

***Efficient order selection algorithms for integer
valued ARMA processes***

Enciso-Mora, Victor and Neal, Peter and Subba Rao,
Tata

2006

MIMS EPrint: **2006.415**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Efficient order selection algorithms for integer valued ARMA processes

Víctor Enciso-Mora*, Peter Neal[†] and T. Subba Rao
University of Manchester

December 7, 2006

Address:

Mathematics Department,
University of Manchester,
PO Box 88,
Manchester,
M60 1QD,
UK

E-mail: P.Neal-2@manchester.ac.uk

Abstract

We consider the problem of model (order) selection for integer valued autoregressive moving-average (INARMA) processes. A very efficient Reversible Jump Markov chain Monte Carlo (RJMCMC) algorithm is constructed for moving between INARMA processes of different order. An alternative in the form of the EM algorithm is given for determining the order of an integer valued autoregressive (INAR) process. Both algorithms are successfully applied to both simulated and real data sets.

Keywords: Integer valued time-series, Reversible jump MCMC, EM algorithm, count data.

*The author would like to thank CONACYT (Consejo Nacional de Ciencia y Tecnología, México) for funding this research.

[†]corresponding author

1 Introduction

There has been a steady increase over the last 5 years in studying Integer-valued time-series. Integer-valued time-series naturally arise in the form of count data over sequential time-points. If the counts are large (magnitude '000s) then approximations of the time-series using continuous time-series models such as the autoregressive moving-average (ARMA) process with Gaussian errors are usually adequate. However, if this counts are of low-frequency (typical value less than 20, say) continuous approximations are totally inadequate. Hence the need to develop time-series models appropriate for low-frequency count data.

An excellent review of integer-valued time-series, and in particular, integer-valued ARMA (INARMA) models is given in McKenzie (2003). Most attention has been restricted to INAR(p) processes due to the complicated form of the likelihood of the full INARMA model. Even with $p = 1$ the likelihood of the INAR(p) process is not trivial to analyse and a number of works have considered statistical inference for this case only, see Franke and Seligmann (1993), Freeland and McCabe (2004) and McCabe and Martin (2005). Recently Jung and Tremayne (2006) have studied the case $p = 2$.

In Neal and Subba Rao (2006), an MCMC algorithm for obtaining samples from the posterior distributions of parameters of a general INARMA(p, q) process with known orders p and q was devised. MCMC is a particularly attractive tool to use in analysing INARMA processes since statistical inference is greatly assisted by the use of data augmentation. In other words, the likelihood for the model parameters given the data is intractable, but given carefully chosen extra information (data) the likelihood becomes tractable and analysis is very straightforward. One of the main limitations of Neal and Subba Rao (2006) is that they assume p and q are known/fixed. Therefore the aim of the current work is to study the case where p and q are themselves parameters of the model to be estimated.

We consider two avenues for treating p and q as parameters in the model. The first is an extension of the MCMC algorithm of Neal and Subba Rao (2006). Since the number of parameters in the model depends upon p and q , it is necessary to develop an MCMC algorithm which can move between different parameter spaces. This can be done very naturally using Reversible Jump (RJ)MCMC which was introduced in Green (1995). The biggest drawback which is often encountered by RJMCMC is problems with moving between different parameter spaces, in that, the acceptance probability of such moves can be very low. We circumvent this problem and develop a very efficient RJMCMC algorithm by exploiting the structure of the INARMA(p, q) model. In particular, we identify a parameter which is well estimated directly from

the data and utilise this parameter in constructing efficient parameter space switching moves.

The second approach also uses data augmentation but in a maximum likelihood estimation (MLE) setting, via the EM algorithm. The EM algorithm can be employed for fixed p and q and the resulting likelihood compared using some model selection criterion such as the BIC (Bayesian Information Criterion). Unfortunately the EM algorithm can only be constructed for INAR(p) processes. One possible solution for INARMA(p, q) processes is to use a Monte-Carlo EM algorithm. However, in this case computation of the likelihood is not possible, and so, model selection, via information criterion, is not possible. We discuss how the RJMCMC algorithm can be used to pinpoint the most parsimonious model and to provide good initial parameter estimates for the EM algorithm.

The paper is structured as follows. In Section 2, the INARMA(p, q) model is briefly described. In Section 3, a review of the MCMC algorithm used in Neal and Subba Rao (2006) is given followed by a description of our RJMCMC algorithm. In particular, we give important details as to how the efficient algorithm is constructed. The EM algorithm is introduced in Section 4. This is followed by results in Section 5 for both simulated test data sets and real life data sets. Finally, in Section 6 we discuss minor modifications and extensions of the work presented in this paper.

2 The INARMA(p, q) Model

The integer valued ARMA (INARMA(p, q)) process is the natural analogue of the ARMA(p, q) process for integer valued time series. For $\{X_t; -\infty < t < \infty\}$, the INARMA(p, q) satisfies the difference equation:

$$X_t = \sum_{i=1}^p \alpha_i \circ X_{t-i} + \sum_{j=1}^q \beta_j \circ Z_{t-j} + Z_t, \quad t \in \mathbb{Z}, \quad (2.1)$$

for some generalised, Steutel and van Harn, operators α_i ($1 \leq i \leq p$) and β_j ($1 \leq j \leq q$) (see, Steutel and van Harn (1979) and Latour (1997)) and Z_t ($-\infty < t < \infty$) are independent and identically distributed according to an arbitrary, but specified, non-negative integer valued random variable Z with $E[Z^2] < \infty$. The operators ensure that the process is integer valued and, for simplicity of exposition, we shall restrict attention to binomial operators. The binomial operator, $\gamma \circ$, for a non-negative integer-valued random variable, W say, is defined as

$$\gamma \circ W = \begin{cases} \text{Bin}(W, \gamma) & W > 0, \\ 0 & W = 0. \end{cases}$$

As noted in Neal and Subba Rao (2006), the procedures described in this paper readily extend to other general Steutel and van Harn operators.

