Particle Markov Chain Monte Carlo Methods

Bjarke Hautop

Student ID: 202005674

Master's Thesis in Statistics

Aarhus University
Supervisor: Jan Pedersen
February 6, 2025

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Introduction

2 Monte Carlo Methods

In this chapter, we provide the theoretical foundation for Particle Markov Chain Monte Carlo (PMCMC). We follow the work of Andrieu et al. [2010], Doucet and Johansen [2009], and Kroese et al. [2013].

Suppose we have a density $\pi_n(x_{1:n})$ for some n, where the normalizing constant is possibly unknown, that is

$$\pi_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n}, \tag{2.1}$$

where $\gamma_n(x_{1:n})$ is the unnormalized density and Z_n is a normalizing constant

$$Z_n = \int \gamma_n(x_{1:n}) dx_{1:n}. \tag{2.2}$$

Let $X_{1:n} \sim \pi_n(x_{1:n})$ and suppose we generate N i.i.d. samples $x_{1:n}^{(1)}, x_{1:n}^{(2)}, \ldots, x_{1:n}^{(N)}$. We can then approximate $\pi_n(x_{1:n})$ by the empirical measure

$$\pi_n^{\mathrm{MC}}(x_{1:n}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_{1:n}^{(i)}}(x_{1:n}),$$

and any marginal $\pi_n(x_k)$ as

$$\pi_n^{\mathrm{MC}}(x_k) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X_k^{(i)}}(x_k).$$

The expectation of any function $H_n : \mathcal{X}^n \to \mathbb{R}$ is given by

$$I_n(H_n) \coloneqq \mathbb{E}_{X_{1:n} \sim \pi_n}[H_n(X_{1:n})] = \int H_n(x_{1:n}) \pi_n(x_{1:n})\, dx_{1:n},$$

and we can estimate it by

$$I_n^{\mathrm{MC}}(H_n) \coloneqq \int H_n(x_{1:n}) \pi_n^{\mathrm{MC}}(x_{1:n}) dx_{1:n} = \frac{1}{N} \sum_{i=1}^{N} H_n(x_{1:n}^{(i)}).$$

However, this requires that we can sample from $\pi_n(x_{1:n})$, which often is not the case when it is a complex high-dimensional distribution.

2.1 Importance Sampling

A way to solve this issue is to use importance sampling (IS). Here we introduce an importance density $q_n(x_{1:n})$ which we can sample from and such that

$$\pi_n(x_{1:n}) > 0 \implies q_n(x_{1:n}) > 0.$$

For the remainder of this chapter, we let $X_{1:n} \sim q_n(x_{1:n})$. Suppose we generate N i.i.d. samples $x_{1:n}^{(1)}, x_{1:n}^{(2)}, \ldots, x_{1:n}^{(N)}$. To correct for the fact that we sample from q_n we define the *unnormalized weight* function

$$w_n(x_{1:n}) \coloneqq \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})}$$

and define the *normalized weight* function

$$W_n^{(i)} \coloneqq \frac{w_n(X_{1:n}^{(i)})}{\sum_{j=1}^N w_n(X_{i:n}^{(j)})}.$$

From (2.1) and (2.2) we get

$$\pi_n(x_{1:n}) = \frac{w_n(x_{1:n}) q_n(x_{1:n})}{Z_n}, \tag{2.3}$$

and

$$Z_n = \int w_n(x_{1:n}) q_n(x_{1:n})\, dx_{1:n}. \tag{2.4}$$

We then define the IS estimators of respectively $\pi_n(x_{1:n})$ and Z_n as

$$\widehat{\pi}_n(x_{1:n}) = \sum_{i=1}^N W_n^{(i)} \delta_{X_{1:n}^{(i)}}(x_{1:n}), \tag{2.5}$$

$$\widehat{Z}_n = \frac{1}{N} \sum_{i=1}^N w_n(X_{1:n}^{(i)}). \tag{2.6}$$

Next, we will show what the relative variance of $\widehat{Z}_n$ is.

