

# SMC for Bayes

Joshua J Bon

2019-05-24

or “Sequential Monte Carlo for static Bayesian model”

## Prerequisites

Assumed knowledge consists of

1. Some experience with Bayesian modelling and inference.
2. Some understanding of Markov chain Monte Carlo algorithms.
3. Tolerance of vague, possibly misleading subheadings.

## Do it for Bayes (some motivation)

Monte Carlo methods (MC) are a suite of probabilistic computational techniques that simulate random variables from probability distributions that have density or mass functions that are difficult to study analytically. For example, such probability laws can have forms that are computationally expensive, or analytically intractable. Analytic intractability primarily manifests as an inability to estimate a multivariate integral, the normalising constants of the target probability distribution. Monte Carlo methods bypass evaluation of this integral by generating draws from the target distribution, from which quantities such as moments or interval probabilities can be approximated. Monte Carlo methods are used in many contexts including in Bayesian statistics to estimate parameter posterior distributions<sup>1</sup>, thereby quantifying uncertainty and enabling inference about the real world.

The justification for Monte Carlo integration appeal to a law of large numbers for integrable functions. The integral, or expectation of  $f$ , assuming it is integrable,

$$E_{\pi}[f(\theta)] = \int_{\Theta} f(\theta)\pi(\theta) d\theta = \lim_{n \rightarrow \infty} n^{-1} \sum_{i=1}^n f(\theta^{(i)}) \quad (1)$$

if the random draws  $\theta^{(i)}$  have distribution  $\pi(\theta)$ . The approximation from the limit is therefore

$$E_{\pi}[f(\theta)] \approx n^{-1} \sum_{i=1}^n f(\theta^{(i)}) \quad \text{when } \theta^{(i)} \sim \pi(\theta)$$

for large enough  $n$ . Modern MC methods focus on simulating from  $\pi(\theta)$  efficiently, and use the above approximation for inference.

<sup>1</sup> A parameter's posterior distribution is the probability the parameter takes certain values after accounting for prior beliefs and available evidence (or data).

Sequential Monte Carlo (SMC) is an attractive and robust MC method that is commonly used in Bayesian inference. It can be preferred to competing methods, such as Markov chain Monte Carlo (MCMC), as it is adaptive, parallelisable, and better equipped to handle multimodality and non-linear dependence in target distributions. SMC, unlike MCMC, can also be easily applied to modelling data that arrives in real-time, often referred to as sequential Bayesian analysis, as opposed to static analysis.

This research will focus on improving the efficiency, robustness, and applicability of sequential Monte Carlo methods, primarily for static Bayesian inference.

## Old school (historical notes)

Contemporary SMC has its origins in importance sampling and particle filtering, which are often motivated by Bayesian statistics and signal processing, respectively. The term ‘Sequential Monte Carlo’, appears at least as early as [Liu and Chen \(1998\)](#) where importance sampling, resampling, and Markov chain methods were combined to estimate dynamic systems<sup>2</sup> in real-time. This work drew heavily from particle filters such as the Bootstrap ([Gordon et al., 1993](#)) and Kalman filters (see [Chen, 2003](#), for example), as well as sequential importance sampling from the time.

<sup>2</sup> State-space models for example.

Whilst the key ingredients of SMC have remained much the same since [Liu and Chen \(1998\)](#), numerous theoretical and algorithmic advances have been made. The work of [Del Moral et al. \(2006\)](#) contains the canonical exposition of modern SMC, describing very general methods with accompanying theory to guide the construction of effective SMC algorithms. The scope of their work was for models with a fixed number of parameters, which is the case for static Bayesian inference.

In general, an SMC algorithm operates by iteratively updating a population of particles (realisations of random variables) to match a sequence of desired distributions. The final distribution in the sequence is the target, which is difficult to sample from directly. The algorithm incrementally progresses particles from one distribution to the next by iteratively reweighting and resampling, followed by refreshing the particles to add diversity. The first two steps are equivalent to importance sampling ([Kroese et al., 2011](#), Ch. 9.7), whilst the third often uses an MCMC kernel to diversify replicated particles in the sample, thereby avoiding degeneracy. We will now describe these steps in more detail.

