Models and algorithms in sequential Monte Carlo

Written by Joshua J Bon in preparation for the SMC Down Under Workshop July 2023.

1 Connecting algorithms to models

We present the Feynman-Kac model in measure-theoretic notation, for generality, and how it is typically presented. For this we need latent-states $x_t \in \mathsf{X}$, in the state space X for $t \in \{1, \dots, n\}$. The probability measures (distributions, M_1, η_t) and Markov kernels (M_t) are defined on the measurable space $(\mathsf{X}, \mathscr{X})$. The Markov kernels represent conditioning by the ordering of their two arguments, i.e. $M_t(x_{t-1}, \mathrm{d}x_t)$ is conditional on x_{t-1} for $t \in \{2, \dots, n\}$.

As for the hidden Markov model, if we assume $\mathsf{X} = \mathbb{R}^d$ and $\mathscr{X}$ is the Borel σ-algebra for X, then we can forgo the measure-theoretic notation in favour of probability density functions $p(\cdot)$ with respect to $\mathbb{R}^d$. The densities represent conditioning by using the vertical line "|", i.e. $p(x_t \mid x_{t-1})$ is conditional on x_{t-1} for $t \in \{2, \dots, n\}$.

We continue with the $\mathsf{X} = \mathbb{R}^d$ assumption for the static Bayesian models. Here, we have a sequence of distributions, defined by the probability densities

$$\pi_1, \pi_2, \dots, \pi_n$$

if we forgo the last iteration of the SMC sampler (i.e. stop algorithm after at $(n-1)$th iteration). The distribution π_n is our target we wish to generate samples from. We present a simple SMC sampler where we begin with the prior distribution and use q_t as a π_t-invariant MCMC kernel for $t \in \{2, \dots, n\}$.

Feynman-Kac model	Hidden Markov model (Bootstrap PF)	Static Bayesian model (Resample-move SMC)
Initial distribution $M_1(\mathrm{d}x_1)$	*Prior density* $p(x_1)$	*Initial / prior density* $\pi_1(x_1)$
Markov kernel $M_t(x_{t-1}, \mathrm{d}x_t)$	*Forward kernel density* $p(x_t \mid x_{t-1})$	*MCMC kernel / mutation density* $q_t(x_t \mid x_{t-1})$
Potential function $G_t(x_t)$	*Likelihood function* $p(y_t \mid x_t)$	*Incremental weighting function* $\dfrac{\pi_{t+1}(x_t \mid y)}{\pi_t(x_t \mid y)}$
Predictive distribution $\eta_t(\mathrm{d}x_t)$	*Predictive density* $p(x_t \mid y_{1:t-1})$	*Time t target density* $\pi_t(x_t \mid y)$
Updated distribution $\hat{\eta}_t(\mathrm{d}x_t)$	*Filtering density* $p(x_t \mid y_{1:t})$	*Time t + 1 target density* $\pi_{t+1}(x_t \mid y)$

Emphasis is often put on the predictive distribution for particle filters, since their context is the online setting where x_t are unobserved latent variable evolving over time. For example, at time $n+1$ you can predict the next state x_{n+1} using $p(x_{n+1} \mid y_n)$.

For the SMC sampler, terminating at time $n-1$, generates a particle approximation for π_n, our target distribution.

1.1 Algorithms

We present several SMC algorithms to highlight their similarities. In practice slight changes to these algorithms may be made for computational efficiency or convenience, and ease of exposition in certain contexts.

We present multinomial resampling, but other options such as systematic resampling are possible. Adaptive resampling can also be used with minor changes to the algorithms presented. For the mechanics of the multinomial resampling we use a categorical distribution to sample the ancestors of the next generation of particles. We denote the categorical distribution by $\text{Cat}(w_1, w_2 \ldots, w_m)$ as the categorical distribution with support $\{1, 2, \ldots, m\}$ with probabilities proportional to unnormalised weights $(w_1, w_2, \ldots, w_m)$ for $w_i \geq 0$.

1.1.1 Bootstrap particle filter for Feynman-Kac model

The simplest particle filter we can apply in general is the Bootstrap Particle Filter. In this presentation we highlight how the ancestors of the particles are used in the resampling step. The resampling step is stylised as a categorical distribution with weights $G_{t-1}(\zeta_{t-1}^i)$ for

Algorithm 1 The Bootstrap Particle Filter (Feynman-Kac model)

1. Sample initial $\zeta_1^i \overset{\text{iid}}{\sim} M_1$ for $i \in \{1, \ldots, N\}$
2. For each time $t = 2, \ldots, n$
   (i) Sample ancestors $A_{t-1}^i \sim \text{Cat}\left(G_{t-1}(\zeta_{t-1}^1), \ldots, G_{t-1}(\zeta_{t-1}^N)\right)$ for $i \in \{1, \ldots, N\}$
   (ii) Sample prediction $\zeta_t^i \sim M_t(\zeta_{t-1}^{A_{t-1}^i}, \cdot)$ for $i \in \{1, \ldots, N\}$

Output: Particles $\zeta_t^{1:N}$ for $t = 1, \ldots, n$

$i \in \{1, \ldots, N\}$. Particle approximations from the BPF are defined as

$$\eta_t^N(f) = N^{-1} \sum_{i=1}^{N} f(\zeta_t^i), \quad \hat{\eta}_t^N(f) = \sum_{i=1}^{N} \frac{f(\zeta_t^i) G_t(\zeta_t^i)}{\sum_{j=1}^{N} G_t(\zeta_t^j)} \tag{1}$$

for $t = \{1, \ldots, n\}$. Note that the final update step with G_n is not applied during the BPF, but we can still use the weighted particle approximation to the updated distribution above.

