MCMC for integer valued ARMA processes

Peter Neal*and T. Subba Rao
University of Manchester

June 7, 2006

Address:

Mathematics Department,

University of Manchester,

Sackville Street,

PO Box 88,

Manchester,

M60 1QD,

UK

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

*corresponding author

Abstract

The Classical statistical inference for integer valued time-series has primarily been restricted to the integer valued autoregressive (INAR) process. Markov chain Monte Carlo (MCMC) methods have been shown to be a useful tool in many branches of statistics and is particularly well suited to integer valued time-series where statistical inference is greatly assisted by data augmentation. Thus in the current work, we outline an efficient MCMC algorithm for a wide class of integer valued autoregressive moving-average (INARMA) processes. Furthermore, we consider noise corrupted integer valued processes and also models with change points. Finally, in order to assess the MCMC algorithms inferential and predictive capabilities we use a range of simulated and real data sets.

Keywords: Integer valued time-series, MCMC, count data.

1 Introduction

Integer valued time-series occur in many different situations, and often in the form of count data. Conventional time-series models such as the standard univariate autoregressive moving-average (ARMA) process with Gaussian errors, assume that the data take values in $\mathbb{R}$. Therefore these standard models are wholely inappropriate for integer valued time-series, especially for low frequency count data where continuous approximations of the discrete process are particularly unsuitable (see, for example, Freeland and McCabe (2004) and McCabe and Martin (2005).) Hence there has been considerable interest in developing and understanding time-series models which are suitable for integer valued processes, see McKenzie (2003), for an excellent review of the subject. This has led to the construction of integer valued ARMA (INARMA) processes, and alternatives such as discrete ARMA (DARMA) processes (see, McKenzie (2003) for a description of the model). Throughout this paper we shall focus on INARMA processes, with a full description being given in Section 2.

Whilst INARMA processes have received a certain amount of attention over the last twenty-five years, they are considerably less well understood and analysed than the standard ARMA processes. In particular, inference for the model parameters governing the INARMA process has received very limited attention. Attention has almost exclusively been restricted to INAR(1) processes, both for classical frequentist inference (see, for example, Franke and Seligmann (1993)) and Bayesian inference (see, for example, Freeland and McCabe (2004) and McCabe and Martin (2005)). Although recently Jung and Tremayne (2006) have considered an INAR(2) process. This is due to the complicated form of the likelihood even for the relatively simple INAR(1) process. One approach that has been taken to avoid such problems is given in Davis *et al.* (2000), where a Poisson regression model is used with a standard ARMA(p,q) time-series forming part of the underlying latent process. Our approach differs radically from Davis *et al.* (2000), in that, we define the process directly in terms of a time-series model, namely the INARMA(p, q) model. To this end MCMC (Markov chain Monte Carlo) greatly assists since MCMC easily incorporates data augmentation which facilitates inference for such models (see, Gilks *et al.* (1996), for a good introduction to the MCMC methodology). Thus we are able to tackle, relatively easily a wide range of time-series models which is a considerable improvement on classical approaches taken in Franke and Seligmann (1993), Freeland and McCabe (2004), McCabe and Martin (2005) and Jung and Tremayne (2006). Note that MCMC has previously been used to tackle a number of time-series problems, see for example, Troughton and Godsill (1997) and Daniels and Cressie (2001).

In analysis of time series data, we are interested, typically, either in the underlying model parameters or the predictive capabilities of the model, or both. Therefore the aim of the current work is to establish a mechanism for conducting inference for both the model parameters and the predictive conditional distribution for a wide range of INARMA processes and their extensions, via MCMC. MCMC is primarily, but not exclusively, used for inference in a Bayesian framework, and in this paper we shall take a Bayesian approach.

In Section 2, the INARMA process is introduced and some extensions of the model are discussed. In Section 3, the likelihood function for the INARMA process is derived, and this is analysed using an MCMC algorithm which we give full details of. This leads to analysis of the data in Section 4. Firstly simulated data and then real life data are used to assess both the capabilities of the MCMC algorithm and of the model. The first real life data set is Westgren's gold particle data set (Westgren (1916)) which has recently been analysed in Jung and Tremayne (2006). The second real life data set concerns the daily scores achieved by a schizophrenic patient on a test of perceptual speed, McCleary and Hay (1980). The patient begins receiving a drug during the course of the study and therefore a change point model is used to assess the affect of the drug on the patient's test scores. Finally, in Section 5 we make some concluding remarks and detail possible extensions.

2 Models

2.1 Integer-valued ARMA processes

An integer valued time-series $\{X_t; -\infty < t < \infty\}$ is called an INARMA(p,q) process, if it is an integer valued autoregressive moving-average process with orders p and q, respectively. The INARMA(p,q) process is given by the following difference equation:

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

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 $E[Z^2] < \infty$. Furthermore the operators are all assumed to be independent. Whilst INAR(p) (integer-valued autoregressive) processes are defined and analysed in Latour (1997) for generalised Steutel and van Harn operators, we shall, for clarity of presentation of the results, restrict attention to binomial operators. The binomial operator, $\gamma\,\circ$, for a non-negative integer-valued random variable, W say, is defined as

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

That is, the operator γ denotes the 'success' probability for the binomial distribution. Clearly for a binomial operator γ, $0 \leq \gamma \leq 1$. The binomial operator is chosen because it is the natural choice of generalised Steutel and van Harn operator in many different situations. (Furthermore, the binomial operator is the operator considered in Steutel and van Harn (1979).) However, most of the results and procedures presented here readily extend to more general, generalised Steutel and van Harn operators with the minimum of fuss.

To ensure that the above INARMA(p,q) process is (second-order) stationary, we require that $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)$ are such that the roots of the p-order polynomial

$$x^p - \alpha_1 x^{p-1} - \ldots - \alpha_{p-1} x - \alpha_p = 0$$

are inside the unit circle (see, Latour (1997)). We shall, however, use the stronger, but easier to check criterion that $\sum_{i=1}^{p} \alpha_i < 1$, to ensure that the process is stationary. Note the parallels with the requirements for stationarity of a real-valued ARMA process. We shall also add a constraint for $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_q)$, in that, we shall assume $\sum_{j=1}^{q} \beta_j < 1$. For a real-valued ARMA process this condition is sufficient for the time series to be invertible. It is an open question whether this condition is sufficient for the INARMA process to be invertible.

One natural extension of the INARMA(p, q) process is the inclusion of a noise term, D_t say. Let $\{W_t; -\infty < t < \infty\}$ denote an INARMA(p, q) process with noise, then

$$W_t = X_t + D_t, \qquad t \in \mathbb{Z}, \tag{2.2}$$

where $\{X_t\}$ is INARMA(p, q) process of the form (2.1) and D_t $(-\infty < t < \infty)$ are independent and identically distributed non-negative integer-valued random variables. Furthermore, we assume that the two processes $\{X_t; -\infty < t < \infty\}$ and $\{D_t; -\infty < t < \infty\}$ are independent. Throughout this paper we assume Z_t and D_t are Poisson distributed for succinctness, however, it is straightforward in principle to consider alternative integer valued distributions.