To ensure that the above INARMA(p, q) process is (second-order) stationary, we require that $\sum_{i=1}^p \alpha_i < 1$. The corresponding constraint on the β 's for invertibility of the time-series is that $\sum_{j=1}^q \beta_j < 1$.

3 MCMC algorithm

3.1 Introduction and review of Neal and Subba Rao (2006) algorithm

We begin by giving a brief summary of the procedure in Neal and Subba Rao (2006) for studying the INARMA(p, q) process for fixed p and q . This is followed by a detailed description of the order switching steps for p and q using reversible jump (RJ) MCMC. The within-order moves (*i.e.* fixed p and q) for updating $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_q)$ and λ can be performed using the algorithm of Neal and Subba Rao (2006).

Inference for the INARMA(p, q) is greatly facilitated by data augmentation. That is, for $t \in \mathbb{Z}$, we represent $\alpha_i \circ X_{t-i}$ ($1 \leq i \leq p$) and $\beta_j \circ Z_{t-j}$ ($1 \leq j \leq q$) by $Y_{t,i}$ and $V_{t,j}$, respectively, with $\mathbf{Y}_t^p = (Y_{t,1}, Y_{t,2}, \dots, Y_{t,p})$ and $\mathbf{V}_t^q = (V_{t,1}, V_{t,2}, \dots, V_{t,q})$. Thus for $t \in \mathbb{Z}$,

$$Z_t = X_t - \sum_{i=1}^p Y_{t,i} - \sum_{j=1}^q V_{t,j}.$$

In Neal and Subba Rao (2006) the augmented data are treated as variables to be updated as part of the MCMC procedure alongside the parameters. In particular, Neal and Subba Rao (2006) gives an efficient procedure for updating the augmented data. The parameters can be sampled using their conditional distributions (see Neal and Subba Rao (2006), (6)–(8)) if the following natural conjugate priors are chosen:

$$\begin{aligned} \pi(\boldsymbol{\alpha}) &= p!; & 0 \leq \alpha_i < 1 \ (1 \leq i \leq p), \quad \sum_{i=1}^p \alpha_i < 1; \\ \pi(\boldsymbol{\beta}) &= q!; & 0 \leq \beta_j < 1 \ (1 \leq j \leq q), \quad \sum_{j=1}^q \beta_j < 1; \\ \pi(\lambda) &\sim \text{Gam}(A_\lambda, B_\lambda); & \text{for some } A_\lambda, B_\lambda > 0. \end{aligned}$$

Note that the above priors on $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are (uninformative) uniform over the permissible parameter ranges. Finally, let $\pi(p)$ ($\pi(q)$) denote the prior probability for AR (MA) order p (q).

Letting $r = \max\{p, q\}$, we have the joint density for the augmented data and the parameter values,

$$f(\mathbf{v}, \mathbf{y}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda | \mathbf{x}) \propto \pi(p)p!\pi(q)q! \prod_{t=1}^n \left\{ \frac{\lambda^{z_t}}{z_t!} e^{-\lambda} \prod_{i=1}^p \left\{ \binom{x_{t-i}}{y_{t,i}} \alpha_i^{y_{t,i}} (1 - \alpha_i)^{x_{t-i} - y_{t,i}} \right\} \right. \\ \left. \times \prod_{j=1}^q \left\{ \binom{z_{t-j}}{v_{t,j}} \beta_j^{v_{t,j}} (1 - \beta_j)^{z_{t-j} - v_{t,j}} \right\} \right\} \lambda^{A_\lambda} e^{-B_\lambda \lambda} \prod_{i=1}^r \left\{ \frac{\lambda^{z_{1-i}}}{z_{1-i}!} e^{-\lambda} \right\} \quad (3.1)$$

subject to $\sum_{i=1}^p y_{t,i} + \sum_{j=1}^q v_{t,j} + z_t = x_t$ ($1 \leq t \leq n$).

3.2 Order determination Algorithm

The algorithm outlined thus far is that given by Neal and Subba Rao (2006). The algorithm of Neal and Subba Rao (2006) is used as a sub-algorithm here for the within-model moves. We now describe the order switching step. At each iteration, we propose to either increase or decrease the AR order by 1 with each move being proposed with probability 0.5. This is subject to constraints at $p = 0$ and $p = p_{max}$. We then do the same for the MA order with constraints at $q = 0$ and $q = q_{max}$.

Before considering the order determination algorithm we consider the INARMA(p, q) model and highlight the key features for our RJMCMC algorithm. Whilst RJMCMC is a very useful tool it is often plagued by inefficient model switching moves. That is, the probability of accepting moves between different models (orders) can be extremely low. As noted in Brooks *et al.* (2003) there is no generic way for constructing efficient RJMCMC algorithms. However, a good understanding of the model under consideration can enable the development of efficient RJMCMC algorithms.

Note that if the time-series is stationary

$$E[X_t] = \sum_{i=1}^p \alpha_i E[X_{t-i}] + \sum_{j=1}^q \beta_j E[Z_{t-j}] + E[Z_t] \quad (3.2)$$

where $E[X_t] = E[X_s]$ for all $s, t \in \mathbb{Z}$. Therefore if $Z_t \sim Po(\lambda)$,

$$E[X_t] = \lambda \left(1 + \sum_{j=1}^q \beta_j \right) / \left(1 - \sum_{i=1}^p \alpha_i \right) = \kappa, \quad \text{say.} \quad (3.3)$$

The key point which we shall exploit is that κ is well-estimated from the data, in that, $\hat{\kappa} = \frac{1}{n} \sum_{i=1}^n x_i$. Therefore whilst information concerning $(\boldsymbol{\alpha}, p)$, $(\boldsymbol{\beta}, q)$ and λ is not readily available, the data is extremely informative about κ . Moreover, the algorithm in Neal and Subba Rao (2006) gives iid observations from the posterior distribution of κ . Therefore a natural suggestion when constructing the order switching step would be to keep κ fixed. This can be implemented very easily as long as we are not proposing to

move the AR (MA) order p (q) either to or from $p = 0$ ($q = 0$). That is, the procedure is very simple whenever we propose order moves within the same model, be it INAR(p), INMA(q) or INARMA(p, q). The procedure for moving between the different models requires a little more care.

