Designing "Particle Filters" for Particle MCMC

Adam M. Johansen
a.m.johansen@warwick.ac.uk
go.warwick.ac.uk/amjohansen/talks/

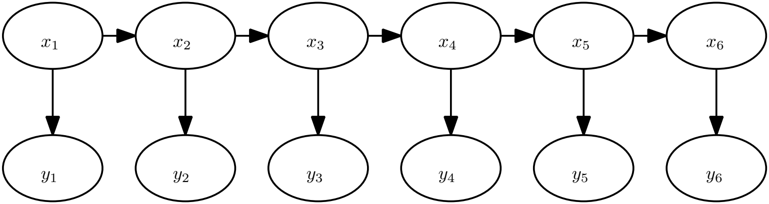
Oxford, March 4th, 2016

Outline

- Background: SMC and PMCMC
- Beyond simple filters:
  - Iterative Lookahead Methods
  - Tempering Blocks in SMC
  - Hierarchical Particle Filters
- Conclusions

Discrete Time Filtering

Online inference for Hidden Markov Models:



- Given *initial distribution* $\mu_\theta(x_1)$, *transition* $f_\theta(x_{n-1}, x_n)$,
- and *likelihood* $g_\theta(x_n, y_n)$,
- use $p_\theta(x_n|y_{1:n})$ to characterize latent state, but,

$$p_\theta(x_n|y_{1:n}) = \frac{\int p_\theta(x_{n-1}|y_{1:n-1}) f_\theta(x_{n-1}, x_n) dx_{n-1} g_\theta(x_n, y_n)}{\int\int p_\theta(x_{n-1}|y_{1:n-1}) f_\theta(x_{n-1}, x'_n) dx_{n-1} g_\theta(x'_n, y_n) dx'_n}$$

and other densities of interest are typically intractable.

Particle Filtering

A Simple "Bootstrap" Particle Filter (Gordon et al. [10])

At $n = 1$:

- Sample $X_1^1, \ldots, X_1^N \sim \mu_\theta$.

For $n > 1$:

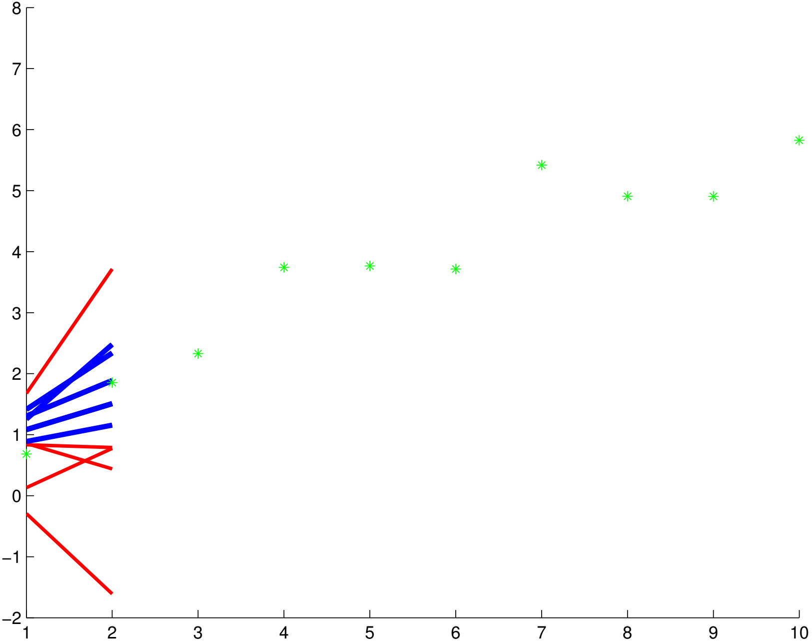
- Sample

$$A_n^1, \ldots, A_n^N \sim \frac{\sum_{j=1}^N g_\theta(X_{n-1}^j, y_{n-1})\delta_j(\cdot)}{\sum_{k=1}^n g_\theta(X_{n-1}^k, y_{n-1})}$$

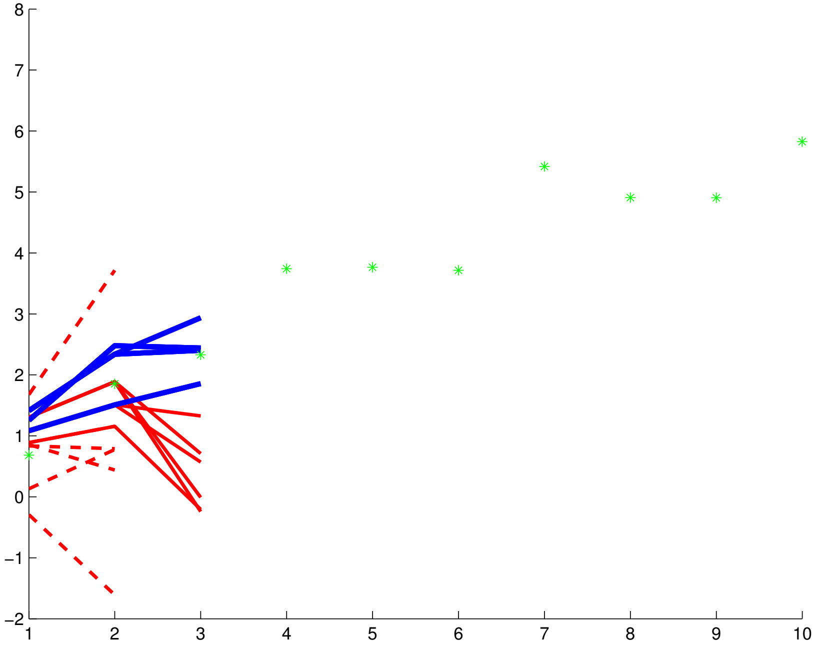
- For i in $1 : N$ sample $X_n^i \sim f_\theta(X_{n-1}^{A_n^i}, \cdot)$.
- Approximate $p_\theta(dx_n | y_{1:n}), p_\theta(y_{1:n})$ with

$$\widehat{p_\theta}(\cdot | y_{1:n}) = \frac{\sum_{j=1}^N g_\theta(X_n^j, y_n)\delta_{X_n^j}}{\sum_{k=1}^n g_\theta(X_n^k, y_n)}, \frac{\widehat{p_\theta}(y_{1:n})}{\widehat{p_\theta}(y_{1:n-1})} = \frac{1}{n} \sum_{j=1}^N g_\theta(X_n^k, y_n)$$

