An introduction to Sequential Monte Carlo

Thang Bui   Jes Frellsen

Department of Engineering
University of Cambridge

Research and Communication Club
6 February 2014

UNIVERSITY OF CAMBRIDGE

Sequential Monte Carlo (SMC) methods

- Initially designed for online inference in dynamical systems
  - Observations arrive sequentially and one needs to update the posterior distribution of hidden variables
  - Analytically tractable solutions are available for linear Gaussian models, but not for complex models
  - Examples: target tracking, time series analysis, computer vision
- Increasingly used to perform inference for a wide range of applications, not just dynamical systems
  - Example: graphical models, population genetic, ...
- SMC methods are scalable, easy to implement and flexible!

Outline

Bibliography I

Andrieu, Christophe, Doucet, Arnaud, & Holenstein, Roman. 2010.
Particle markov chain monte carlo methods.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), **72**(3), 269–342.

Cappé, Olivier, Godsill, Simon J, & Moulines, Eric. 2007.
An overview of existing methods and recent advances in sequential Monte Carlo.
Proceedings of the IEEE, **95**(5), 899–924.

Doucet, Arnaud, De Freitas, Nando, & Gordon, Neil. 2001.
An introduction to sequential Monte Carlo methods.
Pages 3–14 of: Sequential Monte Carlo methods in practice.
Springer.

Gramacy, Robert B., & Polson, Nicholas G. 2011.
Particle Learning of Gaussian Process Models for Sequential Design and Optimization.
Journal of Computational and Graphical Statistics, **20**(1), 102–118.

Holenstein, Roman. 2009.
Particle Markov Chain Monte Carlo.
Ph.D. thesis, The University Of British Columbia.

Bibliography II

Holenstein, Roman, & Doucet, Arnaud. 2007.
Particle Markov Chain Monte Carlo.
Workshop: New directions in Monte Carlo Methods Fleurance, France.

Wilkinson, Richard. 2014.
GPs huh? what are they good for?
Gaussian Process Winter School, Sheffield.

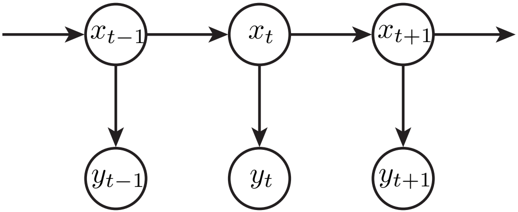
State Space Models

(Doucet *et al.* , 2001; Cappé *et al.* , 2007)

The Markovian, nonlinear, non-Gaussian state space model

- Unobserved signal or states $\{x_t | t \in \mathbb{N}\}$
- Observations or output $\{y_t | t \in \mathbb{N}^+\}$ or $\{y_t | t \in \mathbb{N}\}$

$$
\begin{array}{rll}
P(x_0) & & \\
P(x_t | x_{t-1}) & \text{for } t \geq 1 & \text{(transition probability)} \\
P(y_t | x_t) & \text{for } t \geq 0 & \text{(emission/observation probability)}
\end{array}
$$



Inference for State Space Model

(Doucet *et al.* , 2001; Cappé *et al.* , 2007)

- We are interested the posterior distributions of the unobserved signal

$$\begin{aligned} P(x_{0:t}|y_{0:t}) \quad &- \quad \text{fixed interval smoothing distribution} \\ P(x_{t-L}|y_{0:t}) \quad &- \quad \text{fixed lag smoothing distribution} \\ P(x_t|y_{0:t}) \quad &- \quad \text{filtering distribution} \end{aligned}$$

- and expectations under these posteriors, e.g.

$$\mathbb{E}_{P(x_{0:t}|y_{0:t})}(h_t) = \int h_t(x_{0:t}) P(x_{0:t}|y_{0:t}) \, \mathrm{d}x_{0:t}$$

for some function $h_t : \mathcal{X}^{(t+1)} \to \mathbb{R}^{n_{h_t}}$

Couldn't we use *MCMC*?

(Doucet *et al.* , 2001; Holenstein, 2009)

- Sure, generate N samples from $P(x_{0:t}|y_{0:t})$ using MH
  - Sample a candidate $x'_{0:t}$ from a proposal distribution

$$x'_{0:t} \sim q(x'_{0:t}|x_{0:t})$$

  - Accept the candidate $x'_{0:t}$ with probability

$$\alpha(x'_{0:t}|x_{0:t}) = \min\left[1, \frac{P(x'_{0:t}|y_{0:t})q(x_{0:t}|x'_{0:t})}{P(x_{0:t}|y_{0:t})q(x'_{0:t}|x_{0:t})}\right]$$

- Obtain a set of sample $\{x_{0:t}^{(i)}\}_{i=1}^{N}$
- Calculate empirical estimates for posterior and expectation

$$\tilde{P}(x_{0:t}|y_{0:t}) = \frac{1}{N}\sum_i \delta_{x_{0:t}^{(i)}}(x_{0:t})$$

$$\mathbb{E}_{\tilde{P}(x_{0:t}|y_{0:t})}(h_t) = \int h_t(x_{0:t})\tilde{P}(x_{0:t})\,\mathrm{d}x_{0:t} = \frac{1}{N}\sum_{i=1}^{N} h_t(x_{0:t}^{(i)})$$

Couldn't we use *MCMC*?

