Sequential Monte Carlo Methods

Seong-Hwan Jun
EECS KTH Royal Institute of Technology
SciLifeLab

November 28, 2018

Importance Sampling

Bayesian Inference

Posterior distribution of $x \in \mathcal{X}$ given we observe $y \in \mathcal{Y}$:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

- $p(y|x)$: Data likelihood.
- $p(x)$: Prior distribution.
- $p(y)$: Marginal likelihood.

Goal

Compute expectation of a function h w.r.t. posterior distribution:

$$I = \mathbb{E}_{X \sim p(x|y)}[h(X)] = \int h(x)p(x|y)dx,$$

where h is a known function that we can evaluate.

Example 1: $h(x) = x$,

$$\mathbb{E}_{X \sim p(x|y)}[X] = \int x p(x|y) dx.$$

Example 2: $h(x) = 1[x > a]$ for $x, a \in \mathbb{R}$,

$$\mathbb{E}_{X \sim p(x|y)}[1[X > a]] = \mathbb{P}(X > a|y) = \int 1[x > a] p(x|y) dx.$$

Solution I: Monte Carlo

If $p(x|y)$ is a known distribution such as Normal distribution, we can use Monte carlo approximation to compute the expectation no matter how complex h is:

$$\frac{1}{K}\sum_{k=1}^{K} h(x_k) \to \mathbb{E}_{X\sim p(x|y)}[h(X)] \text{ where } x_k \sim p(x|y) \text{ as } K \to \infty,$$

by the Law of Large Numbers (LLN).

Reality...

- In many cases, integral $p(y) = \int p(x,y)dx$ is not analytically available.
- Hence, $p(x|y)$ is not known in practice.

An important exception is if the likelihood and prior are *conjugate distributions*. In this case, $p(x|y)$ is known and can be sampled from.

Solution II: Importance Sampling

- Find a distribution q that is easy to sample from and $q(x) > 0$ whenever $p(x|y) > 0$.
- Propose $x_k \sim q(x)$ for $k = 1, ..., K$.
- Let $w(x) = p(x|y)/q(x)$. Then,

$$\frac{1}{K}\sum_{k=1}^{K} w(x_k)h(x_k) \to I,$$

by LLN.

Derivation

$$
\begin{aligned}
I &= \int h(x)p(x|y)dx \\
&= \int h(x)\frac{p(x|y)}{q(x)}q(x)dx \\
&= \int h(x)w(x)q(x)dx \\
&= \mathbb{E}_{X\sim q}[h(X)w(X)].
\end{aligned}
$$

Therefore, importance sampling is a Monte Carlo algorithm that approximates the expectation w.r.t to q with test function $h'(x) = h(x)w(x)$.

Self Normalization

Problem: We assumed that $p(x|y)$ can be evaluated. However, $p(y)$ is intractable in practice and hence $p(x|y)$ cannot be evaluated.

- Let $\gamma(x) = p(x, y) = p(y|x)p(x)$ and $Z = p(y)$.
- Let $w(x) = \gamma(x)/Zq(x)$.
- Let $\bar{w}_k = w(x_k)/\sum_j w(x_j)$. Then,

$$\sum_{k=1}^{K} \bar{w}_k h(x_k) \to I.$$

Proof Sketch

$$
\begin{aligned}
\sum_{k=1}^{K} \bar{w}_k h(x_k) &= \sum_{k=1}^{K} \frac{w(x_k)h(x_k)}{\sum_j w(x_j)} \\
&= \sum_{k=1}^{K} h(x_k) \frac{\gamma(x_k)/Zq(x_k)}{\sum_j \gamma(x_j)/Zq(x_j)} \\
&= \frac{K^{-1} \sum_{k=1}^{K} h(x_k) \frac{\gamma(x_k)}{q(x_k)}}{K^{-1} \sum_{j=1}^{K} \frac{\gamma(x_j)}{q(x_j)}}.
\end{aligned}
$$

By LLN, the numerator converges to $\int h(x)p(x,y)dx$.

$$K^{-1}\sum_{k=1}^{K} h(x_k)\frac{\gamma(x_k)}{q(x_k)} \to \mathbb{E}_{X\sim q}\left[\frac{h(X)\gamma(X)}{q(X)}\right].$$

$$\begin{aligned}
\text{RHS} &= \int h(x)\frac{\gamma(x)}{q(x)}q(x)dx \\
&= \int h(x)\gamma(x)dx \\
&= \int h(x)p(x,y)dx.
\end{aligned}$$

Denominator converges to $Z = p(y)$.

$$K^{-1} \sum_{j=1}^{K} \frac{\gamma(x_j)}{q(x_j)} \to \mathbb{E}_{X \sim q}\left[\frac{\gamma(X)}{q(X)}\right].$$

$$\begin{aligned}
\text{RHS} &= \int \frac{\gamma(x)}{q(x)} q(x) dx \\
&= \int p(x, y) dx \\
&= p(y).
\end{aligned}$$

Therefore,

$$\sum_{k=1}^{K} \bar{w}_k h(x_k) \to \frac{\int h(x)p(x,y)dx}{p(y)} = \int h(x)p(x|y)dx.$$

Note: Formal proof would require us to use continuous mapping theorem. See [Durrett, 2010, Thm 3.2.4].

Example: Small Tail Probabilities

From [Robert and Casella, 2013, Example 3.11]. Importance sampling can be useful in many settings beyond Bayesian statistics.

Let $Z \sim N(0,1)$. Estimate $\mathbb{P}(Z > 4.5)$.

- Solution 1: $Z_k \sim N(0,1)$. Compute

$$\frac{1}{K}\sum_k 1[z_k > 4.5].$$

- Solution 2: $X_k \sim q = \text{Exponential}(0.1)$. Compute

$$\frac{1}{K}\sum_k 1[x_k > 4.5]\frac{\phi(x_k)}{q(x_k)}.$$

Note: We need to choose q to cover the region of interest (i.e., x for which $h(x)p(x) > 0$).