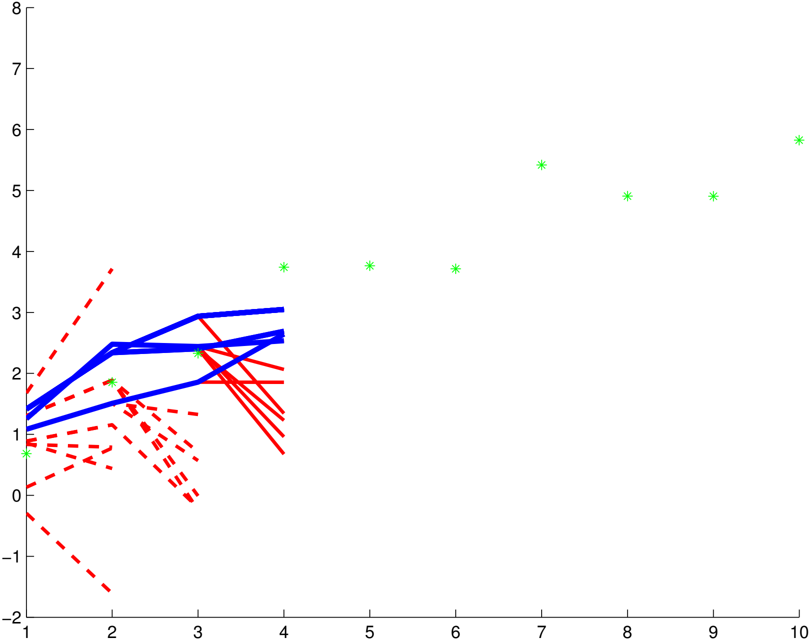
Iteration 2



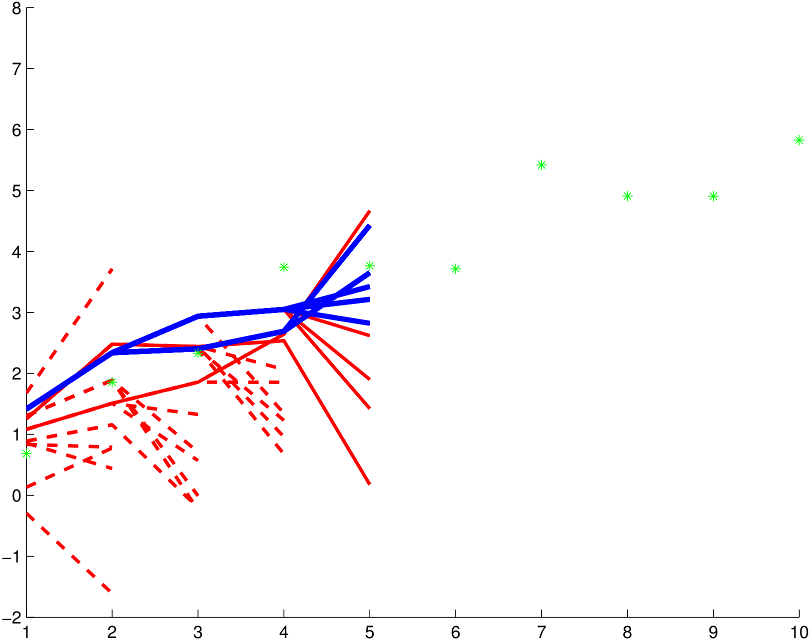
Iteration 3



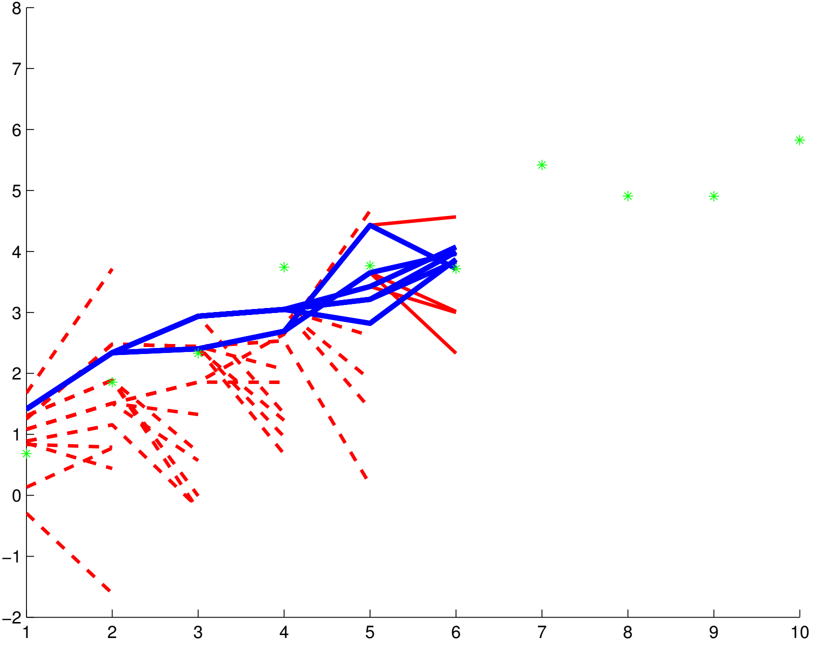
Iteration 4



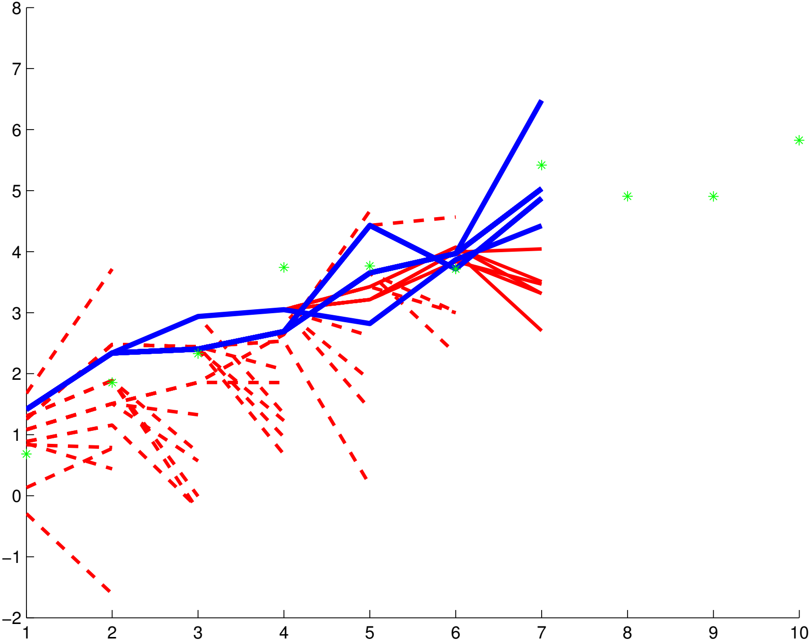
Iteration 5



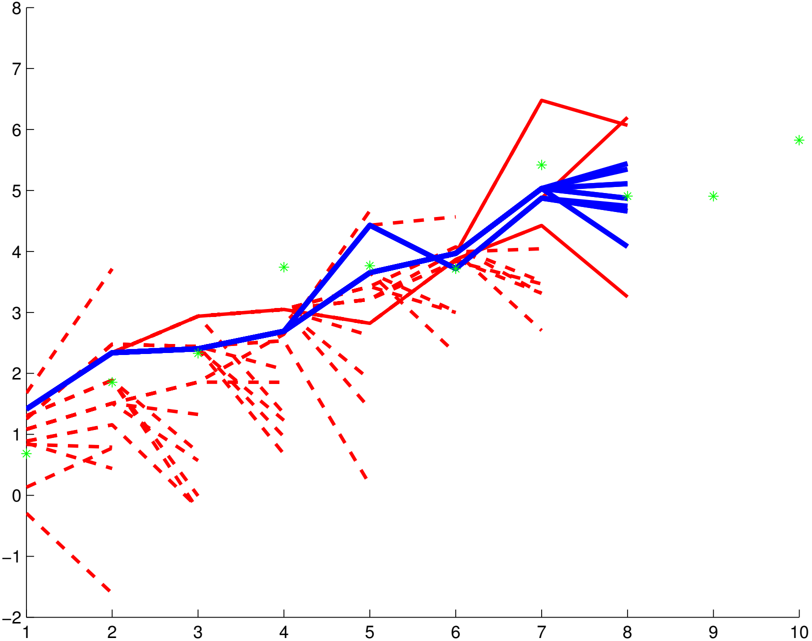
Iteration 6



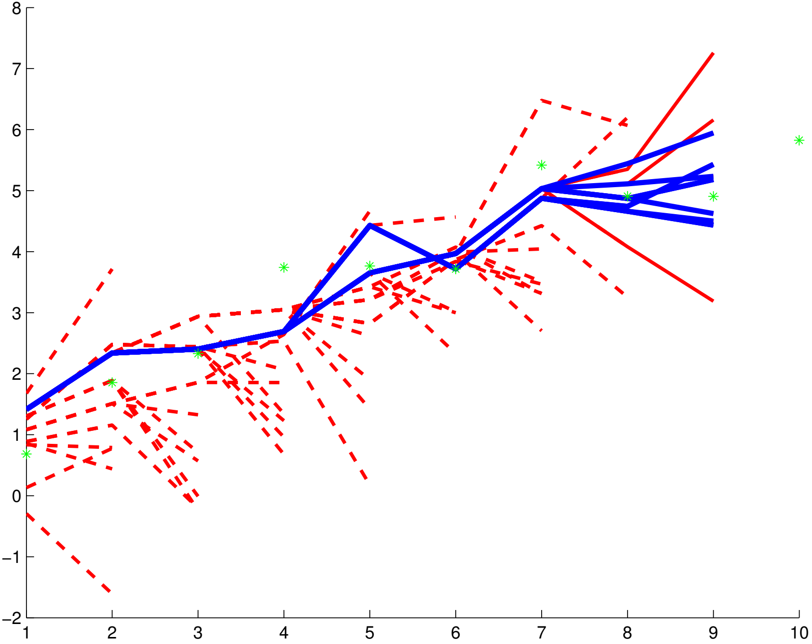
Iteration 7



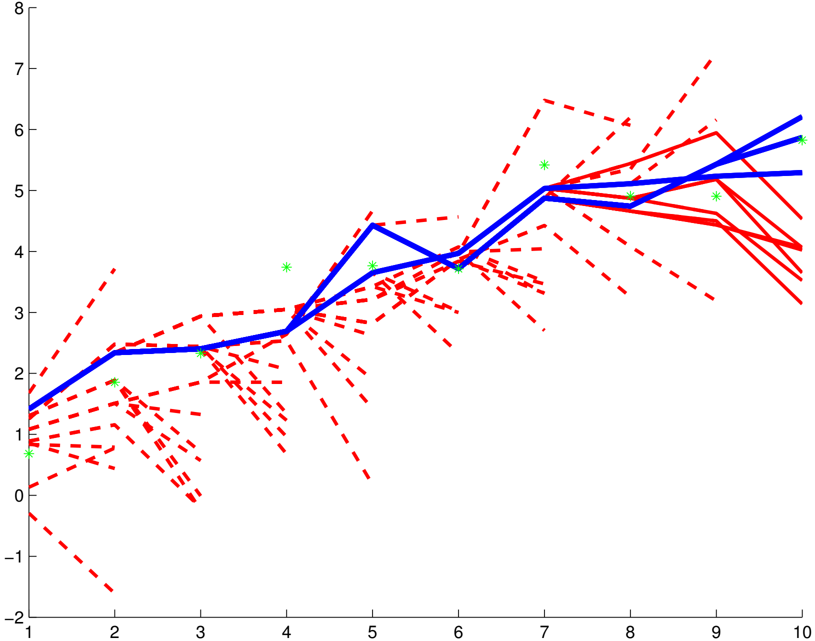
Iteration 8



Iteration 9



Iteration 10



Various Improvements to Particle Filters

- Better Proposal Distributions:
  - Locally optimal (cf. Doucet et al. [6]) case $q(x_n|x_{n-1}, y_n) \propto f_\theta(x_{n-1}, x_n)g(x_n, y_n)$.
  - Incremental importance weights $G_n(x_{n-1}, x_n) = f(x_{n-1}, x_n)g(x_n, y_n)/q(x_{n-1}, x_n)$.
- Auxiliary Particle Filters (Pitt and Shephard [15]) (see also Johansen and Doucet [13])
- Better Resampling Schemes (cf. Douc and Cappé [5])
- "Adaptive" Resampling (cf. Del Moral et al. [4])
- Resample-Move Schemes (MCMC; Gilks and Berzuini [8])
- Tempered Transitions (Godsill and Clapp [9])

All of these retain the *online* character of particle filtering.

Key Properties of Particle Filters

- Fundamentally online: can approximate $p_\theta(x_n|y_{1:n})$ at iteration n at constant cost per iteration.
- Yield good approximations of the filtering distribution:

$$p_\theta(x_n|y_{1:n})$$

  at time n.
- Can approximate the marginal likelihood

$$p_\theta(y_{1:n}) = \int p_\theta(x_{1:n}, y_{1:n}) dx_{1:n}$$

  unbiasedly.

Only the last of these is critical in an offline setting...

Online Particle Filters for Offline Estimation (via PMCMC)

Particle Markov chain Monte Carlo (Andrieu et al. [2])

- Embed SMC within MCMC,
- justified via explicit auxiliary variable construction,
- or in some cases by a pseudomarginal (Andrieu and Roberts [1]) argument.
- Very widely applicable,
- but prone to poor mixing when SMC performs poorly for some θ (Owen et al. [14, Section 4.2.1]).