(Doucet *et al.* , 2001; Holenstein, 2009)

- Unbiased estimates and in most cases nice convergence

$$\mathbb{E}_{\tilde{P}(x_{0:t}|y_{0:t})}(h_t) \xrightarrow{\text{a.s.}} \mathbb{E}_{P(x_{0:t}|y_{0:t})}(h_t) \quad \text{as} \quad N \to \infty$$

- Problem solved!?
- I can be hard to design a good proposal q
    - Single-site updates $q(x'_j|x_{0:t})$ can lead to slow mixing
- What happens if we get a new data point y_{t+1}?
    - We cannot (directly) reuse the samples $\{x_{0:t}^{(i)}\}$
    - We have to run a new MCMC simulations for $P(x_{0:t+1}|y_{0:t+1})$
- MCMC not well-suited for recursive estimation problems

What about *importance sampling*?

(Doucet *et al.* , 2001)

- Generate N i.i.d. samples $\{x_{0:t}^{(i)}\}_{i=1}^{N}$ from an arbitrary importance sampling distribution $\pi(x_{0:t}|y_{0:t})$
- The empirical estimates are

$$\hat{P}(x_{0:t}|y_{0:t}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_{0:t}^{(i)}}(x_{0:t})\tilde{w}_t^{(i)}$$

$$\mathbb{E}_{\hat{P}(x_{0:t}|y_{0:t})}(h_t) = \frac{1}{N}\sum_{i=1}^{N}h_t(x_{0:t}^{(i)})\tilde{w}_t^{(i)}$$

where the importance weights are

$$w(x_{0:t}) = \frac{P(x_{0:t}|y_{0:t})}{\pi(x_{0:t}|y_{0:t})} \quad \text{and} \quad \tilde{w}_t^{(i)} = \frac{w\left(x_{0:t}^{(i)}\right)}{\sum_j w\left(x_{0:t}^{(j)}\right)}$$

What about *importance sampling*?

(Doucet *et al.* , 2001)

- $\mathbb{E}_{\hat{P}(x_{0:t}|y_{0:t})}(h_t)$ is biased, but converges to $\mathbb{E}_{P(x_{0:t}|y_{0:t})}(h_t)$
- Problem solved!?
- Designing a good importance distribution can be hard!
- Still not adequate for recursive estimation
  - When seeing new data y_{t+1}, we cannot reuse the samples and weights for time t

$$\{x_{0:t}^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^N$$

to sample from $P(x_{0:t+1}|y_{0:t+1})$

Sequential importance sampling

(Doucet *et al.* , 2001; Cappé *et al.* , 2007)

- Assume that the importance distribution can be factored as

$$\pi(x_{0:t}|y_{0:t}) = \underbrace{\pi(x_{0:t-1}|y_{0:t-1})}_{\substack{\text{importance distribution} \\ \text{at time } t-1}} \underbrace{\pi(x_t|x_{0:t-1}, y_{0:t})}_{\text{extension to time } t}$$

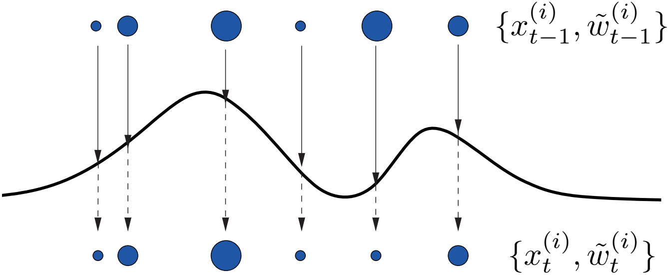
$$= \pi(x_0|y_0) \prod_{k=1}^{t} \pi(x_k|x_{0:k-1}, y_{0:k})$$

- The importance weight can then be evaluated recursively

$$\tilde{w}_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} \frac{P(y_t|x_t^{(i)})P(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{0:t})} \qquad (1)$$

- Given past i.i.d. trajectories $\{x_{0:t-1}^{(i)}|i = 1, \ldots, N\}$ we can
  1. simulate $x_t^{(i)} \sim \pi(x_t|x_{0:t-1}^{(i)}, y_{0:t})$
  2. update the weight $\tilde{w}_t^{(i)}$ for $x_{0:t}^{(i)}$ based on $\tilde{w}_{t-1}^{(i)}$ using eq. (1)
- Note that the extended trajectories $\{x_{0:t}^{(i)}\}$ remain i.i.d.

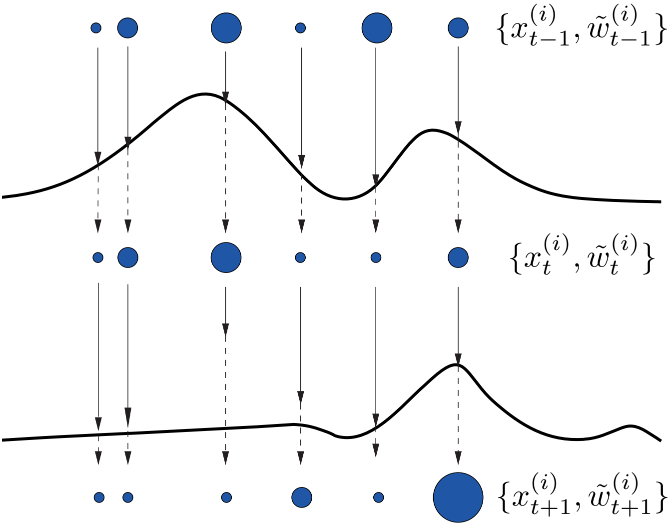
Sequential importance sampling



Adapted from (Doucet *et al.* , 2001)

- Problem solved!?