Thus we begin by describing the algorithm for increasing and decreasing the AR order where neither p nor p' (the new proposed order) are 0. In order to achieve transitions between parameter subspaces that retain detailed balance in the Reversible Jump framework, we require a deterministic invertible mapping that complies with the so-called ‘dimension-matching’ assumption mentioned in Green (1995). A splitting and amalgamation mapping is described below.

Consider the move from p to $p' = p + 1$. One way to keep κ fixed is to choose α' such that

$$\sum_{i=1}^{p'} \alpha'_i = \sum_{i=1}^p \alpha_i$$

with $\beta' = \beta$ and $\lambda' = \lambda$. This can be easily implemented as follows. Let $U \sim U[0, 1]$ and let K be drawn uniformly at random from $\{1, 2, \dots, p\}$. Then for $i \in \{1, 2, \dots, p\}/K$, set $\alpha'_i = \alpha_i$. Let $\alpha'_K = U\alpha_K$ and $\alpha'_{p+1} = (1 - U)\alpha_K$. Thus we are splitting the K^{th} AR term and this should be accompanied by a similar splitting in the corresponding augmented data terms. That is, for $1 \leq t \leq n$, let $S_t \sim \text{Bin}(y_{t,K}, U)$ and set $y'_{t,K} = S_t$, $y'_{t,p+1} = y_{t,K} - S_t$ and $\mathbf{S} = (S_1, S_2, \dots, S_n)$. All other augmented data terms are kept fixed.

For the reverse move from $p + 1$ to $p' = p$. Choose K uniformly at random from $\{1, 2, \dots, p\}$. Set $\alpha'_K = \alpha_K + \alpha_{p+1}$ and for $1 \leq t \leq n$, set $y'_{t,K} = y_{t,K} + y_{t,p+1}$. That is, we amalgamate the $(p+1)^{th}$ AR term into the the K^{th} AR term. Therefore letting $\mathbf{w} = (\mathbf{y}, \mathbf{v}, \mathbf{z})$ (the augmented data) and $\theta = (\alpha, p, \beta, q, \lambda)$ (the parameters), we have that the probability of accepting the move to increase the order is:

$$A = 1 \wedge \frac{f(\theta', \mathbf{w}' | \mathbf{x})}{f(\theta, \mathbf{w} | \mathbf{x})} \times \frac{q(\theta', \mathbf{w}' \rightarrow \theta, \mathbf{w})}{q(\theta, \mathbf{w} \rightarrow \theta', \mathbf{w}')} \times |J| \quad (3.4)$$

where J denotes the Jacobian for the transformation from $(\theta, \mathbf{w}, U, K, \mathbf{S})$ to $(\theta', \mathbf{w}', K)$.

From (3.1), we have that

$$\frac{f(\theta', \mathbf{w}' | \mathbf{x})}{f(\theta, \mathbf{w} | \mathbf{x})} = \frac{\pi(p')p'}{\pi(p)} \times \prod_{t=1}^n \frac{\binom{x_{t-k}}{y'_{t,k}} (\alpha'_k)^{y_{t,K'}} (1 - \alpha'_K)^{x_{t-K} - y_{t,K'}} \binom{x_{t-p'}}{y'_{t,p'}} (\alpha'_{p'})^{y'_{t,p'}} (1 - \alpha'_{p'})^{x_{t-p'} - y'_{t,p'}}}{\binom{x_{t-K}}{y_{t,K}} \alpha_K^{y_{t,K}} (1 - \alpha_K)^{x_{t-K} - y_{t,K}}}. \quad (3.5)$$

For the proposal from (θ, \mathbf{w}) to (θ', \mathbf{w}') , we have that

$$q(\theta, \mathbf{w} \rightarrow \theta', \mathbf{w}') = \frac{1}{2} \times \frac{1}{p} \times 1 \times \prod_{t=1}^n \binom{y_{t,K}}{y'_{t,K}} U^{y_{t,K}} (1 - U)^{y_{t,K} - y'_{t,K}}$$

where the terms correspond to increasing the order, K , U and \mathbf{S} , respectively. The reverse move only depends upon the probability of decreasing the order and choosing K , and so is given by

$$q(\theta', \mathbf{w}' \rightarrow \theta, \mathbf{w}) = \frac{1}{2} \times \frac{1}{p}.$$

Finally, the Jacobian can be factorised into $n + 1$ parts corresponding to the map in $(\boldsymbol{\alpha}, U) \rightarrow \boldsymbol{\alpha}'$ and maps $(y_{t,K}, S_t) \rightarrow (y'_{t,K}, y'_{t,p+1})$ ($1 \leq t \leq n$). This leads to $|J| = \alpha_K \times 1^n = \alpha_K$.

Clearly, the decreasing order move from p to $p' = p - 1$ is simply the reverse of that given above.

For the MA order switching step the procedure is essentially identical to that given above except that we combine and split MA terms. In particular, when changing order we ensure that $\boldsymbol{\beta}'$ is such that

$$\sum_{j=1}^{q'} \beta'_j = \sum_{j=1}^q \beta_j$$

with $\boldsymbol{\alpha}' = \boldsymbol{\alpha}$ and $\lambda' = \lambda$. For the order increase move from q to $q' = q + 1$, we sample K uniformly from $\{1, 2, \dots, q\}$ and $U \sim U(0, 1)$, splitting the K^{th} term with $\beta'_K = U\beta_K$ and $\beta'_{q+1} = (1 - U)\beta_K$. Then split the corresponding augmented data terms, by letting $S_t \sim \text{Bin}(v_{t,K}, U)$ ($1 \leq t \leq n$) and setting $v'_{t,K} = S_t$, $v'_{t,q+1} = v_{t,K} - S_t$ and $\mathbf{S} = (S_1, S_2, \dots, S_n)$. The order decreasing move from q simply involves amalgamating the q^{th} term with one of the lower order MA terms (chosen at random) by combining the β coefficients and the augmented data terms in the natural manner.