2.2 Data

We naturally restrict attention to non-negative integer valued time series data, typically this will take the form of count data. We shall assume that we observe part of the time series and that the data is complete for the section of the time series observed. However, the methodology outlined in Section 3 can easily be extended to incorporate missing data, by considering the missing data as extra parameters which are to be imputed within the model. For the INARMA(p, q) process we assume that the data (observed time-series) are $\mathbf{x} = (x_{1-r}, x_{2-r}, \dots, x_n)$ where $r = \max\{p, q\}$ for some $n \geq 1$. For the INARMA(p, q) with additive noise, the data comprises $\mathbf{w} = (w_{1-r}, w_{2-r}, \dots, w_n)$ for some $n \geq 1$. Without loss of generality, we shall assume that $n > r$.

3 Likelihood and MCMC algorithms

3.1 Likelihood of INARMA(p, q) process

For conciseness, throughout this section we shall assume that $Z \sim Po(\lambda)$. Also we assume that the orders p and q of INARMA(p, q) are known with $p, q > 0$. (The procedure simplifies considerably if either $p = 0$ or $q = 0$.) Thus, we have an observed time series $\mathbf{x} = (x_{1-r}, x_{2-r}, \dots, x_n)$ where $r = \max\{p, q\}$. We are then interested in inference for the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and λ which are assumed to be unknown. Inference for the m-step ahead predictive distribution, $\mathbf{X}_m^{pred} = (X_{n+1}, X_{n+2}, \dots, X_{n+m})$ then follows trivially.

In order to facilitate inference for the INARMA(p,q), it is necessary to augment the data as follows. 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. Thus for $t \in \mathbb{Z}$, we have that

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

For $t \in \mathbb{Z}$, let $\mathbf{Y}_t = (Y_{t,1}, Y_{t,2}, \ldots, Y_{t,p})$ with $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n)$, let $\mathbf{V}_t = (V_{t,1}, V_{t,2}, \ldots, V_{t,q})$ with $\mathbf{V} = (\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_n)$ and let $\mathbf{Z}_t = (Z_1, Z_2, \ldots, Z_t)$. For $t \geq 1$, let $\mathbf{y}_t = (y_{t,1}, y_{t,2}, \ldots, y_{t,p})$ and $\mathbf{v}_t = (v_{t,1}, v_{t,2}, \ldots, v_{t,q})$ with $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)$ and $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n)$. For $q \geq 1$, let $\mathbf{z}_{IN} = (z_{1-q}, z_{2-q}, \ldots, z_0)$, thus $\mathbf{z}_{IN}$ represents the initial values of Z, and for $t \geq 1$, let $\mathbf{z}_t = (z_1, z_2, \ldots, z_t)$ with $\mathbf{z}_0$ corresponding to the empty set.

Note that given $(\mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)$ each of the components $(\mathbf{V}_t, \mathbf{Y}_t, Z_t)$ are independent, where $\mathbf{x}_s = (x_{1-r}, x_{2-r}, \ldots, x_s)$ $(s \geq 0)$. Thus for $t \geq 1$,

$$\begin{aligned}
& f_t(\mathbf{v}_t, \mathbf{y}_t, z_t | \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \\
= \; & f_t(z_t | \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \times \left\{ \prod_{j=1}^{q} f_t(v_{t,j} | \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \right\} \\
& \times \left\{ \prod_{i=1}^{p} f_t(y_{t,i} | \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \right\} \\
= \; & \frac{\lambda^{z_t}}{z_t!} e^{-\lambda} \times \prod_{i=1}^{p} \left\{ \binom{x_{t-i}}{y_{t,i}} \alpha_i^{y_{t,i}} (1-\alpha_i)^{x_{t-i}-y_{t,i}} \right\} \times \prod_{j=1}^{q} \left\{ \binom{z_{t-j}}{v_{t,j}} \beta_j^{v_{t,j}} (1-\beta_j)^{z_{t-j}-v_{t,j}} \right\}.
\end{aligned} \tag{3.1}$$

Since by Bayes' Theorem

$$\begin{aligned}
& f_t(\mathbf{v}_t, \mathbf{y}_t, z_t | x_t, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \\
= \; & \frac{f_t(x_t | \mathbf{v}_t, \mathbf{y}_t, z_t, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)}{f_t(x_t | \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)} f_t(\mathbf{v}_t, \mathbf{y}_t, z_t | \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \mathbf{x}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda),
\end{aligned} \tag{3.2}$$

it follows from (3.1) and (3.2) that for $t \geq 1$,

$$\begin{aligned}
& f_t(\mathbf{v}_t, \mathbf{y}_t, z_t | \mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \\
\propto \; & \frac{\lambda^{z_t}}{z_t!} e^{-\lambda} \times \prod_{i=1}^{p} \left\{ \binom{x_{t-i}}{y_{t,i}} \alpha_i^{y_{t,i}} (1-\alpha_i)^{x_{t-i}-y_{t,i}} \right\} \times \prod_{j=1}^{q} \left\{ \binom{z_{t-j}}{v_{t,j}} \beta_j^{v_{t,j}} (1-\beta_j)^{z_{t-j}-v_{t,j}} \right\},
\end{aligned}$$

subject to the constraints $z_t + \sum_{i=1}^{p} y_{t,i} + \sum_{j=1}^{q} v_{t,j} = x_t$, $v_{t,j} \leq z_{t-j}$ $(1 \leq j \leq q)$ and $y_{t,i} \leq x_{t-i}$ $(1 \leq i \leq p)$. Otherwise $f_t(\mathbf{v}_t, \mathbf{y}_t, z_t | \mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) = 0$. Thus,

$$f(\mathbf{v}, \mathbf{y}, \mathbf{z}_n | \mathbf{x}, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) = \prod_{t=1}^{n} f_t(\mathbf{v}_t, \mathbf{y}_t, z_t | \mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda).$$

However, the parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, λ and $\mathbf{z}_{IN}$ are unknown. Thus in a Bayesian framework it is necessary to assign priors to each of these parameters. The parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and λ are assigned the following natural, independent priors;

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

Therefore we have uninformative, uniform priors over the permissable parameter values for the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and a conjugate Gamma prior for λ. For $1 - r \leq t \leq 0$, we take $\pi(z_t|\lambda, \mathbf{x}) \sim Po(\lambda)$ subject to the constraint that $z_t \leq x_t$ since *a priori* Z_t is Poisson distributed with mean λ, but also by definition $z_t \leq x_t$.

