Bayesian Analysis (2016) **11**, Number 2, pp. 325–352

Exact and Approximate Bayesian Inference for Low Integer-Valued Time Series Models with Intractable Likelihoods

Christopher C. Drovandi[*], Anthony N. Pettitt[†], and Roy A. McCutchan [‡]

Abstract. In this paper we develop a likelihood-free simulation methodology in order to obtain Bayesian inference for models for low integer-valued time series data that have computationally demanding likelihood functions. The algorithm fits within the framework of particle Markov chain Monte Carlo (PMCMC) methods and uses a so-called alive particle filter. The particle filter requires only model simulations and, in this regard, our approach has connections with approximate Bayesian computation (ABC). However, an advantage of using the PMCMC approach in this setting is that simulated data can be matched with data observed one-at-a-time, rather than attempting to match on the full dataset simultaneously or on a low-dimensional non-sufficient summary statistic, which is common practice in ABC. For low integer-valued time series data, we find that it is often computationally feasible to match simulated data with observed data exactly. The alive particle filter uses negative binomial sampling in order to maintain a fixed number of particles. The algorithm creates an unbiased estimate of the likelihood, resulting in exact posterior inferences when included in an MCMC algorithm. In cases where exact matching is computationally prohibitive, a tolerance is introduced as in ABC. This paper further develops the alive particle filter by introducing auxiliary variables so that partially observed and/or non-Markovian models can be accommodated. We demonstrate that Bayesian model choice problems involving such models can be handled with this approach. The methodology is illustrated on a wide variety of models for simulated and real low-count time series data involving a rich set of applications.

Keywords: approximate Bayesian computation, branching process, INARMA model, Markov process, particle filter, particle Markov chain Monte Carlo, pseudo-marginal methods.

1 Introduction

In this paper we develop a simulation methodology to perform exact and approximate Bayesian inference on model parameters for data of low integer-valued time series. The approach relies on simulation from the likelihood to avoid likelihood evaluations, which can be cumbersome or even intractable for some integer-valued time series models in the

[*] ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia, c.drovandi@qut.edu.au

[†] ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia, a.pettitt@qut.edu.au

[‡] Mathematical Sciences School, Queensland University of Technology, Brisbane, Australia, roy.mccutchan@connect.qut.edu.au

© 2016 International Society for Bayesian Analysis DOI: 10.1214/15-BA950

literature (e.g. Markov process and integer autoregressive moving average (INARMA) models). Our approach involves a novel application of the method of Jasra et al. (2013) (see also Persing and Jasra (2014)), who utilise the so-called alive particle filter within a Metropolis–Hastings algorithm to create a new particle Markov chain Monte Carlo (Andrieu et al., 2010) approach.

The motivation for Jasra et al. (2013) and Persing and Jasra (2014) is to estimate the posterior distribution of the static parameter of a state space model. Our motivation for the development of this methodology is different and is to derive a general method for estimating the parameters of complex low integer-valued time series models where previous approximate Bayesian approaches have been used in more limited settings (we discuss such methods and limitations later in this section). To greatly increase the set of models and inference problems our method can be applied to, such as partially observed and non-Markovian models, we extend the alive particle filter to include auxiliary variables. This auxiliary information may include values of missing variables and/or information required to create a Markovian model from a non-Markovian model when necessary.

We demonstrate that the alive particle filter is very useful in the context of low integer-valued time series models as we are able to repeat model simulations until a 'match' is obtained with the observed data and this matching is performed sequentially, one-observation-at-a-time. The alive particle filter includes negative binomial sampling until a certain number of matches are obtained whereas a standard particle filter uses binomial sampling, which may result in no matches. In some cases we show it is possible to perform exact matching between simulated data and observed data. Since the alive particle filter still produces an unbiased estimator of the likelihood for a fixed model parameter, its incorporation within a Metropolis–Hastings algorithm produces a Markov chain that has the correct posterior as its limiting distribution. In other cases it is only computationally feasible to generate close matches. In these situations our method targets an approximate posterior but it is important to note that we are always using the full data so avoiding the need to choose a summary statistic (see Blum et al. (2013)) within a usual approximate Bayesian computation (ABC) algorithm.

Furthermore, we demonstrate that we are able to perform fully Bayesian low integer-valued time series model comparisons using an efficient pseudo-marginal reversible jump (RJ) MCMC algorithm. In one of the examples, we were unable to develop an RJMCMC algorithm on the joint space of parameter and latent variables. However, a viable RJMCMC algorithm was possible via our pseudo-marginal approach.

There are a few other Bayesian methods in the literature that are able to handle some of the inference problems considered in this paper. White et al. (2015) re-structure the full posterior distribution into a form that involves a product of posteriors for single observations. Each component is estimated via simulation with density estimation, and then the components are combined to give an overall posterior approximation. The approach is only applicable for Markov models where all variables in the model are observed. The method of Barthelmé and Chopin (2014) is another summary statistic free ABC method that may be applicable. The method is very fast as it is based on an expectation propagation approximation. However, the method is restricted to posterior distributions that are in the exponential family (an assumption that is difficult to investigate for complicated inference problems). Furthermore, the method of Barthelmé and Chopin (2014) requires further development if some of the variables are unobserved or if the process is non-Markovian. Finally, McKinley et al. (2014) develop pseudo-marginal algorithms for a certain class of stochastic epidemic models. McKinley et al. (2014) propose a matching strategy where auxiliary variables of the model are constrained to be consistent with the observed data. The method adopted here is more generic and is thus applicable to a wider range of models.

The methodology is demonstrated on a rich set of models for simulated data as well as several real datasets. One type of model considered is multivariate Markov processes. The first example is for nosocomial (hospital-acquired) pathogen transmission. This application is mainly for illustrative purposes since for low-valued integer counts the exact likelihood is computationally feasible using the matrix exponential (Moler and van Loan, 2003; Sidje, 1998). However, with these types of models the likelihood computation grows exponentially as more random variables are added to the model (see, for example, Drovandi and Pettitt (2008)). This is illustrated on an example involving an autoregularity gene network, which contains four species.

The second model type considered highlights the utility of developing a pseudo-marginal RJMCMC algorithm. Here the application is to infer the maximum number of offspring that an individual can produce in a branching process when only the population size at each generation is observed. For a fixed upper bound on the offspring distribution, latent variables involving the number of individuals that produce a certain number of offspring at each generation can facilitate a Gibbs sampler. However, it does not appear feasible to move between latent variables of different dimensions. Employing a pseudo-marginal approach allows us to avoid this issue.

The third and final type of model is the integer autoregressive moving average (INARMA) model. The INAR(p) model with order $p = 1$ has a tractable likelihood, but becomes computationally intensive for higher orders. Additionally, our approach can accommodate a moving average component unlike White et al. (2015). Neal and Subba Rao (2007) develop a component-wise Bayesian MCMC algorithm for INARMA models by completing the likelihood with auxiliary variables. This is extended by Enciso-Mora et al. (2009) to an RJMCMC algorithm for model selection between competing INARMA models. Our approach mimics that of an algorithm on the parameter of interest and can therefore avoid potentially poor mixing of the MCMC sampler on the joint space of the model indicator, parameter and latent variables.

This paper is organised as follows. In Section 2, the algorithm is presented. Section 3 considers the examples specified above. Section 4 contains the discussion together with the limitations of the algorithm.

2 Methodology

We denote the observed data (possibly vector) at time $t \in \{1, \ldots, T\}$ as $\boldsymbol{y_t}$ where T is the number of observations. The data represents a low-valued discrete time series. Our approach, similar to ABC, is a simulation based approach that requires simulation from the model and comparison of this simulated data with the observed data. The difference with the common ABC approach is that we match on observations one-at-a-time, making it plausible to exactly or closely match simulated with observed data for the low integer-valued time series data. When exact matching is not feasible, we require an auxiliary variable, $\boldsymbol{s_t}$, which is the simulated version of $\boldsymbol{y_t}$. Exact matching may not be computationally feasible if the proposed model is mis-specified, there are outliers in the data or the observation $\boldsymbol{y_t}$ has several components. Additional auxiliary information in the particle filter is given by $\boldsymbol{x_t}$, which is used to facilitate sequential simulation from the model of interest. In this paper we use $\boldsymbol{x_t}$ to create a Markov model from a non-Markov model and also to represent unobserved variables in a Markov model. The parameter of the model is $\boldsymbol{\theta}$ with prior density $p(\boldsymbol{\theta})$. The likelihood, $p(\boldsymbol{y}|\boldsymbol{\theta})$, where $\boldsymbol{y} = (\boldsymbol{y_1}, \ldots, \boldsymbol{y_T})$, is combined with the prior, $p(\boldsymbol{\theta})$, to produce the posterior, $p(\boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. In situations where exact matching is not feasible, we define an approximate likelihood $p_\epsilon(\boldsymbol{y}|\boldsymbol{\theta})$ where ϵ is an ABC tolerance. An expression for this term, which we refer to as the ABC likelihood, is provided later in this section. In such a situation, our approach targets $p_\epsilon(\boldsymbol{\theta}|\boldsymbol{y}) \propto p_\epsilon(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. Through the introduction of the auxiliary variables $\boldsymbol{s_t}$ and $\boldsymbol{x_t}$, our approach is able to encompass a rich set of models and problems to provide either exact (excluding Monte Carlo error) or approximate Bayesian inferences.

Firstly, the particle filter developed to estimate the likelihood (or approximate likelihood) unbiasedly for the set of models considered in this paper is presented. For completeness, the PMMH algorithm of Andrieu et al. (2010) is also shown in this section.

Finally, following Andrieu and Roberts (2009), we describe how a pseudo-marginal RJMCMC algorithm can be developed for performing fully Bayesian model choice for these types of models. Alternatively, we present an importance sampling estimator of the evidence that can also be used for model comparisons.

2.1 The Alive Bootstrap Particle Filter with Auxiliary Variables

In the most general situation where auxiliary variables are required to create a Markov model and exact matching is not feasible, we define the following (approximate or proxy) likelihood

$$p_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}) = \sum_{\boldsymbol{s_0}} \cdots \sum_{\boldsymbol{s_T}} \sum_{\boldsymbol{x_0}} \cdots \sum_{\boldsymbol{x_T}} \prod_{t=1}^{T} p(\boldsymbol{s_t}, \boldsymbol{x_t}|\boldsymbol{s_{t-1}}, \boldsymbol{x_{t-1}}, \boldsymbol{\theta}) \mathbb{1}(||\boldsymbol{s_t} - \boldsymbol{y_t}|| \leq \epsilon_t),$$

where ϵ_t controls the acceptable level of distance between the observed, $\boldsymbol{y_t}$, and simulated, $\boldsymbol{s_t}$, data, $||\cdot||$ is a suitably chosen norm (which we discuss later), $\boldsymbol{x_t}$ is the extra auxiliary variables that may be required and $\mathbb{1}(\cdot)$ is the indicator function that is equal to one if the argument of the function is true and is equal to zero otherwise. In the following, we present a generic particle filter that is able to efficiently obtain an unbiased estimate of the (approximate) likelihood for a fixed $\boldsymbol{\theta}$ by matching data (within the ABC tolerance) one-at-a-time.

