Nonlinear Filtering using Particles and Quadrature

Andreas Klöckner

May 8, 2007

Outline

Outline

The Nonlinear Filtering Problem

Have:

The Nonlinear Filtering Problem

Have:

- A hidden *Markov process* $\mathbf{x}_t$ for $t = 0, 1, 2, \ldots$ with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$.

The Nonlinear Filtering Problem

Have:

- A hidden *Markov process* $\mathbf{x}_t$ for $t = 0, 1, 2, \ldots$ with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$.
- Conditionally independent *observations* $\mathbf{y}_t$ for $t = 1, 2, 3, \ldots$ characterized by $p(\mathbf{y}_t|\mathbf{x}_t)$.

The Nonlinear Filtering Problem

Have:

- A hidden *Markov process* $\mathbf{x}_t$ for $t = 0, 1, 2, \ldots$ with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$.
- Conditionally independent *observations* $\mathbf{y}_t$ for $t = 1, 2, 3, \ldots$ characterized by $p(\mathbf{y}_t|\mathbf{x}_t)$.

Want:

The Nonlinear Filtering Problem

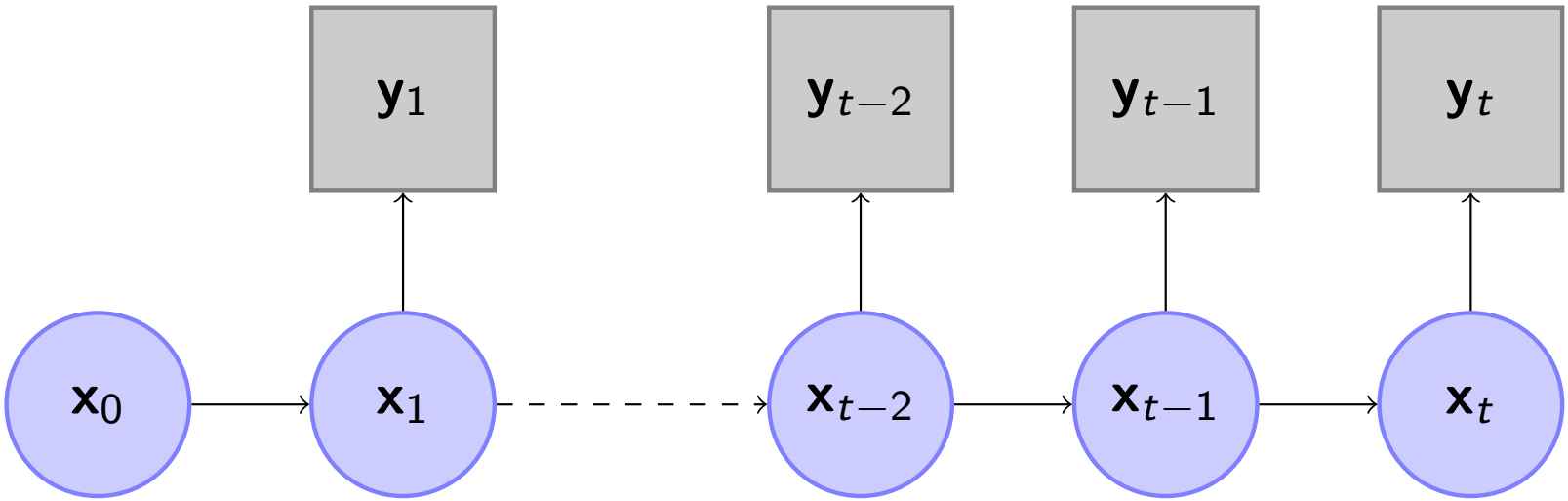
Have:

- A hidden *Markov process* $\mathbf{x}_t$ for $t = 0, 1, 2, \ldots$ with initial distribution $p(\mathbf{x}_0)$ and transition probability $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$.
- Conditionally independent *observations* $\mathbf{y}_t$ for $t = 1, 2, 3, \ldots$ characterized by $p(\mathbf{y}_t|\mathbf{x}_t)$.

