*Ecology*, 93(2), 2012, pp. 256–263


Reports

# Fitting complex population models by combining particle filters with Markov chain Monte Carlo

Jonas Knape[1] and Perry de Valpine

*Department of Environmental Science, Policy and Management, 137 Mulford Hall Number 3114, University of California, Berkeley, California 94720 USA*

*Abstract*. We show how a recent framework combining Markov chain Monte Carlo (MCMC) with particle filters (PFMCMC) may be used to estimate population state-space models. With the purpose of utilizing the strengths of each method, PFMCMC explores hidden states by particle filters, while process and observation parameters are estimated using an MCMC algorithm. PFMCMC is exemplified by analyzing time series data on a red kangaroo (*Macropus rufus*) population in New South Wales, Australia, using MCMC over model parameters based on an adaptive Metropolis-Hastings algorithm. We fit three population models to these data; a density-dependent logistic diffusion model with environmental variance, an unregulated stochastic exponential growth model, and a random-walk model. Bayes factors and posterior model probabilities show that there is little support for density dependence and that the random-walk model is the most parsimonious model. The particle filter Metropolis-Hastings algorithm is a brute-force method that may be used to fit a range of complex population models. Implementation is straightforward and less involved than standard MCMC for many models, and marginal densities for model selection can be obtained with little additional effort. The cost is mainly computational, resulting in long running times that may be improved by parallelizing the algorithm.

*Key words: Markov chain Monte Carlo (MCMC); observation error; particle filter; population dynamics; state-space models.*

## Introduction

Biologists have made great progress in using state-space models for the estimation of dynamic processes from noisy data, such as for population dynamics (King et al. 2009), animal movement (Newman 1998), or individual growth (Fujiwara et al. 2005). These models make a distinction between observed data and true system states by incorporating multiple sources of variation, in particular, environmental stochasticity (process noise) and sampling variation (observation error) (de Valpine and Hastings 2002, Calder et al. 2003). The main challenge of estimating state-space models has been the development of algorithms that appropriately explore the space of possible true system states from which the data could have been sampled. As a result, Markov chain Monte Carlo (MCMC) algorithms have become a popular solution, facilitated by software such as BUGS, including its variants WinBUGS (Lunn et al. 2000), OpenBUGS (Thomas et al. 2006), and JAGS (*available online*).[2]

Particle filters (Cappé et al. 2007), also known as sequential Monte Carlo (SMC; Doucet et al. 2001), are another suite of Monte Carlo techniques for analyzing state-space models that are less commonly used in ecology. Ecological applications of particle filters have focused on, e.g., estimation of structured population models with measurement error (Trenkel et al. 2000, Buckland et al. 2004, Newman et al. 2006), individual growth models (Fujiwara et al. 2005), analyses of movements and positions from data on tagged animals (Andersen et al. 2007, Dowd and Joy 2011), predictions from dynamic ecosystem models (Dowd 2006), and inference about disease dynamics (Ionides et al. 2006). A basic particle filter for estimating hidden states or a likelihood value is easy to implement as long as one can simulate from the model of interest (Newman et al. 2009). Particle filters were originally designed to explore the hidden states of state-space models for given parameter values (Gordon et al. 1993). They can also perform parameter estimation, but there are some practical difficulties associated with this (Cappé et al. 2007). By contrast, Bayesian parameter estimation for state-space models is conceptually simple using MCMC, but programming MCMC can be cumbersome and requires implementing a complete suite of Metropolis-Hastings or Gibbs sampler steps to cover all dimensions. Simple implementations may also suffer from poor mixing due to high posterior correlations between hidden states at successive points in time and/or between hidden states and parameters. In such cases, customized MCMC algorithms can yield more efficient runs, but at

Manuscript received 3 May 2011; revised 9 September 2011; Accepted 10 October 2011. Corresponding Editor: K. B. Newman.
[1] E-mail: jknape@berkeley.edu
[2] http://mcmc-jags.sourceforge.net

the cost of large implementation effort (Newman et al. 2009).

Here, we explored an approach to combining particle filters and MCMC in Bayesian inference (PFMCMC; Fernández-Villaverde and Rubio-Ramírez 2007, Andrieu et al. 2010, Jones et al. 2010) and use it to fit a continuous-time nonlinear state-space model to population data on red kangaroos (*Macropos rufus*). MCMC is used for the model parameters, while a particle filter is used for the system states. When the MCMC algorithm proposes new parameters for acceptance or rejection in a Metropolis-Hastings algorithm, the likelihood is evaluated using a particle filter. The resulting MCMC will have few dimensions, one for each parameter only, but will be computationally demanding because a particle filter with a large number of particles needs to be run for each MCMC update. PFMCMC, therefore, offers a potential trade-off of saving human implementation time at the expense of computing time. When implemented with independent proposal densities (which we do here following an adaptive MCMC burn-in period), this approach has the added advantage of approximating the marginal likelihoods needed for Bayes factors and model selection. PFMCMC algorithms have recently been used to estimate nonlinear discrete time models for population data (Peters et al. 2010) and a predator–prey model for simulated phytoplankton and zooplankton data (Jones et al. 2010).

