Bayesian analysis of order uncertainty in arima models

Article · January 2006

CITATIONS
26

READS
817

2 authors:

Ricardo S Ehlers
University of São Paulo
48 PUBLICATIONS **256** CITATIONS
SEE PROFILE

Stephen P Brooks
Select Statistics
103 PUBLICATIONS **12,637** CITATIONS
SEE PROFILE

Bayesian Analysis of Order Uncertainty in ARIMA Models

R.S. Ehlers†

Federal University of Paraná, Brazil

S.P. Brooks

University of Cambridge, UK

Summary.
In this paper we extend the work of Brooks and Ehlers (2002) and Brooks et al. (2003) by constructing efficient proposal schemes for reversible jump MCMC in the context of autoregressive moving average models. In particular, the full conditional distribution is not available for the added parameters and approximations to it are provided by suggesting an adaptive updating scheme which automatically selects proposal parameter values to improve the efficiency of between-model moves. The performance of the proposed algorithms is assessed by simulation studies and the methodology is illustrated by applying it to a real data set.

Keywords: Bayesian model selection, posterior model probability, Markov chain Monte Carlo, reversible jump MCMC, autoregressive moving average.

1. Introduction

In many applications, in addition to the estimation of model parameters, there is substantial prior uncertainty concerning the choice of models that are the most appropriate for any given data. The classical approach is to use information criteria such as the AIC (Akaike, 1974) to discriminate between competing models. In the Bayesian framework, model uncertainty can be handled in a parametric fashion by indexing all models under consideration, treating this index as another parameter and considering posterior model probabilities and/or Bayes factors (Kass and Raftery 1995). In realistically complex cases the number of competing models may be large and the corresponding analysis complex, so numerical techniques are used to efficiently explore model space.

We assume that a data vector $\boldsymbol{y}$ is observed and can be described by any of M candidate models. Associated with each model is a likelihood function $p(\boldsymbol{y}|\boldsymbol{\theta}^{(k)}, k)$ depending upon an unknown parameter vector $\boldsymbol{\theta}^{(k)}$, where $k \in \{1, \ldots, M\}$ is a model indicator determining the parameter dimension, which may vary from model to model. We shall focus upon the Bayesian framework here and assign prior distributions $p(\boldsymbol{\theta}^{(k)}|k)$ to each parameter vector and a prior distribution $p(k)$ to the model number. We are interested in computing the joint posterior distribution of all unknown quantities, i.e. the model indicator and parameters, denoted by

$$\pi(k, \boldsymbol{\theta}^{(k)}) \propto p(\boldsymbol{y}|\boldsymbol{\theta}^{(k)}, k)p(\boldsymbol{\theta}^{(k)}|k)p(k).$$

Based on a sample from this distribution, marginal posterior model probabilities can be approximated by the sample proportion of models visited. Also, samples from the posterior distribution within each model are automatically available for inference by simply conditioning on samples where the chain is in state k.

We need to simulate Markov chains whose state vector may change dimension as the simulation proceeds. An important innovation in the MCMC literature was the introduction of trans-dimensional algorithms to explore model space. Although alternatives exist (e.g. Stephens, 2000), we shall focus upon the use of the reversible jump MCMC algorithm proposed by Green (1995) for trans-dimensional transitions. We discuss the difficulty in implementing these algorithms efficiently and suggest an adaptive updating scheme which automatically selects proposal parameter values to improve the efficiency of between-model moves.

†*Address for correspondence:* Ricardo Ehlers, Departmento de Estatistica, Universidade Federal do Paraná, 81531-990, Curitiba, PR - Brazil
E-mail: ehlers@est.ufpr.br

1.1. Reversible Jump MCMC

The reversible jump algorithm (Green 1995) is a general strategy for generating samples from the joint posterior distribution $\pi(k, \boldsymbol{\theta}^{(k)})$. This is based upon the standard Metropolis-Hastings approach of proposing a move and defining a probability of accepting that move. In any typical application, both traditional MCMC for within-model moves and reversible jump updates for between-model moves will be employed in order to explore both parameter and model spaces.

There are simple versions of the reversible jump algorithm that can be applied in model discrimination problems. Suppose that the current state of the Markov chain is $(k, \boldsymbol{\theta}^{(k)})$, where $\boldsymbol{\theta}^{(k)}$ has dimension n_k, and we have defined one or more different move types allowing transitions between spaces of different dimensions. A move type r is performed with probability $p_k(r)$ by generating $\boldsymbol{u}$ from a specified proposal density $q(\cdot)$ and setting $(\boldsymbol{\theta}^{(k')}, \boldsymbol{u'}) = g(\boldsymbol{\theta}^{(k)}, \boldsymbol{u})$. Here, g is a specified invertible function and $n_k + |\boldsymbol{u}| = n_{k'} + |\boldsymbol{u'}|$ where $|\boldsymbol{u}|$ denotes the dimension of $\boldsymbol{u}$. Then, we accepted $(k', \boldsymbol{\theta}^{(k')})$ as the new state of the chain with probability $\min(1, A)$ where

$$A = \frac{\pi(k', \boldsymbol{\theta}^{(k')}) p_{k'}(r') q(\boldsymbol{u'})}{\pi(k, \boldsymbol{\theta}^{(k)}) p_k(r) q(\boldsymbol{u})} \left| \frac{\partial g(\boldsymbol{\theta}^{(k)}, \boldsymbol{u})}{\partial(\boldsymbol{\theta}^{(k)}, \boldsymbol{u})} \right| \tag{1}$$

is called the acceptance ratio.

A class of moves most commonly used for transitions between nested models consists of adding or deleting parameters from the current model to the next. In this special case, if we assume that $n_{k'} > n_k$ then $|\boldsymbol{u}| = n_{k'} - n_k$ and the transition from the larger model to the smaller one is entirely deterministic and the acceptance ratio (1) reduces to

$$A = \frac{\pi(k', \boldsymbol{\theta}^{(k')} | \boldsymbol{y}) p_{k'}(r')}{\pi(k, \boldsymbol{\theta}^{(k)} | \boldsymbol{y}) p_k(r) q(\boldsymbol{u})} \left| \frac{\partial g(\boldsymbol{\theta}^{(k)}, \boldsymbol{u})}{\partial(\boldsymbol{\theta}^{(k)}, \boldsymbol{u})} \right| \tag{2}$$

The applications in this paper will focus on a particular implementation in which the increase in dimensionality of the parameter space is of the type $\boldsymbol{\theta}^{(k')} = (\boldsymbol{\theta}^{(k)}, \boldsymbol{u})$. In this case, the Jacobian term is equal to one and the acceptance ratio (2) simplifies to

$$\begin{aligned} A &= \frac{p(\boldsymbol{y} | k', \boldsymbol{\theta}^{(k')})}{p(\boldsymbol{y} | k, \boldsymbol{\theta}^{(k)})} \frac{p(\boldsymbol{\theta}^{(k')} | k') p(k')}{p(\boldsymbol{\theta}^{(k)} | k) p(k)} \frac{p_{k'}(r')}{p_k(r) q(\boldsymbol{u})} \\ &= \text{likelihood ratio} \times \text{prior ratio} \times \text{proposal ratio} \end{aligned} \tag{3}$$

where the posterior densities have been replaced by the appropriate product of prior density and likelihood function.

