Particle filters and curse of dimensionality

Patrick Rebeschini
Joint work with Ramon van Handel
Princeton University

Monte Carlo Inference for Complex Statistical Models
University of Cambridge - April 24, 2014
Nonlinear filtering

Hidden Markov model:

\[ Y_{n-1} \xrightarrow{g} X_{n-1} \xrightarrow{p} X_n \xrightarrow{p} X_{n+1} \]

- Transition density \( P[X_{n+1} \in dx \mid X_n] = p(X_n, x) \varphi(dx) \).
- Observation density \( P[Y_n \in dy \mid X_n] = g(X_n, y) \psi(dy) \).

Goal of nonlinear filtering is to compute \( \pi_n := P[X_n \in \cdot \mid Y_1, \ldots, Y_n] \).
Nonlinear filtering

Hidden Markov model:

\[
\begin{align*}
Y_{n-1} & \xrightarrow{g} X_{n-1} \\
Y_n & \xrightarrow{g} X_n \\
Y_{n+1} & \xrightarrow{g} X_{n+1}
\end{align*}
\]

- **Transition density** \( P[X_{n+1} \in dx | X_n] = p(X_n, x) \varphi(dx) \).
- **Observation density** \( P[Y_n \in dy | X_n] = g(X_n, y) \psi(dy) \).

Goal of nonlinear filtering is to compute \( \pi_n := P[X_n \in \cdot | Y_1, \ldots, Y_n] \).

**Classical applications** (tracking, navigation, vision, communications, finance, \ldots)
- Principled implementation of filtering theory by particle filtering algorithms.

**Data assimilation** (weather forecasting, oceanography, geophysics, \ldots)
- Filtering done almost exclusively using heuristics (3DVAR, EnKF, etc.)
Nonlinear filtering

Hidden Markov model:

\[
\begin{align*}
Y_{n-1} & \xrightarrow{g} X_{n-1} \\
Y_n & \xrightarrow{g} X_n \\
Y_{n+1} & \xrightarrow{g} X_{n+1}
\end{align*}
\]

- Transition density \( P[X_{n+1} \in dx | X_n] = p(X_n, x) \varphi(dx) \).
- Observation density \( P[Y_n \in dy | X_n] = g(X_n, y) \psi(dy) \).

Goal of nonlinear filtering is to compute \( \pi_n := P[X_n \in \cdot | Y_1, \ldots, Y_n] \).

Classical applications (tracking, navigation, vision, communications, finance, \ldots)
- Principled implementation of filtering theory by particle filtering algorithms.

Data assimilation (weather forecasting, oceanography, geophysics, \ldots)
- Filtering done almost exclusively using heuristics (3DVAR, EnKF, etc.)

Q. Fundamental obstacle to filtering in high dimension? Any hope to go beyond?
By the Bayes formula:

\[
\begin{align*}
\pi_{n-1} & \xrightarrow{\text{predict}} P\pi_{n-1} & \xrightarrow{\text{correct}} \pi_n = C_n P\pi_{n-1}
\end{align*}
\]

where

\[
P \mu(dx) = \varphi(dx) \int \mu(dz) p(z, x), \quad C_n \mu(dx) = \frac{\mu(dx) g(x, Y_n)}{\int \mu(dz) g(z, Y_n)}.
\]
SIR particle filter

By the Bayes formula:

\[
\pi_{n-1} \xrightarrow{\text{predict}} P\pi_{n-1} \xrightarrow{\text{correct}} \pi_n = C_n P\pi_{n-1}
\]

where

\[
P\mu(dx) = \varphi(dx) \int \mu(dz) p(z, x), \quad C_n\mu(dx) = \frac{\mu(dx) g(x, Y_n)}{\int \mu(dz) g(z, Y_n)}.
\]

SIR algorithm (Gordon et al. 1993): approximate \(\pi_n\) by empirical measure \(\hat{\pi}_n\)

\[
\hat{\pi}_{n-1} \xrightarrow{\text{predict}} P\hat{\pi}_{n-1} \xrightarrow{\text{sample}} S^N P\hat{\pi}_{n-1} \xrightarrow{\text{correct}} \hat{\pi}_n = C_n S^N P\hat{\pi}_{n-1}
\]

where

\[
S^N\mu(dx) = \frac{1}{N} \sum_{i=1}^{N} \delta_{X(i)}(dx), \quad X(1), \ldots, X(N) \text{ i.i.d. samples} \sim \mu.
\]
A typical iteration of the SIR particle filter looks like this ($N=6$):

\[ \hat{\pi}_n = \sum_{i=1}^{N} W_n(i) \delta X(i), \quad W_n(i) = \frac{g(X(i), Y_n)}{\sum_{i=1}^{N} g(X(i), Y_n)}, \quad x(1), \ldots, x(N) \sim P\hat{\pi}_{n-1}. \]
Good news and bad news