For models with computationally difficult likelihoods, an attractive particle filter is the bootstrap particle filter (Gordon et al., 1993). In this particle filter, only sequential simulation of the data is required and no transition probabilities need to be evaluated. We make use of auxiliary information, $\boldsymbol{x_t}$, in our particle filter to facilitate such sequential simulation when required.

Our implementation of the particle filter maintains exactly N particles throughout via negative binomial resampling until $N+1$ 'matches' are obtained (the rationale for requiring $N+1$ matches is provided below). This type of approach is suggested by Le Gland and Oudjane (2006) in order to avoid degeneracy in the particle filter. The method is referred to as the alive particle filter in Jasra et al. (2013). The natural implementation of this type of filter (resampling until N matches, as done in Le Gland and Oudjane (2006)) results in a biased estimate of the likelihood. However, Jasra et al. (2013) propose a correction to produce an unbiased estimate, which is attractive when using the resulting estimate within an MCMC algorithm (Andrieu and Roberts, 2009). We apply this correction in our paper. The correction involves producing $N+1$ matches and not including the $(N+1)$th match in the particle set. More specifically, denote n_t as the number of simulations from this conditional distribution required to obtain $N+1$ 'matches' for the tth observation. An unbiased estimate of the approximate likelihood is obtained by

$$\hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}) = \prod_{t=1}^{T} \frac{N}{n_t - 1}.$$

It is more numerically stable to sequentially estimate this quantity on the log-scale by considering the following formula

$$\log \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}) = T\log(N) - \sum_{t=1}^{T} \log(n_t - 1).$$

The generic particle filter is shown in Algorithm 1. When exact matching is not plausible and other auxiliary variables are required, then particles need to be defined for $\boldsymbol{s_t}$ and $\boldsymbol{x_t}$ as appropriate. We define these as $\{\boldsymbol{s_t^i}\}_{i=1}^N$ and $\{\boldsymbol{x_t^i}\}_{i=1}^N$. The particle filter keeps track of an approximation to the posterior distribution of the current $\boldsymbol{s_t}$ and/or $\boldsymbol{x_t}$ conditional on the current and previous matches to the observed data that have been performed and the value of $\boldsymbol{\theta}$ via the set of particles.

The particle filter consists of a series of updating and resampling steps. The particle values get updated in light of matching on the next observation. After each iteration the particles are resampled. Due to the uniform ABC discrepancy function employed, the particle weights are always proportional to a constant and do not need to be maintained throughout the algorithm. It is worth noting that even though our particle filter makes use of auxiliary variables, they are only needed to facilitate estimation of the likelihood, which is then used in the MCMC pseudo-marginal sampler detailed in Section 2.2.

We have the following additional remarks about our particle filter:

Remark 1. *If exact sequential matching between observed and simulated data is feasible for observation t, then the auxiliary variable $\boldsymbol{s_t}$ is not required in the algorithm. If the*

Algorithm 1 The alive bootstrap particle filter with auxiliary variables. When $\epsilon = 0$ the algorithm produces an estimate of the true likelihood. When the process is Markovian and exact matching can be performed on all observed variables, the algorithm is technically not a particle filter and Line 8 is not required.

```
Input: A particular value of the parameter, $\boldsymbol{\theta}$, the number of particles, $N$, and the
    time series data, $\boldsymbol{y}$, of length $T$
Output: $\log \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta})$ (i.e. the log of the estimated ABC likelihood)
 1: Initialise $\log \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}) = T\log(N)$
 2: Obtain initial auxiliary simulated data, $\boldsymbol{s_0^i}$, and extra auxiliary variable information,
    $\boldsymbol{x_0^i}$, for $i = 1, \ldots, N$ if necessary
 3: for $t = 1$ to $T$ do
 4:   Set $n_t = 0$
 5:   for $k = 1$ to $N + 1$ do
 6:     matched = ‘no’
 7:     while matched == ‘no’ do
 8:       Resample an index $r$ from the set $\{1, \ldots, N\}$ with equal weights
 9:       Generate $\boldsymbol{s_t^*}$ and $\boldsymbol{x_t^*}$ from $p(\boldsymbol{s_t}, \boldsymbol{x_t}|\boldsymbol{s_{t-1}^r}, \boldsymbol{x_{t-1}^r}, \boldsymbol{\theta})$
10:       Set $n_t = n_t + 1$
11:       if $||\boldsymbol{s_t^*} - \boldsymbol{y_t}|| \leq \epsilon_t$ then
12:         Set $\boldsymbol{s_t^k} = \boldsymbol{s_t^*}$, $\boldsymbol{x_t^k} = \boldsymbol{x_t^*}$ and matched = ‘yes’
13:       end if
14:     end while
15:   end for
16:   Set $\log \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}) = \log \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}) - \log(n_t - 1)$
17: end for
```

model is Markovian and all the variables of the model are observed, then the set of auxiliary variables $\boldsymbol{x_t}$ *for* $t = 0, \ldots, T$ *is not required. If the model is Markov and exact matching is feasible across all time points then the resampling step in Line 8 of Algorithm 1 is not required. In this situation, each probability component in the true likelihood,* $p(\boldsymbol{y_t}|\boldsymbol{y_{t-1}}, \boldsymbol{\theta})$*, is estimated by negative binomial sampling until* $N+1$ *matches are generated and then corrected to ensure the estimate is unbiased. This is the case for the partially observed branching process example in Section 3.2.*

Remark 2. *If the model is Markovian and all the variables of the model are observed (so that the auxiliary variables* $\boldsymbol{x_t}$ *for* $t = 0, \ldots, T$ *are not required), then a slightly different implementation is possible when exact matching is not computationally feasible. Instead of making use of the auxiliary variables* $\boldsymbol{s_t}$ *for* $t = 0, \ldots, T$*, they can be avoided by always simulating forward to time* $t + 1$ *using the observed* $\boldsymbol{y_t}$ *(rather than from the auxiliary variable* $\boldsymbol{s_t}$*). The implementation of this approach is much simpler, but we note that the target distribution will be altered. Although in both implementations the target will not be the true posterior as* $\epsilon > 0$*. We apply this more straightforward implementation in Section 3.2.*

Remark 3. *For a very poor parameter proposal, the model may have very little chance of generating simulated data close to observed data and the particle filter could take an*

Algorithm 2 PMMH algorithm of Andrieu et al. (2010) to simulate from the ABC target. When $\epsilon = 0$ exact inferences are obtained.

```
Input: $\boldsymbol{\theta^0}$ and iters
Output: MCMC output $\boldsymbol{\theta^1}, \ldots, \boldsymbol{\theta^{\mathrm{iters}}}$
Compute $\phi^0 = \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta^0})$ (using Algorithm 1)
for $i = 1$ to iters do
    Propose $\boldsymbol{\theta^*} \sim q(\cdot|\boldsymbol{\theta^{i-1}})$
    Compute $\phi^* = \hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta^*})$ (using Algorithm 1)
    Compute $\alpha = \min\left(1, \frac{\phi^* p(\boldsymbol{\theta^*}) q(\boldsymbol{\theta^{i-1}}|\boldsymbol{\theta^*})}{\phi^{i-1} p(\boldsymbol{\theta^{i-1}}) q(\boldsymbol{\theta^*}|\boldsymbol{\theta^{i-1}})}\right)$
    if $U(0,1) < \alpha$ then
        Set $\phi^i = \phi^*$ and $\boldsymbol{\theta^i} = \boldsymbol{\theta^*}$
    else
        Set $\phi^i = \phi^{i-1}$ and $\boldsymbol{\theta^i} = \boldsymbol{\theta^{i-1}}$
    end if
end for
```

excessively long time or could get stuck. We suggest implementing an intervention in the algorithm which checks the value of n_t in Algorithm 1 and if it becomes excessively large then reject that value of $\boldsymbol{\theta}$ in the PMCMC method specified in Algorithm 2. This could create some bias depending on the problem.

Remark 4. *For time series of low integer values, a reasonable discrepancy function is the L_1-norm (the sum of the absolute differences between every component of the observed and simulated data, since $\boldsymbol{y_t}$ is possibly vector-valued). The L_1-norm has a simple interpretation here; it is the total number of integer units that the simulated data differs from the observed data. In this sense it is easier to test the sensitivity of the results to a change in ϵ_t; a unit increase in ϵ_t allows the simulated data to be away from the observed data by an additional integer unit. An extension of this would be to define a discrepancy function that depends relatively on the observation, e.g. for a scalar observation $||s_t - y_t|| = |s_t - y_t|/(y_t + 1)$, to naturally account for a potentially higher variability for larger counts. It is possible that the optimal discrepancy function is problem dependent and this requires further research.*

2.2 Particle Marginal Metropolis-Hastings Algorithm

The pseudo-marginal algorithm we use is the PMMH algorithm of Andrieu et al. (2010). For completeness, this method is shown in Algorithm 2. The use of MCMC is critical for this approach. A good MCMC proposal distribution will not propose too often parameter values that have a very small chance of generating simulated data close to the observed data, even one-at-a-time as we do here.

A drawback of our approach is that for computational feasibility, a reasonable starting value, $\boldsymbol{\theta^0}$, is required. A very poor starting value will require too much time to generate simulated data close enough to the observed data. In this paper, when the true value of the parameter was unknown, we trialled several starting values until we found one that produced simulated data that 'matched' the observed data. We found this approach feasible for the relatively low dimensional models dealt with in this paper. Alternatively, our method with an increased tolerance matching condition may be used as a pilot run to find a useful part of the parameter space or an adaptive MCMC scheme may be developed that slowly decreases the tolerance. We provide more discussion about this point in Section 4.

The algorithm is based on model simulation and hence is somewhat 'plug-and-play' (He et al., 2010). However, there are some problem specific implementation considerations such as specifying the required auxiliary variables. This becomes clearer in the examples section below. The computational cost of the algorithm is mostly consumed within the particle filter. Therefore, we suggest using a low level implementation (e.g. C, Fortran) of this part of the algorithm.

