Exact Approximation and Particle Filters

Adam M. Johansen et al.

Collaborators Include:
Arnaud Doucet, Axel Finke, Anthony Lee, Nick Whiteley

ERCIM
8th December 2014

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] and SMC^2 [CJP13]

Exact Approximation and Particle Filters:

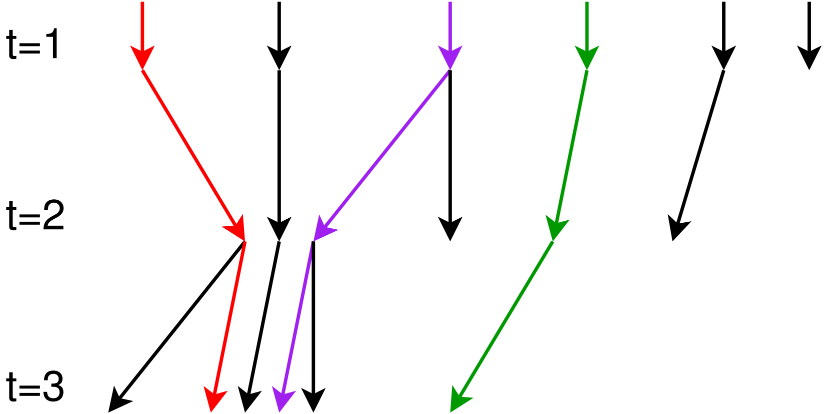
- Approximated RBPFs [CSOL11] *Exactly* [JWD12]
- Hierarchical SMC [JD]
- Pseudomarginal State Augmentation: More of the SAME?

Particle MCMC

- MCMC algorithms which employ SMC proposals [ADH10]
- SMC algorithm as a collection of RVs
  - Values
  - Weights
  - Ancestral Lines
- Construct MCMC algorithms:
  - With many auxiliary variables
  - *Exactly* invariant for distribution on extended space
  - Standard MCMC arguments justify strategy
  - SMC^2 employs the same approach within an SMC setting.

- What else does this allow us to do with SMC?

Ancestral Trees



$a_3^1 = 1$ $a_3^4 = 3$ $a_2^1 = 1$ $a_2^4 = 3$

$b_{3,1:3}^2 = (1, 1, 2)$ $b_{3,1:3}^4 = (3, 3, 4)$ $b_{3,1:3}^6 = (4, 5, 6)$

SMC Distributions

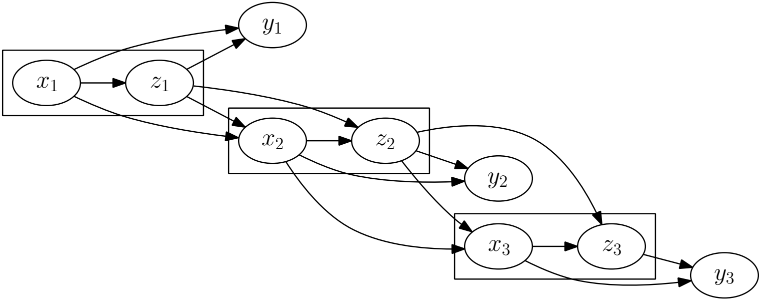
Formally gives rise to the **SMC Distribution**:

$$\psi^M_{n,L}\left(\overline{\mathbf{a}}_{n-L+2:n}, \overline{\mathbf{x}}_{n-L+1:n}, \overline{k}; x_{n-L}\right)$$
$$= \left[\prod_{i=1}^{M} q(\overline{x}^i_{n-L+1} \,|\, \overline{x}_{n-L})\right] \prod_{p=n-L+2}^{n} \left[r(\overline{\mathbf{a}}_p | \overline{\mathbf{w}}_{p-1}) \prod_{i=1}^{M} q\left(\overline{x}^i_p | \overline{x}^{\overline{a}^i_p}_{p-1}\right)\right] r(\overline{k}|\overline{\mathbf{w}}_{\mathbf{n}})$$

and the **conditional SMC Distribution**:

