Epidemics xxx (xxxx) xxxx

Contents lists available at ScienceDirect

Epidemics

journal homepage: www.elsevier.com/locate/epidemics

Introduction to particle Markov-chain Monte Carlo for disease dynamics modellers

Akira Endo[a,*], Edwin van Leeuwen[b], Marc Baguelin[a,b,c]

[a] London School of Hygiene & Tropical Medicine, London, United Kingdom
[b] Public Health England, London, United Kingdom
[c] Imperial College, London, United Kingdom

ARTICLE INFO

Keywords:
Particle Markov-chain Monte Carlo
State-space models
Hidden Markov process
Particle filter
Sequential Monte Carlo

ABSTRACT

The particle Markov-chain Monte Carlo (PMCMC) method is a powerful tool to efficiently explore high-dimensional parameter space using time-series data. We illustrate an overall picture of PMCMC with minimal but sufficient theoretical background to support the readers in the field of biomedical/health science to apply PMCMC to their studies. Some working examples of PMCMC applied to infectious disease dynamic models are presented with R code.

1. Introduction

In many fields of the applied sciences, inference from time-series data is an important class of problems. A standard modelling approach is to develop a mechanistic model (deterministic or stochastic) that captures the temporal dynamics of the target phenomenon, and then optimise the model parameters so that the theory and observation are consistent. If the likelihood of the observed data is available in a functional form, parameter estimation is relatively easy; Markov-chain Monte Carlo (MCMC) is a widely-used approach that can efficiently sample from high-dimensional parameter space. On the other hand, when one wishes to model the observation process (e.g., measurement error) in addition to the dynamic process (i.e. time evolution of the system, which may be subject to process error), oftentimes the model needs to be separated into two parts: the dynamic model that generates the true outcome and the observation model that generates the observed data. The true outcome is not directly observable (hidden variables) and needs to be inferred. There is a method called data augmentation that enables MCMC to handle such hidden variables by simultaneously estimating both the parameters and the hidden variables. However, hidden variables in a time-series analysis with T time steps have T different values which are correlated with each other, and such high correlation between variables tend to substantially slow down the mixing of MCMC.

Particle Markov-chain Monte Carlo (PMCMC) has been proposed to overcome this weakness of MCMC in time-series analyses (Andrieu and Doucet, 2010). To efficiently explore time-varying hidden variables, PMCMC incorporates Sequential Monte Carlo (SMC; also known as particle filtering) (Doucet et al., 2001) into MCMC. SMC is an inference method designed to efficiently estimate time-dependent hidden variables. By combining the strengths of MCMC and SMC, PMCMC serves as a powerful estimation framework to explore high-dimensional parameter space in dynamic models.

Despite potentially high demands for efficient inference tools in time-series analysis, PMCMC has not yet attracted enough attention from practitioners in biomedical/health science fields including biology, ecology, epidemiology and public health. This may be because many of the previous methodological literature on PMCMC required the researcher to have a strong mathematical and statistical background as well as a basic understanding of both MCMC and SMC, which might not always be expected in aforementioned fields (especially in early-career). To fill this gap, in this paper we attempt to provide the readers with an overall picture of the PMCMC framework while including the minimal but sufficient theoretical background of the method so that the reader can be confident when applying PMCMC to their own research questions. Assuming that the readers have a basic understanding of Bayesian inference and MCMC (see, for example, (Funk et al., 2018) in this special issue for an overview), we first introduce in Section 2 the conceptual framework of PMCMC without too much technical detail. We do not include in the main text the technical and theoretical details of SMC, which is the key component of PMCMC, and rather treat it as an external algorithm to outsource the computation of the (marginal) likelihood to. This is to ensure that readers with an understanding of SMC or those who are more interested in the actual application of

* Corresponding author.
E-mail address: akiraendo@outlook.com (A. Endo).

https://doi.org/10.1016/j.epidem.2019.100363
Received 22 February 2019; Received in revised form 26 July 2019; Accepted 19 August 2019

PMCMC can more easily get an overview of the PMCMC method. Interested readers may refer to Appendix B where the technical aspects of the SMC algorithm are described. In Section 3, we present multiple examples of typical inference problems and how PMCMC can be implemented in R to analyse the time-series data.

2. Basic framework of PMCMC

2.1. Overview of PMCMC and suitable inference problems

PMCMC is especially suited for inference of time-series data. The specific class of inference problems targeted by PMCMC is called Hidden Markov Process (HMP; also known as the state-space model) (Mac Donald and Zucchini, 1997). We will describe the properties of HMP in the next section in detail, but in short, HMP is a process where the system has unobservable (hidden) "state variables" which change over time, and the next state depends only on the current state and not on any previous states (Markov-process). Although the state variables are not directly measured, observation data is available which may contain (potentially limited) information on the state variables. Familiar examples of HMP in infectious disease modelling include the commonly-used Susceptible-Infectious-Recovered model (SIR model) (Hethcote, 2000)[1] and its stochastic extensions (Allen, 2008; Cooper and Lipsitch, 2004). The SIR model is based on a set of ordinary differential equations (i.e., its behaviour is solely determined by the current state) and thus constitutes a Markov process. Researchers try to understand the transmission dynamics by applying mathematical models to the observed data; however, when data is generated with measurement errors (e.g., underreporting or diagnostic errors), the true state of the system is not directly reflected in the data at hand. Nevertheless, such data should contain relevant information and we may still be able to infer the true transmission dynamics, given a plausible understanding of the measurement process translating the hidden states into the observed data. PMCMC is a method to efficiently achieve this aim, by inferring the hidden variables alongside the model parameters from indirect observations.

Typically, PMCMC is better suited than other algorithms when the time evolution involves stochasticity and the data is reported with errors (including underreporting and modification). If the time evolution is deterministic or the data is reported without errors, the direct likelihood of the data is usually available and a conventional MCMC approach will suffice. In general, the following conditions result in typical inference problems that may benefit from the use of PMCMC:

1. The user wants to apply a parametric model to time-series data to infer a dynamical system. The model has parameters that are static over the time course as well as time-dependent variables[2] .
2. The data is not a direct observation of the model outcomes and specific observation processes are involved, e.g., measurement error, bias, missing data or partial/modified observation.
3. For a given set of parameters, the time evolution of the system and the observation processes are both modelled as probabilistic distributions and are available in functional forms (at least numerically).
4. The time evolution of the system is a Markov process; i.e., the time evolution is fully predicted by the current state without the previous history.
5. Interdependency between parameters and state variables prevents efficient inference using other methods including MCMC.

In particular, Conditions 2 and 5 are the most crucial points that highlight the need for PMCMC over other methods including MCMC. The existence of hidden state variables, one of the key properties of HMP, often brings a computational difficulty in inference problems. In short, the difficulty arises from the need to "integrate out" the hidden variables; here we illustrate this challenge. Let us consider an HMP where x_t is the hidden state variable and y_t is the observed data at time t. In infectious disease modelling, x_t may be the true number of infectious individuals and y_t the reported case counts, which are potentially multidimensional when the data is stratified by, e.g., age or location. The time evolution of x_t is governed by parameter θ (e.g., transmission rate), and we are interested in the posterior distribution of this parameter: $p(\theta|y_t)$ and in many situation not in the underlying x_t's. Bayes theorem states that $p(\theta|y_t) \propto p(y_t|\theta)p(\theta)$, and we need to compute the likelihood of the data $p(y_t|\theta)$ to perform Bayesian inference. This likelihood is often referred to as the marginal likelihood, as it is a marginal distribution of $p(x_t, y_t|\theta)$ on y_t. For simplicity, let us first consider that there are only two time steps ($t = 1, 2$) and that the initial state x_1 is known (thus all the observation about the process is given by y_2 only). In this two-step case, we need to compute $p(y_2|\theta)$ to infer θ, and this marginal likelihood is given as

$$p(y_2|\theta) = \int p(y_2|x_2, \theta)p(x_2|x_1, \theta)\mathrm{d}x_2, \tag{1}$$

where $p(y_2|x_2, \theta)$ and $p(x_2|x_1, \theta)$ are probabilistic distributions for the observation and time-evolution processes, respectively. Here, the hidden variable x_2 needs to be integrated out to calculate the marginal likelihood giving the posterior of the parameters given y_2. If this integration (over x_2) is readily available as a function of θ in an exact form, Bayesian inference is easily implemented by MCMC or other methods. Such cases are, however, very rare in actual applications, and generally the integration on the right-hand side of Equation (1) needs to be numerically approximated by Monte Carlo sampling (which does not necessarily use a Markov chain), where many samples of x_2 are generated and the integral is computed from the values of $p(y_2|x_2, \theta)p(x_2|x_1, \theta)$ over the samples. The easiest idea might be to sample x_2 from a uniform distribution within the possible range of x_2 and perform the Monte Carlo integration (Press et al., 2007). However, this approach is extremely inefficient as x_2 becomes high-dimensional. To achieve a practical level of efficiency, x_2 must be selectively sampled from plausible regions, i.e. regions with high probability.

The same applies to general cases with more time steps ($t = 1, 2, .. ., T$), but the computational challenge further scales up. The state variable x is now a time series of T steps, and the whole time series $x_{1:T} = \{x_1, x_2, .. ., x_T\}$ needs to be sampled from plausible regions. Compared with the earlier example with only two time steps, finding plausible regions of a long time series of x is increasingly difficult. It becomes unrealistic to independently sample all the hidden states because the distribution of x_t is restricted by the previous state x_{t-1}. If the time series is short enough, the plausible regions of x may be adaptively explored by MCMC targeting the joint distribution $p(x, \theta|y) \propto p(y|x, \theta)p(x|\theta)p(\theta)$ (this approach is known as data-augmentation MCMC Gelfand and Smith, 1990; Neal and Kypraios, 2015); but for longer time series, a small change in each time step accumulates over time, resulting in a massive diversity in the plausible trajectories of x even with the same parameter (see Fig. 1). MCMC is not well suited for sampling from such sparse and highly-correlated parameter spaces and typically suffers from unacceptably slow mixing when applied to HMP inference problems.

Let us use a simple random-walk model as an example to review the above discussion. The example is very simple so that it is easier to follow, but one can easily extend the model by assuming more complex mechanisms. Full documentation for this example, including implementation of SMC and PMCMC, can be found on a Github repository

[1] The SIR model is usually formulated on a continuous time scale, but its numerical computation essentially requires temporal discretisation and can be seen as a discrete-time Markov process.

[2] If instead the model is free of static parameters and only has time-dependent variables to estimate, PMCMC does not have supremacy over Sequential Monte Carlo (SMC).