## Methods

To illustrate the method we use a general state-space notation and therefore consider data in the form of a sequence of vectors or scalars, $y_1, y_2, \ldots, y_T$. For population models these may typically be counts or indices of the number of individuals in a population, possibly with stage structure. The state-space model is defined via hidden stochastic states (scalar or vector valued), $x_1, x_2, \ldots, x_T$, representing the true, but unobserved number of individuals or densities. The hidden states are assumed to follow a population process that has the Markov property $x_{t+1} \mid x_t \sim f_\theta(x_{t+1} \mid x_t)$, where $f_\theta(x_{t+1} \mid x_t)$ are transition densities dependent on a vector of unknown parameters $\theta$. For example, $f_\theta(x_{t+1} \mid x_t)$ would typically represent a population model with environmental stochasticity. The first hidden state $x_1$ follows an initial density $f_\theta(x_1)$. Note that in order to avoid excessive notation, we use $f_\theta$ to denote several different distributions depending on the context. The meaning of the distribution will be clear from its arguments.

Given the hidden state at time $t$, the data at time $t$ are distributed according to an observation distribution $f_\theta(y_t \mid x_t)$, also potentially dependent on the parameter vector $\theta$, so that $y_t \mid x_t \sim f_\theta(y_t \mid x_t)$. The observations corresponding to different points in time are independent given the hidden states. Following standard convention, the collection of all data up to time $t$ will be denoted $y_{1:t}$ ($=\{y_i\colon 1 \leq i \leq t\}$), and similarly, a collection of hidden states up to time $t$ will be denoted $x_{1:t}$. The method we describe uses an MCMC algorithm for the parameters, $\theta$, and so is most easily applied to Bayesian problems where we will assume a prior distribution, $\theta \sim \pi(\theta)$. We note, however, that methods using MCMC output to obtain maximum likelihood estimates could be applied to obtain frequentist estimates (reviewed in de Valpine, *in press*).

The method is outlined in three sections: First we introduce particle filters, then we describe a general particle filter Metropolis-Hastings method, and third, we describe an adaptive Metropolis-Hastings scheme.

### *Particle filters to approximate likelihoods*

Particle filters use simulations to approximate the so-called filtered densities of the hidden states $f_\theta(x_t \mid y_{1:t})$, i.e., the distribution of the hidden state at time $t$, given all information up to time $t$. These densities also allow approximation of the probability of each observation given the previous observations, $f_\theta(y_t \mid y_{1:t-1})$, and the product of these approximates the likelihood. The Kalman filter (Dennis et al. 2006) does this analytically for linear state-space models with normally distributed noises, while the particle filter can be used for more general cases.

The algorithm for the basic particle filter is given as follows:

Initiate by simulating independent vectors $x_{1i}$ from the distribution $f_\theta(x_1)$ for $i = 1, \ldots, N$. The $x_{1i}$'s are called particles. For $t = 1{:}T$, there are three steps.

1) Compute weights for each particle as

$$w_{ti} = f_\theta(y_t \mid x_{ti}). \qquad (1)$$

The weights represent how well each particle at time $t$ corresponds to the observation at time $t$. For a state-space model of population dynamics, each particle is a hidden (true) population vector at one time.

2) Resample the particles according to the weights by drawing with replacement from the particles $x_{ti}$ with probabilities, $p_{ti}$, equal to the normalized weights, $p_{ti} = w_{ti}/\sum_i w_{ti}$. The resampled particles $x'_{ti}$ are approximate draws from the filtered density $f_\theta(x_t \mid y_{1:t})$.

3) Simulate new particles for the next time step starting at the resampled particles and using the transition density. That is, for $i = 1, \ldots, N$, simulate $x_{t+1i}$ from the distribution $f_\theta(x_{t+1} \mid x'_{ti})$. In other words, the population model is used to simulate the population vector forward one time step for each population vector drawn in the resample step. The new particles are approximate draws from the predictive density $f_\theta(x_{t+1} \mid y_{1:t})$, the density of the state of the population at time $t+1$ given all observations up to and including time $t$.

An estimate of the normalized likelihood of the data at the parameter value $\theta$ can be computed directly from the weights of the particle filter by factoring the likelihood as $l(\theta) = p(y_{1:T} \mid \theta) = \Pi_t\, f_\theta(y_t \mid y_{1:t-1})$. The

Reports

factors in this product are estimated through

$$\begin{aligned} f_\theta(y_t \mid y_{1:t-1}) &= \int f_\theta(y_t \mid x_t) f_\theta(x_t \mid y_{1:t-1})\, dx_t \\ &= E_{x_t \mid y_{1:t-1}}(f_\theta(y_t \mid x_t)) \approx 1/N \sum_{i=1}^{N} f_\theta(y_t \mid x_{ti}) \\ &= 1/N \sum_{i=1}^{N} w_{ti} = \hat{f}_\theta(y_t \mid y_{1:t-1}) \end{aligned} \quad (2)$$

where we have used the fact that particles $x_{ti}$ are approximately distributed according to $f_\theta(x_t \mid y_{1:t-1})$ to estimate the expectation by the sample mean. Hence, we may estimate the likelihood as

$$\hat{l}(\theta) = \hat{p}(y_{1:T} \mid \theta) = \prod_{t=1}^{T} \hat{f}_\theta(y_t \mid y_{1:t-1})$$

and it can be shown that this is unbiased for $l(\theta)$ (Doucet et al. 2001).