The paper is organised as follows. In Section 2 we discuss the implementation of Bayesian methods for fitting ARMA models via MCMC, including model order assessment. As we shall see, estimation is complicated computationally since the inclusion of MA terms introduces complex non-linearities in the likelihood function. A detailed description of model parameterisation and updating mechanisms for within and between model moves is provided. Section 3 addresses the problem of order assessment in ARMA models when stationarity and inversibility restrictions are to be imposed. We propose a parameterisation in terms of reciprocal roots of the characteristic equations.

2. Autoregressive Moving Average Processes

Autoregressive moving average (ARMA) processes provide a very useful and parsimonious class of models for describing time series data. For a time series of equally spaced observations y_t, $(t = 1, 2, \ldots, n)$, the general Gaussian ARMA(k, q) model takes the form

$$y_t = \sum_{j=1}^{k} a_j y_{t-j} + \sum_{j=1}^{q} b_j \epsilon_{t-j} + \epsilon_t$$

where the error terms ϵ_t are *i.i.d.* $N(0, \sigma_\epsilon^2)$. ARMA-type models are also relevant for modelling volatility, for example a GARCH(k, q) model can be interpreted as an ARMA(m, k) model for ϵ_t^2

where $m = \max(k, q)$ (see Franses and van Dijk 2000). The process is stationary and inversible if the roots of both the AR and MA characteristic polynomials lie outside the unit circle. These conditions impose a set of restrictions on the ARMA coefficients which are difficult to incorporate into a prior distribution. In this section, we will then assign unconstrained prior distributions to the coefficients.

Here both k and q are unknown parameters and we use reversible jump MCMC for moving between the competing ARMA models with (k, q) playing the role of model indicator and determining the model dimension since σ_ϵ^2 will be present in every model. Upper bounds on model order are fixed *a priori* and we assume that the AR and MA coefficients are *a priori* independent and that each model is equally likely.

Though other forms are available (see for example Box and Jenkins 1976) we shall adopt the following associated likelihood approximation for the ARMA(k, q) model

$$\tilde{p}(\boldsymbol{y}|k, q, \mathbf{a}^{(k)}, \mathbf{b}^{(q)}, \sigma_\epsilon^2) = (2\pi\sigma_\epsilon^2)^{-(n-k_{\max})/2} \exp\left(-\frac{1}{2\sigma_\epsilon^2} \sum_{t=k_{\max}+1}^{n} \epsilon_t^2\right) \tag{4}$$

where $k_{\max}$ is the maximum value allowed for k. We then use the same sample size at different iterations and compute this likelihood function conditional on the first $k_{\max}$ observations. The inclusion of MA terms in the model introduces complicated non-linearities in the likelihood function as each ϵ_{t-j} depends on the whole set of coefficients in a complicated non-linear way. This complexity is inherited by the posterior distribution of the model parameters and approximation methods are necessary to gain posterior inferences.

For the purposes of this paper we use uniform priors for the model order. For fixed values of $k \in \{0, \ldots, k_{\max}\}$ and $q \in \{0, \ldots, q_{\max}\}$, each ARMA coefficient is assumed normally distributed with mean zero and variances σ_a^2 and σ_b^2 while σ_ϵ^2 is assumed inverse-Gamma distributed. Also, all model parameters are assumed to be *a priori* independent and prior inverse-Gamma distributions are specified for the hyperparameters σ_a^2 and σ_b^2. The inverse-Gamma family of prior distributions is conditionally conjugate, i.e. the full posterior conditional distribution is also inverse-Gamma. This conditional conjugacy allows that the variances are easily updated. A common choice in the literature is the non-informative (proper) prior inverse-Gamma(ϵ, ϵ) with small values for ϵ.

Alternatively, when modelling ARMA processes, it is often necessary to impose stationarity and inversibility restrictions. This can be done in several ways (see for example Chib and Greenberg 1994, Barbieri and O'Hagan 1996 and Huerta and West 1999) and we shall focus upon the method which reparameterises the model in terms of the reciprocal roots of the characteristic equations, as in Huerta and West (1999). In this case, the ARMA(k, q) model may be rewritten as

$$\prod_{i=1}^{k}(1 - \lambda_i L)y_t = \prod_{j=1}^{q}(1 - \delta_j L)\epsilon_t \quad t = 1, \ldots, n. \tag{5}$$

where L denotes the lag operator so that $L^i y_t = y_{t-i}$. The λ_i and δ_j (either real or occurring in complex conjugate pairs) are then referred to as the reciprocal roots and the process is stationary and inversible if $|\lambda_i| < 1$, $i = 1, \ldots, k$ and $|\delta_j| < 1$, $j = 1, \ldots, q$. We assume, as is commonly the case, that the roots are distinct and non-zero.

If one or more of the autoregressive roots are unity the resulting models are of great value in representing homogeneous non-stationary time series. If d of these roots are unity and the remainder lie outside the unit circle then the resulting process is called an ARIMA(k, d, q) process. Frequently, small values for k, q and d will be appropriate.

2.1. Within-Model Moves

Within each model, i.e. with k and q fixed, parameters are updated using traditional MCMC methods. In this case, it is easy to show that the full conditional distributions of the variance components remain inverse Gamma and these parameters are then updated by a Gibbs move. We note that the full conditional distribution of σ_ϵ^2 is conditional on the $k_{\max}$ first observations so that it is based on the same number of observations at each iteration.

However, full conditional distributions of standard form for the individual ARMA coefficients cannot be derived analytically. Though the complete conditional distribution of $\mathbf{a}^{(k)}$ may be obtained

analytically and is of standard form this would involve performing a computationally demanding matrix inversion at every iteration. Keeping the updating procedure simple and straightforward is a desirable computational feature, so we update the whole vector of AR coefficients sequentially via the Metropolis-Hastings algorithm by combining univariate moves. We use random walk Metropolis updates with a normal proposal density centered on the current parameter value. Similarly, the full conditional distributions for the individual MA coefficients are not of standard form and again we use random walk Metropolis updates.

2.2. Model Order Assessment

In the classical time series literature the usual approach to order determination is to fit models of different orders and decide on an adequate model order based on the residuals. Information criteria such as AIC are based on an estimate of the error variance σ_ϵ^2 conditional on k and q, and on a quantity that depends on k and q whose role is to penalise high order models. The value of (k, q) for which the criterion is minimal is chosen as the appropriate model order.

The problem of order uncertainty in pure AR and ARMA models has been addressed recently using reversible jump MCMC methods (e.g. Troughton and Godsill 1997 and Barbieri and O'Hagan 1996) or a stochastic search variable selection approach (e.g. Barnett et al. 1996). More recently, (Philippe 2001) applied an algorithm developed in Stephens (2000) based on the construction of a continuous time Markov birth-death process to deal with order uncertainty in AR models.

In this paper, we use reversible jump MCMC for moving between the different possible models. Though not necessary, we shall assume that the values of σ_ϵ^2, σ_a^2 and σ_b^2 remain unchanged under model moves. The updating scheme is implemented in two steps by firstly updating the AR coefficients via random walk Metropolis, and then proposing to add a new coefficient or deleting an existing one. In the second step, this same scheme is applied to the MA coefficients. At each iteration a random choice between the birth or death move is made with probability 1/2. The death of an AR coefficient is rejected with probability 1 when the current model is either ARMA(1,0) or ARMA(0,q) for any value of q. Similarly the death of an MA coefficient is rejected with probability 1 if the current model is either ARMA(0,1) or ARMA(k,0) for any value of k. Also, a birth move in the AR component is rejected with probability 1 when $k = k_{\max}$, and likewise for a birth move in the MA component when $q = q_{\max}$.