Finally, we want the algorithm to move between the INARMA(p, q) and the sub-models INAR(p) and INMA(q). In order to do this we again keep κ fixed but can't simply apply a splitting/combination algorithm as described above. We consider the move from INAR(p) to INARMA($p, 1$). In this case we keep $\boldsymbol{\alpha}$ fixed but let λ vary. Let $U \sim U[0, 1]$ and set $\beta'_1 = U$ and $\lambda' = \lambda/(1 + U)$. The updating of the augmented terms is sequential with $S_t \sim \text{Bin}(z_t \wedge z'_{t-1}, U/(1 + U))$, $v'_{t,1} = S_t$ and $z'_t = z_t - S_t$ ($1 \leq t \leq n$). The natural reverse move from INARMA($p, 1$) to INAR(p) involves setting $\lambda' = \lambda(1 + \beta_1)$ and combining the augmented data terms with $z'_t = z_t + v_{t,1}$ ($1 \leq t \leq n$).

The move between the INMA(q) to INARMA($1, q$) is more intricate. In this case we keep $\boldsymbol{\beta}$ fixed but let λ and (\mathbf{v}, \mathbf{z}) vary. Let $U \sim U[0, 1]$ and set $\alpha'_1 = U$ and $\lambda' = \lambda(1 - U)$. For $1 \leq t \leq n$ and $0 \leq j \leq q$, let $S_{t,j} \sim \text{Bin}(v_{t,j}, U)$ with the convention $v_{t,0} \equiv z_t$. Set $y'_{t,1} = \sum_{j=0}^q S_{t,j}$ and set $v'_{t,j} = v_{t,j} - S_{t,j}$ ($0 \leq j \leq q$). Therefore we are taking into account the fact that decreasing λ leads to a decrease in \mathbf{v} as well as \mathbf{z} for fixed $\boldsymbol{\beta}$. For the reverse move from INARMA($1, q$) to INMA(q), set $\lambda' = \lambda/(1 - \alpha_1)$ keeping $\boldsymbol{\beta}$ fixed. Let $\mathbf{S}_t = (S_{t,0}, S_{t,1}, \dots, S_{t,q}) \sim \text{Multinomial}(y_{t,1}, \boldsymbol{\gamma})$ where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1, \dots, \gamma_q)$ and $\gamma_k = \beta_k / \sum_{j=0}^q \beta_j$ ($0 \leq k \leq q$) with the convention that $\beta_0 = 1$. Finally, for $1 \leq t \leq n$, set $v'_{t,j} = v_{t,j} + S_{t,j}$ ($0 \leq j \leq q$).

4 EM Algorithm

Whilst MCMC is not an exclusively Bayesian statistical tool, it is primarily used within the Bayesian paradigm and this is how it is used in Section 3. The aim therefore of this section is to look beyond the Bayesian paradigm towards Maximum Likelihood Estimation (MLE) of the parameters and classical model selection, via information criterion. Since the likelihood is intractable given the data \mathbf{x} , one possibility is data augmentation as in Section 3. Data augmentation in the MLE framework is usually performed using the EM algorithm (see Dempster *et al.* (1977)). The natural augmentation is $\mathbf{w} = (\mathbf{v}, \mathbf{y}, \mathbf{z})$ as used in Section 3 which gives us the likelihood proportional to (3.1) with $A_\lambda = B_\lambda = 0$. Then the MLE for $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)$ are:

$$\begin{aligned}\hat{\alpha}_i &= \sum_{t=1}^n y_{t,i} / \sum_{t=1}^n x_{t-i} \quad (1 \leq i \leq p) \\ \hat{\beta}_j &= \sum_{t=1}^n v_{t,j} / \sum_{t=1}^n z_{t-j} \quad (1 \leq j \leq q) \\ \hat{\lambda} &= \frac{1}{n} \sum_{t=1}^n z_t.\end{aligned}\tag{4.1}$$

However the EM algorithm relies upon being able to compute the expected values of \mathbf{w} given \mathbf{x} and $\boldsymbol{\theta}$. For the INARMA(p, q) model the dependence of $(\mathbf{v}_t, \mathbf{y}_t, z_t)$ upon \mathbf{z}_{t-1} makes the expectation step infeasible. A Monte-Carlo EM algorithm can be used but this is very slow due to the slow convergence of the parameters and the high rejection rate in the rejection sampler used to estimate the E-step.

For the INAR(p) model it is possible to compute the expected values of \mathbf{y} given \mathbf{x} and $\boldsymbol{\theta}$. Furthermore, the likelihood of $\boldsymbol{\theta}$ can be computed as a by-product of the EM algorithm. The EM algorithm can only be used for the INAR(p) with fixed order p . However, computation of the log-likelihood enables us to compare different orders of p using model selection tools, in particular, the BIC (Bayesian Information Criterion). This is done for comparing different INAR(p) models in Section 5. The EM algorithm is easy to implement and for $p = 1$, the algorithm is very effective being able to compute the E-step and converging rapidly to the parameter MLEs. However, the efficiency of the algorithm rapidly drops away as p increases. This is due in part to the fact that computation of the E-step, which requires calculations of the probabilities for all the possibilities for the missing data, increases exponentially with p . This is accompanied by much slower convergence of the parameters of the MLE which are necessarily heavily correlated. Finally as p increases computer precision becomes an issue since for $1 \leq t \leq n$, each individual possibility for the missing data (\mathbf{y}_t, z_t) have (very) small probability. This manifests itself by seeing, on occasions, slight decreases in the calculation of the log-likelihood, despite the algorithm being correct. Using Fortran 77 with double precision arithmetic, we saw these problems begin to enter the output

for $p = 4$. These problems can be overcome by using multiple precision arithmetic in calculation of the outcome probabilities, and hence, the log-likelihood. Multiple precision arithmetic has been used effectively to calculate very small epidemic probabilities accurately, see Demiris and O'Neill (2006).

