Asymptotic Genealogies of Sequential Monte Carlo Algorithms

Suzie Brown

26 April 2019

State space models

Hidden process $X_0, \ldots, X_T \in \mathcal{X}$
Observable process $Y_0, \ldots, Y_T \in \mathcal{Y}$

$$\begin{tikzcd}
 & Y_{t-1} & Y_t & \\
\cdots \arrow[r, "f"] & X_{t-1} \arrow[u, "g"'] \arrow[r, "f"] & X_t \arrow[u, "g"'] \arrow[r, "f"] & \cdots
\end{tikzcd}$$

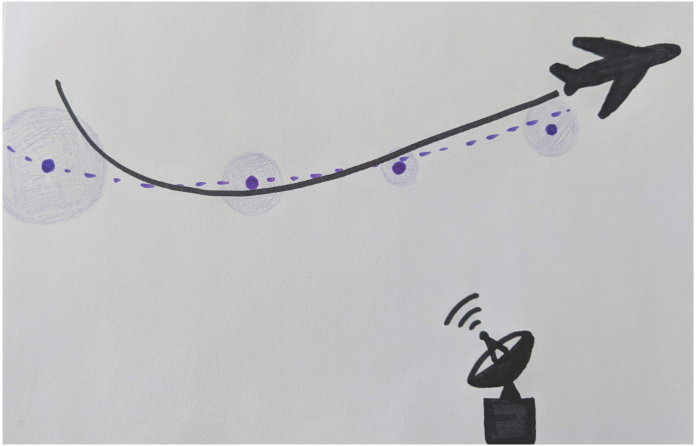
State space models

Hidden process $X_0, \ldots, X_T \in \mathcal{X}$
Observable process $Y_0, \ldots, Y_T \in \mathcal{Y}$

$$
\begin{tikzcd}
 & Y_{t-1} & Y_t & \\
\cdots \arrow[r, "f"] & X_{t-1} \arrow[u, "g"'] \arrow[r, "f"] & X_t \arrow[u, "g"'] \arrow[r, "f"] & \cdots
\end{tikzcd}
$$

$$
\begin{aligned}
X_0 &\sim \mu(\cdot) \\
X_{t+1} \mid (X_t = x_t) &\sim f(\cdot|x_t) \\
Y_t \mid (X_t = x_t) &\sim g(\cdot|x_t)
\end{aligned}
$$

Target tracking



use noisy radar data to infer position/trajectory of aircraft:

- f models how aircraft moves
- g models uncertainty in radar measurements

Inference problems

Filtering: where is it now? $p(x_t|y_{0:t})$

Prediction: where will it go next? $p(x_{t+1}|y_{0:t})$

Smoothing: where has it been? $p(x_{0:t}|y_{0:t})$

Smoothing is "harder" than filtering/prediction [1].

Deterministic solutions

Kalman filter [2]

Under a linear Gaussian model:

$$\begin{aligned} X_0 &\sim \mathcal{N}(0, \Sigma_0) \\ X_{t+1} \mid (X_t = x_t) &\sim \mathcal{N}(Ax_t, \Sigma_x) \\ Y_t \mid (X_t = x_t) &\sim \mathcal{N}(Bx_t, \Sigma_y) \end{aligned}$$

we can recursively compute filtering distributions. Then a backward pass of RTS smoother [3] provides smoothing distributions.

This class of models is rather restrictive.

Deterministic solutions

Kalman filter [2]

Under a linear Gaussian model:

$$X_0 \sim \mathcal{N}(0, \Sigma_0)$$
$$X_{t+1} \mid (X_t = x_t) \sim \mathcal{N}(Ax_t, \Sigma_x)$$
$$Y_t \mid (X_t = x_t) \sim \mathcal{N}(Bx_t, \Sigma_y)$$

we can recursively compute filtering distributions. Then a backward pass of RTS smoother [3] provides smoothing distributions.

This class of models is rather restrictive.