Thus, the full joint likelihood of $(\mathbf{V}, \mathbf{Y}, \mathbf{Z}_n, \mathbf{Z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)$ given the data $\mathbf{x}$, is given by,

$$
f(\mathbf{v}, \mathbf{y}, \mathbf{z}_n, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda|\mathbf{x}) = f(\mathbf{v}, \mathbf{y}, \mathbf{z}_n|\mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda, \mathbf{x})\pi(\boldsymbol{\alpha})\pi(\boldsymbol{\beta})\pi(\mathbf{z}_{IN}|\lambda, \mathbf{x})\pi(\lambda).
$$

Therefore it follows that

$$
\begin{aligned}
f(\mathbf{v}, \mathbf{y}, \mathbf{z}, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda|\mathbf{x}) \quad \propto \quad & \prod_{t=1}^{n} \left\{ \frac{\lambda^{z_t}}{z_t!} e^{-\lambda} \prod_{i=1}^{p} \left\{ \binom{x_{t-i}}{y_{t,i}} \alpha_i^{y_{t,i}} (1-\alpha_i)^{x_{t-i}-y_{t,i}} \right\} \right. \\
\times \quad & \left. \prod_{j=1}^{q} \left\{ \binom{z_{t-j}}{v_{t,j}} \beta_j^{v_{t,j}} (1-\beta_j)^{z_{t-j}-v_{t,j}} \right\} \right\} \lambda^{A_\lambda} e^{-B_\lambda \lambda} \prod_{i=1}^{r} \left\{ \frac{\lambda^{z_{1-i}}}{z_{1-i}!} e^{-\lambda} \right\} \quad (3.3)
\end{aligned}
$$

subject to the aforementioned constraints. The following posterior distributions are then readily obtained from (3.3):

$$
\pi(\alpha_i|\mathbf{v}, \mathbf{y}, \mathbf{z}_n, \mathbf{z}_{IN}, \boldsymbol{\alpha}_{i-}, \boldsymbol{\beta}, \lambda, \mathbf{x}) \sim Beta\left(1 + \sum_{t=1}^{n} y_{t,i}, 1 + \sum_{t=1}^{n} (x_{t-i} - y_{t,i})\right) \qquad (1 \leq i \leq p), \qquad (3.4)
$$

where $\boldsymbol{\alpha}_{i-} = (\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots \alpha_p)$ and subject to the constraint that $\sum \alpha_k < 1$;

$$
\pi(\beta_j|\mathbf{v}, \mathbf{y}, \mathbf{z}_n, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}_{j-}, \lambda, \mathbf{x}) \sim Beta\left(1 + \sum_{t=1}^{n} v_{t,j}, 1 + \sum_{t=1}^{n} (z_{t-j} - v_{t,j})\right) \qquad (1 \leq j \leq q), \qquad (3.5)
$$

where $\boldsymbol{\beta}_{j-} = (\beta_1, \ldots, \beta_{j-1}, \beta_{j+1}, \ldots \beta_q)$ and subject to the constraint that $\sum \beta_l < 1$; and

$$
\pi(\lambda|\mathbf{v}, \mathbf{y}, \mathbf{z}_n, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{x}) \sim Gam\left(A_\lambda + \sum_{t=1-q}^{n} z_t, B_\lambda + n + q\right). \qquad (3.6)
$$

3.2 MCMC algorithm

We are now in a position to outline the MCMC algorithm.

1. *Update* $\boldsymbol{\alpha}$: One component at a time, either in a fixed or random order. We sample α_i using the following Gibbs/rejection sampling procedure:

   (i) Draw a proposed value for α_i from the Beta distribution given in (3.4);

   (ii) If α_i is drawn such that $\sum \alpha_k < 1$, accept the proposed value of α_i, and update the value of α_i. Otherwise repeat step (i).

2. *Update* $\boldsymbol{\beta}$: One component at a time, either in a fixed or random order. We sample β_j using the following Gibbs/rejection sampling procedure:

   (i) Draw a proposed value for β_j from the Beta distribution given in (3.5);

   (ii) If β_j is drawn such that $\sum \beta_l < 1$, accept the proposed value of β_j, and update the value of β_j. Otherwise repeat step (i).

3. *Update* λ: Sample λ from its conditional distribution given by (3.6).

4. *Update* $(\mathbf{v}, \mathbf{y}, \mathbf{z}_n)$: One set of components $(\mathbf{v}_t, \mathbf{y}_t, z_t)$ at a time, either in a fixed or random order. We use the following independence proposal Metropolis-Hastings procedure:

   (i) For $1 \leq i \leq p$, draw $y'_{t,i}$ from $Bin(x_{t-i}, \alpha_i)$;

   (ii) For $1 \leq j \leq q$, draw $v'_{t,j}$ from $Bin(z_{t-j}, \beta_j)$;

   (iii) If $x_t < \sum_{j=1}^{q} v'_{t,j} + \sum_{i=1}^{p} y'_{t,i}$, then repeat steps ($i$) and ($ii$);

   (iv) Set $z'_t = x_t - \left(\sum_{j=1}^{q} v'_{t,j} + \sum_{i=1}^{p} y'_{t,i}\right)$;

   Thus the proposal distribution for $(\mathbf{v}'_t, \mathbf{y}'_t, z'_t)$ is independent of $(\mathbf{v}_t, \mathbf{y}_t, z_t)$ and depends only upon $(\mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta})$. Therefore let $q_t(\mathbf{v}'_t, \mathbf{y}'_t, z'_t|\mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ denote the conditional proposal probability for $(\mathbf{v}'_t, \mathbf{y}'_t, z'_t)$ given $(\mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta})$.

   (v) Calculate the acceptance probability, A, for the move:

$$A = 1 \wedge \frac{f(\mathbf{v}', \mathbf{y}', \mathbf{z}'_n|\mathbf{x}, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)}{f(\mathbf{v}, \mathbf{y}, \mathbf{z}_n|\mathbf{x}, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)} \times \frac{q_t(\mathbf{v}_t, \mathbf{y}_t, z_t|\mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{q_t(\mathbf{v}'_t, \mathbf{y}'_t, z'_t|\mathbf{x}, \mathbf{z}_{IN}, \mathbf{z}_{t-1}, \boldsymbol{\alpha}, \boldsymbol{\beta})} \tag{3.7}$$

   Note that for $s \neq t$, $(\mathbf{v}'_s, \mathbf{y}'_s, z'_s) = (\mathbf{v}_s, \mathbf{y}_s, z_s)$. Thus the proposal distribution has been chosen in steps (i) and (ii) such that most of the terms on the right-hand side of (3.7) cancel, and so, we have that

$$A = 1 \wedge \frac{z_t!}{z_t'!} \lambda^{z_t' - z_t} B_t,$$

where $B_n = 1$ and for $1 \leq t \leq n-1$,

$$B_t = \begin{cases} \prod_{j=1}^{q \wedge (n-t)} \left\{ \frac{z_t'!(z_t - v_{t+j,j})!}{z_t!(z_t' - v_{t+j,j}')!} (1-\beta_j)^{z_t' - z_t} \right\} & \text{if } z_t' \geq v_{t+j,j} \ (1 \leq j \leq q \wedge (n-t)), \\ 0 & \text{otherwise.} \end{cases}$$

Therefore if the move is accepted, set $(\mathbf{v}_t, \mathbf{y}_t, z_t)$ equal to $(\mathbf{v}_t', \mathbf{y}_t', z_t')$, otherwise leave $(\mathbf{v}_t, \mathbf{y}_t, z_t)$ unchanged.