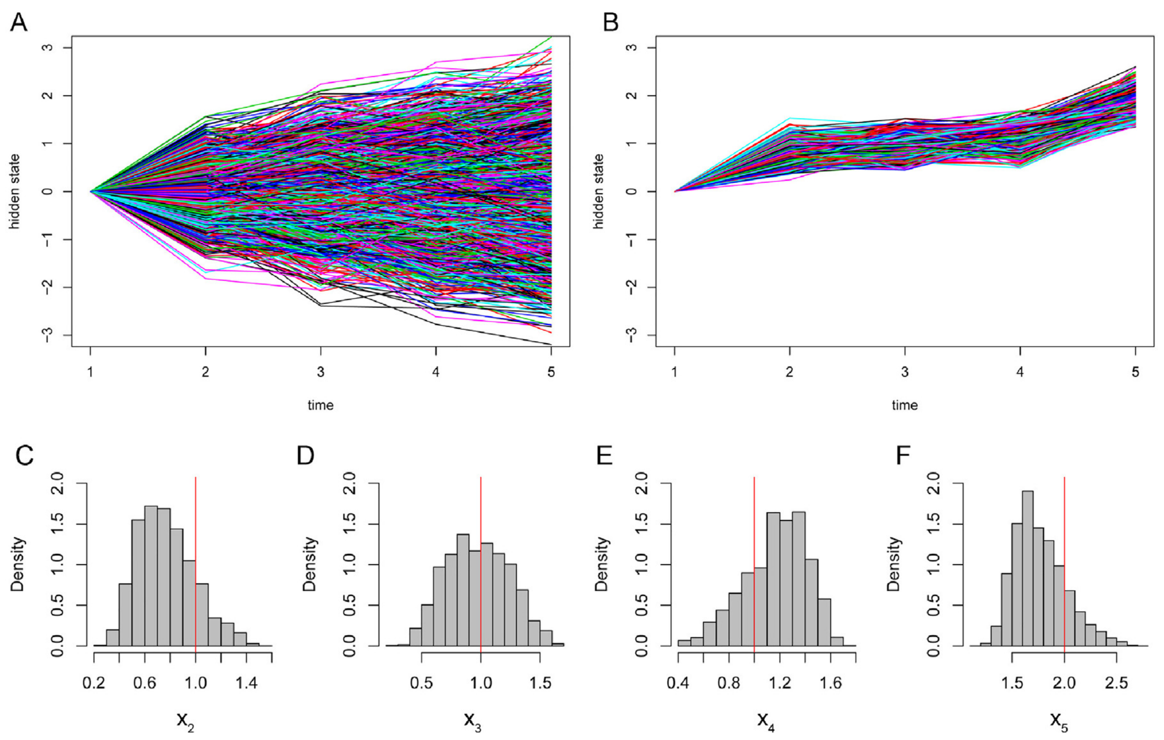

Fig. 1. Example of a hidden Markov process. A random-walk hidden Markov process model where the state variable x_t evolves following a Gaussian kernel and is observed with a Gaussian measurement error and rounding. (A) A large sample of simulated trajectories among all the possible trajectories of $x_{1:5}$. (B) Simulated trajectories consistent with data $y_{1:5} = (0, 1, 1, 1, 2)$. (C)–(D) Histograms of $x_{1:5}$ consistent with $y_{1:5}$. The red lines represent the observed data.

(https://github.com/akira-endo/Intro-PMCMC). In this model, we assume that x_t is normally distributed around the previous state, i.e., $x_t \sim \mathcal{N}(x_{t-1}, \sigma)$, with a non-random initial state $x_1 = 0$. The data is observed as a rounded integer with a Gaussian measurement error $y_t = \mathrm{round}(\mathcal{N}(x_{t-1}, 0.1))$. First, consider a two-time-step case ($T = 2$) where the observation was $(y_1, y_2) = (0, 1)$. Although a substantial amount of information is lost because of the crude observation process, we can still extract certain clues from the data. Noting that y_1 does not provide additional information as x_1 is known, Equation (1) translates into

$$p(y_2 = 1|\sigma) = \int r(1; \mathcal{N}(x_2, 0.1))\mathcal{N}(x_2; 0, \sigma)\mathrm{d}x_2, \tag{2}$$

where $r(1; \mathcal{N}(x_2, 0.1)) = \Pr[0.5 \le y < 1.5; y \sim \mathcal{N}(x_2, 0.1))]$. This example is actually a rare case where the likelihood function is available in a functional form:

$$p(y_2 = 1|\sigma) = \Phi(1.5; 0, \sqrt{\sigma^2 + 0.01}) - \Phi(0.5; 0, \sqrt{\sigma^2 + 0.01}), \tag{3}$$

where $\Phi(x;\mu, \sigma)$ is the cumulative normal distribution, and σ can be estimated with this likelihood. However, this holds only when $T = 2$, and adding a third time step makes the functional-form likelihood unavailable. If Equation (3) were not available, the integration (Eq. (2)) would have to be approximated by Monte Carlo integration, i.e., by drawing samples $\{X_2\}$ uniformly from the range [0, 2][3] and averaging the value $\frac{\mathcal{N}(x_2; 0, \sigma)}{\mathcal{U}(x_2; 0, 2)}$. This might work for such a short time series (with a 1-dimensional variable), but not for a longer time series. Consider $T = 5$, $\sigma = 0.5$ and observed data $y_{1:5} = (0, 1, 1, 1, 2)$. Even with only five time steps, we can observe a wide variety of time series $x_{1:5}$ consistent with the observation (Fig. 1). That is, if we uniformly sample from the possible time series shown in Fig. 1A, the majority of the samples will have very small likelihood values and be of little use. Moreover, because of the high correlation between the successive states, the data-augmentation MCMC will not be efficient.

PMCMC has overcome this challenge in exploring and marginalising out time-evolving hidden variables by employing sequential Monte Carlo (SMC) as a sub-algorithm inside MCMC. As SMC exploits the temporal structure of the data for the inference of time-evolving variables, PMCMC making use of SMC can also utilise its strengths. Section 2.3 will illustrate how SMC is coupled with MCMC to accelerate the sampling of the hidden state variables.

2.2. Hidden Markov process

In this section, we will describe the structure and the properties of Hidden Markov Process (HMP). HMP is also referred to as the state-space model and is characterised as a Markov process where the internal state is not directly observed. A Markov process is a stochastic process whose future time evolution is predicted solely by the current state and is independent of previous states. Let x_t represent the state variable of the system following a Markov process. Conditioned to x_t, the probabilistic distribution of x_{t+1}, the state variable at the next time step, is written as a function of the current state x_t:

$$p(x_{t+1}|x_t) = f(x_{t+1}|x_t) \tag{4}$$

For example, a recursive sequence $a_{t+1} = a_t + 1$ is interpreted as a (non-random) Markov process, while $a_{t+1} = a_t + a_{t-1}$ is not, because the transition from t to $t + 1$ depends also on the past state a_{t-1}. A Markov process is thus characterised by the state variable x_t and the time evolution process $f(x_{t+1}|x_t)$. In infectious disease models where x_t denotes the true incidence, $f(x_{t+1}|x_t)$ may represent the process of transmission.

In HMP, an observation process is added to the Markov process (Fig. 2). The state variable x_t is no longer directly observed, and observation y_t is now given as a random variable depending on x_t. The

[3] because x_2 is unlikely to fall outside of this range given $y_2 = 1$. Technically we introduce a very small bias here as the x_2 could fall outside of this interval due to the measurement error. Given that the standard deviation of the normal component of the observation error is 0.1, this probability may be almost negligible.

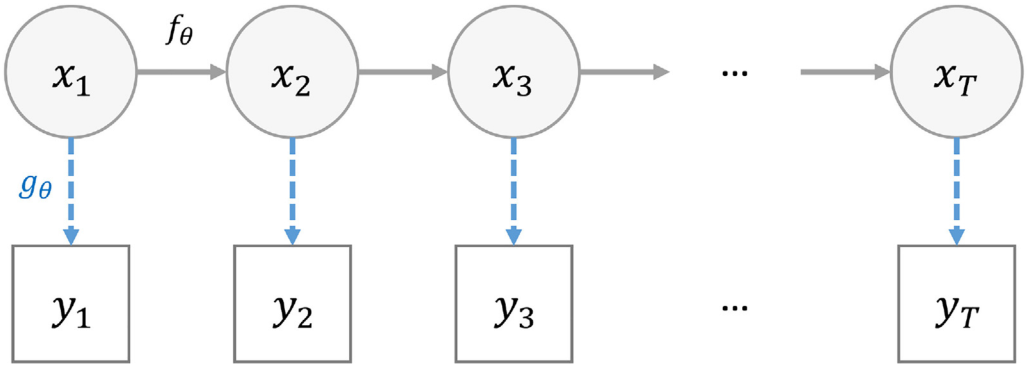

Fig. 2. Schematic image of a hidden Markov process. x denotes hidden state variables and y denotes observations. Time evolution and observation processes are characterised by probabilistic distributions f_θ and g_θ.

probabilistic distribution of observing y_t is a function of x_t:

$$p(y_t|x_t) = g(y_t|x_t). \tag{5}$$

The observation process $g(y_t|x_t)$ may reflect inherent factors involved in observation, e.g., measurement error, bias or partial/modified observation. A typical example of observation process $g(y_t|x_t)$ in infectious disease modelling is underreporting, because not everyone who is sick will visit the GP/hospital and/or will not always be diagnosed correctly. This is often represented by a binomial process with a certain reporting probability ϵ; i.e., $g(y_t|x_t) = \text{Bin}(y_t; x_t, \epsilon)$. Inference on HMP requires extracting indirect clues on the hidden state x_t from y_t, taking in account the possible modification $g(y_t|x_t)$. The main components of HMP are: time-varying hidden state variables x_t, its time evolution process $f(x_{t+1}|x_t)$, time-series observations y_t and the observation process $g(y_t|x_t)$. In inference problems employing the HMP framework the hidden state x_t and the time evolution process $f(x_{t+1}|x_t)$ are usually of interest. The function f is modelled with a set of parameters θ ($f = f_\theta$) and θ is estimated from data as well as x_t. In some cases, the observation process g_θ (also shaped by parameters) may also be investigated. For example, transmissibility β in the SIR model determines the time evolution process $f(x_{t+1}|x_t)$ and is usually of interest in modelling studies. In addition, when underreporting of disease incidence is expected, one might wish to simultaneously estimate the reporting probability which characterises the observation process g_θ.

The importance of the Markov property may not be self-evident. However, it plays a vital role in reducing the complexity of the model structure; the Markov property allows for the use of SMC, which is relatively simple and computationally-inexpensive. Appendix A describes the implications of the Markov property for interested readers.