5 Results

5.1 Simulation study

We conducted a simulation study in order to assess the performance of the algorithms described in Sections 3 and 4. The models used in the simulation study, along with their true parameters, are presented in Table 5.1.

Model	n	α	β	λ
AR(3)	400	(0.4,0.1,0.3)	-	2
MA(3)	400	-	(0.4,0.1,0.3)	2
ARMA(2,1)	400	(0.4,0.2)	(0.3)	2

Table 5.1 A selection of models used in simulation study.

For each data set, the RJMCMC algorithm was run to obtain a sample of size 100,000 following a burn-in period of 10,000 iterations with $p_{max} = q_{max} = 10$. The priors are chosen as specified in Section 3.1 with $\pi(\lambda) \sim \text{Gam}(1, 1)$. For each of the models, data sets of length 400 were generated. Analysis was done based on the first n observations for $n = 100, 200, 400$.

The choice of prior for p and q is an important consideration. In particular, as either p and q increases by 1, the number of parameters in the model increases by $n + 1$, corresponding to one new AR or MA parameter and n new augmented data values. Therefore we found that a BIC-like penalisation of the model order produced excellent results for simulated data sets. That is, we set $\pi(p) \propto n^{-p/2}$ and $\pi(q) \propto n^{-q/2}$. Such priors upon p and q produced consistent results as the length of $n (\geq 400)$ varies. (We tested $n = 400, 800$ and 1600 .) Alternative priors $\pi(p), \pi(q) \sim \text{Poisson}(\theta)$ and $\pi(p), \pi(q) \sim \text{Geometric}(\theta)$ were also tested but were not as effective as the BIC based prior.

The results produced by the RJMCMC algorithm can be seen in figures 1–3 which show histograms of the estimated posterior densities for p and q when $n = 100$ and $n = 400$. The corresponding estimates of α, β and λ for the true models when $n = 400$ are given in table 5.2.

For the AR(3) data set (figure 1) we can see that for $n = 100$ the marginal posterior probabilities for $p = 3$ and $q = 0$ (the true orders) are approximately 0.60 and 0.75, respectively. When the sample size increases from $n = 100$ to $n = 400$ the data becomes much more informative and consequently the performance of the algorithm is much better, with corresponding marginal posterior probabilities for $p = 3$ and $q = 0$ exceeding 0.99 and 0.90, respectively.

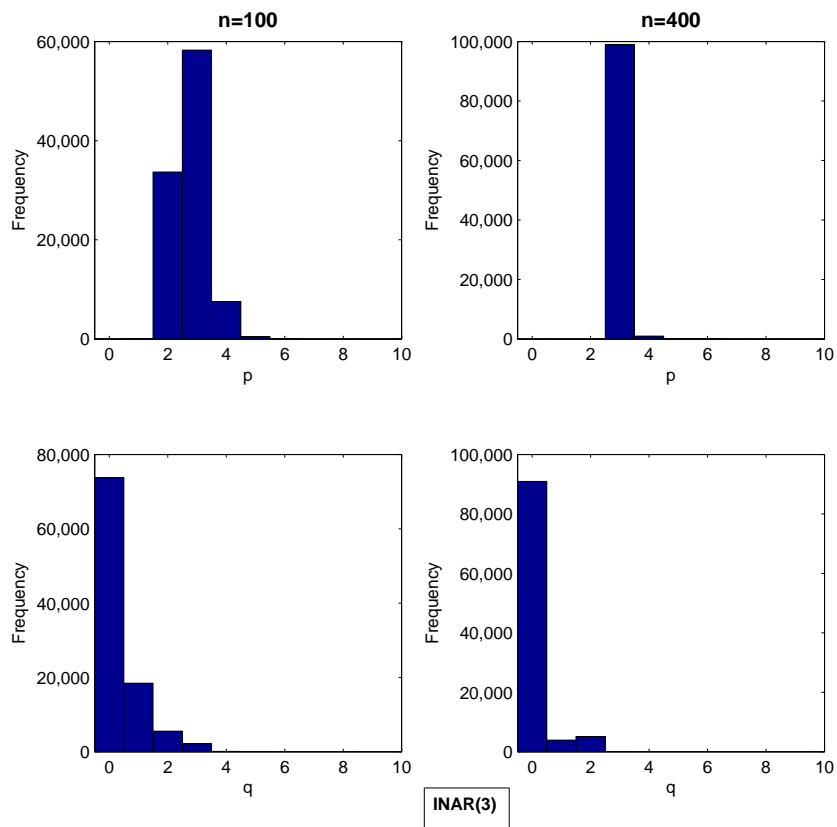


Figure 1: Posterior density plots for the distributions of p and q when the true model is INAR(3).

It is worth noting that it is the joint posterior distribution of p and q which is of particular interest. Table 5.3 shows the estimated posterior probabilities for the three most visited models when $n = 400$ with the posterior probability of the simulated model (INAR(3)) exceeding 0.90.

Analogous analysis for the case where the simulated model is an INMA(3), figure 2, shows a considerable improvement when we increase the sample size from $n = 100$ to $n = 400$. In particular, for $n = 400$ the posterior probability of the INMA(3) model exceeds 0.98.

In the ARMA case we can see that the algorithm once again favours the simulated model, with the

Model	Probability
INAR(3)	0.902
INARMA(3,2)	0.049
INARMA(3,1)	0.039

Table 5.3 Model probabilities when simulated data comes from an INAR(3), $n = 400$.