Sequential importance sampling



Adapted from (Doucet *et al.* , 2001)

- Weights become highly degenerated after few steps

Sequential importance resampling

(Doucet *et al.* , 2001; Cappé *et al.* , 2007)

- Key idea to eliminate weight degeneracy
  1. Eliminate particles with low importance weights
  2. Multiply particles with high importance weights
- Introduce a resampling each time step (or "occasionally")
- Resample a new trajectory $\{x'^{(i)}_{0:t} | i = 1, \ldots, N\}$
  - Draw N samples from

$$\hat{P}(x_{0:t}|y_{0:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}_{0:t}}(x_{0:t}) \tilde{w}^{(i)}_t$$

  - The weights of the new samples are $\tilde{w}'^{(i)}_t = \frac{1}{N}$
- The new empirical (unweighted) distribution a time step t

$$\hat{P}'(x_{0:t}|y_{0:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x^{(i)}_{0:t}}(x_{0:t}) N^{(i)}_t$$

where $N^{(i)}_t$ is the number of copies of $x^{(i)}_{0:t}$.

  - $N^{(i)}_t$ is sampled for a multinomial with parameters $w^{(i)}_t$

Sequential importance resampling

(Doucet *et al.* , 2001; Cappé *et al.* , 2007)

for $i = 1, \ldots, N$ **do**

  Sample $x_0^{(i)} \sim \pi(x_0 | y_0)$

  $w_0^{(i)} \leftarrow \frac{P(y_0 | x_0^{(i)}) P(x_0^{(i)})}{\pi(x_0^{(i)} | y_0)}$

for $t = 1, \ldots, T$ **do**

  Importance sampling step

  for $i = 1, \ldots, N$ **do**

    Sample $\tilde{x}_t^{(i)} \sim \pi(x_t | x_{0:t-1}, y_{0:t})$

    $\tilde{x}_{0:t}^{(i)} \leftarrow (x_{0:t-1}^{(i)}, \tilde{x}_t^{(i)})$

    $\tilde{w}_t^{(i)} \leftarrow w_{t-1}^{(i)} \frac{P(y_t | x_t^{(i)}) P(x_t^{(i)} | x_{t-1}^{(i)})}{\pi(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{0:t})}$

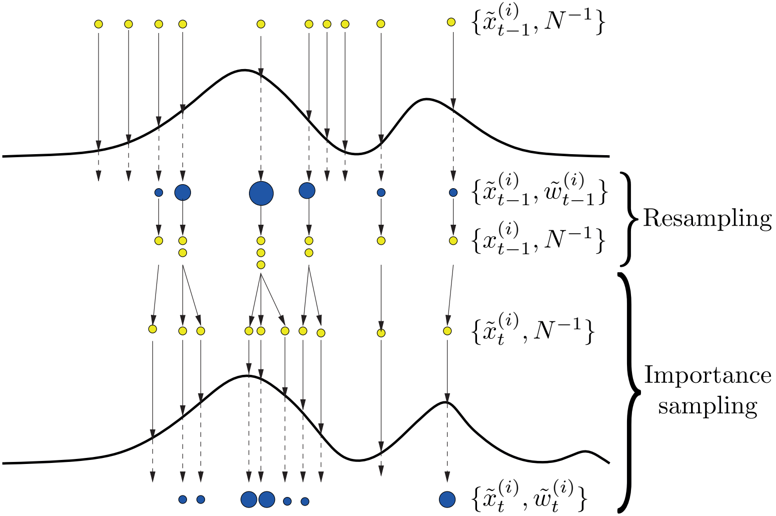
  Resampling/selection step

  Sample N particles $\{x_{0:t}^{(i)}\}$ from $\{\tilde{x}_{0:t}^{(i)}\}$ according to $\{\tilde{w}_t^{(i)}\}$

  $w_t^{(i)} \leftarrow \frac{1}{N}$ **for** $i = 1, \ldots, N$

return $\{x_{0:t}^{(i)}\}_{i=1}^{N}$

Sequential importance resampling

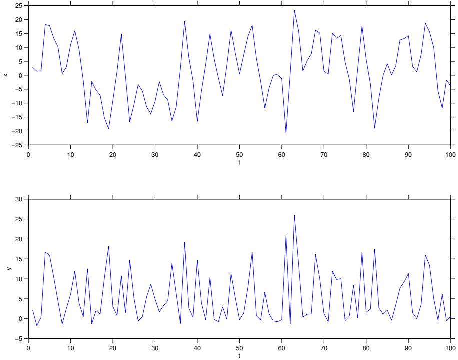


$i = 1, \ldots, N$ and $N = 10$, figure modified from (Doucet *et al.* , 2001)