$$\widetilde{\psi}^M_{n,L}\left(\widetilde{\mathbf{a}}^{\ominus k}_{n-L+2:n}, \widetilde{\mathbf{x}}^{\ominus k}_{n-L+1:n}; x_{n-L} \,\middle\|\, \widetilde{b}^k_{n-L+1:n-1}, k, \widetilde{x}^k_{n-L+1:n}\right)$$
$$= \frac{\psi^M_{n,L}(\widetilde{\mathbf{a}}_{n-L+2:n}, \widetilde{\mathbf{x}}_{n-L+1:n}, k; x_{n-L})}{q\left(\widetilde{x}^{\widetilde{b}^k_{n,n-L+1}}_{n-L+1} | x_{n-L}\right) \left[\prod_{p=n-L+2}^{n} r\left(\widetilde{b}^k_{n,p} | \widetilde{\mathbf{w}}_{\mathbf{p-1}}\right) q\left(\widetilde{x}^{\widetilde{b}^k_{n,p}}_p | \widetilde{x}^{\widetilde{b}^n_{n,p-1}}_{p-1}\right)\right] r(k|\widetilde{\mathbf{w}}_n)}$$

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:

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

Formal Solutions

- Filtering and Prediction Recursions:

$$p(x_n, z_n|y_{1:n}) = \frac{p(x_n, z_n|y_{1:n-1})g(y_n|x_n, z_n)}{\int p(x_n', z_n'|y_{1:n-1})g(y_n|x_n', z_n')d(x_n', z_n')}$$

$$p(x_{n+1}, z_{n+1}|y_{1:n}) = \int p(x_n, z_n|y_{1:n})f(x_{n+1}, z_{n+1}|x_n, z_n)d(x_n, z_n)$$

- Smoothing:

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

A Simple SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^i, (X,Z)_{1:n-1}^i\}$ for $i = 1, \ldots, N$:
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}, \widetilde{Z})_{1:n-1}^i\}$.
  - Sample $(X,Z)_n^i \sim q_n(\cdot|(\widetilde{X}, \widetilde{Z})_{n-1}^i)$
  - Weight $W_n^i \propto \frac{f((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)g(y_n|(X,Z)_n^i)}{q_n((X,Z)_n^i|(\widetilde{X},\widetilde{Z})_{n-1}^i)}$

Actually:

- Resample efficiently.
- Only resample when necessary.
- ...

A Rao-Blackwellized SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^{X,i}, (X_{1:n-1}^i, p(z_{1:n-1}|X_{1:n-1}^i, y_{1:n-1})\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}_{1:n-1}^i, p(z_{1:n-1}|\widetilde{X}_{1:n-1}^i, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^i)$
  - Set $X_{1:n}^i \leftarrow (\widetilde{X}_{1:n-1}^i, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{p(X_n^i, y_n|\widetilde{X}_{n-1}^i)}{q_n(X_n^i|\widetilde{X}_{n-1}^i)}$
  - Compute $p(z_{1:n}|y_{1:n}, X_{1:n}^i)$.

Requires analytically tractable substructure.

An Approximate Rao-Blackwellized SIR Filter

Algorithmically, at iteration n:

- Given $\{W_{n-1}^{X,i}, (X_{1:n-1}^{i}, \widehat{p}(z_{1:n-1}|X_{1:n-1}^{i}, y_{1:n-1})\}$
- **Resample**, obtaining $\{\frac{1}{N}, (\widetilde{X}_{1:n-1}^{i}, \widehat{p}(z_{1:n-1}|\widetilde{X}_{1:n-1}^{i}, y_{1:n-1}))\}$.
- For $i = 1, \ldots, N$:
  - Sample $X_n^i \sim q_n(\cdot|\widetilde{X}_{n-1}^{i})$
  - Set $X_{1:n}^{i} \leftarrow (\widetilde{X}_{1:n-1}^{i}, X_n^i)$.
  - Weight $W_n^{X,i} \propto \frac{\widehat{p}(X_n^i, y_n|\widetilde{X}_{n-1}^{i})}{q_n(X_n^i|\widetilde{X}_{n-1}^{i})}$
  - Compute $\widehat{p}(z_{1:n}|y_{1:n}, X_{1:n}^{i})$.