Birth moves are proposed by sampling a new coefficient from the proposal density $N(0, \sigma_q^2)$ and keeping the other coefficient values unaltered so that the Jacobian of the transformation is equal to one. We then use (3) with the *a priori* independence assumption to accept or reject this move. Note that the prior ratio for the model order is always 1 in this case since we are using uniform priors. Similarly, death moves are proposed by deleting the excess coefficient and evaluating the prior and proposal densities at these values.

We also propose what we call arima moves by proposing a new value for the number of differences d and updating k accordingly. For example, we can propose a move from ARIMA$(k, 0, q)$ to ARIMA$(k-1, 1, q)$ or ARIMA$(k-2, 2, q)$, so we allow unit roots (if they exist) to be either complex or real. The criteria for proposing arima moves are as follows: we randomly choose one root which is greater (in absolute value) than a prespecified lower bound L and propose 1 or 2 differences depending on the root being real or complex (this implies deleting 1 or 2 roots). Otherwise, the number of differences is decreased by 1 or 2, which implies adding 1 or 2 roots by sampling from U(-1,-L) or U(L,1) with probability 1/2.

2.3. Efficient Proposals

The performance of the resulting Markov chain, in particular the ability to jump between models, will depend critically upon the choice of the proposal distribution. While for within-model moves it is fairly easy to choose proposals that lead both to high acceptance rates and rapid mixing, this is considerably more difficult for trans-dimensional algorithms as there is no Euclidean structure between models to guide proposal choices. In practice, the proposals are typically tuned on the basis of short pilot runs.

There have been several recent suggestions as to how to construct efficient proposals in trans-dimensional MCMC (see Green 2003, for a review on these methods). In particular, Brooks et al.

(2003) develop methods to try and find parameters for the proposal distribution based upon a Taylor series expansion of the acceptance ratio for certain canonical jumps. Their method is an attempt to translate the natural ideas for proposal construction from a Euclidean space to the union of model spaces. Brooks and Ehlers (2002) talk about how to choose optimal values for the proposal parameters in AR models and we extend their development to choose optimal values for the proposal parameters in ARMA models.

We begin by considering jumps in which one or more parameters is added or deleted from the model, by generating the required number of new parameters but leaving the existing ones unaltered. Suppose that we are currently in model ARMA(k, q) and we must generate a new value k' to which we will propose a jump. It is often desirable to place higher probability on models closest to the current one so as to avoid spending too much time proposing moves to models with very low posterior mass (Troughton and Godsill 1997). This can be accomplished by using a discretised Laplacian distribution so that the distribution for k' is given by

$$p(k') \propto \exp(-\beta|k - k'|), \quad k' \in [1, \ldots, k_{\max}],$$

where $\beta \geq 0$ denotes a scale parameter. Of course, taking $\beta = 0$, we obtain the uniform proposal. However, if we take $\beta = 0.5$, say, then a jump to $k \pm 1$ is three times more likely than a jump to $k \pm 3$, for example. Without loss of generality, let us suppose that $k' > k$, and that we generate $\boldsymbol{u} = (u_1, \ldots, u_{k'-k})$ from some proposal density q.

Another type of move to be explored here is what Brooks and Ehlers (2002) call a non-deterministic down move. In this case we propose a jump from k to k' by generating the whole vector of AR coefficients in the k'-dimensional space, so that moves to lower dimensional spaces are no longer deterministic. We note that in this case the current values of $\mathbf{a}^{(k)}$ will not be used to determine the new values. In terms of dimension matching, this is equivalent to setting the change of variables as $\mathbf{a}' = \boldsymbol{u}$ and $\boldsymbol{u}' = \mathbf{a}$, which has unity Jacobian. This move would then be accepted with probability $\min(1, A)$ where, from (1),

$$A = \frac{p(\boldsymbol{y}|k', q, \boldsymbol{u}, \mathbf{b}, \sigma_\epsilon^2)}{p(\boldsymbol{y}|k, q, \mathbf{a}, \mathbf{b}, \sigma_\epsilon^2)} \frac{p(\boldsymbol{u}|k', \sigma_a^2)}{p(\mathbf{a}|k, \sigma_a^2)} \frac{r_{k',k} q(\mathbf{a})}{r_{k,k'} q(\boldsymbol{u})} \tag{6}$$

and this remains unchanged whether or not $k < k'$. Similar comments follow for the MA component.

For this type of model move, taking the proposal density for $\boldsymbol{u}$ to be a multivariate normal density then expressions for the mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{C}$ are given in the appendix. Note that we need to calculate the proposal parameters for both the proposed move and the corresponding reverse (non-deterministic) move in order to evaluate this acceptance ratio.

Having described all the steps necessary for the MCMC simulations, we now present both a simulation study and the analysis of a real data set where we give details concerning prior specification and examine the performance of the updating schemes described in this section.

2.4. Simulation Study

To assess the performance of our RJMCMC algorithm for order selection problems in ARMA models we simulated 20 data sets from AR(3), MA(3) and ARMA(3,3) stationary processes with 1000 data points. For each data set we ran our full updating scheme during 1,000,000 iterations (discarding the first 500,000 as burn-in) using the first 20, 50, 100, 500 and 1000 observations and recorded the estimated posterior probability of the true model. We also recorded the proportion of time that the algorithm correctly selects the true model for each sample size. These are shown in Table 1 where, for each process, the first row shows the average posterior probability of the true model while the second row shows the proportion of correct choices. It is clear that, as the sample size increases the performance of the algorithm improves. Acceptable performances seem to be achieved for at least 200 observations.

2.5. A Real Data Example

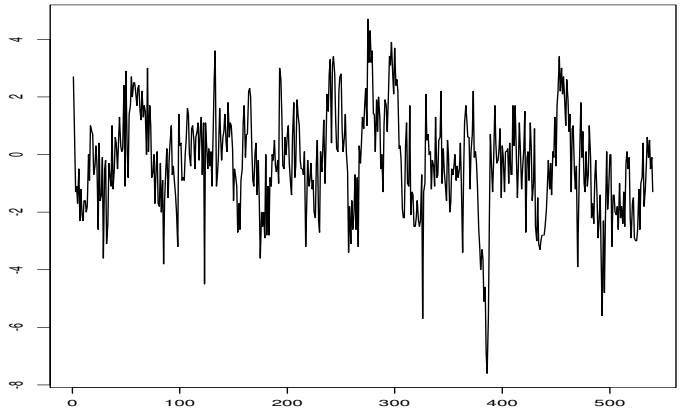
In this section we illustrate our updating scheme for ARMA models on one data set which has been analysed in the time series literature: the Southern oscillation index (SOI). The series appears in

Table 1. Changes in model probabilities and proportion of correct model for simulated AR(3), MA(3) and ARMA(3,3) processes as the sample size increases.

	Sample size					
model	20	50	100	200	500	1000
AR(3)	0.0561	0.1837	0.2842	0.4401	0.6034	0.7336
	0.1429	0.8095	0.9048	0.7619	0.9048	0.9524
MA(3)	0.0364	0.0438	0.0736	0.1028	0.2189	0.3367
	0.0476	0.0476	0.2857	0.3333	0.8095	0.8571
ARMA(3,3)	0.0202	0.0285	0.0524	0.1232	0.2663	0.3967
	0.0000	0.0000	0.0476	0.3333	0.8571	0.9524

Trenberth and Hoar (1996) and Huerta and West (1999) and consists of 540 monthly observations registered from 1950 to 1995 and is related to sea surface temperature. The original SOI series is plotted in Figure 1.

