Monte Carlo Approximation of Monte Carlo Filters

Adam M. Johansen*, Arnaud Doucet‡ and Nick Whiteley†

*University of Warwick Department of Statistics,
†University of Bristol School of Mathematics,
‡University of Oxford Department of Statistics

a.m.johansen@warwick.ac.uk
http://go.warwick.ac.uk/amjohansen/talks

January 8th, 2013
Monte Carlo Approximation of Monte Carlo Filters

Context & Outline

Filtering in State-Space Models:
- SIR Particle Filters [GSS93]
- Rao-Blackwellized Particle Filters [AD02, CL00]
- Block-Sampling Particle Filters [DBS06]

Exact Approximation of Monte Carlo Algorithms:
- Particle MCMC [ADH10]

Approximating the RBPF
- Approximated Rao-Blackwellized Particle Filters [CSOL11]
- Exactly-approximated RBPFs [JWD12]

Approximating the BSPF
- Local SMC [JD13]
The Structure of the Problem
Hidden Markov Models / State Space Models

- Unobserved Markov chain \( \{ X_n \} \) transition \( f \).
- Observed process \( \{ Y_n \} \) conditional density \( g \).
- Density:

\[
p(x_{1:n}, y_{1:n}) = f_1(x_1)g(y_1|x_1) \prod_{i=2}^{n} f(x_i|x_{i-1})g(y_i|x_i).
\]
Motivating Examples

- **Tracking, e.g. ACV Model:**
  - **States:** \( x_n = [s_n^x \ u_n^x \ s_n^y \ u_n^y]^T \)
  - **Dynamics:** \( x_n = Ax_{n-1} + \epsilon_n \)
  - **Observation:** \( y_n = Bx_n + \nu_n \)

- **Stochastic Volatility, e.g.:**
  - **States:** \( x_n \) is latent volatility.
  - **Dynamics:** \( f(x_i|x_{i-1}) = \mathcal{N}(\phi x_{i-1}, \sigma^2) \)
  - **Observations:** \( g(y_i|x_i) = \mathcal{N}(0, \beta^2 \exp(x_i)) \)
Formally:

\[
p(x_{1:n} | y_{1:n}) = \frac{p(x_{1:n-1} | y_{1:n-1}) f(x_n | x_{n-1}) g(y_n | x_n)}{\int g(y_n | x'_n) f(x'_n | x_{n-1}) p(x'_{n-1} | y_{1:n-1}) dx'_{n-1:n}}
\]

Importance Sampling in The HMM Setting

- Given \( p(x_{1:n} | y_{1:n}) \) for \( n = 1, 2, \ldots \)
- Choose \( q_n(x_{1:n}) = q_n(x_n | x_{1:n-1}) q_{n-1}(x_{1:n-1}) \).
- Weight:

\[
w_n(x_{1:n}) \propto \frac{p(x_{1:n} | y_{1:n})}{q_n(x_{1:n})} \propto w_{n-1}(x_{1:n-1}) \frac{f(x_n | x_{n-1}) g(y_n | x_n)}{q_n(x_n | x_{n-1})}
\]
Sequential Importance Sampling – Prediction & Update

▶ A first “particle filter”:
  ▶ Simple default: $q_n(x_n|x_{n-1}) = f(x_n|x_{n-1})$.
  ▶ Importance weighting becomes:
    
    $$w_n(x_{1:n}) = w_{n-1}(x_{1:n-1}) \times g(y_n|x_n)$$

▶ Algorithmically, at iteration $n$:
  ▶ Given $\{ W_{n-1}^i, X_{1:n-1}^i \}$ for $i = 1, \ldots, N$:
    ▶ Sample $X_n^i \sim f(\cdot|X_{n-1}^i)$ (prediction)
    ▶ Weight $W_n^i \propto W_{n-1}^i g(y_n|X_n^i)$ (update)
Resampling

- Simplest approach:
  \[ \tilde{X}_n^i \sim \text{iid} \sum_{j=1}^{n} W_n^j \delta_{X_n^j} \]

- Replace \( \{ W_n^i, X_n^i \}_{i=1}^{N} \) with \( \{ \frac{1}{N}, \tilde{X}_n^i \}_{i=1}^{N} \).
- Lower variance options preferable.