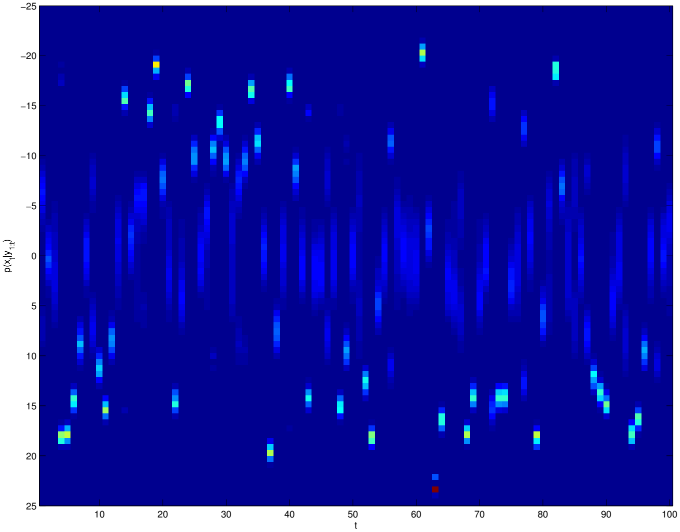
Example - A dynamical system

$$\begin{aligned} x_t &= \tfrac{1}{2}x_{t-1} + 25\tfrac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + u_t \\ y_t &= \tfrac{x_t^2}{20} + v_t \end{aligned}$$

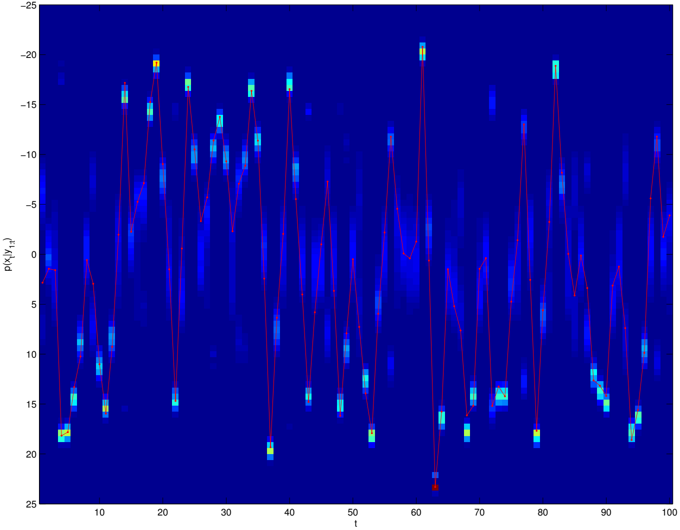
where
$x_0 \sim \mathcal{N}(0, \sigma_0^2), u_t \sim \mathcal{N}(0, \sigma_u^2), v_t \sim \mathcal{N}(0, \sigma_v^2), \sigma_0^2 = \sigma_u^2 = 10, \sigma_v^2 = 1.$



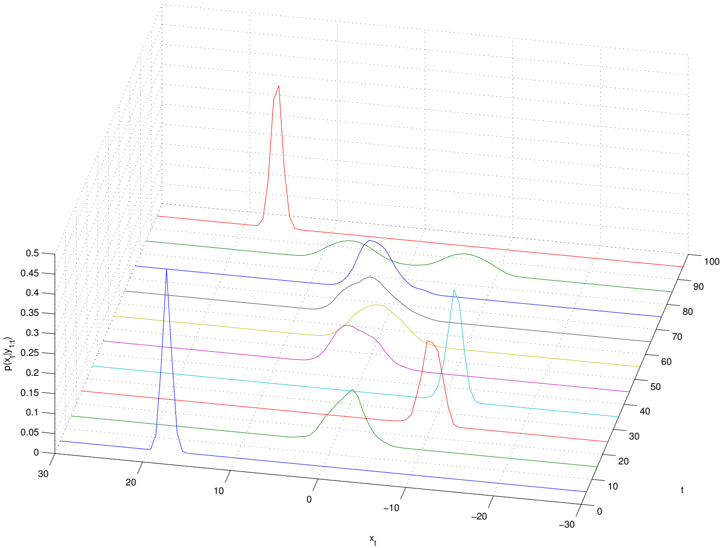
Posterior distribution of states



Posterior distribution of states



Posterior distribution of states



Proposal

- Bootstrap filter uses $\pi_t(x_t|x_{t-1}, y_t) = P(x_t|x_{t-1})$ which leads to a simple form for the importance weight update: $w_t^{(i)} \propto w_{t-1}^{(i)} P(y_t|x_t^{(i)})$
  - The weight update depends on the new proposed state and the observation!
  - Uninformative observation can lead to poor performance
- Optimal proposal:

$$\pi_t(x_t|x_{t-1}, y_t) = P(x_t|x_{t-1}, y_t)$$

therefore: $w_t^{(i)} \propto w_{t-1}^{(i)} P(y_t|x_{t-1}) = \int P(y_t|x_t) P(x_t|x_{t-1}) \mathrm{d}x_t$

  - The weight update depends on the previous state and the observation
  - Analytically intractable integral, need to resort to approximation techniques.