Fig. 1. Southern oscillation index. 540 measurements taken between 1950-1995 of the difference of the departure from the long-term monthly mean sea level pressures at Tahiti and Darwin.



In order to compare the different methods for specifying proposal parameters we fit ARMA models to this data set with both AR and MA orders varying from 0 to 5, thus considering a set of competing parsimonious models. The variance of the proposal distribution for model moves was set, after pilot-tuning, as $\sigma^2 = 0.01$, and we then used an approximate likelihood based on fixed error terms for proposing model moves based on the second order method of Brooks et al. (2003). The within-model moves (with k and q fixed) were performed by updating the AR and MA coefficients via the random-walk Metropolis algorithm as described in Section 2.1 using proposal variances $\sigma_n^2 = 0.1$. We assigned inverse Gamma priors with parameters 0.01 for the error and prior variances.

Following Brooks et al. (2003) and Brooks and Ehlers (2002) we compare each algorithm in terms of the mean between-model acceptance rates ($\bar{\alpha}$) for the AR and MA components, the effective sample size (ESS) and the total computation time. We compute the effective sample size by monitoring a unique scalar that retains its interpretation across all ARMA(k, q) models. A suitable choice is to calculate these diagnostics for the model indicator $(k_{\max} + 1)q + k$ as this assumes a unique value for each possible model. Note that for pure AR and MA models this reduces to the usual model order k or q.

Then, 1,000,000 iterations of the algorithm were run discarding the first 500,000 as burn-in. The results of these simulations are presented in Table 2. We can see that the effective sample size is quite small for both unidimensional updating schemes, and it is lower for the second order method which may indicate lower performance in terms of mixing. Even so, the acceptance rates for both the AR

Table 2. SOI data. Comparing performance of different algorithms via acceptance rates $\bar{\alpha}$ for the AR and MA components, effective sample size ESS and computation time.

method	$\bar{\alpha}$		ESS	time (min.)
pilot tuning	0.10	0.16	1645	12.6765
2nd order	0.12	0.19	913	14.3577
partial updating	0.08	0.11	3035	15.8525
full updating	0.11	0.11	5437	15.3960

Table 3. Posterior model order probabilities for the SOI data based on 500,000 iterations after a 500,000 burn-in. Top model highlighted in bold.

	MA order					
AR order	0	1	2	3	4	5
1	0.0000	**0.4428**	0.0259	0.0042	0.0031	0.0056
2	0.0178	0.0360	0.0717	0.0141	0.0102	0.0136
3	0.0971	0.0254	0.0151	0.0097	0.0057	0.0066
4	0.0817	0.0336	0.0124	0.0063	0.0034	0.0033
5	0.0239	0.0153	0.0063	0.0040	0.0024	0.0028

and MA components increased with the second order method compared to pilot tuning. Comparing multidimensional move schemes, full updating clearly performs better than partial updating with a higher acceptance rate for the AR component and better mixing.

This provides evidence that using the suggested proposals for ARMA models can be more efficient than pilot tuning despite being based on an approximate likelihood. In any case, using the (approximate) optimal proposals we are allowing the reversible jump sampler to adapt along iterations.

The posterior distribution of model order for the full updating scheme appears in Table 3 where we can see that when we include MA terms in the model the ARMA(1,1) is identified as the most likely one with decreasing posterior support for models AR(3), AR(4), ARMA(2,2), ARMA(2,1), AR(3,1) and ARMA(4,1) and low probabilities at other models. Of course for this case an exaustive search over the 35 possible ARMA models can be performed and the classical information criteria (AIC and BIC) also select model ARMA(1,1).

3. Imposing Stationarity and Inversibility

When reparameterising the model in terms of reciprocal roots as in (5) then updating one (or a conjugate pair) of the reciprocal roots the whole vector of coefficients (either AR or MA) is changed. Also it is not necessary to impose an identifying ordering on the roots since the vector of coefficients is the same irrespective of ordering. The set of all variables in the problem is now $(\boldsymbol{y}, k, q, \boldsymbol{\lambda}, \boldsymbol{\delta}, \sigma_\epsilon^2)$ and the likelihood function has the same form in terms of ARMA coefficients and is evaluated by first mapping from $\boldsymbol{\lambda}$ to $\mathbf{a}$ and from $\boldsymbol{\delta}$ to $\mathbf{b}$ since $p(\boldsymbol{y}|k, q, \boldsymbol{\lambda}, \boldsymbol{\delta}, \sigma_\epsilon^2) = p(\boldsymbol{y}|k, q, \mathbf{a}(\boldsymbol{\lambda}), \mathbf{b}(\boldsymbol{\delta}), \sigma_\epsilon^2)$. In the next sections we present details of priors and proposals in terms of the AR reciprocal roots and the development for the MA component is analogous.

3.1. Parameter Priors

Conditional on model order we assume independent priors for the real and any pairs of complex conjugate reciprocal roots. A real reciprocal root r has a continuous prior density over the support (-1,1) while a pair of complex conjugates $(\lambda_j, \lambda_{j^*})$ can be written as

$$\begin{aligned} \lambda_j &= r\cos\theta + ir\sin\theta \\ \lambda_{j^*} &= r\cos\theta - ir\sin\theta \end{aligned}$$

and here we specify the prior distribution of the complex pair in terms of the two defining parameters, θ and r, over a support in the stationary region. So, assuming that θ lies in the interval $(0, \pi)$ then $r \in (-1, 1)$ as in the case for real roots. Here we shall also assume that θ and r are *a priori* independent so that the prior density for the complex pair is given by

$$p(\lambda_j, \lambda_{j^*}) = p(\theta)p(r) \left| \frac{\partial(\lambda_j, \lambda_{j^*})}{\partial(\theta, r)} \right|^{-1}.$$

While we can consider a variety of priors for r we will rarely, if ever, have any prior information concerning θ and we shall therefore assume a $U(0, \pi)$ prior distribution throughout.

To place prior information on r, we can reparameterise the reciprocal roots by taking a real quantity x without restriction and try and find a suitable function that maps the real line onto the interval (-1,1). Here we use the function

$$r = \frac{2e^x}{1 + e^x} - 1 \tag{7}$$

with inverse $x = \log((1 + r)/(1 - r))$. We can now place suitable priors on x, noting that very large values of $|x|$ correspond to values of $|r|$ very close to 1. A $N(0, \sigma_a^2)$ provides a reasonable family of prior distributions, centred on zero (corresponding to $r = 0$), with σ_a^2 determining the shape of the prior distribution of r. This becomes more concentrated around zero as σ_a^2 decreases and becomes U-shaped and more concentrated near -1 and 1 as σ_a^2 increases thus describing a broad range of prior beliefs. We refer to this as a logistic-based prior.

For this prior specification the full conditional distribution of the hyperparameter σ_a^2 can be obtained analytically and has a standard form. Under the assumed conditionally independent prior and assigning an inverse Gamma prior to σ_a^2 its full conditional distribution is given by

$$IG\left(\alpha + \frac{n_r + n_c}{2}, \beta + \frac{1}{2}\left[\sum_{i:\lambda_i \in \mathbb{R}} x_i^2 + \sum_{j:\lambda_j \in \mathbb{C}} x_j^2\right]\right)$$

where n_r and n_c are the number of real roots and the number of complex conjugate pairs respectively, and α and β are the prior parameters.