- Is valid for *very* general SMC algorithms.

Iterative Lookahead Methods[1]: Motivation

- Online algorithms can only perform so well:

$$p(x_{1:n}|y_{1:T}) = \int p(x_{1:T}|y_{1:T})dx_{n+1:T} \neq p(x_{1:n}|y_{1:n})$$

- We'd benefit from targetting the smoothing distributions:

$$\tilde{\pi}_n(x_{1:n}) = p(x_{1:n}|y_{1:T}) \propto p(x_{1:n}|y_{1:n})p(y_{n+1:T}|x_n)$$

  in place of the filtering distributions:

$$\pi_n(x_{1:n}) = p(x_{1:n}|y_{1:n})$$

- But this is *really* hard: can we *approximate* $p(y_{n+1:T}|x_n)$?

[1] Joint work with Pieralberto Guarniero and Anthony Lee

Twisting the HMM (complements (Whiteley and Lee [16]))

Given (μ, f, g) and $y_{1:T}$, introducing $\boldsymbol{\psi} := (\psi_1, \psi_2, \ldots, \psi_T), \psi_t \in \mathcal{C}_b(\mathsf{X}, (0, \infty))$ and

$$\tilde{\psi}_0 := \int_{\mathsf{X}} \mu(x_1) \psi_1(x_1) \, dx_1 \quad \tilde{\psi}_t(x_t) := \int_{\mathsf{X}} f(x_t, x_{t+1}) \psi_{t+1}(x_{t+1}) \, dx_{t+1}$$

(and $\tilde{\psi}_T \equiv 1$) we obtain $(\mu_1^{\boldsymbol{\psi}}, \{f_t^{\boldsymbol{\psi}}\}, \{g_t^{\boldsymbol{\psi}}\})$, with

$$\mu_1^{\boldsymbol{\psi}}(x_1) := \frac{\mu(x_1)\psi_1(x_1)}{\tilde{\psi}_0}, \qquad f_t^{\boldsymbol{\psi}}(x_{t-1}, x_t) := \frac{f(x_{t-1}, x_t) \psi_t(x_t)}{\tilde{\psi}_{t-1}(x_{t-1})}$$

and the sequence of non-negative functions

$$g_1^{\boldsymbol{\psi}}(x_1) := g(x_1, y_1) \frac{\tilde{\psi}_1(x_1)}{\psi_1(x_1)} \tilde{\psi}_0, \qquad g_t^{\boldsymbol{\psi}}(x_t) := g(x_t, y_t) \frac{\tilde{\psi}_t(x_t)}{\psi_t(x_t)}.$$

Proposition

For any sequence of bounded, continuous and positive functions $\boldsymbol{\psi}$, let

$$Z_{\boldsymbol{\psi}} := \int_{\mathsf{X}^T} \mu_1^{\boldsymbol{\psi}}(x_1)\, g_1^{\boldsymbol{\psi}}(x_1) \prod_{t=2}^{T} f_t^{\boldsymbol{\psi}}(x_{t-1}, x_t)\, g_t^{\boldsymbol{\psi}}(x_t)\, dx_{1:T}.$$

Then, $Z_{\boldsymbol{\psi}} = p_\theta(y_{1:T})$ for any such $\boldsymbol{\psi}$.