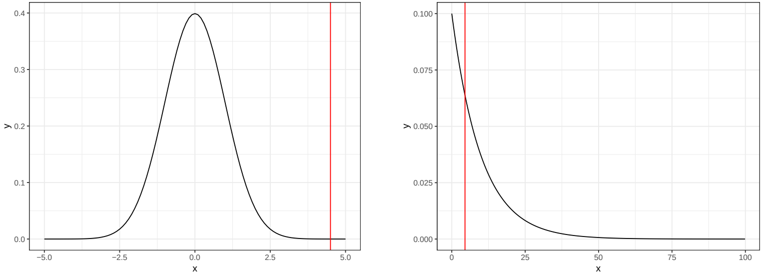


Figure: Left: Standard normal distribution. Right: Exp(0.1). Red vertical line is the threshold, $a = 4.5$.

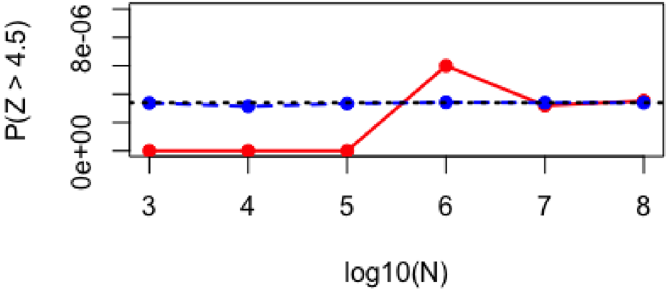


Figure: Monte Carlo stimate of $P(X > 4.5)$. Red: Simple Monte Carlo sampling. Blue dashed: Importance sampling. Dotted black: truth.

Brief Summary

- Monte Carlo sampling can be used to approximate complex integral numerically.
- IS can improve efficiency in terms of number of samples required.
- IS can be useful with just a simple q even when direct sampling is not possible.
- IS yields estimate of the marginal likelihood: $Z = p(y)$.
- Remember to choose q such that $q(x) > 0$ whenever $p(x) > 0$ (or $h(x)p(x) > 0$).

Sequential IS

Sequential Importance Sampling

- Now, let $\boldsymbol{x} = (x_1, ..., x_d)$ be a d-dimensional vector.
- Goal: Compute

$$I = \int h(x_1, ..., x_d) p(x_1, ..., x_d | \boldsymbol{y}) dx_1, ..., dx_d.$$

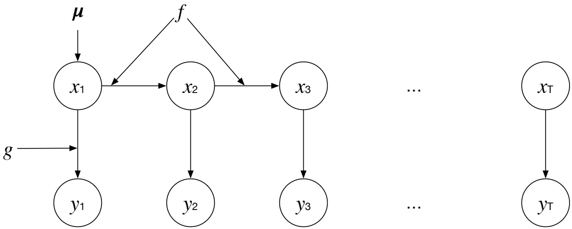
- Solution I: Importance sampling. We need to find a multivariate proposal distribution that is easy to sample from.
- If $x_i \in \mathbb{R}$, we may be able to use *multivariate Normal* distribution. But for general setting, finding q may be difficult.
- Also, curse of dimensionality: number of samples needed to sufficiently approximate the integral grows exponentially with dimension.

Idea: Propose one dimension at a time from $x_i \sim q_i$ for $i = 1, ..., d$.

Proposal:	$x_i^k \sim q_i$
Sample extenion:	$\boldsymbol{x}_{1:i}^k = (x_1^k, ..., x_{i-1}^k, x_i^k)$
Weight computation:	$w(x_i^k) = \frac{\gamma_i(x_{1:i}^k)}{\nu_i(x_{1:i}^k)}$,

where $\nu_i(x_{1:i}) = \prod_{j=1}^{i} q_j(x_j|x_{1:j-1})$ and $x_{i:j} = (x_i, ..., x_j)$ for $0 < i < j$.

Application: Hidden Markov Model



$$x_1 \sim \mu(x_1)$$
$$x_t|x_{t-1} \sim f(x_t|x_{t-1}) \text{ for } t = 2, ..., T$$
$$y_t|x_t \sim g(y_t|x_t) \text{ for } t = 1, ..., T.$$

- $p(\boldsymbol{x}) = \mu(x_1) \prod_{t=2}^{T} f(x_t | x_{t-1})$
- $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{T} g(y_t | x_t)$
- $\gamma(\boldsymbol{x}) = p(\boldsymbol{x}, \boldsymbol{y}) = \mu(x_1) g(y_1|x_1) \prod_{t=2}^{T} f(x_t|x_{t-1}) g(y_t|x_t)$.
- $Z = p(\boldsymbol{y}) = \int p(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x}$

Recursive weight Update

$$
\begin{aligned}
w(x_{1:t}) &= \frac{\gamma_t(x_{1:t})}{\nu_t(x_{1:t})} \\
&= \frac{\gamma_{t-1}(x_{1:t-1})}{\nu_{t-1}(x_{1:t-1})} \frac{f(x_t|x_{t-1})g(y_t|x_t)}{q_t(x_t|x_{1:t-1})} \\
&= w(x_{1:t-1})\alpha(x_{1:t-1}, x_t).
\end{aligned}
$$

Therefore, we should store the weight from previous iteration and compute only the weight update function $\alpha(x_{1:t-1}, x_t)$ at current iteration.

Proposal

- Prior: $q_t = f(x_t|x_{t-1})$.
  - Weight function: $\alpha(x_{1:t-1}, x_t) = g(y_t|x_t)$.
  - Pro: Simplicity.
  - Con: May require large number of samples if $f(x_t|x_{t-1})$ differs significantly from $p(x_t|x_{1:t-1}, y_t)$.