2.3 Model Choice

Andrieu and Roberts (2009) demonstrate that efficient RJMCMC algorithms can be developed using pseudo-marginal approaches. One advantage of this approach is that it eliminates the need to find a suitable and efficient proposal distribution for a trans-dimensional latent variable, which is very difficult to obtain in complex problems. Secondly, working with an approach that mimics an algorithm on the marginal space of the parameter of interest means that adjacent models are closer in dimension, making it more efficient to jump between them. Drovandi et al. (2014) present a recent application of the utility of performing such marginalisation within RJMCMC algorithms. In this paper we demonstrate that this is also quite a useful approach in dealing with fully Bayesian model choice problems for low integer-valued time series models.

We extend the notation to handle Bayesian model choice problems. We consider a finite set of M models indexed by m, $m \in \{1, \ldots, M\}$. The prior probability of model m is denoted by $p(m)$. The parameter relating to model m, $\boldsymbol{\theta_m}$, has a prior distribution $p(\boldsymbol{\theta_m}|m)$. The (ABC) likelihood for model m is $p_\epsilon(\boldsymbol{y}|\boldsymbol{\theta_m}, m)$ (the true likelihood is obtained when $\epsilon = 0$). Interest is in the (approximate) posterior model probabilities, $p_\epsilon(m|\boldsymbol{y})$, and also in the posterior distributions of the parameter for each model, $p_\epsilon(\boldsymbol{\theta_m}|m, \boldsymbol{y})$, for $m = 1, \ldots, M$.

The RJMCMC algorithm (see Green (1995), and Hastie and Green (2012) for a recent tutorial) can be used for sampling over a transdimensional parameter space. The RJMCMC algorithm that we apply involves both within-model and between-model moves. In our context, within-model moves may be performed as shown in Algorithm 2. The between model move also uses an unbiased estimate of the target based on a proposed parameter value at the proposed model. Andrieu and Roberts (2009) demonstrate that this RJMCMC approach has as its limiting distribution the desired posterior distribution. Care needs to be taken when computing the Metropolis–Hastings ratio, which we detail in Section 3 for the relevant applications as required.

For low dimensional problems, a simpler approach may be to use an importance sampling estimator of the evidence where the importance weights involve an unbiased estimate of the ABC likelihood. Dropping the dependence on m for notational convenience, draw B independent samples from an importance sampling density, $g(\boldsymbol{\theta})$, as $\boldsymbol{\theta}_i$, $i = 1, \dots, B$. An unbiased estimate of the (approximate) evidence, $p_\epsilon(\boldsymbol{y})$ is given by

$$\hat{p}_\epsilon(\boldsymbol{y}) = \frac{1}{B}\sum_{i=1}^{B} \frac{\hat{p}_\epsilon(\boldsymbol{y}|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i)}{g(\boldsymbol{\theta}_i)}, \tag{1}$$

where $p_\epsilon(\boldsymbol{y}) = \int_{\boldsymbol{\theta}} p_\epsilon(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. The reader is referred to Tran et al. (2013) for a justification.

3 Examples

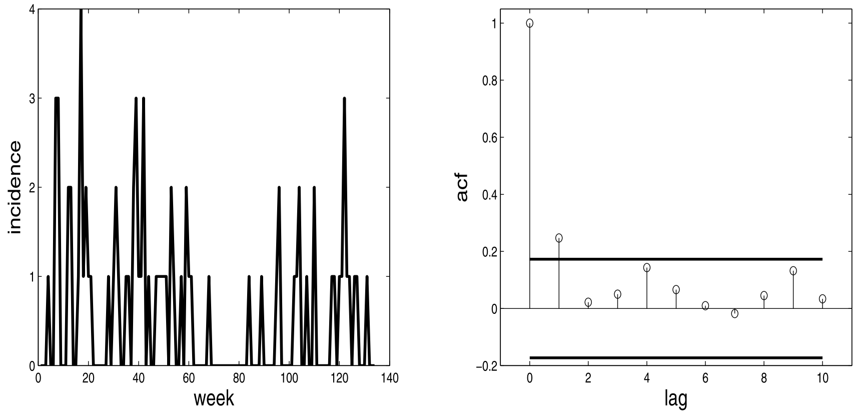
The method is demonstrated below on several models and applications. Where appropriate, our approach is compared with a gold standard to provide a validation of the method. To test for evidence against convergence of our PMCMC algorithm, we used the Heidelberg and Welch diagnostic tests within the Coda R package (Plummer et al., 2006). The diagnostics consist of a stationarity test and a halfwidth test. We found that for all our examples there was no evidence against convergence.

The tuning involved in pseudo-marginal methods is slightly more involved than usual MCMC as the number of particles in the particle filter has to be chosen as well as the MCMC proposal distribution. We performed some pre-runs to obtain tuning parameters that led to reasonable MCMC mixing that was assessed visually. Table 1 in Appendix G of the Supplementary Material (Drovandi et al., 2015) shows the specific details of the proposal distributions that were used for all our examples (both in this paper and the Supplementary Material) for completeness.

3.1 Spread of Nosocomial Pathogens

Drovandi and Pettitt (2008) consider a stochastic model to help explain the spread of the pathogen Methicillin-resistant *Staphylococcus aureus* (MRSA) within a hospital ward. The model suggests that the colonisation of MRSA in patients is facilitated by health-care workers through possible lack of hand hygiene. The model consists of random variables for the number of colonised patients, $Y_p(t)$, and colonised health-care workers, $Y_h(t)$, and assumes a constant ward size of R and a patient to health-care ratio of unity. The data consist of weekly incidence (the number of new cases) of colonisation within the ward, which can be routinely collected by hospitals. These data for a 184 week period at the Princess Alexandra Hospital, Brisbane, Australia are shown in Figure 1(a). The model of Drovandi and Pettitt (2008) includes a counter variable $N(t)$ for the incidence so that there is a correspondence between the model and the data. Note that $N(t)$ is reset to 0 at the beginning of each time interval.

To simplify the model, McBryde et al. (2007) consider a so-called pseudo-equilibrium approximation (also applied in Drovandi and Pettitt (2008)) where the mean of the colonised health-care worker population is considered and the rate of change of this



(a) data

(b) acf

Figure 1: (a) Weekly incidence counts of MRSA colonisation in patients in a hospital ward at the Princess Alexandra Hospital, Brisbane, Australia. (b) Autocorrelation function with estimated 95% CI as solid horizontal lines.

population is set to 0. This provides an equation that deterministically relates the discrete colonised patient variable to the now-continuous colonised health-care population. Drovandi and Pettitt (2011) show that this provides a good approximation. This reduces the original trivariate stochastic process to a bivariate one involving $(Y_p(t), N(t))$. Given the current values of the states, $Y_p(t) = i$ and $N(t) = k$, and a small time interval Δ_t, the probabilities of various combinations of the states at time $t + \Delta_t$ are given by

$$\begin{aligned} P(Y_p = i+1, N = k) &= \mu\sigma(R-i)\Delta_t + o(\Delta_t), \\ P(Y_p = i+1, N = k+1) &= f(\bar{Y}_h)(R-i)\Delta_t + o(\Delta_t), \\ P(Y_p = i-1, N = k) &= \mu'(1-\sigma)i\Delta_t + o(\Delta_t), \end{aligned} \tag{2}$$

where $\mu' = 1/10.6$ and $\mu = 1/4$ are the per-capita colonised and uncolonised patients discharge rates, respectively, $\sigma = 0.03$ is the probability that a patient is colonised upon admission, $f(\bar{Y}_h)$ models the transmission process and $\bar{Y}_h$ is the approximated colonised health-care worker population. Lee et al. (2015) consider two different models for the transmission process. The first is a standard mass action assumption that $f(\bar{Y}_h) = \phi_s \bar{Y}_h$ (referred to as the Standard model) and the second uses the assumption of Greenwood (1931) whereby provided that at least one person is colonised, there is a constant colonisation pressure for the corresponding susceptible group so that $f(\bar{Y}_h) = \phi_g \mathbb{1}(\bar{Y}_h > 0)$ (the Greenwood model hereafter). For illustration, we use the priors, $\phi_s \sim U(0, 0.1)$ and $\phi_g \sim U(0, 0.02)$. The expressions for $\bar{Y}_h(t)$ for the Standard and Greenwood models are given by

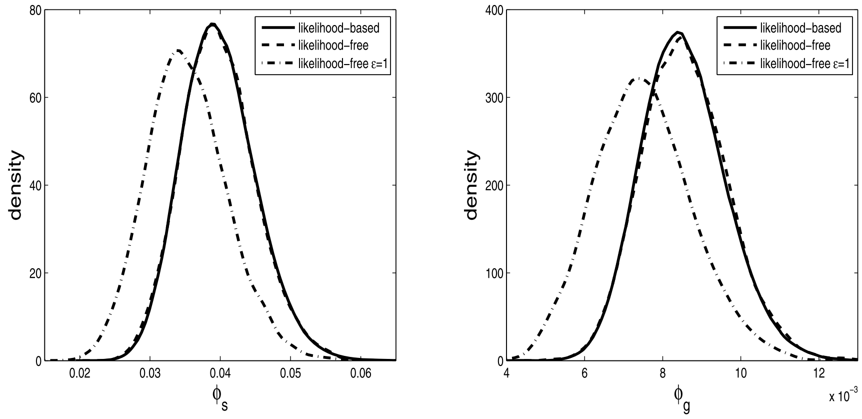
$$\bar{Y}_h(t) = \frac{RY_p(t)}{\frac{hR}{p_{ph}(1-h)} + Y_p(t)} \qquad \text{and}$$

$$\bar{Y}_h(t) = \frac{R\mathbb{1}(Y_p(t) > 0)}{\frac{hR}{p_{ph}(1-h)} + \mathbb{1}(Y_p(t) > 0)},$$

respectively, where $h = 0.59$ is the hand hygiene compliance and $p_{ph} = 0.13$ is the probability of a transmission from patient to health-care worker. Note that for the Greenwood model $\bar{Y}_h(t) = 0$ if $Y_p(t) = 0$.

This example provides a useful application for testing out our method for inferring model parameters and also calculating posterior model probabilities as the likelihood can be calculated here using the approach of Drovandi and Pettitt (2008). It is important to note that the likelihood-based analysis depends upon setting an upper bound for the incidence, L (our simulation based approach does not require this). The maximum value of the incidence in the dataset is 4 (see Figure 1). We found that the results were essentially identical when $L = 5$ and $L = 6$ so we report results based on $L = 5$. For our approach we used $N = 100$ and matched the incidence data exactly. The number of colonised patients is unobserved throughout the process so we introduce $\boldsymbol{x_t} = Y_p(t)$ as an auxiliary variable in the particle filter. We ran our method separately for the Standard and Greenwood models for 100,000 iterations following a burn-in period of 1000 iterations and used normal random walk proposals. To validate our method we ran the likelihood-based approach for 1,000,000 MCMC iterations with the same proposal distributions. Figure 2 demonstrates agreement between our likelihood-free method and the likelihood-based results.

