Sequential Monte Carlo methods
Lecture 16 – SMC samplers

Fredrik Lindsten, Linköping University
2019-08-30
Simulate a Markov chain which is designed in such a way that its stationary distribution coincides with the target distribution.

An MCMC sampler generates the Markov chain \( \{x[m]\}_{m=1}^M \) by:

- **Initialize:** set \( x[1] \) arbitrarily.
- **For** \( m = 2 \text{ to } M \): sample \( x[m] \sim \kappa(x[m-1], x^*) \).

\( \kappa(x, x^*) \) is a **Markov kernel** on \( \mathcal{X} \), i.e. a conditional distribution for the next state \( x^* \) given the current state \( x \).
Algorithm 1 Pseudo-marginal Metropolis Hastings

1. **Initialize** ($m = 1$): Set $\theta[1]$ and run a particle filter for $\hat{z}[1]$.

2. **For** $m = 2$ to $M$, **iterate**:
   a. Sample $\theta' \sim q(\theta | \theta[m - 1])$.
   b. Sample $\hat{z}' \sim \psi(z | \theta', y_{1:T})$ (i.e. run a particle filter).
   c. With probability
      \[
      \alpha = \min \left( 1, \frac{\hat{z}' p(\theta')}{{\hat{z}[m - 1]} p(\theta[m - 1]) \frac{q(\theta[m - 1] | \theta')}{q(\theta' | \theta[m - 1])}} \right)
      \]
      set $\{\theta[m], \hat{z}[m]\} \leftarrow \{\theta', \hat{z}'\}$ (accept candidate sample) and with prob. $1 - \alpha$ set $\{\theta[m], \hat{z}[m]\} \leftarrow \{\theta[m - 1], \hat{z}[m - 1]\}$ (reject candidate sample).
Particle Gibbs kernel: A Markov kernel $\kappa_{N, \theta}(x_0:T, x_0^*:T)$ on $\mathcal{X}^{T+1}$.

Particle Gibbs: Run a particle filter, but at each time step

- sample only $N - 1$ particles in the standard way.
- set the $N$th particle deterministically: $x_t^N = x_t$ and $a_t^N = N$.
- At final time $t = T$, output $x_0^*:T = x_0^b:T$ with $b \sim C(\{w_i\}_{i=1}^N)$

Ancestor sampling: Replace “set $a_t^N = N$” with: sample $a_t^N \in \{1, \ldots, N\}$ with

$$
P(A_t^N = j) \propto w_t^j p(x_t | x_{t-1}^j, \theta).$$

Can significantly improve mixing for small $N$/large $T$. 
SMC can be used to approximate a sequence of probability distributions \( \{\pi_k(x_{0:k})\}_{k \geq 0} \) on a sequence of probability spaces of increasing dimension, \( \mathcal{X}_{0:k} = \mathcal{X}_{0:k-1} \times \mathcal{X}_k \).

At iteration \( k \): given \( \{x_{0:k-1}^i, w_{k-1}^i\}_{i=1}^N \)

**Resampling:** Sample \( a_k^i \) with \( \mathbb{P}(a_k^i = j) = \nu_{k-1}^j, j = 1, \ldots, N \).

**Propagation:** \( x_k^i \sim q_k(x_k \mid x_{1:k-1}^{a_k^i}) \) and \( x_{0:k}^i = (x_{0:k-1}^{a_k^i}, x_k^i) \).

**Weighting:** \( w_k^i \propto \frac{w_{k-1}^{a_k^i}}{\nu_{k-1}^{a_k^i}} \frac{\tilde{\pi}_k(x_{0:k}^i)}{\tilde{\pi}_k(x_{0:k-1}^{a_k^i})q_k(x_k^i \mid x_{0:k-1}^{a_k^i})} \).
Aim: See how we can use SMC for inference even in the absence of any sequential structure in the model.

Outline:

1. Problem formulation
2. The annealing/tempering idea
3. Constructing the “SMC sampler”
4. User aspects
Let $\mathcal{X}$ be a space on which a probability density $\gamma$ is defined. Let $\tilde{\gamma}$ be an unnormalized version of the density, as $\gamma(x) = \frac{\tilde{\gamma}(x)}{Z}$. Assume that only $\tilde{\gamma}(x)$ can be evaluated pointwise.

**Goal:** Generate $N$ samples $x^i \in \mathcal{X}$ from the density $\gamma(x)$.

**ex)** Typical situation: $\gamma(x) = p(x|y)$, $\tilde{\gamma}(x) = p(x, y)$ and $Z = p(y)$. 

![Graph of $\gamma(x)$ vs $x$]
Most common solution

MCMC?

SMC sampler is an alternative!
Metropolis–Hastings targeting $\gamma$ (a reminder)

for $k = 1, 2, \ldots$

Propose a new sample $x'$ from a proposal $r(x' | x_k)$

Compute acceptance rate $\alpha = \min(1, \frac{\gamma(x') r(x_k | x')} {\gamma(x_k) r(x' | x_k)})$

Set $x_{k+1} \leftarrow x'$ with probability $\alpha$, otherwise set $x_{k+1} \leftarrow x_k$
Annealing/tempering

Sequential Monte Carlo needs something **sequential**. Construct a sequence which transitions ‘smoothly’ in $K$ steps from a simple initial $\gamma_0(x)$ to the sought $\gamma_K(x) \equiv \gamma(x)$.