- Adapted: $q_t = p(x_t|x_{1:t-1}, y_t)$

$$p(x_t|x_{1:t-1}, y_t) = \frac{p(x_t, y_t|x_{1:t-1})}{p(y_t|x_{1:t-1})} = \frac{g(y_t|x_t)f(x_t|x_{t-1})}{\int g(y_t|x_t)f(x_t|x_{t-1})dx_t}.$$

  - Weight update function: $p(y_t|x_{1:t-1})^{-1}$.
  - Pro: Makes use of the latest observation to build a smart proposal. Generally requires less number of samples compared to prior (for example, to attain similar accuracy of approximation).
  - Con: Need to analytically compute $p(y_t|x_{1:t-1})$.

Example: Stochastic Volatility Model

$$
\begin{aligned}
X_1 &\sim \mathcal{N}(x_1|0,\sigma^2) \\
X_t|(X_{t-1}=x_{t-1}) &\sim \mathcal{N}(x_t|\phi x_{t-1},\sigma^2), && t=2,\dots,T, \\
Y_t|(X_t=x_t) &\sim \mathcal{N}(y_t|0,\beta^2\exp(x_t)), && t=2,\dots,T.
\end{aligned}
$$

- X_t: Unobserved volatility of an asset (e.g., stock price).
- Y_t: Observed change in the price of the asset.

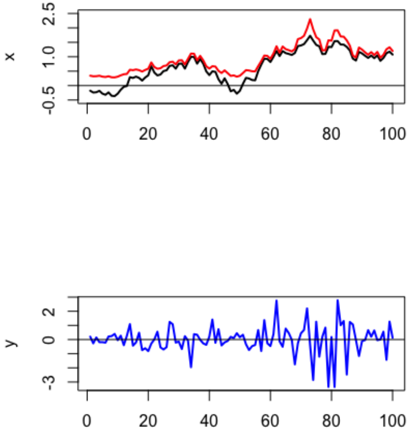
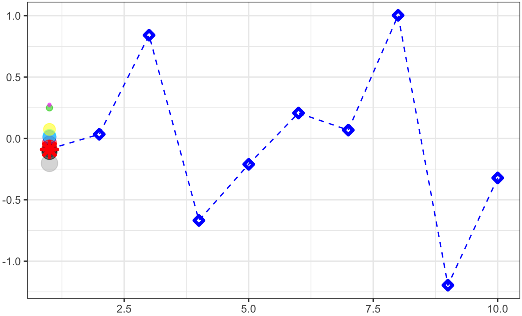
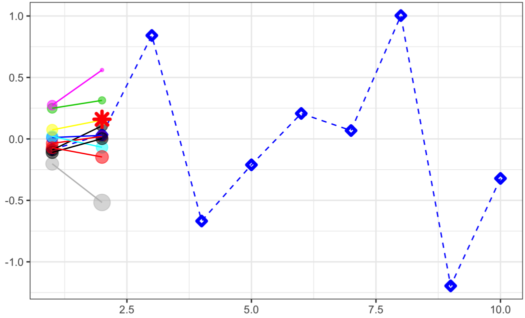
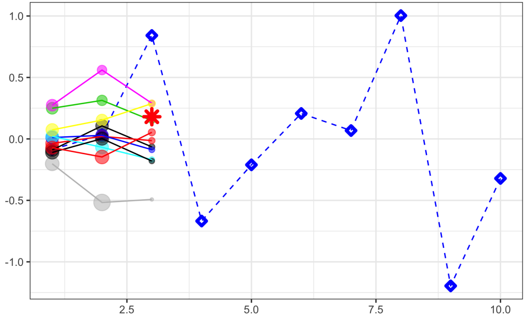


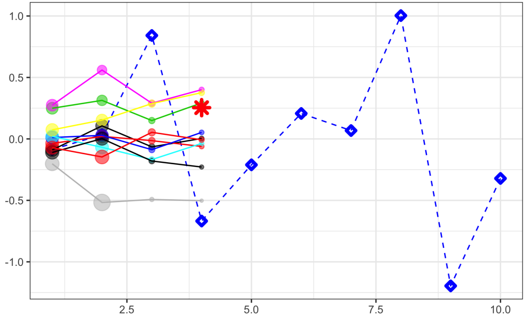
Figure: Top: X_t in black and variance of the observation i.e., $\beta^2 exp(x_t)$ in red. Bottom: Observation.