5. *Update* $\mathbf{z}_{IN}$: One component at a time, either in a fixed or random order. The procedure to update z_t $(1-q \leq t \leq 0)$ is as follows:

   (i) Draw z_t' from $Po(\lambda)$;

   (ii) If $z_t' > x_t$, repeat step (i);

   Thus the proposal probability is

$$q(z_t'|\lambda, \mathbf{x}) = \frac{\pi(z_t'|\lambda)}{F_\lambda(x_t)}$$

   where $F_\lambda(\cdot)$ denotes the cumulative distribution function of a Poisson random variable with mean λ.

   (iii) Calculate the acceptance probability, A, for the move:

$$\begin{aligned} A &= 1 \wedge \frac{f(\mathbf{v}, \mathbf{y}, \mathbf{z}_n | \mathbf{x}, \mathbf{z}_{IN}', \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \pi(z_t'|\lambda)}{f(\mathbf{v}, \mathbf{y}, \mathbf{z}_n | \mathbf{x}, \mathbf{z}_{IN}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda) \pi(z_t|\lambda)} \times \frac{q(z_t|\lambda, \mathbf{x})}{q(z_t'|\lambda, \mathbf{x})} \\ &= \begin{cases} 1 \wedge \prod_{j=1-t}^{q} \left\{ \frac{z_t'!(z_t - v_{t+j,j})!}{z_t!(z_t' - v_{t+j,j}')!} (1-\beta_j)^{z_t' - z_t} \right\} & \text{if } z_t' \geq v_{t+j,j} \ (1-t \leq j \leq q), \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Thus we have outlined an MCMC algorithm to obtain samples from the posterior distributions of both the model parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda)$ and the augmented data $(\mathbf{v}, \mathbf{y}, \mathbf{z}_n, \mathbf{z}_{IN})$. Typically, we restrict attention to the posterior distribution of the model parameters $\pi(\boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda|\mathbf{x})$ which is easily obtained from the above algorithm.

In the analysis of time series data, we are often interested in the predictive capabilities of the model. In a Bayesian context, we are interested in the m-step ahead predictive distribution, $\mathbf{X}_m^{pred}$, that is, $\pi(\mathbf{x}_m^{pred}|\mathbf{x})$ where $\mathbf{x}_m^{pred} = (x_{n+1}, x_{n+2}, \ldots, x_{n+m})$. Clearly,

$$\begin{aligned} \pi(\mathbf{x}_m^{pred}|\mathbf{x}) &= \int_{\mathbf{0}}^{\mathbf{1}} \int_{\mathbf{0}}^{\mathbf{1}} \int_0^\infty \sum_{\mathbf{k}} \sum_{\mathbf{l}} \pi(\mathbf{x}_m^{pred}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda, \mathbf{z}_{IN} = \mathbf{k}, \mathbf{z}_n = \mathbf{l}, \mathbf{x}) \\ &\quad \times \pi(\boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda, \mathbf{z}_{IN} = \mathbf{k}, \mathbf{z}_n = \mathbf{l}|\mathbf{x})\, d\lambda\, d\boldsymbol{\beta}\, d\boldsymbol{\alpha}. \end{aligned}$$

Thus to obtain samples from $\mathbf{X}_m^{pred}$, the MCMC algorithm can easily be extended as follows.

6. *Update* $\mathbf{X}_m^{pred}$: One component at a time, sequentially from x_{n+1} through to x_{n+m}. For $1 \leq k \leq m$, draw $z_{n+k} \sim Po(\lambda)$, $y_{n+k,i} \sim Bin(x_{n+k-i}, \alpha_i)$ $(1 \leq i \leq p)$ and $v_{n+k,j} \sim Bin(z_{n+k-j}, \beta_j)$ $(1 \leq j \leq q)$. Then set,

$$x_{n+k} = z_{n+k} + \sum_{i=1}^{p} y_{n+k,i} + \sum_{j=1}^{q} v_{n+k,j}.$$

   Thus we have sampled (z_{n+k}, x_{n+k}) from its conditional distribution,

$$\pi(z_{n+k}, x_{n+k} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \lambda, \mathbf{z}_n, \mathbf{z}_{IN}, \mathbf{x}, \mathbf{x}_{k-1}^{pred}).$$

Therefore from step 6 of the algorithm it is trivial to obtain a sample form the posterior, predictive distribution, $\mathbf{X}_m^{pred}$, $\pi(\mathbf{x}_m^{pred}|\mathbf{x})$.

3.3 INAR(p) + Noise

We restrict attention to INAR(p) processes with Poisson noise, $D_t \sim Po(\mu)$, for ease of exposition. The major difference from the standard INAR(p) process is that the output $\mathbf{x} = (x_{1-p}, x_{2-p}, \ldots, x_n)$ from the INAR process is unobserved, thus we augment the data by including $\mathbf{x}$. Note that given $\mathbf{x}$, we obtain $\mathbf{d} = (d_{1-p}, d_{2-p}, \ldots, d_n)$ since $w_t = x_t + d_t$.

Suppose that $\boldsymbol{\alpha}$, λ, μ and $\mathbf{x}$ are known. Then, for $t \geq 1$,

$$f_t(d_t, \mathbf{y}_t, z_t | \mathbf{w}, \mathbf{x}, \boldsymbol{\alpha}, \lambda, \mu) \propto \frac{\lambda^{z_t}}{z_t!} e^{-\lambda} \frac{\mu^{d_t}}{d_t!} e^{-\mu} \prod_{i=1}^{p} \binom{x_{t-i}}{y_{t,i}} \alpha_i^{y_{t,i}} (1-\alpha_i)^{x_{t-i}-y_{t,i}}.$$

subject to the constraints $d_t = w_t - x_t$, $\sum_{i=1}^{p} y_{t,i} + z_t = x_t$ and $y_{t,i} \leq x_{t-i}$ $(1 \leq i \leq p)$. Otherwise $f(d_t, \mathbf{y}_t, z_t | \mathbf{w}, \mathbf{x}, \boldsymbol{\alpha}, \lambda, \mu) = 0$. Thus

$$f(\mathbf{d}, \mathbf{y}, \mathbf{z}_n | \mathbf{w}, \mathbf{x}, \boldsymbol{\alpha}, \lambda, \mu) = \prod_{t=1}^{n} f_t(d_t, \mathbf{y}_t, z_t | \mathbf{w}, \mathbf{x}, \boldsymbol{\alpha}, \lambda, \mu),$$