It is important to note that the requirements for implementing the above algorithm for a specific model are straightforward. Essentially, the requirements boil down to the following: (1) Random numbers can be generated from the distributions $f_\theta(x_{t+1} \mid x'_{ti})$ and (2) the densities of the observations $f_\theta(y_t \mid x_{ti})$ can be computed. Hence, even if the densities $f_\theta(x_{t+1} \mid x'_{ti})$ are not available, the particle filter can be implemented; put more simply, it needs to be possible to simulate the dynamics, but not necessarily to calculate the probability of each outcome. An example of such a situation will be given below in the subsection *Example: diffusion models of kangaroo population dynamics*. (Forward simulation from a population model can also be used to generate block updates for MCMC algorithms [Harrison et al. 2011].)

Particle degeneracy, and hence, inefficiency of the filter, occurs if the drawn particles $x_{ti}$ are largely inconsistent with the observed data $y_t$. This may happen if, for instance, the state projections, $f_\theta(x_t \mid x_{t-1})$, are very informative (small variance) and/or if the data are very informative about the hidden states. Typically, this leads to most particles having weights close to zero and just a few particles having significant nonzero weights. In such cases, the resampled particles do not provide a good approximation to the filtered density and results in estimators (of $l(\theta)$ and the hidden states) with large variances.

Refinements of the particle filter that attempt to avoid particle degeneracy are available. Perhaps the most general and commonly used refinement is the auxiliary particle filter (Pitt and Shephard 1999), which is simple to implement and can significantly improve performance for many problems (but may, in fact, reduce it for others). Other advanced particle filters tend to be more restricted to model type than the basic and auxiliary filters.

### *Particle Metropolis-Hastings*

As presented in the previous subsection, the particle filter estimates the likelihood of the state-space model for a given parameter vector $\theta$, but does not provide an estimate of the parameters themselves. Bayesian estimation of $\theta$ can be incorporated in particle filter algorithms by including $\theta$ along with the state vectors as particles for weighted resampling (Liu and West 2001, Thomas et al. 2005), but doing so is often challenging. Here, we instead used MCMC via a Metropolis-Hastings algorithm over the parameters $\theta$ to obtain Bayesian estimates. This approach has been suggested by Fernández-Villaverde and Rubio-Ramírez (2007), by Jones et al. (2010) and by Andrieu et al. (2010), who further provide extensions and a rigorous theoretical justification. Bayesian inference about $\theta$ is based on the posterior distribution of $\theta$ given the data $y_{1:T}$, i.e., on $p(\theta \mid y_{1:T})$, which is defined by the model and priors via Bayes theorem, $p(\theta \mid y_{1:T}) = p(y_{1:T} \mid \theta)\, \pi(\theta)/p(y_{1:T})$. If the likelihood $p(y_{1:T} \mid \theta)$ could be computed exactly, a Metropolis-Hastings algorithm could be used to obtain a sequence of random variables, $\theta_{(1)}$, $\theta_{(2)}$, $\theta_{(3)}$, ... that could be treated as approximate draws from the posterior distribution. This would be done via the following algorithm:

Start with any initial vector $\theta_{(1)}$. For $i = 2{:}M$, there are two steps.

1) Generate a vector of proposed parameter values, $\theta^*_{(i)}$, from some distribution that may depend on the previous value, $g(\theta \mid \theta_{(i-1)})$.

2) Draw a random number $u$ from the uniform distribution on the interval (0, 1). If

$$u < \frac{p(y_{1:T} \mid \theta^*_{(i)})\pi(\theta^*_{(i)})g(\theta_{(i-1)} \mid \theta^*_{(i)})}{p(y_{1:T} \mid \theta_{(i-1)})\pi(\theta_{(i-1)})g(\theta^*_{(i)} \mid \theta_{(i-1)})} \quad (3)$$

then accept the proposed value by setting $\theta_{(i)} = \theta^*_{(i)}$. Otherwise reject the proposed value and retain the previous value, i.e., set $\theta_{(i)} = \theta_{(i-1)}$.

Under the circumstances considered here we cannot compute $p(y_{1:T} \mid \theta)$ exactly, but Andrieu et al. (2010) showed that the algorithm still provides valid inferences if the likelihoods $p(y_{1:T} \mid \theta)$ are replaced by their particle filter estimates, $\hat{p}(y_{1:T} \mid \theta)$, in Eq. 3. Hence, we can use MCMC over a parameter space of much smaller dimension than would have been the case had we also had to sample over the hidden states. The choice of proposal distributions, $g(\theta \mid \theta_{(i-1)})$, will be discussed in the next subsection.