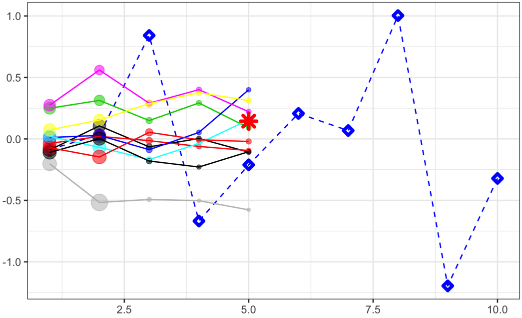
Illustration of SIS



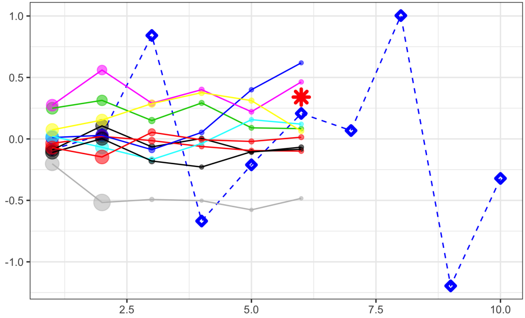

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

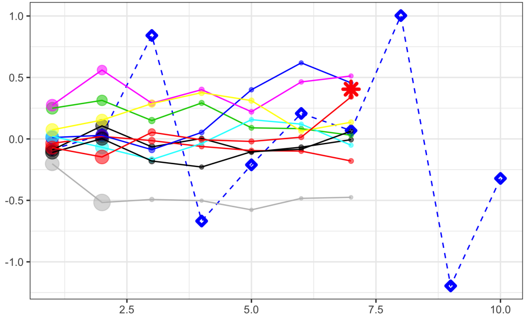

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

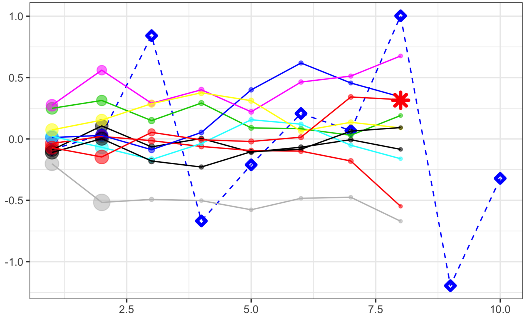

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

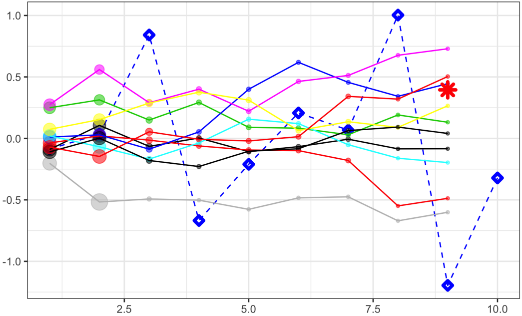


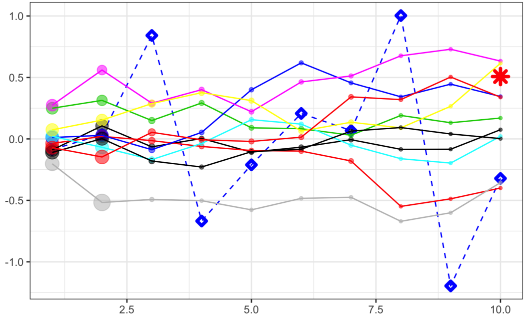
1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0



1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

Brief Summary

- SIS can be useful in settings where we need to approximate high dimensional integral.
- Particularly useful if the model exhibits a temporal structure.
- Weights decay with T. For large T, SIS usually does not work well (contradictory to the first point).
- Only a handful of samples become relevant as T increases, leading to waste of computational resources.

SMC

SIS with Resampling

- Idea: Interleave resampling step to choose promising particles.
- Use the weights to select the particles.
- Sequential Monte Carlo methods refer to a class of algorithms that involve sequential proposal, weight computation, followed by (optional) resampling.
- Best tutorial to get started in SMC (in my opinion): [Doucet and Johansen, 2009].

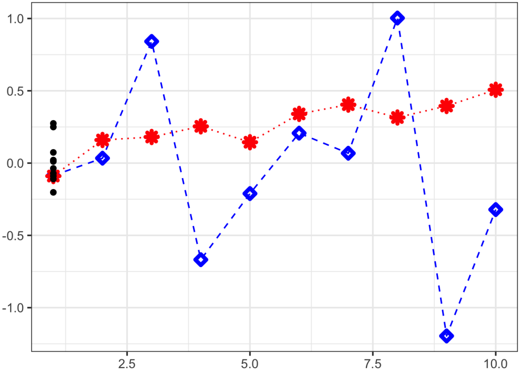
$t = 1$:

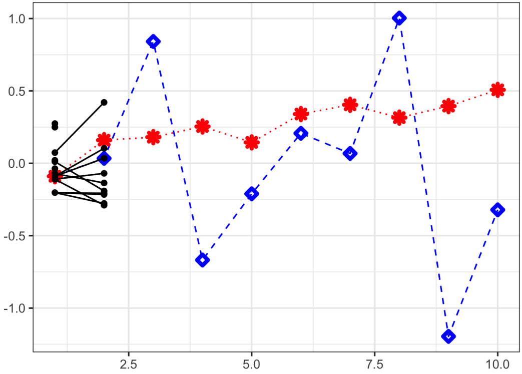
- Proposal: $x_1^k \sim q_1(x_1)$.
- Weight computation: $w(x_1^k) = \alpha(x_1^k)$.
- Weight normalization: $\bar{w}_1^k = w(x_1^k) / \sum_{k'} w(x_1^{k'})$.

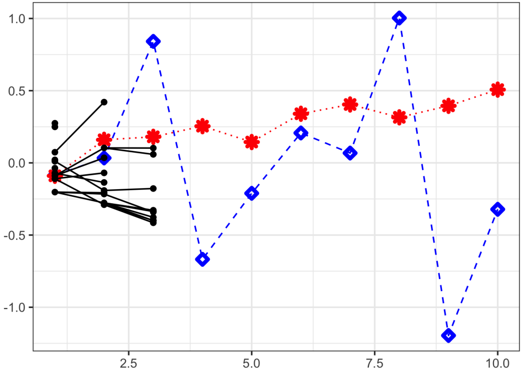
$t \geq 2$:

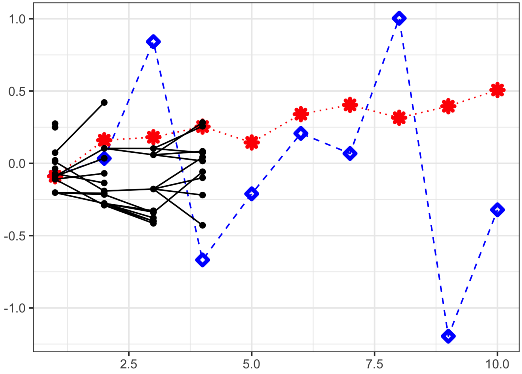
- Resampling: $j \sim \text{Multinomial}(\bar{w}_{t-1}^1, ..., \bar{w}_{t-1}^K)$.
- Proposal: $x_t^k \sim q_t(x_t | x_{1:t-1}^j)$.
- Extension: $\boldsymbol{x}^k = (x_{1:t-1}^j, x_t^k)$.
- Weight computation: $w(x_{1:t}^k) = \alpha(x_{1:t-1}^j, x_t^k)$.
- Normalize the weights: $\bar{w}_t^k = w(x_{1:t}^k) / \sum_{k'} w(x_{1:t}^{k'})$.