For $\boldsymbol{\alpha}$ and λ we use the same prior as in Section 3.1. However, we now also require a prior for μ. For conjugacy, we set $\pi(\mu) \sim Gam(A_\mu, B_\mu)$ for some $A_\mu, B_\mu > 0$. Therefore the posterior distribution for μ is:

$$\pi(\mu | \mathbf{w}, \mathbf{d}, \mathbf{x}, \mathbf{y}, \mathbf{z}_n, \boldsymbol{\alpha}, \lambda) \sim Gam\left(A_\mu + \sum_{t=1-p}^{n} d_t, B_\mu + n + p\right). \tag{3.8}$$

The MCMC algorithm described in Section 3.2 can easily be adapted to include the noise term. The parameters $(\boldsymbol{\alpha}, \lambda, \mathbf{y}, \mathbf{z}_n)$ are updated exactly as in Section 3.2 using the augmented data $\mathbf{x}$ (and with

$q = 0$). Thus all that is required is to outline the procedure for updating the parameters $(\mathbf{x}, \mathbf{d}, \mu)$. For simplicity in exposition, let $\mathbf{x}_n = (x_1, x_2, \ldots, x_n)$ ($\mathbf{d}_n = (d_1, d_2, \ldots, d_n)$) and $\mathbf{x}_{IN} = (x_{1-p}, x_{2-p}, \ldots, x_0)$ ($\mathbf{d}_{IN} = (d_{1-p}, d_{2-p}, \ldots, d_0)$), so that $\mathbf{x} = (\mathbf{x}_{IN}, \mathbf{x}_n)$ ($\mathbf{d} = (\mathbf{d}_{IN}, \mathbf{d}_n)$). The following steps should be appended to the MCMC algorithm in Section 3.2, most naturally after step 5.

1. *Updating* $(\mathbf{x}_n, \mathbf{d}_n, \mathbf{z}_n)$: One set of components (x_t, d_t, z_t) at a time, either in a fixed or random order. We use the following independence proposal Metropolis-Hastings procedure:

   (i) Draw d_t' from $Po(\mu)$ and set $x_t' = w_t - d_t'$;

   (ii) If $x_t' < \sum_{i=1}^{p} y_{t,i}$, then repeat step (i);

   (iii) Set $z_t' = x_t' - \sum_{i=1}^{p} y_{t,i}$;

   Note that the proposal distribution for (x_t', d_t', z_t') depends only on μ. Therefore we write $q(x_t', d_t', z_t'|\mu)$ for the proposal probability.

   (iv) Calculate the acceptance probability, A, for the move:

$$A = 1 \wedge \frac{f(\mathbf{d}', \mathbf{y}, \mathbf{z}_n'|\mathbf{w}, \mathbf{x}', \boldsymbol{\alpha}, \lambda, \mu)}{f(\mathbf{d}, \mathbf{y}, \mathbf{z}_n|\mathbf{w}, \mathbf{x}, \boldsymbol{\alpha}, \lambda, \mu)} \times \frac{q(d_t, z_t, x_t|\mu)}{q(d_t', z_t', x_t'|\mu)} \tag{3.9}$$

   Again the proposal has been chosen such that most of the terms on the right-hand side of (3.9) cancel, leaving

$$A = 1 \wedge \frac{z_t!}{z_t'!} \lambda^{z_t' - z_t} B_t$$

   where $B_n = 1$ and for $1 \leq t \leq n-1$,

$$B_t = \begin{cases} \prod_{i=1}^{p \wedge (n-t)} \left\{ \frac{x_t'!(x_t - y_{t+i,i})!}{x_t!(x_t' - y_{t+i,i})!} (1-\alpha_i)^{x_t' - x_t} \right\} & \text{if } x_t' \geq y_{t+i,i} \ (1 \leq i \leq p \wedge (n-t)); \\ 0 & \text{otherwise.} \end{cases}$$

   Therefore if the move is accepted, set (x_t, d_t, z_t) equal to (x_t', d_t', z_t'), otherwise leave (x_t, d_t, z_t) unchanged.

2. *Updating* $(\mathbf{x}_{IN}, \mathbf{d}_{IN})$: One set of components (x_t, d_t) at a time, either in a fixed or random order. The procedure is similar to that used for $(\mathbf{x}_n, \mathbf{d}_n, \mathbf{z}_n)$ and is as follows:

   (i) Draw d_t' from $Po(\mu)$ and set $x_t' = w_t - d_t'$;

   (ii) If $x_t' < 0$, then repeat step (i);

   Let $q(d_t', x_t'|\mu)$ denote the proposal distribution.

(iii) Calculate the acceptance probability, A, for the move:

$$A = 1 \wedge \frac{f(\mathbf{d}', \mathbf{y}, \mathbf{z}_n | \mathbf{w}, \mathbf{x}', \boldsymbol{\alpha}, \lambda, \mu)}{f(\mathbf{d}, \mathbf{y}, \mathbf{z}_n | \mathbf{w}, \mathbf{x}, \boldsymbol{\alpha}, \lambda, \mu)} \times \frac{q(d_t, x_t | \mu)}{q(d'_t, x'_t | \mu)}$$

This reduces to

$$A = \begin{cases} 1 \wedge \prod_{i=1-t}^{p} \left\{ \frac{x'_t!(x_t - y_{t+i,i})!}{x_t!(x'_t - y_{t+i,i})!} (1-\alpha_i)^{x'_t - x_t} \right\} & \text{if } x'_t \geq y_{t+i,i} \ (1-t \leq i \leq p); \\ 0 & \text{otherwise.} \end{cases}$$

Therefore if the move is accepted, set (x_t, d_t) equal to (x'_t, d'_t), otherwise leave (x_t, d_t) unchanged.

3. *Update* μ: Sample μ from its conditional distribution given by (3.8).

Then obtaining samples from the posterior predictive distributions of $\mathbf{W}_m^{pred} = (W_{n+1}, W_{n+2}, \ldots, W_{n+m})$ or $\mathbf{X}_m^{pred}$ or both is straightforward, and involves adding the following step to the above MCMC algorithm.