## New wave (modern SMC)

SMC uses a series of densities,  $\pi_1, \pi_2, \dots, \pi_T$ , to iteratively generate a realised set of random variables (or particles) whose distribution approaches a desired law, say the density  $\pi(\theta) = \pi_T(\theta)$  for random variable  $\theta \in \Theta$ . The

sequence, indexed by  $t = 1, 2, \dots, T$ , has corresponding particles and weights described by

$$\{\theta_t^{(i)}, w_t^{(i)}\}_{i=1}^I$$

for some  $t$  and the number of particles  $I > 1$ . The SMC algorithm is designed such that a particle at index  $i$  and iteration  $t$ ,  $\theta_t^{(i)}$ , has distribution with density  $\pi_t(\theta)$ . The starting density,  $\pi_1(\theta)$ , is chosen so that it can be simulated from easily, whilst the density we are targeting is that of  $\pi(\theta) = \pi_T(\theta)$ . We proceed through a series of densities (in truth, particles with the current density as their distribution) that are increasingly complex. Each subsequent density is “near” enough to the proceeding one so that evolving the particles from one to the next is feasible. The weights are such that  $w_t^{(i)} > 0$ , and  $\sum_{i=1}^I w_t^{(i)} = 1$  for  $t = 1, 2, \dots, T$ .

The exact sequence of distributions to move from  $\pi_1(\theta)$  to  $\pi_T(\theta)$  can be defined in a number of ways. The most popular schedules are

- *data annealing*, where  $\pi_t(\theta) \propto p(\theta | y_1, \dots, y_t)$ , where the target is  $\pi(\theta) = p(\theta | y_1, \dots, y_T)$  and  $T = N$  the number of data observations; and
- *geometric annealing*, where  $\pi_t(\theta) \propto \pi(\theta)^{\phi_t} \mu(\theta)^{1-\phi_t}$  for  $0 = \phi_1 < \phi_2 < \dots < \phi_{T-1} < \phi_T = 1$ , and  $\mu(\theta)$  is the starting distribution of the particles.
- *likelihood annealing* (a special case of geometric for static Bayesian models), where  $\pi_t(\theta | \mathbf{y}) \propto \pi(\mathbf{y} | \theta)^{\phi_t} p(\theta)$ , the likelihood function is  $\pi(\mathbf{y} | \theta)$ , and  $p(\theta)$  is the prior distribution. The annealing parameter  $\phi_t$  is an in geometric annealing.

Data annealing was first used in [Chopin \(2002\)](#) for static Bayesian models, whilst geometric annealing has its origins in papers by [Gelman and Meng \(1998\)](#) as well as [Neal \(2001\)](#). The components that make up an SMC algorithm, can be roughly divided into three iterated steps:

1. *Reweight* particles for the next distribution in the sequence;
2. *Resample* particles to duplicate “good” particles and remove “bad” ones;
3. *Refresh* particles to increase diversity after particles duplicated above.

We will now outline these steps in more detail.

### Reweight

The reweighting step is based on importance sampling, which can be seen as a more general form of the Monte Carlo integration identity. Specifically, the expectation in (1) can be rewritten as

$$E_\pi[f(\theta)] = \int_{\Theta} f(\theta) \pi(\theta) d\theta = \int_{\Theta} f(\theta) \frac{\pi(\theta)}{\pi_0(\theta)} \pi_0(\theta) d\theta = E_{\pi_0}[w(\theta) f(\theta)]$$

for some density  $\pi_0(\theta)$  with the same support as  $\pi(\theta)$ , where  $w(\theta) = \frac{\pi(\theta)}{\pi_0(\theta)}$ .