This algorithm describes how to obtain inference about the lower level parameters $\theta$, but does not detail how to proceed with the hidden states, $x$. If inference about the hidden states is not of interest, the particles used in each evaluation of the filter may simply be dropped. However, the particles at each iteration of the MCMC may be used to construct posterior draws from the hidden states in order to compute estimates of quantities such as posterior mean trajectories. The details of how to do this are described in Andrieu et al. (2010).

Reports

### *Independence proposals constructed from an initial adaptive Metropolis-Hastings phase*

Each update of the particle Metropolis-Hastings algorithm involves a run of the particle filter and is therefore computationally costly. Because of this, there is much to gain in spending as few iterations as possible on burn-in and mixing. To address this, we used an adaptive scheme (see e.g., Andrieu and Thoms 2008 and Roberts and Rosenthal 2009 for introductions to adaptive MCMC), which is a simplified version of the one given in Pitt et al. (2010). The scheme is based on three steps. The first uses a random walk Metropolis-Hastings (Robert and Casella 2004) in which the variances of the steps are updated at regular intervals. This is used for initial exploration of the posterior distribution. The second uses an independence Metropolis-Hastings algorithm (Robert and Casella 2004) in which the proposal, $g(\theta)$, is a mixture of normal distributions with the components updated based on the history of the MCMC. This is used for more refined tuning of independence samplers prior to the final step. Adapting the proposals according to the history of the chain breaks the Markov property and may cause flawed inference in the sense that sample means may not converge to posterior expectations. It has been proved that the scheme we use, in fact, still provides output with desired properties (Pitt et al. 2010). Despite this, we prefer to stop the adaptation after a burn-in period and only use the output generated after that period for inferences; this is the third step of our MCMC scheme. Stopping the adaptation may result in slightly less efficient mixing and the need to discard all output generated during the burn-in period, but circumvents the need for a technical investigation of the algorithm since convergence is guaranteed by standard MCMC theory. Details about the proposals and adaptations are given in the Appendix.

### *Marginal densities*

A convenient by-product of the independence particle Metropolis-Hastings sampler is that it allows a simple means of estimating the marginal density of the data. Typically, such estimates are difficult to obtain from MCMC output and may need to be computed separately (Han and Carlin 2001). The marginal density can be estimated by importance sampling using

$$\begin{aligned} p(y_{1:T}) &= \int p(y_{1:T} \mid \theta)\pi(\theta)\, d\theta = \int \frac{p(y_{1:T} \mid \theta)\pi(\theta)}{g(\theta)} g(\theta)\, d\theta \\ &= E_g\left(\frac{p(y_{1:T} \mid \theta)\pi(\theta)}{g(\theta)}\right) \approx \frac{1}{M}\sum_i \frac{p(y_{1:T} \mid \theta^*_{(i)})\pi(\theta^*_{(i)})}{g(\theta^*_{(i)})} \\ &\approx \frac{1}{M}\sum_i \frac{\hat{p}(y_{1:T} \mid \theta^*_{(i)})\pi(\theta^*_{(i)})}{g(\theta^*_{(i)})}. \end{aligned} \tag{4}$$

Note that the estimate is based not on the MCMC output, but on every value of $\theta$ drawn from $g$, and that the prior needs to be properly normalized. In other words, since all of the terms on the right-hand side are computed within the independence particle Metropolis-Hastings algorithm, the marginal density estimate can be obtained with virtually no additional computational burden.

## Example: Diffusion Models of Kangaroo Population Dynamics

### *Stochastic differential equation models of population dynamics*

As an example of how particle Metropolis-Hastings methods can be used to fit population models, we analyzed the dynamics of a red kangaroo (*Macropus rufus*) population in New South Wales, Australia. Data consist of double total transect counts performed on consecutive days at 41 occasions spaced at irregular time intervals ranging from two to six months from 1973 to 1984. The data are given in Caughley et al. (1987) and have been previously analyzed by Knape et al. (2011) using a Gompertz state-space model ignoring the irregular sampling. The irregularly sampled data and the general nonseasonal nature of red kangaroo demography (Caughley et al. 1987) might suggest that continuous time stochastic population models would be more appropriate. For our purposes, this provides a useful illustration of the particle filter MCMC method because we can simulate from such models, but it can be difficult to obtain good MCMCs for them.

We fit three, nested, population models to the kangaroo data. The most general model accounts for potential effects of density dependence and is a logistic diffusion model with environmental variance (Dennis and Costantino 1988). The state model may thus be defined by the stochastic differential equation (interpreted in the Itô sense):

$$dN_t/N_t = (r + \sigma^2/2 - bN_t)dt + \sigma dW_t \tag{5}$$

where $N_t$ is the population size at time $t$, $r$ and $b > 0$ are regressors describing the effects of population size on the growth rate, $\sigma$ is a scaling constant for the growth rate variance, and $W_t$ is white noise Brownian motion. The stationary distribution of $N_t$ under this model exists and is a gamma distribution with mean $r/b$ if $r > 0$, while the population size will tend to zero if $r < 0$ (Dennis and Costantino 1988).