Theorem 2.1 (Relative variance of $\mathbb{V}(\widehat{Z}_n)$)**.** The relative variance of the IS estimate of the normalizing constant Z_n is given by

$$\frac{\mathbb{V}(\widehat{Z}_n)}{Z_n^2} = \frac{1}{N}\Big(\int \frac{\pi_n^2(x_{1:n})}{q_n(x_{1:n})}\, dx_{1:n} - 1 \Big).$$

Proof. The variance of $\widehat{Z}_n$ is

$$\begin{aligned}
\mathbb{V}(\widehat{Z}_n) &= \frac{1}{N} \mathbb{V}\big(w_n(X_{1:n})\big) \\
&= \frac{1}{N} \mathbb{E}[w_n^2(X_{1:n})] - (\mathbb{E}[w_n(X_{1:n})])^2 \\
&= \frac{1}{N}\Big(\int \frac{\gamma_n^2(x_{1:n})}{q_n^2(x_{1:n})} q_n(x_{1:n})\, dx_{1:n} - Z_n^2 \Big) \\
&= \frac{1}{N}\Big(\int \frac{\gamma_n^2(x_{1:n})}{q_n(x_{1:n})}\, dx_{1:n} - Z_n^2 \Big) \\
&= \frac{1}{N}\Big(\int \frac{Z_n^2 \pi_n^2(x_{1:n})}{q_n(x_{1:n})}\, dx_{1:n} - Z_n^2 \Big) \\
&= \frac{Z_n^2}{N}\Big(\int \frac{\pi_n^2(x_{1:n})}{q_n(x_{1:n})}\, dx_{1:n} - 1 \Big).
\end{aligned}$$

Dividing by Z_n^2 gives the result. □

Furthermore, we can also estimate $I_n(H_n)$ by

$$I_n^{\mathrm{IS}}(H_n) \coloneqq \int H_n(x_{1:n})\widehat{\pi}(x_{1:n})\,dx_{1:n} = \sum_{i=1}^{N} W_n^{(i)} H_n(X_{1:n}^{(i)}) = \frac{\frac{1}{N}\sum_{i=1}^{N} w_n(X_{1:n}^{(i)})H_n(X_{1:n}^{(i)})}{\frac{1}{N}\sum_{i=1}^{N} w_n(X_{1:n}^{(i)})}.$$

Note, that the numerator is an unbiased estimate of $Z_n I_n$ and the denominator an unbiased estimate of Z_n. Thus, we have a ratio of unbiased estimates, which is not unbiased. However, it is still consistent, which follows by using the law of large numbers and properties of a.s. convergence.

A natural choice for an importance density $q_n(x_{1:n})$ is one that minimizes the variance of $\widehat{Z}_n$. As shown in Theorem 2.2, this minimum variance is achieved when

$$q_n(x_{1:n}) = \pi_n(x_{1:n}).$$

However, we cannot select this, as this was the reason we used IS in the first place. Nonetheless, this result indicates that the importance density should closely resemble the target density.

We could now sample from $\pi_n(x_{1:n})$ using the above method. However, to generate a sequence of samples for each n, we would have that each step would grow linearly in n, as generating samples from $\pi_{n+1}(x_{1:n+1})$ depends on the previous samples up to time n. This makes such an algorithm unfeasible in practice.

Theorem 2.2 (Minimum Variance of IS)**.** The variance $\mathbb{V}[\widehat{Z}_n]$ is minimized if

$$q_n(x_{1:n}) = \pi_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n}.$$

The proof is done by using Lagrange multiplier, and is inspired by Shimao [2018].

Proof. By independence we have,

$$\mathbb{V}[\widehat{Z}_n] = \frac{1}{N}\,\mathbb{V}\left[w_n(X_{1:n})\right].$$

We then have

$$\mathbb{V}[w_n(X_{1:n})] = \mathbb{E}\left[\frac{\gamma_n(X_{1:n})^2}{q_n(X_{1:n})^2}\right] - Z_n^2 = \int \frac{\gamma_n(x_{1:n})^2}{q_n(x_{1:n})}\,dx_{1:n} - Z_n^2.$$

Minimizing $\mathbb{V}[\widehat{Z}_n]$ is thus equivalent to minimizing

$$J(q_n) = \int \frac{\gamma_n(x_{1:n})^2}{q_n(x_{1:n})}\,dx_{1:n},$$

subject to the constraint

$$\int q_n(x_{1:n})\,dx_{1:n} = 1.$$

We now introduce a Lagrange multiplier λ and form the Lagrangian

$$L(q_n, \lambda) = \int \frac{\gamma_n(x_{1:n})^2}{q_n(x_{1:n})}\,dx_{1:n} + \lambda\left(\int q_n(x_{1:n})\,dx_{1:n} - 1\right).$$