3.2. Within-Model Moves

For k and q fixed, it is easy to see that the full conditional distribution of σ_ϵ^2 has the same inverse Gamma form as in Section 2.1 with the vector of ARMA coefficients computed from the reciprocal roots. This parameter is then updated by a Gibbs move.

In order to update the ARMA coefficients we randomly choose one of the reciprocal roots and use Metropolis-Hastings updates with the proposal density centered on the current value as follows. If the chosen λ_j is real we propose a new value by sampling λ_j' from $U[\max(\lambda_j - \delta, -1), \min(\lambda_j + \delta, 1)]$. Of course when $\lambda_j - \delta < -1$ and $\lambda_j + \delta > 1$ the proposal distribution is simply the $U(-1, 1)$ and the proposal ratio in the acceptance probability is equal to 1.

A similar approach is adopted when the chosen λ_j is complex, we propose a new value for the pair $(\lambda_j, \lambda_{j^*})$ by sampling θ^* from $U[\max(0, \theta-\delta), \min(\pi, \theta+\delta)$ and r^* from $U[\max(r-\delta, -1), \min(r+\delta), 1]$, and setting the proposed new values as $r^* \cos\theta^* \pm ir^* \sin\theta^*$.

The above schemes ensure that the new values of the reciprocal roots (either real or complex), are proposed in a neighbourhood of the current ones and are restricted to stationarity.

3.3. Model Priors

For particular values of k and q the root structure is not unique, except for $k = 1$ or $q = 1$. For example, if $k = 4$ we can have 4 real roots, 2 real and 2 complex roots or 4 complex roots in the AR component, so that there are 3 possible configurations of real and complex roots corresponding to $k = 4$. Therefore, in order to assign a uniform prior on the AR order, the prior probability for a certain value of k should be split uniformly over the possible configurations of real and complex roots corresponding to that order. Likewise for the MA component.

If k is even, then the roots can be divided into $d = k/2$ pairs and each of them can be either real or complex. Since the order is irrelevant and the number of pairs of one type (real or complex) can vary from 0 to d it follows that the number of possible configurations is given by $d + 1$. If k is odd, there are $d = [k/2]$ pairs of roots, where $[x]$ denotes the integer part of x, plus one real root and the number of possible configurations is again $d + 1$ since the number of pairs of one type varies from 0 to d. Therefore, given the value of model order k, it follows that

$$\mathbb{P}(r \text{ real and } c \text{ complex roots}) \propto \frac{1}{[k/2]+1}.$$

This prior specification differs from Huerta and West (1999) where a uniform distribution is assigned to the possible configurations of real and complex roots thus leading to a non-uniform prior distribution on model order.

3.4. Between-Model Moves

We employ model moves that increase or decrease the model dimension by one or two by proposing the addition (or deletion) of one real or a pair of complex conjugate roots. The reciprocal roots common to the current and proposed model remain unchanged. Model moves are performed in two steps by first deciding on the birth or death of roots and then deciding on a single real or a pair of conjugate complex roots to be added or deleted. So, four model move types are allowed: real birth, complex birth, real death and complex death. Here each move type is proposed with the same probability 1/4 so that they cancel out in the proposal ratio.

Suppose that we propose a move from ARMA(k, q) to ARMA$(k+1, q)$ by adding one real reciprocal root r sampled from a continuous distribution over the support (-1,1). The models are nested in terms of reciprocal roots and these are assumed *a priori* independent.

Suppose now that we propose adding a pair of complex reciprocal roots $(u, \bar{u})$ where

$$\begin{aligned} u &= r\cos\theta + ir\sin\theta \\ \bar{u} &= r\cos\theta - ir\sin\theta. \end{aligned}$$

Here, we specify the proposal distribution of the complex pair in terms of the two defining parameters θ and r over the support $(0, \pi) \times (-1, 1)$. So, we shall propose new values not for $(u, \bar{u})$ directly but for (θ, r) so that the Jacobian term will cancel in the acceptance ratio with that arising from the prior.

Note that, under this parameterisation and updating scheme, the models can be treated as nested so that the Jacobian of the transformation from $(\lambda_1, \ldots, \lambda_k)$ to either $(\lambda_1, \ldots, \lambda_k, r)$ or $(\lambda_1, \ldots, \lambda_k, u, \bar{u})$ equals 1 and does not appear in the proposal ratios. Also, the likelihood ratio is computed by first mapping the set of reciprocal roots to $(a'_1, \ldots, a'_{k+1})$ or $(a'_1, \ldots, a'_{k+2})$.

Conversely, a real (or complex) death move is proposed by randomly selecting one of the real (or complex) roots and deleting it (or the pair of complex conjugates).

3.4.1. Proposals

Here we consider three families of proposal densities. The first proposal samples a value for the new root r by generating a realisation from a $N(\mu, \sigma^2)$ distribution truncated to the interval $(-1, 1)$. Our second proposal samples $u \sim \beta(\alpha_1, \alpha_2)$ and sets $r = 2u - 1$. Our third proposal samples a new value for

$$\rho = \log\left(\frac{1+r}{1-r}\right),$$

from a $N(\mu, \sigma^2)$ distribution and maps to r using function (7). We refer to these as truncated normal, beta-based and logistic-based proposals respectively. Each of these proposals are characterised by two proposal parameters that can be determined by pilot tuning or using the methods described in Brooks and Ehlers 2002.

Of course, for either real or complex cases if we take both prior and proposal in the same family (logistic-based) the Jacobian term will cancel with that arising from the prior. We can also take the prior as a proposal for θ in which case we sample a new value from a $U(0, \pi)$ and $q(\theta)$ cancels with $p(\theta)$ in the acceptance ratio.

3.5. Efficient Proposals for Reciprocal Roots

Brooks and Ehlers (2002) also talk about how to choose optimal values for the proposal parameters in AR models parametrised in terms of reciprocal roots. Since our scheme updates the AR and MA components separately we can extend those developments to choose proposal parameters in ARMA models too.

Suppose that we propose a move from ARMA(k, q) to ARMA$(k+1, q)$ by adding one real reciprocal root r. Then using the representation (5) and denoting the error terms in the higher dimensional model by ϵ'_t we obtain

$$(1 - rL)\prod_{i=1}^{k}(1 - \lambda_i L)y_t = \prod_{j=1}^{q}(1 - \delta_j L)\epsilon'_t$$

so that $\epsilon'_t = (1 - rL)\epsilon_t = \epsilon_t - r\epsilon_{t-1}$ where the ϵ_t denote the error terms in the original model. Thus, the likelihood function under the larger model is simply,

$$L(\boldsymbol{y} \mid k, q, \boldsymbol{\lambda}, \boldsymbol{\delta}, \sigma_\epsilon^2) \propto \exp\left[-\frac{1}{2\sigma_\epsilon^2}\sum(\epsilon_t - r\epsilon_{t-1})^2\right]$$

where the proportionality constant includes terms that do not depend on r. This is exactly the same expression that appears in Brooks and Ehlers (2002) with the error terms redefined here for ARMA models. So, we can use the expressions given there for the various combinations of prior and proposal distributions.