In many cases there is not sufficient information in time series data to estimate several process parameters of a population model with precision. We therefore also considered two reduced versions of Eq. 5 without density feedback, one for which $b = 0$ and one for which $r = b = 0$. For $b = 0$, Eq. 5 can be solved analytically and gives

$$N_t = N_1 \exp[r(t-1) + \sigma \mathrm{RW}_t] \tag{6}$$

where $\mathrm{RW}_t$ is a Gaussian random walk starting at 0 at

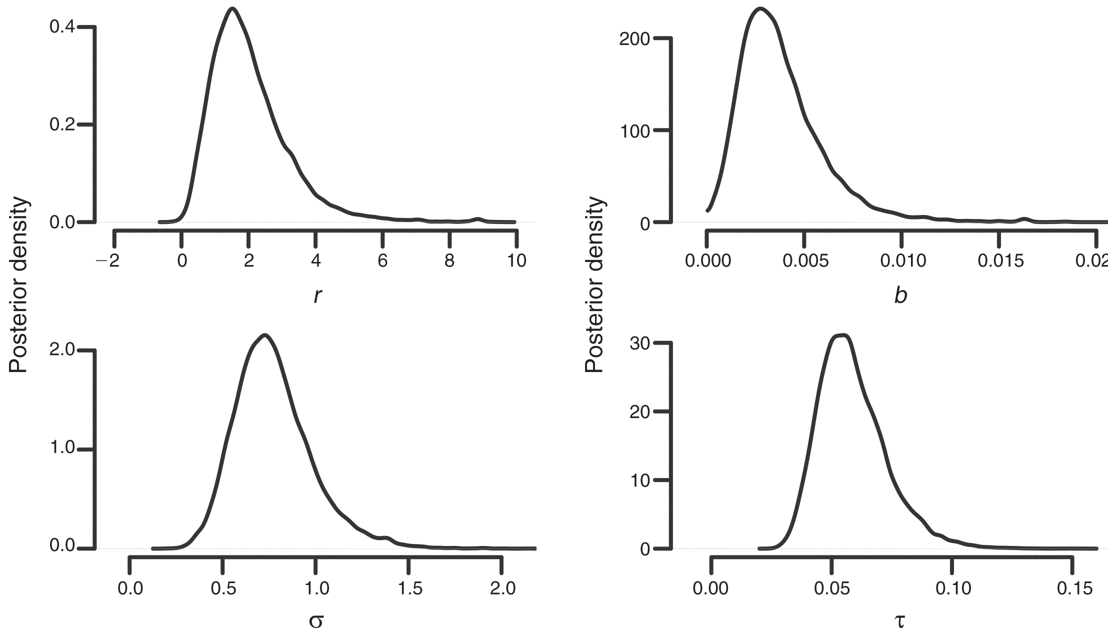


FIG. 1. Kernel density estimates of the posterior distributions for the parameters of the logistic diffusion model of the kangaroo data. Definitions: $r$, small population growth rate; $b$, negative density feedback; $\sigma$ process variance parameter; $\tau$, observation error overdispersion parameter.

time 1 and with variance $t - 1$ at time $t$. Eq. 6 describes unregulated stochastic exponential growth, a model commonly used for population viability analyses (Dennis et al. 1991, Holmes 2004). When $r$ is also equal to zero, this is a continuous time random walk (Brownian motion) on the log scale.

For all three population models, we assumed that the double counts at sampling occasion $t$, $y_{t1}$, and $y_{t2}$, correspond to noisy discrete observations of this model such that $y_{t1} \mid N_t$ and $y_{t2} \mid N_t$ are independent and negatively binomially distributed. The negative binomial distribution was parameterized so that

$$E(y_{ti} \mid N_t) = N_t \qquad V(y_{ti} \mid N_t) = N_t + \tau N_t^2$$

(Lindén and Mäntyniemi 2011) with $\tau$ a parameter to be estimated ($\tau = 0$ corresponds to a Poisson model).

To implement the particle filter for these models, computations of the observation densities $f(y_{t1}, y_{t2} \mid N_t)$ to obtain the particle weights (Eq. 1) and simulations from the population model $f(N_{t+1} \mid N_t)$ are needed. Computations of the weights are straightforward by conditional independence of the observations. Simulation of the population model is trivial in the exponential and random-walk cases using Eq. 6, but is more complicated for the logistic model (Eq. 5). In the latter case, the transition densities $f(N_{t+1} \mid N_t)$ are not available in closed form, but can still be simulated from, for instance, using an approximate Euler discretization or exactly using rejection sampling when $r + \sigma^2/2$ is positive (Beskos et al. 2006). We mainly used the rejection sampling algorithm, but replaced it with an Euler approximation with 100 time steps per year for negative $r + \sigma^2/2$ and for certain parameter combinations where the algorithm was slow. R code for fitting the random-walk model with drift is given in the Supplement.