2.3. *Sequential Monte Carlo as a sub-algorithm to approximate the marginal likelihood*

The biggest challenge in the inference of Hidden Markov Process (HMP) is to efficiently sample hidden state variables. To obtain the marginal likelihood $p(y_{1:T}|\theta)$, the time-dependent hidden variables x_t have to be sampled and marginalised out. Because of the strong dependency between the successive states x_t and x_{t-1}, sampling the whole time series $x_{1:T}$ at once is an inefficient way to explore the large variable space (as shown in Fig. 1). A straightforward remedy for this problem is "step-wise sampling" of x_t using observation y_t at each time step. Although the full time series $x_{1:T}$ may exhibit substantial variability, a one-step transition from x_{t-1} to x_t is usually relatively straightforward. Rather than sampling the whole time series $x_{1:T}$, one can sequentially narrow down the distribution of x_t with observation y_t. By shadowing the time evolution process while calibrating the hidden state step-by-step, plausible realisations of $x_{1:T}$ are efficiently generated. Once a set of plausible trajectories of $x_{1:T}$ is available, the likelihood $p(y_{1:T}|\theta)$ is computed based on the generalised form of Equation (1):

$$p(y_{1:T}|\theta) = \int p(y_{1:T}|x_{1:T}, \theta)p(x_{1:T}|\theta)\mathrm{d}x_{1:T}. \tag{6}$$

Sequential Monte Carlo (SMC; also known as particle filtering) is designed to implement this "step-wise" sampling. PMCMC outsources the whole sampling process of $x_{1:T}$ to SMC given the current parameter θ and data $y_{1:T}$. With supplied θ and $y_{1:T}$, SMC produces a set of samples for the time series (or "trajectories") $\{X_{1:T}\}$ and the corresponding marginal likelihood $p(y_{1:T}|\theta)$ sequentially. Samples X_t at each time step are sometimes called "particles". We outlined the technical details of SMC in Appendix B for interested readers, but to summarise, the SMC algorithm consists of the following steps:

- For $t = 1, 2, .., T$, sample the current state x_t using the samples of the previous state x_{t-1} and the current observation y_t. The target distribution to be sampled from is characterised by the probabilities for the time evolution and observation processes:

$$p(x_t, |x_{t-1}, y_t, \theta) \propto p(y_t|x_t, \theta)p(x_t|x_{t-1}, \theta) = g_\theta(y_t|x_t)f_\theta(x_t|x_{t-1}) \tag{7}$$

- Using the produced samples of time series $x_{1:T}$, approximate the marginal likelihood $\hat{p}(y_{1:T}|\theta)$ based on the following equation:

$$\begin{aligned} p(y_{1:T}|\theta) &= p(y_1|\theta)\prod_{t=2}^{T} p(y_t|y_{1:t-1}, \theta) \\ &= \int p(y_1|x_1, \theta)p(x_1, \theta)\mathrm{d}x_t \prod_{t=2}^{T}\int p(y_t|x_t, \theta)p(x_t|y_{1:t-1}, \theta)\mathrm{d}x_t. \end{aligned} \tag{8}$$

The key component in this algorithm is sampling x_t from the distribution (Eq. (7)). The bootstrap filter, the standard algorithm for SMC (detailed in Appendix B), achieves this by generating particles based on $f_\theta(x_t|x_{t-1})$ and then filter those particles based on $g_\theta(y_t|x_t)$ so that the resulting samples follow Eq. (7). Fig. 3 illustrates how the SMC algorithm works when applied to the random-walk model in Section 2.1. For each time step, the candidate particles are generated from the previous state: $x_t \sim \mathcal{N}(x_{t-1}, \sigma)$, and then only those consistent with the observation are selected and stored. Comparison with Fig. 1 shows that the SMC efficiently produces samples of x_t consistent with the data y_t from a diverse space.

2.4. *PMCMC algorithm*

SMC being used as the sub-algorithm, the algorithm of PMCMC (where standard Metropolis-Hastings is used as the MCMC algorithm) is given below. Algorithm 1 shows pseudocode; application examples in R will be presented in Section 3.

1. Arbitrarily choose an initial parameter value $\theta^{(0)}$ (step $n = 0$). Run

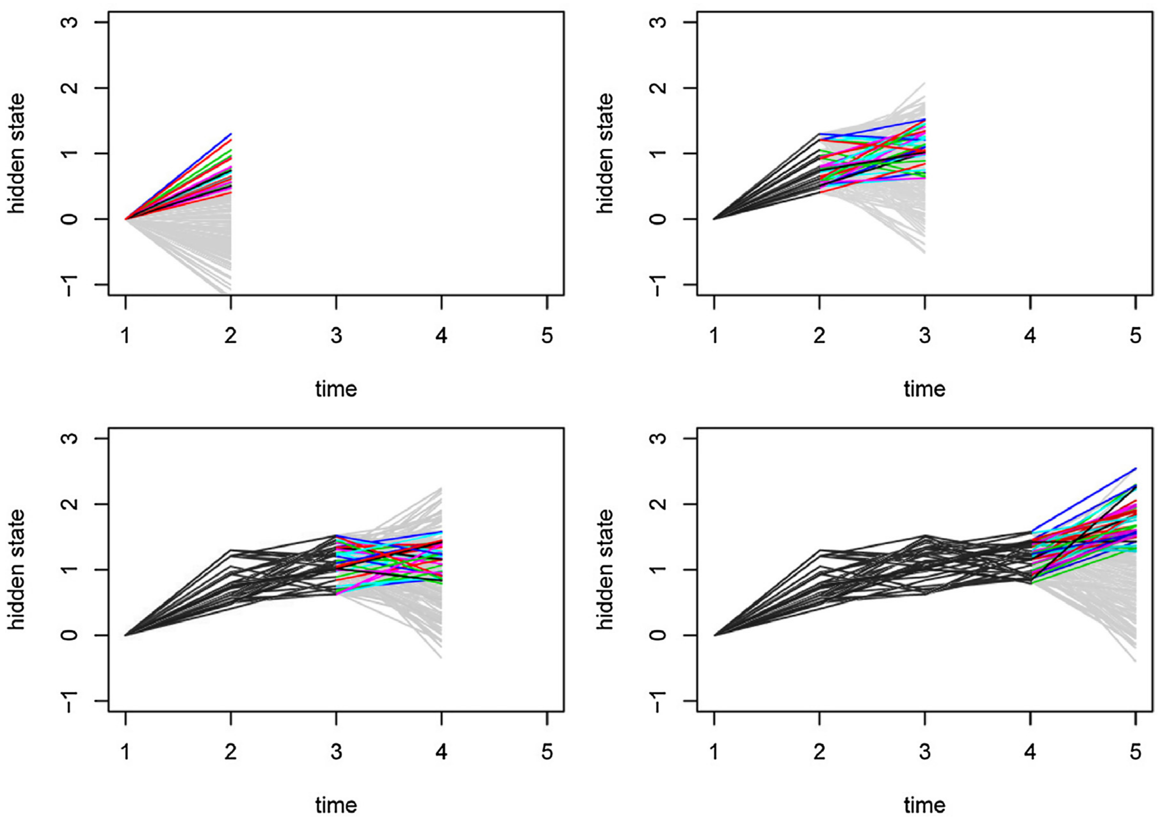

Fig. 3. Sequential Monte Carlo (SMC) procedures applied to the random-walk model. Candidate particles are generated by the time evolution process f_θ and then filtered given observation g_θ. Coloured and grey lines correspond to accepted and rejected particles, respectively.

SMC to generate samples $\{X_{1:T}\}$ and the approximated marginal likelihood $\hat{p}(y_{1:T}|\theta^{(0)})$. Randomly choose one trajectory $x_{1:T}^{(0)}$ from $\{X_{1:T}\}$.

2. For step $n \geq 1$, propose a new parameter value $\theta^{*(n)}$ by sampling from the proposal distribution $q(\theta^{*(n)}; \theta^{(n-1)})$ based on the previous value $\theta^{(n-1)}$.
3. Run SMC to generate samples $\{X_{1:T}^*\}$ and the approximated marginal likelihood $\hat{p}(y_{1:T}|\theta^{*(n)})$. Randomly choose one trajectory $x_{1:T}^{*(n)}$ from $\{X_{1:T}^*\}$ as a candidate for a MCMC sample for this step[4] .
4. Compare the marginal likelihood with that in the previous step $\hat{p}(y_{1:T}|\theta_{n-1})$. With probability
$$\min\left[1, \frac{\hat{p}(y_{1:T}|\theta_n^*)}{\hat{p}(y_{1:T}|\theta_{n-1})}\frac{q(\theta_{n-1}; \theta_n^*)}{q(\theta_n^*; \theta_{n-1})}\right], \tag{9}$$
update $\theta^{(n)} = \theta^{*(n)}$ and $x_{1:T}^{(n)} = x_{1:T}^{*(n)}$. Otherwise, keep the values from the previous step: $\theta^{(n)} = \theta^{(n-1)}$ and $x_{1:T}^{(n)} = x_{1:T}^{(n-1)}$.
5. Repeat 2.-4. until the Markov-chain converges.

Here, SMC only serves as a function which returns the approximated marginal likelihood $\hat{p}(y_{1:T}|\theta^{*(n)})$ and a sample of $x_{1:T}$ (the algorithm of SMC is detailed in Appendix B). The other part of the algorithm is the same as the Metropolis-Hastings MCMC. An experienced user of MCMC can easily incorporate most known MCMC techniques, e.g., block sampling, adaptive MCMC or data augmentation. One exception is Gibbs sampling: if the conditional samplers for θ and x (i.e., $p(\theta|x_{1:T}, y_{1:T})$ and $p(x_{1:T}|\theta, y_{1:T})$) are available, one may wish to alternately sample θ and x to improve the efficiency of the MCMC steps. However, combining PMCMC and Gibbs sampling requires a specific technique which Andrieu et al. referred to as "the conditional SMC update". For details of this procedure, refer to the section "Particle Gibbs sampler" in the original paper (Andrieu and Doucet, 2010).

There is an important property that must be noted about the approximated marginal likelihood $\hat{p}(y_{1:T}|\theta^{*(n)})$ returned by SMC. So far, to communicate a handy overview on the idea of PMCMC, we have not put much emphasis on the fact that $\hat{p}(y_{1:T}|\theta^{*(n)})$ is not the exact marginal likelihood but an approximation, which randomly fluctuates at each implementation due to its Monte Carlo nature. PMCMC works in the exact manner we have illustrated if the employed SMC is well designed and returns sufficiently accurate $\hat{p}(y_{1:T}|\theta^{*(n)})$; even if this does not hold, repeated implementation of SMC in the MCMC iterations assures that the final MCMC samples converge to the target distribution $p(\theta, x_{1:T}|y_{1:T})$. It has been shown that the number of particles used in SMC, which determines how good an approximation is, does not affect the distribution to which PMCMC is converging. This property implies that an SMC employed in PMCMC does not need to be tuned as carefully as one directly applied to inference problems on its own. Section 4 in the original paper (Andrieu and Doucet, 2010) by Andrieu et al. provides a formal demonstration.