Gamma distributions were fitted to the posterior distribution which were then used as importance distributions to obtain an importance sampling estimate of the evidence for each model (see (1)). For each model, the estimated evidences were based on 10,000 importance samples. We obtained very similar estimates of the evidences for each model when using either the true likelihood or the unbiased simulated likelihood. Both the



(a) Standard (b) Greenwood

Figure 2: (a) Posterior distribution of ϕ_s for the Standard model. (b) Posterior distribution of ϕ_g for the Greenwood model.

likelihood-based and simulation-based approaches produced a posterior model probability, to two decimal places, of 0.85 in favour of the Standard model.

Figure 2 also displays the approximate posteriors when using $\epsilon = 1$, which biases the results towards smaller values of the parameters and increases the uncertainty. We note the strong impact on the posteriors is a result of $\epsilon = 1$ not being a very stringent matching criterion given the very small counts in the dataset. Using a similar process to that above, we found that the approximate posterior model probability was 0.82 for the Standard model.

Finally, we investigated the impact of starting values and between-run variability. For the standard model, we initialised two chains at $\phi_s = 0.01$ and $\phi_s = 0.1$, respectively. From Figure 2(a) it can be seen that these starting values are out in the extreme tails of the posterior support. The results are shown in Appendix A of the Supplementary Material (Drovandi et al., 2015). In short, the method converges from both starting values and there is very little between-run variability in the posterior density estimates for ϕ_s.

Another illustrative example involving infectious diseases is provided in Appendix B of the Supplementary Material (Drovandi et al., 2015).

3.2 Partially Observed Branching Processes

This example demonstrates the utility of pseudo-marginal algorithms for transdimensional simulation problems involving low integer-valued time series models. Consider a standard Galton–Watson branching process where the population size at the end of generation t, Y_t, can be determined via the random sum of random variables

$$Y_t = \sum_{i=0}^{Y_{t-1}} X_{t,i},$$

where $X_{t,i}$ is the number of offspring produced by individual i during the tth generation. Here the set of random variables $\{X_{t,i} : \forall t, i\}$ are independent and identically distributed discrete random variables. We propose to use a flexible non-parametric probability mass function for the bounded offspring distribution, where $p(X_{t,i} = k) = p_k$ for $k = 0, \ldots, K$ ($K < \infty$). Denote the vector of probabilities as $\boldsymbol{\theta_K} = (p_0, \ldots, p_K)$.

The most realistic scenario is that only the population sizes at each generation are observed, and the number of offspring produced by each individual is hidden. We denote the observed data as $\boldsymbol{y} = (y_1, \ldots, y_T)$ where y_t is a realisation of Y_t and T is the total number of generations that the process is observed for. In this case the model is Markovian, but the transition probability, $p(y_t|y_{t-1}, \boldsymbol{\theta})$, involves a summation over all combinations of the offspring realisations that are consistent with the observed data. However, it is very fast to simulate from this distribution and thus the likelihood contribution can be estimated in an unbiased way via simulation.

When the maximum number of offspring possible, K, is finite and fixed, González et al. (2013) show that inference for $\boldsymbol{\theta_K}$ can proceed by introducing a set of latent variables, $\{n_{t,k} : t = 1, \ldots, T, k = 0, \ldots, K\}$, where $n_{t,k}$ represents the number of times that k offspring are produced during the tth generation. The latent variables must satisfy the constraints $\sum_{k=0}^{K} n_{t,k} = y_{t-1}$ and $\sum_{k=0}^{K} k n_{t,k} = y_t$. Using this set-up, a Gibbs sampler can be implemented that draws $\boldsymbol{\theta_K}$ from a Dirichlet distribution conditional on the current set of latent variables, then draws the set of latent variables from a multinomial distribution conditional on the current $\boldsymbol{\theta_K}$ and also constrained to be consistent with the observed data. The reader is referred to González et al. (2013) for more details.

Here we also consider conducting inference for the parameter K, the maximum offspring size, creating a transdimensional simulation problem. Consider a proposal where K is increased by one. It is relatively straightforward to propose a $\boldsymbol{\theta_{K+1}}$ whose elements sum to 1 based on a slight modification to $\boldsymbol{\theta_K}$ (see below). However, we were unable to propose a local update to expand the set of latent variables in such a way that maintains the integrity of the constraints. It appears that the only available proposal is to re-generate the full set of latent variables conditional on the proposed $\boldsymbol{\theta_{K+1}}$. However, the acceptance probability of this move will involve the evaluation of the intractable likelihood.

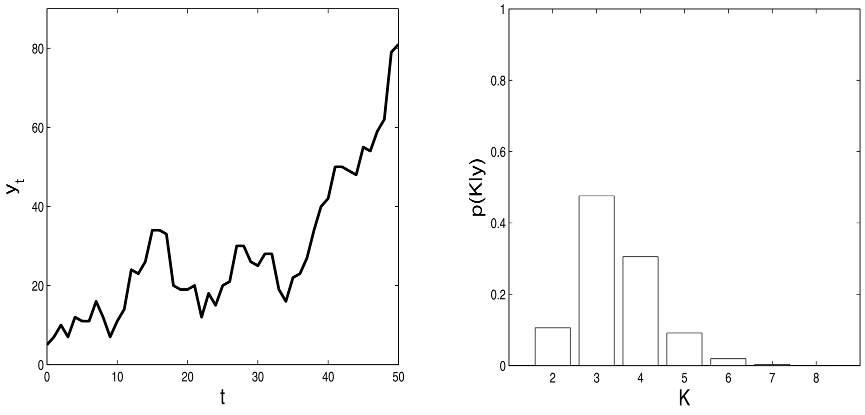
We can estimate the likelihood unbiasedly via simulation, which allows us to overcome this issue, and so we perform inference for K via an RJMCMC method that mimics an algorithm on the marginal space of $\{K, \boldsymbol{\theta_K}\}$. The prior distribution for K is discrete uniform between $K_{\min}$ and $K_{\max}$. The prior for $\boldsymbol{\theta_K}$ is Dirichlet with parameter $\boldsymbol{\alpha_0^K}$ of length $K+1$. We denote the probability density of this prior as $D(\boldsymbol{\theta_K}; \boldsymbol{\alpha_0^K})$ for simplicity. In this work we assign Jeffreys' prior on $\boldsymbol{\theta_K}$ for each value of K, which involves setting all values in $\boldsymbol{\alpha_0^K}$ equal to 0.5. This algorithm consists of both within-model and between-model moves. For the within-model move, a proposed vector for the offspring distribution is obtained via a Dirichlet random walk with a mean given by the current parameter and variance controlled by a single tuning parameter. The between-model move involves a change in dimension. Consider the following birth move to create a $K+1$ model from a K model. Draw a random variable $r \sim U(b, 1)$ (we use $b = 0.9$ after some experimentation) and then the proposed $\boldsymbol{\theta_{K+1}}$ is given by

$$\boldsymbol{\theta_{K+1}} = (p_0^{K+1}, \dots, p_K^{K+1}, p_{K+1}^{K+1}) = (rp_0^K, \dots, rp_K^K, 1-r),$$

where p_i^j denotes the probability of i offspring when the upper bound of the offspring distribution is $K = j$. The absolute value of the Jacobian of this transformation is equal to $1/r^{K+1}$. The likelihood is estimated unbiasedly based on the proposal via simulation. The acceptance probability of this move is

$$\alpha_{K \to K+1} = \min\left(1, \frac{\hat{p}(\boldsymbol{y}|\boldsymbol{\theta_{K+1}}) D(\boldsymbol{\theta_{K+1}}; \boldsymbol{\alpha_0^{K+1}})(1-b) p_{K+1 \to K}}{\hat{p}(\boldsymbol{y}|\boldsymbol{\theta_K}) D(\boldsymbol{\theta_K}; \boldsymbol{\alpha_0^K}) r^{K+1} p_{K \to K+1}}\right), \tag{3}$$

where $p_{i \to j}$ is the probability of proposing a model where $K = j$ from a model where $K = i$. Here we have $p_{i \to j} = 0.5$ if $j = i+1$ or $j = i-1$ and 0 otherwise except for $p_{K_{\min} \to K_{\min}+1} = p_{K_{\max} \to K_{\max}-1} = 1$. The corresponding death move from $K+1$ to K removes the p_{K+1}^{K+1} probability and re-scales the remaining probabilities by a factor of $1/(1 - p_{K+1}^{K+1})$. The Metropolis–Hastings ratio (the second term in brackets in (3)) for the death move is the reciprocal of the one for the birth move.



(a) data

(b) posterior for K

Figure 3: (a) Branching process data. (b) Posterior distribution for K.

For analysis, we simulated $T = 50$ generations of data using the offspring distribution $\boldsymbol{\theta} = (0.35, 0.4, 0.140625, 0.0625, 0.03125, 0.015625)$ and $y_0 = 5$. That is, the true value of K is 5. The data is shown in Figure 3(a). We forced exact matching with the observed data. Further, the model is Markovian so we do not require $\boldsymbol{s_t}$ or $\boldsymbol{x_t}$. The likelihood is simply estimated unbiasedly via simulation. If it took more than 100,000 simulations to match on a particular observation, that parameter configuration was rejected. It was important to implement this intervention here, since the between-model step often made proposals that were not consistent with the data. The smallest value of K that could generate the data with a non-zero probability is 2, so we set $K_{\min} = 2$. The upper limit is set at $K_{\max} = 10$. The algorithm was run for 50,000 iterations starting at the true parameter. Each iteration consisted of a within-model move followed by a between-model move. The acceptance rate of the between model move was 12%. The resulting posterior distribution for K is shown in Figure 3(b). The posterior mean of the average number of offspring produced by an individual (averaged over K) was roughly 1.0572 (true value is 1.071875) with a posterior standard deviation of 0.03. The posterior mean estimate is consistent with the expected number of offspring required to produce 81 individuals after 50 generations, $(81/5)^{0.02} \approx 1.0573$.