The (variance) optimal choice is:

$$\psi_t^*(x_t) := g(x_t, y_t)\, \mathbb{E}\left[\prod_{p=t+1}^{T} g(X_p, y_p) \middle| \{X_t = x_t\} \right], \quad x_t \in \mathsf{X},$$

for $t \in \{1, \ldots, T-1\}$. Then, $Z_{\boldsymbol{\psi}^*}^N = p(y_{1:T})$ with probability 1.

ψ−Auxiliary Particle Filters (Guarniero et al. [11])

ψ-Auxiliary Particle Filter

1. Sample $\xi_1^i \sim \mu^{\psi}$ independently for $i \in \{1, \ldots, N\}$.
2. For $t = 2, \ldots, T$, sample independently

$$\xi_t^i \sim \frac{\sum_{j=1}^{N} g_{t-1}^{\psi}(\xi_{t-1}^j) f_t^{\psi}(\xi_{t-1}^j, \cdot)}{\sum_{j=1}^{N} g_{t-1}^{\psi}(\xi_{t-1}^j)}, \qquad i \in \{1, \ldots, N\}.$$

Necessary features of ψ

1. It is possible to sample from f_t^{ψ}.
2. It is possible to evaluate g_t^{ψ}.
3. To be useful: $\mathbb{V}(Z_{\psi}^N)$ must be small.

A Recursive Approximtion

Proposition

The sequence $\boldsymbol{\psi}^*$ *satisfies* $\psi_T^*(x_T) = g(x_T, y_T)$, $x_T \in \mathsf{X}$ *and*

$$\psi_t^*(x_t) = g(x_t, y_t)\, f\left(x_t, \psi_{t+1}^*\right), \quad x_t \in \mathsf{X}, \quad t \in \{1, \ldots, T-1\}.$$

Algorithm 1 Recursive function approximations

For $t = T, \ldots, 1$:

1. Set $\psi_t^i \leftarrow g\left(\xi_t^i, y_t\right) f\left(\xi_t^i, \psi_{t+1}\right)$ for $i \in \{1, \ldots, N\}$.
2. Choose ψ_t as a member of Ψ on the basis of $\xi_t^{1:N}$ and $\psi_t^{1:N}$.

Iterated Auxiliary Particle Filters (Guarniero et al. [11])

Algorithm 2 An iterated auxiliary particle filter (param's: N_0, k, τ)

1. Initialize: set ${\psi_t}^0 = \mathbf{1}$. $l \leftarrow 0$.
2. Repeat:
   - 2.1 Run a $\boldsymbol{\psi}^l$-APF with N_l particles; set $\hat{Z}_l \leftarrow Z^{N_l}_{\boldsymbol{\psi}^l}$.
   - 2.2 If $l > k$ and $\mathrm{sd}(\hat{Z}_{l-k:l})/\mathrm{mean}(\hat{Z}_{l-k:l}) < \tau$, go to 3.
   - 2.3 Compute $\boldsymbol{\psi}^{l+1}$ using Algorithm 1.
   - 2.4 If $N_{l-k} = N_l$ and the sequence $\hat{Z}_{l-k:l}$ is not monotonically increasing, set $N_{l+1} \leftarrow 2N_l$.
     Otherwise, set $N_{l+1} \leftarrow N_l$.
   - 2.5 Set $l \leftarrow l + 1$. Go to 2a.
3. Run a $\boldsymbol{\psi}^l$-APF. Return $\hat{Z} := Z^{N_l}_{\boldsymbol{\psi}}$.

An Elementary Implementation

Function Approximation

- Numerically obtain:

$$(m_t^*, \Sigma_t^*, \lambda_t^*) = \arg\min_{(m,\Sigma,\lambda)} \sum_{i=1}^{N} (\mathcal{N}(x_t^i, m, \Sigma) - \lambda \psi_t^i)^2$$

- Set:

$$\psi_t(x_t) := \mathcal{N}(x_t; m_t^*, \Sigma_t^*) + c(N, m_t^*, \Sigma_t^*).$$

Stopping Rule

- $k = 3$ or $k = 5$ in the following examples
- $\tau = 0.5$

Resampling

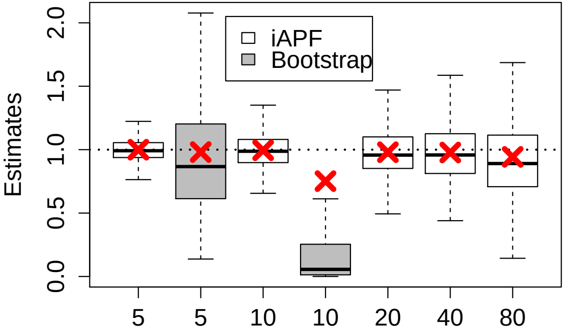
- Multinomial when ESS $< N/2$.

A Linear Gaussian Model: Behaviour with Dimension

$$\mu = \mathcal{N}(\cdot; \mathbf{0}, I_d) \qquad f(x, \cdot) = \mathcal{N}(\cdot; Ax, I_d)$$

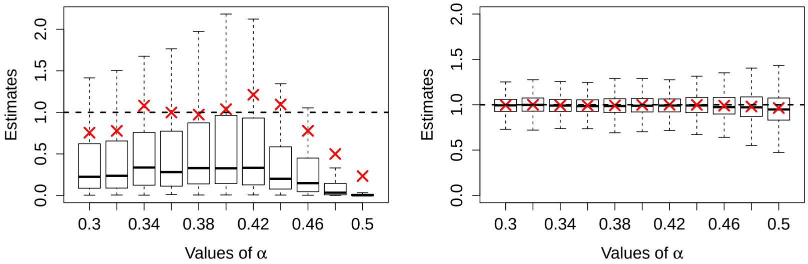
$$\text{and } g(x, \cdot) = \mathcal{N}(\cdot; x, I_d) \qquad \text{where } A_{ij} = 0.42^{|i-j|+1},$$

Box plots of $\hat{Z}/Z$ for different dim(X) (1000 replicates; $T = 100$).



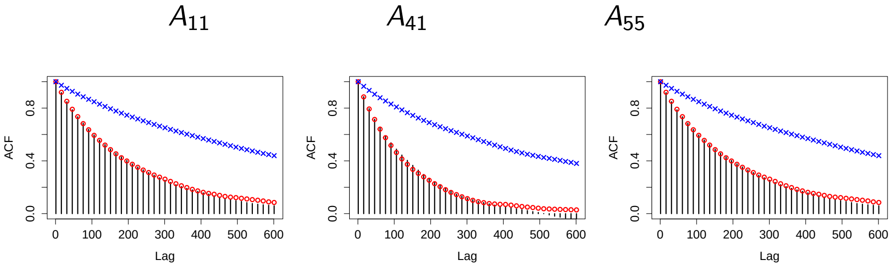
Linear Gaussian Model: Sensitivity to Parameters

Fixing $d = 10$: Bootstrap ($N = 50,000$) / iAPF ($N_0 = 1,000$)



Box plots of $\frac{\hat{Z}}{Z}$ for different values of the parameter α using 1000 replicates.