Taking the functional derivative with respect to $q_n(x_{1:n})$ and using chain rule we have

$$\frac{\delta L}{\delta q_n(x_{1:n})} = -\frac{\gamma_n(x_{1:n})^2}{q_n(x_{1:n})^2} + \lambda = 0.$$

Solving for $q_n(x_{1:n})$ yields

$$q_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{\sqrt{\lambda}}.$$

Enforcing the normalization condition we get

$$\int \frac{\gamma_n(x_{1:n})}{\sqrt{\lambda}}\, dx_{1:n} = 1 \quad \Longrightarrow \quad \frac{Z_n}{\sqrt{\lambda}} = 1,$$

so that $\sqrt{\lambda} = Z_n$. Therefore,

$$q_n(x_{1:n}) = \frac{\gamma_n(x_{1:n})}{Z_n} = \pi_n(x_{1:n}).$$

This completes the proof. □

2.2 Sequential Importance Sampling

Sequential Importance Sampling (SIS) builds upon the basic idea of IS by exploiting a Markov structure of the importance distribution. Instead of sampling the full trajectory at once, SIS extends the particle trajectories sequentially, updating the importance weights recursively. This leads to significant computational savings.

Formally, we let our importance distribution have a Markov structure, that is

$$q_n(x_{1:n}) = q_1(x_1)q_2(x_2|x_1)\dots q_n(x_k|x_{1:k-1}).$$

Each element in the set $\{x_{1:t}\}$ we will refer to as a particle. We define for $n \geq 2$ the *incremental importance weight* as

$$\alpha_n(x_{1:n}) \coloneqq \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})}.$$

The unnormalized weights can then be written recursively as

$$\begin{aligned} w_n(x_{1:n}) &= \frac{\gamma_n(x_{1:n})}{q_n(x_{1:n})} \\ &= \frac{\gamma_{n-1}(x_{1:n-1})}{q_{n-1}(x_{1:n-1})} \frac{\gamma_n(x_{1:n})}{\gamma_{n-1}(x_{1:n-1})q_n(x_n|x_{1:n-1})} \\ &= w_{n-1}(x_{1:n-1}) \cdot \alpha_n(x_{1:n}) \\ &= w_1(x_1) \prod_{k=2}^{n} \alpha_k(x_{1:k}). \end{aligned} \tag{2.7}$$

Summarizing, the SIS method is described in Algorithm 2.3.

Algorithm 2.3 Sequential Importance Sampling (SIS)

1: Generate particles $x_1^{(i)}$ from $q_1(x_1)$ for $i = 1, \dots, N$
2: **for** each time step $n = 2, \dots, T$ **do**
3: **for** each particle $i = 1, \dots, N$ **do**
4: Generate particles $x_n^{(i)}$ from $q_n(x_n | x_{1:n-1}^{(i)})$
5: Compute the incremental importance weight:

$$\alpha_n^{(i)} = \frac{\gamma_n(x_{1:n}^{(i)})}{\gamma_{n-1}(x_{1:n-1}^{(i)}) q_n(x_n^{(i)} | x_{1:n-1}^{(i)})}$$

6: Update the particle weight: $w_n^{(i)} = w_{n-1}^{(i)} \cdot \alpha_n^{(i)}$
7: **end for**
8: Normalize the weights: $W_n^{(i)} \leftarrow \frac{w_n^{(i)}}{\sum_{i=1}^N w_n^{(i)}}$
9: **end for**

However, this method has a severe drawback, that the relative estimated variance $\mathbb{V}(\widehat{Z}_n)/Z_n^2$ increases exponentially in n even in simple examples. Example 2.4 illustrates this issue.