To help validate our results we implemented a variant of the Expectation Propagation (EP) ABC algorithm of Barthelmé and Chopin (2014) assuming a Dirichlet distribution for $p(\boldsymbol{\theta_K}|\boldsymbol{y}, K)$ (see Appendix C of the Supplementary Material (Drovandi et al., 2015) for more detail). A by-product of the algorithm is an estimate of the evidence, which we convert into posterior model probabilities (considering values of $K = 2, \ldots, 8$). We found that there was some Monte Carlo variability associated with the results so we repeated the algorithm 10 times for each value of K. In Table 1, we report the mean and standard deviation of the posterior model probability estimates. We found that the EP ABC results produced the same modal value of $K = 3$ as our method but had less support for larger values of K.

K	2	3	4	5	≥ 6	time (hrs)
$\epsilon = 0$	0.11	0.48	0.30	0.09	0.02	5.9
$\epsilon = 1$	0.15	0.48	0.26	0.09	0.02	2.9
$\epsilon = 2$	0.28	0.39	0.24	0.07	0.02	1.9
$\epsilon = 3$	0.33	0.42	0.19	0.05	0.01	1.6
EP ABC	0.16(0.08)	0.66(0.08)	0.14(0.06)	0.03(0.01)	0.00(0.00)	2.4[a]

Table 1: Posterior probabilities, $p(K = k|\boldsymbol{y})$, in the branching process application for different values of ϵ. Note the probabilities have been rounded to two decimal places. Also shown are the computing times in hours. The final row consists of the EP ABC results, where the algorithm was replicated 10 times. Shown is the mean(standard deviation) of the 10 runs.

[a] Total computing time for all 10 repetitions of the EP ABC algorithm.

The results suggest that the data are able to be explained using $K = 3$. We repeated the analysis on a second dataset generated from the model with the same parameter configuration (results not shown). The posterior for K again suggested a mode at $K = 3$ (but this time no support for $K = 2$ and a similar support for $K = 3$ and $K = 4$). There are two possible reasons for this. Firstly, the true value of K is 5; however, p_4 and p_5 are rather small. Secondly, the choice of the prior distribution on $\boldsymbol{\theta_K}$ may be having an effect on the posterior for K. This requires further investigation and is not a focus of this paper. We note that the method recovers the expected number of offspring reasonably well, which is important for predicting extinction of the population.

We also investigated the effect of introducing ϵ on the posterior model probabilities (with the same value of ϵ used for each observation). We note that we used the implementation described in Remark 2. Table 1 presents the results for several ϵ values (based on the same MCMC specifications as for the $\epsilon = 0$ case). It is evident from the table that increasing the value of ϵ biases the results further towards a simpler explanation of the data (that is, a lower value of K). There was also a steady reduction in the computing time as ϵ increased.

Finally, we investigated the impact of starting values and between-run variability. For $K = 5$, we drew $\boldsymbol{\theta_K}$ values randomly until we obtained four parameter sets that produced simulated data that matched with the observed data in the alive particle filter. We then ran our RJMCMC algorithm for 101,000 iterations using the first 1,000 iterations as burn-in. For all four of these starting values we obtained very similar posterior distribution estimates for K (results are shown in Appendix D of the Supplementary Material (Drovandi et al., 2015)) and also for each of the marginals of $\boldsymbol{\theta_K}$. This demonstrates that the results are not sensitive to the initial values provided that a reasonable parameter set is used.

3.3 Integer Autoregressive Moving Average Models

The integer autoregressive moving average (INARMA) model is the discrete version of the popular ARMA model for stationary Gaussian time series. The INARMA(p, q)

model is given by

$$Y_t = \sum_{i=1}^{p} \alpha_i \circ Y_{t-i} + Z_t + \sum_{j=1}^{q} \beta_j \circ Z_{t-j},$$

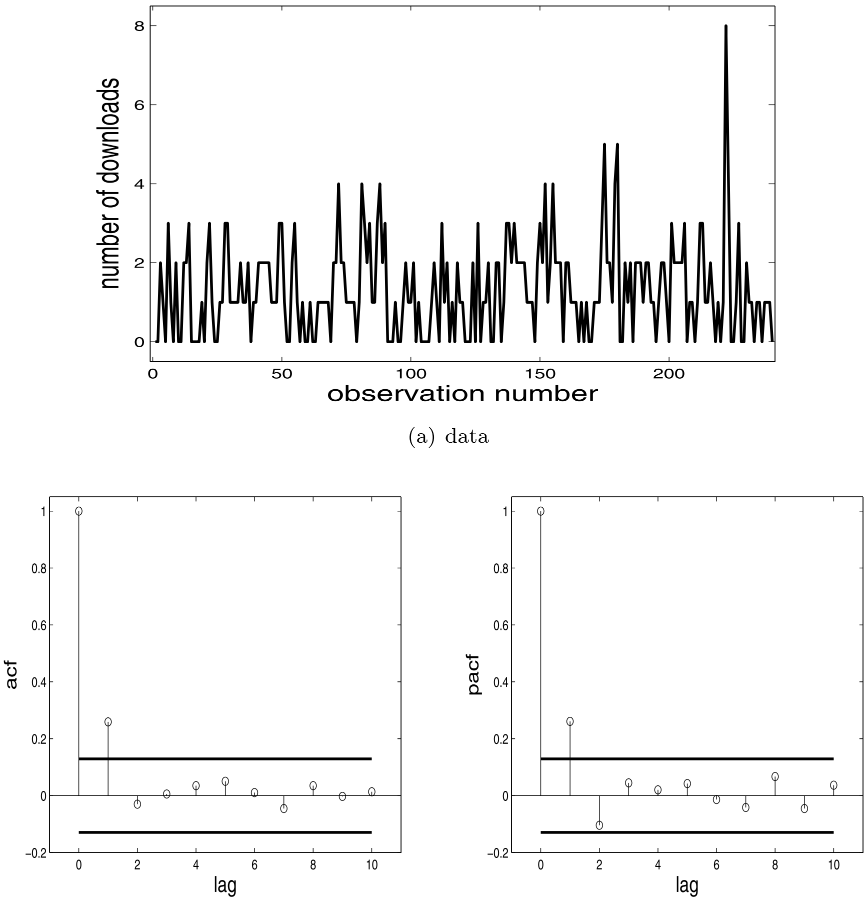
where $\circ$ is the binomial thinning operator (that is, if $W = \alpha \circ Y$, then $W \sim \text{Binomial}(Y, \alpha)$) and Z_t for $t \in \mathcal{N}$ is a sequence of independent and identically distributed discrete random variables. A popular choice is $Z_t \overset{\text{iid}}{\sim} \text{Poisson}(\lambda)$, which is adopted here. The likelihood is cumbersome for all but the INAR(1) model, which involves the convolution of a binomial and a Poisson random variable. Some applications of these types of models are to counts of road accidents (Pedeli and Karlis, 2011) and animal health (Jazi et al., 2012), for example. From a Bayesian perspective, such models have been studied via introducing auxiliary variables to form the complete likelihood and using MCMC for joint posterior simulation and RJMCMC for selecting the model order. Our method attempts to mimic a sampler on the marginal space of $\boldsymbol{\theta} = (p, \alpha_1, \ldots, \alpha_p, q, \beta_1, \ldots, \beta_q)$ only. The approach of White et al. (2015) can handle INAR(p) models but cannot accommodate a moving average component due to the lack of Markov structure in the resulting model. Our particle filter uses auxiliary variables to allow the addition of the moving average component. Below we consider two models that have a first order moving average component. For these cases the auxiliary variable is $\boldsymbol{x_t} = Z_t$. Here we enforce exact matching so that $\boldsymbol{s_t}$ is not required.

We analyse the number of web address downloads at a computer at the University of Würzburg (see Martin et al. (2014)). From the data, autocorrelation function and partial autocorrelation function (see Figure 4), it is evident that the INAR(1), INMA(1) and INARMA(1,1) are all plausible models. However, none of the models are able to replicate the observed value of 8 in this dataset, suggesting that this point is an outlier with respect to these models (see Eduarda Silva and Pereira (2012) for confirmation of this). This observation was changed to a 5 for our purposes. We assume that $Y_0 = 0$ where required.

Here we define the INAR(1), INMA(1) and INARMA(1,1) models as m_1, m_2 and m_3, respectively, and are assumed equally likely a priori. The parameter for each of these models is given by $\boldsymbol{\theta_{m_1}} = (\alpha_{m_1}, \lambda_{m_1})$, $\boldsymbol{\theta_{m_2}} = (\beta_{m_2}, \lambda_{m_2})$ and $\boldsymbol{\theta_{m_3}} = (\alpha_{m_3}, \beta_{m_3}, \lambda_{m_3})$, respectively. Here we implement a pseudo-marginal RJMCMC algorithm in order to estimate the posterior model probabilities. Assume that the current model is m_i. At each iteration a proposal was made to one of the other models, say m_j, with equal probability. The parameter, $\boldsymbol{\theta_{m_j}}$, was drawn independently using the results from an initial within-model run (see below for more details). We denote this proposal density as q_{m_j}. The acceptance probability of this move is given by

$$\alpha_{i \to j} = \min\left(1, \frac{\hat{p}(\boldsymbol{y}|\boldsymbol{\theta_{m_j}}, m_j) p(\boldsymbol{\theta_{m_j}}|m_j) q_{m_i}(\boldsymbol{\theta_{m_i}})}{\hat{p}(\boldsymbol{y}|\boldsymbol{\theta_{m_i}}, m_i) p(\boldsymbol{\theta_{m_i}}|m_i) q_{m_j}(\boldsymbol{\theta_{m_j}})}\right).$$

For all models the parameters on (0,1) had a Uniform$(0, 1)$ prior whilst the λ parameter had an improper prior on $\mathbb{R}^+$ proportional to a constant. The use of an improper



(a) data

(b) acf

(c) pacf

Figure 4: (a) The number of web address downloads every 2 minutes. (b) Autocorrelation function. (c) Partial autocorrelation function. Data source: Martin et al. (2014). The solid horizontal lines in (b) and (c) are the estimated 95% CI.

prior for λ is valid here as we assume the same arbitrary constant for each model and thus have cancelling in the marginal likelihood ratio (Pericchi, 2005). A within-model algorithm was run for the models INAR(1) and INMA(1) using 21,000 iterations whilst 31,000 iterations were used for the INARMA(1,1) model, discarding the first 1000 as burn-in in all cases. Multivariate normal random walks were used.