Linear Gaussian Model: PMMH Empirical Autocorrelations



In this case:

$$\begin{aligned} d &= 5 \\ \mu &= \mathcal{N}\left(\cdot; \mathbf{0}, I_d\right) \\ f(x,\cdot) &= \mathcal{N}\left(\cdot; Ax, I_d\right) \\ \text{and } g(x,\cdot) &= \mathcal{N}\left(\cdot; x, 0.25 I_d\right) \end{aligned}$$

$$A = \begin{pmatrix} 0.9 & 0 & 0 & 0 & 0 \\ 0.3 & 0.7 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0.6 & 0 & 0 \\ 0.4 & 0.1 & 0.1 & 0.3 & 0 \\ 0.1 & 0.2 & 0.5 & 0.2 & 0 \end{pmatrix},$$

treated as unknown lower triangular matrix.

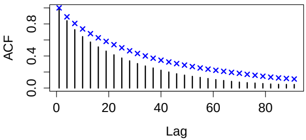
Stochastic Volatility

- A simple stochastic volatility model is defined by:

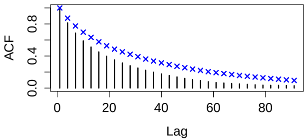
$$\begin{aligned} \mu(\cdot) =& \mathcal{N}(\cdot; 0, \sigma^2/(1-\alpha)^2) \\ f(x, \cdot) =& \mathcal{N}(\cdot; \alpha x, \sigma^2) \\ \text{and } g(x, \cdot) =& \mathcal{N}(\cdot; 0, \beta^2 \exp(x)), \end{aligned}$$

  where $\alpha \in (0, 1)$, $\beta > 0$ and $\sigma^2 > 0$ are unknown.
- Considered $T = 945$ observations $y_{1:T}$ corresponding to the mean-corrected daily returns for the GBP/USD exchange rate from 1/10/81 to 28/6/85.

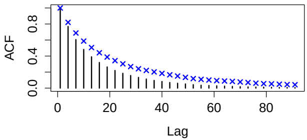
Estimated PMCMC Autocorrelation



a

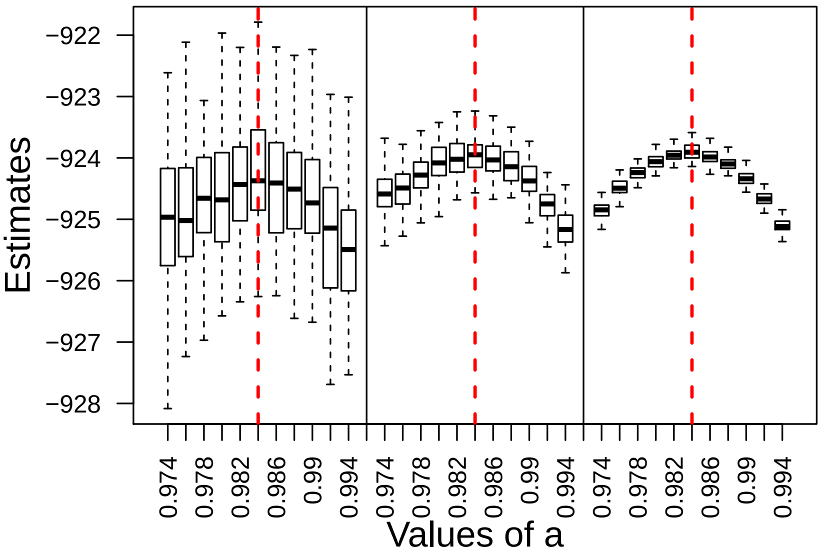


σ

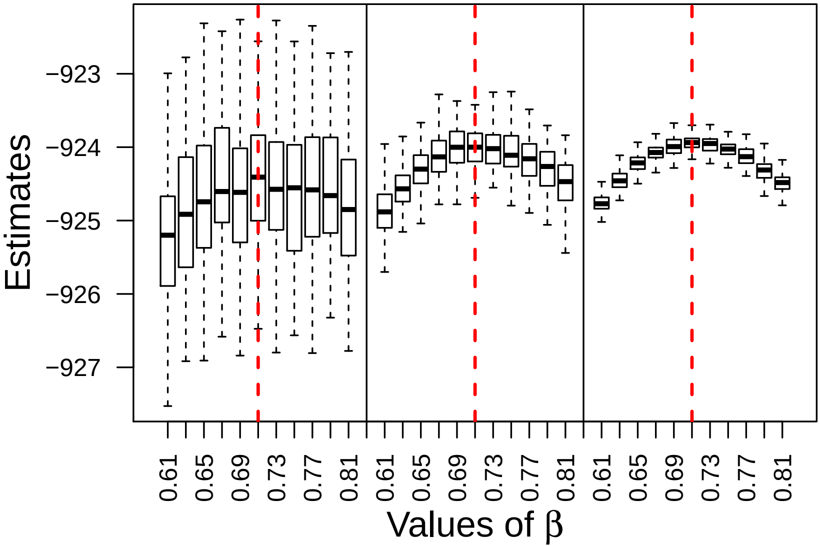


β

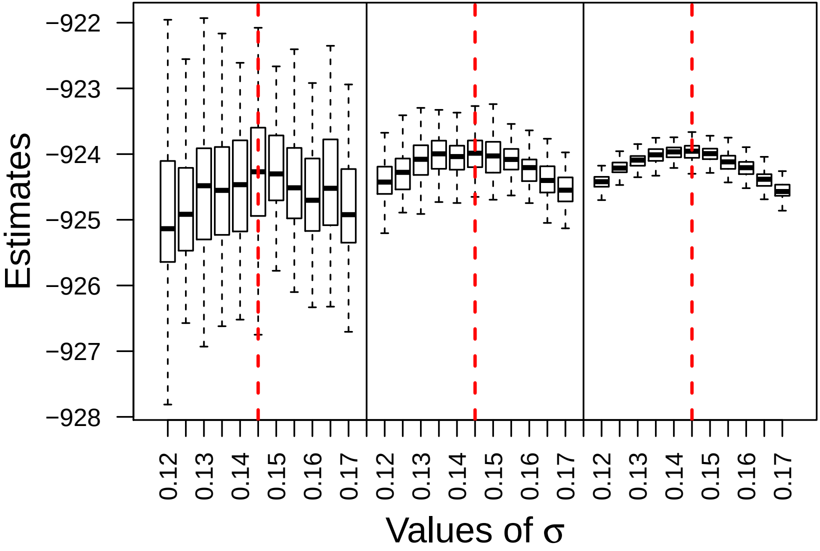
Boostrap $N = 1,000$.
iAPF $N_0 = 100$.
Comparable cost.
150,000 PMCMC iterations.



$Bootstrap : N = 1,000 \ / \ N = 10,000 \ / \ iAPF, N_0 = 100$



$Bootstrap : N = 1,000 \ / \ N = 10,000 \ / \ iAPF, N_0 = 100$



$Bootstrap : N = 1,000\ /\ N = 10,000\ /\ iAPF, N_0 = 100$

Block-Sampling and Tempering in Particle Filters

Tempered Transitions (Godsill and Clapp [9])

- Introduce each likelihood term gradually, targetting:

$$\pi_{n,m}(x_{1:n}) \propto p(x_{1:n}|y_{1:n-1})p(y_n|x_n)^{\beta_m}$$

  between $p(x_{1:n-1}|y_{1:n-1})$ and $p(x_{1:n}|y_{1:n})$.
- Can improve performance — but to a limited extent.

Block Sampling (Doucet et al. [7])

- Essentially uses $y_{n+1:n+L}$ in proposing x_{n+1}.
- Can *dramatically* improve performance,
- but requires good analytic approximation of $p(x_{n+1:n+L}|x_n, y_{n+1:n+L})$.