Extended Kalman filter [4]

In non-linear Gaussian models, use a local linear approximation and apply Kalman filter.

This requires gradients and performs poorly in models that are very non-linear.

Deterministic solutions

Kalman filter [2]

Under a linear Gaussian model:

$$
\begin{aligned}
X_0 &\sim \mathcal{N}(0, \Sigma_0) \\
X_{t+1} \mid (X_t = x_t) &\sim \mathcal{N}(A x_t, \Sigma_x) \\
Y_t \mid (X_t = x_t) &\sim \mathcal{N}(B x_t, \Sigma_y)
\end{aligned}
$$

we can recursively compute filtering distributions. Then a backward pass of RTS smoother [3] provides smoothing distributions.

This class of models is rather restrictive.

Extended Kalman filter [4]

In non-linear Gaussian models, use a local linear approximation and apply Kalman filter.

This requires gradients and performs poorly in models that are very non-linear.

Unscented Kalman filter [5]

For highly non-linear Gaussian models, replace the prediction step of Kalman filter by propagating a representative set of points through f.

Deterministic solutions

- Kalman filter is optimal for linear Gaussian models, but extended/unscented Kalman filter are no longer optimal.

Deterministic solutions

- Kalman filter is optimal for linear Gaussian models, but extended/unscented Kalman filter are no longer optimal.
- All of these methods require a Gaussian model. Deterministic solutions are available for some other conjugate families [6], but these are still restrictive.

Deterministic solutions

- Kalman filter is optimal for linear Gaussian models, but extended/unscented Kalman filter are no longer optimal.
- All of these methods require a Gaussian model. Deterministic solutions are available for some other conjugate families [6], but these are still restrictive.
- Solutions are also available in the case that $\mathcal{X}$ is finite (integrals become sums), but we will mostly consider $\mathcal{X}$ to be a continuous space.

Deterministic solutions

- Kalman filter is optimal for linear Gaussian models, but extended/unscented Kalman filter are no longer optimal.
- All of these methods require a Gaussian model. Deterministic solutions are available for some other conjugate families [6], but these are still restrictive.
- Solutions are also available in the case that $\mathcal{X}$ is finite (integrals become sums), but we will mostly consider $\mathcal{X}$ to be a continuous space.

Sequential Monte Carlo (SMC) provides general-purpose (stochastic) methods that do not require a tractable model.

Deterministic solutions

- Kalman filter is optimal for linear Gaussian models, but extended/unscented Kalman filter are no longer optimal.
- All of these methods require a Gaussian model. Deterministic solutions are available for some other conjugate families [6], but these are still restrictive.
- Solutions are also available in the case that $\mathcal{X}$ is finite (integrals become sums), but we will mostly consider $\mathcal{X}$ to be a continuous space.

Sequential Monte Carlo (SMC) provides general-purpose (stochastic) methods that do not require a tractable model.

Only requires sampling from $f(\cdot|x)$, and pointwise evaluation of $g(y|x)$ up to a normalising constant for each y.

Sequential Monte Carlo

Prior: $p(x_{0:t}) = \mu(x_0) \prod_{i=1}^{t} f(x_i | x_{i-1})$

Likelihood: $p(y_{0:t} | x_{0:t}) = \prod_{i=0}^{t} g(y_i | x_i)$

Posterior: $p(x_{0:t} | y_{0:t}) \propto \mu(x_0) g(y_0 | x_0) \prod_{i=1}^{t} f(x_i | x_{i-1}) g(y_i | x_i)$

Sequential Monte Carlo

Prior: $p(x_{0:t}) = \mu(x_0) \prod_{i=1}^{t} f(x_i|x_{i-1})$

Likelihood: $p(y_{0:t}|x_{0:t}) = \prod_{i=0}^{t} g(y_i|x_i)$

Posterior: $p(x_{0:t}|y_{0:t}) \propto \mu(x_0)g(y_0|x_0) \prod_{i=1}^{t} f(x_i|x_{i-1})g(y_i|x_i)$

- Represent posterior distribution at time t with N particles.
- Posterior factorises sequentially, avoiding increase of dimension with T.