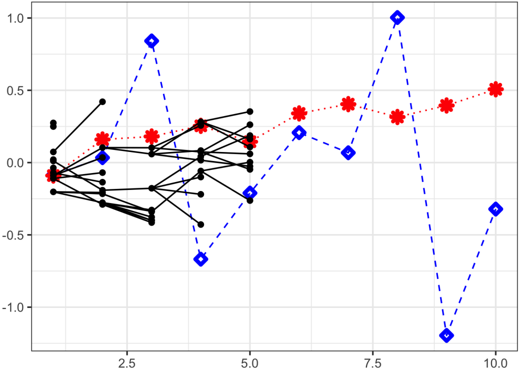
Illustration of SMC on SV Model

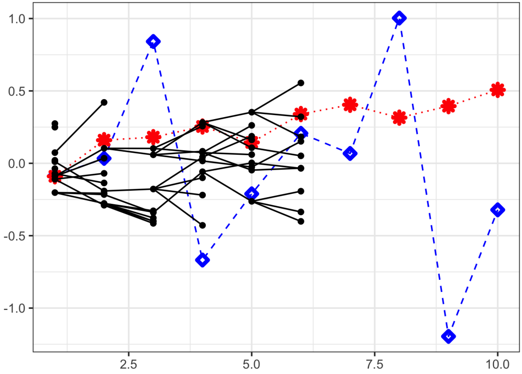


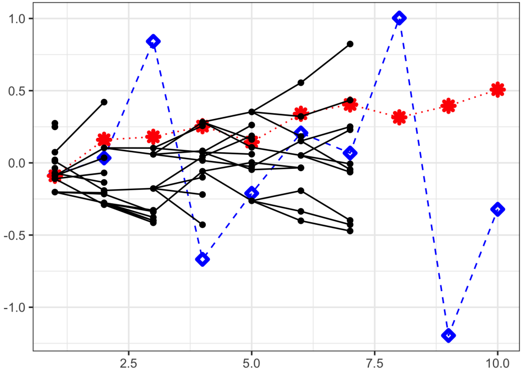

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

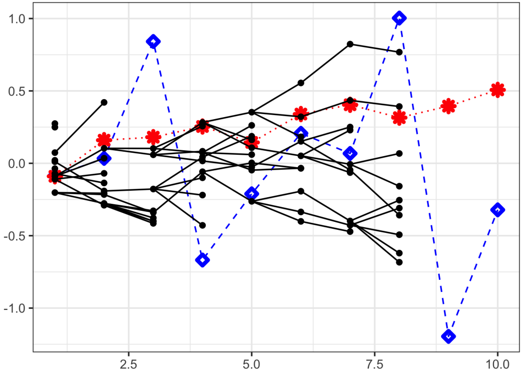

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

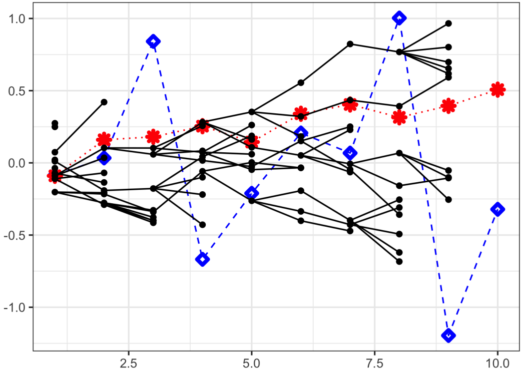

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

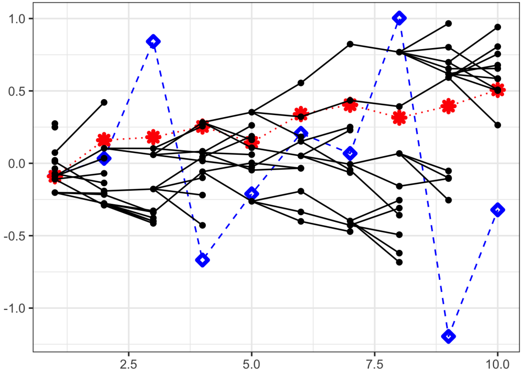

1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0



1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0


1.0
0.5
0.0
-0.5
-1.0
2.5
5.0
7.5
10.0

Filtering

- Samples and the weights can be used to approximate the *filtering* distribution:

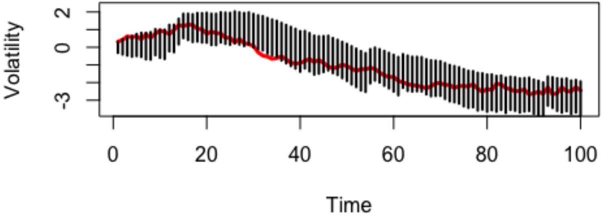
$$\hat{p}(x_t|y_{1:t}) = \sum_{k=1}^{K} \bar{w}_t^k \delta_{x_t^k}(x_t) \text{ for } t = 1, ..., T.$$

or after resampling:

$$\hat{p}(x_t|y_{1:t}) = \frac{1}{K} \sum_{k=1}^{K} \delta_{x_t^k}(x_t) \text{ for } t = 1, ..., T.$$

Effectiveness of SMC on SV Model

Ran with $10,000$ particles. Computed empirical 95% confidence interval. Contains the true x_t about 93% of the time.



Predictive Distribution

- The generated samples can be used to build a predictive distribution:

$$p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(x_t|y_{1:t})dx_t.$$

Therefore, take the test function $h(x_{t+1}) = p(x_{t+1}|x_t)$ (e.g., $p(x_{t+1}|x_t) = f(x_{t+1}|x_t)$ in HMM application) and,

$$\hat{p}(x_{t+1}|y_{1:t}) = \sum_{k=1}^{K} p(x_{t+1})\bar{w}_t^k \delta_{x_t^k}(x_t) \text{ for } t = 1, ..., T.$$

Applications

- Online estimation: as the observation arrives, infer the latent state.
  - E.g., fraud detection, missile tracking, robot localization, etc.
- An extension of SMC [Del Moral et al., 2006], can be used in problems that do not exhibit temporal structure.
  - Phylogenetic inference [Bouchard-Côté et al., 2012].
  - Graph matching [Jun et al., 2017].
- Inference over graphical models [Naesseth et al., 2014].
- Probabilistic programming [Murray et al., 2017].