Not hard to get an easy error bound:

\[ \| \pi_n - \hat{\pi}_n \| := \sup_{\| f \|_\infty \leq 1} \left( E |\pi_n(f) - \hat{\pi}_n(f)|^2 \right)^{1/2} \leq \frac{C}{\sqrt{N}}. \]

But key to performance is the constant \( C \).

**Good surprise:** typically \( C \) does not depend on time \( n \).

- Explains why particle filters work so well in **classical applications**.
- First results in *Del Moral and Guionnet 2001*.

**Bad surprise:** typically \( C \) is exponential in (effective) model dimension \( d \).

- Particle filters have proved to be largely useless for **data assimilation**.
- First results in *Bickel, Li, Bengtsson 2008*.

No principled way forward has been proposed...
Stability

Why does error not accumulate over time?
Stability

Why does error not accumulate over time?

Recall $\pi_n = F_n \pi_{n-1}$ and $\hat{\pi}_n = \hat{F}_n \hat{\pi}_{n-1}$, where $F_n := C_n P$ and $\hat{F}_n := C_n S^N P$.

$$\| F_n \cdots F_1 \pi_0 - \hat{F}_n \cdots \hat{F}_1 \pi_0 \| \leq \sum_{r=1}^{n} \| F_{n:r+1} F_r \hat{\pi}_{r-1} - F_{n:r+1} \hat{F}_r \hat{\pi}_{r-1} \|$$

contribution of error in step $r$
Stability

Why does error not accumulate over time?

Recall \( \pi_n = F_n \pi_{n-1} \) and \( \hat{\pi}_n = \hat{F}_n \hat{\pi}_{n-1} \), where \( F_n := C_n P \) and \( \hat{F}_n := C_n S^N P \).

\[
\| F_n \cdots F_1 \pi_0 - \hat{F}_n \cdots \hat{F}_1 \pi_0 \| \leq \sum_{r=1}^{n} \left( \| F_{n:r+1} F_r \hat{\pi}_{r-1} - F_{n:r+1} \hat{F}_r \hat{\pi}_{r-1} \| \right)
\]

contribution of error in step \( r \)

Proposition (Del Moral, Guionnet 2001)

Suppose that \( \varepsilon \leq p(z, x) \leq \varepsilon^{-1} \). Then the filter is stable:

\[
\| F_{n:r+1} \mu - F_{n:r+1} \nu \| \leq \varepsilon^{-2} (1 - \varepsilon^2)^{n-r} \| \mu - \nu \|.
\]
Stability

Why does error not accumulate over time?

Recall \( \pi_n = F_n \pi_{n-1} \) and \( \hat{\pi}_n = \hat{F}_n \hat{\pi}_{n-1} \), where \( F_n := C_n P \) and \( \hat{F}_n := C_n S^N P \).

\[
\left\| \underbrace{F_n \cdots F_1 \pi_0}_{\pi_n} - \underbrace{\hat{F}_n \cdots \hat{F}_1 \pi_0}_{\hat{\pi}_n} \right\| \leq \sum_{r=1}^{n} \left\| \underbrace{F_{n:r+1} F_r \hat{\pi}_{r-1}}_{\text{contribution of error in step } r} - \underbrace{F_{n:r+1} \hat{F}_r \hat{\pi}_{r-1}} \right\| \leq \frac{C}{\sqrt{N}}.
\]

Proposition (Del Moral, Guionnet 2001)

Suppose that \( \varepsilon \leq p(z, x) \leq \varepsilon^{-1} \). Then the filter is stable:

\[
\left\| F_{n:r+1} \mu - F_{n:r+1} \nu \right\| \leq \varepsilon^{-2} (1 - \varepsilon^2)^{n-r} \| \mu - \nu \|.
\]
Stability

Why does error not accumulate over time?