**For example:**

- If $\gamma(x)$ is a posterior $\gamma(x) \propto p(y \mid x)p(x)$, then $\gamma_k(x) \propto p(y \mid x)^{\tau_k} p(x)$, $\tau_k = k/K$ (likelihood tempering)

- If $\gamma(x)$ depends on some data $y_{1:K}$ as $\gamma(x) = p(x \mid y_{1:K})$, then $\gamma_k(x) = p(x \mid y_{1:k})$ (data tempering)
**Intuition:** Track the evolving sequence $\gamma_0, \gamma_1, \ldots, \gamma_K$ using a weighting - resampling - propagation scheme.

**How do we do it?** This sequence (unlike the state inference problem) is not defined as a state-space model, neither does it fall into the general SMC formulation (yet).
Let's try to make use of the sequence $\gamma_0, \gamma_1, \ldots, \gamma_K$:

Sample $x^i$ from $\gamma_0$ and set $\tilde{w}_0(x^i) = 1$

for $k = 1$ to $K$

Evaluate $\tilde{w}_k^i(x^i) = \frac{\gamma_k(x^i)}{\gamma_{k-1}(x^i)} \tilde{w}_{k-1}^i$

Valid but inefficient: effectively importance sampling with proposal $\gamma_0$ and target $\gamma$. 

[Diagram showing the process with a red line indicating the importance sampling and a dotted line showing the target distribution.]
Sample $x_0^i$ from $\gamma_0$

for $k = 1$ to $K$

Use some Markov kernel $\kappa_k$ to sample new $x_k^i$ from $\kappa_k(x_{k-1}^i, x_k^i)$

Set weights $w_k^i \propto \frac{\gamma_k(x_k^i)}{\eta_k(x_k^i)}$ and normalize

where $\eta_k(x_k^i)$ is $\eta_k(x_k) = \int_{\mathcal{X}_k} \gamma_0(x_0) \prod_{j=1}^{k} \kappa_j(x_{j-1}, x_j) dx_{0:k-1}$.

In most cases intractable
Recall:

SMC can be used to approximate a sequence of probability distributions

\[ \{ \pi_k(x_{0:k}) \}_{k \geq 0} \]

don a sequence of probability spaces of increasing dimension,

\[ \mathcal{X}_{0:k} = \mathcal{X}_{0:k-1} \times \mathcal{X}_k. \]

- The intermediate target distributions can be chosen arbitrarily.
- We need to be able to recover the original distribution of interest (here \( \gamma_K(x) \equiv \gamma(x) \)) at iteration \( K \).
We have a sequence of distributions $\gamma_0(x), \gamma_1(x), \ldots, \gamma_K(x)$ evolving smoothly from something simple ($\gamma_0$) to the target distribution of interest ($\gamma_K$).

**Problem with directly applying SMC to this sequence:** The $\gamma_k$’s are all defined on the same space $\mathcal{X}$.

SMC requires spaces of **increasing dimension**.
Introduce a backward kernel $\lambda_{k-1}(x_k, x_{k-1})$, and define

\[
\begin{align*}
\pi_0(x_0) &= \gamma_0(x_0), \quad x_0 \in \mathcal{X} \\
\pi_1(x_{0:1}) &= \gamma_1(x_1)\lambda_0(x_1, x_0), \quad x_{0:1} \in \mathcal{X} \times \mathcal{X} = \mathcal{X}^2 \\
&\vdots \\
\pi_K(x_{0:K}) &= \gamma_K(x_K) \prod_{k=1}^{K} \lambda_{k-1}(x_k, x_{k-1}), \quad x_{0:K} \in \mathcal{X}^{K+1}
\end{align*}
\]

- $\gamma_k$ is defined on $\mathcal{X}$, whereas $\pi_k$ is defined on $\mathcal{X}^{k+1}$
- The marginal with index $k$ of $\pi_k(x_{0:k})$ is $\int \pi_k(x_{0:k}) dx_{0:k-1} = \gamma_k(x_k)$
- The marginal with index $j < k$ of $\pi_k(x_{0:k})$ is $\int \pi_k(x_{0:k}) dx_{0:j-1,j+1:k} \neq \gamma_j(x_j)$
We now have a sequence $\pi_0, \pi_1, \ldots, \pi_K$ (with marginals that we are interested in) defined on the spaces $\mathcal{X}, \mathcal{X}^2, \ldots, \mathcal{X}^{K+1}$.

Use the general SMC scheme!

