Brief introduction of particle MCMC

Fei Lu
feilu@math.jhu.edu
Started: 5/10/2018, Last Updated: 6/19/2018

Abstract: Introduction to particle MCMC algorithms [?], especially, the particle Gibbs with ancestor sampling [?].

Contents

For simplicity, we introduce the framework of PMCMC considering the inference in state-space models (SSMs), also known as hidden Markov models (HMMs). Hereby we use the convention that capital letters denote random variables and lower case letter denote the values of the random variables in corresponding capital letters. The state is a hidden Markov process $X_{1:N}$ with initial distribution $X_1 \sim p_\theta(x_1)$ and transition probability density

$$X_{n+1}|(X_n = x_n) \sim p_\theta(x_{n+1}|x_n),$$

for some static parameter $\theta \in \Theta$. Hereafter we use $p(z)$ to denote the density of the random variables Z. This process can be a times series or a dynamical system. The process is observed through another process $Y_{1:N}$ (often resulted from nonlinear measurement functions with measurement errors) , which is assumed to be conditionally independent given $X_{1:N}$ with1:N marginal probability density

$$Y_n|(X_n = x_n) = p_\theta(y_n|x_n).$$

For example, if the state-space model is of the form

$$X_n = f(\theta, X_{n-1}) + V_n, \tag{1}$$
$$Y_n = g(\theta, X_n) + W_n, \tag{2}$$

with $V_n \sim \mathcal{N}(0, \sigma_V^2)$ independent of $W_n \sim \mathcal{N}(0, \sigma_W^2)$, then

$$p_\theta(x_n|x_{n-1}) \propto e^{-\frac{|x_n - f(\theta, x_{n-1})|^2}{2\sigma_V^2}}, \quad p_\theta(y_n|x_n) \propto e^{-\frac{|y_n - g(\theta, x_n)|^2}{2\sigma_W^2}}.$$

In the evaluation of these densities, we evaluate the deterministic maps f and g.

The goal is to estimate the states and the parameter from the observations $y_{1:N}$. In Bayesian inference, we estimate the posterior of $(\theta, x_{1:N})$ conditional on $y_{1:N}$:

$$p(\theta, x_{1:N}|y_{1:N}) \propto p_\theta(x_{1:N}|y_{1:N})p(\theta),$$

where $p_\theta(x_{1:N}|y_{1:N})$ denote the density of $X_{1:N}$ conditional on $y_{1:N}$ and θ, i.e.

$$p_\theta(x_{1:N}|y_{1:N}) = p_\theta(x_1) \prod_{n=2}^{T} p_\theta(x_n|x_{n-1}) \prod_{n=1}^{T} p_\theta(y_n|x_n).$$

If the parameter is know, then the goal is to estimate $p_\theta(x_{1:N}|y_{1:N})$.

There are various SMC and MCMC methods to sample from $p(\theta, x_{1:N}|y_{1:N})$, commonly used strategies are (i) by alternatively updating the state components $x_{1:N}$ conditional on θ and θ conditional on $x_{1:N}$, or (ii) by adding artificial dynamics to θ. We focus on the sampling of $p_\theta(x_{1:N}|y_{1:N})$, since sampling from $p(\theta|y_{1:N}, x_{1:N})$ is often feasible.

1.2 Sequential Monte Carlo.

SMC methods approximate the target density $p_\theta(x_{1:N}|y_{1:N})$ sequentially by weighted random samples called particles, (hereafter in this subsection we drop the subindex θ to simplify the notation)

$$\widehat{p}(x_{1:n}|y_{1:n}) := \sum_{m=1}^{M} W_n^m \delta_{X_{1:n}^m}(dx_{1:n}).$$

These weighted samples are generated sequentially by importance sampling based on the recurrent formation

$$p(x_{1:n}|y_{1:n}) = p(x_{1:n-1}|y_{1:n-1}) \frac{p\left(y_n|x_n\right) p(x_n|x_{n-1})}{p(y_n|y_{1:n-1})}. \tag{3}$$