We put independent uniform priors over the interval (0, 10) on the parameters $b$, $\sigma$, and $\tau$ and a uniform prior over the interval (−10, 10) on $r$. The initial adaptive random walk Metropolis-Hastings algorithm was run for 4000 iterations for each model and with the proposal variance updated at each iteration. We then ran the adaptive independent Metropolis-Hastings phase for 16 000 iterations with proposal updates at every thousand iterations. After the adaptive phase, that was later discarded as burn-in, the proposal distribution at iteration 20 000 was fixed and used for 30 000 iterations. Plots of MCMC chains indicate no issues with convergence or mixing (Appendix: Fig. A1). Posterior densities of the parameters for the full model are given in Fig. 1.

To check the sensitivity to priors we also fitted the model under uniform priors on the parameters over the interval (0, 100). The alternative prior revealed that there was a heavy right tail to the posterior distributions of $r$, $b$, and $\sigma$, corresponding to posterior draws with simultaneously large values for each of these parameters. This would represent a population that is highly reactive and unpredictable even at short timescales, which does not seem biologically plausible.

*Model selection*

Denoting the logistic model (Eq. 5) as $M_1$, the exponential growth model (Eq. 6) as $M_2$, and the random-walk model as $M_3$ and giving them uniform prior probabilities, $\pi(M_1) = \pi(M_2) = \pi(M_3) = 1/3$,

TABLE 1. Bayes factors for pairwise comparisons (given as preference for rows over columns) between the geometric Brownian motion without drift (Model M2), geometric Brownian motion with drift (Model M3), and logistic diffusion models of the kangaroo data and posterior model probabilities under a uniform prior on the models.

| Models | M1 (logistic) | M2 (exponential) | M3 (random walk) | Model probabilities | log(marginal density) |
|---|---|---|---|---|---|
| M1 (logistic) | | 1/95 | 1/4968 | 0.00 | −556.2 |
| M2 (exponential) | 95 | | 1/52 | 0.02 | −551.6 |
| M3 (random walk) | 4968 | 52 | | 0.98 | −547.7 |

posterior model probabilities may be estimated as

$$p(M_i \mid y_{1:T}) = \frac{p(y_{1:T} \mid M_i)\pi(M_i)}{\sum_{j=1}^{3} p(y_{1:T} \mid M_j)\pi(M_j)}$$

by plugging in estimates of $p(y_{1:T} \mid M_i)$ computed by importance sampling from Eq. 4. Pairwise comparisons using Bayes factors are further given by

$$\mathrm{BF}_{ij} = \frac{p(y_{1:T} \mid M_i)}{p(y_{1:T} \mid M_j)}$$

with the interpretation that the probability of the data is $\mathrm{BF}_{ij}$ times higher under model $M_i$ than under model $M_j$. Among the three models under consideration, the random-walk model ($M_3$) was highly preferred over the logistic ($M_1$) and exponential ($M_2$) models (Table 1, Fig. 2). Thus, neither the exponential increase, nor the negative density feedback adds sufficient model fit. We note that, as pointed out by others (e.g., Link and Barker 2006), posterior model probabilities and Bayes factors can be sensitive to the priors on the parameters. This was the case for the logistic model $M_1$. Under the alternative uniform priors over the interval (–100, 100) for $r$ and (0, 100) for the other parameters the marginal density was a factor $10^3$ times smaller than under the original prior.

## DISCUSSION

The particle Metropolis-Hastings algorithm illustrated here provides a brute-force method for fitting complex state-space population models. The method is, arguably, relatively simple to implement since it utilizes only basic analytical information about the model under consideration. This also makes modification of the algorithm to fit alternative models easy. Once an implementation of the algorithm has been developed for one model, it is, in many cases, sufficient to change two or three lines of code to fit an alternative model. The method further provides a way of fitting models for which the transition densities, $f(x_t \mid x_{t-1})$, are not easy to write analytically, but can be simulated from. Other benefits are that the marginal density of the data needed to compute Bayes factors and posterior model probabilities for Bayesian model selection can be computed. This stems from the fact that particle filters provide estimates of properly normalized likelihoods.

The main drawback of PFMCMC is that it is computationally demanding. Running the particle Metropolis-Hastings algorithm in R for 50 000 iterations took ~20 hours for the logistic model (with simulations from the transition densities implemented in C) and two hours for the exponential growth and random-walk models on a three GHz personal computer. Such running times may seem prohibitive, but will be reduced as computers become faster. Running times may also be decreased by parallelizing the algorithm using, e.g., clusters, multi-core processors, or graphics cards (Lee et al. 2011). We have had initial success with implementing a simple parallelized PFMCMC in R.