Suppose now that we propose a move from ARMA(k, q) to ARMA$(k + 2, q)$ by adding a pair of complex reciprocal roots $(u, \bar{u})$. Then, it is easy to show that the error terms in the higher dimensional model can be written as $\epsilon'_t = \epsilon_t - 2r\cos\theta\epsilon_{t-1} + r^2\epsilon_{t-2}$ and the likelihood function for θ and r is again given by the same expression as in Brooks and Ehlers (2002). Their second order method suggests taking the prior of θ as a proposal and they give expressions for the proposal parameters of r for various combinations of prior and proposal distributions.

3.5.1. Updating the MA Component

Consider now jumps that alter the dimension of the MA component and we begin by considering a move from ARMA(k, q) to ARMA$(k, q+1)$ by adding one real reciprocal root, r. Then the representation for the higher dimensional model is

$$\prod_{i=1}^{k}(1 - \lambda_i L)y_t = (1 - rL)\prod_{j=1}^{q}(1 - \delta_j L)\epsilon'_t$$

so that $\epsilon_t = (1 - rL)\epsilon'_t = \epsilon'_t - r\epsilon'_{t-1}$ where the ϵ_t denote the error terms in the original model. Thus, the derivatives needed to apply the second order method are not available since each error term ϵ'_t depends on r in a complicated non-linear way. Here, we approximate these derivatives by treating ϵ'_{t-1} as if it were fixed in the larger model. In this case, the likelihood function under the larger model is given by,

$$L(\boldsymbol{y} \mid k, q, \boldsymbol{\lambda}, \boldsymbol{\delta}, \sigma_\epsilon^2) \propto \exp\left[-\frac{1}{2\sigma_\epsilon^2}\sum(\epsilon_t + r\epsilon_{t-1})^2\right].$$

To apply the second order method we take first and second derivatives of the logarithm of the likelihood with respect to r.

3.6. Simulation Study

To assess the performance of our RJMCMC algorithm for order selection problems in ARIMA models we turn to the simulated AR(3), MA(3) and ARMA(3,3) data sets of Section 2.4. For each data set and each of the three proposal families we ran our algorithm during 1,000,000 iterations (discarding the first 500,000 as burn-in) using the first 20, 50, 100, 500 and 1000 observations and recorded the estimated posterior probability of the true model. Here the maximum ARMA model orders are $k_{\max} = q_{\max} = 5$ and $d = 0, 1, 2$, so the number of possible ARIMA models is quite large and an exhaustive enumeration would be cumbersome. We also recorded the proportion of time that the

Table 4. Changes in model probabilities and proportion of correct model for a simulated AR(3) processe as the sample size increases considering the three families of proposals.

	Sample size					
proposal	20	50	100	200	500	1000
Truncated normal	0.0092	0.0398	0.1074	0.2677	0.4565	0.5480
	0.0000	0.1500	0.3500	0.6000	0.9000	0.9500
Beta-based	0.0096	0.0441	0.1059	0.2702	0.4812	0.5404
	0.0000	0.1500	0.3500	0.6500	0.9500	0.9000
Logistic-based	0.0092	0.0414	0.1058	0.2628	0.4822	0.5436
	0.0000	0.1500	0.3000	0.6000	0.9500	0.9000

algorithm correctly selects the true model for each sample size. In Table 4 we show the results for the simulated AR(3) processes where, for each proposal distribution the first row refers to the average posterior probability of the true model while the second row shows the proportion of correct choices.

Clearly, the performance of the algorithm improves as the sample size increases and a quite similar pattern is observed for the three proposals considered. Acceptable performances seem to be achieved for at least 200 observations.

3.7. The SOI Data Revisited

In this section, we illustrate the algorithms proposed to sample model parameters and model order in the parameterisation in terms of reciprocal roots with a real data set. The Southern Oscillation Index (SOI) data, described in Section 2.5, is analysed here. The analysis is based on an ARIMA model with maximum order $k_{\max} = q_{\max} = 5$ and $d_{\max} = 2$ posterior inference is based on 500,000 samples after discarding the initial 500,000 as burn-in.

The posterior distribution of model order appears in Table 5, for each of the three proposal families of distributions and $d = 0, 1$. We can see that the ARIMA(1,0,1) model is identified as the most likely one with much lower posterior support for other models (as in the previous parameterisation). Also, ARIMA$(p, 2, q)$ models were not visited often enough after the burn-in period. This behaviour is quite similar for the three different proposal families.

4. Discussion

In this paper we illustrate the Bayesian approach to simultaneous parameter estimation and model order selection for the class of ARIMA time series models. In particular, we address the problem of order selection in a MCMC framework via reversible jump algorithms.

We presented an alternative parameterisation in terms of reciprocal roots of the characteristic equations. This allowed us to enforce stationarity and inversibility constraints in the model parameters in a very straightforward way. Since the stationary/inversible region in convex if each set of reciprocal roots generated via MCMC satisfies those constraints, then so does their means. So, the parameter estimates are guaranteed to impose stationarity/inversibility.

Even for fixed dimension, the presence of moving average terms in the model introduces complex non-linearities in the likelihood function and classical estimation of model parameters would require numerical opmization methods to be used. In the Bayesian approach, the posterior density of model parameters is of the same complexity as the likelihood function and cannot be directly computed, thus approximation methods are necessary to derive posterior inferences on these parameters. Another difficulty in ARMA models concerns the problem of roots cancellation, i.e. there may be common factors cancelling out if there are similar AR and MA roots. This is a generic phenomenon with ARMA models and the likelihood function is very badly behaved if we overparameterise.

It has been shown via simulated examples that the approach can reliably select the best model for reasonable sample sizes, and it has performed well with a real data set. The approach developed in this paper can be extended to other classes of models (e.g. threshold autoregression, smooth transition autoregression and stochastic volatility models) and this is object of current and future research.

Table 5. Posterior model order probabilities for the SOI data based on 500,000 iterations after a 500,000 burn-in. Top model highlighted in bold.

			MA order					
Proposal	d	AR order	0	1	2	3	4	5
Truncated normal	0	1	0.0000	**0.2245**	0.0518	0.0503	0.0398	0.0336
		2	0.0024	0.0164	0.0556	0.0292	0.0249	0.0184
		3	0.0111	0.0173	0.0374	0.0286	0.0247	0.0186
		4	0.0130	0.0083	0.0178	0.0134	0.0140	0.0104
		5	0.0200	0.0071	0.0148	0.0113	0.0103	0.0085
	1	0	0.0000	0.0135	0.0089	0.0156	0.0147	0.0193
		1	0.0000	0.0023	0.0042	0.0067	0.0078	0.0105
		2	0.0001	0.0018	0.0032	0.0053	0.0067	0.0086
		3	0.0004	0.0020	0.0024	0.0041	0.0042	0.0058
		4	0.0004	0.0012	0.0024	0.0034	0.0050	0.0060
Beta-based	0	1	0.0000	**0.2383**	0.0555	0.0560	0.0378	0.0353
		2	0.0020	0.0181	0.0417	0.0239	0.0218	0.0159
		3	0.0086	0.0161	0.0355	0.0258	0.0258	0.0183
		4	0.0108	0.0070	0.0199	0.0124	0.0136	0.0092
		5	0.0105	0.0057	0.0197	0.0105	0.0106	0.0083
	1	0	0.0000	0.0147	0.0095	0.0176	0.0171	0.0235
		1	0.0000	0.0023	0.0044	0.0076	0.0086	0.0117
		2	0.0000	0.0018	0.0036	0.0065	0.0076	0.0098
		3	0.0003	0.0016	0.0025	0.0047	0.0054	0.0074
		4	0.0003	0.0012	0.0022	0.0037	0.0043	0.0060
Logistic-based	0	1	0.0000	**0.2675**	0.0575	0.0598	0.0393	0.0355
		2	0.0017	0.0207	0.0323	0.0234	0.0173	0.0168
		3	0.0142	0.0208	0.0216	0.0236	0.0206	0.0183
		4	0.0103	0.0094	0.0117	0.0109	0.0106	0.0101
		5	0.0102	0.0078	0.0101	0.0089	0.0086	0.0082
	1	0	0.0000	0.0157	0.0107	0.0189	0.0193	0.0240
		1	0.0000	0.0031	0.0042	0.0076	0.0089	0.0124
		2	0.0001	0.0019	0.0028	0.0061	0.0078	0.0105
		3	0.0001	0.0012	0.0022	0.0043	0.0050	0.0068
		4	0.0001	0.0011	0.0023	0.0045	0.0046	0.0057