Algorithmically, at iteration \( n \):
- Given \( \{ W_{n-1}^i, X_{1:n-1}^i \}_{i=1}^{N} \):
- **Resample** \( \rightarrow \{ \frac{1}{N}, \tilde{X}_{1:n-1}^i \} \).
- For \( i = 1, \ldots, N \):
  - Sample \( X_n^i \sim q_n(\cdot | \tilde{X}_{n-1}^i) \)
  - Weight \( W_n^i \propto \frac{f(X_n^i | \tilde{X}_{n-1}^i) g(y_n | X_n^i)}{q_n(X_n^i | \tilde{X}_{n-1}^i)} \)
Monte Carlo Approximation of Monte Carlo Filters

Background

Hidden Markov Models

Iteration 2
Monte Carlo Approximation of Monte Carlo Filters

Background

Hidden Markov Models

Iteration 3
Monte Carlo Approximation of Monte Carlo Filters

Background

Hidden Markov Models

Iteration 4
Monte Carlo Approximation of Monte Carlo Filters

Background

Hidden Markov Models

Iteration 5

![Graph showing iteration 5 with various points and lines representing data.](image-url)
Monte Carlo Approximation of Monte Carlo Filters

Background

Hidden Markov Models

Iteration 7
Iteration 8
Monte Carlo Approximation of Monte Carlo Filters

Background

Hidden Markov Models

Iteration 9
Rao-Blackwellised Particle Filters
A (Rather Broad) Class of Hidden Markov Models

Unobserved Markov chain $\{(X_n, Z_n)\}$ transition $f$.

Observed process $\{Y_n\}$ conditional density $g$.

Density:

$$p(x_{1:n}, z_{1:n}, y_{1:n}) = f_1(x_1, z_1)g(y_1|x_1, z_1) \prod_{i=2}^{n} f(x_i, z_i|x_{i-1}, z_{i-1})g(y_i|x_i, z_i).$$
Simple Solutions

Formally:
\[ p((x, z)_{1:n}|y_{1:n}) \propto p((x, z)_{1:n-1}|y_{1:n-1}) f((x, z)_n|(x, z)_{n-1}) g(y_n|(x, z)_n) \]

An SIR Filter: Algorithmically, at iteration \( n \):
- Given \( \{W_{n-1}^i, (X, Z)_{1:n-1}^i\} \) for \( i = 1, \ldots, N \):
- **Resample**, obtaining \( \{\frac{1}{N}, (\tilde{X}, \tilde{Z})_{1:n-1}^i\} \).
  - Sample \( (X, Z)_n^i \sim q_n(\cdot|(\tilde{X}, \tilde{Z})_{1:n-1}^i) \)
  - **Weight** \( W_n^i \propto \frac{f((X, Z)_n^i|(\tilde{X}, \tilde{Z})_{1:n-1}^i) g(y_n|(X, Z)_n^i)}{q_n((X, Z)_n^i|(\tilde{X}, \tilde{Z})_{1:n-1}^i)} \)
A Rao-Blackwellized SIR Filter

Algorithmically, at iteration $n$:

- **Given** $\{W^X_{n-1}^i, (X^i_1:n-1, p(z_1:n-1|X^i_1:n-1, y_1:n-1))\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\tilde{X}^i_1:n-1, p(z_1:n-1|\tilde{X}^i_1:n-1, y_1:n-1))\}$.
- **For** $i = 1, \ldots, N$:
  - Sample $X^i_n \sim q_n(\cdot|\tilde{X}^i_{n-1})$
  - Set $X^i_1:n \leftarrow (\tilde{X}^i_1:n-1, X^i_n)$.
  - **Weight** $W^X_{n}^i \propto \frac{p(X^i_n; y_n|\tilde{X}^i_{n-1})}{q_n(X^i_n|\tilde{X}^i_{n-1})}$
  - Compute $p(z_1:n|y_{1:n}, X^i_1:n)$.