Recall $\pi_n = F_n \pi_{n-1}$ and $\hat{\pi}_n = \hat{F}_n \hat{\pi}_{n-1}$, where $F_n := C_n P$ and $\hat{F}_n := C_n S^N P$.

$$\| F_n \cdots F_1 \pi_0 - \hat{F}_n \cdots \hat{F}_1 \pi_0 \| \leq \sum_{r=1}^{n} \| F_{n:r+1} F_r \hat{\pi}_{r-1} - F_{n:r+1} \hat{F}_r \hat{\pi}_{r-1} \| \leq \frac{C}{\sqrt{N}}.$$

Proposition (Del Moral, Guionnet 2001)

Suppose that $\varepsilon \leq p(z, x) \leq \varepsilon^{-1}$. Then the filter is stable:

$$\| F_{n:r+1} \mu - F_{n:r+1} \nu \| \leq \varepsilon^{-2} (1 - \varepsilon^2)^{n-r} \| \mu - \nu \|.$$

Key idea

Filter stability is a dissipation mechanism for approximation errors in time.

$$P(X_n \in \cdot | X_0, Y_1, \ldots, Y_n) \approx P(X_n \in \cdot | Y_1, \ldots, Y_n) \quad \text{for } n \text{ large.}$$
A trivial high-dimensional model

What do we mean by dimension?

\[ Y_{n-1} \rightarrow X_{n-1} \rightarrow X_n \rightarrow X_{n+1} \rightarrow Y_{n+1} \]

\[ \text{time} \]
A trivial high-dimensional model

What do we mean by dimension?
Curse of dimensionality

Crude one-step error bound:

\[ \|F_n \mu - \hat{F}_n \mu\| = \left\| \frac{g(x, Y_n) (P \mu)(dx)}{\int g(x, Y_n) (P \mu)(dx)} - \frac{g(x, Y_n) (S^N P \mu)(dx)}{\int g(x, Y_n) (S^N P \mu)(dx)} \right\| \leq \frac{\sup g}{\inf g} \frac{2}{\sqrt{N}}, \]

where we used

\[ \|\mu - S^N \mu\| \leq \frac{1}{\sqrt{N}}. \]
Curse of dimensionality

Crude one-step error bound:

\[ \| F_n \mu - \hat{F}_n \mu \| = \left\| \frac{g(x, Y_n)(P\mu)(dx)}{\int g(x, Y_n)(P\mu)(dx)} - \frac{g(x, Y_n)(S^N P\mu)(dx)}{\int g(x, Y_n)(S^N P\mu)(dx)} \right\| \leq \frac{\sup g}{\inf g} \frac{2}{\sqrt{N}}, \]

where we used

\[ \| \mu - S^N \mu \| \leq \frac{1}{\sqrt{N}}. \]

In trivial model,

\[ g(x, Y_n) = \prod_{i=1}^{d} g_0(x^i, Y^i_n) \implies \frac{\sup g}{\inf g} = \left( \frac{\sup g_0}{\inf g_0} \right)^d = e^{d \text{osc}[\log g_0]}. \]
Curse of dimensionality

Crude one-step error bound:

\[ \| F_n \mu - \hat{F}_n \mu \| = \left\| \frac{\int g(x, Y_n) (P \mu)(dx)}{\int g(x, Y_n) (P \mu)(dx)} - \frac{\int g(x, Y_n) (S^N P \mu)(dx)}{\int g(x, Y_n) (S^N P \mu)(dx)} \right\| \leq \frac{\sup g}{\inf g} \frac{2}{\sqrt{N}}, \]

where we used

\[ \| \mu - S^N \mu \| \leq \frac{1}{\sqrt{N}}. \]

In trivial model,

\[ g(x, Y_n) = \prod_{i=1}^{d} g_0(x^i, Y_n^i) \implies \frac{\sup g}{\inf g} = \left( \frac{\sup g_0}{\inf g_0} \right)^d = e^{d \text{osc}[\log g_0]}. \]

Theorem (Bickel, Li, Bengtsson 2008)

Bad things happen unless \( \log N \gtrsim \text{var} [\log g] \sim d \text{var} [\log g_0]. \)
Curse of dimensionality: example

Symmetric random walk in $\mathbb{Z}^1$ ($N = 4$):

\[ \hat{\pi}_0 = \delta_{\{\vec{0}\}} \]

\[ S^N P \hat{\pi}_0 \]

\[ \hat{\pi}_1 = C_1 S^N P \hat{\pi}_0 \]

Symmetric random walk in $\mathbb{Z}^2$ ($N = 4$):

\[ \hat{\pi}_0 = \delta_{\{\vec{0}\}} \]

\[ S^N P \hat{\pi}_0 \]

\[ \hat{\pi}_1 = C_1 S^N P \hat{\pi}_0 \]

In $d$ dimension, only $N/2^d$ particles will have relevant weights.
Fundamental problem: as $d \to \infty$, prob. measures become mutually singular.