5. Acknowledgements

The work of the first author was funded by the National Council for Scientific and Technological Development - CNPq, of the Ministry of Science and Technology of Brazil. The work of the second author was supported by the UK Engineering and Physical Sciences Research Council, under grant number AF/000537.

A. Efficient Proposals for ARMA Models

Here we seek to generalise the efficient construction of proposal distributions in Brooks and Ehlers (2002) to ARMA models. We concentrate on their second order method which involves setting to zero the first and second derivatives of the acceptance ratio. In this case, each error term ϵ_t depends on the whole set of ARMA coefficients in a complicated non-linear way. So, in order to apply the optimal proposal methods in this context we need to make some simplifying assumptions. We shall assume that, when proposing to add new coefficients or delete existing ones, the previous values of the error term are kept fixed.

A.1. Updating the AR component

Consider jumps from ARMA(k,q) to ARMA(k',q) where $k' \in \{k+1,\ldots,k_{\max}\}$ by sampling a vector of random variables $\boldsymbol{u} = (u_1,\ldots,u_{k'-k})$ and keeping the MA component fixed so that the new vector of ARMA coefficients is given by $(\mathbf{a}^{(k)}, \boldsymbol{u}, \mathbf{b}^{(q)})$.

Defining the $(n-k_{\max}) \times q$ matrix of errors

$$\mathbf{E} = \begin{bmatrix} \epsilon_{k_{\max}} & \cdots & \epsilon_{k_{\max}-q+1} \\ \vdots & & \vdots \\ \epsilon_{n-1} & \cdots & \epsilon_{n-q} \end{bmatrix}$$

the Gaussian autoregressive moving average model of order (k',q) can be written as

$$\boldsymbol{y} = \mathbf{Y}^*\mathbf{a} + \mathbf{Y}\boldsymbol{u} + \mathbf{E}\mathbf{b} + \boldsymbol{\epsilon}$$

where $\boldsymbol{y} = (y_{k_{\max}+1},\ldots,y_n)'$, $\boldsymbol{\epsilon} = (\epsilon_{k_{\max}+1},\ldots,\epsilon_n)'$ and

$$\mathbf{Y}^* = \begin{bmatrix} y_{k_{\max}} & \cdots & y_{k_{\max}-k+1} \\ \vdots & & \vdots \\ y_{n-1} & \cdots & y_{n-k} \end{bmatrix} \quad \text{and} \quad \mathbf{Y} = \begin{bmatrix} y_{k_{\max}-k} & \cdots & y_{k_{\max}-k'+1} \\ \vdots & & \vdots \\ y_{n-k-1} & \cdots & y_{n-k'} \end{bmatrix}$$

Then, for $\boldsymbol{u}$ sampled from a multivariate Normal distribution with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{C}$ and using the *a priori* independence assumption, the acceptance ratio is given by

$$A_{k,k'} \propto \exp\left[-\frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}^* - \mathbf{Y}\boldsymbol{u})'(\boldsymbol{\epsilon}^* - \mathbf{Y}\boldsymbol{u})\right] \exp\left[-\frac{1}{2\sigma_a^2}\boldsymbol{u}'\boldsymbol{u}\right] \exp\left[\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu})'\boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu})\right] \qquad (8)$$

where $\boldsymbol{\epsilon}^* = \boldsymbol{y} - \mathbf{Y}^*\mathbf{a} - \mathbf{E}\mathbf{b}$ and the terms in the proportionality constant do not depend on $\boldsymbol{u}$.

The second order method can be applied by setting to zero the first and second order derivatives of $\log A_{k,k'}$ with respect to $\boldsymbol{u}$ and ignoring the zeroth order term. However, these derivatives are not available since each error term depends on the whole set of coefficients in a complicated non-linear way. So, $\partial\boldsymbol{\epsilon}^*/\partial\boldsymbol{u}$ is too complex and we approximate by treating $\mathbf{E}$ as if it were fixed in order to get $\boldsymbol{\mu}$ and $\boldsymbol{C}$. We then obtain that

$$\begin{aligned} \nabla \log A_{k,k'} &= \sigma_\epsilon^{-2}\mathbf{Y}'(\boldsymbol{\epsilon}^* - \mathbf{Y}\boldsymbol{u}) - \sigma_a^{-2}\boldsymbol{u} + \boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) \\ \nabla^2 \log A_{k,k'} &= -\sigma_\epsilon^{-2}\mathbf{Y}'\mathbf{Y} - \sigma_a^{-2}\mathbf{I}_{k'-k} + \boldsymbol{C}^{-1}. \end{aligned}$$

Of course, since $\log A_{k,k'}$ is a quadratic function of $\boldsymbol{u}$, the second derivative does not depend on the value of $\boldsymbol{u}$ and setting it to zero we obtain

$$\boldsymbol{C}^{-1} = \sigma_\epsilon^{-2}\mathbf{Y}'\mathbf{Y} + \sigma_a^{-2}\mathbf{I}_{k'-k}.$$

Similarly, setting the first derivative to zero and using the above expression for $\boldsymbol{C}^{-1}$ it follows that

$$\begin{aligned}\boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) &= \sigma_a^{-2}\mathbf{I}_{k'-k}\boldsymbol{u} - \sigma_\epsilon^{-2}\mathbf{Y}'(\boldsymbol{\epsilon}^* - \mathbf{Y}\boldsymbol{u}) \\ &= (\sigma_\epsilon^{-2}\mathbf{Y}'\mathbf{Y} + \sigma_a^{-2}\mathbf{I}_{k'-k})\boldsymbol{u} - \sigma_\epsilon^{-2}\mathbf{Y}'\boldsymbol{\epsilon}^* = \boldsymbol{C}^{-1}\boldsymbol{u} - \sigma_\epsilon^{-2}\mathbf{Y}'\boldsymbol{\epsilon}^*\end{aligned}$$

so the proposal mean is given by $\boldsymbol{\mu} = \sigma_\epsilon^{-2}\boldsymbol{C}\mathbf{Y}'(\boldsymbol{y} - \mathbf{Y}^*\mathbf{a} - \mathbf{E}\mathbf{b})$. Note also that, only the proposal mean depends on the current values of the MA coefficients $\mathbf{b}$.

A.2. Full Updating the AR component

We can also propose a jump from ARMA(k, q) to ARMA(k', q) by generating new values for the whole vector of AR coefficients directly in the k'-dimensional space while keeping the MA component fixed.

In this full updating scheme, the expressions for the proposal mean vector and variance matrix in the AR component are simply obtained from the expressions for partial updating by dropping the terms that depend on $\mathbf{a}$ and using the approximation based on $\mathbf{E}$ fixed again. So,

$$\begin{aligned}\boldsymbol{C}^{-1} &= \sigma_\epsilon^{-2}\mathbf{Y}'\mathbf{Y} + \sigma_a^{-2}\mathbf{I}_{k'} \\ \boldsymbol{\mu} &= \sigma_\epsilon^{-2}\boldsymbol{C}\mathbf{Y}'(\boldsymbol{y} - \mathbf{E}\mathbf{b})\end{aligned}$$

where $\mathbf{Y}$ and $\mathbf{E}$ are $(n - k_{\max}) \times k'$ and $(n - k_{\max}) \times q$ matrices respectively.

A.3. Updating the MA Component

When proposing a birth move in the MA component, from order q to order $q' \in \{q+1, \ldots, q_{\max}\}$, with partial updating we generate a vector of random variables $\boldsymbol{u} = (u_1, \ldots, u_{q'-q})$ so that the new vector of ARMA coefficients is given by $(a_1, \ldots, a_k, b_1, \ldots, b_q, u_1, \ldots, u_{q'-q})$. The Gaussian autoregressive moving average model of order (k, q') can be written in matrix form as

$$\boldsymbol{y} = \mathbf{Y}\mathbf{a} + \mathbf{E}^*\mathbf{b} + \mathbf{E}\boldsymbol{u} + \boldsymbol{\epsilon}.$$

where the matrix $\mathbf{Y}$ is now $(n - k_{\max}) \times k$ and the $(n - k_{\max}) \times (q' - q)$ matrix $\mathbf{E}$ is defined as

$$\mathbf{E} = \begin{bmatrix} \epsilon_{k_{\max}-q} & \cdots & \epsilon_{k_{\max}-q'+1} \\ \vdots & & \vdots \\ \epsilon_{n-q-1} & \cdots & \epsilon_{n-q'} \end{bmatrix}.$$

The acceptance ratio is given by

$$A_{q,q'} \propto \exp\left[-\frac{1}{2\sigma_\epsilon^2}(\boldsymbol{\epsilon}^* - \mathbf{E}\boldsymbol{u})'(\boldsymbol{\epsilon}^* - \mathbf{E}\boldsymbol{u})\right] \exp\left[-\frac{1}{2\sigma_b^2}\boldsymbol{u}'\boldsymbol{u}\right] \exp\left[\frac{1}{2}(\boldsymbol{u}-\boldsymbol{\mu})'\boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu})\right]$$

where $\boldsymbol{\epsilon}^* = \boldsymbol{y} - \mathbf{Y}\mathbf{a} - \mathbf{E}^*\mathbf{b}$ is the error term in the lower dimensional ARMA(k, q) model and the terms in the proportionality constant do not depend on $\boldsymbol{u}$.

Using the approximation based on $\mathbf{E}^*$ and $\mathbf{E}$ fixed, the first and second order derivatives of $\log A_{q,q'}$ with respect to $\boldsymbol{u}$ are given by

$$\begin{aligned}\nabla \log A_{q,q'} &= \sigma_\epsilon^{-2}\mathbf{E}'(\boldsymbol{\epsilon}^* - \mathbf{E}\boldsymbol{u}) - \sigma_b^{-2}\boldsymbol{u} + \boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) \\ \nabla^2 \log A_{q,q'} &= -\sigma_\epsilon^{-2}\mathbf{E}'\mathbf{E} - \sigma_b^{-2}\mathbf{I}_{q'-q} + \boldsymbol{C}^{-1}.\end{aligned}$$

Setting the second derivative, which does not depend on the value of $\boldsymbol{u}$, to zero we obtain

$$\boldsymbol{C}^{-1} = \sigma_\epsilon^{-2}\mathbf{E}'\mathbf{E} + \sigma_b^{-2}\mathbf{I}_{q'-q}.$$

Similarly, setting the first derivative to zero and using the above expression for $\boldsymbol{C}^{-1}$ it follows that

$$\boldsymbol{\mu} = \sigma_\epsilon^{-2}\boldsymbol{C}\mathbf{E}'(\boldsymbol{y} - \mathbf{Y}\mathbf{a} - \mathbf{E}^*\mathbf{b}).$$

A.4. Full Updating the MA Component

Finally, in an MA component full updating scheme we propose a move from q to q' by generating new values for the whole vector of MA coefficients conditional on the current AR coefficients. The proposal mean vector and variance matrix are then obtained by dropping terms depending on $\mathbf{b}$ in the expressions for partial updating, i.e.,

$$\begin{aligned} \boldsymbol{C}^{-1} &= \sigma_\epsilon^{-2}\mathbf{E}'\mathbf{E} + \sigma_b^{-2}\mathbf{I}_{q'} \\ \boldsymbol{\mu} &= \sigma_\epsilon^{-2}\boldsymbol{C}\mathbf{E}'(\boldsymbol{y} - \mathbf{Y}\mathbf{a}) \end{aligned}$$

where $\mathbf{Y}$ and $\mathbf{E}$ are $(n - k_{\max}) \times k$ and $(n - k_{\max}) \times q'$ matrices respectively, and this is assuming again $\mathbf{E}$ fixed.

References

Barbieri, M. and A. O'Hagan (1996). A reversible jump MCMC sampler for Bayesian analysis of ARMA time series. Technical report, Dipartamento di Statistica, Probabilità e Statistiche Applicate, Università "La Sapienza", Roma and Department of Mathematics, University of Nottingham.

Barnett, G., R. Kohn, and S. Sheather (1996). Bayesian estimation of an autoregressive model using Markov chain Monte Carlo. *Journal of Econometrics 74*(2), 237–254.

Box, G. E. P. and G. M. Jenkins (1976). *Time Series Analysis: Forecasting and Control* (Revised ed.). Holden-Day, Oakland, California.

Brooks, S. P. and R. S. Ehlers (2002). Efficient construction of reversible jump MCMC proposals for autoregressive time series models. Technical report, University of Cambridge.

Brooks, S. P., P. Giudici, and G. O. Roberts (2003). Efficient construction of reversible jump MCMC proposal distributions (with discussion). *Journal of the Royal Statistical Society, Series B 65*, 1–37.

Chib, S. and E. Greenberg (1994). Bayes inference in regression models with ARMA(p, q) errors. *Journal of Econometrics 64*, 183–206.

Franses, P. and D. van Dijk (2000). *Non-linear time series models in empirical finance*. Cambridge University Press: Cambridge.

Green, P. J. (1995). Reversible jump MCMC computation and Bayesian model determination. *Biometrika 82*, 711–732.

Green, P. J. (2003). Efficient construction of reversible jump MCMC proposal distributions. In *Highly Structured Stochastic Systems*. Oxford University Press.

Huerta, G. and M. West (1999). Priors and component structures in autoregressive time series models. *Journal of the Royal Statistical Society, Series B 51*, 881–899.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*, 773–795.

Philippe, A. (2001). Bayesian model selection for autoregressive moving average processes. Technical report, IRMA, Lille.

Trenberth, K. E. and T. J. Hoar (1996). The 1990-1995 el Niño southern oscillation event: Longest on record. *Geophysical Research Letter 23*, 57–60.

Troughton, P. and S. Godsill (1997). Reversible jump sampler for autoregressive time series, employing full conditionals to achieve efficient model space moves. Technical report, Cambridge University Engineering Department. CUED/F-INFENG/TR.304.