Example 2.4. Consider the case where $\mathcal{X} = \mathbb{R}$ and let the density $\pi_n(x_{1:n})$ be given by

$$\pi_n(x_{1:n}) = \prod_{k=1}^n \pi_n(x_k) = \prod_{k=1}^n N(x_k; 0, 1).$$

Thus, the unnormalized density $\gamma_n(x_{1:n})$ is given by

$$\gamma_n = \prod_{k=1}^n \exp\Big(-\frac{x_k^2}{2}\Big),$$

and the normalizing constant Z_n is

$$Z_n = (2\pi)^{n/2}.$$

Ignoring that we could easily sample from π_n, we select the importance distribution q_n to sample from as

$$q_n(x_{1:n}) = \prod_{k=1}^n \pi_n(x_k) = \prod_{k=1}^n N(x_k; 0, \sigma^2)$$

and we are interested in estimating Z_n. Recall from Theorem 2.1 that the relative variance of $\widehat{Z}_n$ is given by

$$\frac{\mathbb{V}(\widehat{Z}_n)}{Z_n^2} = \frac{1}{N}\Big(\int \frac{\pi_n^2(x_{1:n})}{q_n(x_{1:n})}\, dx_{1:n} - 1\Big).$$

In our case this factors over the coordinates and we have

$$\begin{aligned}
\int \frac{\pi_n(x_{1:n})^2}{q_n(x_{1:n})}\, dx_{1:n} &= \prod_{k=1}^n \int \frac{1/(2\pi)\exp(-x^2)}{1/\sqrt{2\pi\sigma^2}\exp(-x^2/(2\sigma^2))}\, dx_{1:k} \\
&= \left[\frac{\sqrt{2\pi\sigma^2}}{2\pi}\int \exp\Big(-x^2 + \frac{x^2}{2\sigma^2}\Big)\, dx\right]^n \\
&= \left[\frac{\sqrt{2\pi\sigma^2}}{2\pi}\int \exp\Big(-\Big(1 - \frac{1}{\sigma^2}\Big)x^2\Big)\, dx\right]^n.
\end{aligned}$$

The integral is finite if and only if $1 - 1/(2\sigma^2) > 0$, that is $\sigma^2 > 1/2$. In this case, it is equal to

$$\int_{-\infty}^{\infty} \exp\Big(-\Big(1 - \frac{1}{\sigma^2}\Big)x^2\Big)\, dx = \sqrt{\frac{\pi}{1 - 1/(2\sigma^2)}}.$$

Thus, for $\sigma^2 > 1/2$ we have

$$\begin{aligned}\int \frac{\pi_n(x_{1:n})^2}{q_n(x_{1:n})}\, dx_{1:n} &= \left(\frac{\sqrt{2\pi\sigma^2}}{2\pi}\sqrt{\frac{\pi}{1 - 1/(2\sigma^2)}}\right)^n \\ &= \left(\sqrt{\frac{\sigma^4}{2\sigma^2 - 1}}\right)^n \\ &= \Big(\frac{\sigma^4}{2\sigma^2 - 1}\Big)^{n/2}.\end{aligned}$$

Finally, we can conclude that for $\sigma^2 > 1/2$ the relative variance becomes $\mathbb{V}[\widehat{Z}_n] < \infty$ and

$$\frac{\mathbb{V}[\widehat{Z}_n]}{Z_n^2} = \frac{1}{N}\left[\Big(\frac{\sigma^4}{2\sigma^2 - 1}\Big)^{n/2} - 1\right].$$

Note that for any $1/2 < \sigma^2 \neq 1$ we have that $\sigma^4/(2\sigma^2 - 1) > 1$, and thus the variance increases exponentially with n.

For example, choosing $\sigma^2 = 1.2$ then we have a reasonably good importance distribution as $q_k(x_k) \approx \pi_n(x_k)$. However,

$$N\,\mathbb{V}[\widehat{Z}_n]/Z_n^2 \approx (1.103)^{n/2},$$

which for $n = 1000$ is roughly equal to $1.9 \cdot 10^{21}$. So we would need to use $N \approx 2 \cdot 10^{23}$ particles to obtain a relative variance of 0.01.

2.3 Resampling

Over time, many particles receive negligible weight, leading to a situation where only a few particles dominate the estimate. This phenomenon, known as weight degeneracy, can degrade the performance of the estimator. The idea for resampling is to get rid of particles with low weights with a high probability, so the focus is spent on high-probability regions instead of carrying forward particles with very low weights. This typically reduces the variance, see for instance Example 2.7.