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 [7].

Block-Tempering (Johansen [12])

- We could combine blocking and tempering strategies.
- Run a simple SMC sampler (Del Moral et al. [3]) targetting:

$$\pi^{\theta}_{t,r}(x_{1:t\wedge T}) = \mu_\theta(x_1)^{\beta^1_{(t,r)}} g_\theta(y_1|x_1)^{\gamma^1_{(t,r)}} \cdot \prod_{s=2}^{T\wedge t} f_\theta(x_s|x_{s-1})^{\beta^s_{(t,r)}} g_\theta(y_s|x_s)^{\gamma^s_{(t,r)}}, \tag{1}$$

  where $\{\beta^s_{(t,r)}\}$ and $\{\gamma^s_{(t,r)}\}$ are $[0,1]$-valued
- for $s \in [\![1, T]\!]$, $r \in [\![1, R]\!]$ and $t \in [\![1, T']\!]$, with $T' = T + L$
- for some $R, L \in \mathbb{N}$.
- Can be *validly* embedded within PMCMC:
  - Terminal likelihood estimate is unbiased.
  - Explicit auxiliary variable construction is possible.

Two Simple Block-Tempering Strategies

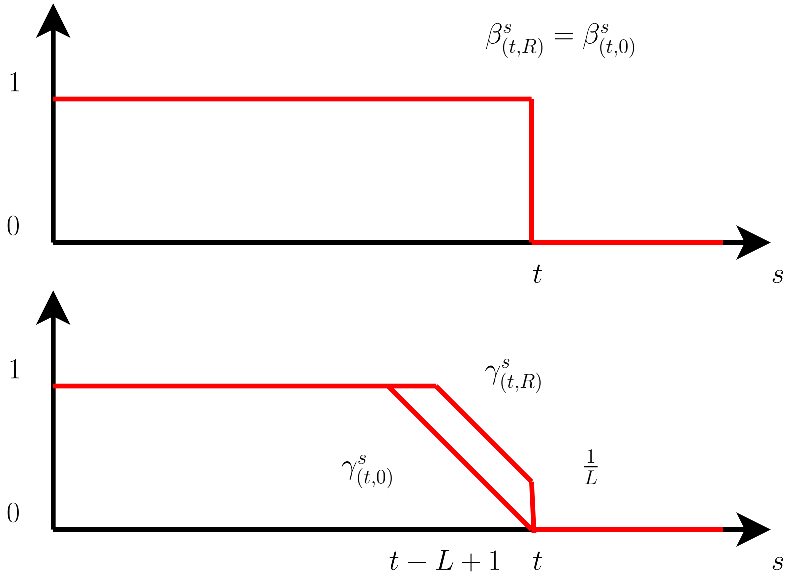
Tempering both likelihood and transition probabilities

$$\beta^s_{(t,r)} = \gamma^s_{(t,r)} = \left(1 \wedge \frac{R(t-s)+r}{RL}\right) \vee 0 \tag{2}$$

Tempering only the observation density

$$\beta^s_{(t,r)} = \mathbb{I}\{s \leq t\} \qquad \gamma^s_{(t,r)} = \left(1 \wedge \frac{R(t-s)+r}{RL}\right) \vee 0, \tag{3}$$

Tempering Only the Observation Density

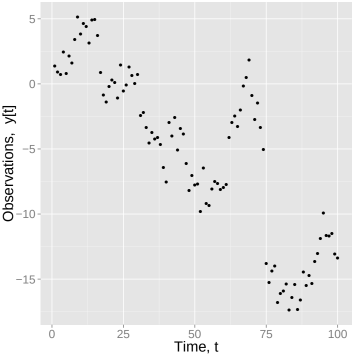


Illustrative Example

- Univariate Linear Gaussian SSM:

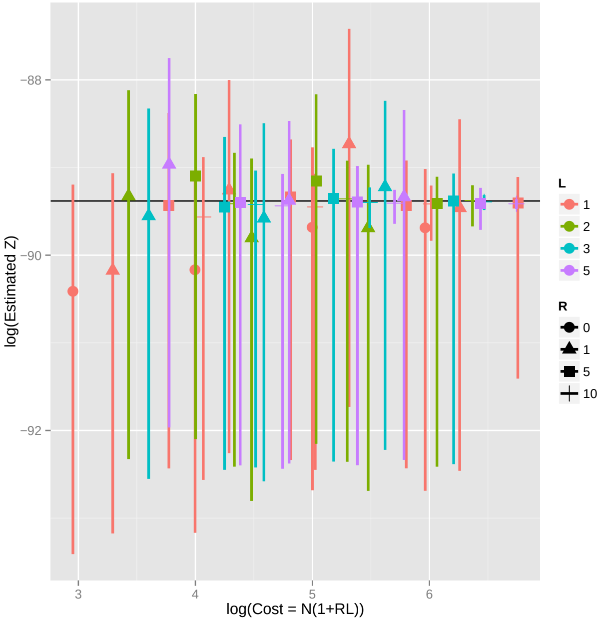
$$\begin{aligned} \text{transition} \quad f(x'|x) =& \mathcal{N}(x'; x, 1) \\ \text{likelihood} \quad g(y|x) =& \mathcal{N}(y; x, 1) \end{aligned}$$

- Artificial jump in observation seqeuence at time 75.
- Cartoon of *model misspecification*
- — a key difficulty with PMCMC.
- Temper only likelihood.
- Use single-site Metropolis-Within Gibbs (standard normal proposal) MCMC moves.

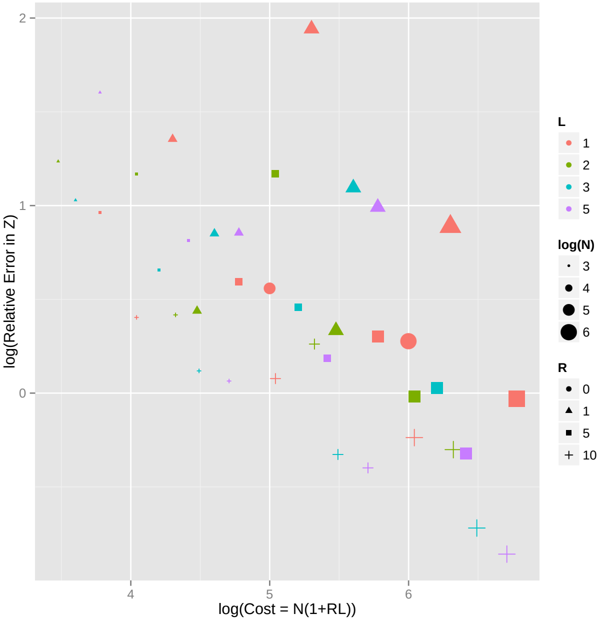


True Filtering and Smoothing Distributions

Estimating the Normalizing Constant, $\widehat{Z}$



Relative Error in $\widehat{Z}$ Against Computational Effort



Performance:

$$\begin{aligned} &(R = 1, L = 1) \\ \prec &(R > 1, L = 1) \\ \prec &(R > 1, L > 1) \end{aligned}$$

Hierarchical Particle Filters[2]

- (Optimal) block sampler requires samples from (approximately)

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

- and evaluations of

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

- Particle filters can provide sample approximations of

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

- and of

$$p(y_{n-L+1:n}|x_{n-L}).$$

- Can we use particle filters hierarchically?

[2] Joint work with Arnaud Doucet. . .