Algorithm 1. Particle Markov-chain Monte Carlo (PMCMC)

```
$\theta^{(0)} \longleftarrow \theta_0$
$\hat{p}_\theta(y_{1:t}), \{X_{1:T}\} \longleftarrow \mathrm{SMC}(\theta^{(0)})$
$\pi^{(0)} \longleftarrow \hat{p}_\theta(y_{1:t})$
sample $x_{1:T}^{(0)} \sim \{X_{1:T}\}$
  for $n = 1, .., ., N$ do
  $\theta^{*(n)} \sim q(\cdot|\theta^{(n-1)})$
  $\hat{p}_\theta(y_{1:t}), \{X_{1:T}\} \longleftarrow \mathrm{SMC}(\theta^{*(n)})$
  $\pi^{*(n)} \longleftarrow \hat{p}_\theta(y_{1:t})$
  sample $x_{1:T}^{*(n)} \sim \{X_{1:T}\}$
  $U \longleftarrow \min\left\{1, \frac{\pi^{*(n)}}{\pi^{(n-1)}} \cdot \frac{q(\theta^{(n-1)} \mid \theta^{*(n)})}{q(\theta^{*(n)} \mid \theta^{(n-1)})}\right\}$
```

[4] Only one sample from $\{X_{1:T}^*\}$ is stored to construct a MCMC chain for $x_{1:T}$. Random sampling at each MCMC step assures that the resulting MCMC chain is invariant

```
    $u \longleftarrow Unif(0, 1)$
    if $u < U$ then
        $\theta^{(n)} \longleftarrow \theta^{*(n)}$
        $x_{1:T}^{(n)} \longleftarrow x_{1:T}^{*(n)}$
        $\pi^{(n)} \longleftarrow \pi^{*(n)}$
    else
        $\theta^{(n)} \longleftarrow \theta^{(n-1)}$
        $x_{1:T}^{(n)} \longleftarrow x_{1:T}^{(n-1)}$
        $\pi^{(n)} \longleftarrow \pi^{(n-1)}$
    end if
end for
```

2.5. *Practical considerations for implementation*

2.5.1. *Tuning of PMCMC: the number of particles*

An important decision when performing PMCMC is the number of particles to use when fitting to the data (Pitt et al., 2012). Too few particles will lead to a large variation in the marginal likelihood estimation and, therefore, to poor mixing in the MCMC chain. On the other hand, increasing the number of particles results in a higher computational load. If the computational load is tolerable, one could choose an arbitrary number (which is sufficiently large for SMC to approximate the marginal likelihood); in a typical inference problem, 100-1000 particles usually work well. Otherwise, one could also optimise the number of particles using certain package functionalities (e.g., RBi.-helpers). A general rule-of-thumb adopted by the RBi.helpers R package (see Section 2.5.4) is to choose the number of particles which keeps the variance of the log-likelihood around the mode below one.

2.5.2. *Computation time*

The computation time of PMCMC is highly dependent on the model you are trying to fit, but in typical cases, fitting can easily take from a couple of hours to a day. The computational complexities of the model components (f and g), the number of particles employed and the tuning of the MCMC component (e.g., choosing an appropriate optimal proposal distribution) are important determinants of computational time and should be considered to optimise PMCMC. Particle-based methods are relatively easier to parallelise; if multicore processors are available, one could parallelise the algorithm to reduce the total run time (which may be automatically done by some packages). Also note that some tools (e.g., LibBi) support the use of graphics processing units (GPUs), which have far greater computational performance than ordinary central processing units (CPUs) in terms of parallelisation (Murray, 2012, 2013; Murray et al., 2016). Although using GPUs may require additional technical knowledge, it can substantially reduce the computational time (even by a factor of 50-100 in some cases) if implemented successfully.

2.5.3. *Using packages vs writing from scratch*

There are a number of open source implementations of the PMCMC algorithm available, such as LibBi, RBi, NIMBLE and POMP (see the next section). Which one to use depends a lot on what environment you are comfortable with. It is also possible to write an implementation from scratch, but this can be challenging as it requires both technical skills and a thorough understanding of the algorithm. Meanwhile, writing one or two simple PMCMC implementations (e.g., examples in this paper) from scratch will be helpful for a beginner to get familiar with the method. Once you are clearer of the overall picture by prototyping simple working examples, you can reduce the amount of work by using existing tools while avoiding the risk of misuse.

2.5.4. *Useful tools*

Here are some of the tools available for the implementation of PMCMC (though not inclusive), which may be useful for the readers who wish to try out PMCMC in their own studies. All listed tools support MCMC and SMC as well as PMCMC.

- LibBi (http://libbi.org/): a C+ + based software library for HMP (Murray, 2013)
- NIMBLE (https://r-nimble.org/): an R package for Bayesian inference based on the BUGS language [M. Auzenbergs et al.]
- pomp (https://kingaa.github.io/pomp/): an R package for HMP
- RBi (https://cran.r-project.org/web/packages/rbi/): an R package for using LibBi in R. Some useful functionalities are available in the RBi.helpers package (https://github.com/sbfnk/RBi.helpers).

3. Examples

In this section, we provide two examples of PMCMC implementation from epidemic modelling. The first example is the Reed-Frost model, a simple epidemic model with stochasticity. The second example is the Dureau model, a compartmental model with time-varying parameters to model transmission. The first example will illustrate how PMCMC (and SMC as a part of it) is combined with epidemic models, and the second example will show an application of PMCMC to more complex models, which may be more likely to be used in real epidemic settings. For each example, implementation procedures and practical considerations (including how PMCMC was tuned) are discussed.

3.1. *Reed-Frost model*

3.1.1. *Model and Data*

The Reed-Frost model is a discrete-time stochastic model with each time step representing a new generation of infectious agents (Abbey, 1952), which can be understood as a discrete, stochastic version of Susceptible-Infectious-Recovered model. During the period between two time steps $t-1$ and t (one generation), each of the susceptible individuals has a probability p of getting infected by any of the infectious individuals. The probability of escaping infection from all the infectious individuals is $(1-p)^{I_{t-1}}$ and thus the number of infectious individuals at the next time step (i.e. the ones who did not escape infection) can be drawn from a binomial distribution with probability $1-(1-p)^{I_{t-1}}$.

$$\begin{aligned} I_t &\sim \mathrm{Bin}(S_{t-1}, 1-(1-p)^{I_{t-1}}), \\ S_t &= S_{t-1} - I_t. \end{aligned} \tag{10}$$

The Reed-Frost model can be made into a hidden Markov process (HMP) by using (S_n, I_n) as the state variables and employing an observation process. Here we assume that we only detect a fraction of the infected individuals; the observed case counts are assumed to follow a negative binomial distribution of size s with a mean probability of detection p_{obs}. The HMP is then represented as

$$\begin{aligned} X_t &= (S_t, I_t), \\ Y_t &\sim \mathrm{NegBin}(\mathrm{mean} = p_{\mathrm{obs}} \cdot I_t, \mathrm{size} = s). \end{aligned} \tag{11}$$

We generated a simulated dataset with the model (Eq. (11)), and fitted the model using both SMC and PMCMC. The total population N $(= S + I)$ was set as $N = 10,000$, and the initial state (S_1, I_1) was set by drawing from a Poisson distribution, i.e., $I_1 \sim \mathrm{Pois}(\lambda = 5)$, $S_1 = N - I_1$. The person-to-person probability of infection p was set at $p = 0.00015$. A simulated data of fifty time steps was produced.

3.1.2. *Implementation and tuning*

As the computational load of the model was trivial, the number of particles was simply set to a sufficiently large number (1,000 particles). The code was written from scratch and can be found on a Github repository (https://github.com/akira-endo/Intro-PMCMC).

3.1.3. *Results*

3.1.3.1. *SMC*. Using the true parameter value $p = 0.00015$ for the probability of transmission, we applied a SMC to the simulated data. The inference of the hidden states is shown in Fig. 4. The algorithm can

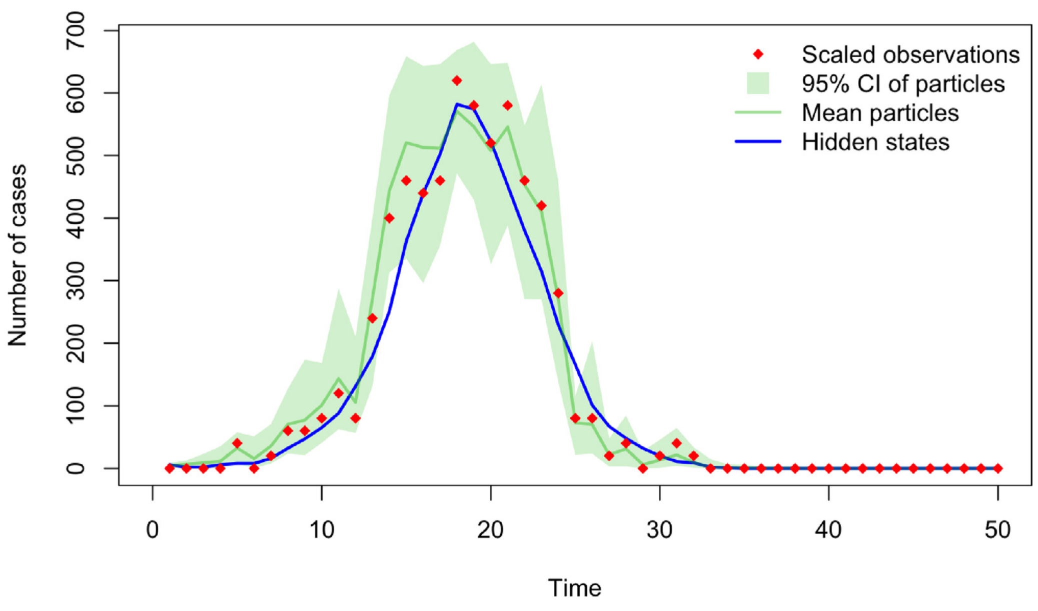

Fig. 4. Inference of the Reed-Frost model by SMC. The blue line represents the hidden states of the Reed-Frost model. The red diamonds are the observations from the system. The green lines are the mean trajectories of the particles resulting from the SMC algorithm, and the green ribbons denote the 95% credible intervals of these particles.