That is, at each time n, one first draws a sample of X_n^m for each $m = 1, \ldots, M$ from an easy to sample importance density $q(x_n|y_n, X_{n-1}^m)$ (approximating the "incremental density" which is proportional to $p\left(y_n|x_n\right) p(x_n|X_{n-1}^m)$), and computes incremental weights

$$w_n^m = \frac{p(X_n^m|X_{n-1}^m)p(y_n|X_n^m))}{q(X_n^m|y_n, X_{n-1}^m)},$$

which take account the discrepancy between the two densities. One then assigns normalized weights $\{W_n^m \propto W_{n-1}^m w_n^m\}_{m=1}^M$ to the concatenated sample trajectories $\{X_{1:n}^m\}_{m=1}^M$.

A clear drawback of the above procedure is that all but one of the weights $\{W_n^m\}$ will become close to zero as the number of iterations increases, due to the multiplication and normalization operations. To avoid this, one replaces the unevenly weighted samples $\{(X_{n-1}^m, W_{n-1}^m)\}$ by uniformly weighted samples from the approximate density $\widehat{p}_\theta(x_{n-1}|y_{1:n-1})$. This is the well-known resampling step. More precisely, this is carried out as follows:

(i) draw random indices $\{A_{n-1}^m\}_{m=1}^M$ according the discrete probability distribution $\mathbb{F}(\cdot|\mathbf{W}_{n-1})$ on the set $\{1, \ldots, M\}$, which is defined as

$$\mathbb{F}(A_{n-1} = k|\mathbf{W}_{n-1}) = W_{n-1}^k, \text{ for } k = 1, \ldots, M.$$

(ii) for each m, draw a sample X_n^m from $q(x_n|y_n, X_{n-1}^{A_{n-1}^m})$ and set $X_{1:n}^m := (X_{n-1}^{A_{n-1}^m}, X_n^m)$;

(iii) compute and normalize the weights

$$w_n^m := w_n(X_{1:n}^m) = \frac{p(X_n^m|X_{n-1}^{A_{n-1}^m})p(y_n|X_n^m))}{q(X_n^m|y_n, X_{n-1}^{A_{n-1}^m})}, \quad W_n^m = \frac{w_n^m}{\sum_{k=1}^M w_n^k}. \tag{4}$$

The above SMC sampler is summarized in Algorithm **??**.

Drawbacks of SMC. Note that while the resampling technique prevents W_T^m from being degenerate, the SMC approximation suffers the degeneracy (or depletion) problem: the marginal distribution $\widehat{p}(x_n|(y_{1:N}))$ gets to concentrated on a single particle as $N - n$ increases because each resampling step reduces the number of distinct particles of x_n. Therefore, the joint density $p(x_{1:N}|y_{1:N})$ deteriorates as time N increases.

Algorithm 1 Sequential Monte Carlo with resampling. (Each step is for $m = 1, \dots, M$.)

Output: Samples $\{(X_{1:N}^m, W_T^m)\}_{m=1}^M$

Draw samples $X_1^m \sim q(x_1|y_1)$.

Compute and normalize the weights: $w_1^m = \frac{p_\theta(X_1^m)p_\theta(y_1|X_1^m))}{q(X_1^m|y_1)}$, $W_1^m = \frac{w_1^m}{\sum_{k=1}^M w_1^k}$.

for $n = 2 : N$ **do**

Draw samples $A_{n-1}^m \sim \mathbb{F}(\cdot|\mathbf{W}_{n-1})$.

Draw samples $X_n^m \sim q(x_n|y_n, X_{n-1}^{A_{n-1}^m})$ and set $X_{1:n}^m := (X_{n-1}^{A_{n-1}^m}, X_n^m)$.