Formally, the IS approximation $\widehat{\pi}_n(x_{1:n})$ of the target distribution $\pi_n(x_{1:n})$ is based on weighted samples drawn from $q_n(x_{1:n})$. Consequently, these samples are not distributed according to $\pi_n(x_{1:n})$. To obtain samples from $\pi_n(x_{1:n})$, we can resample $X_{1:n}^{(i)}$ with probability $W_n^{(i)}$; that is, we resample from our approximation $\widehat{\pi}_n(x_{1:n})$ to approximately obtain samples from $\pi_n(x_{1:n})$. In practice, to obtain N samples, we can select $N_n^{(i)}$ offspring for each particle $X_{1:n}^{(i)}$ such that

$$N_n^{(1:n)} = (N_n^{(1)}, \ldots, N_n^{(N)}) \sim \text{Multinomial}(N, W_n^{(1:N)}).$$

Using the recursive structure of the unnormalized weights from (2.7) a natural way to estimate Z_n is to define

$$\widetilde{Z}_1 \coloneqq \frac{1}{N}\sum_{i=1}^{N} w_1(x_1^i),$$

and for $n \geq 2$ estimate Z_n recursively by

$$\widetilde{Z}_n \coloneqq \widetilde{Z}_{n-1}\alpha_n^{\mathrm{MC}}, \tag{2.8}$$

where α_n^{MC} is the standard MC estimate of α_n. We can still estimate $\pi_n(x_{1:n})$ by (2.5).

A generic SISR algorithm is given in Algorithm 2.5.

Algorithm 2.5 Sequential Importance Sampling with Resampling (SISR)

1: Generate particles $x_1^{(i)}$ from $q_1(x_1)$ for $i = 1, \ldots, N$
2: **for** each time step $n = 2, \ldots, T$ **do**
3: **for** each particle $i = 1, \ldots, N$ **do**
4: Generate particles $x_n^{(i)}$ from $q_n(x_n|x_{1:n-1}^{(i)})$
5: Compute the incremental importance weight:

$$\alpha_n^{(i)} = \frac{\gamma_n(x_{1:n}^{(i)})}{\gamma_{n-1}(x_{1:n-1}^{(i)})q_n(x_n^{(i)}|x_{1:n-1}^{(i)})}$$

6: Update the particle weight: $w_n^{(i)} = w_{n-1}^{(i)} \cdot \alpha_n^{(i)}$
7: **end for**
8: Normalize the weights: $W_n^{(i)} \leftarrow \frac{w_n^{(i)}}{\sum_{i=1}^N w_n^{(i)}}$
9: Draw N indices $\{a^{(i)}\}_{i=1}^N$ from $\{1, \ldots, N\}$ according to the probabilities $\{W_n^{(i)}\}$.
10: Set $X_{1:n}^{(i)} \leftarrow X_{1:n}^{(a^{(i)})}$ for all i.
11: Reset weights: $w_n^{(i)} = 1$.
12: **end for**

Now we provide a result similar to Theorem 2.1 for the case with resampling.

Theorem 2.6 (Relative Asymptotic Variance with Resampling)**.** The relative asymptotic variance of the IS estimate of the normalizing constant Z_n with resampling at every time step is

$$\frac{\mathbb{V}(\widetilde{Z}_n)}{Z_n^2} = \frac{1}{N}\left[\left(\int \frac{\pi_1^2(x_1)}{q_1(x_1)}\,dx_1 - 1\right) + \sum_{k=2}^{n}\left(\int \frac{\pi_k^2(x_{1:k})}{\pi_{k-1}(x_{1:k-1})\,q_k(x_k|x_{1:k-1})}\,dx_{k-1:k} - 1\right)\right].$$

A formal proof is omitted for brevity (it can be derived using the Feynman–Kac framework; see, e.g., Moral [2004]).

Notably, comparing Theorem 2.6 with Theorem 2.1 reveals that while resampling introduces additional variance, the resampling has a resetting property of the systems. Thus, we get that the associated errors accumulate linearly rather than multiplicatively. This linear propagation can offer enhanced stability, especially in high-dimensional settings where multiplicative error growth might otherwise lead to significant degradation in performance. We continue Example 2.4 using Theorem 2.6 to highlight this.