Then by stochastic approximation, we can write

$$E_{\pi}[f(\boldsymbol{\theta})] \approx \sum_{i=1}^n w_i f(\boldsymbol{\theta}^{(i)}) \quad \text{when } \boldsymbol{\theta}^{(i)} \sim \pi_0(\boldsymbol{\theta}) \quad (2)$$

where  $w_i = \frac{\pi(\boldsymbol{\theta}^{(i)})}{n\pi_0(\boldsymbol{\theta}^{(i)})}$ , for large  $n$ . The density  $\pi_0(\boldsymbol{\theta})$  is called the importance distribution. It is from this approximation that importance sampling, and its derivatives are borne.<sup>3</sup>

In relation to SMC, importance sampling is performed iteratively, as in sequential or annealed importance sampling (see Neal, 2001, and references therein). At step  $t + 1$  the previous particles and weights are  $\{\boldsymbol{\theta}_t^{(i)}, w_t^{(i)}\}_{i=1}^I$ . These particles are then evaluated against the new targeted density  $\pi_{t+1}(\boldsymbol{\theta})$ . That is the previous density,  $\pi_t(\boldsymbol{\theta})$  is the importance distribution for the next density  $\pi_{t+1}(\boldsymbol{\theta})$ . Due to the sequential nature of the algorithm, the random samples from  $\pi_t(\boldsymbol{\theta})$  have already been generated. However, each particle will be differently suited to the new density compared to the old one, so the weights must be updated.

In general determining the weights is non-trivial, but an auxiliary parameter approach can be used (Del Moral et al., 2006, Sec. 3.1). There is an important simplification available when the move step is an MCMC kernel with invariant distribution  $\pi_{t+1}$  for a static model. Under this condition, the process of updating the unnormalised weights,  $W_{t+1}$  can be performed using the formula (Del Moral et al., 2006, Sec. 3.3.2.3)

$$W_{t+1}^{(i)} = W_t^{(i)} \frac{\gamma_{t+1}(\boldsymbol{\theta}_{t+1}^{(i)})}{\gamma_t(\boldsymbol{\theta}_t^{(i)})}$$

where  $\gamma_t(\boldsymbol{\theta})$  is the unnormalised target density at time  $t$ , i.e.  $\pi_t(\boldsymbol{\theta}) = \gamma_t(\boldsymbol{\theta})/Z_t$  and  $Z_t$  is the normalising constant. The weights can then be normalised by

$$w_t^{(i)} = \frac{W_t^{(i)}}{\sum_{i=1}^I W_t^{(i)}}.$$

### Resample

The particle weights will degenerate each iteration and eventually most of the weight will be concentrated over a small set of particles. This is the main draw back of standard importance sampling or sequential importance sampling without resampling and refreshment. Degenerate weights cause two problems for SMC and other importance sampling algorithms:

1. It is inefficient to use computational resources to store and manipulate particles with a small weights. In particular, particles with small weights contribute very little to the Monte Carlo integration.
2. Large discrepancies in weight increases the variance of the estimate of the integral. Or in other words, the estimate is less accurate on average.

<sup>3</sup> Sufficient conditions for workable importance samplers are given in Geweke (1989) and relate to the finiteness of the expected value under the importance distribution and supremum of the ratio of target distribution to importance distribution.

Resampling addresses the first issue by taking a random sample of the particles with probability proportional to the current weight. This preserves the expectation, removes weights with low values, and duplicates those with high values. There are three main strategies for resampling:

1. *Multinomial resampling*: Sample  $I$  particles from the current particles with replacement, with the probability of selecting a given particle proportional to its current weight. After which each particle has unit weight.
2. *Stratified resampling*: Partition the particles into  $m$  sets by weight, such that each block (of the partition) has the same total weight, and then sample  $I/m$  particles from each block<sup>4</sup> (Kitagawa, 1996). After which each particle has unit weight.
3. *Residual resampling*: Randomly round the current weights to integer values, such that the mean is preserved (see for example, Liu and Chen, 1998). Duplicate particles based on their integer weight, after which each new particle has unit weight.

<sup>4</sup> Assuming  $I/m$  is an integer.

Systematic resampling is also popular and can be viewed as a simplified version of stratified sampling. Other variations are also possible but they often lack an intuitive description, unlike the three examples given above. Generally, resampling strategies are functions of empirical inverse tail probabilities (Gerber et al., 2019) which is a standard basis for describing, comparing, and efficiently implementing them.