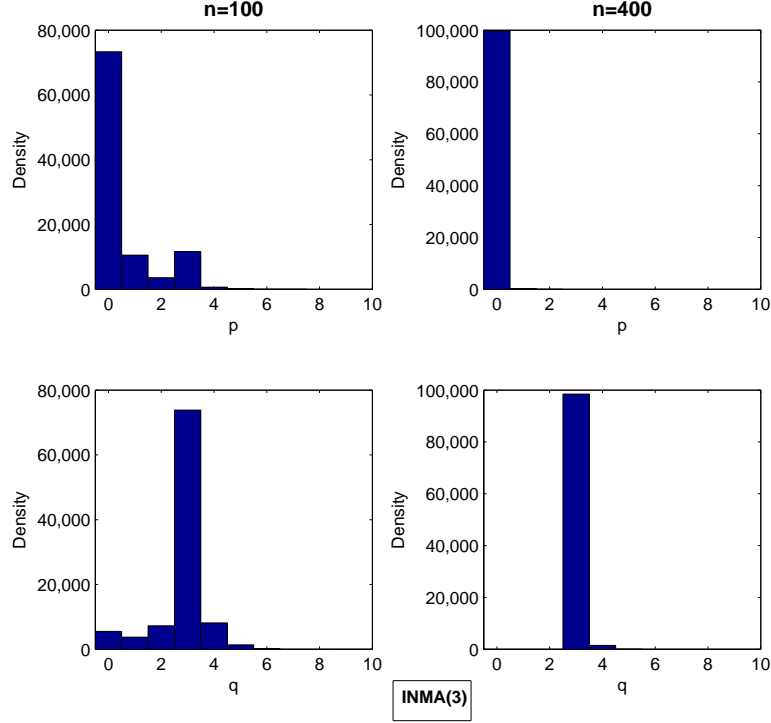


Figure 2: Posterior density plots for the distributions of p and q when the true model is INMA(3).

posterior probability for the INARMA(2,1) exceeding 0.95 when $n = 400$. As mentioned in Neal and Subba Rao (2006) the parameter estimates for the MA parameters are more volatile than those for the AR parameters since the data is more informative about the latter parameters. This is confirmed by the increased size of the estimated standard deviations of $\hat{\beta}_1$, as a consequence the algorithm also underestimates λ .

As mentioned in Neal and Subba Rao (2006) we can assess whether the MCMC algorithm has converged and consequently that the Markov Chain is sampling from its stationary distribution by visual inspection of the time series plots of the MCMC samples. The main diagnostic is for the model switching step. (For fixed p and q the algorithm is that given in Neal and Subba Rao (2006).) In all the above analysis

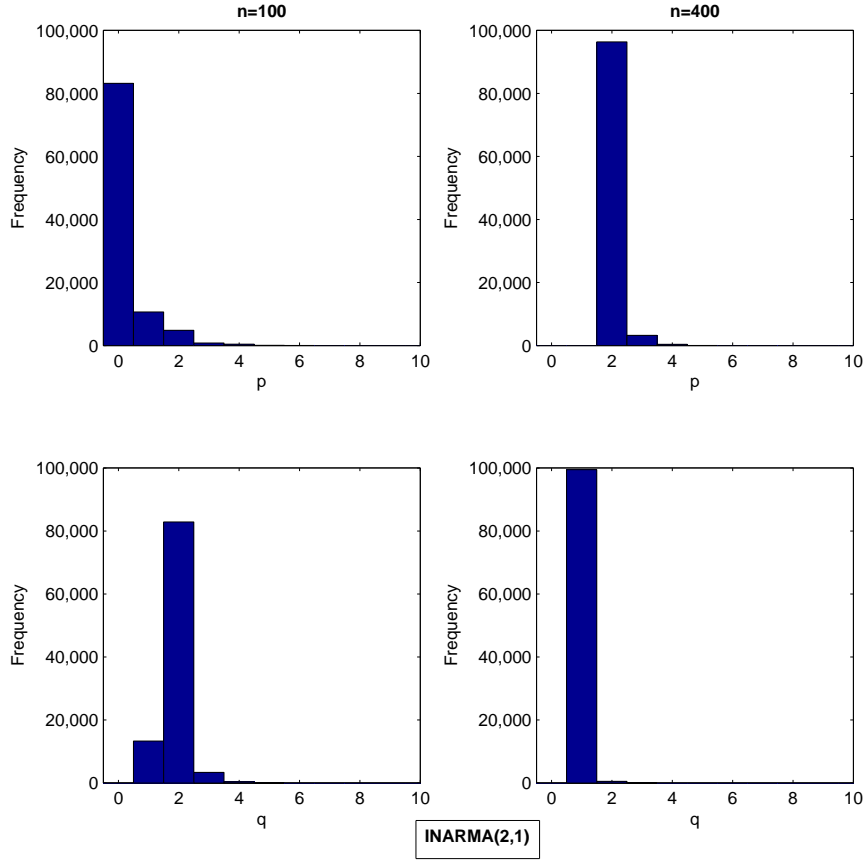


Figure 3: Posterior density plots for the distributions of p and q when the true model is INARMA(2,1).

Model	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\lambda}$
AR(3)			
Mean	(0.4473, 0.1418, 0.2439)	-	1.667
SD	(0.0418, 0.0475, 0.0453)	-	0.3207
MA(3)			
Mean	-	(0.3743, 0.1861, 0.3582)	1.937
SD	-	(0.0774, 0.0786, 0.0728)	0.1066
ARMA(2,1)			
Mean	(0.408, 0.195)	(0.624)	1.787
SD	(0.0634, 0.0473)	(0.2142)	0.2684

Table 5.2 Mean and standard deviations of posterior parameters estimates, $n = 400$.

we have started from a saturated model INARMA(5,5) with the algorithm converging quickly (within the burn-in) period to the simulated model. However, to demonstrate the robustness of the algorithm, we analysed the INAR(3) data set with the algorithm started in the INMA(3) model. Figure 4 shows a run of size 50,000 with no burn-in for this case. The algorithm appears to converge to the posterior distribution within less than 30,000 iterations.