Is approximate; how does error accumulate?

Exactly Approximated Rao-Blackwellized SIR Filter

At time $n = 1$

- Sample, $X_1^i \sim q^x(\cdot | y_1)$.
- Sample, $Z_1^{i,j} \sim q^z\left(\cdot | X_1^i, y_1\right)$.
- Compute and normalise the local weights

$$w_1^z\left(X_1^i, Z_1^{i,j}\right) := \frac{p(X_1^i, y_1, Z_1^{i,j})}{q^z\left(Z_1^{i,j} \middle| X_1^i, y_1\right)}, W_1^{z,i,j} := \frac{w_1^z\left(X_1^i, Z_1^{i,j}\right)}{\sum_{k=1}^M w_1^z\left(X_1^i, Z_1^{i,k}\right)}$$

$$\text{define } \widehat{p}(X_1^i, y_1) := \frac{1}{M}\sum_{j=1}^M w_1^z\left(X_1^i, Z_1^{i,j}\right).$$

- Compute and normalise the top-level weights

$$w_1^x\left(X_1^i\right) := \frac{\widehat{p}(X_1^i, y_1)}{q^x\left(X_1^i | y_1\right)}, \; W_1^{x,i} := \frac{w_1^x\left(X_1^i\right)}{\sum_{k=1}^N w_1^x\left(X_1^k\right)}.$$

At times $n > 2$, resample and do essentially the same again...

Toy Example: Model

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

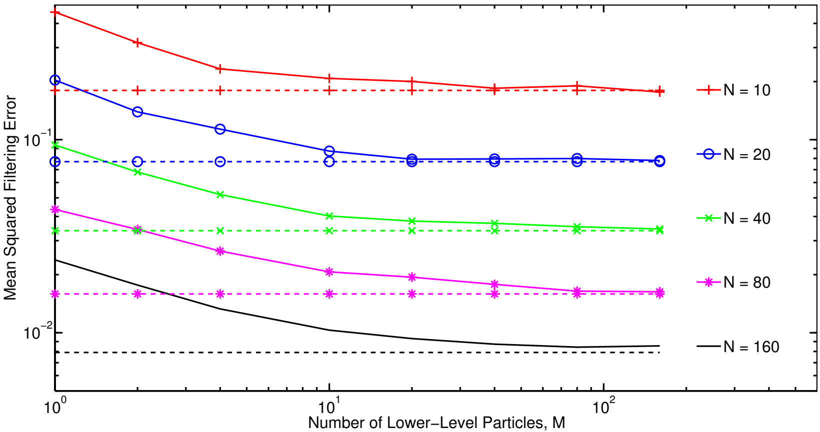
$$\mu(x_1, z_1) = \mathcal{N}\left(\begin{pmatrix} x_1 \\ z_1 \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

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

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

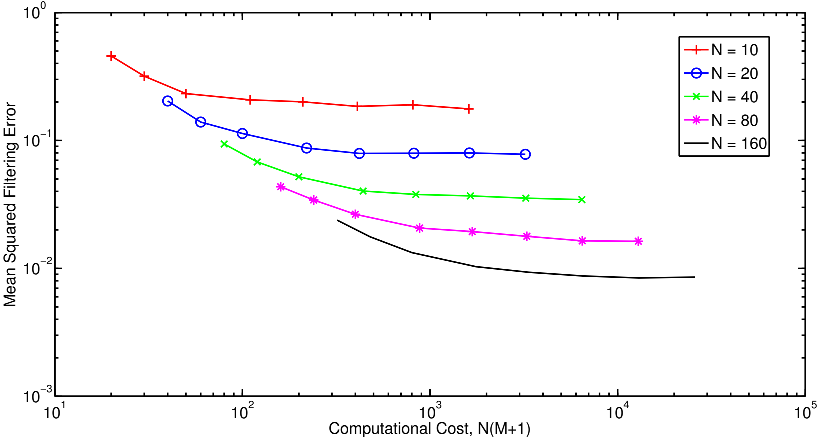
Consider IMSE (relative to optimal filter) of filtering estimate of first coordinate marginals.