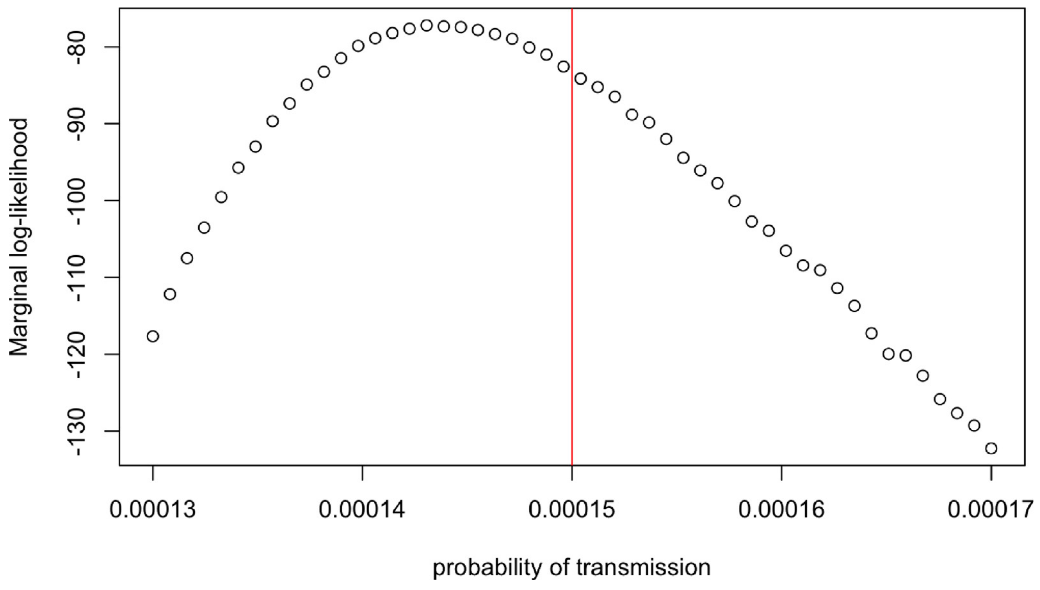

Fig. 5. Marginal log-likelihood calculated for some values of the transmission parameter near the “true” parameter used to generate the simulated data. The SMC algorithm with 5,000 particles was used. The vertical red line indicates the transmission parameter used to generate the simulated data.

reconstruct the hidden states, which can be in return used to compute the marginal likelihood. In Fig. 5, the marginal log-likelihood is calculated for different value of the transmission parameter. With enough particles the estimation provides a smooth profile of the marginal likelihood as expected.

3.1.3.2. PMCMC. PMCMC (with the Metropolis-Hastings MCMC algorithm) was performed to estimate the parameter *p*. The PMCMC algorithm is able to perform inference of the parameters from the model by using the marginal likelihood approximation computed from the SMC step. In Fig. 6, an example of such a Markov chain is shown. The chain is able to explore the region of a high density of the posterior for the parameters of the model after convergence.

3.2. Dureau model

3.2.1. Introduction

This example is a re-implementation of the model first presented by Dureau et al. (2013). The pandemic influenza epidemic in 2009 in the UK exhibited two waves, which cannot be explained by the typical SEIR type of mechanisms used to describe influenza transmission and the study explored whether this could be explained by changes in the transmission rate over time, due to school holidays. They explored a number of models. In this example, we will explore their simplest model. The data used was a weekly time-series of the clinical cases derived from the official number provided by Public Health England (Baguelin et al., 2010) in the UK during the 2009 pandemic. The code and data for this example can be found on a Github repository (https://github.com/akira-endo/Intro-PMCMC)

3.2.2. Model

The model is borrowed from the original study (Dureau et al., 2013). The modelled states are: susceptibles (*S*), exposed (*E*), infected (*I*) and recovered (*R*).

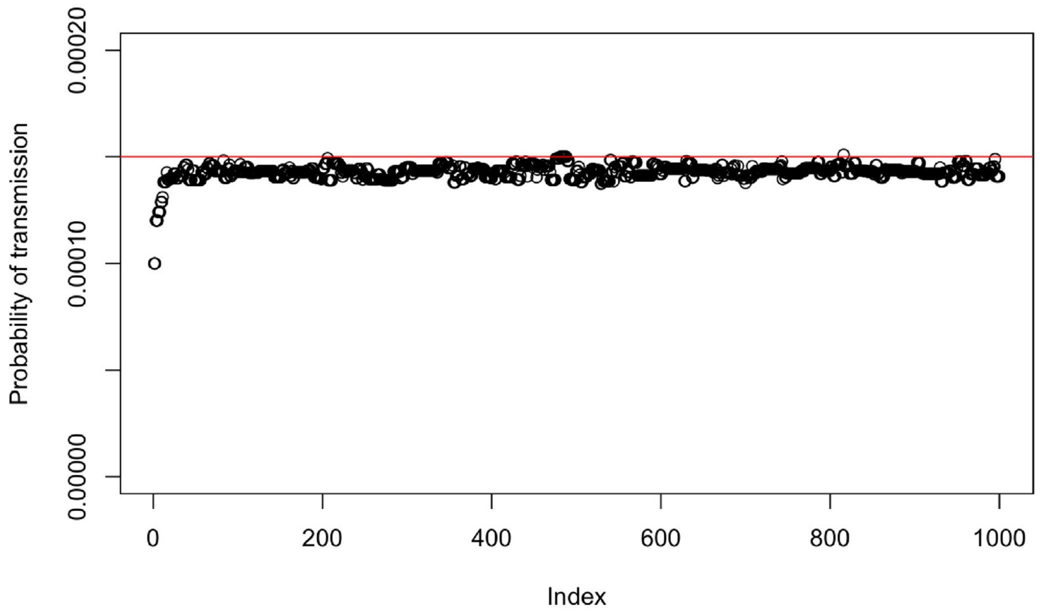

Fig. 6. Markov chain for the transmission parameter. The chain is expected to consist posterior samples based on the likelihood approximated by SMC (shown in Fig. 5).

$$
\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= -\beta S(t)\frac{I(t)}{N}\\
\frac{\mathrm{d}E}{\mathrm{d}t} &= \beta S(t)\frac{I(t)}{N} - \mathrm{k}E(t)\\
\frac{\mathrm{d}I}{\mathrm{d}t} &= \mathrm{k}E(t) - \gamma I(t)\\
\frac{\mathrm{d}R}{\mathrm{d}t} &= \gamma I(t)\\
\frac{\mathrm{d}Z}{\mathrm{d}t} &= \mathrm{k}E(t) - Z\delta(t \bmod 7)\\
\frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma \mathrm{dW}\\
\beta &= e^{x(t)}
\end{aligned}
\tag{12}
$$

with Z the cumulative number of new infections per seven days and x the log-transformed transmissibility. $\delta(t\text{mod}7)$ is the Dirac delta function, which causes Z to reset to zero when $t\text{mod}7 = 0$, i.e., every seven days. The latent and infectious periods correspond to $\frac{1}{k}$ and $\frac{1}{\gamma}$, respectively. The amplitude of the stochasticity in the transmission rate is controlled by the parameter σ.

3.2.3. Likelihood

The likelihood of the data was assumed to be log-normally distributed in the original manuscript. Not everyone infected ends up visiting a GP, either due to asymptomatic infections or the patients' likelihood to consult a GP. Therefore the authors assumed that the true incidence was ten times higher than the number of GP visits. This correction was further supported by a serological survey (see Dureau et al., 2013 for further details).

$$\mathcal{L}(Z|y_i, \tau) = \mathcal{N}(\log(y_i), \log(Z/10), \tau)$$

3.2.4. Priors

Most parameters in the model have a wide uniform prior, except for the latent period, which was assumed to be normal distributed with mean 1.59 and standard deviation 0.02, the infectious period, which was normal distributed with mean 1.08 and standard deviation 0.075 and the initial recovered (immune) population, which was normal distributed with a mean of 0.15 and standard deviation of 0.15 (truncated between 0 and 1; Dureau et al., 2013).

3.2.5. Implementation and tuning

The model was fitted using the RBi and RBi.helpers packages, which were designed to interface with LibBi from within R (Murray, 2013). To ensure efficient mixing of the PMCMC algorithm we took the following approach. First, we took 1000 samples with the proposal distribution set to sample from the prior. This means that instead of taking proposal samples close to the current parameter values, proposal samples are taken from the whole prior distribution. As a result, the algorithm sampled from a much wider distribution, making it more likely to find a good starting value. This is a sensible approach if you have no a priori indication of a valid starting value.

Next, we called the adapt_particles and adapt_proposal functions provided by the RBi.helpers package. The first function runs the model with an increasing number of particles until the variance around the log likelihood is lower than 1. This ensured that the variance is not so high that it hampers mixing, but still kept the number of particles as low as possible, to reduce computational load. We started with a low number of particles (16 particles), and after running adapt_particles we found that 128 particles were sufficient to realise low log-likelihood variance. The adapt_proposal function adapts the parameter proposals, such that the number of accepted samples during the MCMC step is reasonable (in this case we set the acceptable limits between 0.05 and 0.4 of the proposals accepted). After these adaptions, we ran the inference for 5000 samples as a burn-in period. Finally we ran the full inference with 5000 samples and thinning every 5 samples, resulting in a 1000 posterior samples in total. The run time was just over 1 hour on a workstation from 2015, using 8 cores in parallel.

3.2.6. Results

Fig. 7 shows the results of the data fitting. The top panel shows the incidence data (red dots), with two distinct epidemic waves, and the model prediction. As shown, the model was able to reproduce both waves. The middle panel shows transmissibility over time, with an apparent dip between day 50 and 100. This dip can be confirmed by comparing the transmissibility to the transmissibility at time 0 (bottom panel), which shows that between day 50 and 100 the transmissibility is below the starting transmissibility in all cases. The dip in transmissibility coincides with the holiday period and the decline of the first epidemiological wave.