Apart from its computational burden, the limitations of the particle Metropolis-Hastings algorithm are similar to the limitations of particle filters in exploring the hidden states of state-space models for given parameter values. For instance, particle filters tend to not perform well in situations where the observations are high dimensional or very informative about the hidden states, leading to particle degeneracy. While

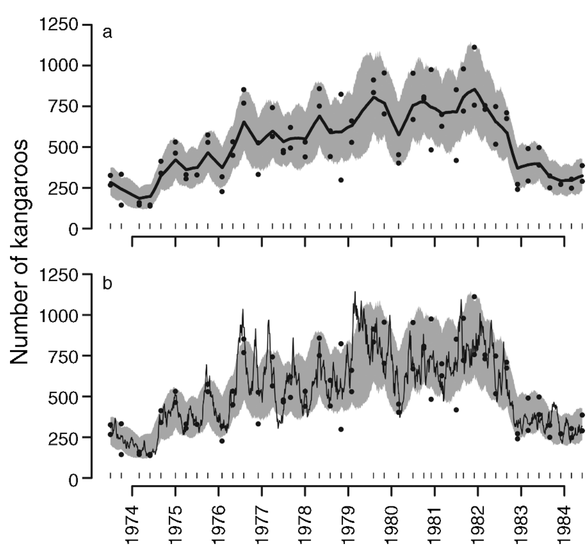


FIG. 2. (a) The posterior mean trajectory and (b) a random posterior trajectory of the kangaroo population under the best-fitting model (random walk). Points represent observed numbers, and the short black vertical lines above the $x$-axis indicate the (irregular) times of sampling. The shaded areas comprise 95% highest posterior density intervals for the population.

Reports

Andrieu et al. (2010) showed that degeneracy of the particle filter does not invalidate the PFMCMC, extreme degeneracy may lead to poor mixing of the underlying MCMC algorithm. Hence, individual based mark–recapture models that can be formulated as state-space models (e.g., Gimenez et al. 2007) may be out of reach for this method. On the other hand, since parameter values are fixed during particle filter runs, degeneracy may be less prominent compared to methods where parameter estimation is incorporated as part of the particle filter (Liu and West 2001, Thomas et al. 2005).

Particle filter Metropolis-Hastings is a simple method for fitting nonlinear population models in situations where standard MCMC software performs poorly or is difficult to program or when Bayesian model selection is desired. Nonlinear population dynamical models and observation errors in data are common in ecology. Particle MCMC methods therefore have good potential for finding a wide range of applications.

## Literature Cited


Andersen, K. H., A. Nielsen, U. H. Thygesen, H. Hinrichsen, and S. Neuenfeldt. 2007. Using the particle filter to geolocate Atlantic cod (*Gadus morhua*) in the Baltic Sea, with special emphasis on determining uncertainty. Canadian Journal of Fisheries and Aquatic Sciences 64:618–627.

Andrieu, C., A. Doucet, and R. Holenstein. 2010. Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society B 72:269–342.

Andrieu, C., and J. Thoms. 2008. A tutorial on adaptive MCMC. Statistics and Computing 18:343–373.

Beskos, A., O. Papaspiliopoulos, and G. O. Roberts. 2006. Retrospective exact simulation of diffusion sample paths with applications. Bernoulli 12:1077–1098.

Buckland, S., K. Newman, L. Thomas, and N. Koesters. 2004. State-space models for the dynamics of wild animal populations. Ecological Modelling 171:157–175.

Calder, C., M. Lavine, P. Muller, and J. Clark. 2003. Incorporating multiple sources of stochasticity into dynamic population models. Ecology 84:1395–1402.

Cappé, O., S. Godsill, and E. Moulines. 2007. An overview of existing methods and recent advances in sequential Monte Carlo. Proceedings of the IEEE 95:899–924.

Caughley, G., N. Shepherd, and J. Short, editors. 1987. Kangaroos, their ecology and management in the sheep rangelands of Australia. Cambridge University Press, Cambridge, UK.

Dennis, B., and R. Costantino. 1988. Analysis of steady-state populations with the gamma-abundance model: application to Tribolium. Ecology 69:1200–1213.

Dennis, B., P. Munholland, and J. Scott. 1991. Estimation of growth and extinction parameters for endangered species. Ecological Monographs 61:115–143.

Dennis, B., J. Ponciano, S. Lele, M. Taper, and D. Staples. 2006. Estimating density dependence, process noise, and observation error. Ecological Monographs 76:323–341.

de Valpine, P. *In press*. Frequentist analysis of hierarchical models for population dynamics and demographic data. Journal of Ornithology. doi 10.1007/s10336-010-0642-5

de Valpine, P., and A. Hastings. 2002. Fitting population models incorporating process noise and observation error. Ecological Monographs 72:57–76.

Doucet, A., N. De Freitas, and N. Gordon. 2001. Sequential Monte Carlo methods in practice. Springer-Verlag, New York, New York, USA.

Dowd, M. 2006. A sequential Monte Carlo approach for marine ecological prediction. Environmetrics 17:435–455.

Dowd, M., and R. Joy. 2011. Estimating behavioral parameters in animal movement models using a state augmented particle filter. Ecology 92:568–575.