Sample $x^i \sim \gamma_0(x_0)$ and set weights $w_0^i = 1/N$

for $k = 1$ to $K$

- If ESS too low, resample and set $w_k^i = 1/N$
- Use Markov kernel $\kappa_k$ to sample $x_k^i$
- Set weights $\tilde{w}_k^i = w_{k-1}^i \omega(x_{k-1:k})$ and normalize to $w_k^i$

Here, $\omega(x_{k-1:k}) = \frac{\gamma_k(x_k)\lambda_{k-1}(x_k, x_{k-1})}{\gamma_{k-1}(x_{k-1})\kappa_k(x_{k-1}, x_k)}$
How do we choose the backward kernels $\lambda_{k-1}$?

Assume that $\kappa_k$ is an **MCMC kernel** with stationary distribution $\gamma_k$. We can then select $\lambda_{k-1}$ as its **reversal**:

$$
\lambda_{k-1}(x_k, x_{k-1}) = \frac{\gamma_k(x_{k-1})\kappa_k(x_{k-1}, x_k)}{\gamma_k(x_k)}
$$

$$
\Rightarrow
$$

$$
\omega(x_{k-1:k}) = \frac{\gamma_k(x_k)}{\gamma_{k-1}(x_{k-1})} \frac{\gamma_k(x_{k-1})\kappa_k(x_{k-1}, x_k)}{\gamma_k(x_k)\kappa_k(x_{k-1}, x_k)} = \frac{\gamma_k(x_{k-1})}{\gamma_{k-1}(x_{k-1})}
$$
Design choices made: $\kappa_k$ and $\lambda_{k-1}$

The $\gamma_k$-invariant Markov kernel $\kappa_k$ is **one option** for propagating the samples, $q_k$ in the general SMC framework.

The backward kernel $\lambda_{k-1}$ is **part of the model specification** of $\pi_0, \ldots, \pi_K$, in the SMC context. (But since we are only interested in a marginal of $\pi_K$ not depending on $\lambda_{k-1}$, it may appear to be part of the inference algorithm rather than the model.)
The final SMC sampler

Sample $x_0^i$ from $\gamma_0$ and set weights $w_0^i = 1/N$

for $k = 1$ to $K$

Set weights $w_k^i \propto w_{k-1}^i \frac{\gamma_k(x_{k-1}^i)}{\gamma_{k-1}(x_{k-1}^i)}$ and normalize

If ESS too low, resample and set $w_k^i = 1/N$

Sample $x_k^i$ from MCMC kernel with stationary distribution $\gamma_k$. 

Estimating $Z$

For notational convenience, we have implicitly assumed we can evaluate $\gamma(x)$ exactly for any $x \in \mathcal{X}$. The SMC is also applicable if we only can evaluate $\tilde{\gamma}(x)$, where $\gamma(x) = \frac{\tilde{\gamma}(x)}{Z}$.

If $Z = Z_K$ is of interest, we can estimate $Z_K/Z_0$ as

$$
\frac{\hat{Z}_K}{Z_0} = \prod_{k=1}^{K} \frac{\hat{Z}_{k-1}}{Z_k}
$$

where

$$
\frac{\hat{Z}_{k-1}}{Z_k} = \sum_{i=1}^{N} w_{k-1}^{i} \frac{\gamma_k(x_{k-1}^{i})}{\gamma_{k-1}(x_{k-1}^{i})}
$$

and $Z_0$ is the normalizing constant of the user-chosen $\gamma_0$.

(Superior to annealed importance sampling.)
SMC sampler vs MCMC

### MCMC (Metropolis-Hastings)

Set initial $x_0$

**for** $k = 1, \ldots$

- Propose a new sample $x'$ from $r(x' \mid x_k)$
- Compute $\alpha = \min(1, \frac{\gamma(x')}{{\gamma(x_k)} r(x' \mid x_k)})$
- Set $x_{k+1} \leftarrow x'$ with probability $\alpha$, otherwise $x_{k+1} \leftarrow x_k$

**end**

### SMC sampler

Sample $x_0^i$ from $\gamma_0$ and set weights $w_0^i = 1/N$

**for** $k = 1$ to $K$

- Set $\tilde{w}_k^i = w_{k-1}^i \frac{\gamma_k(x_k^i)}{\gamma_k(x_k^i) r(x_k^i \mid x_k)}$ and normalize
- If ESS too low, resample and set $w_k^i = 1/N$

Sample $x_k^i$ by $\gamma_k$-invariant Metropolis-Hastings

**end**
1. Design a simulated annealing sequence (e.g., likelihood or data tempering)
2. Design MCMC kernel $\kappa_k$ (typically Metropolis-Hastings) for $\gamma_k$
3. Design backward kernel $\lambda_{k-1}$. Simplest choice is as the reversal of $\kappa_k$, but other options are available.
4. Run the SMC sampler!
Automatic adaptation

Adaptation of the MCMC kernels:


Adaptation of the tempering sequence:

Some further developments

Approximate Bayesian computations (ABC):


Use SMC sampler for unknown parameters in a state-space model:

A few concepts to summarize lecture 16

- SMC sampler is an alternative to MCMC
- The simulated annealing sequence is key
- The formal construction is made possible by the use of backward kernels $\lambda_k$