Approximation of the RBPF



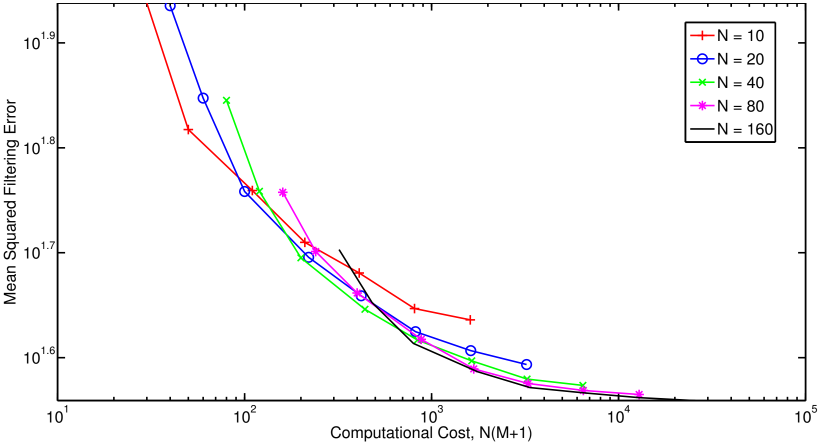
For $\sigma_x^2 = \sigma_z^2 = 1$.

Computational Performance



For $\sigma_x^2 = \sigma_z^2 = 1$.

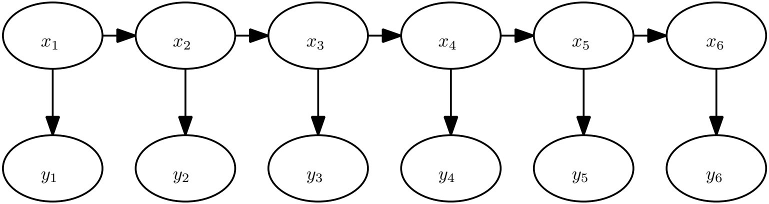
Computational Performance



For $\sigma_x^2 = 10^2$ and $\sigma_z^2 = 0.1^2$.

What About Other HMMs / Algorithms?

Returning to:



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

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

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

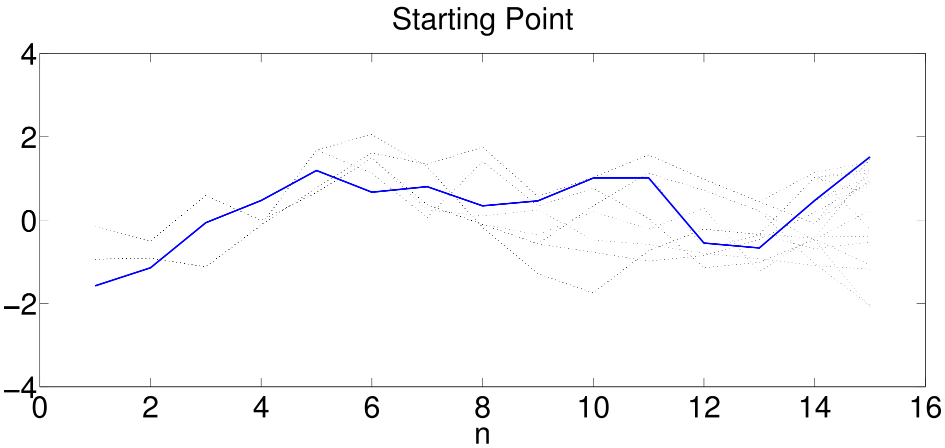
$$W(x_{1:n}) = \frac{p(x_{1:n}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})q(x_{n-L+1:n}|x_{n-L}, y_{1:n-L+1:n})}$$