Smoothing

- For a complex SSM, the posterior distribution of state variables can be *smoothed* by including future observations.
- The joint smoothing distribution can be factorised:

$$
\begin{aligned}
P(x_{0:T}|y_{0:T}) &= P(x_T|y_{0:T}) \prod_{t=0}^{T-1} P(x_t|x_{t+1}, y_{0:t}) \\
&\propto P(x_T|y_{0:T}) \prod_{t=0}^{T-1} \underbrace{P(x_t|y_{0:t})}_{\text{filtering distribution}} \underbrace{P(x_{t+1}|x_t)}_{\text{likelihood of future state}}
\end{aligned}
$$

Hence, the weight update:

$$\hat{w}_t^{(i)}(x_{t+1}) = w_t^{(i)} P(x_{t+1}|x_t)$$

Particle smoother

Algorithm:

- Run forward simulation to obtain particle paths $\{x_t^{(i)}, w_t^{(i)}\}_{i=1,\cdots,N;t=1,\cdots,T}$
- Draw $\tilde{x}_T$ from $\hat{P}(x_T|y_{0:T})$
- Repeat:
  - Adjust and normalise the filtering weights $w_t^{(i)}$:

$$\hat{w}_t^{(i)} = w_t^{(i)} P(\tilde{x}_{t+1}|x_t)$$

  - Draw a random sample $\tilde{x}_t$ from $\hat{P}(x_{t:T}|y_{0:T})$

The sequence $(\tilde{x}_0, \tilde{x}_2, \cdots, \tilde{x}_T)$ is a random draw from the approximate distribution $\hat{P}(x_{0:T}|y_{0:T})$ [$\mathcal{O}(NT)$]

MAP estimation

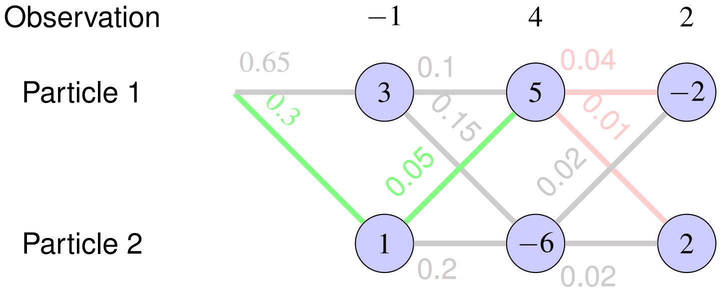
- Maximum a posteriori (MAP) estimate:

$$\underset{x_{0:T}}{\operatorname{argmax}}\, P(x_{0:T}|y_{0:T}) = \underset{x_{0:T}}{\operatorname{argmax}}\, P(x_0) \prod_{t=1}^{T} P(x_t|x_{t-1}) \prod_{t=0}^{T} P(y_t|x_t)$$

  Question: Can we just choose particle trajectory with largest weights?
  Answer: NO!

- Assume a discrete particle grid, $x_t \in {x_t^{(i)}}_{1 \leq i \leq N}$, the approximation can be interpreted as a **Hidden Markov Model** with N states.
  MAP estimate can be found using the Viterbi algorithm
  - Keep track of the probability of the most likely path so far
  - Keep track of the last state index of the most likely path so far

Viterbi algorithm for MAP estimation



Path probability update:

$$\alpha_t^{(j)} = \alpha_{t-1}^{(j)} P(x_t^{(i)} | x_{t-1}^{(i)}) P(y_t | x_t^{(i)})$$

Parameter estimation

- Consider SSMs that have $P(x_t|x_{t-1}, \theta), P(y_t|x_t, \theta)$ where θ is a static parameter vector and one wishes to estimate θ.
- Marginal likelihood:

$$l(y_{0:T}|\theta) \quad = \quad \int p(y_{0:T}, x_{0:T}|\theta) \mathrm{d}x_{0:T}$$

- Optimise $l(y_{0:T}|\theta)$ using the EM algorithm:
  - E-step:

$$\hat{\tau}(\theta, \theta_k) = \sum_{i=1}^{N} w_T^{(i,\theta)} \sum_{t=0}^{T-1} s_{t,\theta}(x_t^{(i,\theta_k)}, x_{t+1}^{(i,\theta_k)})$$

    where

$$s_{t,\theta}(x_t, x_{t+1}) = \log(P(x_{t+1}|x_t, \theta)) + \log(P(y_{t+1}|x_{t+1}, \theta))$$

  - Optimise $\hat{\tau}(\theta|\theta_k)$ to update θ_k.

A generic SMC algorithm

(Holenstein, 2009)

- We want to sample from a target distribution $\pi(\boldsymbol{x})$, $\boldsymbol{x} \in \mathcal{X}^p$
- Assume we have a sequence of bridging distributions of increasing dimension

$$\{\pi_n(\boldsymbol{x}_n)\}_{n=1}^p = \{\pi_1(x_1), \pi_2(x_1, x_2), \ldots, \pi_p(x_1, \ldots, x_p)\}$$

  where

  - $\pi_n(\boldsymbol{x}_n) = Z_n^{-1}\gamma_n(\boldsymbol{x}_n)$
  - $\pi_p(\boldsymbol{x}_p) = \pi(\boldsymbol{x})$