Parametric statistical models were then fitted to the within-model posterior distributions in order to determine suitable proposal distributions, $q_{m_i}(\boldsymbol{\theta}_{\boldsymbol{m_i}})$ for $i = 1, 2, 3$, to use in the reversible jump. Normal distributions were fitted to the posterior samples of each of the INAR(1) and INMA(1) model parameters. Gamma distributions were fitted for α_{m_3} and β_{m_3} while a normal distribution was fitted for λ_{m_3} for the INARMA(1,1) model. The RJMCMC was run for 50,000 iterations with a starting value used from the within-model runs (so no burn-in was required). The posterior probability of the INAR(1) and INMA(1) models was roughly 0.41 while the INARMA(1,1) had probability 0.18. These posterior distributions can be seen in Figure 5. Note that Figures 5(a) and 5(b) show agreement between our approach and the use of the exact likelihood for the INAR(1) model.

Another example on INARMA models is provided in Appendix E of the Supplementary Material (Drovandi et al., 2015). In this appendix, we also investigate the issue of multiple starting points and between-run variability.

3.4 Autoregulatory Gene Network Example

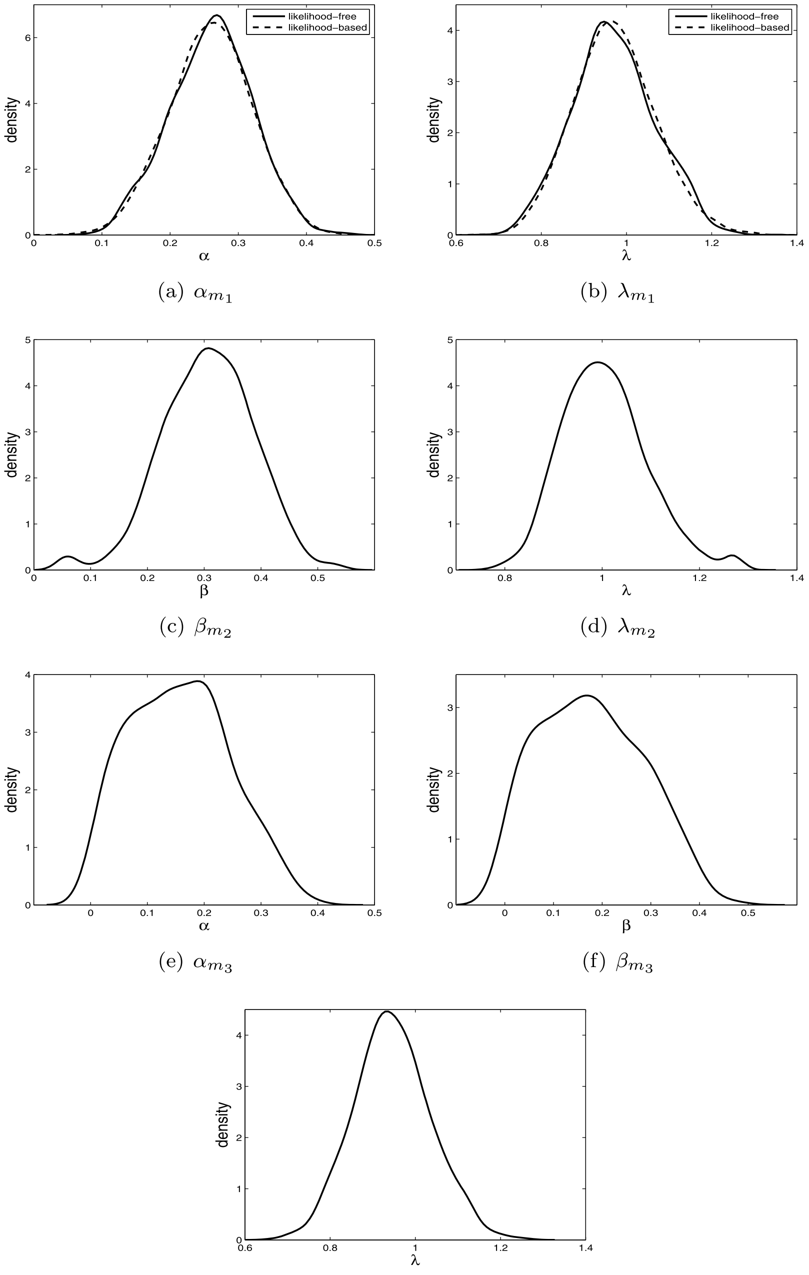
Golightly and Wilkinson (2005) consider a Markov jump process for an autoregulatory gene network that consists of four species (DNA, RNA, P and P_2) (see Golightly and Wilkinson (2005) for more details about this application). There are eight possible reactions in the system, which are given by

$$\begin{aligned}
\mathrm{DNA} + \mathrm{P}_2 &\xrightarrow{c_1 \mathrm{DNA} \times \mathrm{P}_2} \mathrm{DNA} \cdot P_2,\ c_1 = 0.1,\\
\mathrm{DNA} \cdot \mathrm{P}_2 &\xrightarrow{c_2 (k - \mathrm{DNA})} \mathrm{DNA} + \mathrm{P}_2,\ c_2 = 0.7,\\
\mathrm{DNA} &\xrightarrow{c_3 \mathrm{DNA}} \mathrm{DNA} + \mathrm{RNA},\ c_3 = 0.35,\\
\mathrm{RNA} &\xrightarrow{c_4 \mathrm{RNA}} \mathrm{RNA} + \mathrm{P},\ c_4 = 0.2,\\
2\mathrm{P} &\xrightarrow{c_5 0.5 \times \mathrm{P}(\mathrm{P}-1)} \mathrm{P}_2,\ c_5 = 0.1,\\
\mathrm{P}_2 &\xrightarrow{c_6 \mathrm{P}_2} 2\mathrm{P},\ c_6 = 0.9,\\
\mathrm{RNA} &\xrightarrow{c_7 \mathrm{RNA}} \emptyset,\ c_7 = 0.3,\\
\mathrm{P} &\xrightarrow{c_8 \mathrm{P}} \emptyset,\ c_8 = 0.1,
\end{aligned}$$

where the values assigned to the parameters are the true values of the parameters used to simulate some observed data (with $k = 10$ assumed known). Initial values of 5, 8, 8 and 8 of DNA, RNA, P and P_2, respectively, were used to simulate 100 observations where the species populations were observed at 0.5 second time intervals. The initial values were assumed known. Even though the observed counts are small (Figure 6), due to the large number of species, this model does not allow a computationally tractable likelihood function.

We considered two scenarios: the first where all species are observed at these equally spaced time points (referred to as fully observed) and where only the RNA and P species are observed (referred to as partially observed). We assumed that the data are observed without error. As in Fearnhead et al. (2014), we selected half-cauchy priors on the parameters, $p(c_i) \propto 1/(1 + (4c_i)^2)$, $c_i > 0$ for $i = 1, \ldots, 8$. Note that in the MCMC we sampled over the re-parameterised space of $\boldsymbol{\theta}$ where $\theta_i = \log c_i$.

One approach to perform inference for such models is to assume that each species is observed with Gaussian error with a standard deviation of σ (Holenstein, 2009).



(a) α_{m_1}

(b) λ_{m_1}

(c) β_{m_2}

(d) λ_{m_2}

(e) α_{m_3}

(f) β_{m_3}

(g) λ_{m_3}

Figure 5: Posterior densities for when the INAR(1), INMA(1) and INARMA(1,1) models are fitted to the downloads data in Figure 4. (a) and (b) Posteriors for INAR(1) model. (c) and (d) Posteriors for INMA(1) model. (e), (f) and (g) Posteriors for INARMA(1,1) model.

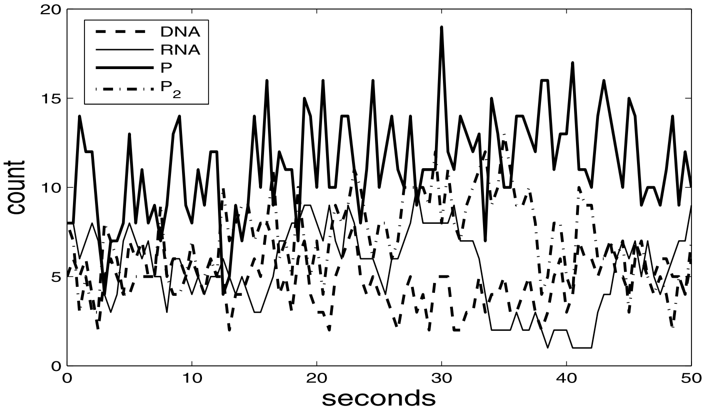
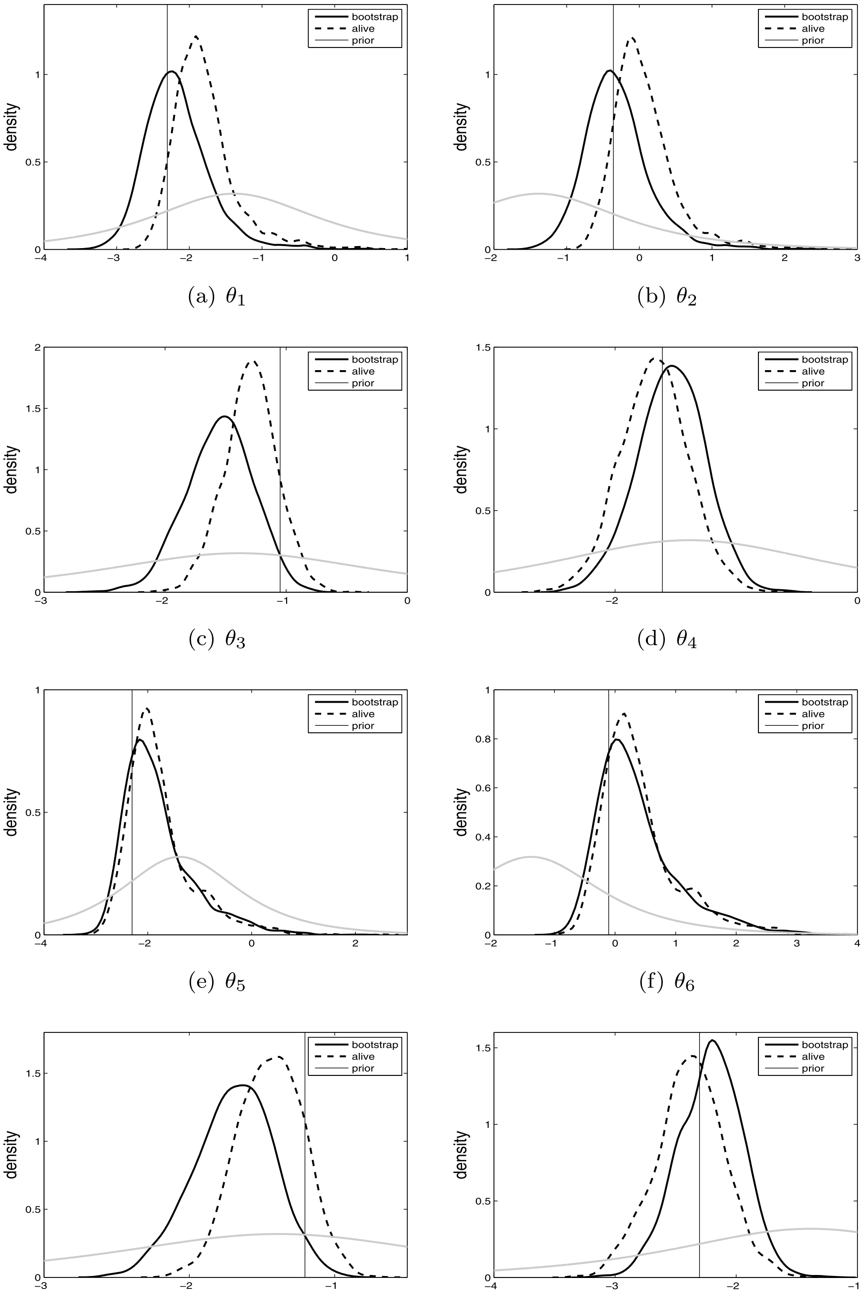


Figure 6: Data for the autoregulatory gene network example.