Compute the normalized weights W_n^m according to (**??**).

end for

1.3 Standard Markov Chain Monte Carlo

MCMC approximates the target density $p_\theta(x_{1:N}|y_{1:N})$ by constructing a Markov chain $(X_{1:N}(i), i \geqslant 1)$ with the target distribution as invariant distribution. The major difficulty is the design of high-dimensional proposal densities $p(x'_{1:N}|x_{1:N})$ for possible moves of the chain. Local strategies are often used, and a standard practice consists of dividing the high-dimensional distribution into small blocks and updating each of these blocks in turn. For example, we can divide $x_{1:N}$ into blocks of length K and update each block $x_{n+1:n+K}$ according to an MCMC step of invariant density

$$p(x_{n+1:n+K}|y_{1:N}, x_{1:n}, x_{n+K+1:N}) \propto \prod_{k=n}^{n+K} p(x_{k+1}|x_k) \prod_{k=n+1}^{n+K} p(y_k|x_k).$$

To design efficient proposal densities (in a Metropolis-Hastings update) for this density, the size K of the block is often limited, and this slows down the exploration of the support of $p_\theta(x_{1:N}|y_{1:N})$.

Comparison of MCMC and SMC: in the above MCMC local strategy, the distribution of the local block $p(x_{n+1:n+K}|y_{1:N}, x_{1:n}, x_{n+K+1:N})$ depends on all the observations and the other parts of the trajectory, and it can be costly to compute the proposal density; in SMC (when viewed as a local strategy), the distribution of each "local block" $p(x_n|y_{1:n}, x_{1:n-1})$ only conditional on the past, and it can be easily computed but pays the price of leading to degeneracy.

2 Particle MCMC

The framework of particle MCMC (PMCMC) introduced in [**?**] is a systematic combination of sequential Monte Carlo (SMC) and MCMC methods, exploiting the strengths of both techniques.[1]

The PMCMC samplers use SMC algorithms to design efficient high-dimensional proposal distributions for MCMC algorithms, and use Markov chain moves to guide the trajectories proposed by the SMC algorithms to explore the target distribution. Two types of methods have been proposed for the Markov move chain step, and they lead to two types of PMCMC samplers:

- The *particle marginal Metropolis-Hastings samplers* (PMMH) that use the accept-reject algorithm involving the marginal density $p(\theta|y_{1:N}) \propto p_\theta(y_{1:N})p(\theta)$, and this justifies the name

[1] Key point: When sampling high-dimensional distribution, both SMC and MCMC methods use *local strategies* (such as sequential sampling or Gibbs methods, all based on conditional sampling) to break up the original sampling problem into smaller and simpler ones. Such local strategies ignore some of the *global features* of the target distribution, resulting potentially poor performance (e.g. degeneracy of SMC, and XXX of Gibbs MCMC). How global features are captured in MCMC? How pMCMC overcome this problem?

PMMH. Here $p(\theta)$ is the prior, and the marginal likelihood $p_\theta(y_{1:N})$ is estimated by SMC algorithms.

- The *particle Gibbs sampler* (PG) that uses a novel a conditional SMC update [**?**], as well as its variant, the ***particle Gibbs with ancestor sampling*** (PGAS) sampler [**?**] which improves the conditional SMC update.

These samplers can be directly used for inference in state-space models and has been extended to various model settings. The PGAS has shown to outperform PMMH and PG (when the SMC step uses the simple sequential importance sampling with resampling algorithm, in which the importance densities are the prior densities from the state model, see e.g. [**?**, **?**]). Therefore we focus on PGAS and PG below and leave the description of the PMMH in Appendix **??**.

2.1 Particle Gibbs with ancestor sampling (PGAS)

The PGAS and PG samplers use a conditional SMC update step to realize the transition between two steps of the Markov chain, admitting the target distribution invariant and avoiding the accept-reject step in MH. The framework of these samplers is as follows:

- Initialization: set $\theta(1)$ ~~$, X_{1:N}(1)$~~ arbitrarily. Run an SMC to generate weighted samples $\{X_{1:N}^m, W_N^m\}_{m=1}^M$ for $p_{\theta(1)}(x_{1:N}|y_{1:N})$ and draw $X_{1:N}(1)$ from these weighted samples.
- Iteration: for $t \geqslant 1$:
  1. Sample $\theta(t+1) \sim p(\cdot|y_{1:N}, X_{1:N}(t))$
  2. Run a conditional SMC algorithm conditional on $X_{1:N}(t)$ (this method will be explained below): to generate weighted samples $\{X_{1:N}^m, W_N^m\}_{m=1}^M$ for the approximate target density
  $$\widehat{p}_{\theta(t+1), X_{1:N}(t)}(x_{1:N}|y_{1:N}) = \sum_{m=1}^{M} W_N^m \delta_{X_{1:N}^m}(dx_{1:N}),$$
  and to draw a sample $X_{1:N}(t+1) \sim \widehat{p}_{\theta(t+1), X_{1:N}(t)}(x_{1:N}|y_{1:N})$.

