An introduction to Sequential Monte Carlo

Thang Bui     Jes Frellsen

Department of Engineering
University of Cambridge

Research and Communication Club
6 February 2014
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

Motivation
References

Introduction
MCMC and importance sampling
Sequential importance sampling and resampling
Example: A dynamical system
Proposal
Smoothing
MAP estimation
Parameter estimation

A generic SMC algorithm

Particle MCMC

Particle learning for GP regression

Summary
Andrieu, Christophe, Doucet, Arnaud, & Holenstein, Roman. 2010.  
Particle markov chain monte carlo methods.  

An overview of existing methods and recent advances in sequential Monte Carlo.  

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.  

Holenstein, Roman. 2009.  
*Particle Markov Chain Monte Carlo.*  
Holenstein, Roman, & Doucet, Arnaud. 2007.  
*Particle Markov Chain Monte Carlo.*  
Workshop: New directions in Monte Carlo Methods Fleurance, France.

*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 \( \{y_t | t \in \mathbb{N}\} \)

\[
\begin{align*}
P(x_0) \\
P(x_t | x_{t-1}) & \quad \text{for } t \geq 1 \quad \text{(transition probability)} \\
P(y_t | x_t) & \quad \text{for } t \geq 0 \quad \text{(emission/observation probability)}
\end{align*}
\]
We are interested in the posterior distributions of the unobserved signal

\[ P(x_{0:t} | y_{0:t}) \] — fixed interval smoothing distribution

\[ P(x_{t-L} | y_{0:t}) \] — fixed lag smoothing distribution

\[ P(x_t | y_{0:t}) \] — filtering distribution

and expectations under these posteriors, e.g.

\[ \mathbb{E}_{P(x_{0:t} | y_{0:t})}(h_t) = \int h_t(x_{0:t}) P(x_{0:t} | y_{0:t}) \, dx_{0:t} \]

for some function \( h_t : \mathcal{X}^{(t+1)} \rightarrow \mathbb{R}^{n_{ht}} \)
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 $\{x^{(i)}_{0:t}\}_{i=1}^{N}$
- Calculate empirical estimates for posterior and expectation
  \[
  \tilde{P}(x_{0:t}|y_{0:t}) = \frac{1}{N} \sum_{i} \delta_{x^{(i)}_{0:t}}(x_{0:t})
  \]
  \[
  \mathbb{E}\tilde{P}(x_{0:t}|y_{0:t})(h_t) = \int h_t(x_{0:t})\tilde{P}(x_{0:t}) \, dx_{0:t} = \frac{1}{N} \sum_{i=1}^{N} h_t(x^{(i)}_{0:t})
  \]
 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

$$w(x_{0:t}) = \frac{P(x_{0:t}|y_{0:t})}{\pi(x_{0:t}|y_{0:t})}$$

and

$$\tilde{w}_t^{(i)} = \frac{w(x_{0:t}^{(i)})}{\sum_j w(x_{0:t}^{(j)})}$$
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
\[ \pi(x_{0:t} \mid y_{0:t}) = \frac{\pi(x_{0:t-1} \mid y_{0:t-1}) \pi(x_t \mid x_{0:t-1}, y_{0:t})}{\pi(x_0 \mid y_0)} \prod_{k=1}^{t} \pi(x_k \mid x_{0:k-1}, y_{0:k}) \]

The importance weight can then be evaluated recursively
\[ \tilde{w}_t(i) \propto \tilde{w}_{t-1}(i) \frac{P(y_t \mid x_t(i))P(x_t(i) \mid x_{t-1}(i))}{\pi(x_t(i) \mid x_{0:t-1}, y_{0:t})} \]

Given past i.i.d. trajectories \( \{x_{0:t-1}^{(i)} \mid i = 1, \ldots, N \} \) we can
1. simulate \( x_t^{(i)} \sim \pi(x_t \mid x_{0:t-1}^{(i)}, y_{0:t}) \)
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

\[
\{ x^{(i)}_t, \tilde{w}^{(i)}_t \}
\]

\[
\{ x^{(i)}_{t-1}, \tilde{w}^{(i)}_{t-1} \}
\]

