix when using the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 226–233, 1982.

G. Masarotto and C. Varin. Gaussian Copula Marginal Regression. *Electronic Journal of Statistics*, 6:1517–1549, 2012.

G. Masarotto and C. Varin. Gaussian copula regression in R. *Journal of Statistical Software, Articles*, 77(8):1–26, 2017. ISSN 1548-7660. doi: 10.18637/jss.v077.i08. URL `https://www.jstatsoft.org/v077/i08`.

E. McKenzie. Some simple models for discrete variate time series. *JAWRA Journal of the American Water Resources Association*, 21(4):645–650, 1985. ISSN 1752–1688.

G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*, volume 382. John Wiley & Sons, 2007.

P. Neal and T. Subba Rao. MCMC for Integer-valued ARMA processes. *Journal of Time Series Analysis*, 28(1):92–110, 2007.

M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, 1981.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`.

S. M. Ross. *Simulation*. Academic Press, 2012.

M. Shum. Notes on ghk. *Available at http://www.its.caltech.edu/ mshum/gradio/ghk_desc.pdf*, 2008.

P. X.-K. Song. Multivariate dispersion models generated from gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320, 2000.

M. Tan, G.-L. Tian, and H.-B. Fang. An efficient MCEM algorithm for fitting generalized linear mixed models for correlated binary data. *Journal of Statistical Computation and Simulation*, 77(11):929–943, 2007.

G. C. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

S. Wilhelm and B. Manjunath. tmvtnorm: A package for the truncated multivariate normal distribution. *sigma*, 2:2, 2010.

C. F. J. Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

S. L. Zeger. A regression model for time series of counts. *Biometrika*, 75(4): 621–629, 1988.