Resampling Algorithms

Can reduce variance of the estimator by using better resampling algorithms [Douc and Cappé, 2005]:

- Stratified Resampling.
- Residual Resampling.
- Implementing stratified resampling is a homework question :)

PMCMC

Particle MCMC

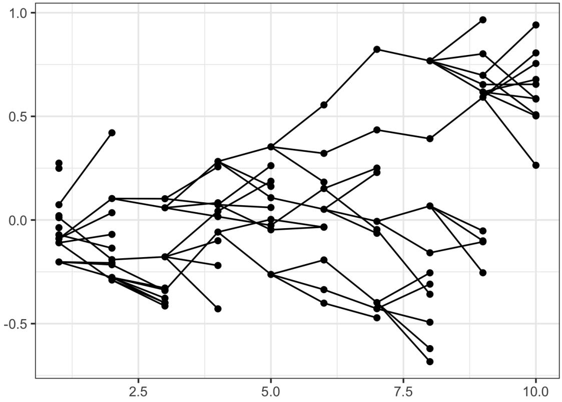
PMCMC is proposed in [Andrieu et al., 2010]. A very important development in computational statistics in recent years (recommend to read up to section 2 – to solve homework questions).

- Goal: Jointly infer the model parameters θ and the latent variables $\boldsymbol{x}$.
- Basic Idea: Construct Metropolis-Hastings or Gibbs sampler on $(\theta^n, \boldsymbol{x}^n)$.

Particle Genealogy

- SMC through resampling induces genealogy of particles.
- Let a_t^k denote the index of the ancestor of particle k (for $t = 1, ..., T-1$).
- Let b_t^k denote the index of the k-th particle genealogy at generation t.
  - $b_T^k = k$.
  - If $b_1^k = i$, then if you trace k-th genealogy all the way up to the first iteration, its index among that population is i.
  - If $b_t^k = i$, then if you trace k-th genealogy all the way up to the t-th iteration of SMC, its index is i.
  - Note: $b_t^k = a_t^{b_{t+1}^k}$.

Illustration of Particle Genealogy



Particle Marginal Metropolis Hastings

Initialization step:

- Initialize $\theta(0)$ arbitrarily.
- $\{Z^*, \bar{w}^k, \boldsymbol{x}^k\}_{k=1}^K \leftarrow \text{SMC}(\theta(0))$.
- Sample $k' \sim \bar{w}^k$.
- Set $\boldsymbol{x}(0) = \boldsymbol{x}^{k'}$.
- Set $\hat{Z}(0) = Z^*$.

For $n = 1, ..., \text{numIter}$:

- Propose $\theta^* | \theta(n-1) \sim q_\theta(\cdot | \theta(n-1))$.
- $\{\hat{Z}^*, \bar{w}^k, \boldsymbol{x}^k\}_{k=1}^K \leftarrow \text{SMC}(\theta^*)$.
- Sample $k' \sim \bar{w}^k$.
- Compute MH acceptance ratio:

$$\alpha = \min\left\{1, \frac{\hat{Z}^*}{\hat{Z}(n-1)} \frac{p(\theta^*)}{p(\theta(n-1))} \frac{q_\theta(\theta(n-1)|\theta^*)}{q_\theta(\theta^*|\theta(n-1))}\right\}$$

- With probability α, set

$$(\theta(n), \boldsymbol{x}(n), \hat{Z}(n)) = (\theta^*, \boldsymbol{x}^{k'}, \hat{Z}^*).$$

- Otherwise, set

$$(\theta(n), \boldsymbol{x}(n), \hat{Z}(n)) = (\theta(n-1), \boldsymbol{x}(n-1), \hat{Z}(n-1)).$$

Why does PMMH work?

- PMMH is a Metropolis-Hastings algorithm on the extended space: $\Theta \times \mathcal{X}_{1:T}$ with $\theta \in \Theta, x_{1:T} \in \mathcal{X}_{1:T}$.
- Posterior distribution: $p(\theta, x_{1:T}|y) = \gamma(\theta, x_{1:T})/Z(\theta)$, where $Z(\theta) = p(y|\theta)$.
- SMC can be viewed as constructing an efficient proposal for $x_{1:T}$.
- Accept-reject is used to move the state of the Markov chain.

Derivation

- Proposal on θ^*: $q_\theta(\cdot|\theta_{n-1})$.
- Proposal of SMC: $q_t(x_t|x_{1:t-1}, \theta)$.
- Resampling distribution: $r(a_t^k|\bar{\boldsymbol{w}}_{t-1})$.
  - Important condition: $P(a_t^k = k|\bar{\boldsymbol{w}}_{t-1}) = \bar{w}_{t-1}^k$. In other words, resampling should be unbiased.
- There are other regularity conditions, which are rather mild and can be satified with relative ease in most applications. See [Andrieu et al., 2010] for more details.

- Distribution of all variables generated by SMC given θ:

$$\psi(x_{1:T}^{1:K}, a_{1:T-1}^{1:K}|\theta) = \prod_{k=1}^{K} q_1(x_1^k|\theta) \prod_{t=2}^{T} \prod_{k=1}^{K} r(a_t^k|\bar{\boldsymbol{w}}_{t-1}) q_t(x_t^k|x_{1:t-1}^{a_{t-1}^k}, \theta).$$

- Proposal distribution on the extended space:

$$q(\theta^*, k, x_{1:T}^{1:K}, a_{1:T-1}^{1:K}|\theta_{n-1}) = q_\theta(\theta^*|\theta_{n-1}) \bar{w}_T^k \psi(x_{1:T}^{1:K}, a_{1:T-1}^{1:K}|\theta^*).$$