The core of PGAS and PG samplers is the conditional SMC algorithm. Intuitively, the conditional SMC algorithm is a standard SMC such that in each time step, one of the M particle is taken from the reference path, and the remaining $M-1$ particles are generated as usual, therefore the $M-1$ particles interact with the reference path through the resampling steps.

In the PG sampler, the reference path $X_{1:N}(t)$ is retained through the resampling steps. This is accomplished by deterministically setting $X_{1:N}^M = X_{1:N}(t)$ and $A_n^M = M$ for all n, and by sampling the remaining $M-1$ particles according a standard SMC algorithm. The reference path interacts with the other paths by contributing a weight w_n^M. This is the key to ensure the PG Markov chain converge to the target distribution. However, since the reference path is always retained, the resampling steps tend to direct the other particles towards the reference path therefore lead to poor mixing.

The PGAS sampler increases the mixing of the chain by allowing the reference path to be connected with other particles, i.e. assigning a history to the partial reference path $X_{n:N}(t)$. This is accomplished by sampling a random value for the index variable A_{n-1}^M, which is referred as an *ancestor sampling step*. The distribution of the index A_{n-1}^M is determined by the likelihood

of connecting $X_{n:N}(t)$ to the particles $\{X_{1:n-1}^m\}_{m=1}^M$ which leads to weights

$$\widetilde{w}_{n-1|N}^m = w_{n-1}^m \frac{p_{\theta(t+1)}(X_{1:n-1}^m, X_{n:N}(t)|y_{1:N})}{p_{\theta(t+1)}(X_{1:n-1}^m|y_{1:n-1})}, \ \widetilde{W}_{n-1|N}^m = \frac{\widetilde{w}_{n-1}^m}{\sum_{k=1}^M \widetilde{w}_{n-1}^k} \tag{5}$$

Here the expression of the weights can be seen as an application of the Bayes' theorem, where the importance weight w_{n-1}^m is the prior probability of the particle $X_{1:n-1}^m$, and the ratio between the target densities is the likelihood of that $X_{n:N}(t)$ is originated from $X_{1:n-1}^m$ (see Remark **??** about its computation). In short, A_{n-1}^M will be drawn from $\mathbb{F}(A_{n-1}^M = k|\widetilde{\mathbf{W}}_{n-1|N}) = \widetilde{W}_{n-1|N}^k$.

The above procedure for conditional SMC with ancestor sampling of PGAS is summarized in Algorithm **??**. Removing line **??**, the ancestor sampling step, we get the PG sampler. While the PG sampler retains the reference path, the PGAS sampler tends to break the reference path into pieces, due to allowing the past of the reference path to be resampled.

Algorithm 2 Conditional SMC in PGAS.

Input: $X_{1:N}(t)$ and $\theta := \theta(t+1)$.
Output: $X_{(1:N)}(t+1)$.

Initialize the particles in SMC:

1: Set $X_1^M = X_1(t)$ and draw samples $\{X_1^m\}_{m=1}^{M-1} \sim q_\theta(x_1|y_1)$.
2: Compute the weights $w_1^m = \frac{p_\theta(X_1^m)p_\theta(y_1|X_1^m))}{q_\theta(X_1^m|y_1)}$, $W_1^m = \frac{w_1^m}{\sum_{k=1}^M w_1^k}$ for $m = 1:M$.
3: **for** $n = 2:N$ **do**
4: Draw samples $\{A_{n-1}^m\}_{m=1}^{M-1} \sim \mathbb{F}(\cdot|\mathbf{W}_{n-1})$.
5: Set $X_n^M = X_n(t)$ and draw samples $X_n^m \sim q(x_n|y_n, X_{n-1}^{A_{n-1}^m})$ for $m = 1:M-1$.
6: Draw $A_{n-1}^M \sim \mathbb{F}(\cdot|\widetilde{\mathbf{W}}_{n-1|N})$, where the weights in $\widetilde{\mathbf{W}}_{n-1|N}$ are computed in (**??**).
7: Set $X_{1:n}^m := (X_{n-1}^{A_{n-1}^m}, X_n^m)$ for $m = 1:M$.
8: Compute the normalized weights W_n^m according to (**??**).
9: **end for**
10: Draw A_N with $\mathbb{F}(\cdot|\mathbf{W}_N)$.
11: **return** $X_{(1:N)}(t+1) = X_{1:N}^{A_N}$.

Remark 1 *When the observation model is $Y_n = f(X_n) + W_n$, the ratio $\frac{p_\theta(X_{1:n-1}^m, X_{n:N}(t)|y_{1:N})}{p_\theta(X_{1:n-1}^m|y_{1:n-1})}$ in the ancestor sampling is straightforward to evaluate as follows. Note that:*

$$\begin{aligned}\frac{p(x_{1:N}|y_{1:N})}{p(x_{1:n-1}|y_{1:n-1})} &= \frac{p(x_{1:N}, y_{1:N})}{p(x_{1:n-1}, y_{1:n-1})p(y_{1:N}|y_{1:n-1})} \\ \propto \frac{p(y_{1:N}|x_{1:N})p(x_{1:N})}{p(y_{1:n-1}|x_{1:n-1})p(x_{1:n-1})} &= p(y_{n:N}|x_{n:N})p(x_{n:N}|x_{1:n-1}),\end{aligned}$$

where the conditional independence between observations has been used, i.e. $p(y_{1:k}|x_{1:k}) = \prod_{i=1}^k p(y_i|x_i)$. Note further that $p(y_{n:N}|X_{n:N}(t))$ does not depend on m, therefore,

$$\frac{p_\theta(X_{1:n-1}^m, X_{n:N}(t)|y_{1:N})}{p_\theta(X_{1:n-1}^m|y_{1:n-1})} \propto p(X_{n:N}(t)|X_{1:n-1}^m).$$

If the state model is Markov, then the ratio equals to $p_{\theta(t+1)}(X_n(t)|X_{n-1}^m)$; if the state model is non-Markov, then the ratio equals to $p_{\theta(t+1)}(X_{n:N}(t)|X_{1:n-1}^m)$.

Similar computation goes to the case when the observation depends on multiple time steps of the states, e.g.

$$Y_{n_i} = f(x_{n_{i-1}+1:n_i}) + W_{n_i}, \quad n_i - n_{i-1} > 1, i = 1, \cdots, L,$$

and the major difference is how one updates the particles in SMC. [TO Nils: I think we should use the above function for observation, which can be viewed as a right-point approximation of the integral: $\int_{x_{n_{i-1}}}^{x_{n_i}} g(x)dx \approx \sum_{j=n_{i-1}+1}^{n_i} g(x_j)\Delta x$. We can also use left-point approximation. The point is that y_{n_i} depends on either $x_{n_{i-1}}$ or x_{n_i}, but not both. Otherwise, we may have more complicated SMC updates.]

In this case, upon observing y_{n_i} at time n_i, the SMC draws samples $X^m_{n_{i-1}+1:n_i}$ (the states that y_{n_i} depends on) from an importance density $q(x_{n_{i-1}+1:n_i}|y_{n_i}, x_{1:n_{i-1}})$ and assigns weights

$$w^m_{n_i} := w_{n_i}(X^m_{1:n_i}) = \frac{p(X^m_{n_{i-1}+1:n_i}|X^{A^m_{n_{i-1}}}_{1:n_{i-1}})p(y_{n_i}|X^m_{n_{i-1}+1:n_i}))}{q(X^m_{n_{i-1}+1:n_i}|y_{n_i}, X^{A^m_{n_{i-1}}}_{n_{i-1}})}, \quad W^m_{n_i} = \frac{w^m_{n_i}}{\sum_{k=1}^M w^k_{n_i}}. \tag{6}$$