Sequential Monte Carlo

Algorithm

After initialisation, iterate these steps:

- **Propagate:** move the particles through the transition f
- **Calculate weights:** weight each particle according to g
- **Resample:** duplicate high-weight particles and kill off low-weight ones

Algorithm

After initialisation, iterate these steps:

- **Propagate:** move the particles through the transition f
- **Calculate weights:** weight each particle according to g
- **Resample:** duplicate high-weight particles and kill off low-weight ones

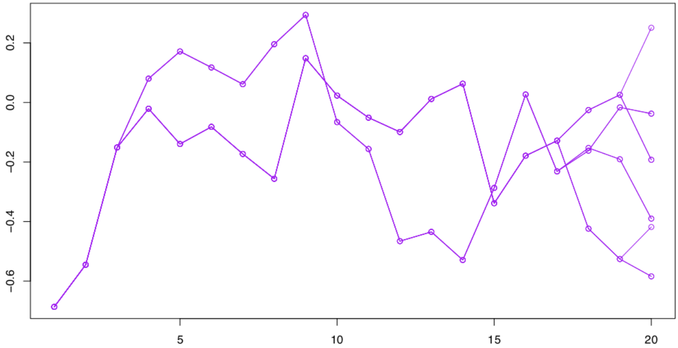
Approximate posterior distribution $p(dx_{0:t}|y_{0:t})$ by the empirical measure of the particles:

$$\hat{p}(dx_{0:t}|y_{0:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{X_{0:t}^{(i)}}(dx_{0:t})$$

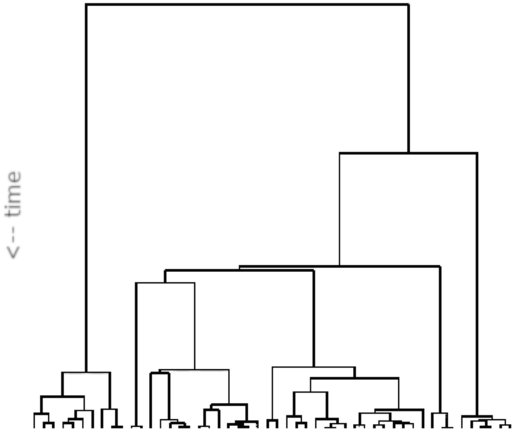
Ancestral degeneracy

For smoothing we need a sample of trajectories.

Resampling means that trajectories of time T particles coalesce backwards in time.

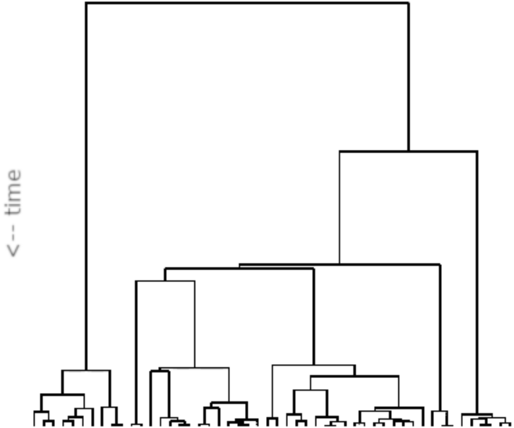


Kingman's coalescent



Looking backwards in time, each pair of lineages coalesces with unit rate.

Kingman's coalescent



Looking backwards in time, each pair of lineages coalesces with unit rate.

This is the limiting coalescent process in many population models as $N \to \infty$ [7, 8].

Coalescent for SMC

- Analysing the coalescent can help us to understand the performance of various SMC algorithms.

Coalescent for SMC

- Analysing the coalescent can help us to understand the performance of various SMC algorithms.
- The limiting coalescent of an SMC algorithm may depend on the resampling mechanism.