Requires analytically tractable substructure.
An Approximate Rao-Blackwellized SIR Filter

Algorithmically, at iteration $n$:

- **Given** $\{W_{n-1}^{X,i}, (X_{1:n-1}^i, \hat{p}(z_{1:n-1}|X_{1:n-1}^i, y_{1:n-1}))\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\tilde{X}_{1:n-1}^i, \hat{p}(z_{1:n-1}|\tilde{X}_{1:n-1}^i, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\tilde{X}_{n-1}^i)$
  - Set $X_{1:n}^i \leftarrow (\tilde{X}_{1:n-1}^i, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{\hat{p}(X_n^i, y_n|\tilde{X}_{n-1}^i)}{q_n(X_n^i|\tilde{X}_{n-1}^i)}$
  - Compute $\hat{p}(z_{1:n}|y_{1:n}, X_{1:n}^i)$.
- Our proposal: use $N$ “local” $M$-particle filters to provide $\hat{p}(z_{1:n}|y_{1:n}, X_{1:n}^i)$ and $\hat{p}(X_n^i, y_n|\tilde{X}_{n-1}^i)$.

Is approximate; how does error accumulate?
How can this be justified?

- As an extended space SIR algorithm.
- Via unbiased estimation arguments.

Note also the $M = 1$ and $M \to \infty$ cases.

How does this differ from CSOL11?

Principally in the *local* weights, benefits including:

- Valid ($N$-consistent) for all $M \geq 1$ rather than ($M, N$)-consistent.
- Computational cost $O(MN)$ rather than $O(M^2N)$.
- Only requires knowledge of joint behaviour of $x$ or $z$; doesn’t require say $p(x_n|x_{n-1}, z_{n-1})$. 

Toy Example: Model

We use a simulated sequence of 100 observations from the model defined by the densities:

\[
\mu(x_1, z_1) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ z_1 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)
\]

\[
f(x_n, z_n|x_{n-1}, z_{n-1}) = \mathcal{N}\left(\begin{pmatrix} x_n \\ z_n \end{pmatrix}; \begin{pmatrix} x_{n-1} \\ z_{n-1} \end{pmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)
\]

\[
g(y_n|x_n, z_n) = \mathcal{N}\left(y_n; \begin{pmatrix} x_n \\ z_n \end{pmatrix}, \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_z^2 \end{bmatrix}\right)
\]

Consider IMSE (relative to optimal filter) of filtering estimate of first coordinate marginals.
Approximation of the RBPF

For $\sigma_x^2 = \sigma_z^2 = 1$. 
Monte Carlo Approximation of Monte Carlo Filters

- Approximating the RBPF
- Exact Approximation of the RBPF

Computational Performance

For $\sigma_x^2 = \sigma_z^2 = 1$. 
Monte Carlo Approximation of Monte Carlo Filters

Approximating the RBPF

Exact Approximation of the RBPF

Computational Performance

For $\sigma_x^2 = 10^2$ and $\sigma_z^2 = 0.1^2$. 
Local Particle Filters
What About Other HMMs / Algorithms?

Returning to:

- Unobserved Markov chain \( \{X_n\} \) transition \( f \).
- Observed process \( \{Y_n\} \) conditional density \( g \).
- Density:

\[
p(x_{1:n}, y_{1:n}) = f_1(x_1)g(y_1|x_1) \prod_{i=2}^{n} f(x_i|x_{i-1})g(y_i|z_i).
\]
Block Sampling: An Idealised Approach

At time $n$, given $x_{1:n-1}$; discard $x_{n-L+1:n-1}$:

- Sample from $q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})$.
- Weight with

$$W(x_{1:n}) = \frac{p(x_{1:n}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})q(x_{n-L+1:n}|x_{n-L}, y_{1:n-L+1:n})}$$

- Optimally,

$$q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n}) = p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})$$

$$W(x_{1:n}) \propto \frac{p(x_{1:n-L}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})} = p(y_n|x_{1:n-L}, y_{n-L+1:n-1})$$

- Typically intractable; auxiliary variable approach in [DBS06].
Motivating Example

- **Model:**
  \[ f(x_i|x_{i-1}) = \mathcal{N}(\phi x_{i-1}, \sigma^2) \]
  \[ g(y_i|x_i) = \mathcal{N}(0, \beta^2 \exp(x_i)) \]

- **Simple Bootstrap SIR Algorithm:**
  - **Proposal:**
    \[ q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1}) \]
  - **Weighting:**
    \[ W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1}) g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t) \]

  - Resample residually every iteration.