- A sequence of importance densities on $\mathcal{X}$

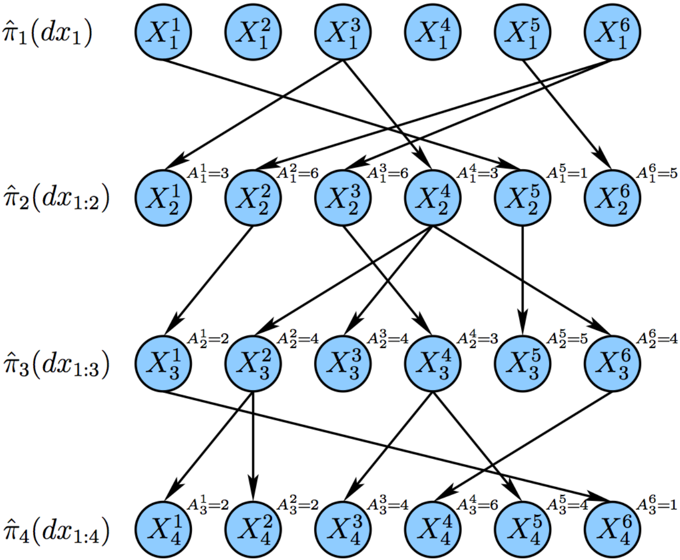
$$\underbrace{M_1(x_1)}_{\text{for initial sample}} \quad , \quad \underbrace{\{M_n(x_n|\boldsymbol{x}_{n-1})\}_{n=2}^p}_{\substack{\text{for extending } \boldsymbol{x}_{n-1} \in \mathcal{X}^{n-1} \\ \text{by sampling } x_n \in \mathcal{X}}}$$

- A resampling distribution

$$r(\boldsymbol{A}_n|\boldsymbol{w}_n), \quad \boldsymbol{A}_n \in \{1, \ldots, N\}^N \text{ and } \boldsymbol{w}_n \in [0,1]^N$$

  where $\boldsymbol{A}_{n-1}^i$ is the parent at time $n-1$ of some particle $\boldsymbol{X}_n^i$

A generic SMC algorithm



From (Holenstein, 2009)

A generic SMC algorithm

At $n = 1$

Sample $\mathbf{X}_1^i \sim M_1(\cdot)$

Update and normalise the weights

$$w_1\left(\mathbf{X}_1^i\right) := \frac{\gamma_1(\mathbf{X}_1^i)}{M_1(\mathbf{X}_1^i)},\ W_1^i = \frac{w_1\left(\mathbf{X}_1^i\right)}{\sum_{k=1}^N w_1\left(\mathbf{X}_1^k\right)}.$$

For $n = 2, ..., p$ **do**

Sample $\mathbf{A}_{n-1} \sim r\left(\cdot|\mathbf{W}_{n-1}\right)$

Sample $X_n^i \sim M_n(\mathbf{X}_{n-1}^{A_{n-1}^i}, \cdot)$ and set $\mathbf{X}_n^i = (\mathbf{X}_{n-1}^{A_{n-1}^i}, X_n^i)$

Update and normalise the weights

$$w_n\left(\mathbf{X}_n^i\right) := \frac{\gamma_n\left(\mathbf{X}_n^i\right)}{\gamma_{n-1}\left(\mathbf{X}_{n-1}^{A_{n-1}^i}\right) M_n\left(\mathbf{X}_{n-1}^{A_{n-1}^i}, X_n^i\right)},$$

$$W_n^i = \frac{w_n\left(\mathbf{X}_n^i\right)}{\sum_{k=1}^N w_n\left(\mathbf{X}_n^k\right)}$$

From (Holenstein, 2009)

A generic SMC algorithm

(Holenstein, 2009)

- Again we can calculate empirical estimates for target and the normalization constant ($\pi(\boldsymbol{x}) = Z^{-1}\gamma(\boldsymbol{x})$)

$$\hat{\pi}^N(\boldsymbol{x}) = \sum_{i=1}^{N} \delta_{\boldsymbol{x}_p^{(i)}}(\boldsymbol{x}) W_p^{(i)}$$

$$\hat{Z}^N(\boldsymbol{x}) = \prod_{n=1}^{p} \left(\frac{1}{N} \sum_{i=1}^{N} w_n(X_n^{(i)}) \right)$$

- Convergence can be shown under weak assumptions

$$\hat{\pi}^N(\boldsymbol{x}) \xrightarrow{\text{a.s.}} \pi(\boldsymbol{x}) \quad \text{as} \quad N \to \infty$$

$$\hat{Z}^N \xrightarrow{\text{a.s.}} Z \quad \text{as} \quad N \to \infty$$

- The SIR algorithm for state space models is a special case of this generic SMC algorithm

Motivation for Particle MCMC

(Holenstein & Doucet, 2007; Holenstein, 2009)

- Let's return to the problem om sampling from a target

$$\pi(\boldsymbol{x}), \text{ where } \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$$

  using MCMC
- Single-site proposal $q(x'_j|\boldsymbol{x})$
  - Easy to design
  - Often leads to slow mixing
- It would be more efficient, if we could update larger blocks
  - Such proposals are harder to construct
- We could use SMC as a proposal distribution!

Particle Metropolis Hastings Sampler

Initialisation $i = 0$

Run an SMC algorithm targeting $\pi(\mathbf{x})$

sample $\mathbf{X}(0) \sim \hat{\pi}^N(\cdot)$ and compute $\hat{Z}^N(0)$

For iteration $i \geq 1$

Run an SMC algorithm targeting $\pi(\mathbf{x})$, sample $\mathbf{X}^* \sim \hat{\pi}^N(\cdot)$ and compute $\hat{Z}^{N,*}$

With probability

$$1 \wedge \frac{\hat{Z}^{N,*}}{\hat{Z}^N(i-1)}, \tag{3.3}$$

set $\mathbf{X}(i) = \mathbf{X}^*$ and $\hat{Z}^N(i) = \hat{Z}^{N,*}$, otherwise set $\mathbf{X}(i) = \mathbf{X}(i-1)$ and $\hat{Z}^N(i) = \hat{Z}^N(i-1)$