Coalescent for SMC

- Analysing the coalescent can help us to understand the performance of various SMC algorithms.
- The limiting coalescent of an SMC algorithm may depend on the resampling mechanism.
- For multinomial resampling, the limiting coalescent is a scaled Kingman coalescent [9].

Coalescent for SMC

- Analysing the coalescent can help us to understand the performance of various SMC algorithms.
- The limiting coalescent of an SMC algorithm may depend on the resampling mechanism.
- For multinomial resampling, the limiting coalescent is a scaled Kingman coalescent [9].
- What about other (more used) resampling schemes?

Resampling

Requirements:

- The total number of particles N remains fixed.
- The particles after resampling are equally weighted.
- The resampling scheme is unbiased; that is, the expected number of offspring of each particle i is equal to $Nw_t^{(i)}$.

Resampling

Multinomial: $v_t^{(1:N)} \stackrel{d}{=} \text{Multinomial}(N, w_t^{(1:N)})$

Resampling

Multinomial: $v_t^{(1:N)} \stackrel{d}{=} \text{Multinomial}(N, w_t^{(1:N)})$

Residual [10]: $v_t^{(1:N)} \stackrel{d}{=} \lfloor N w_t^{(1:N)} \rfloor + \text{Multinomial}(R, r_t^{(1:N)}/R)$

$$r_t^{(i)} := (N w_t^{(i)} - \lfloor N w_t^{(i)} \rfloor)$$

$$R := \sum r_t^{(i)}$$

Resampling

Multinomial: $v_t^{(1:N)} \stackrel{d}{=} \text{Multinomial}(N, w_t^{(1:N)})$

Residual [10]: $v_t^{(1:N)} \stackrel{d}{=} \lfloor N w_t^{(1:N)} \rfloor + \text{Multinomial}(R, r_t^{(1:N)}/R)$

$$r_t^{(i)} := (N w_t^{(i)} - \lfloor N w_t^{(i)} \rfloor)$$

$$R := \sum r_t^{(i)}$$

- Residual resampling yields lower Monte Carlo variance than multinomial resampling [11].
- Residual resampling is widely used by practitioners.
- Analysing the coalescent for residual resampling is a work in progress.

Particle Gibbs [12]

Hidden Markov model where transition depends on a hyperparameter θ:

$$
\begin{aligned}
&\theta \sim \nu(\cdot) \\
&X_0 \sim \mu(\cdot) \\
&X_{t+1} \mid (X_t = x_t) \sim f_\theta(\cdot|x_t) \qquad t = 0, \ldots, T-1 \\
&Y_t \mid (X_t = x_t) \sim g(\cdot|x_t) \qquad t = 0, \ldots, T
\end{aligned}
$$

Particle Gibbs [12]

Hidden Markov model where transition depends on a hyperparameter θ:

$$
\begin{aligned}
&\theta \sim \nu(\cdot) \\
&X_0 \sim \mu(\cdot) \\
&X_{t+1} \mid (X_t = x_t) \sim f_\theta(\cdot|x_t) \qquad t = 0, \ldots, T-1 \\
&Y_t \mid (X_t = x_t) \sim g(\cdot|x_t) \qquad t = 0, \ldots, T
\end{aligned}
$$

Gibbs sampler: alternately sample from $p(\theta|x_{0:t}, y_{0:t})$ and $p(x_{0:t}|\theta, y_{0:t})$.

Particle Gibbs [12]

Hidden Markov model where transition depends on a hyperparameter θ:

$$
\begin{aligned}
&\theta \sim \nu(\cdot) \\
&X_0 \sim \mu(\cdot) \\
&X_{t+1} \mid (X_t = x_t) \sim f_\theta(\cdot|x_t) \qquad t = 0, \dots, T-1 \\
&Y_t \mid (X_t = x_t) \sim g(\cdot|x_t) \qquad t = 0, \dots, T
\end{aligned}
$$

Gibbs sampler: alternately sample from $p(\theta|x_{0:t}, y_{0:t})$ and $p(x_{0:t}|\theta, y_{0:t})$.