Weight degeneracy: *importance sampling* can not fundamentally address the issue due to the *recursive* nature of the filtering problem (see also Snyder 2011).

Q. Is there any hope?
A more realistic high-dimensional model

\[ Y_{i+1}^{n-1} \rightarrow X_{i+1}^{n-1} \rightarrow \ldots \rightarrow X_{i-1}^{n-1} \rightarrow X_i^{n-1} \rightarrow X_i^n \rightarrow \ldots \rightarrow X_i^n \rightarrow X_i^{n+1} \rightarrow \ldots \rightarrow X_i^{n+1} \rightarrow Y_i^{n+1} \rightarrow Y_i^n \rightarrow \ldots \rightarrow Y_i^n \rightarrow Y_i^{n+1} \rightarrow \ldots \rightarrow Y_i^{n+1} \rightarrow Y_{i+1}^{n+1} \]

\[ \text{dimension} \rightarrow \ldots \rightarrow \text{time} \rightarrow \ldots \]
A more realistic high-dimensional model

\[ Y_{i+1} \rightarrow X_{i+1} \rightarrow Y_i \rightarrow X_i \rightarrow Y_{i-1} \rightarrow X_{i-1} \ldots \]

\[ \vdots \]

\[ \Rightarrow \]

dimension

time
Local hidden Markov model:

- Spatial index set is a graph $G = (V, E)$.
- $X_n = (X^n_v)_{v \in V}$, $Y_n = (Y^n_v)_{v \in V}$.
- Transition and observations densities are

$$p(z, x) = \prod_{v \in V} p^v(z, x^v),$$
$$g(z, y) = \prod_{v \in V} g^v(x^v, y^v),$$

where $p^v(z, x^v)$ depends only on $\{z^w : w \text{ is in neighborhood of } v\}$. 

$G = (V, E)$
Decay of correlations

Unlike in trivial model, \((X_v^n, Y_v^n)_{v \in V}\) are **not** independent!
Decay of correlations

Unlike in trivial model, \((X_n^v, Y_n^v)_{v \in V}\) are not independent!

What property shall we look for?
Decay of correlations

Unlike in trivial model, \((X_n^v, Y_n^v)_{v \in V}\) are not independent!

What property shall we look for?

- In many cases such models still exhibit decay of correlations:

  \((X_n^v, Y_n^v)\) and \((X_n^w, Y_n^w)\) are “nearly” independent when \(\text{dist}(v, w)\) is large.
Decay of correlations

Unlike in trivial model, \((X_v^n, Y_v^n)_{v \in V}\) are not independent!

What property shall we look for?

- In many cases such models still exhibit decay of correlations:

\[(X_v^n, Y_v^n) \text{ and } (X_w^n, Y_w^n) \text{ are “nearly” independent when } \text{dist}(v, w) \text{ is large.}\]

Key idea

In particular, for \(b\) large:

\[
P(X_v^n \in \cdot | Y_1, \ldots, Y_n) \approx P(X_v^n \in \cdot | Y_1^w, \ldots, Y_n^w, \text{dist}(v, w) \leq b).
\]
Resampling is the key to exploit filter stability algorithmically.
Resampling is the key to exploit filter stability algorithmically.

**Q.** What can we do to exploit the decay of correlations property?
Block particle filter

Resampling is the key to exploit filter stability algorithmically.

Q. What can we do to exploit the decay of correlations property?

A. LOCALIZE!

\[ \hat{\pi}_n \leftarrow \text{predict} \quad \rightarrow \quad P_\hat{\pi}_n \leftarrow \text{sample} \quad \rightarrow \quad \text{block} \quad \rightarrow \quad BS \quad \rightarrow \quad \text{correct} \quad \rightarrow \quad \hat{\pi}_n = C_n BS \quad \text{where} \quad B_\mu(dx) = \bigotimes_{K \in K} \mu(dx_K) \]
Resampling is the key to exploit filter stability algorithmically.