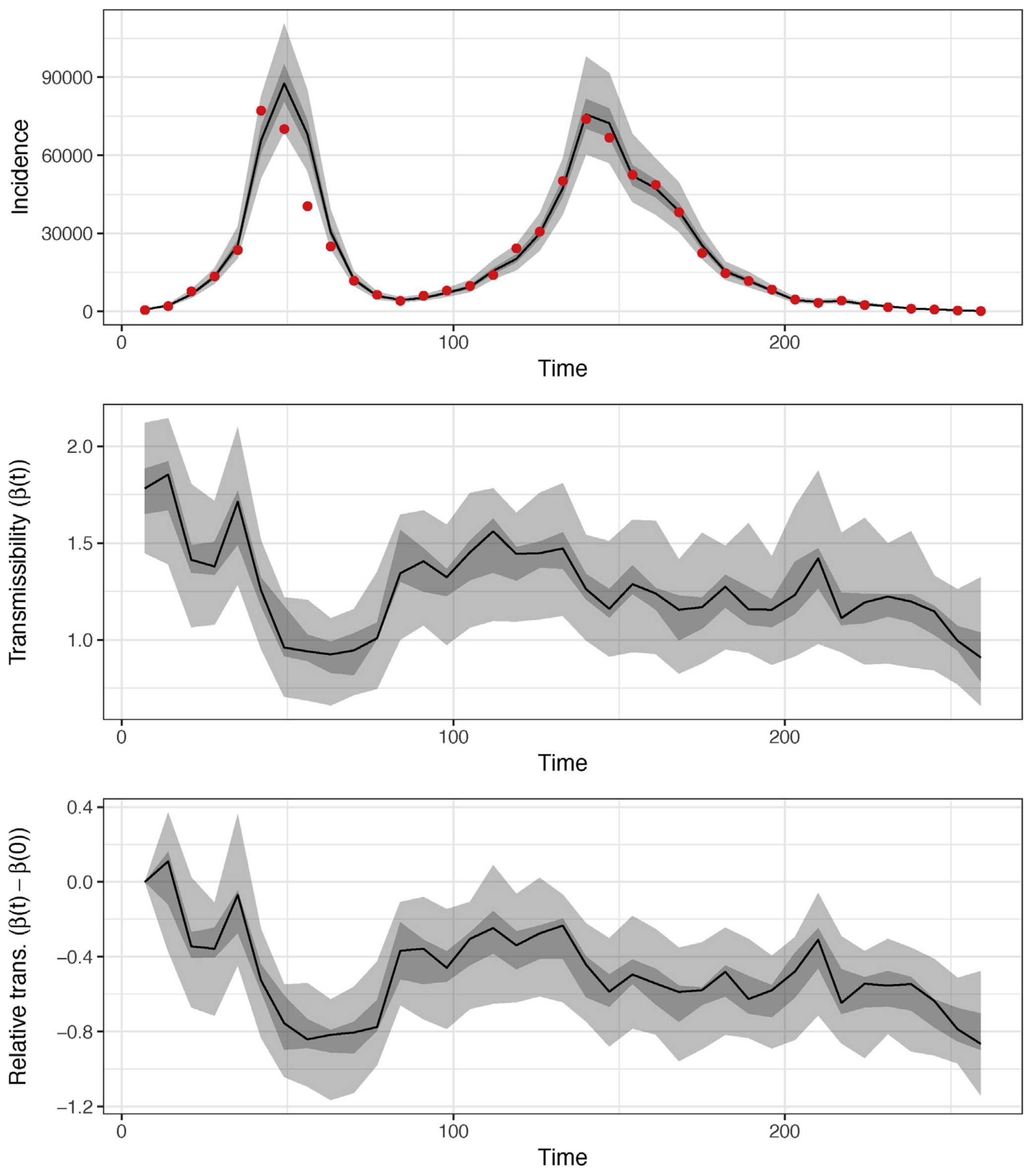

Fig. 7. Model inference results. Top panel shows the GP consultation results, with the points showing the actual data points. The ribbons represent the 95% and 50% credible intervals in incidence, and the black line shows the median. The middle panel shows the transmissibility over time and the bottom panel the change in transmissibility relative to the initial transmissibility.

4. Discussion

PMCMC is an estimation method that combines SMC with MCMC and is especially suited for inference of HMP, whose likelihood is usually intractable. PMCMC uses particles to efficiently sample plausible trajectories of hidden state variables to integrate them out, while (potentially high-dimensional) parameters are explored by the MCMC framework. As HMP is a model with time evolution, their parameters and hidden variables are usually intercorrelated. When applied to HMPs, PMCMC generally performs better than data-augmentation MCMC, which typically exhibits slow mixing when parameters and hidden variables are highly correlated.

Infectious disease dynamics often involve stochastic time evolution of unobserved variables (e.g., the true number of infections which may not be fully reported). However, deterministic SIR model, which is widely used for computational convenience, neglects the possible stochastic fluctuation of the system. Deterministic models are also unable to capture stochastic behaviours in situations with small numbers (e.g., extinction) (Roberts et al., 2015). PMCMC will be a feasible approach to such complex models with stochasticity. Camacho et al. (2015) estimated the temporal trend of Ebola virus disease in Sierra Leone using a stochastic SEIR model. They accounted for temporal variation in the transmission rates which was assumed to follow the Wiener process, and PMCMC is well suited for such model settings. A similar framework was also used for real-time forecasting of the Ebola outbreak in Africa (Funk et al., 2018). Other applications include a multi-host dynamics model of bubonic plague (Didelot et al., 2017) and a phylodynamic analysis of a poliovirus outbreak (Li et al., 2017).

Other related methods may be used in place of PMCMC. Iterated filtering (Ionides et al., 2006, 2015) is a frequentist-type alternative of PMCMC, which finds the maximum likelihood estimate instead of posterior samples. The extended Kalman filter (Jazwinski, 1970) could replace the SMC component in PMCMC; this method is computationally simpler than SMC but is subject to certain assumptions (Gaussianity and approximate linearity). SMC^2 (Chopin et al., 2013) is a method based on a similar framework to PMCMC, where the MCMC component of PMCMC is replaced with another SMC. That is, instead of iteratively updating parameter θ, SMC^2 generates a set of particles in a joint space

$\theta \times x$ and then filters them to approximate a joint distribution $p(x, \theta|y)$. SMC^2 may be particularly useful when the original particle distribution is localised due to informative priors.

The limitations of PMCMC must be noted. PMCMC is an extension of MCMC and thus it inherits the weaknesses of MCMC including high computational load, the requirement of a well-chosen proposal distribution and slow mixing in the presence of correlation between parameters. PMCMC requires that the time-evolution and observation processes (f and g) are given and tractable; if they are intractable, alternate likelihood-free methods need to be sought, e.g., approximate Bayesian computation (ABC). Moreover, as PMCMC uses a finite number of particles, the approximation can become poor when the hidden state x is high-dimensional (e.g., a spatiotemporal model with a large number of spatial units). In such cases, parametric approaches including the extended Kalman filter and integrated nested Laplace approximation (Rue et al., 2009; Blangiardo et al., 2013) may be more appropriate.

When applied to the appropriate subset of inference problems, PMCMC can serve as an efficient sampling method for HMPs with intractable likelihood. PMCMC enables infectious disease modellers to employ more complex epidemic models to account for the stochastic and partially-observed nature of real-world outbreaks.

Acknowledgement

We thank Naomi R Waterlow for proofreading the manuscript. AE receives financial support fromThe Nakajima Foundation. The authors thank the UK National Institute for Health Research Health Protection Research Unit (NIHR HPRU) in Modelling Methodology at Imperial College London in partnership with Public Health England (PHE) for funding (grant HPRU-2012-10080). EvL also acknowledges the NIHR HPRU Respiratory Infections for funding. The funders had no role in conceptualisation, preparation of the manuscript or decision to publish. The authors declare that no competing interest exists.

Appendix A. Importance of Markov property for the use of SMC

The Markov property of HMP, which assures that $f(x_{t+1}|x_t)$ is a function of only the current state x_t and not of the past states, provides a useful attribute in estimation. In sequential Monte Carlo (SMC), which is employed as a sub-algorithm of PMCMC, the observation y_t is used to infer the hidden state x_t (inferring the hidden states from the observations is sometimes referred in the literature as “filtering”). Due to the Markov property, y_t is independent of the previous states $x_{1:t-1}$ if x_t is given. This means, inversely, that the observation of the future y_{t+c} $(c \geq 1)$ does not affect the posterior probability of the current state x_t given x_{t+1}. It is shown as follows, noting that $p(y_{t+c}|x_{t+1}, x_t) = p(y_{t+c}|x_{t+1})$,

$$\begin{aligned} p(x_t|x_{t+1}, y_{t+c}) &= \frac{p(y_{t+c}|x_{t+1})p(x_{t+1}|x_t)p(x_t)}{p(x_{t+1}, y_{t+c})} \\ &= p(x_t|x_{t+1}). \end{aligned} \tag{13}$$

In the SMC algorithm, as seen in Sections 2.3 and B, x_t is sampled sequentially in a forward direction. Eq. (13) promises that once x_t is sampled and fixed, subsequent observations $y_{t:T}$ does not directly affect the distribution of the past states $x_{1:t-1}$. A Bayesian update of the distribution of x_t alone can thus represent all the possible influence of y_t on $x_{1:t}$, allowing for the sequential sampling without moving back and forth.

Appendix B. Inside structure of SMC

For readers interested in the technical aspects of SMC, here we outline the most basic SMC algorithm: the bootstrap filter. The bootstrap filter (BF) is a special case of methods called sequential importance (re)sampling (SIS/SIR), which infer hidden state variables by sequentially filtering particles based on importance sampling (IS). After introducing the basic settings, we first describe the concept of IS and then introduce SIS/SIR algorithms using BF as an example. Lastly, we mention some approaches to addressing the so-called particle degeneracy problem, which may cause the SMC algorithms to fail.

B.1 Basic settings of SMC

SMC (also known as particle filtering) is a method particularly designed for inference on hidden Markov processes (HMPs; see the main text, Section 2.2). SMC can efficiently produce samples of the hidden states $x_{1:T}$ from the conditional distribution $p_\theta(x_{1:T}|y_{1:T})$, and also approximate the marginal likelihood $p_\theta(y_{1:T})$. Data $y_{1:T}$, the initial state samples from the prior distribution $p_\theta(x_1)$ and the time evolution/observation processes $f_\theta(x_t|x_{t-1})$ and $g_\theta(y_t|x_t)$ need to be defined for SMC. SMC then sequentially samples hidden state variables x_t from the step-wise conditional probability $p_\theta(x_t|y_{1:t})$. Note that the contribution of θ to the system is fully represented by f_θ and g_θ. Therefore, within SMC, we can assume that θ is fixed (so are f_θ and g_θ). The notation $p_\theta(\cdot) = p(\cdot|\theta)$ is used to indicate that θ conditions the probability but can be left out from consideration.