This allows a standard particle MCMC approach with a bootstrap filter to be applied. More accurate inferences are obtained with a lower value of σ but the (approximate) likelihood is more computationally demanding to estimate precisely. We compare our approach (called alive) with this particle MCMC method (referred to as bootstrap).

In the completely observed case, exact matching on the four species simultaneously was not feasible. Here we used $\epsilon_t = 1$ for all t. The alive approach used $N = 100$ particles and we rejected any proposal that needed at least 100,000 simulations to produce a match on any observation. For the bootstrap method, we selected the number of particles so that the overall computation time was similar to the alive approach. The value of σ was chosen so that the bootstrap likelihood was estimated to a similar precision as the alive likelihood (standard deviation of approximately 1 for the log-likelihood based on the true parameter). Thus for the bootstrap method we used $N = 6000$ and $\sigma = 0.85$. We ran both methods for 100,000 iterations starting at the true value of the parameter. Both approaches required about 75 hours of computation. Posterior distributions for the parameters (with the true values and priors overlaid) are shown in Figure 7. It can be seen that the alive method results in more precise inferences for the parameters θ_3 and θ_7 whereas the results for the other parameters are relatively similar.

When the data are partially observed, we found that it was feasible to produce exact matching and thus set $\epsilon_t = 0$ for all t. Again $N = 100$ was used for the alive approach and we rejected any proposal that needed at least 100,000 simulations to produce a match on any observation. Using a similar process to above, we chose $N = 6000$ and $\sigma = 0.45$ for the bootstrap method. The computing times for the alive and bootstrap methods were 98 and 93 hours, respectively. The extra computation time compared with the completely observed scenario could be attributed to an increase in time to simulate from the model as the less informative data allows for larger values of the rate parameters. The posterior approximations are shown in Figure 8. The bootstrap and alive approaches produced similar results for the parameters θ_1, θ_2, θ_3, θ_5 and θ_6 but due to the exact matching the alive approach resulted in more precise inferences for the parameters θ_4, θ_7 and θ_8.



(a) θ_1

(b) θ_2

(c) θ_3

(d) θ_4

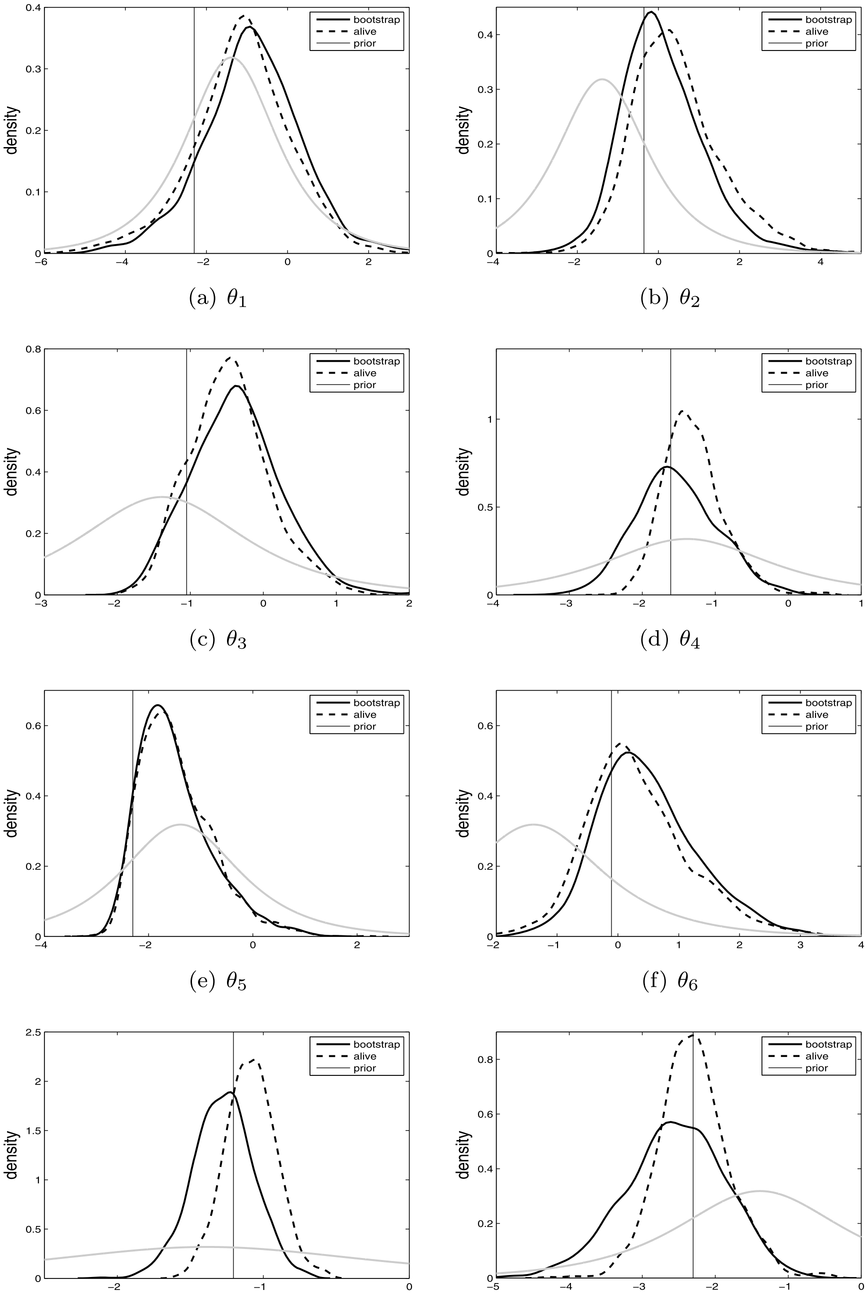
(e) θ_5

(f) θ_6

(g) θ_7

(h) θ_8

Figure 7: Posterior densities for the gene network example with all species observed. Results are shown for the bootstrap method (solid lines) and alive method (dashed lines). The prior distributions are shown in grey.



(a) θ_1

(b) θ_2

(c) θ_3

(d) θ_4

(e) θ_5

(f) θ_6

(g) θ_7

(h) θ_8

Figure 8: Posterior densities for the gene network example with DNA and P species observed only. Results are shown for the bootstrap method (solid lines) and alive method (dashed lines). The prior distributions are shown in grey.

4 Discussion

Here a general Bayesian methodology was presented to perform inference on parameters of models for low integer-valued time series data with intractable likelihoods. The method can work on partially observed processes and non-Markovian models through the use of auxiliary variables in the particle filter. Since the proposals can be made directly on the space of $\boldsymbol{\theta}$ alone, efficient reversible jump proposals can be developed straightforwardly, which leads to robust fully-Bayesian model comparisons. When the observed data cannot be matched exactly, the algorithm allows for ABC inference by introducing a tolerance. No summary statistics are required. Our algorithm represents an alternative to the data augmentation approaches of Neal and Subba Rao (2007) and Enciso-Mora et al. (2009) for INARMA models.

As was seen in one of the examples, the method can fail if there are outliers (with respect to a particular model). The algorithm can become stuck constantly trying to generate data from the model to match the observed outlier. However, if an outlier exists it may be that there is sufficient reason to remove such a data point or possibly that the model is not an adequate representation of the true underlying process. However, the method we propose can be used to identify if a model is not appropriate for the data and identify outliers in the data. For example, the outlier in the downloads data (Figure 4) was quickly noticed. See also one of the examples in Appendix F of the Supplementary Material (Drovandi et al., 2015). A real advantage of these methods is that they have a built in predictive check so that poorly fitting models are discovered quickly. A related issue is that in our RJMCMC branching process application in Section 3.2, the between model proposals often were not consistent with the data, as is often the case in RJMCMC applications. Currently we are working on including the alive particle filter into the SMC^2 algorithm of Chopin et al. (2013) so that the evidence of each model can be estimated individually and no between-model moves are required.

As mentioned in the introduction of the paper, the method can only be applied to a certain class of non-Markovian models. A model not within this class is the discrete time series models of Cui and Lund (2009). Here independent renewal processes are run in parallel and are superposed. To simulate the model, each renewal process must be simulated to at least the current observation but can often go beyond. Applying our method to this class of models may accept simulations that match the current observation but produce a random mis-match in the future. Despite this, we demonstrate in Appendix F of the Supplementary Material (Drovandi et al., 2015) that we are able to obtain good approximate Bayesian inference for such models.

As has been stated in the paper, our method relies upon a starting value where it is computationally feasible to match simulated with observed data. When the model provides a reasonable fit to the data, this amounts to finding a parameter value within the space of non-negligible posterior support. We note that there are other MCMC methods that suffer from this same problem. For example, MCMC ABC (Marjoram et al., 2003) requires the same kind of stringent matching condition. In addition, in pseudo-marginal methods, it is often computationally difficult to obtain precise estimates of the likelihood in regions of negligible posterior support. This can lead to grossly overestimated likelihoods that result in the chain getting 'stuck' when only local MCMC moves are applied. In this paper, several starting values were tested until a suitable one was found. Our other suggestion was to begin with a more relaxed matching tolerance. Here we suggest other approaches that might be useful. Approximate and fast methods, such as EP ABC, may be applied first to find a starting value. However, methods like EP ABC are not as widely applicable as our method. Our current work on incorporating the alive filter within SMC^2 (Chopin et al., 2013) may also help address this issue since the method uses a population of 'particles' across the parameter space.