4. *Update* $(\mathbf{x}_m^{pred}, \mathbf{w}_m^{pred})$: One pair of components at a time, sequentially from (x_{n+1}, w_{n+1}) through to (x_{n+m}, w_{n+m}). Then for $1 \leq k \leq m$, draw $z_{n+k} \sim Po(\lambda)$, $y_{n+k,i} \sim Bin(x_{n+k-i}, \alpha_i)$ $(1 \leq i \leq p)$ and $d_{n+k} \sim Po(\mu)$. Finally, set

$$(x_{n+k}, w_{n+k}) = \left(z_{n+k} + \sum_{i=1}^{p} y_{n+k,i}, z_{n+k} + \sum_{i=1}^{p} y_{n+k,i} + d_{n+k} \right).$$

4 Results

4.1 Simulation study

In order to assess the strengths and weaknesses of the MCMC algorithms outlined in Section 3, we conducted an extensive simulation study. A representative sample from the simulation study is given in Table 1 along with the true model parameters.

Model	n	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	λ	Noise
$AR(3)$	500	$(0.2, 0.3, 0.15)$	$-$	2	$-$
$MA(3)$	500	$-$	$(0.2, 0.3, 0.15)$	2	$-$
$ARMA(2,2)$	500	$(0.4, 0.2)$	$(0.4, 0.2)$	2	$-$
$AR(3)$ + Noise	500	$(0.2, 0.3, 0.15)$	$-$	2	$Po(1)$

Table 1. A selection of models from the simulation study.

For each data set, the MCMC algorithm was run to obtain a sample of size 1000000 following a burn-in period of 10000 iterations. The burn-in period is essential to ensure that the estimates of the posterior distribution are unaffected by the choice of starting values of the Markov chain. The priors were chosen as outlined in Section 3, with $\pi(\lambda) \sim Gam(1,1)$ and $\pi(\mu) \sim Gam(1,1)$.

MCMC methods generate dependent samples from the posterior distribution of interest. Therefore it is essential that a large enough sample is obtained to offset the loss in information (when compared with an independent and identically distributed sample from the posterior distribution) due to having a dependent sample. Formal investigation of the performance of the MCMC algorithms, if so desired, can be done using the integrated autocorrelation function or variance of batch means, see Geyer (1992) for details. However, visual inspection of the time series plots of the MCMC sample is usually sufficient to detect that the Markov chain is sampling from its stationary distribution (the posterior distribution). Also the time series plots allow us to determine how well the Markov chain is *mixing*, with better mixing corresponding to less dependence between successive observations of the posterior distribution, and hence, reduced Monte Carlo error. In Figure 1, time series plots of the first 10000 observations for λ from the MCMC sample for each of the four data sets are plotted.

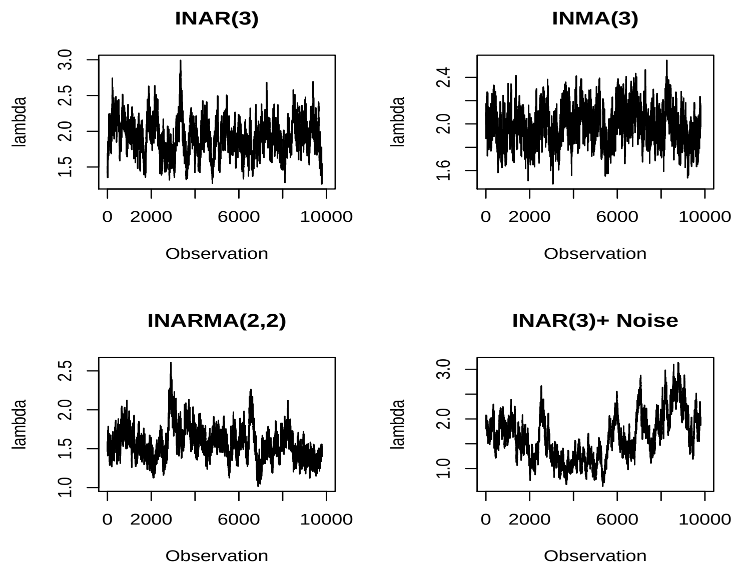

Figure 1. Time series plots for the first 10000 observations of λ, for the $INAR(3)$, $INMA(3)$, $INARMA(2,2)$ and $INAR(3) + Noise$ data sets, respectively.

The mixing of the Markov chain (time series plot) with respect to λ is indicative of the MCMC algorithms performance with respect to the other parameters. The algorithms are shown to perform well for the INAR and INMA data sets and not so well for the INARMA and INAR + Noise data sets. The performance of the MCMC algorithm is closely related to the information content contained within the data about the various parameters. That is, the more informative the data are about the underlying model parameters, the better the MCMC algorithms performs. Thus the MCMC algorithm performs very well for INAR(p) processes since the data $(x_{1-p}, x_{2-p}, \dots, x_n)$ is very informative about the parameters driving the process. However, the MCMC algorithm performs as well for the INMA(q) process (or even better in the above example) since although the key underlying process $(z_{1-q}, z_{2-q}, \dots, z_n)$ is unobserved and needs to be imputed, the data is very informative about $(z_{1-q}, z_{2-q}, \dots, z_n)$. This is most clearly seen in Table 2 below which lists the estimated mean and standard deviations of the posterior distributions based on the samples obtained from the MCMC algorithms.

Model		$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	λ	μ
$AR(3)$	Mean	$(0.2096, 0.3180, 0.1064)$	–	1.926	–
	St. Dev.	$(0.0442, 0.0405, 0.0431)$	–	0.234	–
$MA(3)$	Mean	–	$(0.1717, 0.3065, 0.1588)$	2.016	–
	St. Dev.	–	$(0.0552, 0.0621, 0.0592)$	0.153	–
$ARMA(2)$	Mean	$(0.3524, 0.3096)$	$(0.5966, 0.1877)$	1.561	–
	St. Dev.	$(0.0652, 0.0589)$	$(0.206, 0.135)$	0.227	–
$AR(3) + \text{Noise}$	Mean	$(0.1759, 0.3923, 0.1342)$	–	1.416	1.887
	St. Dev.	$(0.0676, 0.0741, 0.0686)$	–	0.491	0.808

Table 2. Estimated mean and standard deviations of parameter posterior distributions.

For the INARMA(2,2) data set, the data are more informative about the AR parameters than the MA parameters. This is generally the case observed for INARMA(p, q) data sets. This is clearly seen in the estimated variances of the parameter posterior distributions and also by observing the mixing of the Markov chain for the AR and MA parameters.

The presence of a noise parameter also affects the MCMC algorithm as one would expect. As in Section 3.3, we have restricted attention to INAR(p) processes with noise. Although the noise and signal processes are independent, the joint posterior distributions for the parameters of the two processes are dependent. Therefore the stronger the noise is, the harder it is to detect the parameters of the signal (underlying INAR(p)) process. This leads to worse estimation of these parameters.

4.2 Real life data

The main purpose for introducing the MCMC algorithms in Section 3 was to apply the methodology to real life data. In this section, we apply the methodology of Section 3 to two real life examples Westgren's gold particle data set, see Westgren (1916) and Jung and Tremayne (2006) and a Schizophrenic patients test scores, see McCleary and Hay (1980).