- SMC is an appropriate method for sampling from $p(x_{0:t}|\theta, y_{0:t})$
- To target the correct posterior distribution, need to use *conditional SMC*.

Conditional SMC

- One “immortal” trajectory is conditioned to survive all of the resampling steps.
- Intuitively, this could have a significant effect on the coalescent.

Conditional SMC

- One "immortal" trajectory is conditioned to survive all of the resampling steps.
- Intuitively, this could have a significant effect on the coalescent.
- But we have this result:

Corollary 1

Under the conditions of [9, Lemma 3] , the genealogy of any n particles from a conditional SMC algorithm with multinomial resampling converges to Kingman's n-coalescent in the sense of finite-dimensional distributions, under an appropriate time-scaling.

Conditional SMC

- One "immortal" trajectory is conditioned to survive all of the resampling steps.
- Intuitively, this could have a significant effect on the coalescent.
- But we have this result:

Corollary 1

Under the conditions of [9, Lemma 3] , the genealogy of any n particles from a conditional SMC algorithm with multinomial resampling converges to Kingman's n-coalescent in the sense of finite-dimensional distributions, under an appropriate time-scaling.

- Intuition: as $N \to \infty$, there is zero probability that an arbitrary sample of fixed size n contains the immortal particle.

Summary

- SMC is widely used, but doesn't perform well for smoothing.

Summary

- SMC is widely used, but doesn't perform well for smoothing.
- Analysis of SMC coalescents allows us to quantify the problem of ancestral degeneracy.

Summary

- SMC is widely used, but doesn't perform well for smoothing.
- Analysis of SMC coalescents allows us to quantify the problem of ancestral degeneracy.
- Proved: SMC with multinomial resampling $\Rightarrow$ Kingman coalescent.
- (And simulations suggest it is a good model even for small N.)

Summary

- SMC is widely used, but doesn't perform well for smoothing.
- Analysis of SMC coalescents allows us to quantify the problem of ancestral degeneracy.
- Proved: SMC with multinomial resampling $\Rightarrow$ Kingman coalescent.
- (And simulations suggest it is a good model even for small N.)
- Proved: conditional SMC (with multinomial resampling) $\Rightarrow$ Kingman coalescent.
- In progress: investigate by simulations whether this is a good model for finite N.

Summary

- SMC is widely used, but doesn't perform well for smoothing.
- Analysis of SMC coalescents allows us to quantify the problem of ancestral degeneracy.
- Proved: SMC with multinomial resampling $\Rightarrow$ Kingman coalescent.
- (And simulations suggest it is a good model even for small N.)
- Proved: conditional SMC (with multinomial resampling) $\Rightarrow$ Kingman coalescent.
- In progress: investigate by simulations whether this is a good model for finite N.
- In progress: analysis of residual resampling (and other resampling schemes).

References I

[1] 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.

[2] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[3] Herbert E Rauch, CT Striebel, and F Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

[4] Andrew H Jazwinski. *Stochastic Processes and Filtering Theory*. Courier Corporation, 2007.

[5] Eric A Wan and Rudolph Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158. IEEE, 2000.

[6] Paolo Vidoni. Exponential family state space models based on a conjugate latent process. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):213–221, 1999.

References II

[7] John Kingman. The coalescent. *Stochastic Processes and Their Applications*, 13(3):235–248, 1982.

[8] Martin Möhle. Robustness results for the coalescent. *Journal of Applied Probability*, 35(2):438–447, 1998.

[9] Jere Koskela, Paul A Jenkins, Adam M Johansen, and Dario Spanò. Asymptotic genealogies of interacting particle systems with an application to sequential Monte Carlo. *arXiv preprint arXiv:1804.01811*, 2018.

[10] Jun S Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

[11] Randal Douc, Olivier Cappé, and Eric Moulines. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[12] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.