Fernández-Villaverde, J., and J. F. Rubio-Ramírez. 2007. Estimating macroeconomic models: a likelihood approach. Review of Economic Studies 74:1059–1087.

Fujiwara, M., B. Kendall, R. Nisbet, and W. Bennett. 2005. Analysis of size trajectory data using an energetic-based growth model. Ecology 86:1441–1451.

Gimenez, O., V. Rossi, R. Choquet, C. Dehais, B. Doris, H. Varella, J.-P. Vila, and R. Pradel. 2007. State-space modelling of data on marked individuals. Ecological Modelling 206:431–438.

Gordon, N. J., D. J. Salmond, and A. F. M. Smith, editors. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings of Radar and Signal Processing 140:107–113.

Han, C., and B. P. Carlin. 2001. Markov chain Monte Carlo methods for computing Bayes factors: a comparative review. Journal of the American Statistical Association 96:1122–1132.

Harrison, P., I. Hanski, and O. Ovaskainen. 2011. Bayesian state-space modeling of metapopulation dynamics in the Glanville fritillary butterfly. Ecological Monographs 81:581–598.

Holmes, E. E. 2004. Beyond theory to application and evaluation: diffusion approximations for population viability analysis. Ecological Applications 14:1272–1293.

Ionides, E., C. Breto, and A. King. 2006. Inference for nonlinear dynamical systems. Proceedings of the National Academy of Sciences USA 103:18438–18443.

Jones, E., J. Parslow, and L. Murray. 2010. A Bayesian approach to state and parameter estimation in a phytoplankton-zooplankton model. Australian Meteorological and Oceanographic Journal 59:7–16.

King, R., B. Morgan, O. Gimenez, and S. Brooks. 2009. Bayesian analysis for population ecology. CRC Press, Boca Raton, Florida, USA.

Knape, J., N. Jonzén, and M. Sköld. 2011. On observation distributions for state space models of population survey data. Journal of Animal Ecology 80:1269–1277.

Lee, A., C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes. 2011. On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. Journal of Computational and Graphical Statistics 19:769–789.

Lindén, A., and S. Mäntyniemi. 2011. Using negative binomial distribution to model overdispersion in ecological count data. Ecology 92:1414–1421.

Link, W. A., and R. J. Barker. 2006. Model weights and the foundations of multimodel inference. Ecology 87:2626–2635.

Liu, J., and M. West. 2001. Combining parameter and state estimation in simulation based filtering. Pages 197–223 *in* A. Doucet, N. de Freitas, and N. Gordon, editors. Sequential Monte Carlo methods in practice. Springer-Verlag, New York, New York, USA.

Lunn, D. J., A. Thomas, N. Best, and D. Spiegelhalter. 2000. WinBUGS. A Bayesian modelling framework: concepts, structure, and extensibility. Statistics and Computing 10:325–337.

Newman, K. B. 1998. State-space modeling of animal movement and mortality with application to Salmon. Biometrics 54:1290–1314.

Newman, K. B., S. T. Buckland, S. T. Lindley, L. Thomas, and C. Fernández. 2006. Hidden process models for animal population dynamics. Ecological Applications 16:74–86.

Newman, K. B., C. Fernández, L. Thomas, and S. T. Buckland. 2009. Monte Carlo inference for state-space models of wild animal populations. Biometrics 65:572–583.

Peters, G. W., G. R. Hosack, and K. R. Hayes. 2010. Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo (AdPMCMC). Preprint arXiv:1005.2238v1. Cornell University Library, Ithaca, New York, USA.

Pitt, M. K., and N. Shephard. 1999. Filtering via simulation: auxiliary particle filters. Journal of the American Statistical Association 94:590–599.

Pitt, M., R. Silva, P. Giordani, and R. Kohn. 2010. Auxiliary particle filtering within adaptive Metropolis Hastings sampling. Preprint arXiv:1006.1914v1. Cornell University Library, Ithaca, New York, USA.

Robert, C. P., and G. Casella. 2004. Monte Carlo statistical methods. Springer, New York, New York, USA.

Roberts, G. O., and J. S. Rosenthal. 2009. Examples of adaptive MCMC. Journal of Computational and Graphical Statistics 18:349–367.

Thomas, A., B. O'Hara, U. Ligges, and S. Sturtz. 2006. Making BUGS open. R News 6:12–17.

Thomas, L., S. T. Buckland, K. B. Newman, and J. Harwood. 2005. A unified framework for modelling wildlife population dynamics. Australian and New Zealand Journal of Statistics 47:19–34.

Trenkel, V. M., D. A. Elston, and S. Buckland. 2000. Fitting population dynamics models to count and cull data using sequential importance sampling. Journal of the American Statistical Association 95:363–374.

## Supplemental Material

### Appendix

An adaptive Metropolis-Hastings scheme (*Ecological Archives* E093-025-A1).

### Supplement

R code for fitting the random-walk state-space model using particle filter MCMC (*Ecological Archives* E093-025-S1).

Reports