4.2.1 Westgren's Gold Particle data set

This data set consists of 380 counts of gold particles in a solution at equidistant points in time. The data was originally published in Westgren (1916) but has recently been analysed in Jung and Tremayne (2006) using method of moments estimation of an INAR(2) model. Jung and Tremayne (2006) claim that the INAR(2) is an adequate model for this data basing their analysis on the first 370 observations with the remaining 10 observations used for predictive purposes. Recent extensions of this paper incorporating order determination into the MCMC algorithm suggest that the INAR(2) model is indeed the most suitable INAR model for this data, see Enciso-Mora *et al.* (2006) for more details.

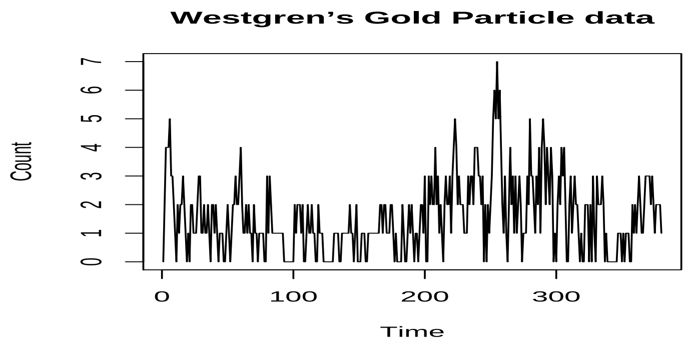

Figure 2. Westgren's gold particle data set.

Our aim here is to compare our approach with Jung and Tremayne (2006). In particular, we fit an INAR(2) model to the first 370 observations and estimate the model parameters for the INAR(2) model. We then obtain the (joint-)predictive distribution for the remaining 10 observations. In Jung and Tremayne (2006) two INAR(2) models are considered; these are the INAR(2)-AA model (Alzaid and Al-Osh (1990)) and INAR(2)-DL model (Du and Li (1991)). We have focussed upon the Du and Li model, although in principle the MCMC methodology can be adapted to the Alzaid and Al-Osh model.

The MCMC algorithm was run in order to obtain samples of size 50000 from the posterior parameter distribution with a burn-in period of 10000 iterations. These samples were then used to simulate 50000 realisations of the data points $371, 372, \ldots, 380$. The estimated posterior means and standard deviations for the model parameters are given in the table below. These results are comparable with the parameter estimates and bootstrap standard errors obtained in Jung and Tremayne (2006).

Parameter	α_1	α_2	λ
Mean	0.463	0.187	0.550
St. dev.	0.0475	0.0540	0.0719

Table 3. Estimates of INAR(2) parameters for Westgren's data set.

For the (joint-)predictive distribution of the last 10 data points, we obtained identical results to Jung and Tremayne (2006) for the median predicted values. Below are histograms of the estimated posterior densities of data points 371, 372 and 380 along with a histogram of the empirical distribution of Westgren's data. It is worth noting the close agreement between the predictive distribution for data point 380 and the empirical distribution of the observed data.

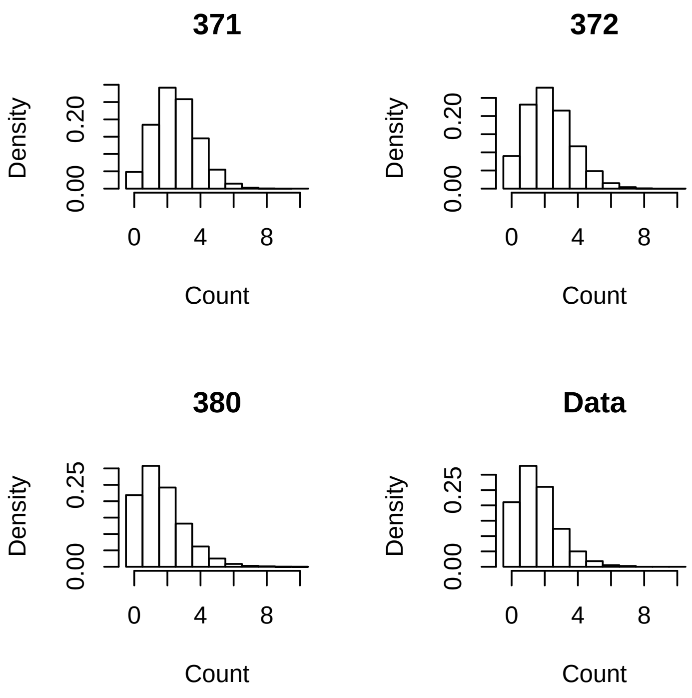

Figure 3. Histograms of predictive distributions of data points 371, 372 and 380 and the empirical distribution of Westgren's data.

4.2.2 Schizophrenic patient

The second example concerns the daily observations of the score achieved by a schizophrenic patient on a test of perceptual speed, see, McCleary and Hay (1980). The data consists of 120 consecutive daily scores. However, from the 61^{st} day onwards the patient began receiving a powerful tranquilizer (chloropromazine) that could be expected to reduce perceptual speed. The data are presented in Figure 4.

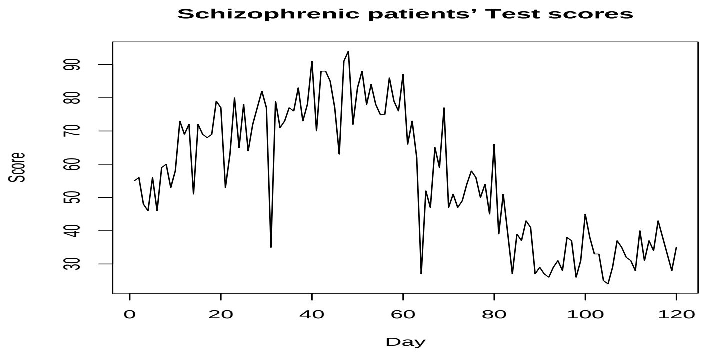

Figure 4. A time-series plot of the daily test score achieved by schizophrenic patient.

The data clearly shows a reduction in the patients' test score after day 61. Therefore we propose to model the patients' test score using an INAR(1) process with different parameters (α_1, λ) for before and after receiving chloropromazine. Thus the model we used is:

$$X_t = \begin{cases} \alpha_1^B \circ X_{t-1} + Z_t & \text{if } t \leq 60 \\ \alpha_1^A \circ X_{t-1} + Z_t & \text{if } t \geq 61 \end{cases}$$

where Z_t $(1 \leq t \leq 120)$ are independent and for $t \leq 60$, $Z_t \sim Po(\lambda^B)$ and for $t \geq 61$, $Z_t \sim Po(\lambda^A)$. Thus B and A denote *before* and *after* treatment, respectively.