- Optimally,

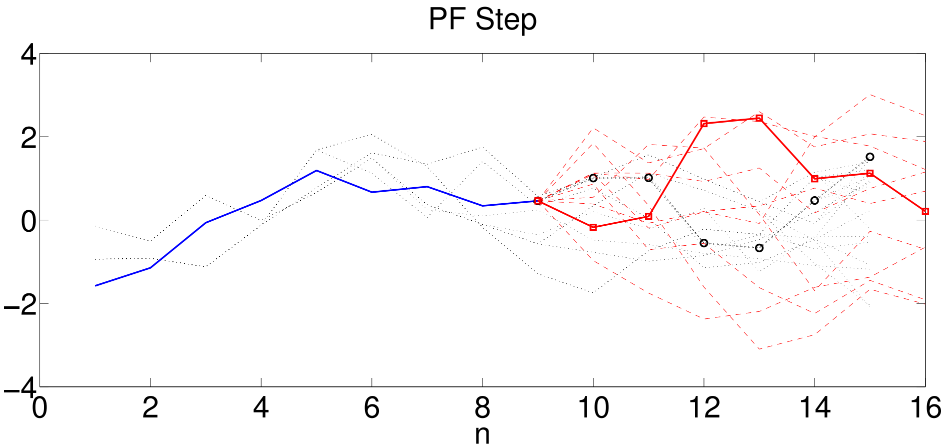
$$\begin{aligned} q(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n}) &= p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n}) \\ W(x_{1:n}) \propto \frac{p(x_{1:n-L}|y_{1:n})}{p(x_{1:n-L}|y_{1:n-1})} &= p(y_n|x_{1:n-L}, y_{n-L+1:n-1}) \end{aligned}$$

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

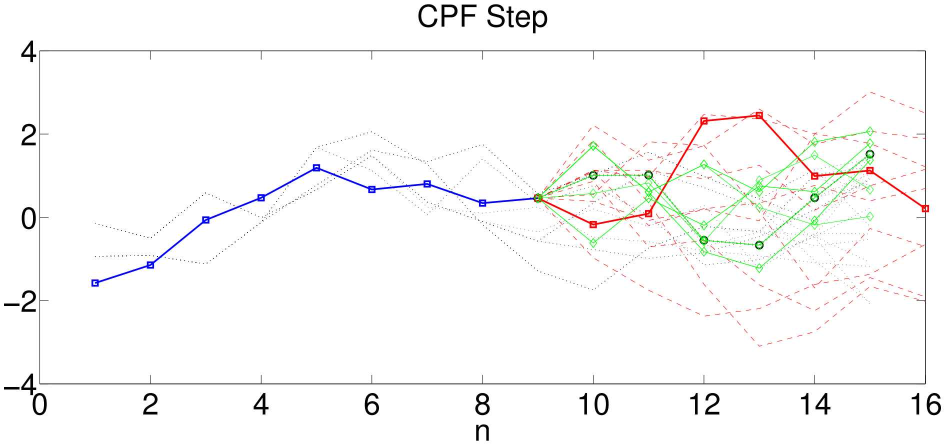
Local Particle Filtering: Current Trajectories



Local Particle Filtering: PF Proposal



Local Particle Filtering: CPF Auxiliary Proposal



Local SMC

- Not just a Random Weight Particle Filter.
- Propose from:

$$\mathcal{U}_{1:M}^{\otimes n-1}(b_{1:n-2}, k) p(x_{1:n-1}|y_{1:n-1}) \psi_{n,L}^{M}(\overline{\mathbf{a}}_{n-L+2:n}, \overline{\mathbf{x}}_{n-L+1:n}, \overline{k}; x_{n-L})$$
$$\widetilde{\psi}_{n-1,L-1}^{M}\left(\widetilde{\mathbf{a}}_{n-L+2:n-1}^{\ominus k}, \widetilde{\mathbf{x}}_{n-L+1:n-1}^{\ominus k}; x_{n-L} \,||\, b_{n-L+2:n-1}, x_{n-L+1:n-1}\right)$$