- Target distribution of MH algorithm:

$$\tilde{p}(\theta, k, x_{1:T}^{1:K}, a_{1:T}^{1:K-1}) = \frac{p(\theta, x_{1:T}^k)}{K^T} \frac{\psi(x_{1:T}^{1:K}, a_{1:T-1}^{1:K}|\theta)}{q_1(x_1^{b_1^k}|\theta) \prod_{t=2}^{T} r(b_{t-1}^k|\bar{\boldsymbol{w}}_{t-1}) q_t(x_t^{b_t^k}|x_{1:t-1}^{b_{t-1}^k}, \theta)}$$

Derivation continued

$$
\begin{aligned}
\frac{\tilde{p}(\theta^*, k, x_{1:T}^{1:K}, a_{1:T}^{1:K-1})}{q(\theta^*, k, x_{1:T}^{1:K}, a_{1:T}^{1:K-1}|\theta_{n-1})} &= \frac{K^{-T}p(\theta^*, x_{1:T}^k)}{q_\theta(\theta^*|\theta_{n-1})\bar{w}_T^k q_1(x_1^{b_1^k}|\theta^*)\prod_{t=2}^T r(b_{t-1}^k|\bar{\boldsymbol{w}}_{t-1})q_t(x_t^{b_t^k}|x_{1:t-1}^{b_{t-1}^k}, \theta^*)} \\
&= \frac{p(\theta^*)}{q(\theta^*|\theta_{n-1})} \frac{p(x_{1:T}^k|\theta^*)}{q_1(x_1^{b_1^k})\prod_{t=2}^T q_t(x_t^{b_t^k}|x_{1:t-1}^{b_{t-1}^k})} \frac{K^{-T}}{\prod_{t=1}^T \bar{w}_t^{b_t^k}} \\
&= \frac{p(\theta^*)}{q(\theta^*|\theta_{n-1})} \frac{p(x_{1:T}^k|\theta^*)}{q_1(x_1^{b_1^k})\prod_{t=2}^T q_t(x_t^{b_t^k}|x_{1:t-1}^{b_{t-1}^k})} \frac{K^{-T}\prod_{t=1}^T \sum_j w_t^j}{\prod_{t=1}^T w_t^{b_t^k}} \\
&= \frac{p(\theta^*)}{q(\theta^*|\theta_{n-1})} \frac{p(x_{1:T}^k|\theta^*)}{q_1(x_1^{b_1^k})\prod_{t=2}^T q_t(x_t^{b_t^k}|x_{1:t-1}^{b_{t-1}^k})} \frac{\hat{Z}}{\prod_{t=1}^T w_t^{b_t^k}} \qquad (1)
\end{aligned}
$$

Note:

$$q_1(x_1^{b_1^k}|\theta^*) \prod_{t=2}^{T} q_t(x_t^{b_t^k}|x_{1:t-1}^{b_{t-1}^k}, \theta^*) \times \prod_{t=1}^{T} w_t^{b_t^k} = q(x_{1:T}^k|\theta^*) \times \frac{\gamma(x_{1:t}^k|\theta^*)}{q(x_{1:T}^k|\theta^*)}$$
$$= Zp(x_{1:T}^k|\theta^*).$$

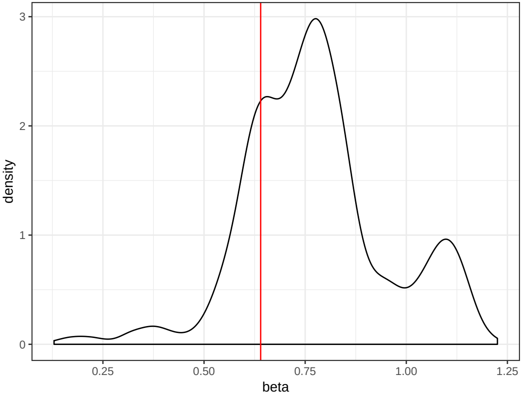
Therefore,

$$\text{Eq (1)} = \frac{p(\theta^*)}{q(\theta^*|\theta_{n-1})} \frac{\hat{Z}(\theta^*)}{Z}.$$

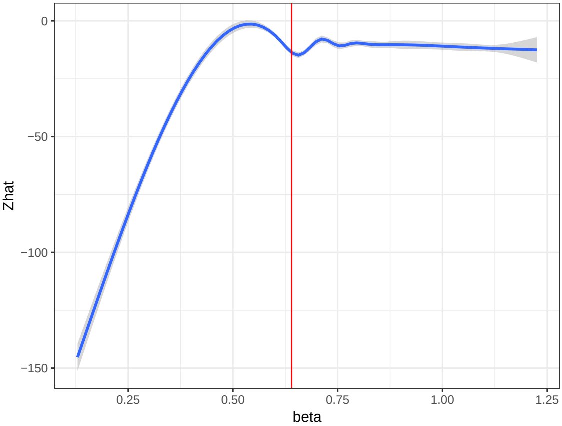
The acceptance ratio arises by carrying out symmetric derivation on $q/\tilde{p}$ followed by multiplication of the resulting quantity with Eq (1).

Example: Stochastic volatility

Assume $\phi = 1, \sigma = 0.16$. Suppose β is unknown. We can perform PMMH to sample β.
Chose $p(\beta) = \text{Uniform}(0, 2)$ and
$q_\beta(\cdot|\beta(n-1)) = N(\beta(n-1), 0.02)$. Ran for 2,000 iterations using 200 particles for SMC.



Plot of $\beta(n)$ vs $\hat{Z}(n-1)$ (smoothed). Red line indicates the value of β used to generate the data.



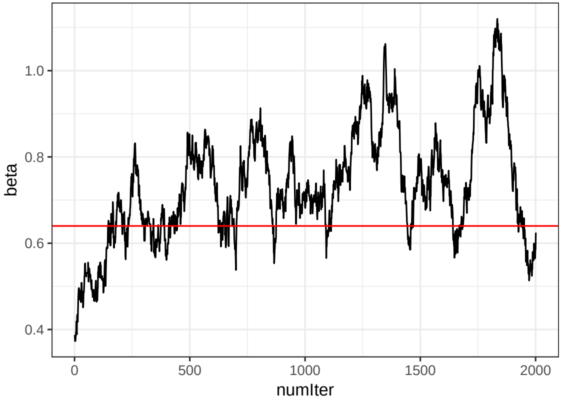


Figure: The trace plot of β sampled using PMMH.