Adapted from (Doucet et al., 2001)

- Problem solved!?
Sequential importance sampling

\[ \{x^{(i)}_{t-1}, \tilde{w}^{(i)}_{t-1}\} \]

\[ \{x^{(i)}_t, \tilde{w}^{(i)}_t\} \]

\[ \{x^{(i)}_{t+1}, \tilde{w}^{(i)}_{t+1}\} \]

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 \( \{x_{0:t}^{(i)} | i = 1, \ldots, N \} \)
  - Draw \( N \) samples from

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

- The weights of the new samples are \( \tilde{w}_t^{(i)} = \frac{1}{N} \)

- The new empirical (unweighted) distribution a time step \( t \)

\[
\hat{P}'(x_{0:t} | y_{0:t}) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_{0:t}^{(i)}}(x_{0:t}) N_t^{(i)}
\]

where \( N_t^{(i)} \) is the number of copies of \( x_{0:t}^{(i)} \).

- \( N_t^{(i)} \) is sampled for a multinomial with parameters \( w_t^{(i)} \)
Sequential importance resampling
(Doucet et al., 2001; Cappé et al., 2007)

1: \textbf{for } i = 1, \ldots, N \textbf{ do}
2: \hspace{1em} Sample \( x_0^{(i)} \sim \pi(x_0|y_0) \)
3: \hspace{1em} \( w_0^{(i)} \leftarrow \frac{P(y_0|x_0^{(i)})P(x_0^{(i)})}{\pi(x_0^{(i)}|y_0)} \)
4: \textbf{for } t = 1, \ldots, T \textbf{ do}
5: \hspace{1em} \textbf{importance sampling step}
6: \hspace{2em} \textbf{for } i = 1, \ldots, N \textbf{ do}
7: \hspace{3em} Sample \( \tilde{x}_t^{(i)} \sim \pi(x_t|x_{0:t-1}, y_{0:t}) \)
8: \hspace{3em} \( \tilde{x}_{0:t}^{(i)} \leftarrow (x_{0:t-1}^{(i)}, \tilde{x}_t^{(i)}) \)
9: \hspace{3em} \( \tilde{w}_t^{(i)} \leftarrow w_t^{(i)} \frac{P(y_t|x_t^{(i)})P(x_t^{(i)}|x_{t-1}^{(i)})}{\pi(x_t^{(i)}|x_{0:t-1}, y_{0:t})} \)
10: \textbf{resampling/selection step}
11: \hspace{1em} \textbf{for } i = 1, \ldots, N \textbf{ do}
12: \hspace{2em} Sample \( N \) particles \( \{x_{0:t}^{(i)}\} \) from \( \{\tilde{x}_{0:t}^{(i)}\} \) according to \( \{\tilde{w}_t^{(i)}\} \)
13: \hspace{1em} \( w_t^{(i)} \leftarrow \frac{1}{N} \)
14: \textbf{return } \{x_{0:t}^{(i)}\}_{i=1}^N
Sequential importance resampling

\[
\{\tilde{x}^{(i)}_{t-1}, N^{-1}\}
\]
\[
\{\tilde{x}^{(i)}_{t-1}, \tilde{\tau}^{(i)}_{t-1}\}
\]
\[
\{x^{(i)}_{t-1}, N^{-1}\}
\]
\[
\{\tilde{x}^{(i)}_{t}, N^{-1}\}
\]
\[
\{\tilde{x}^{(i)}_{t}, \tilde{\tau}^{(i)}_{t}\}
\]

\(i = 1, \ldots, N\) and \(N = 10\), figure modified from (Doucet et al., 2001)
Example - A dynamical system

\[ x_t = \frac{1}{2}x_{t-1} + 25 \frac{x_{t-1}}{1+x_{t-1}^2} + 8 \cos(1.2t) + u_t \]

\[ y_t = \frac{x_t^2}{20} + v_t \]