1.1.2 Bootstrap particle filter for Hidden Markov model

In this presentation we highlight the weight interpretation of the resampling step. We also include the calculation of the terminal weights in the algorithm, but this may not always be necessary (e.g. no observation at time n).

Algorithm 2 The Bootstrap Particle Filter (HMM model)

1. Sample initial $\zeta_1^i \overset{\text{iid}}{\sim} p(x_1)$ for $i \in \{1,\dots,N\}$
2. For each time $t = 2,\dots,n$
   (i) Calculate weights $w_{t-1}^i = p(y_{t-1} \,|\, x_{t-1} = \zeta_{t-1}^i)$ for $i \in \{1,\dots,N\}$
   (ii) Sample ancestors $A_{t-1}^i \sim \text{Cat}\left(w_{t-1}^1,\dots,w_{t-1}^N\right)$ for $i \in \{1,\dots,N\}$
   (iii) Sample prediction $\zeta_t^i \sim p(x_t \,|\, x_{t-1} = \hat{\zeta}_{t-1}^{A_{t-1}^i})$ for $i \in \{1,\dots,N\}$
3. Calculate terminal weights w_n^i as in step 2(i)

Output: Particles $\zeta_t^{1:N}$ and weights $w_t^{1:N}$ for $t = 1,\dots,n$

Particle approximations from the BPF (2) are defined as

$$\mathbb{E}(f(X_t) \,|\, Y_{1:t-1} = y_{1:t-1}) \approx \eta_t^N(f) = \frac{\sum_{i=1}^N f(\zeta_t^i)}{N}$$
$$\mathbb{E}(f(X_t) \,|\, Y_{1:t} = y_{1:t}) \approx \hat{\eta}_t^N(f) = \frac{\sum_{i=1}^N f(\zeta_t^i) w_t^i}{\sum_{i=1}^N w_t^i}$$

for $t = \{2,\dots,n\}$.

1.1.3 SMC sampler as a Bootstrap Particle Filter

This is the resample-move (or sequential importance sampling resampling) algorithm as a bootstrap particle filter. General sequential Monte Carlo samplers can also be framed in this way.

Algorithm 3 SMC sampler

1. Sample initial $\zeta_1^i \overset{\text{iid}}{\sim} \pi_1(x_1)$ for $i \in \{1,\dots,N\}$
2. For each time $t = 2,\dots,n$
   (i) Calculate weights $w_{t-1}^i = \frac{\pi_t(\zeta_{t-1}^i)}{\pi_{t-1}(\zeta_{t-1}^i)}$ for $i \in \{1,\dots,N\}$
   (ii) Sample ancestors $A_{t-1}^i \sim \text{Cat}\left(w_{t-1}^1,\dots,w_{t-1}^N\right)$ for $i \in \{1,\dots,N\}$
   (iii) Sample MCMC $\zeta_t^i \sim q_t(x_t \,|\, x_{t-1} = \hat{\zeta}_{t-1}^{A_{t-1}^i})$ for $i \in \{1,\dots,N\}$

Output: Particles $\zeta_n^{1:N}$ (or $\zeta_{n-1}^{1:N}$ and weights $w_{n-1}^{1:N}$)

1.2 Unnormalised measures

Beginning with $\gamma_1 = M_1$, the *predictive* (unnormalised) measure can be defined recursively as

$$\gamma_t(S) = \int \gamma_{t-1}(\mathrm{d}x) G_{t-1}(x) M_t(x, S), \quad S \in \mathscr{X}$$

for $t \in \{2, \ldots, n\}$. These measures relate to the predictive distribution by $\eta_t(\mathrm{d}x_t) = \gamma_t(\mathrm{d}x_t)/Z_t$ where $Z_t = \int \gamma_t(\mathrm{d}x_t)$ whilst the *updated* (unnormalised) measures are

$$\hat{\gamma}_t(\mathrm{d}x_t) = \gamma_t(\mathrm{d}x_t) G_t(x_t)$$

relating to the updated distribution $\hat{\eta}_t(\mathrm{d}x_t) = \hat{\gamma}_t(\mathrm{d}x_t)/\hat{Z}_t$ where $\hat{Z}_t = \int \hat{\gamma}_t(\mathrm{d}x_t)$ for $t \in \{1, \ldots, n\}$.

1.3 Normalising constant

Using boldface we will define the path distribution and potential function as

$$\mathbf{M}_n(\mathrm{d}x_{1:n}) = M_1(\mathrm{d}x_1) \prod_{t=2}^{n} M_t(x_{t-1}, \mathrm{d}x_t),$$
$$\mathbf{G}_n(x_{1:n}) = \prod_{t=1}^{n} G_t(x_t),$$

respectively.

The normalising constants can also be defined as

$$Z_n = \int \mathbf{G}_{n-1}(x_{1:n-1}) \mathbf{M}_{n-1}(\mathrm{d}x_{1:n-1}) \tag{2}$$
$$\hat{Z}_n = \int \mathbf{G}_n(x_{1:n}) \mathbf{M}_n(\mathrm{d}x_{1:n}) \tag{3}$$

or as path expectations

$$Z_n = \mathbb{E}\left[\mathbf{G}_{n-1}(X_{1:n-1})\right], \qquad X_{1:n-1} \sim \mathbf{M}_{n-1} \tag{4}$$
$$\hat{Z}_n = \mathbb{E}\left[\mathbf{G}_n(X_{1:n})\right], \qquad X_{1:n} \sim \mathbf{M}_n. \tag{5}$$

The "hat" denotes that this quantity is the normalising constant for the updated distribution, rather than the predictive distribution. Notation for SMC samplers typically drops the "hat" on $\hat{Z}_n$ since only the updated distribution is of interest.