In the ancestor sampling step, the weights that provides a distribution of the ancestor are

$$\widetilde{w}^m_{n_{i-1}|N} = w^m_{n_{i-1}} \frac{p_{\theta(t+1)}(X^m_{1:n_{i-1}}, X_{n_{i-1}+1:N}(t)|y_{n_{1:L}})}{p_{\theta(t+1)}(X^m_{1:n_{i-1}}|y_{n_{i-1}})}, \widetilde{W}^m_{n_{i-1}|N} = \frac{\widetilde{w}^m_{n_{i-1}}}{\sum_{k=1}^M \widetilde{w}^k_{n_{i-1}}}. \tag{7}$$

The algorithm is summarized in Algorithm **??**. *As before, the ratio between the target distributions is the likelihood of connecting the partial reference path $X_{n_{i-1}+1:N}(t)$ with the samples $X^m_{1:n_{i-1}}$, and its computation becomes*

$$\begin{aligned}
&\frac{p(x_{1:n_{i-1}}, x_{n_{i-1}+1:N}|y_{n_{1:L}})}{p(x_{1:n_{i-1}}|y_{n_{1:i-1}})} = \frac{p(x_{1:n_{i-1}}, x_{n_{i-1}+1:N}, y_{n_{1:L}})}{p(x_{1:n_{i-1}}, y_{n_{1:i-1}})p(y_{n_{1:L}}|y_{n_{1:i-1}})} \\
\propto &\frac{p(y_{n_{1:L}}|x_{1:n_{i-1}}, x_{n_{i-1}+1:N})p(x_{n_{i-1}+1:N})}{p(y_{n_{1:i-1}}|x_{1:n_{i-1}})p(x_{1:n_{i-1}})} = p(y_{n_{i:L}}|x_{n_{i-1}+1:N})p(x_{n_{i-1}+1:N}|x_{1:n_{i-1}}),
\end{aligned}$$

where the conditional independence between observations has been used, i.e. $p(y_{n_{j:k}}|x_{n_j+1:n_k}) = \prod_{i=j}^{k-1} p(y_{n_i}|x_{n_i+1:n_{i+1}})$. Noticing again that that $p(y_{n_{i:L}}|X_{n_{i-1}+1:N})$ does not depend on m, we obtain

$$\frac{p_{\theta(t+1)}(X^m_{1:n_{i-1}}, X_{n_{i-1}+1:N}(t)|y_{n_{1:L}})}{p_{\theta(t+1)}(X^m_{1:n_{i-1}}|y_{n_{i-1}})} \propto p(X_{n_{i-1}+1:N}(t)|X^m_{1:n_{i-1}}).$$

If the state model is Markov, then the ratio equals to $p_{\theta(t+1)}(X_{n_{i-1}+1}(t)|X^m_{n_{i-1}})$; if the state model is non-Markov, then the ratio equals to $p_{\theta(t+1)}(X_{n_{i-1}+1:N}(t)|X^m_{1:n_{i-1}})$.

Remark 2 (About splitting the state model to deterministic + random parts) *. In computation, one can run 1-step the deterministic part of the state model (i.e. f in $X_n = f(\theta, X_{n-1}) + V_n$), and use it in both the sample drawing (by adding the random forces to it) and the evaluation of $p_{\theta(t+1)}(X_{n_{i-1}+1}(t)|X^m_{n_{i-1}})$ in the ancestor sampling. However, for observations with time gap, we should note that samples of a block of states $X^m_{n_{i-1}+1:n_i}$ have to be generated by iterating the stochastic state model, and NOT by iterating the deterministic map f and add multi-step noise.*

Remark 3 (Avoiding singular weights in computation) *In practice, the weights $\{w^m_n\}_{m=1}^M$ and the ratios (or increment weights in other types of SMC that do not resample every step)*