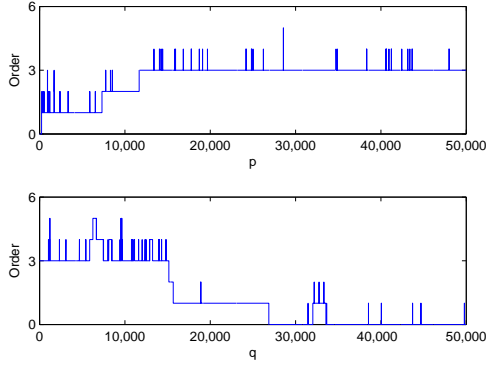


Figure 4: Time series plots of \hat{p} and \hat{q} for an INAR(3) data set with starting values $p_0 = 0$, $q_0=3$.

Finally, the EM algorithm of Section 4 was run for the AR(3) data set with $p = 1, 2, 3, 4$. Using the BIC, the AR(3) model was clearly selected with corresponding MLEs; $\hat{\alpha} = (0.4534, 0.1359, 0.2343)$ and $\hat{\lambda} = 1.7592$. These estimates are very close the posterior means obtained from the RJMCMC algorithm given in Table 5.2.

5.2 Real life data

5.2.1 Westgren Data Set

The first data set we consider is the Westgren gold particle data set analysed in Jung and Tremayne (2006) and Neal and Subba Rao (2006). Figure 5 shows a plot of this data set which consists of 370 counts of gold particles in a solution at equidistant points in time. In Jung and Tremayne (2006), it is concluded that an INAR(2) model fits the data adequately. Estimation of the parameters then proceeds via the method of moments.

We ran the RJMCMC algorithm to obtain a sample of size 100,000 from the posterior distribution of the orders p and q following a burn-in of 10,000 iterations. We use the same priors as in Section 5.1, namely, $\pi(p) \propto n^{-p/2}$, $\pi(q) \propto n^{-q/2}$ and $\pi(\lambda) \sim Gam(1, 1)$.

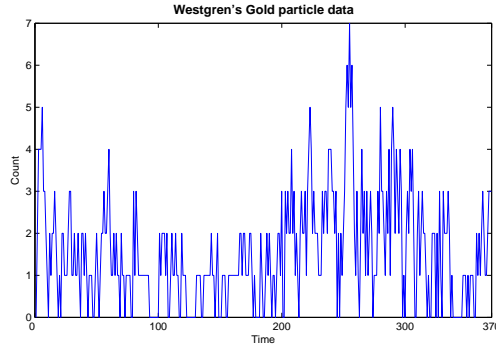


Figure 5: Westgren gold particle data set.

Figure 6 gives posterior histograms for the orders p and q . The results support the findings of Jung and Tremayne (2006) with the posterior probabilities of the three most likely models being 0.669, 0.249 and 0.056 for the INAR(2), INAR(3) and INAR(4) models, respectively. Table 5.4 gives the parameters estimates obtained from the RJMCMC algorithm for the INAR(2) and INAR(3) models along with the results obtained in Jung and Tremayne (2006). The parameter estimates for the INAR(2) model are very close to those obtained in Jung and Tremayne (2006). An advantage of our approach is that we can just as easily provide estimates for the INAR(3) or any other model that the RJMCMC algorithm suggests whereas the approach taken in Jung and Tremayne (2006) is limited by the complex form of the likelihood of the INAR(p) model for large p .

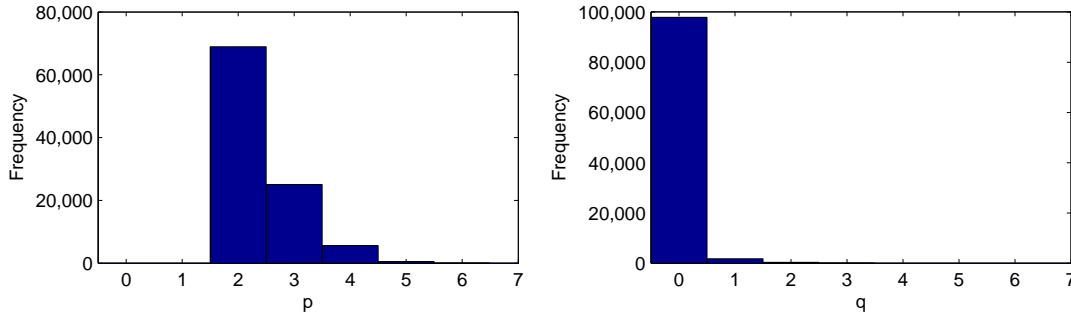


Figure 6: Posterior density plots for the distributions of p and q for the Westgren gold particle data set.

The EM algorithm was also applied to the Westgren data set for INAR(p) with $p=1,2,3,4$. Using the BIC, model INAR(2) was selected just ahead of model INAR(3). The corresponding MLEs are again very close to the posterior means given in Table 5.4. In particular, for INAR(2) $\hat{\alpha} = (0.4716, 0.1798)$ and $\hat{\lambda} = 0.5450$ and for INAR(3), $\hat{\alpha} = (0.4672, 0.1433, 0.0954)$ and $\hat{\lambda} = 0.4546$.

Parameter	α_1	α_2	α_3	λ
INAR(2)	0.465 (0.0485)	0.183 (0.0543)	-	0.554 (0.0720)
INAR(3)	0.456 (0.0477)	0.149 (0.0551)	0.100 (0.0478)	0.462 (0.0755)
JT-INAR(2)	0.453 (-)	0.213 (-)	-	0.518 (-)

Table 5.4 Mean (standard deviations) of parameter estimates for the Westgren data set.

5.2.2 Cut Injury Data Set

The second data set under consideration is the cut injury data set analysed in Zhu and Joe (2006). This data set consists of the monthly number of claims of short-term disability benefits made by injured workers to the British Columbia Workers' Compensation Board from January 1985 to December 1994. This data set focuses particularly on cut related injuries in the logging industry. Figure 7 shows the time series plot of this data set.