SMC Distributions

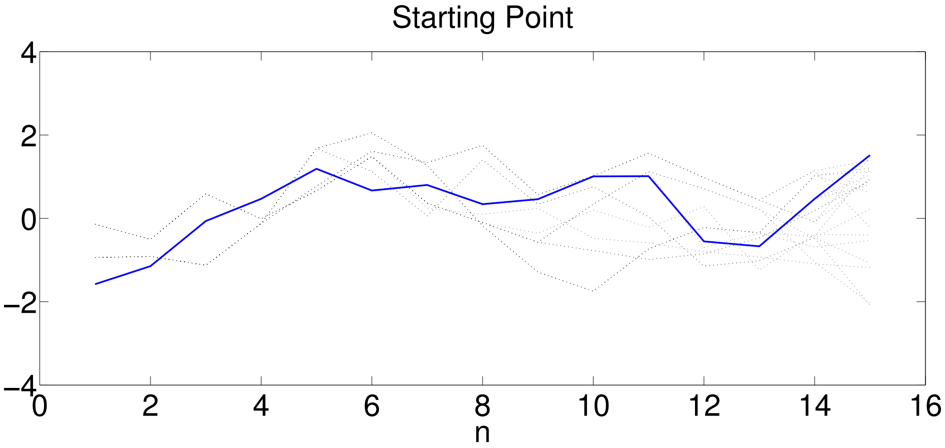
Formally gives rise to the **SMC Distribution**:

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

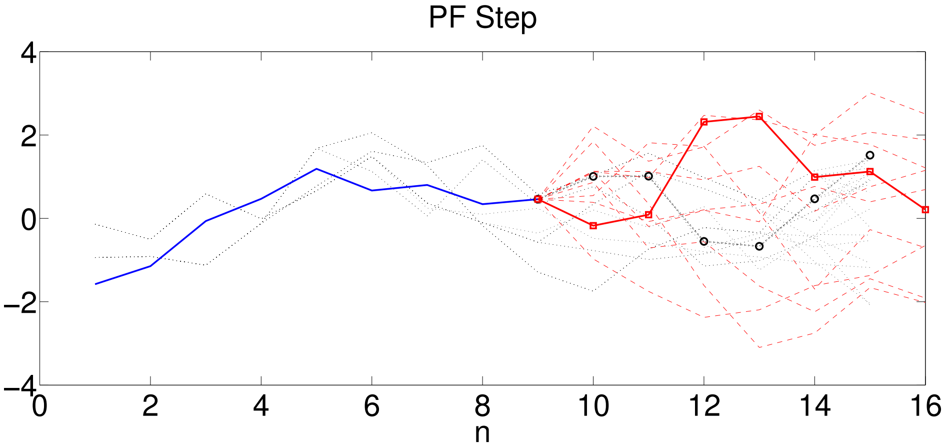
and the **conditional SMC Distribution**:

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

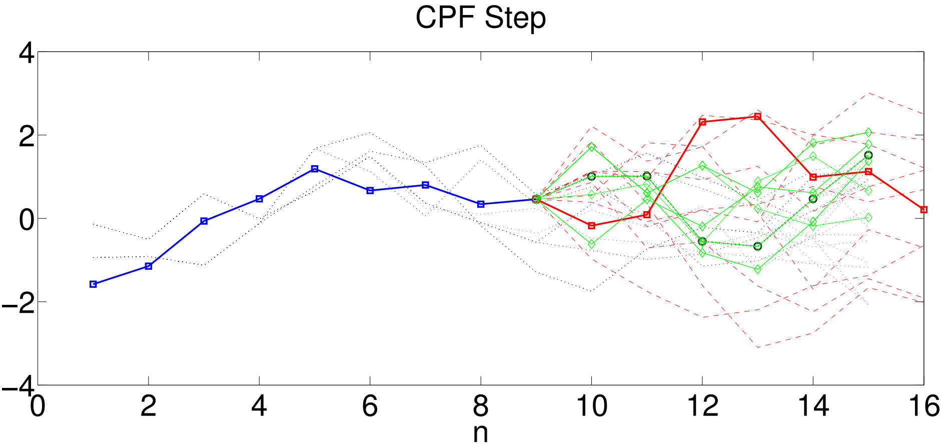
Local Particle Filtering: Current Trajectories



Local Particle Filtering: PF Proposal



Local Particle Filtering: CPF Auxiliary Proposal



Local SMC: Version 1

- 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:

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

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

Local SMC: Version 2

Problems with this PF+CPF scheme:

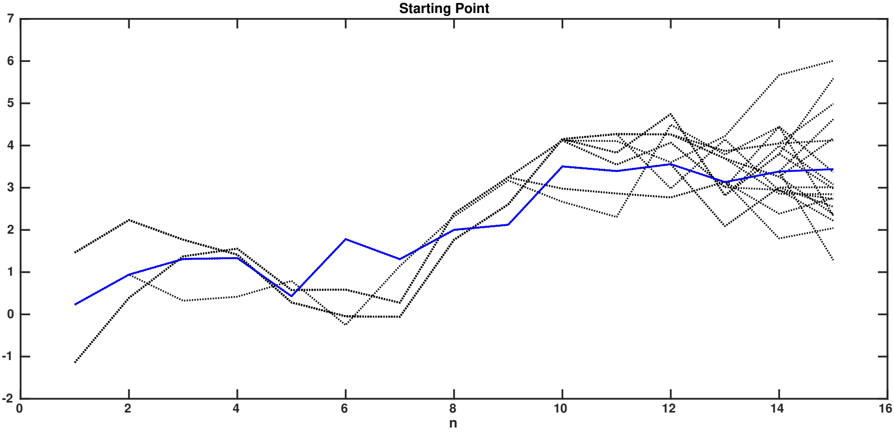
- Expensive to run *2 filters* per proposal...
- and large M is required...
- can't we do better?

Using a non-standard CPF/PF proposal is preferable:

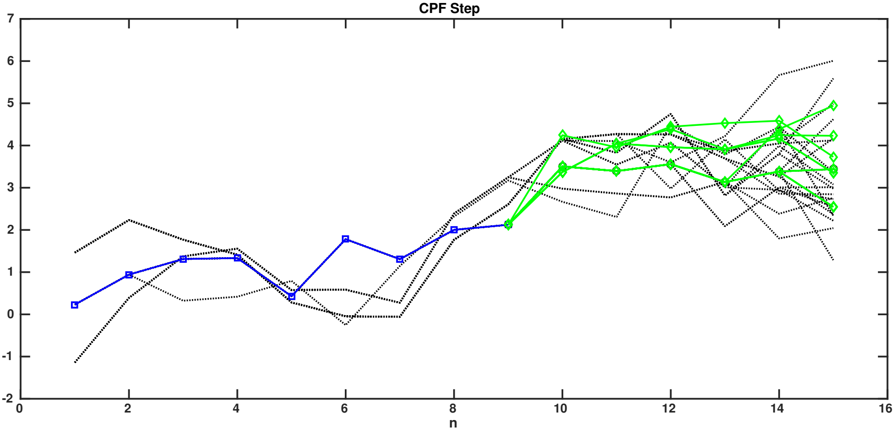
- Running a CPF from time $n - L + 1$ to n,
- and extending it's paths with a PF step to $n + 1$,
- invoking a careful auxiliary variable construction,
- we reduce computational cost *and* variance.
- Leads to importance weights:

$$\propto \frac{\hat{p}(y_{n-L+1:n}|x_{n-L}^{b^k_{n-1,n-L}})}{\hat{p}(y_{n-L+1:n-1}|x_{n-L}^{b^k_{n-1,n-L}})}$$

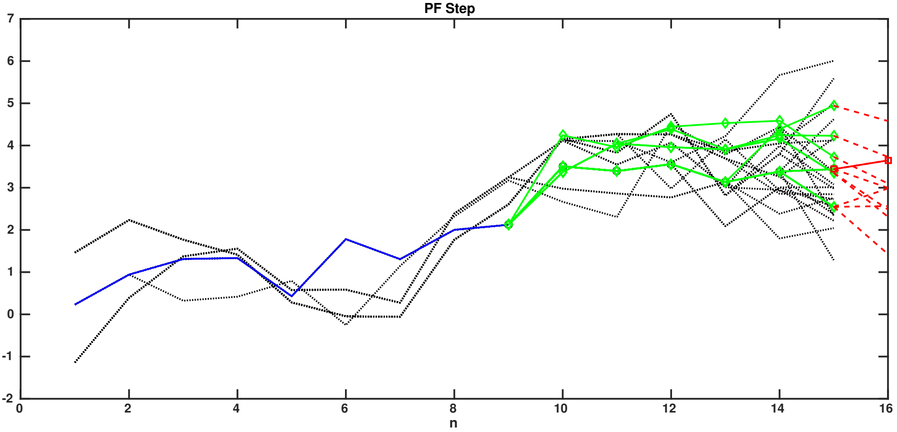
Hierarchical Particle Filtering: Current Trajectories



Hierarchical Particle Filtering: CPF Proposal Component



Hierarchical Particle Filtering: PF Proposal Component



A Toy Model: Linear Gaussian HMM

- Linear, Gaussian state transition:

$$f(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1}, 1)$$

- and likelihood

$$g(y_t|x_t) = \mathcal{N}(y_t; x_t, 1)$$

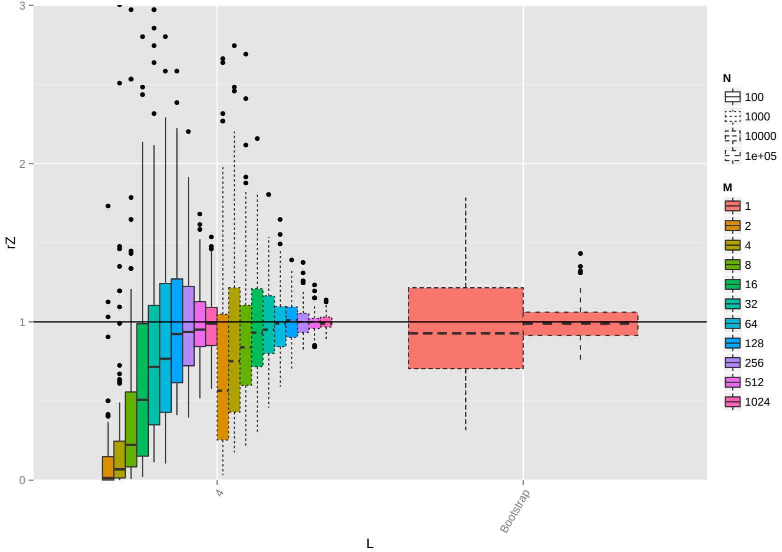
- Analytically: Kalman filter/smoother/etc.

- Simple bootstrap HPF:
  - Local proposal:

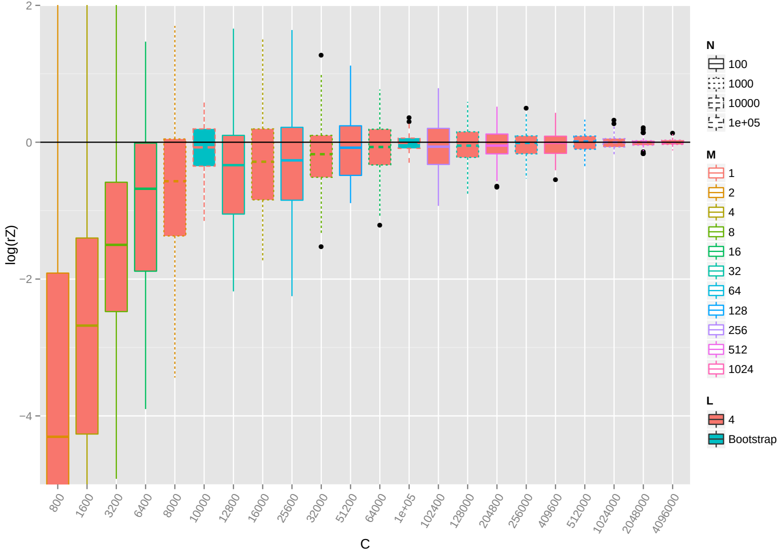
$$q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$$

  - Weighting:

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



3
2
1
0
rZ
4
Bootstrap
L
N
100
1000
10000
1e+05
M
1
2
4
8
16
32
64
128
256
512
1024



log(rZ)
2
0
−2
−4
C
800
1600
3200
6400
8000
10000
12800
16000
25600
32000
51200
64000
1e+05
102400
128000
204800
256000
409600
512000
1024000
2048000
4096000
N
100
1000
10000
1e+05
M
1
2
4
8
16
32
64
128
256
512
1024
L
4
Bootstrap

Conclusions

- To fully realise the potential of PMCMC we should exploit its flexibility.
- Even very simple variants on the standard particle filter can significantly improve performance.
- Three particular possibilities were discussed here:
  - iAPF
    - The iAPF can improve performance substantially in some settings.
    - Extending the extent of its applicability is ongoing work.
    - In principle any *function approximation* scheme can be employed: provided that f_t^{ψ} can be sampled from and g_t^{ψ} evaluated.
  - Block-Tempering
    - Can significantly improve performance with misspecified models.
    - Requires MCMC kernels and introduces 2 tuning parameters
  - Hierarchical Particle Filters
    - Dramatically reduce the need for resampling.
    - Computationally rather costly.

References

[1] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Annals of Statistics*, 37(2):697–725, 2009.

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

[3] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B*, 63(3):411–436, 2006.

[4] P. Del Moral, A. Doucet, and A. Jasra. On adaptive resampling procedures for sequential Monte Carlo methods. *Bernoulli*, 18(1), 2012. URL `http://hal.inria.fr/inria-00332436/PDF/RR-6700.pdf`.

[5] R. Douc and O. Cappé. Comparison of resampling schemes for particle filters. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, volume I, pages 64–69. IEEE, 2005.

[6] A. Doucet, S. Godsill, and C. Andrieu. On sequential simulation-based methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[7] 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.

[8] W. R. Gilks and C. Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society B*, 63: 127–146, 2001.

[9] S. Godsill and T. Clapp. Improvement strategies for Monte Carlo particle filters. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, pages 139–158. Springer Verlag, New York, 2001.

[10] 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.

[11] P. Guarniero, A. M. Johansen, and A. Lee. The iterated auxiliary particle filter. ArXiv mathematics e-print 1511.06286, ArXiv Mathematics e-prints, 2015.

[12] A. M. Johansen. On blocks, tempering and particle MCMC for systems identification. In Y. Zhao, editor, *Proceedings of 17th IFAC Symposium on System Identification*, pages 969–974, Beijing, China., 2015. IFAC. Invited submission.

[13] A. M. Johansen and A. Doucet. A note on the auxiliary particle filter. *Statistics and Probability Letters*, 78(12):1498–1504, September 2008. URL `http://dx.doi.org/10.1016/j.spl.2008.01.032`.

[14] J. Owen, D. J. Wilkinson, and C. S. Gillespie. Likelihood free inference for markov processes: a comparison. *Statistical Applications in Genetics and Molecular Biology*, 2015. In press.

[15] M. K. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[16] Nick Whiteley and Anthony Lee. Twisted particle filters. *Annals of Statistics*, 42 (1):115–141, 2014.