Example 2.7 (Example 2.4 continued)**.** Using Theorem 2.6 and using derivation done in Example 2.4 we have that it is finite for $\sigma^2 > 1/2$ and the relative variance using

resampling at every time step is approximately equal to

$$\frac{\mathbb{V}(\widetilde{Z}_n)}{Z_n^2} \approx \frac{1}{N}\left[\left(\int \frac{\pi_1^2(x_1)}{q_1(x_1)}\,dx_1 - 1\right) + \sum_{k=2}^{n}\left(\int \frac{\pi_k^2(x_{1:k})}{\pi_{k-1}(x_{1:k-1})\,q_k(x_k|x_{1:k-1})}\,dx_{k-1:k} - 1\right)\right]$$
$$= \frac{n}{N}\left[\left(\frac{\sigma^4}{2\sigma^2 - 1}^{1/2}\right) - 1\right]$$

which is linear in n in contrast to the exponential growth of the IS estimate of the relative variance. If we again select $\sigma^2 = 1.2$ then to obtain a relative variance of 0.01 we only need $N \approx 10^4$ particles instead of the $N \approx 2 \cdot 10^{23}$ particles that were needed for the IS estimate to obtain the same precision. That is, we obtain an improvement by 19 orders of magnitude.

This setup favors SMC massively since the density $\pi_n(x_{1:n})$) factorizes. A more realistic example is given in Example 2.9.

Reducing variance of Resampling

While resampling helps mitigate weight degeneracy by focusing computational effort on high-probability regions, it introduces additional variance into the algorithm. In this section, we describe two distinct techniques that can reduce this extra variance, thereby improving the overall efficiency of the SISR algorithm. Notably, these approaches are complementary and can be applied simultaneously.

We can reduce the variance introduced during resampling by leveraging sampling concepts such as stratified sampling. In stratified sampling, the interval $[0, 1]$ is divided into K equal strata, and one uniform random number is drawn from each subinterval. Formally, for $i = 1, \ldots, K$, we draw

$$u_i \sim \text{Uniform}\left(\frac{i-1}{K}, \frac{i}{K}\right).$$

Each u_i is then used to select a particle based on the cumulative normalized weights.

Another way to reduce the variance of the SISR algorithm is to only do the resampling step when we have many particles with low weights. We will call this Sequential Importance Sampling with Adaptive Resampling (SISAR). A common metric used to decide when to trigger a resampling step is the *effective sample size* (ESS), first introduced by Liu and Chen [1995]. The ESS at time n is defined as

$$\text{ESS}_n \coloneqq \frac{1}{\sum_{i=1}^{N}\left(W_n^{(i)}\right)^2}.$$

ESS can take a value between 1 and N. When the ESS falls below a predetermined threshold, N_τ, often chosen as $N_\tau = N/2$, it is an indication that most of the weight is carried by only a few particles, and resampling is then warranted.

Resampling within SMC

In the context of the SIS algorithm, resampling is integrated as an additional step to refresh the particle set, given that the resampling condition is met. A generic algorithm incorporating adaptive resampling is described in Algorithm 2.8.

Algorithm 2.8 Sequential Importance Sampling with Adaptive Resampling (SISAR)

Generate particles $x_1^{(i)}$ from $q_1(x_1)$ for $i = 1, \ldots, N$
for each time step $n = 2, \ldots, T$ **do**
    for each particle $i = 1, \ldots, N$ **do**
        Generate particles $x_n^{(i)}$ from $q_n(x_n | x_{1:n-1}^{(i)})$
        Compute the incremental importance weight:

$$\alpha_n^{(i)} = \frac{\gamma_n(x_{1:n}^{(i)})}{\gamma_{n-1}(x_{1:n-1}^{(i)}) q_n(x_n^{(i)} | x_{1:n-1}^{(i)})}$$

        Update the particle weight: $w_n^{(i)} = w_{n-1}^{(i)} \cdot \alpha_n^{(i)}$
    end for
    Normalize the weights: $W_n^{(i)} \leftarrow \frac{w_n^{(i)}}{\sum_{i=1}^N w_n^{(i)}}$
    if Resampling condition is met **then**
        Draw N indices $\{a^{(i)}\}_{i=1}^N$ from $\{1, \ldots, N\}$ according to the probabilities $\{W_n^{(i)}\}$.
        Set $x_{1:n}^{(i)} \leftarrow x_{1:n}^{(a^{(i)})}$ for all i.
        Reset weights: $w_n^{(i)} = 1$.
    end if
end for

Again, we can at any step n estimate $\pi_n(x_{1:n})$ and Z_n by respectively (2.5) and (2.8).