The approach proposed in this paper is most suited to datasets where the time series consists of very low integer values such that it is not very computationally intensive to obtain close matches between the observed and simulated data. Furthermore, our method is most appropriate for models where sequential simulation is possible (perhaps via the introduction of some auxiliary variables) and where events occurring in the future beyond the current simulation time do not need to be stored (however, we show in Appendices A and D of the Supplementary Material (Drovandi et al., 2015) that useful posterior approximations can be obtained for such models). The method may be applicable to data with large counts or continuous data, but would require some development. Clearly, exact matching will be computationally prohibitive in such situations. Therefore, ABC will be required, potentially with a different discrepancy function (for example, a discretised Gaussian kernel) utilised here that is more appropriate for large values. The variance of the data at some time points may be greater than others, and this needs to be considered. We are currently exploring this in other research. Partially observed data and non-Markovian models could be handled in the same way via auxiliary variables in the particle filter. There are, of course, situations where models produce near chaotic data (see, for example, Wood (2010)), and methods which attempt to match simulated with observed datasets are not feasible.

Supplementary Material

Supplementary Material for Exact and Approximate Bayesian Inference for Low Integer-Valued Time Series Models with Intractable Likelihoods (DOI: 10.1214/15-BA950SUPP; .pdf).

References

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342. MR2758115. doi: http://dx.doi.org/10.1111/j.1467-9868.2009.00736.x. 326, 328, 331

Andrieu, C. and Roberts, G. O. (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *The Annals of Statistics*, 37(2): 697–725. MR2502648. doi: http://dx.doi.org/10.1214/07-AOS574. 328, 329, 332

Barthelmé, S. and Chopin, N. (2014). "Expectation propagation for likelihood-free inference." *Journal of the American Statistical Association*, 109(505): 315–333.

MR3180566. doi: http://dx.doi.org/10.1080/01621459.2013.864178. 326, 327, 338

Blum, M. G. B., Nunes, M. A., Prangle, D., and Sisson, S. A. (2013). "A comparative review of dimension reduction methods in approximate Bayesian computation." *Statistical Science*, 28: 189–208. MR3112405. 326

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). "SMC^2: an efficient algorithm for sequential analysis of state space models." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3): 397–426. MR3065473. doi: http://dx.doi.org/10.1111/j.1467-9868.2012.01046.x. 347, 348

Cui, Y. and Lund, R. (2009). "A new look at time series of counts." *Biometrika*, 96(4): 781–792. MR2564490. doi: http://dx.doi.org/10.1093/biomet/asp057. 347, 352

Drovandi, C. C. and Pettitt, A. N. (2008). "Multivariate Markov process models for the transmission of Methicillin-resistant *Staphylococcus aureus* in a hospital ward." *Biometrics*, 64(3): 851–859. MR2526636. doi: http://dx.doi.org/10.1111/j.1541-0420.2007.00933.x. 327, 333, 335

— (2011). "Using approximate Bayesian computation to estimate transmission rates of nosocomial pathogens." *Statistical Communications in Infectious Diseases*, 3(1): 2. MR2819387. doi: http://dx.doi.org/10.2202/1948-4690.1025. 334

Drovandi, C. C., Pettitt, A. N., Henderson, R. D., and McCombe, P. A. (2014). "Marginal reversible jump Markov chain Monte Carlo with application to motor unit number estimation." *Computational Statistics & Data Analysis*, 72: 128–146. MR3139352. doi: http://dx.doi.org/10.1016/j.csda.2013.11.003. 332

Drovandi, C. C., Pettitt, A. N., and McCutchan, R. A. (2015). "Supplementary Material for Exact and Approximate Bayesian Inference for Low Integer-Valued Time Series Models with Intractable Likelihoods." *Bayesian Analysis*. doi: http://dx.doi.org/10.1214/15-BA950SUPP. 333, 336, 338, 339, 342, 347, 348

Eduarda Silva, M. and Pereira, I. (2012). "Detection of additive outliers in Poisson Integer-valued autoregressive time series." arXiv:1204.6516v1. 340

Enciso-Mora, V., Neal, P., and Subba Rao, T. (2009). "Efficient order selection algorithms for integer-valued ARMA processes." *Journal of Time Series Analysis*, 30(1): 1–18. MR2488633. doi: http://dx.doi.org/10.1111/j.1467-9892.2008.00592.x. 327, 347

Fearnhead, P., Giagos, V., and Sherlock, C. (2014). "Inference for reaction networks using the linear noise approximation." *Biometrics*, 70(2): 457–466. MR3258050. doi: http://dx.doi.org/10.1111/biom.12152. 342

Golightly, A. and Wilkinson, D. J. (2005). "Bayesian inference for stochastic kinetic models using a diffusion approximation." *Biometrics*, 61(3): 781–788. MR2196166. doi: http://dx.doi.org/10.1111/j.1541-0420.2005.00345.x. 342

González, M., Gutiérrez, C., Martínez, R., and Inés, M. (2013). "Bayesian inference for controlled branching processes through MCMC and ABC methodologies." *Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, 107(2): 459–473. MR3199723. 336, 337

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." In: *Radar and Signal Processing, IEE Proceedings F*, volume 140, 107–113. 328

Green, P. J. (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82(4): 711–732. MR1380810. doi: http://dx.doi.org/10.1093/biomet/82.4.711. 332

Greenwood, M. (1931). "On the statistical measure of infectiousness." *The Journal of Hygiene*, (3): 336–351. 334

Hastie, D. I. and Green, P. J. (2012). "Model choice using reversible jump Markov chain Monte Carlo." *Statistica Neerlandica*, 66(3): 309–338. MR2955422. doi: http://dx.doi.org/10.1111/j.1467-9574.2012.00516.x. 332

He, D., Ionides, E. L., and King, A. A. (2010). "Plug-and-play inference for disease dynamics: measles in large and small populations as a case study." *Journal of the Royal Society Interface*, 7(43): 271–283. 332

Holenstein, R. (2009). "Particle Markov chain Monte Carlo." Ph.D. thesis, The University Of British Columbia. 342

Jasra, A., Lee, A., Yau, C., and Zhang, X. (2013). "The alive particle filter." arXiv:1304.0151. 326, 329

Jazi, M. A., Jones, G., and Lai, C.-D. (2012). "First-order integer valued AR processes with zero inflated Poisson innovations." *Journal of Time Series Analysis*, 33(6): 954–963. MR2991911. doi: http://dx.doi.org/10.1111/j.1467-9892.2012.00809.x. 340

Le Gland, F. and Oudjane, N. (2006). "A Sequential Particle Algorithm that Keeps the Particle System Alive." In: *Stochastic Hybrid Systems: Theory and Safety Critical Applications*, volume 337 of *Lecture Notes in Control and Information Sciences*, Springer, 351–389. MR2246656. doi: http://dx.doi.org/10.1007/11587392_11. 329

Lee, X. J., Drovandi, C. C., and Pettitt, A. N. (2015). "Model choice problems using approximate Bayesian computation with applications to pathogen transmission data sets." *Biometrics*, 71(1): 198–207. 334

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). "Markov chain Monte Carlo without likelihoods." *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328. 347

Martin, V. L., Tremayne, A. R., and Jung, R. C. (2014). "Efficient method of moments estimators for integer time series models." *Journal of Time Series Analysis*, 35(6): 491–516. 340, 341

McBryde, E. S., Pettitt, A. N., and McElwain, D. L. S. (2007). "A stochastic mathematical model of Methicillin-resistant *Staphylococcus aureus* transmission in an intensive care unit: predicting the impact of interventions." *Journal of Theoretical Biology*, 245(3): 470–481. MR2306474. doi: http://dx.doi.org/10.1016/j.jtbi.2006.11.008. 333

McKinley, T. J., Ross, J. V., Deardon, R., and Cook, A. R. (2014). "Simulation-based Bayesian inference for epidemic models." *Computational Statistics & Data Analysis*, 71: 434–447. MR3131981. doi: http://dx.doi.org/10.1016/j.csda.2012.12.012. 327

Moler, C. and van Loan, C. (2003). "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later." *SIAM Review*, 45(1): 3–49. MR1981253. doi: http://dx.doi.org/10.1137/S00361445024180. 327

Neal, P. and Subba Rao, T. (2007). "MCMC for integer-valued ARMA processes." *Journal of Time Series Analysis*, 28(1): 92–110. MR2332852. doi: http://dx.doi.org/10.1111/j.1467-9892.2006.00500.x. 327, 347

Pedeli, X. and Karlis, D. (2011). "A bivariate INAR(1) process with application." *Statistical Modelling*, 11(4): 325–349. MR2906704. doi: http://dx.doi.org/10.1177/1471082X1001100403. 340

Pericchi, L. R. (2005). "Model selection and hypothesis testing based on objective probabilities and Bayes factors." In: *Bayesian thinking: modeling and computation*, Elsevier B. V., 115–149. MR2490524. doi: http://dx.doi.org/10.1016/S0169-7161(05)25004-6. 341

Persing, A. and Jasra, A. (2014). "Twisting the alive particle filter." *Methodology and Computing in Applied Probability*, in press. doi: http://dx.doi.org/10.1007/s11009-014-9422-7. 326

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). "CODA: Convergence Diagnosis and Output Analysis for MCMC." *R News*, 6(1): 7–11. http://CRAN.R-project.org/doc/Rnews/ 333

Sidje, R. B. (1998). "Expokit: a software package for computing matrix exponentials." *ACM Transactions on Mathematical Software (TOMS)*, 24(1): 130–156. 327

Tran, M.-N., Scharth, M., Pitt, M. K., and Kohn, R. (2013). "Importance sampling squared for Bayesian inference in latent variable models." arXiv:1309.3339v3. 333

White, S. R., Kypraios, T., and Preston, S. P. (2015). "Piecewise approximate Bayesian computation: fast inference for discretely observed Markov models using a factorised posterior distribution." *Statistics and Computing*, 25(2): 289–301. MR3306707. doi: http://dx.doi.org/10.1007/s11222-013-9432-2. 326, 327, 340

Wood, S. N. (2010). "Statistical inference for noisy nonlinear ecological dynamic systems." *Nature*, 466(7310): 1102–1104. 348

Acknowledgments

The authors were supported by the Australian Research Council (ARC) DP110100159 and the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS). The authors are grateful to Candice Hincksman for useful discussions on the model of Cui and Lund (2009). The first author is grateful to Simon Barthelmé for consultation about the EP ABC algorithm. Comments and suggestions from an associate editor and two referees have led to an improved paper. The RJMCMC proposal in Section 3.2 was suggested by a referee.