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

\[ p(x_t | y_{1:t}) \]
Posterior distribution of states
Posterior distribution of states

\[ p(x_t | y_{1:t}) \]
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:

\[
w_t^{(i)} \propto w_{t-1}^{(i)} P(y_t|x_{t-1}) = \int P(y_t|x_t)P(x_t|x_{t-1})dx_t
\]

- The weight update depends on the **previous state** and the observation

- Analytically intractable integral, need to resort to approximation techniques.
For a complex SSM, the posterior distribution of state variables can be *smoothed* by including future observations.

The joint smoothing distribution can be factorised:

\[
P(x_{0:T}|y_{0:T}) = P(x_T|y_{0:T}) \prod_{t=0}^{T-1} P(x_t|x_{t+1}, y_{0:t})
\]

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

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
  \[\{x_t^{(i)}, w_t^{(i)}\}_{i=1,\ldots,N; t=1,\ldots,T}\]
- 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 \(\hat{P}(x_{0:T}|y_{0:T}) [O(NT)] \)
MAP estimation

- Maximum a posteriori (MAP) estimate:

\[
\arg\max_{x_{0:T}} P(x_{0:T} \mid y_{0:T}) = \arg\max_{x_0} P(x_0) \prod_{t=1}^{T} P(x_t \mid x_{t-1}) \prod_{t=0}^{T} P(y_t \mid x_t)
\]

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 $\theta$ is a static parameter vector and one wishes to estimate $\theta$.

- Marginal likelihood:
  
  $$l(y_{0:T}|\theta) = \int p(y_{0:T}, x_{0:T}|\theta)dx_{0:T}$$

- Optimise $l(y_{0:T}|\theta)$ using the EM algorithm:
  
  - E-step:
    
    $$\hat{\tau}(\theta, \theta_k) = \sum_{i=1}^{N} w_T^{(i,\theta)} \sum_{t=0}^{T-1} s_{t,\theta}(x^{(i,\theta_k)}_{t}, x^{(i,\theta_k)}_{t+1})$$

    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 $\hat{\tau} \left( \theta|\theta_k \right)$ to update $\theta_k$. 
A generic SMC algorithm
(Holenstein, 2009)

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

$$\left\{ \pi_n(x_n) \right\}_{n=1}^p = \left\{ \pi_1(x_1), \pi_2(x_1, x_2), \ldots, \pi_p(x_1, \ldots, x_p) \right\}$$

where

- $\pi_n(x_n) = Z_n^{-1}\gamma_n(x_n)$
- $\pi_p(x_p) = \pi(x)$

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

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

- A resampling distribution

$$r(A_n|w_n), \quad A_n \in \{1, \ldots, N\}^N \text{ and } w_n \in [0, 1]^N$$

where $A_{n-1}^i$ is the parent at time $n - 1$ of some particle $X_n^i$
A generic SMC algorithm

\[ \hat{\pi}_1(dx_1) \]

\[ \hat{\pi}_2(dx_{1:2}) \]

\[ \hat{\pi}_3(dx_{1:3}) \]

\[ \hat{\pi}_4(dx_{1:4}) \]

From (Holenstein, 2009)
A generic SMC algorithm

1. At $n = 1$
   2. Sample $X_1^i \sim M_1(\cdot)$
   3. Update and normalise the weights:
      \[
      w_1(X_1^i) := \frac{\gamma_1(X_1^i)}{M_1(X_1^i)}, \quad W_1^i = \frac{w_1(X_1^i)}{\sum_{k=1}^{N} w_1(X_k^i)}.
      \]