- Target:

$$\mathcal{U}_{1:M}^{\otimes n}(b_{1:n-L}, \overline{b}_{n,n-L+1:n-1}^{\bar{k}}, \bar{k}) p(x_{1:n-L}, \overline{x}_{n-L+1:n}^{\overline{b}_{n,n-L+1:n}^{\bar{k}}}|y_{1:n})$$
$$\widetilde{\psi}_{n,L}^{M}\left(\overline{\mathbf{a}}_{n-L+2:n}^{\ominus \overline{k}}, \overline{\mathbf{x}}_{n-L+1:n}^{\ominus \overline{k}}; x_{n-L} \,\middle|\middle|\, \overline{b}_{n,n-L+1:n}^{\bar{k}}, \overline{x}_{n-L+1:n}^{\overline{b}_{n,n-L+1:n}^{\bar{k}}}\right)$$
$$\psi_{n-1,L-1}^{M}\left(\widetilde{\mathbf{a}}_{n-L+2:n-1}, \widetilde{\mathbf{x}}_{n-L+1:n-1}, k; x_{n-L}\right).$$

- Weight: $\bar{Z}_{n-L+1:n} / \widetilde{Z}_{n-L+1:n-1}$.

Stochastic Volatility Bootstrap Local SMC

- Model:

$$f(x_i|x_{i-1}) = \mathcal{N}\left(\phi x_{i-1}, \sigma^2\right)$$
$$g(y_i|x_i) = \mathcal{N}\left(0, \beta^2 \exp(x_i)\right)$$

- Top Level:
  - Local SMC proposal.
  - Stratified resampling when ESS$< N/2$.
- Local SMC Proposal:
  - Proposal:

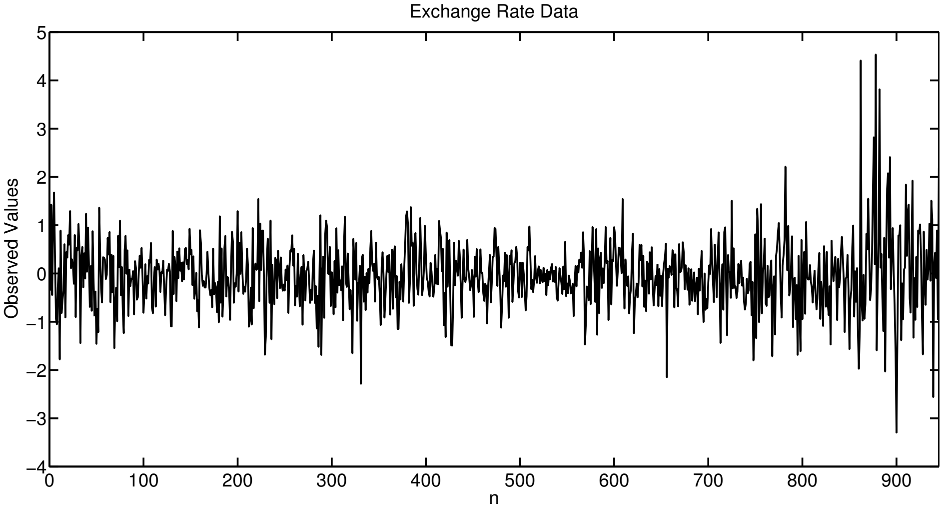
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