Example 2.9 (Example with weight degeneracy?)**.** Make example

3 Particle Markov Chain Monte Carlo Methods

Rewrite

Suppose we have a hidden Markov model (HMM), also called a state-space model (SSM). Specifically, we consider a discrete-time Markov process $\{X_n; n \geq 1\}$, where $X_n \subset \mathcal{X}^n$ for each n. It is characterized by an initial distribution

$$X_1 \sim \mu(x_1)$$

and the transition density

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

for some parameter $\theta \in \Theta$. Our goal is to infer the latent states $\{X_n\}$, given a sequence of noisy observations $\{Y_n; n \geq 1\}$, where $Y_n \subset \mathcal{Y}^n$ for each n. The observation Y_n is assumed to be conditionally independent given X_n, meaning that for $1 \leq n \leq m$,

$$Y_n|(X_1 = x_1, \ldots, X_n = x_n, \ldots, X_m = x_m) \sim g_\theta(y_n|X_n = x_n).$$

Let $y_{1:T} = (y_1, \ldots, y_T)$ denote the sequence of observations up to time $T \geq 1$. If $\theta \in \Theta$ is known then Bayesian inference gives us the posterior density as $p_\theta(x_{1:T}|y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T})$ where

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

When θ is unknown we give a prior density $p(\theta)$ to θ and Bayesian inference then relies on the joint posterior density

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

In most cases, neither $p_\theta(x_{1:T}, y_{1:T})$ nor $p(\theta, x_{1:T}|y_{1:T})$ admit closed-form solutions, necessitating the use of Monte Carlo methods for inference. In most cases, directly sampling the densities is infeasible due to the high-dimensional nature of the state space and the complexity of the model. In the case of a known $\theta \in \Theta$ sequential Monte Carlo (SMC) is a class of algorithms to approximate sequentially the sequence of posterior densities $\{p_\theta(x_{1:n}|y_{1:n}; n \geq 1)\}$ and the sequence of marginal likelihoods $\{p_\theta(y_{1:n}); n \geq 1\}$.

SMC methods are a general class of Monte Carlo methods that sample sequentially from a sequence of target probability densities $\{\pi_n(x_{1:n}); n \geq 1\}$, where $\pi_n(x_{1:n}) \subset \mathcal{X}^n$ for each n. In many practical applications, the normalizing constant Z_n is unknown, and one of the advantages of SMC is that it also provides an estimate $\widehat{Z}_n$ alongside approximations of the target distribution. Specifically, at time $n = 1$ we get an approximation of $\pi_1(x_1)$ and an estimate of Z_1, then at time $n = 2$ we get an approximation of $\pi_2(x_{1:2})$ and an estimate of Z_2 and so on. Here for instance we could have $\gamma_n(x_{1:n}) = p(x_{1:n}, y_{1:n})$ and $Z_n = p(y_{1:n})$, thus $\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$.

3.1 Filtering

4 Simulation Studies

5 Application to Stochastic SIR Models

6 Conclusion and Future Work

Bibliography

C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3):269–342, 05 2010. doi: 10.1111/j.1467-9868.2009.00736.x. URL https://doi.org/10.1111/j.1467-9868.2009.00736.x.

A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12, 01 2009.

D. Kroese, T. Taimre, and Z. Botev. *Handbook of Monte Carlo Methods*. Wiley Series in Probability and Statistics. Wiley, 2013. ISBN 9781118014950. URL https://books.google.dk/books?id=Trj9HQ7G8TUC.

J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995. doi: 10.1080/01621459.1995.10476549. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476549.

P. Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Probability and Its Applications. Springer New York, 2004. ISBN 9780387202686. URL https://books.google.dk/books?id=8LypfuG8ZLYC.

Shimao. How is this minimum variance worked out for this importance sampling estimator? Cross Validated, https://stats.stackexchange.com/q/324834, 2018. (Version: 2018-01-24).