Condtional SMC

- Let $(x^k_{1:T}, b^k_{1:T})$ be a genealogy of a particle. Allow it to survive resampling steps for each $t = 1, ..., T-1$.
- Perform SMC as usual for $k' \neq k$.
- This procedure can be viewed as sampling auxiliary variables $(x^{-k}_{1:T}, b^{-k}_{1:T})$ given $(x^k_{1:T}, b^k_{1:T})$ from the following density:

$$p(x^{-k}_{1:T}, a^{-k}_{1:T-1} | x^k_{1:T}, b^k_{1:T}) = \frac{\psi(x^{1:K}_{1:T}, a^{1:K}_{1:T-1})}{q_1(x_1^{b_1^k}) \prod_{t=2}^{T} r(b^k_{t-1} | \bar{\boldsymbol{w}}_{t-1}) q_t(x_t^{b_t^k} | x_{t-1}^{b_{t-1}^k})}.$$

Illustration of CSMC

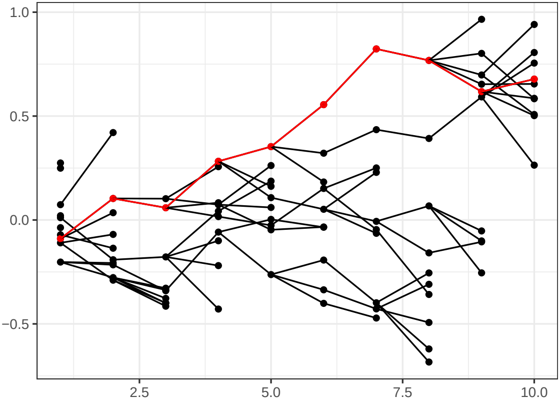


Figure: Lineage of $k = 7$-th particle is fixed throughout the iterations of conditional SMC.

Particle Gibbs

If we can derive closed form distribution for $\theta(n)|x_{1:T}$, then we can alternate sampling of model parameters using Gibbs sampling and latent variables using conditional SMC.

- Initialize $\theta(0), x_{1:T}(0), b_{1:T}(0)$.
- Sample $\theta(n)|x_{1:T}(n-1) \sim p(\theta|x^k_{1:T})$
- Sample $(x^{-k}_{1:T}, b^{-k}_{1:T})|x_{1:T}(n-1), b_{1:T}(n-1), \theta(n) \sim$ $\mathrm{CSMC}(x_{1:T}(n-1), b_{1:T-1}(n-1), \theta(n))$.
- Sample $k \sim r(\cdot|\bar{\boldsymbol{w}})$.

Implementation note: conditional SMC can deterministically set $a^1_{1:T-1} = (1, ..., 1)$ and $x^1_{1:T} = x_{1:T}(n-1)$ and sample for $2:N$. See [Chopin et al., 2015, Remark 1].

Stochastic Volatility Model for β

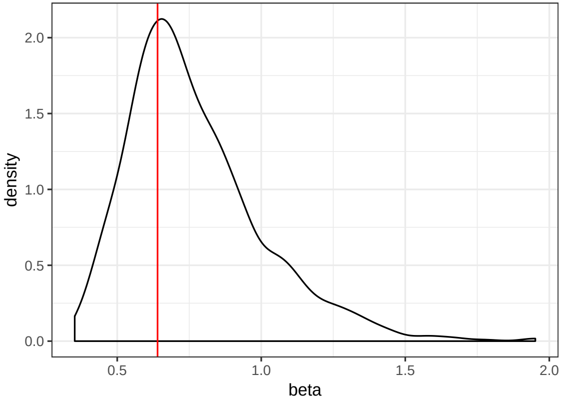


Figure: Plot of posterior distribution over β obtained by running PG for 1,000 iterations with 200 particles.

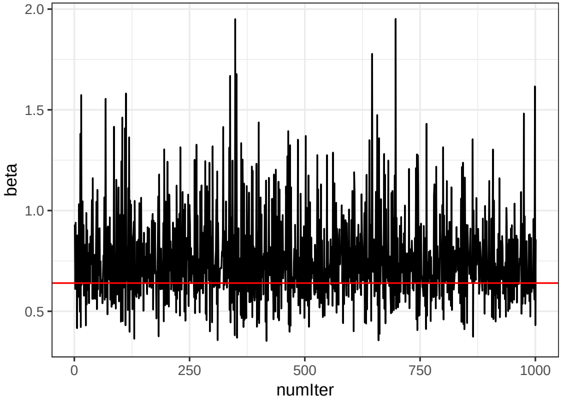


Figure: Trace plot of β sampled using PG.

Useful Resources

- A course given by Arnaud Doucet: `http://www.stats.ox.ac.uk/~doucet/samsi_course.html`.
- A bit more recent course given by SMC group in Uppsala: `http://www.it.uu.se/research/systems_and_control/education/2017/smc/`.

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, 2010.

A. Bouchard-Côté, S. Sankararaman, and M. I. Jordan. Phylogenetic inference via sequential monte carlo. *Systematic biology*, 61(4):579–593, 2012.

N. Chopin, S. S. Singh, et al. On particle gibbs sampling. *Bernoulli*, 21(3):1855–1883, 2015.

P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

R. Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

S.-H. Jun, S. W. Wong, J. Zidek, and A. Bouchard-Côté. Sequential graph matching with sequential monte carlo. In *Artificial Intelligence and Statistics*, pages 1075–1084, 2017.

L. M. Murray, D. Lundén, J. Kudlicka, D. Broman, and T. B. Schön. Delayed sampling and automatic rao-blackwellization of probabilistic programs. *arXiv preprint arXiv:1708.07787*, 2017.

C. A. Naesseth, F. Lindsten, and T. B. Schön. Sequential monte carlo for graphical models. In *Advances in Neural Information Processing Systems*, pages 1862–1870, 2014.

C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.