4. For $n = 2, \ldots, p$ do
   5. Sample $A_{n-1} \sim r(\cdot | W_{n-1})$
   6. Sample $X_n^i \sim M_n(X_{n-1}^{A_{n-1}}^i, \cdot)$ and set $X_n^i = (X_{n-1}^{A_{n-1}^{-1}}^i, X_n^i)$
   7. Update and normalise the weights:
      \[
      w_n(X_n^i) := \frac{\gamma_n(X_n^i)}{\gamma_{n-1}(X_{n-1}^{A_{n-1}^{-1}}) M_n(X_{n-1}^{A_{n-1}^{-1}}, X_n^i)},
      \]
      \[
      W_n^i = \frac{w_n(X_n^i)}{\sum_{k=1}^{N} w_n(X_k^i)}.
      \]

From (Holenstein, 2009)
A generic SMC algorithm
(Holenstein, 2009)

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

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

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

▶ Convergence can be shown under weak assumptions

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

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

▶ 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 of sampling from a target

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

using MCMC

- Single-site proposal $q(x'_j|x)$
  - 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

1. **Initialisation** \( i = 0 \)
2. Run an SMC algorithm targeting \( \pi(x) \)
3. Sample \( X(0) \sim \pi^N(\cdot) \) and compute \( \hat{Z}^N(0) \)

4. **For iteration** \( i \geq 1 \)
5. Run an SMC algorithm targeting \( \pi(x) \), sample \( X^* \sim \pi^N(\cdot) \) and compute \( \hat{Z}^{N,*} \)
6. With probability

   \[ 1 \wedge \frac{\hat{Z}^{N,*}}{\hat{Z}^N(i - 1)}, \]  

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

\[ \pi(\theta, x) = \frac{\gamma(\theta, x)}{Z} \]

- Assume that sampling form

  - \( \pi(\theta|x) \) is easy
  - \( \pi(x|\theta) \) is hard

- The PG Sampler uses SMC to sample from \( \pi(x|\theta) \)
  1. Sample \( \theta(i) \sim \pi(\theta|x(i - 1)) \)
  2. Sample \( x(i) \sim \hat{\pi}^N(x|\theta(i)) \)

- If sampling from \( \pi(\theta|x) \) is not easy?
  - We can use a MH update for \( \theta \)
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:

$$P(y^* | \hat{\theta}, \mathcal{D}, x^*)$$  \hspace{1cm} (2)

or

$$P(y^* | \mathcal{D}, x^*) = \int P(y^* | \theta, \mathcal{D}, x^*) P(\theta | \mathcal{D}) d\theta$$  \hspace{1cm} (3)

▶ Estimate model hyperparameters $\theta_n$ using ML (2) or use sampling to find the posterior distribution (3)
▶ Find the inverse of the covariance matrix $K^{-1}_n$.
▶ 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 \(\theta_{n+1}\).

- a naive implementation costs \(O(n^3)\)
- 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:

$$P(S_{n+1} | D_{1:n+1}) = \int P(S_{n+1} | S_n, D_{n+1})P(S_n | D_{1:n+1})dS_n$$

$$\propto \int P(S_{n+1} | S_n, D_{n+1})P(D_{n+1} | S_n)P(S_n | D_{1:n})dS_n$$

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

$$w_i \sim P(D_{n+1} | S_n^{(i)}) = P(y_{n+1} | x_{n+1}, D_n, \theta_n^{(i)})$$

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

$$S_{n+1}^{(i)} \sim P(S_{n+1} | S_n^{\zeta(i)}, D_{1:n+1})$$
Propagation

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

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

then

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

where

$$g_n(x) = -\mu(x)K_n^{-1}k(x)$$

$$\mu_n(x) = [k(x, x) - k^\top(x)K_n^{-1}k(x)]^{-1}$$

- Use Cholesky update for stability
- Cost: $O(n^2)$
Illustrative result 1 - Prediction

From (Gramacy & Polson, 2011)
Illustrative result 2 - Particle locations

Samples of range (d) and nugget (g)

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).
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 distributions of the parameters in a non-linear state space model
  - Particle learning of the hyperparameters in a GP model

Thank you for your attention!