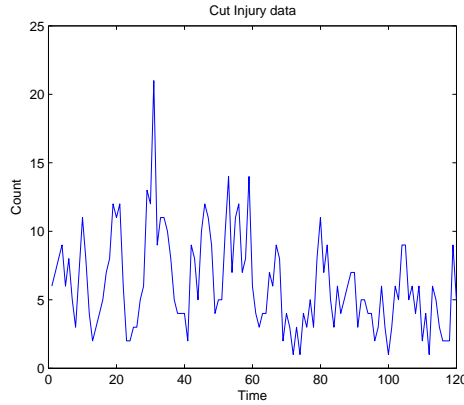


Figure 7: Cut injury in logging industry data set.

Zhu and Joe (2006) fit several models to this data set all of them being of autoregressive type. Since the data is monthly, exploratory analysis was used to check for seasonality. Plotting the autocorrelation and partial autocorrelation functions (ACF and PACF) gave weak evidence for a non-zero lag 12 correlation (an annual effect). However, for the purposes of this paper this seasonality effect will be ignored. Furthermore, the ACF and PACF gave support for both INMA(2) and INMA(3) models for the data.

The RJMCMC algorithm was run to obtain a sample of size 100,000 from the posterior distribution of the orders p and q following a burn-in of 10,000 iterations. We used the same priors as before.

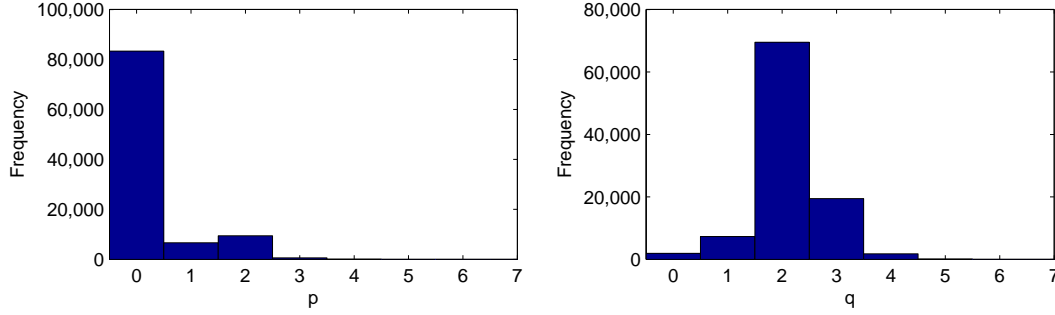


Figure 8: Posterior density plots for the distributions of p and q for the Cut injury data set.

Figure 8 gives histograms for the marginal posterior distributions of p and q . From the joint posterior distribution the INMA(2) and INMA(3) models have posterior probabilities 0.627 and 0.164, respectively. No other model has posterior probability exceeding 0.05. These findings are supported by our initial exploratory analysis using the ACF and PACF plots. Table 5.5 shows the estimated posterior means and standard deviations obtained for the INMA(2) and INMA(3) models.

Parameter	β_1	β_2	β_3	λ
INMA(2)	0.547 (0.0801)	0.353 (0.0883)	-	3.089 (0.2110)
INMA(3)	0.498(0.0859)	0.3032 (0.0818)	0.1402 (0.0745)	2.941 (0.1861)

Table 5.5 Mean (standard deviations) of parameter estimates for the Cut injury data set.

6 Discussion

We have presented an efficient RJMCMC algorithm for conducting inference for the orders p and q of an INARMA(p, q) process. The algorithm has shown to be very successful at detecting the correct order for simulated data sets and produced good results for real life data sets. Furthermore, the algorithm freely moves between the INARMA(p, q) and the sub-models INAR(p) and INMA(q). An alternative to MCMC has been given in the form of the EM algorithm for INAR(p) processes. This gave very similar results to the MCMC algorithm where comparisons were applicable.

Finally, the RJMCMC methodology can be used to analyse other integer-valued time series models. In particular, the authors are considering the inclusion of explanatory variables into the INARMA(p, q) model. In such a case the question as to whether or not to include an explanatory variable can be

considered. The RJMCMC methodology is also clearly applicable to time series with an unknown number of parameter change-points, see Green (1995), Section 4.

Acknowledgements

We thank Robert Jung for providing us with Westgren’s gold particle data set.

References

- [1] Brooks, S. Giudici, P. and Roberts, G. (2003) Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *J. R. Statist. Soc. B* **65**, 3–55.
- [2] Demiris, N. and O’Neill, P. (2006) Computation of final outcome probabilities for the generalised stochastic epidemic. *Stat. Comput.* **16** 309–317.
- [3] Dempster, A. P. ,Laird, N. M. and Rubin, D. B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Statist. Soc. B* **39**, 1–38.
- [4] Franke, J. and Seligmann, T. (1993) Conditional maximum likelihood estimates for INAR(1) processes and their application to modelling epileptic seizure counts. *Time Series Analysis*. (ed. T. Subba Rao.) Chapman and Hall, New York. pp. 310–330.
- [5] Freeland, R. K. and McCabe, B. P. M. (2004) Forecasting discrete valued low count time series. *International Journal of Forecasting* **20**, 427–434.
- [6] Green P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711–732.
- [7] Latour, A. (1997) The multivariate GINAR(p) process. *Advances in Applied Probability* **29**, 228–248
- [8] Jung, R.C. and Tremayne, A.R. (2006) Coherent forecasting in integer time series models. *International Journal of Forecasting* **22**, 223–238
- [9] McCabe, B. P. M. and Martin, G. M. (2005) Bayesian predictions of low count time series. *International Journal of Forecasting* **22**, 315–330.
- [10] McKenzie, E. (2003) Discrete variate time series. *Stochastic Processes: Modelling and Simulation*. (eds. D. N. Shanbhag and C. R. Rao.) Elsevier. pp. 573–606.
- [11] Neal, P. and Subba Rao, T. (2006) MCMC for integer-valued ARMA processes. *To appear in J. Time Ser. Anal.*
- [12] Steutel F. and van Harn, K. (1979) Discrete analogues of self-decomposability and stability. *Annals of Probability* **7**, 893–899
- [13] Zhu R. and Joe H. (2006) Modelling count data time series with markov processes based on binomial thinning. *Journal of Time Series Analysis* **27**, 725–738