From (Holenstein, 2009)

Particle Metropolis Hastings (PMH) Sampler

(Holenstein, 2009)

- Standard independent MH algorithm ($q(x'|x) = q(x')$)
- Target $\tilde{\pi}^N$ and proposal q^N defined on an extended space

$$\frac{\tilde{\pi}^N(\cdot)}{q^N(\cdot)} = \frac{\hat{Z}^N}{Z}$$

which leads to the acceptance ratio

$$\alpha = \min\left[1, \frac{\hat{Z}^{N,*}}{\hat{Z}^N(i-1)}\right]$$

- Note that $\alpha \to 1$ as $N \to \infty$, since $\hat{Z}^N \to Z$ as $N \to \infty$

Particle Gibbs (PG) Sampler

(Holenstein, 2009)

- Assume that we are interested in sampling from

$$\pi(\theta, \boldsymbol{x}) = \frac{\gamma(\theta, \boldsymbol{x})}{Z}$$

- Assume that sampling form
  - $\pi(\theta|\boldsymbol{x})$ is easy
  - $\pi(\boldsymbol{x}|\theta)$ is hard
- The PG Sampler uses SMC to sample from $\pi(\boldsymbol{x}|\theta)$
  1. Sample $\theta(i) \sim \pi(\theta|\boldsymbol{x}(i-1))$
  2. Sample $\boldsymbol{x}(i) \sim \hat{\pi}^N(\boldsymbol{x}|\theta(i))$
- If sampling from $\pi(\theta|\boldsymbol{x})$ is not easy?
  - We can use a MH update for θ

Parameter estimation a state space models using PG

(Andrieu *et al.* , 2010)

- (Re)consider the non-linear state space model

$$x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + V_t$$

$$y_t = \frac{x_t^2}{20} + W_t$$

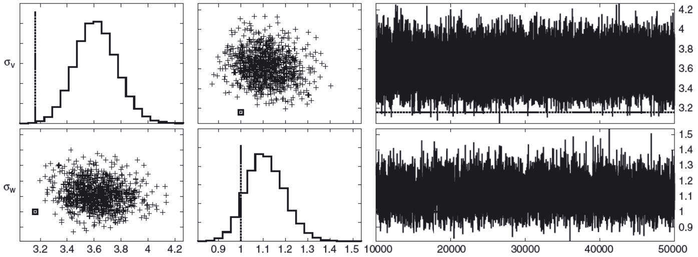
where $x_0 \sim \mathcal{N}(0, \sigma_0^2)$, $V_t \sim \mathcal{N}(0, \sigma_V^2)$ and $W_t \sim \mathcal{N}(0, \sigma_W^2)$

- Assume that $\theta = (\sigma_V^2, \sigma_W^2)$ is unknown
- Simulate $y_{1:T}$ for $T = 500$, $\sigma_0^2 = 5$, $\sigma_V^2 = 10$ and $\sigma_W^2 = 1$
- Sample from $P(\theta, x_{1:t}|y_{1:t})$ using
  - Particle Gibbs sampler, with importance dist. $f_\theta(x_n|x_{n-1})$
  - One-at-a-time MH sampler, with proposal $f_\theta(x_n|x_{n-1})$
- The algorithms ran for 50,000 iterations (burn-in of 10,000)
  - Vague inverse-Gamma priors for $\theta = (\sigma_V^2, \sigma_W^2)$

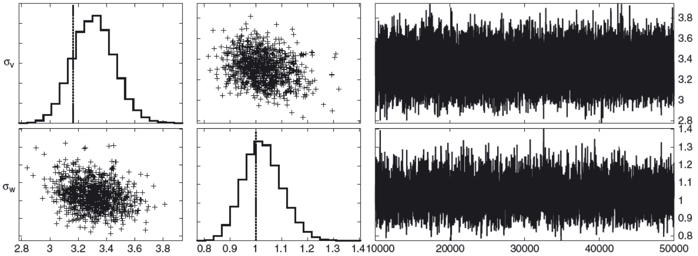
Parameter estimation a state space models using PG

(Andrieu *et al.* , 2010)

One-at-a-time Metropolis Hastings



Particle Gibbs Sampler



Particle learning for GP regression – motivation

Training a GP using data: $\mathcal{D}_{1:n} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ and make prediction:

$$P(y^*|\hat{\theta}, \mathcal{D}, x^*) \tag{2}$$

or

$$P(y^*|\mathcal{D}, x^*) = \int P(y^*|\theta, \mathcal{D}, x^*)P(\theta|\mathcal{D})\mathrm{d}\theta \tag{3}$$

- Estimate model hyperparameters θ_n using ML (2) or use sampling to find the posterior distribution (3)
- Find the inverse of the covariance matrix K_n^{-1}.
- Computational cost $\mathcal{O}(n^3)$.

Sequential update

Given a new observation pair (x_{n+1}, y_{n+1}) that we want to use in our training set, need to find K_{n+1}^{-1} and re-estimate hyperparameters θ_{n+1}.

- a naive implementation costs $\mathcal{O}(n^3)$
- need an efficient approach that makes use of the sequential nature of data.