$\{\alpha_n^m\}_{m=1}^M$ in (??) and (??) can be singular in the sense that $\sum_{m=1}^M w_n^m \alpha_n^m = 0$ due to numerical precision. This would happen very often if any probability function is highly concentrated and the normalization step only makes the situation worse. One way to avoid such a problem is to store the logarithm exponents and minus the maximum of these exponents before normalization. That is, we store c_n^m, a_n^m's of $w_n^m = \frac{e^{c_n^m}}{\sum_{k=1}^M e^{c_n^k}}$ and $\alpha_n^m = \frac{e^{a_n^m}}{\sum_{k=1}^M e^{a_n^k}}$, and compute $\widetilde{w}_n^m \propto e^{c_n^m + a_n^m - \max_k\{c_n^k + a_n^k\}}$.

Algorithm 3 Conditional SMC in PGAS with sparse observations $Y_{n_i} = f(x_{n_{i-1}+1:n_i}) + W_{n_i}$.

Input: $X_{1:N}(t)$ and $\theta := \theta(t+1)$.
Output: $X_{(1:N)}(t+1)$.

Initialize the particles in SMC:

1: Set $X_{1:n_1}^M = X_{1:n_1}(t)$ and draw samples $\{X_{1:n_1}^m\}_{m=1}^{M-1} \sim q_\theta(x_{1:n_1}|y_{n_1})$.
2: Compute the weights $w_{n_1}^m = \frac{p_\theta(X_{1:n_1}^m) p_\theta(y_{n_1}|X_{1:n_1}^m))}{q_\theta(X_{1:n_1}^m|y_{n_1})}$, $W_{n_1}^m = \frac{w_{n_1}^m}{\sum_{k=1}^M w_{n_1}^k}$ for $m = 1:M$.
3: **for** $i = 2:L$ **do**
4: Draw samples $\{A_{n_{i-1}}^m\}_{m=1}^{M-1} \sim \mathbb{F}(\cdot|\mathbf{W}_{n_{i-1}})$.
5: Set $X_{n_{i-1}+1:n_i}^M = X_{n_{i-1}+1:n_i}(t)$ and draw samples $X_{n_{i-1}+1:n_i}^m \sim q(x_{n_{i-1}+1:n_i}|y_{n_i}, X_{n_{i-1}}^{A_{n_{i-1}}^m})$ for $m = 1:M-1$.
6: Draw $A_{n_{i-1}}^M \sim \mathbb{F}(\cdot|\widetilde{\mathbf{W}}_{n_{i-1}|N})$, where the weights in $\widetilde{\mathbf{W}}_{n-1|N}$ are computed in (??).
7: Set $X_{1:n_i}^m := (X_{1:n_{i-1}}^{A_{n_{i-1}}^m}, X_{n_{i-1}+1:n_i}^m)$ for $m = 1:M$.
8: Compute the normalized weights $W_{n_i}^m$ according to (??).
9: **end for**
10: Draw A_N with $\mathbb{F}(\cdot|\mathbf{W}_N)$.
11: **return** $X_{(1:N)}(t+1) = X_{1:N}^{A_N}$.

3 Appendix

3.1 MCMC: the Metropolis-Hastings Algorithm

The most widely used MCMC algorithm is the Metropolis-Hastings algorithm [**?**], and we refer to [**?**] for an overview of MCMC methods. It prescribes a transition rule for the a Markov chain, by using a proposal function (a probability transition function) to suggest possible moves and by an acceptance-rejection rule to ensure the invariant distribution is the target distribution π. Suppose that the chain starts with an initial state $x^{(0)}$, the Metropolis algorithm iterates the following two steps:

- Perturb the original state to a new state x'. That is, generate a new state x' from a probability transition function $p(x(t), x')$.
- Accept the state with rate $r(x(t), x')$, where

$$r(x, x') := \min\left\{1, \frac{\pi(x')p(x', x)}{\pi(x)p(x, x')}\right\}.$$

  That is, generate a random number $U \sim \text{Uniform}[0, 1]$, and let $x(t+1) = x'$ if $U \leqslant r(x(t), x')$ and $x(t+1) = x(t)$ otherwise.