In SMC, samples of x_t are called “particles”. The basic idea of SMC is to sequentially update the distribution of x_t with f_θ, g_θ and $y_{1:t}$ for $t = 1, 2, ..., T$ to produce a set of “time-series particles” $x_{1:T}$. Ignoring the actual procedures to generate samples from given distributions for now, the algorithm of SMC is described as follows:

1. For $t = 1$, sample the initial state x_1 from

$$p_\theta(x_1|y_1) \propto g_\theta(y_1|x_1)p_\theta(x_1), \tag{14}$$

and compute the initial marginal likelihood as

$$p_\theta(y_1) = \int g_\theta(y_1|x_1)p_\theta(x_1)\mathrm{d}x_1. \tag{15}$$

2. For $t = 2, .. ., T$, given the previous state $p_\theta(x_{t-1}|y_{1:t-1})$, sample x_t from

$$\begin{aligned} p_\theta(x_t|y_{1:t}) &\propto g_\theta(y_t|x_t)p_\theta(x_t|y_{1:t-1}) \\ &= g_\theta(y_t|x_t)\int f_\theta(x_t|x_{t-1})p_\theta(x_{t-1}|y_{1:t-1})\mathrm{d}x_{t-1}, \end{aligned} \tag{16}$$

and compute the marginal likelihood as

$$\begin{aligned} p_\theta(y_{1:t}) &= p_\theta(y_{1:t-1})p_\theta(y_t|y_{1:t-1}) \\ &= p_\theta(y_{1:t-1})\int g_\theta(y_t|x_t)p_\theta(x_t|y_{1:t-1})\mathrm{d}x_t. \end{aligned} \tag{17}$$

Although the formulations are relatively straightforward, it is not always easy to directly sample from these distributions. The key concept, which SMC uses to sample from the distribution defined as above, is to generate and then filter particles each time step so that the resulting particles represent the target distribution (this is where the term "particle filtering" originates). First, candidate particles $\{X'_t\}$ are randomly generated. Particles for the previous state $\{X_{t-1}\}$ and the time evolution process f_θ are usually used to generate $\{X'_t\}$. $\{X'_t\}$ does not generally follow the target distribution $p_\theta(x_t|y_{1:t})$; SMC produces from $\{X'_t\}$ a new set of samples $\{X_t\}$ that represents the target distribution $p_\theta(x_t|y_{1:t})$. This process of converting the candidate particles $\{X'_t\}$ into the proper particles $\{X_t\}$ is referred to as "filtering". In the most standard SMC, the bootstrap filter (BF), one of the resampling algorithms called importance sampling resampling (ISR), is used to filter particles.

B.2 Importance sampling/importance sampling resampling

Importance sampling (IS) and importance sampling resampling (ISR) are generic sampling methods for probabilistic distributions which are difficult to sample from but whose density functions are available. Let $\pi(x)$ be the target distribution from which we cannot directly sample. Instead of sampling from $\pi(x)$, an easily-sampled distribution $\pi'(x)$ is used as a proposal distribution. Samples of size N generated from $\pi'(x)$, $\{X'^{(i)}\}$ $(i = 1, 2, .. ., N)$, are then given "importance weights" $w^{(i)}$ such that

$$\begin{aligned} w^{(i)} &\propto \frac{\pi(X'^{(i)})}{\pi'(X'^{(i)})}, \\ \sum_{i=1}^{N} w^{(i)} &= 1. \end{aligned} \tag{18}$$

Combined with the importance weights, $\{w^{(i)}, X'^{(i)}\}$ is considered as a (weighted) sample set following the target distribution $\pi(x)$.

Theoretically, the proposal distribution $\pi'(x)$ can be arbitrarily chosen if a sufficient number of samples $\{X'^{(i)}\}$ are generated. However in practice, $\pi'(x)$ must be similar to $\pi(x)$ so that a finite number of weighted samples $\{w^{(i)}, X'^{(i)}\}$ closely approximates the target distribution. If $\pi(x)$ and $\pi'(x)$ are very different, only a few samples in $\{X'^{(i)}\}$ will fall within the likely region of $\pi(x)$, and the weights of most samples become nearly zero. Meanwhile, x with too low $\pi'(x)$ to be realised is never sampled regardless of $\pi(x)$. Importance sampling with an ill-chosen proposal distribution (which usually yields bipolarised weights) results in a scarcity of informative samples, and thus the weighted samples do not properly represent the target distribution (Fig. 8). Choosing a proposal distribution that resembles the target distribution is important to ensure that the weighted samples are representative. Kish's effective sample size (ESS) (Kish, 1965) is one of the useful indicators to evaluate whether a weighted sample set is a good approximation of the target distribution.

Weighted samples can be bothersome to handle. In such cases, weighted samples obtained from IS are resampled to yield unweighted samples: N samples are drawn from the weighted samples $\{w^{(i)}, X'^{(i)}\}$ with replacement, and then the new samples $\{X^{(i)}\}$ will be an unweighted sample set following $\pi(x)$. By resampling, particles with low weights are filtered out, so that the more informative samples remain in the set of particles. This approach is called importance sampling resampling (ISR).

B.3 Sequential importance sampling/resampling and the bootstrap filter

Sequential importance sampling (SIS) or sequential importance resampling (SIR) are natural extensions of IS/ISR for the SMC framework. They sequentially generate particles following the conditional distribution $p_\theta(x_t|y_{1:t})$ using the particles of the previous state ($\{w_{t-1}^{(i)}, X_{t-1}'^{(i)}\}$ in SIS, $\{X_{t-1}^{(i)}\}$ in SIR) which follows $p_\theta(x_{t-1}|y_{1:t-1})$. Although SIS is faster than SIR because it can avoid the computational burden caused by the resampling process, SIS is known to be ill-performing for longer time series due to the so-called "particle degeneracy" problem. We have seen in Fig. 8 that in IS, the resulting weighted samples are not representative when the proposal distribution is not close enough to the target distribution. Unlike SIR, which "resets" the weight w_t each step, SIS keeps updating the weight w_{t-1} sequentially over time. Because SIS repeatedly implements IS, which always decreases the ESS of the weighted samples (unless $\pi(x) = \pi'(x)$), at the limit $T \to \infty$ the ESS of the SIS samples converge to 1, i.e., all samples but one are effectively lost. SIR is preferred to SIS as an algorithm for SMC for this reason, and thus we limit our focus to the bootstrap filter (BF) in this

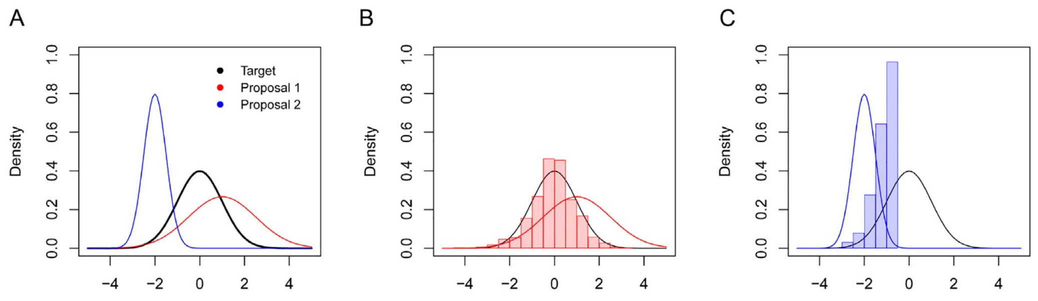

Fig. 8. Importance sampling resampling (ISR) using different proposal distributions. (A) The target distribution $\mathcal{N}(0, 1)$ and the proposal distributions $\mathcal{N}(1, 1.5)$ and $\mathcal{N}(-2, 0.5)$. (B) (C) Samples obtained by the importance sampling resampling method. For each proposal distribution, 500 samples were drawn, weighted and resampled.

paper. See, for example, Doucet and Johansen (2011) for the SIS algorithm and its weaknesses.

Equation (16) implies that samples obtained by applying f_θ to the previous particles, $X_t'^{(i)} \sim f_\theta(x_t|X_{t-1}^{(i)})$, can be used as a proposal distribution for ISR. Noting that $g_\theta(y_t|x_t) \propto \frac{p_\theta(x_t \mid y_{1:t})}{p_\theta(x_t \mid y_{1:t-1})}$, the importance weights $w_t^{(i)}$ is computed from the observation probability g_θ as

$$\begin{aligned} w_t^{(i)} &\propto g_\theta(y_t|X_t'^{(i)}), \\ \sum_i w_t^{(i)} &= 1. \end{aligned} \tag{19}$$

Weighted particles $\{w_{t-1}^{(i)}, X_t'^{(i)}\}$ are then resampled according to $\{w_t^{(i)}\}$ to produce an unweighted sample set $\{X_t^{(i)}\}$. In the end, starting from the initial particles generated from $p_\theta(x_1)$, samples of hidden trajectories $\{X_{1:t}\}$ are obtained by sequentially applying ISR to the particles. f_θ is used to propose candidate particles, which are then weighted by g_θ and resampled. In the above algorithm, particles with very small weights are removed (filtered out) from the sample set; combined with its conceptual similarity to the bootstrap method, this type of SIR using $f_\theta(x_t|X_{t-1}^{(i)})$ as a proposal distribution is called the bootstrap filter.

Finally, an SMC algorithm employing BF is described below (see Algorithm 2 for a pseudocode).

1. For $t = 1$, approximate the initial marginal likelihood as the sum of g_θ for the initial prior samples $\{X_1'^{(i)}\} \sim p_\theta(x_1)$:

$$\hat{p}_\theta(y_1) = \sum_{i=1}^{N} g_\theta(y_1|X_1'^{(i)}), \tag{20}$$

   and obtain the posterior samples $\{X_1^{(i)}\}$ by resampling $\{X_1'^{(i)}\}$ with weights

$$w_1^{(i)} = \frac{g_\theta(y_1|X_1'^{(i)})}{\hat{p}_\theta(y_1)}. \tag{21}$$

2. For $t = 2, .. ., T$, generate samples $\{X_t'^{(i)}\}$ by applying f_θ to each of the previous samples $\{X_{t-1}^{(i)}\}$:

$$X_t'^{(i)} \sim f_\theta(\cdot|X_{t-1}^{(i)}), \tag{22}$$

   approximate the incremental marginal likelihood as

$$\hat{p}_\theta(y_t|y_{1:t-1}) = \frac{1}{N}\sum_{i=1}^{N} g_\theta(y_t|X_t'^{(i)}), \tag{23}$$

   and resample from $\{X_t'^{(i)}\}$ with weights

$$w_t^{(i)} = \frac{g_\theta(y_1|X_t'^{(i)})}{\hat{p}_\theta(y_t|y_{1:t-1})}, \tag{24}$$

   to obtain $\{X_t^{(i)}\}$. When resampling, also reorganise $\{X_{1:t-1}^{(i)}\}$ so that $\{X_{1:t}^{(i)}\}$ for each i constitutes a trajectory of the same particle.
3. After completing T steps, return the samples of trajectory $\{X_{1:t}^{(i)}\}$ and the approximate marginal likelihood $\hat{p}_\theta(y_{1:t}) = \hat{p}_\theta(y_1)\prod_{t=2}^{T}\hat{p}_\theta(y_t|y_{1:t-1})$

The algorithm presented above is the simplest BF algorithm, and there are many other variant algorithms for SMC, including

- Adaptive resampling: particles are resampled when only certain resampling criteria are satisfied to reduce the computational burden involved in resampling
- Auxillary particle filtering: future information y_{t+1} is also used to sample x_t for a better proposal distribution

both of which are introduced in Doucet and Johansen (2011).