Particle learning for GP regression

(Gramacy & Polson, 2011; Wilkinson, 2014)

Sufficient information for each particle $S_n^{(i)} = \{K_n^{(i)}, \theta_n^{(i)}\}$
Two-step update based on:

$$\begin{aligned} P(S_{n+1}|\mathcal{D}_{1:n+1}) &= \int P(S_{n+1}|S_n, \mathcal{D}_{n+1}) P(S_n|\mathcal{D}_{1:n+1}) \mathrm{d}S_n \\ &\propto \int P(S_{n+1}|S_n, \mathcal{D}_{n+1}) P(\mathcal{D}_{n+1}|S_n) P(S_n|\mathcal{D}_{1:n}) \mathrm{d}S_n \end{aligned}$$

1. **Resample** indices $\{i\}_{i=1}^N$ with replacement to obtain new indices $\{\zeta(i)\}_{i=1}^N$ according to weights

$$w_i \sim P(\mathcal{D}_{n+1}|S_n^{(i)}) = P(y_{n+1}|x_{n+1}, \mathcal{D}_n, \theta_n^{(i)})$$

2. **Propagate** sufficient information from S_n to S_{n+1}

$$S_{n+1}^{(i)} \sim P(S_{n+1}|S_n^{\zeta(i)}, \mathcal{D}_{1:n+1})$$

Propagation

- Parameters θ_n are static and can be deterministically copied from $S_n^{\zeta(i)}$ to $S_{n+1}^{(i)}$.
- Covariance matrix rank-one update to build K_{n+1}^{-1} from K_n^{-1}:

$$K_{n+1} = \begin{bmatrix} K_n & k(x_{n+1}) \\ k^\top(x_{n+1}) & k(x_{n+1}, x_{n+1}) \end{bmatrix}$$

then

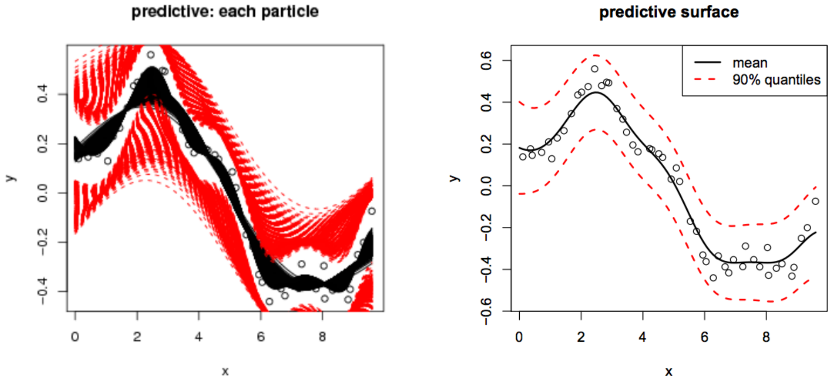
$$K_{n+1}^{-1} = \begin{bmatrix} K_n^{-1} + g_n(x_{n+1}) g_n^\top(x_{n+1}) / \mu_n(x_{n+1}) & g_n(x_{n+1}) \\ g_n^\top(x_{n+1}) & \mu_n(x_{n+1}) \end{bmatrix}$$

where

$$\begin{aligned} g_n(x) &= -\mu(x) K_n^{-1} k(x) \\ \mu_n(x) &= [k(x,x) - k^\top(x) K_n^{-1} k(x)]^{-1} \end{aligned}$$

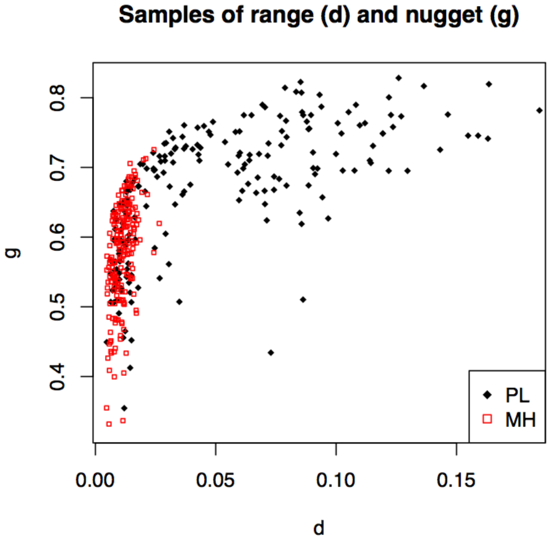
- Use Cholesky update for stability
- Cost: $\mathcal{O}(n^2)$

Illustrative result 1 - Prediction



From (Gramacy & Polson, 2011)

Illustrative result 2 - Particle locations



From (Gramacy & Polson, 2011)

SMC for learning GP models

Advantages:

- Fast for sequential learning problems

Disadvantages:

- Particle degeneracy/depletion
  - Use MCMC sampler to augment the propagate and *rejuvenate* the particles after m sequential updates.
- The predictive distribution given model hyperparameters needs to be analytically tractable [See resample step]

Similar treatment for classification can be found in (Gramacy & Polson, 2011).

Summary

- SMC is a powerful method for sampling from distributions with sequential nature
  - Online learning in state space models
  - Sample from high dimensional distributions
  - As proposal distribution in MCMC
- We presented two concrete examples of using SMC
  - Particle Gibbs for sampling from the posterior distribtuions of the parameters in a non-linear state space model
  - Particle learning of the hyperparameters in a GP model
- Thank you for your attention!