Note that the probability transition function p does not have to be symmetric, and the only requirement is that $p(x, y) > 0$ if and only if $p(y, x) > 0$.

Algorithm 4 Metropolis-Hastings algorithm.

Output: A Markov chain $x(1:T)$ (with the target density $\pi(x)$ as invariant density).

Initialize the chain $x(1)$.

for $t = 2 : T$ **do**

Draw a sample from the proposal distribution $x' \sim p(x(t-1), x')$.

Draw $U \sim \text{Uniform}[0, 1]$ and let

$$x(t) = \begin{cases} x', & \text{if } U \leqslant r(x(t-1), x'); \\ x(t-1), & \text{otherwise.} \end{cases}$$

where the acceptance ratio is $r(x, x') := \min\left\{1, \frac{\pi(x')p(x', x)}{\pi(x)p(x, x')}\right\}$.

end for

3.2 PMMH: particle marginal Metropolis-Hastings samplers

To sample $p(\theta, x_{1:N}|y_{1:N}) = p(\theta|y_{1:N})p_\theta(x_{1:N}|y_{1:N})$, the PMMH sampler jointly update θ and $x_{1:N}$ by the proposal density

$$q(\theta^*, x^*_{1:N}|\theta, x_{1:N}) = q(\theta^*|\theta)p_{\theta^*}(x^*_{1:N}|y_{1:N}).$$

That is, one first draw a sample θ^* from $q(\theta^*|\theta)$, then draw a sample $x^*_{1:N}$ from the approximate density $\widehat{p}_{\theta^*}(x_{1:N}|y_{1:N})$ generated by SMC. The MH acceptance ratio is

$$\frac{p(\theta^*, x^*_{1:N}|y_{1:N})q(\theta, x_{1:N}|\theta^*, x^*_{1:N})}{p(\theta, x_{1:N}|y_{1:N})q(\theta^*, x^*_{1:N}|\theta, x_{1:N})} = \frac{p_{\theta^*}(y_{1:N})p(\theta^*)q(\theta|\theta^*)}{p_\theta(y_{1:N})p(\theta)q(\theta^*|\theta)}.$$

Note that this ratio effectively uses the marginal density $p(\theta|y_{1:N}) \propto p_\theta(y_{1:N})p(\theta)$, and this justifies the name PMMH. The likelihood $p_{\theta*}(y_{1:N})$ is approximated by XXXX

The PMMH sampler is as follows.

Algorithm 5 particle marginal Metropolis-Hastings sampler (PMMH).

1: Initialize the chain:
2: Draw a sample $\theta(1) \sim p(\theta)$;
3: Run an SMC algorithm to generate $\widehat{p}_{\theta(1)}(x_{1:N}|y_{1:N})$. Draw a sample $X_{1:N}(1) \sim \widehat{p}_{\theta(1)}(\cdot|y_{1:N})$ and compute $\widehat{p}_{\theta(1)}(y_{1:N})$.
4: **for** $t = 2 : T$ **do**
5: Sample $\theta^* \sim q(\cdot|\theta(t-1))$;
6: Run an SMC algorithm to generate $\widehat{p}_{\theta*}(x_{1:N}|y_{1:N})$. Draw a sample $X^*_{1:N} \sim \widehat{p}_{\theta*}(\cdot|y_{1:N})$ and compute $\widehat{p}_{\theta*}(y_{1:N})$.
7: Accept $(\theta^*, X^*_{1:N})$ with probability

$$r := \frac{\widehat{p}_{\theta*}(y_{1:N})p(\theta^*)q(\theta(t-1)|\theta^*)}{\widehat{p}_{\theta(t-1)}(y_{1:N})p(\theta(t-1))q(\theta^*|\theta(t-1))}$$

That is, draw $U \sim \text{Uniform}[0,1]$ and let $(\theta(t), X_{1:N}(t)) = \begin{cases} (\theta^*, X^*_{1:N}), & \text{if } U \leqslant r; \\ (\theta(t-1), X_{1:N}(t-1)), & \text{otherwise.} \end{cases}$

8: **end for**

3.3 Convergence of PG and PGAS