Algorithm 2. Sequential Monte Carlo (SMC) with the bootstrap filter (BF)

```
Input: $\theta$, $N$
$X_1'^{(i)} \sim p_\theta(x_1)$, for $i = 1, .. ., N$
$w_1^{(i)} \longleftarrow g_\theta(y_1|X_1'^{(i)})$, for $i = 1, .. ., N$
$\bar{\boldsymbol{w}}_1 \longleftarrow \left(\frac{w_1^{(i)}}{\sum_{i=1}^{N} w_1^{(i)}}\right)_{i=1,...,N}$
$I^{(i)} \sim$ Multinom({1, ...,N}, $\bar{\boldsymbol{w}}_1$), for $i = 1, .. ., N$    ▷ *Multinomial resampling
$X_1^{(i)} \longleftarrow X_1'^{(I^{(i)})}$
$l_1 \longleftarrow \sum\hat{}_{i=1}^{N} w_1^{(i)}$
for $t = 2, .. ., T$ do
  $X_t'^{(i)} \sim f_\theta(\cdot|X_{t-1}^{(i)})$, for $i = 1, .. ., N$
  $w_t^{(i)} \longleftarrow g_\theta(y_t|X_t'^{(i)})$, for $i = 1, .. ., N$
  $\bar{\boldsymbol{w}}_1 \longleftarrow \left(\frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}\right)_{i=1,...,N}$
  $I^{(i)} \sim$ Multinom({1, ...,N}, $\bar{\boldsymbol{w}}_t$), for $i = 1, .. ., N$    ▷ *Multinomial resampling
  $X_t^{(i)} \longleftarrow X_t'^{(I^{(i)})}$
```

```
    $l_t \longleftarrow \sum_{i=1}^{\hat{N}} w_t^{(i)})$
end for
$\hat{p}_\theta(y_{1:T}) \longleftarrow \sum_{t=1}^{T} l_t$
Output:$\{X_{1:T}\}$, $\hat{p}_\theta(y_{1:T})$
```

*Other resampling methods may be used; see Section "Methods to handle diversity in particles".

B.4 Methods to handle diversity in particles

SIR methods, including BF, resample the particles at each time step to filter out particles corresponding to unlikely posterior probabilities $p_\theta(x_t|y_{1:t})$. Although this helps SIR to keep only particles within the plausible regions of $p_\theta(x_t|y_{1:t})$, it comes with another problem over time. Resampling with replacement results in replicate samples. This means, in the long run, particles with higher probabilities have so many duplicates that the resulting particles may lack diversity. This loss of diversity caused by resampling is also referred to as particle degeneracy. Possible options to alleviate this degeneracy is to use alternative resampling methods (residual, stratified or systematic resampling Douc and Cappe, 2005) in place of the multinomial resampling. Algorithm 3 shows the most common method: systematic resampling, where an interval [0,1] is divided into subregions proportional to the weights and equally-spaced N points are used to sample from those subregions. The resample-move method (Gilks and Berzuini, 2001) is another common approach that addresses the particle degeneracy. In the resample-move method, the resampled particles are jittered around the neighbouring region (See Doucet and Johansen, 2011; Gilks and Berzuini, 2001 for technical details). The resample-move method propagates the particles with higher weights so that the important regions of the posterior $p_\theta(x_t|y_{1:t})$ are more densely sampled while avoiding particle degeneracy.

Algorithm 3. Systematic resampling

```
Input: $\bar{w} = (w^{(1)}, ..., w^{(N)})$
$u \sim Unif(0, 1)$
for $i = 1, .. ., N$ do
    $I^{(i)} \longleftarrow$ (The smallest integer $I$ that satisfies $\frac{i-1+u}{N} \le \sum_{j=1}^{I} w^{(j)})$
end for
```

References

Abbey, H., 1952. An examination of the Reed-Frost theory of epidemics. Human Biol. 24 (3), 201–233 URL http://view.ncbi.nlm.nih.gov/pubmed/12990130.

Allen, L.J.S., 2008. An Introduction to Stochastic Epidemic Models. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 81–130. https://doi.org/10.1007/978-3-540-78911-6_3.

Andrieu, C., Doucet, A., 2019. Particle Markov chain Monte Carlo methods. J. Royal Stat. Soc.. Series B: Stat. Methodol.

Baguelin, M., Hoek, A.J.V., Jit, M., Flasche, S., White, P.J., Edmunds, W.J., 2010. Vaccination against Pandemic Influenza A/H1N1v in England: A Real-Time Economic Evaluation. Vaccine 28 (12), 2370–2384. https://doi.org/10.1016/j.vaccine.2010.01.002.

Blangiardo, M., Cameletti, M., Baio, G., Rue, H., 2013. Spatial and spatio-temporal models with R-INLA, Spatial and Spatio-temporal. Epidemiology. https://doi.org/10.1016/j.sste.2012.12.001.

Camacho, A., Kucharski, A., Aki-Sawyerr, Y., White, N.M., Flasche, S., Baguelin, M., Pollington, T., Carney, J.R., Glover, R., Smout, E., Tiffany, A., Edmunds, W.J., Funk, S., 2015. Temporal changes in ebola transmission in sierra leone and implications for control requirements: A real-time modelling study. PLoS Curr. https://doi.org/10.1371/currents.outbreaks.406ae55e83ec0b5193e3085.

Chopin, N., Jacob, P.E., Papaspiliopoulos, O., 2013. SMC2: An efficient algorithm for sequential analysis of state space models. J. Royal Stat. Soc. Series B: Stat. Methodol. https://doi.org/10.1111/j.1467-9868.2012.01046.x.

Cooper, B., Lipsitch, M., The analysis of hospital infection data using hidden Markov models, Biostatistics, doi:10.1093/biostatistics/5.2.223.

Didelot, X., Whittles, L.K., Hall, T., 2017. Model-based analysis of an outbreak of bubonic plague in Cairo in 1801. J Royal Soc Interface. https://doi.org/10.1098/rsif.2017.0160.

Douc, R., Cappe, O., 2005. Comparison of resampling schemes for particle filtering. ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005 64–69. https://doi.org/10.1109/ISPA.2005.195385.

A. Doucet, A.M. Johansen, A tutorial on particle filtering and smoothing: fifteen years later.

Doucet, A., de Freitas, N., Gordon, N., 2001. An Introduction to Sequential Monte Carlo Methods. Springer New York, New York, NY, pp. 3–14. https://doi.org/10.1007/978-1-4757-3437-9_1.

Dureau J., Kalogeropoulos K., Baguelin M., Capturing the time-varying drivers of an epidemic using stochastic dynamical systems 14 (3) 541-555. arXiv:23292757, doi:10.1093/biostatistics/kxs052. URL http://biostatistics.oxfordjournals.org/content/14/3/541.

Funk, S., Camacho, A., Kucharski, A.J., Eggo, R.M., Edmunds, W.J., 2018. Real-time forecasting of infectious disease dynamics with a stochastic semi-mechanistic model. Epidemics. https://doi.org/10.1016/j.epidem.2016.11.003.

Gelfand, A.E., Smith, A.F.M., 1990. Sampling-based approaches to calculating marginal densities. J Am Stat Assoc 85 (410), 398–409. http://www.jstor.org/stable/2289776.

Gilks, W.R., Berzuini, C., 2001. Following a moving target - Monte Carlo inference for dynamic Bayesian models. J. Royal Stat. Soc. Series B: Stat. Methodol. https://doi.org/10.1111/1467-9868.00280.

Hethcote, H.W., 2000. The mathematics of infectious diseases. SIAM Rev. 42, 599–653.

Ionides, E.L., Breto, C., King, A.A., Inference for nonlinear dynamical systems, Proc. Natl. Acad. Sci., doi:10.1073/pnas.0603181103.

E. L. Ionides, D. Nguyen, Y. AtchadLtdc:&rsquo;e:rtdc, S. Stoev, A.A. King, Inference for dynamic and latent variable models via iterated, perturbed Bayes maps, Proceedings of the National Academy of Sciencesdoi:10.1073/pnas.1410597112.

Jazwinski, 1970. Stochastic processes and filtering theory. https://doi.org/10.1109/TAC.1972.1100136.

Kish, L., 1965. Survey sampling /Leslie Kish. Wiley New York, Chichester.

Li, L.M., Grassly, N.C., Fraser, C., 2017. Quantifying transmission heterogeneity using both pathogen phylogenies and incidence time series. Mol Biol Evol. https://doi.org/10.1093/molbev/msx195.

MacDonald, I.L., Zucchini, W., Hidden Markov and other models for discrete-valued time series.

Murray, L.M., Lee, A., Jacob, P.E., 2016. Parallel Resampling in the Particle Filter. J. Comput. Graph. Stat. 25 (3), 789–805. https://doi.org/10.1080/10618600.2015.1062015.

Murray, L., 2012. GPU Acceleration of Runge-Kutta Integrators. IEEE Trans. Parallel Distrib. Syst. 23 (1), 94–101. https://doi.org/10.1109/TPDS.2011.61.

Murray, L.M., Bayesian State-Space Modelling on High-Performance Hardware Using LibBi, arXiv:1306.3277 [stat]arXiv:1306.3277.

Neal, P., Kypraios, T., 2015. Exact bayesian inference via data augmentation. Stat. Comput. 25 (2), 333–347. https://doi.org/10.1007/s11222-013-9435-z.

Pitt, M.K., dos Santos Silva, R., Giordani, P., Kohn, R., 2012. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. J. Econometr. 171 (2), 134–151. https://doi.org/10.1016/j.jeconom.2012.06.004. Bayesian Models, Methods and Applications.

Press, W., Teukolsky, S., Vetterling, W., Flannery, B., 2007. Numerical Recipes: The Art of Scientific Computing, 3rd Edition. Cambridge University Press URL http://nr.com/.

Roberts, M., Andreasen, V., Lloyd, A., Pellis, L., 2015. Nine challenges for deterministic epidemic models. Epidemics 10, 49–53. https://doi.org/10.1016/j.epidem.2014.09.006. challenges in Modelling Infectious DIsease Dynamics.

Rue, H., Martino, S., Chopin, N., 2009. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. J. Royal Stat. Soc.. Series B: Stat. Methodol. https://doi.org/10.1111/j.1467-9868.2008.00700.x.