- **Is hierarchical SMC possible here?**
Local Particle Filtering: First Particle
Local Particle Filtering: SMC Proposal
Local Particle Filtering: CSMC Auxiliary Proposal
The Justification of Local SMC

- Propose from:

\[
\begin{align*}
\mathcal{U}^{\otimes n-1}_{1:M}(b_{1:n-2}, k) & p(x_{1:n-1} | y_{1:n-1}) \psi^M_{n,L}(\bar{a}_{n-L+2:n}, \bar{x}_{n-L+1:n}, \bar{k}; x_{n-L}) \\
\tilde{\psi}^M_{n-1,L-1}(\bar{a}_{n-L+2:n-1} \otimes^{k} \bar{x}_{n-L+1:n-1}; x_{n-L} \parallel b_{n-L+2:n-1}, x_{n-L+1:n-1})
\end{align*}
\]

- Target:

\[
\begin{align*}
\mathcal{U}^{\otimes n}_{1:M}(b_{1:n-L}, \bar{b}^k_{n,n-L+1:n-1}, \bar{k}) & p(x_{1:n-L}, \bar{x}_{n-L+1:n} | y_{1:n}) \\
\tilde{\psi}^M_{n,L}(\bar{a}_{n-L+2:n} \otimes^{k} \bar{x}_{n-L+1:n}; x_{n-L} \parallel \bar{b}^k_{n,n-L+1:n}, \bar{x}_{n-L+1:n}) \\
\psi^M_{n-1,L-1}(\bar{a}_{n-L+2:n-1}, \bar{x}_{n-L+1:n-1}, k; x_{n-L})
\end{align*}
\]

- Weight: \(\tilde{Z}_{n-L+1:n} / \tilde{Z}_{n-L+1:n-1}\).
Stochastic Volatility Bootstrap Local SMC

- **Model:**
  \[ f(x_i|x_{i-1}) = \mathcal{N}(\phi x_{i-1}, \sigma^2) \]
  \[ g(y_i|x_i) = \mathcal{N}(0, \beta^2 \exp(x_i)) \]

- **Top Level:**
  - Local SMC proposal.
  - Stratified resampling when ESS < \( N/2 \).

- **Local SMC Proposal:**
  - Proposal:
    \[ q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1}) \]
  - Weighting:
    \[ W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t) \]
  - Resample residually every iteration.
SV Simulated Data

Simulated Data

Observed Values

n
SV Bootstrap Local SMC: $M=100$

![Graph showing the average number of unique values for different values of L (2, 4, 6, 10) as a function of n.](#)
SV Bootstrap Local SMC: $M=1000$

The graph shows the average number of unique values for different values of $L$ when $N=100$ and $M=1000$. The legend indicates the following:

- Solid line: $L=2$
- Dashed line: $L=4$
- Dotted line: $L=6$
- Dash-dotted line: $L=10$
- Purple solid line: $L=14$
- Pink solid line: $L=18$

The x-axis represents $n$, and the y-axis represents the average number of unique values.
SV Bootstrap Local SMC: $M=10000$

![Diagram showing the average number of unique values for different values of $L$.

- $L = 2$
- $L = 4$
- $L = 6$
- $L = 10$
- $L = 14$
- $L = 18$
- $L = 22$
- $L = 26$]
SV Exchange Rate Data

Exchange Rate Data

Observed Values

Exchange Rate Data
SV Bootstrap Local SMC: M=100

![Graph showing the average number of unique values for different values of L (2, 4, 6, 10) with N = 100 and M = 100]
SV Bootstrap Local SMC: $M=1000$

![Graph showing average number of unique values for different values of L.](image)
SV Bootstrap Local SMC: M=10000

N=100, M=10,000

- L = 2
- L = 4
- L = 6
- L = 10
- L = 14
- L = 18
- L = 22
- L = 26
SV Exchange Rate Data

Exchange Rate Data

Observed Values

n
In Conclusion

- SMC can be used hierarchically.
- Software implementation is not difficult [Joh09].
- The Rao-Blackwellized particle filter can be approximated exactly.
  - Can reduce estimator variance at fixed cost.
  - Attractive for distributed/parallel implementation.
  - Allows combination of different sorts of particle filter.
  - Can be combined with other techniques for parameter estimation etc..
- The optimal block-sampling particle filter can be approximated exactly.
  - Requiring only simulation from the transition and evaluation of the likelihood.
  - Easy to parallelise.
  - Low storage cost.
References I