It is well known that stratified and residual resampling outperform multinomial by reducing the variance of the particles. For general SMC algorithms, stratified sampling is recommended as it is more efficient and has stronger proven theoretical guarantees of convergence than its peers (Gerber et al., 2019). Other specialised resampling techniques also exist for certain situations, for example when resampling particles in a distributed computer setting (Lee and Whiteley, 2016; Murray et al., 2016).

Whilst resampling addresses the first issue, the second issue requires a combination of resampling and refreshment to be effectively dealt with. We detail how particles are refreshed below.

### Refresh

After resampling, there are  $I$  particles with unit weight. However, many of them may be duplicated. Refreshment<sup>5</sup> increases the diversity of the particles by allowing them to traverse the parameter space. Using a combination of resampling and refreshment addresses the second issue mentioned above: Particles with relatively large weights are duplicated then diversified so that degeneracy in the particles can be avoided and the variance of the integral estimator can be decreased.

<sup>5</sup> Refreshment is also known as mutation, move step, or jittering among other names.

We focus on the situation where an MCMC kernel is used for refreshment because this is the most common practise for static Bayesian models. The resampled particles will be  $\{\theta_{t+1}^{(i)}\}_{i=1}^I$  with unit weights. An MCMC kernel, or transition probability density (or mass) function with the Markovian prop-

erty,  $k(\theta \mid \theta_{t+1})$ , can be used if the stationary (or invariant) distribution of the Markov chain defined by  $k$  is  $\pi_{t+1}$  where  $t + 1$  is the current time. We update each particle by sampling from this kernel with  $\theta_{t+1}$  equal to the current position. This step can be iterated multiple times, and the last value is taken as the current position of each particle. The global current time is not incremented as the particle in the chain still has a  $\pi_{t+1}$  law, due to the MCMC kernel's invariance. Note that the weights need to be updated after mutation, but this can generally be left until the next iteration of the algorithm.

Whilst we have described mutation as the last step, this does not diminish its importance from an algorithmic and historical perspective. In fact the origins of SMC for static models are just as much based on resample-move algorithms (Chopin, 2002, for example), as they are in annealed importance sampling (Neal, 2001).

## Improve. Adapt. Overcome.

Several enhancements are used in modern SMC algorithms to dramatically increase their efficiency, and are outlined in the subsections below. These improvements result from dynamic SMC algorithms which adapt using the current population of particles. Whilst they can improve efficiency, adaptive SMC algorithms are not covered by the theoretical convergence guarantees developed for their non-adaptive counterparts. The theory for some special cases of adaptivity has been developed by Del Moral et al. (2012) and Beskos et al. (2016). When this theory is not available the issue can be circumvented using a pilot run of the entire SMC algorithm. The adapted parameters from the pilot run are fixed for a second, non-adaptive version of SMC, which will be covered by the general convergence results for SMC in static models (see for example, Del Moral et al., 2006).

## References

- Alexandros Beskos, Ajay Jasra, Nikolas Kantas, Alexandre Thiery, et al. On the convergence of adaptive sequential monte carlo methods. *The Annals of Applied Probability*, 26(2):1111–1146, 2016.
- Zhe Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics: A Journal of Theoretical and Applied Statistics*, 182(1):1–69, 2003.
- Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.
- Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

- Pierre Del Moral, Arnaud Doucet, Ajay Jasra, et al. On adaptive resampling strategies for sequential monte carlo methods. *Bernoulli*, 18(1):252–278, 2012.
- Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, 13(2):163–185, 1998.
- Mathieu Gerber, Nicolas Chopin, and Nick Whiteley. Negative association, ordering and convergence of resampling methods. *Annals of Statistics*, (forthcoming), 2019.
- John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–1339, 1989.
- Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.
- Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1): 1–25, 1996.
- Dirk P Kroese, Thomas Taimre, and Zdravko I Botev. *Handbook of Monte Carlo methods*. John Wiley & Sons, Hoboken, New Jersey, 2011.
- Anthony Lee and Nick Whiteley. Forest resampling for distributed sequential Monte Carlo. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 9(4):230–248, 2016.
- Jun S Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.
- Lawrence M Murray, Anthony Lee, and Pierre E Jacob. Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics*, 25(3): 789–805, 2016.
- Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.