**Q.** What can we do to exploit the decay of correlations property?

**A.** LOCALIZE!

Partition $V$ into blocks $\mathcal{K}$.

\[
\begin{align*}
\hat{\pi}_{n-1} & \xrightarrow{\text{predict}} P\hat{\pi}_{n-1} \\
& \xrightarrow{\text{sample}} S^N P\hat{\pi}_{n-1} \\
& \xrightarrow{\text{block}} B S^N P\hat{\pi}_{n-1} \\
& \xrightarrow{\text{correct}} \hat{\pi}_n = C_n B S^N P\hat{\pi}_{n-1}
\end{align*}
\]

where $B \mu(dx) = \bigotimes_{K \in \mathcal{K}} \mu(dx^K)$. 

\[
\mu(dx) = \bigotimes_{K \in \mathcal{K}} \mu(dx^K).
\]
SIR particle filter (recall)

A typical iteration of the SIR particle filter looks like this (N=6):

\[
\hat{\pi}_n = \sum_{i=1}^{N} W_n(i) \delta X(i),
\]

\[
W_n(i) = \frac{\prod_{v \in V} g^v(X^v(i), Y^v_n)}{\sum_{i=1}^{N} \prod_{v \in V} g^v(X^v(i), Y^v_n)},
\]
A typical iteration of the block particle filter looks like this (\textbf{still }\text{N=6)}:

\[ \hat{\pi}_n = \bigotimes_{K \in \mathcal{K}} \sum_{i=1}^{N} W^K_n(i) \delta^{X^K(i)}, \quad W^K_n(i) = \frac{\prod_{v \in K} g^v(X^v(i), Y^v_n)}{\sum_{i=1}^{N} \prod_{v \in K} g^v(X^v(i), Y^v_n)}, \]
Main result (Rebeschini, van Handel, 2013)

Suppose that $\varepsilon \leq p^v(z, x^v) \leq \varepsilon^{-1}$ for all $v$. Then, for any $J \subseteq K \in \mathcal{K}$,

$$\|\pi_n - \hat{\pi}_n\|_J \leq c_1 e^{-\gamma_1 \text{dist}(J, K^c)} + c_2 \frac{e^{\gamma_2 \text{card}(K)}}{\sqrt{N}},$$

provided $\varepsilon \geq \varepsilon_0$ (where $\varepsilon_0, c_1, c_2, \gamma_1, \gamma_2$ do not depend on $\text{card}(V)$ or $n$).

Optimizing block size yields consistent algorithm uniformly in time and dimension.

Key idea:

Decay of correlations is a dissipation mechanism for approximation errors in space.
Table 1. Tracker failures observed in the 10,400 frame test sequence

<table>
<thead>
<tr>
<th>Tracker</th>
<th>Number of Samples</th>
<th>Number of Failures</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMC</td>
<td>50</td>
<td>123</td>
</tr>
<tr>
<td>MCMC</td>
<td>100</td>
<td>49</td>
</tr>
<tr>
<td>MCMC</td>
<td>200</td>
<td>28</td>
</tr>
<tr>
<td>MCMC</td>
<td>1000</td>
<td>16</td>
</tr>
<tr>
<td>joint particle filter</td>
<td>50</td>
<td>544</td>
</tr>
<tr>
<td>joint particle filter</td>
<td>100</td>
<td>519</td>
</tr>
<tr>
<td>joint particle filter</td>
<td>200</td>
<td>479</td>
</tr>
<tr>
<td>joint particle filter</td>
<td>1000</td>
<td>392</td>
</tr>
</tbody>
</table>
Fig. 1. An example of partitioning a traffic network into subnetworks for parallelized simulation/particle filtering.

Fig. 4. The filter root mean square error as a function of the number of particles for the centralized filter (top), approach 1 (middle), and approach 2 (bottom). The dots connected by the solid line indicate the mean, and...
Factored particles (Ng, Peshkin, Pfeffer 2002)

But not the same algorithm! Fundamental difference.

Factored Particles for Scalable Monitoring

Brenda Ng, Leonid Peshkin, Avi Pfeffer*
{bmnng, pesha, avi}@eecs.harvard.edu
Division of Engineering and Applied Sciences
Harvard University, Cambridge MA 02138

Abstract

Particle filtering (PF) has the attractive property that the approximation converges to the exact solution in the limit of an infinite number of samples. It also has an “any-real-time” property that the number of samples can be adjusted to fill however much time is available for inference at each time point. However, the variance of the method can be very high, particularly in high dimensional spaces, and too many particles might be needed for accurate approximation.

But not the same algorithm! Fundamental difference.
Main message: there is no fundamental obstacle to filtering in high dimension in systems that exhibit (conditional) decay of correlations.

First algorithm and mathematical analysis, decidedly proof of concept.

Guiding principle for algorithm design (blocking is one of many possible spatial regularizations to exploit decay of correlations; bias-variance trade-off).

Preprints:

