High-dimensional inference using nested particle filters
— Nested sequential Monte Carlo methods

Christian A. Naesseth

Joint work with Fredrik Lindsten and Thomas B. Schön
1 Background – sequential Monte Carlo
2 2D MRF – nested SMC applied
3 Experiments – spatio-temporal MRFs
Nonlinear filtering

The filtering problem for a nonlinear state space model,

\[ x_{t+1} | x_t \sim f(x_{t+1} | x_t), \]
\[ y_t | x_t \sim g(y_t | x_t), \]

amounts to computing

\[
p(x_t | y_{1:t}) = \frac{g(y_t | x_t) \int f(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1}}{p(y_t | y_{1:t-1})}
\]

for \( t = 1, 2, \ldots \)
The bootstrap filter

The bootstrap particle filter approximates $p(x_t | y_{1:t})$ by

$$\hat{P}^N (x_t | y_{1:t}) := \sum_{i=1}^N \frac{W_t^i}{\sum_{\ell} W_t^\ell} \delta X_t^i (x_t).$$
The bootstrap filter

The bootstrap particle filter approximates \( p(x_t | y_{1:t}) \) by

\[
\hat{p}^N(x_t | y_{1:t}) := \sum_{i=1}^{N} \frac{W_t^i}{\sum_{\ell} W_t^\ell} \delta_x(x_t)^i.
\]

- **Resampling:** \( \{(X_{t-1}^i, W_{t-1}^i)\}_{i=1}^{N} \rightarrow \{(\tilde{X}_{t-1}^i, 1/N)\}_{i=1}^{N} \).
The bootstrap filter

The bootstrap particle filter approximates \( p(x_t | y_{1:t}) \) by

\[
\hat{p}^N(x_t | y_{1:t}) := \sum_{i=1}^{N} \frac{W^i_t}{\ell W^\ell_t} \delta_{X^i_t}(x_t).
\]

- **Resampling:** \( \{(X^i_{t-1}, W^i_{t-1})\}_{i=1}^{N} \rightarrow \{(\tilde{X}^i_{t-1}, 1/N)\}_{i=1}^{N} \).  
- **Propagation:** \( X^i_t \sim f(x_t | \tilde{X}^i_{t-1}) \).
The bootstrap filter

The *bootstrap particle filter* approximates \( p(x_t \mid y_{1:t}) \) by

\[
\hat{p}^N(x_t \mid y_{1:t}) := \sum_{i=1}^{N} \frac{W_i^t}{\sum_{\ell} W_{\ell}^t} \delta_X(x_t).
\]

- **Resampling**: \( \{ (X_{t-1}^i, W_{t-1}^i) \}_{i=1}^N \rightarrow \{ (\tilde{X}_{t-1}^i, 1/N) \}_{i=1}^N \).
- **Propagation**: \( X_t^i \sim f(x_t \mid \tilde{X}_{t-1}^i) \).
- **Weighting**: \( W_t^i = g(y_t \mid X_t^i) \).

\[
\Rightarrow \{ (X_t^i, W_t^i) \}_{i=1}^N
\]
Particle filters in high dimension

• Known to perform poorly in high (say, $d \gtrsim 10$) dimensions.
Particle filters in high dimension

- Known to perform poorly in high (say, $d \gtrsim 10$) dimensions.
- ex) Spatio-temporal model: $g(y_t \mid x_t) = \prod_{k=1}^{d} g(y_{t,k} \mid x_{t,k})$. 

![Diagram of a spatio-temporal model with nodes $X_1$ to $X_6$ and arrows indicating the flow of information.](image-url)
Particle filters in high dimension

- Known to perform poorly in high (say, \( d \gtrsim 10 \)) dimensions.
- \textit{ex}) Spatio-temporal model: \( g(y_t | x_t) = \prod_{k=1}^{d} g(y_{t,k} | x_{t,k}) \).
  \[
  \begin{align*}
  X_1 & \quad X_2 & \quad X_3 & \quad X_4 & \quad X_5 & \quad X_6 \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  & \quad & \quad & \quad & \quad & \quad \\
  \end{align*}
  \]

- \( f(x_t | x_{t-1}) \) is typically an \textit{extremely} bad proposal distribution in HD.
Particle filters in high dimension

- Known to perform poorly in high (say, $d \gtrsim 10$) dimensions.
- *ex* Spatio-temporal model: $g(y_t \mid x_t) = \prod_{k=1}^{d} g(y_{t,k} \mid x_{t,k})$.