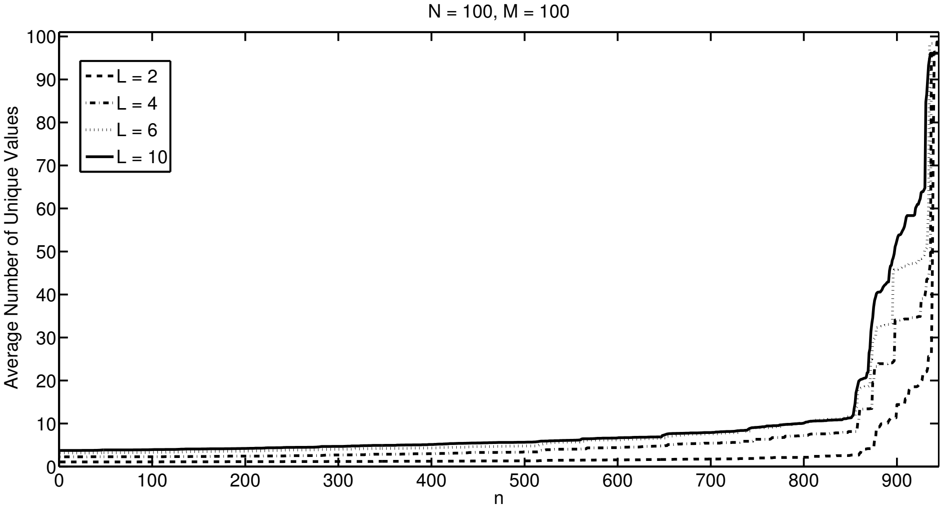
$$W(x_{t-1}, x_t) \propto \frac{f(x_t|x_{t-1})g(y_t|x_t)}{f(x_t|x_{t-1})} = g(y_t|x_t)$$

  - Resample residually every iteration.

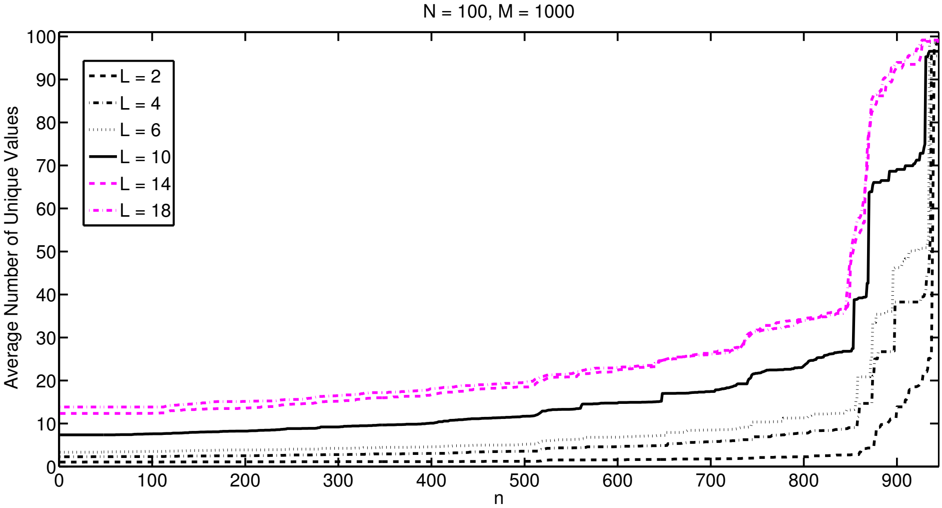
SV Exchange Rata Data



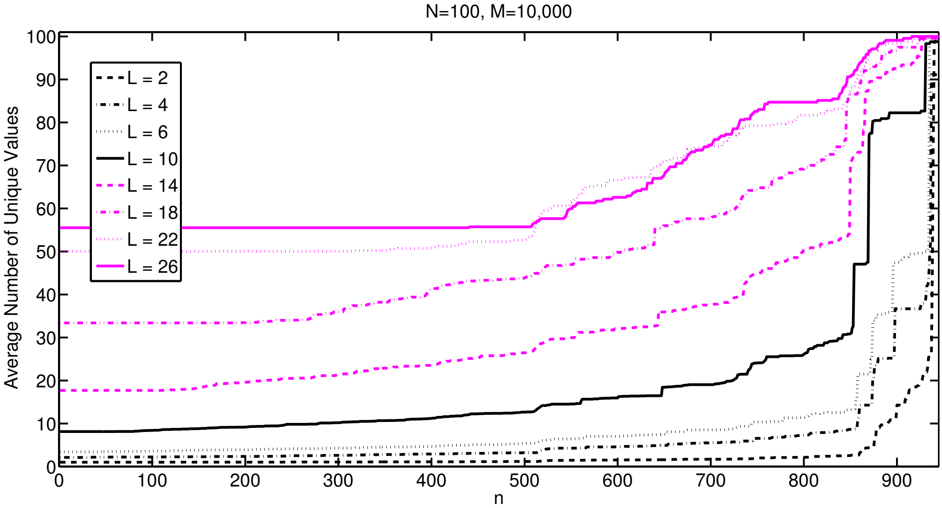
SV Bootstrap Local SMC: M=100



SV Bootstrap Local SMC: M=1000



SV Bootstrap Local SMC: M=10000



Some Heuristics

Recent calculations suggest that, under *appropriate* assumptions, at fixed cost $(2L-1) \cdot M \cdot N$:

- Optimal L is determined solely by the mixing of the HMM.
- Optimal M is a linear function of L.
- N can then be obtained from M, L and available budget.

In practice:

- L can be chosen using pilot runs,
- and M fine-tuned once L is chosen.

In Conclusion

- SMC can be used hierarchically.
- Software implementation is not difficult [Joh09, Zho13].
- The Rao-Blackwellized particle filter can be approximated *exactly*
- The optimal block-sampling particle filter can be approximated *exactly*
- Many other things are possible...

going beyond *unbiased random weighting*.

References I

[AD02] C. Andrieu and A. Doucet. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society B*, 64(4):827–836, 2002.

[ADH10] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo. *Journal of the Royal Statistical Society B*, 72(3):269–342, 2010.

[CJP13] N. Chopin, P. Jacob, and O. Papaspiliopoulos. SMC^2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society B*, 75(3):397–426, 2013.

[CL00] R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society B*, 62(3):493–508, 2000.

[CSOL11] T. Chen, T. Schön, H. Ohlsson, and L. Ljung. Decentralized particle filter with arbitrary state decomposition. *IEEE Transactions on Signal Processing*, 59(2):465–478, February 2011.

[DBS06] A. Doucet, M. Briers, and S. Sénécal. Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711, 2006.

[GSS93] N. J. Gordon, S. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.

[JD] A. M. Johansen and A. Doucet. The hierarchical particle filter. Unpublished Manuscript.

[Joh09] A. M. Johansen. SMCTC: Sequential Monte Carlo in C++. *Journal of Statistical Software*, 30(6):1–41, April 2009.

[JWD12] A. M. Johansen, N. Whiteley, and A. Doucet. Exact approximation of Rao-Blackwellised particle filters. In *Proceedings of 16th IFAC Symposium on Systems Identification*, Brussels, Belgium, July 2012. IFAC, IFAC.

[Zho13] Y. Zhou. vSMC: Parallel sequential Monte Carlo in C++. Mathematics e-print 1306.5583, ArXiv, 2013.

Key Identity

$$
\frac{\psi_{n,L}^{M}(\mathbf{a}_{n-L+2:n}, \mathbf{x}_{n-L+1:n}, k; x_{n-L})}{p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})\widetilde{\psi}_{n,L}^{M}(\mathbf{a}^{\ominus k}_{n-L+2:n}, \mathbf{x}^{\ominus k}_{n-L+1:n}, k; x_{n-L}\,||\ldots\,)}
$$

$$
= \frac{q\left(x_{n-L+1}^{b^k_{n,n-L+1}}|x_{n-L}\right)\left[\prod_{p=n-L+2}^{n} r\left(b^k_{n,p}|\mathbf{w}_{\mathbf{p-1}}\right) q\left(x_p^{b^k_{n,p}}|x_{p-1}^{b^n_{n,p-1}}\right)\right] r(k|\mathbf{w}_n)}{p(x_{n-L+1:n}|x_{n-L}, y_{n-L+1:n})}
$$

$$
= \widehat{Z}_{n-L+1:n}/p(y_{n-L+1:n}|x_{n-L})
$$