Adapting the likelihood and consequently the MCMC algorithm in Section 3 to incorporate a change point where the parameter values change is trivial especially in this example where the change point is known. However, the methodology could also be extended to situations where the position and number of change points is unknown.

The MCMC algorithm was run in order to obtain samples of size 50000 from the posterior parameter distribution with a burn-in period of 10000 iterations. The main aim in this example is to study the affects of chloropromazine on the parameters (α_1, λ), however, the posterior predictive distribution can also be obtained as before.

The samples from α_1^B and α_1^A produced indistinguishable posterior density plots. Thus suggesting that the parameter α_1 is unaffected by the introduction of treatment and that the data should be modelled with $\alpha_1^B = \alpha_1^A = \alpha_1$. Therefore the MCMC algorithm was run as before except that a common, α_1 parameter, for both before and after treatment was used. Posterior density plots for the model parameters from the two algorithms are given in Figure 5.

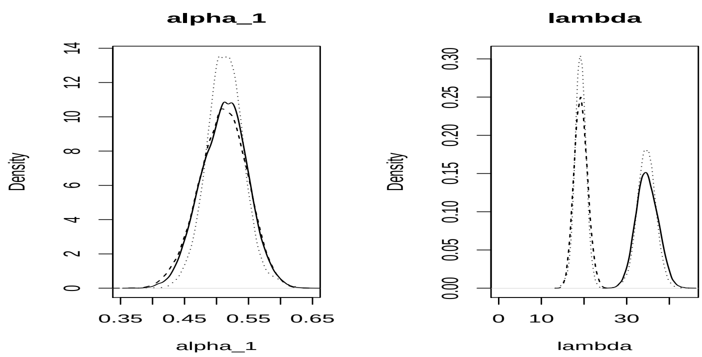

Figure 5. Posterior density plots for the INAR(1) parameters for the test score, before treatment (solid lines) and after treatment (dashed lines) in the case ($\alpha_1^A \neq \alpha_1^B$) and the corresponding parameters (dotted lines) in the case ($\alpha_1^A = \alpha_1^B = \alpha_1$).

The two MCMC algorithms produce indistinguishable results in terms of posterior means for α_1, λ^B and λ^A. Since a common α_1 parameter can be assumed, the reduction in test scores is therefore accounted for by a reduction in the λ parameter. The posterior means of λ^B and λ^A are 34.7 (34.8) and 19.2 (19.3), respectively, when a common α_1 parameter is used ($\alpha_1^B \neq \alpha_1^A$).

Note that the posterior distribution for α_1 ($\alpha_1 = \alpha_1^A = \alpha_1^B$) is more compact than the individual posterior distributions for α_1^B and α_1^A ($\alpha_1^A \neq \alpha_1^B$). This is since inference for α_1 is based on time-series data of length 120 whereas inference for α_1^B and α_1^A time-series data of length 60. This has a knock-on effect of producing more compact (and peaked) posterior distributions for λ^B and λ^A in the case of a common α_1 parameter.

5 Conclusions

This paper has introduced an efficient MCMC mechanism for conducting inference for INARMA(p, q) processes where p and q are both known. We have been able to do inference for models which would be very tricky using conventional methods but are very straightforward using MCMC. The methodology introduced here, as previously mentioned, can readily be extended to incorporate noise, change points and more general, generalised Steutal and van Harn operators. Thus as a consequence this paper only scratches the surface of the potential applications of MCMC to integer valued time-series.

Some very natural extensions of the current work present themselves. Firstly, developing inference for INARMA(p, q) processes where the parameters p and q are unknown. This is the subject of ongoing research where a reversible jump (RJ) MCMC algorithm has successfully been constructed, see Enciso-Mora *et al.* (2006) for details. RJMCMC could also prove useful in extensions of Section 4.2.2 where both the number and location of parameter change points are unknown. Throughout this paper we have assumed that the time-series under consideration is stationary. However, this is not necessary for the above methodology and we could easily adapt the MCMC algorithm to allow for, for example, seasonality or a linear trend. Finally it would be interesting to study multivariate INARMA(p, q) processes. This should be, in principle at least, a fairly straightforward extension of the MCMC algorithms in Section 3. All these extensions should be fairly routine using MCMC algorithms but would prove extremely difficult problems to tackle using more conventional methods.

Acknowledgements

We thank Robert Jung for providing us with Westgren's gold particle data set. The schizophrenic patient data set was obtained from the Time Series Data Library:
http://www-personal.buseco.monash.edu.au/∼hyndman/TSDL/

References

Daniels, M. J. and Cressie, N. (2001) A hierarchical approach to Covariance function estimation for Time Series. *Journal of Time Series Analysis* 22, 253–266.

Davis, R. A., Dunsmuir, W. T. M. and Wang, Y. (2000) On autocorrelation in a Poisson regression model. *Biometrika* 87, 491–505.

Du, J. and Li, Y. (1991) The integer-valued autoregressive (INAR(p)) model. *Journal of Time Series Analysis* 12, 129–142.

Enciso-Mora, V., Neal, P. and Subba Rao, T. (2006) Order determination and model selection for INARMA(p, q) processes. *In preparation.*

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. pp. 310–330.

Freeland, R. K. and McCabe, B. P. M. (2004) Forecasting discrete valued low count time series. *International Journal of Forecasting* 20, 427–434.

Geyer, C.J. (1992) Practical Markov chain Monte Carlo (with discussion). *Statistical Science* 7, 473–511.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds.) (1995) *Markov chain Monte Carlo in Practice.* Chapman and Hall.

Jung, R. C. and Tremayne, A. R. (2006) Coherent forecasting in integer time series models. *International Journal of Forecasting* 22, 223–238.

Latour, A. (1997) The multivariate GINAR(p) process. *Advances in Applied Probability* 29, 228–248.

McCabe, B. P. M. and Martin, G. M. (2005) Bayesian predictions of low count time series. *International Journal of Forecasting* 22, 315–330.

McCleary, R. and Hay, R. A. (1980) *Applied Time Series Analysis for the Social Sciences.* Sage Publications.

McKenzie, E. (2003) Discrete variate time series. *Stochastic Processes: Modelling and Simulation.* (eds. D. N. Shanbhag and C. R. Rao.) Elsevier. pp. 573–606.

Steutel, F. W. and van Harn, K. (1979) Discrete analogues of self-decomposability and stability. *Annals of Probability* 7, 893–899.

Troughton, P. T. and Godsill, S. J. (1997) Bayesian model selection for time series using Markov chain Monte Carlo. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing 1997, vol. V.* 3733–3736.

Westgren, A. (1916). Die Veränderungsgeschwindigkeit der lokalen Teilchenkonzentration in kollioden Systemen (Erste Mitteilung). *Arkiv för Matematik, Astronomi och Fysik* 11, 1–24.