- $f(x_t \mid x_{t-1})$ is typically an *extremely* bad proposal distribution in HD.

Does a better proposal distribution improve our result?
Preview of the idea

- Optimal *proposals* and *resampling weights* — known conceptually but intractable to compute.
- Deterministic (e.g., Gaussian) approximations available, but often inadequate in high dimensions.
Preview of the idea

- Optimal *proposals* and *resampling weights* — known conceptually but intractable to compute.
- Deterministic (e.g., Gaussian) approximations available, but often inadequate in high dimensions.

Idea behind Nested SMC:
- Use *SMC* to approximate the optimal proposals and resampling weights.
Preview of the idea

- Optimal *proposals* and *resampling weights* — known conceptually but intractable to compute.
- Deterministic (e.g., Gaussian) approximations available, but often inadequate in high dimensions.

Idea behind Nested SMC:

- Use *SMC* to approximate the optimal proposals and resampling weights.
- Sampling distribution not available on closed form — still possible to obtain a valid algorithm!
Preview of the idea

- Optimal *proposals* and *resampling weights* — known conceptually but intractable to compute.
- Deterministic (e.g., Gaussian) approximations available, but often inadequate in high dimensions.

Idea behind Nested SMC:

- Use *SMC* to approximate the optimal proposals and resampling weights.
- Sampling distribution not available on closed form — still possible to obtain a valid algorithm!
- Nested SMC satisfies the conditions on the proposal approximation ⇒ possible to use within itself (nesting to arbitrary degree).
Fully adapted auxiliary SMC sampler

Let $\bar{\pi}_t(x_{1:t}) = \mathcal{Z}_t^{-1} \pi_t(x_{1:t})$ for $t = 1, 2, \ldots$ be a sequence of target distributions.
Fully adapted auxiliary SMC sampler

Let $\bar{\pi}_t(x_{1:t}) = Z_t^{-1} \pi_t(x_{1:t})$ for $t = 1, 2, \ldots$ be a sequence of target distributions.

- **Optimal proposal:**
  
  $\bar{q}_t(x_t \mid x_{1:t-1}) = Z_t^{-1}(x_{1:t-1})q_t(x_t \mid x_{1:t-1})$, where

  $$
  q_t(x_t \mid x_{1:t-1}) := \frac{\pi_t(x_{1:t})}{\pi_{t-1}(x_{1:t-1})} \quad [= g(y_t \mid x_t)f(x_t \mid x_{t-1})]
  $$

- **Optimal resampling weights:**

  $\tilde{W}_i^t := Z_t(x_{1:t}) = p(y_t \mid x_{t-1})$ results in an unweighted set of particles $\{X_i\}_{i=1}^N$, such that $\bar{\pi}_N(x_{1:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{X_i(x_{1:t})}$ approximates $\bar{\pi}_t(x_{1:t})$. 


Fully adapted auxiliary SMC sampler

Let \( \bar{\pi}_t(x_{1:t}) = Z_t^{-1} \pi_t(x_{1:t}) \) for \( t = 1, 2, \ldots \) be a sequence of target distributions.

- **Optimal proposal:**
  \[
  \bar{q}_t(x_t | x_{1:t-1}) = Z_t^{-1}(x_{1:t-1}) q_t(x_t | x_{1:t-1}),
  \]
  where
  \[
  q_t(x_t | x_{1:t-1}) := \frac{\pi_t(x_{1:t})}{\pi_{t-1}(x_{1:t-1})} \quad [= g(y_t | x_t) f(x_t | x_{t-1})]
  \]

- **Optimal resampling weights:**
  \[
  \bar{W}^i_{t-1} := Z_t(X^i_{1:t-1}) \quad [= p(y_t | X^i_{t-1})]
  \]
Fully adapted auxiliary SMC sampler

Let $\bar{\pi}_t(x_{1:t}) = \mathcal{Z}_t^{-1} \pi_t(x_{1:t})$ for $t = 1, 2, \ldots$ be a sequence of target distributions.

- **Optimal proposal:**
  $$\bar{q}_t(x_t | x_{1:t-1}) = Z_t^{-1}(x_{1:t-1}) q_t(x_t | x_{1:t-1}),$$
  where
  $$q_t(x_t | x_{1:t-1}) := \frac{\pi_t(x_{1:t})}{\pi_{t-1}(x_{1:t-1})} \quad \Rightarrow g(y_t | x_t) f(x_t | x_{t-1})$$

- **Optimal resampling weights:**
  $$\bar{W}_t^i := Z_t(X_{t-1}^i) \quad \Rightarrow p(y_t | X_{t-1}^i)$$

Results in an *unweighted* set of particles $\{X_{1:t}^i\}_{i=1}^N$, such that $\bar{\pi}_t^N(x_{1:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{X_{1:t}^i}(x_{1:t})$ approximates $\bar{\pi}_t$. 
Fully adapted and nested SMC

Given \( \{X^{i}_{1:t-1}\}_{i=1}^{N} \) targeting \( \bar{\pi}_{t-1}(x_{1:t-1}) \) and a class \( Q \) with functions \( \text{GetZ()} \) and \( \text{Simulate()} \) that generates *properly weighted* samples:

\[ \hat{Z}^{i}_{t} = q_{\text{GetZ}}. \]

\[ X^{i}_{t} = q_{\text{Simulate}}. \]

\[ \{X^{i}_{1:t-1}\}_{i=1}^{N} \text{ targeting } \bar{\pi}_{t-1}(x_{1:t-1}). \]
Fully adapted and nested SMC

Given \( \{X^i_{1:t-1}\}_{i=1}^N \) targeting \( \bar{\pi}_{t-1}(x_{1:t-1}) \) and a class \( Q \) with functions GetZ() and Simulate() that generates *properly weighted* samples:

<table>
<thead>
<tr>
<th>Fully adapted SMC</th>
<th>Nested SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialisation:</td>
<td>( q^i = Q(q_t(\cdot</td>
</tr>
</tbody>
</table>

□ CLT with std MC rate \( 1/\sqrt{N} \), asymptotic variance depends on \( M \).

\( \square \) Unbiased estimator of \( Z_t \):

\[ \hat{Z}_t = \hat{Z}_{t-1} \times \frac{1}{N} \sum_{i=1}^N \hat{Z}_i^t. \]
Fully adapted and nested SMC

Given \( \{X^i_{1:t-1}\}_{i=1}^N \) targeting \( \bar{\pi}_{t-1}(x_{1:t-1}) \) and a class \( Q \) with functions \( \text{GetZ()} \) and \( \text{Simulate()} \) that generates properly weighted samples:

\[
\begin{array}{ll}
\text{Fully adapted SMC} & \text{Nested SMC} \\
\hline
\text{Initialisation:} & - \\
\text{Resampling weights:} & q^i = Q(q_t(\cdot | X^i_{1:t-1}), M) \\
& \hat{Z}_t^i = q^i.\text{GetZ()} \\
\end{array}
\]

\( \square \) CLT with std MC rate \( 1/\sqrt{N} \), asymptotic variance depends on \( M \).

\( \square \) Unbiased estimator of \( Z_t \): \( \hat{Z}_t = \hat{Z}_t - 1 \times \frac{1}{N} \sum_{i=1}^N \hat{Z}_t^i \).
Fully adapted and nested SMC

Given \( \{X^i_{1:t-1}\}_{i=1}^N \) targeting \( \bar{\pi}_{t-1}(x_{1:t-1}) \) and a class \( Q \) with functions GetZ() and Simulate() that generates *properly weighted* samples:

<table>
<thead>
<tr>
<th>Fully adapted SMC</th>
<th>Nested SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>( q^i = Q(q_t(\cdot</td>
</tr>
<tr>
<td>\textbf{Initialisation:}</td>
<td>( Z_t(X^i_{1:t-1}) )</td>
</tr>
<tr>
<td>\textbf{Resampling weights:}</td>
<td>( X^i_t \sim q_t(x_t</td>
</tr>
<tr>
<td>\textbf{Propagation:}</td>
<td>-</td>
</tr>
</tbody>
</table>
Fully adapted and nested SMC

Given \( \{X_{1:t-1}^i\}_{i=1}^N \) targeting \( \bar{\pi}_{t-1}(x_{1:t-1}) \) and a class \( Q \) with functions GetZ() and Simulate() that generates *properly weighted* samples:

<table>
<thead>
<tr>
<th>Fully adapted SMC</th>
<th>Nested SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initialisation:</strong></td>
<td>—</td>
</tr>
<tr>
<td><strong>Resampling weights:</strong></td>
<td>( Z_t(X_{1:t-1}^i) )</td>
</tr>
<tr>
<td><strong>Propagation:</strong></td>
<td>( X_t^i \sim \bar{q}_t(x_t</td>
</tr>
</tbody>
</table>

\( \Rightarrow \{X_{1:t}^i\}_{i=1}^N \) targeting \( \bar{\pi}_t(x_{1:t}) \).
Fully adapted and nested SMC

Given \( \{X^i_{1:t-1}\}_{i=1}^N \) targeting \( \bar{\pi}_{t-1}(x_{1:t-1}) \) and a class Q with functions \( \text{GetZ}() \) and \( \text{Simulate}() \) that generates properly weighted samples:

<table>
<thead>
<tr>
<th>Fully adapted SMC</th>
<th>Nested SMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X^i_t \sim \tilde{q}_t(x_t</td>
<td>X^A^i_{1:t-1}) )</td>
</tr>
</tbody>
</table>

\[ q^i = Q(q_t(\cdot | X^i_{1:t-1}), M) \]

\( \hat{Z}^i_t = q^i \cdot \text{GetZ}() \)

\[ \Rightarrow \{X^i_{1:t}\}_{i=1}^N \text{ targeting } \bar{\pi}_t(x_{1:t}). \]

\( \square \) CLT with std MC rate \( 1/\sqrt{N} \), asymptotic variance depends on \( M \).

\( \square \) Unbiased estimator of \( Z_t \): \( \hat{Z}_t = \hat{Z}_{t-1} \times \frac{1}{N} \sum_{i=1} \hat{Z}^i_t. \)
2D Markov Random Field

1 spatial + 1 temporal dimension

\[
\bar{\pi}_t(x_{1:t}) = \frac{1}{Z_t} \varphi_1(x_1) \prod_{s=2}^{t} \{ \varphi_s(x_s) \psi_s(x_{s-1}, x_s) \}.
\]
2D MRF – Nested SMC implementation (I/III)

Optimal proposals given by:

\[ q_t(x_t | x_{t-1}) = \phi_t(x_t) \psi_t(x_{t-1}, x_t) \]
2D MRF – Nested SMC implementation (I/III)

Optimal proposals given by:

\[ q_t(x_t \mid x_{t-1}) = \phi_t(x_t)\psi_t(x_{t-1}, x_t) \]

\[ = \left\{ \prod_{k=1}^{d} G_{t,k}(x_{t,k}) \prod_{k=2}^{d} m(x_{t,k-1}, x_{t,k}) \right\} \left\{ \prod_{k=1}^{d} \psi(x_{t-1,k}, x_{t,k}) \right\} \]
2D MRF – Nested SMC implementation (II/III)

Proposed algorithm:

**Step 1: Resampling**
(Optimal weights $\tilde{W}_t^i = Z_t(X_{t-1}^i)$ with $Z_t(x_{t-1}) = \int q_t(x_{t} | x_{t-1}) dx_t$.)
2D MRF – Nested SMC implementation (II/III)

Proposed algorithm:

**Step 1: Resampling**

(Optimal weights \( \tilde{W}^i_{t-1} = Z_t(X^i_{t-1}) \) with 

\[ Z_t(x_{t-1}) = \int q_t(x_t | x_{t-1}) dx_t. \]

- For each particle \( \{X^i_{t-1}\}_{i=1}^N \):
  - Run PF with \( M \) particles for target \( q_t(x_t | X^i_{t-1}) \).
2D MRF – Nested SMC implementation (II/III)

Proposed algorithm:

**Step 1: Resampling**

(Optimal weights $\tilde{W}_{t-1}^i = Z_t(X_{t-1}^i)$ with $Z_t(x_{t-1}) = \int q_t(x_t | x_{t-1}) dx_t$.)

- For each particle $\{X_{t-1}^i\}_{i=1}^N$:
  - Run PF with $M$ particles for target $q_t(x_t | X_{t-1}^i)$.
  - Estimate normalising constant:
    $$\hat{Z}_t^i = \prod_{k=1}^d \left\{ \frac{1}{N} \sum_{j=1}^M W_k^{i,j} \right\}.$$
2D MRF – Nested SMC implementation (II/III)

Proposed algorithm:

**Step 1: Resampling**

(Optimal weights $\tilde{W}_{t-1}^i = Z_t(X_{t-1}^i)$ with $Z_t(x_{t-1}) = \int q_t(x_t | x_{t-1}) dx_t$.)

- For each particle $\{X_{t-1}^i\}_{i=1}^N$:
  - Run PF with $M$ particles for target $q_t(x_t | X_{t-1}^i)$.
  - Estimate normalising constant:
    $$\widehat{Z}_t^i = \prod_{k=1}^d \left\{ \frac{1}{N} \sum_{j=1}^M W_{k,j}^i \right\}.$$  
  - Resample $\{X_{t-1}^i\}_{i=1}^N$ and corresponding PFs based on $\{\widehat{Z}_t^i\}_{i=1}^N$. 

2D MRF – Nested SMC implementation (III/III)

Step 2: Propagation

- Assume particle $X_{t-1}^i$ resampled $n_t^i$ times.
- For $i = 1, \ldots, N$, generate $n_t^i$ descendants of $X_{t-1}^i$ by backward simulation.
2D MRF – Nested SMC implementation (III/III)

Step 2: Propagation

- Assume particle $X_{t-1}^i$ resampled $n_t^i$ times.
- For $i = 1, \ldots, N$, generate $n_t^i$ descendants of $X_{t-1}^i$ by backward simulation:
  - $P(X_{t,d}' = X_{t,d}^{i,j}) = W_{t,d}^{i,j}$ ($j = 1, \ldots, M$).
2D MRF – Nested SMC implementation (III/III)

Step 2: Propagation

• Assume particle $X_{t-1}^i$ resampled $n_t^i$ times.

• For $i = 1, \ldots, N$, generate $n_t^i$ descendants of $X_{t-1}^i$ by backward simulation:
  
  - $\mathbb{P}(X_{t,d} = X_{t,d}^{i,j}) = W_{t,d}^{i,j}$ \quad ($j = 1, \ldots, M$).
  
  - For $k = d - 1$ to 1,
    
    $$
    \mathbb{P}(X_{t,k} = X_{t,k}^{i,j}) = \frac{W_{t,k}^{i,j}m(X_{t,k}^{i,j}, X'_{t,k+1})}{\sum_{\ell=1}^{M} W_{t,k}^{i,\ell} m(X_{t,k}^{i,\ell}, X'_{t,k+1})} \quad (j = 1, \ldots, M).
    $$

⇒ $X_{t}^{'} \approx \bar{q}_t(\cdot | X_{t-1}^i)$. Results in $N$ unweighted particles: \{ $X_{t}^i$ \}$_{i=1}^{N}$. 
2D MRF – Nested SMC implementation (III/III)

Step 2: Propagation

- Assume particle $X^{i}_{t-1}$ resampled $n^{i}_{t}$ times.
- For $i = 1, \ldots, N$, generate $n^{i}_{t}$ descendants of $X^{i}_{t-1}$ by backward simulation:
  - $P(X^{i,j}_t = X^{i,j}_{t,d}) = W^{i,j}_t \quad (j = 1, \ldots, M)$.
  - For $k = d - 1$ to 1,

$$P(X^{i,j}_t = X^{i,j}_{t,k}) = \frac{W^{i,j}_t m(X^{i,j}_{t,k}, X^{i,j}_{t,k+1})}{\sum_{\ell=1}^{M} W^{i,\ell}_t m(X^{i,\ell}_{t,k}, X^{i,\ell}_{t,k+1})} \quad (j = 1, \ldots, M).$$

$\Rightarrow \quad X^{'}_t = X^{'}_{t,1:d} \approx \bar{q}_t(\cdot \mid X^{i}_{t-1}).$

Results in $N$ unweighted particles: $\{X^{i}_{t}\}_{i=1}^{N}$
ex) Gaussian spatio-temporal model

Gaussian spatio-temporal model in the form of a 2D MRF, $d \times t$, i.e. \( \dim x_t = d \).

\[
p(x_{1:t}, y_{1:t}) \propto \prod_{s=1}^{t} \begin{cases} 
\mathcal{N}(y_s; x_s, \tau^{-1}I) & \text{if } G \\
\mathcal{N}(x_s; ax_{s-1}, I) & \text{if } \psi \\
\mathcal{N}(x_s; 0, \Sigma) & \text{if } m
\end{cases}
\]

where \( \Sigma^{-1} \) is a banded matrix (reflecting local dependencies).
ex) Gaussian spatio-temporal model

\[ d = 50 \quad d = 100 \quad d = 200 \]

Figure: Median (over dimension) effective sample size (ESS) and 15–85% percentiles. \( N = 500 \) and \( M = 2d \). (Results for 100 independent runs.)

\[ \text{ESS}_{t,k} := \left( \mathbb{E} \left[ \frac{(\hat{x}_{t,k} - \mu_{t,k})^2}{\sigma_{t,k}^2} \right] \right)^{-1} \]

ex) Spatio-temporal model for drought prediction

- System state $x_t = \{x_{t,k,\ell}\}_{k=1,\ell=1}^{K,L}$, i.e., dimension is $d = K \times L$.
- Binary variables: $x_{t,k,\ell} = 0$ (normal state) or $x_{t,k,\ell} = 1$ (drought).
- Yearly Gaussian observations of precipitation at each site.
ex) Spatio-temporal model for drought prediction

Exploit the rectangular structure in three levels:

Level 1: Instantiate a Nested SMC sampler targeting the full posterior filtering distribution.

Level 2: To sample $x_t$, we run a Nested SMC sampler, operating on the “columns” $x_{t,1:K,\ell}$, $\ell = 1, \ldots, L$.

Level 3: To sample each column $x_{t,1:K,\ell}$ we run a third level of SMC, operating on the individual components $x_{t,k,\ell}$, $k = 1, \ldots, K$. 
ex) Spatio-temporal model for drought prediction

- Data from the Sahel region in Africa for years 1950–2000.
- \( \{K, L\} = \{24, 44\} \)  
  \( (\Rightarrow d = 1056). \)
- \( \{N, M_1, M_2\} = \{100, 40, 20\}. \)

Figure: Sahel region in 1989.
Wrapping up

Summary:

- NSMC allows us to \textit{“exactly approximate”} a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) \( M \) of particles in the “internal” filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from “tens” to “hundreds” (?).
Wrapping up

Summary:

• NSMC allows us to “exactly approximate” a fully adapted SMC sampler.
• Forward-backward strategy for lattice models.
• Provably correct for any number(s) $M$ of particles in the “internal” filter(s).
• Modular to an arbitrary degree.
• Pushes the dimension-limit for SMC from “tens” to “hundreds” (?).
Wrapping up

Summary:

- NSMC allows us to "exactly approximate" a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) $M$ of particles in the "internal" filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from “tens” to “hundreds” (?).
Wrapping up

Summary:

- NSMC allows us to "exactly approximate" a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) $M$ of particles in the "internal" filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from "tens" to "hundreds" (?).
Wrapping up

Summary:

• NSMC allows us to "exactly approximate" a fully adapted SMC sampler.

• Forward-backward strategy for lattice models.

• Provably correct for any number(s) $M$ of particles in the "internal" filter(s).

• Modular to an arbitrary degree.

• Pushes the dimension-limit for SMC from "tens" to "hundreds" (?)
Wrapping up

Summary:

- NSMC allows us to "exactly approximate" a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) $M$ of particles in the "internal" filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from “tens” to “hundreds” (?)..

Worth to note:

- Can straightforwardly be used with, e.g., Particle MCMC for learning.
- Computational complexity of the method is $N \times M$ (for two layers). However, much more efficient than a bootstrap PF with $N \times M$ particles!
- NSMC does not beat the curse of dimensionality!
- Can we push the limit even further with blocking and localisation strategies?
Wrapping up

Summary:

- NSMC allows us to \textit{“exactly approximate”} a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) $M$ of particles in the “internal” filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from “tens” to “hundreds” (?)..

Worth to note:

- Can straightforwardly be used with, e.g., Particle MCMC for learning.
- Computational complexity of the method is $N \times M$ (for two layers). However, \textit{much more efficient} than a bootstrap PF with $N \times M$ particles!
- NSMC \textit{does not} beat the curse of dimensionality!
- Can we push the limit even further with blocking and localisation strategies?
Wrapping up

Summary:

- NSMC allows us to “exactly approximate” a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) \( M \) of particles in the “internal” filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from “tens” to “hundreds” (?)..

Worth to note:

- Can straightforwardly be used with, e.g., Particle MCMC for learning.
- Computational complexity of the method is \( N \times M \) (for two layers). However, \textit{much more efficient} than a bootstrap PF with \( N \times M \) particles!
- NSMC \textit{does not} beat the curse of dimensionality!
- Can we push the limit even further with blocking and localisation strategies?
Wrapping up

Summary:

- NSMC allows us to "exactly approximate" a fully adapted SMC sampler.
- Forward-backward strategy for lattice models.
- Provably correct for any number(s) $M$ of particles in the "internal" filter(s).
- Modular to an arbitrary degree.
- Pushes the dimension-limit for SMC from “tens” to “hundreds” (?) .

Worth to note:

- Can straightforwardly be used with, e.g., Particle MCMC for learning.
- Computational complexity of the method is $N \times M$ (for two layers). However, much more efficient than a bootstrap PF with $N \times M$ particles!
- NSMC does not beat the curse of dimensionality!
- Can we push the limit even further with blocking and localisation strategies?
References

C. A. Naesseth, F. Lindsten, and T. B. Schön.
Nested sequential Monte Carlo methods.

A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou.
A stable particle filter in high-dimensions.

P. Fearnhead, O. Papaspiliopoulos, G. O. Roberts, and A. Stuart.
Random-weight particle filtering of continuous time processes.

N. Chopin, P. E. Jacob, and O. Papaspiliopoulos.
SMC$^2$: an efficient algorithm for sequential analysis of state-space models.

A. Johansen, N. Whiteley, and A. Doucet.
Exact approximation of Rao-Blackwellised particle filters.
In Proceedings of the 16th IFAC Symposium on System Identification (SYSID), Brussels, Belgium, July 2012.

C. Vergé, C. Dubarry, P. Del Moral, and E. Moulines.
On parallel implementation of sequential Monte Carlo methods: the island particle model.
Thank you!

www.liu.se