Want:

- (An estimate of) The *posterior distribution* $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

Process and Observation, schematically



Bayes' Formula

Recall:

Bayes' Formula

Recall:

$p(X|Y)$

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)}$$

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int P(Y \cap X)\mathrm{d}X}$$

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int P(Y \cap X)\mathrm{d}X} = \frac{p(Y|X)p(X)}{\int p(Y|X)p(X)\mathrm{d}X}$$

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int P(Y \cap X)\mathrm{d}X} = \frac{p(Y|X)p(X)}{\int p(Y|X)p(X)\mathrm{d}X}$$

So in our setting:

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int P(Y \cap X)\mathrm{d}X} = \frac{p(Y|X)p(X)}{\int p(Y|X)p(X)\mathrm{d}X}$$

So in our setting:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})\mathrm{d}\mathbf{x}_{0:t}}$$

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int P(Y \cap X)\mathrm{d}X} = \frac{p(Y|X)p(X)}{\int p(Y|X)p(X)\mathrm{d}X}$$

So in our setting:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})\mathrm{d}\mathbf{x}_{0:t}}$$

Not usable for on-line processing.

Bayes' Formula

Recall:

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int P(Y \cap X)\mathrm{d}X} = \frac{p(Y|X)p(X)}{\int p(Y|X)p(X)\mathrm{d}X}$$

So in our setting:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})\mathrm{d}\mathbf{x}_{0:t}}$$

Not usable for on-line processing. Need a recursive algorithm.

Joint Posterior Update

$$\frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}$$

Joint Posterior Update

$$= \frac{\frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}}{\frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})}} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})}$$

Joint Posterior Update

$$
\begin{aligned}
& \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \\
= \quad & \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})} \\
\underline{\text{CI}} \quad & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t-1})}
\end{aligned}
$$

Joint Posterior Update

$$
\begin{aligned}
& \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \\
= \quad & \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})} \\
\underline{\underline{\text{CI}}} \quad & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t-1})} \\
\underline{\underline{\text{Markov}}} \quad & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t})}
\end{aligned}
$$

Joint Posterior Update

$$
\begin{aligned}
& \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \\
= \quad & \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})} \\
\underline{\text{CI}} \quad & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t-1})} \\
\underline{\text{Markov}} \quad & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t})} \\
= \quad & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} \cdot p(\mathbf{y}_{1:t-1})
\end{aligned}
$$

Joint Posterior Update

$$
\begin{aligned}
& \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})} \\
= & \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{1:t-1}|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1})} \\
\underline{\text{CI}} & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t-1})} \\
\underline{\text{Markov}} & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} \cdot \frac{p(\mathbf{y}_{1:t-1})}{p(\mathbf{x}_{0:t})} \\
= & \frac{p(\mathbf{y}_t|\mathbf{x}_{0:t})p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_{1:t})} \cdot p(\mathbf{y}_{1:t-1}) \\
= & \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}
\end{aligned}
$$

Joint Posterior Update

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$$

Joint Posterior Update

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$$

We can expand the denominator as

Joint Posterior Update

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$$

We can expand the denominator as

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

Joint Posterior Update

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$$

We can expand the denominator as

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

Two drawbacks:

Joint Posterior Update

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$$

We can expand the denominator as

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

Two drawbacks:

- Denominator not explicitly known

Joint Posterior Update

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$$

We can expand the denominator as

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

Two drawbacks:

- Denominator not explicitly known
- Don't care about the *joint* posterior–only want $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

Recursive Posterior Update

$$p(\mathbf{x}_t | \mathbf{y}_{1:t})$$

Recursive Posterior Update

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t-1}$$

Recursive Posterior Update

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t-1}$$

$$\overset{\text{JointUp}}{=} \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1}$$

Recursive Posterior Update

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t-1}$$

$$\overset{\text{JointUp}}{=} \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1}$$

$$= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1}$$

Recursive Posterior Update

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t-1}$$

$$\overset{\text{JointUp}}{=} \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1}$$

$$= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1}$$

$$\overset{\text{Markov}}{=} \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1}$$

Recursive Posterior Update

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t-1} \\
&\overset{\text{JointUp}}{=} \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1} \\
&= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1} \\
&\overset{\text{Markov}}{=} \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1} \\
&= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_t|\mathbf{y}_{1:t-1})
\end{aligned}
$$

Recursive Posterior Update

$$
\begin{aligned}
p(\mathbf{x}_t|\mathbf{y}_{1:t}) &= \int p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_{0:t-1} \\
&\overset{\text{JointUp}}{=} \int \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1} \\
&= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1} \\
&\overset{\text{Markov}}{=} \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \int p(\mathbf{x}_t|\mathbf{x}_{0:t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{0:t-1} \\
&= \frac{p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \\
&= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}
\end{aligned}
$$

The Complete Recursion

Prediction $t \longrightarrow$ *Posterior* t:

The Complete Recursion

Prediction $t \rightarrow$ *Posterior* t:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}$$

The Complete Recursion

Prediction $t \rightarrow$ *Posterior* t:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}$$

Posterior $t-1 \rightarrow$ *Prediction* t:

The Complete Recursion

Prediction $t \rightarrow$ *Posterior* t:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}$$

Posterior $t-1 \rightarrow$ *Prediction* t:

Start with

The Complete Recursion

Prediction $t \rightarrow$ *Posterior* t:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}$$

Posterior $t-1 \rightarrow$ *Prediction* t:
Start with

$$p(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1})\mathrm{d}\mathbf{x}_{t-1},$$

The Complete Recursion

Prediction $t \rightarrow$ *Posterior* t:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}$$

Posterior $t-1 \rightarrow$ *Prediction* t:
Start with

$$p(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1})\mathrm{d}\mathbf{x}_{t-1},$$

and obtain

The Complete Recursion

Prediction $t \rightarrow$ *Posterior* t:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t}$$

Posterior $t-1 \rightarrow$ *Prediction* t:
Start with

$$p(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1})\mathrm{d}\mathbf{x}_{t-1},$$

and obtain

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_{t-1}.$$

Outline

Our Example Process

Throughout this presentation, we will stick to the following model:

Our Example Process

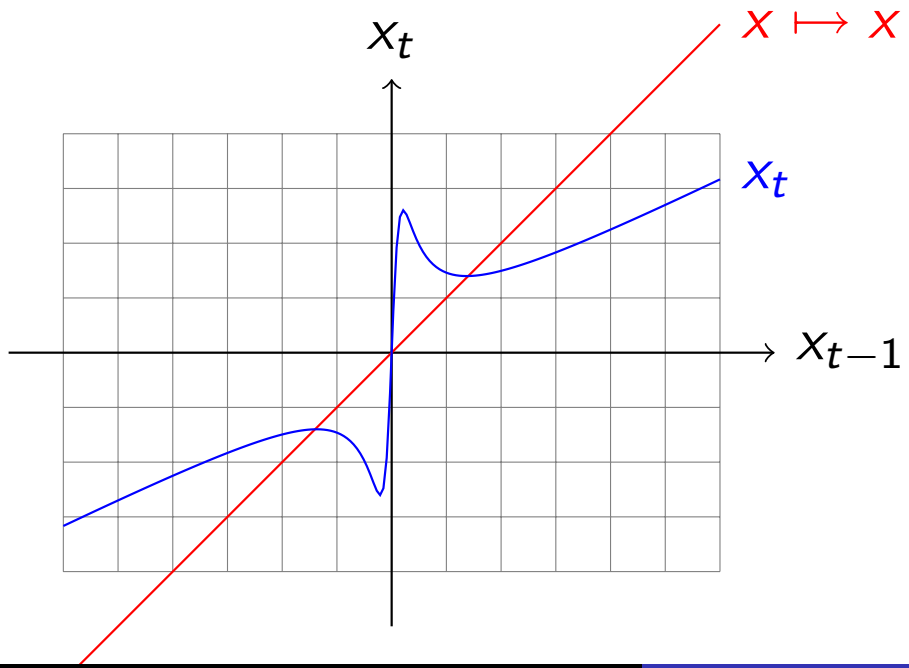
Throughout this presentation, we will stick to the following model:

$$x_t = \frac{1}{2} x_{t-1} + 25 \frac{x_{t-1}}{1 + x_{t-1}^2}$$

Our Example Process

Throughout this presentation, we will stick to the following model:

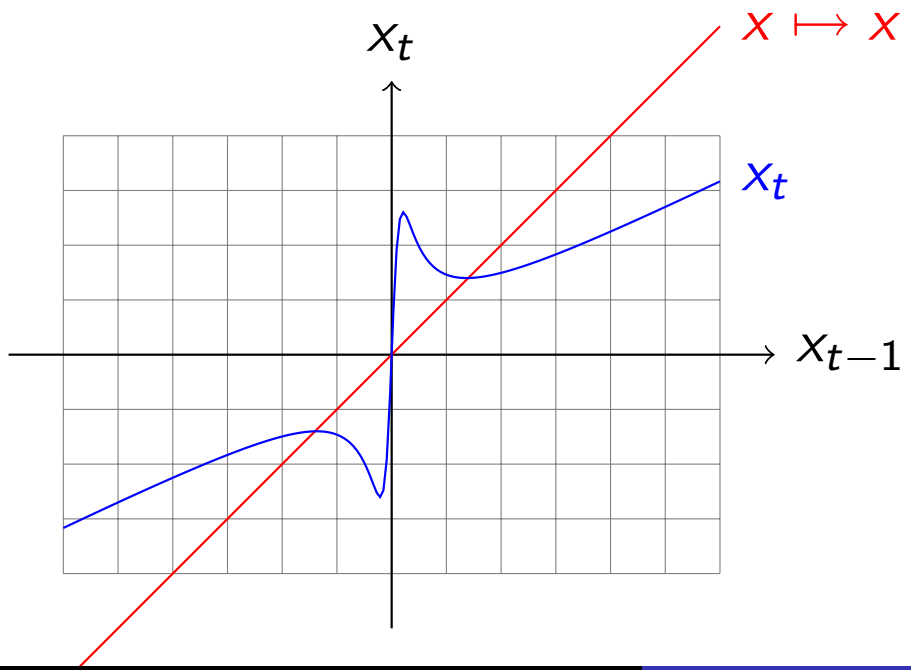
$$x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2}$$



Our Example Process

Throughout this presentation, we will stick to the following model:

$$x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8cos(1.2t) + v_t,$$



Our Example Process

Throughout this presentation, we will stick to the following model:

$$\begin{aligned} x_t &= \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8cos(1.2t) + v_t, \\ y_t &= \frac{x_t^2}{20} + w_t, \end{aligned}$$

Our Example Process

Throughout this presentation, we will stick to the following model:

$$\begin{aligned} x_t &= \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8cos(1.2t) + v_t, \\ y_t &= \frac{x_t^2}{20} + w_t, \end{aligned}$$

where $x_0 \sim \mathcal{N}(0, \sigma_1^2)$, $v_t \sim \mathcal{N}(0, \sigma_v^2)$, $w_t \sim \mathcal{N}(0, \sigma_w^2)$.

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Then insert that into the prediction formula:

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Then insert that into the prediction formula:

$$\hat{p}(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}p(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})$$

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Then insert that into the prediction formula:

$$\hat{p}(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}p(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})$$

and the update formula:

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Then insert that into the prediction formula:

$$\hat{p}(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t} p(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})$$

and the update formula:

$$\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{1}{\tilde{C}_{t+1}\cdot N_t}\sum_{i=1}^{N_t} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}).$$

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Then insert that into the prediction formula:

$$\hat{p}(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t} p(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})$$

and the update formula:

$$\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{1}{\tilde{C}_{t+1}\cdot N_t}\sum_{i=1}^{N_t} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}).$$

Problem:

Particle Approximations

Assume a discrete approximation of the posterior density

$$\hat{p}(\mathrm{d}\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t}\delta_{\mathbf{x}_t^{(i)}}(\mathrm{d}\mathbf{x}_t).$$

Then insert that into the prediction formula:

$$\hat{p}(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \frac{1}{N_t}\sum_{i=1}^{N_t} p(\mathrm{d}\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)})$$

and the update formula:

$$\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{1}{\tilde{C}_{t+1}\cdot N_t}\sum_{i=1}^{N_t} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}).$$

Problem: (Generally) Nontrivial to sample from!

Importance Sampling

Importance Sampling lets you sample. . .

Importance Sampling

Importance Sampling lets you sample...

- ...from an (almost) arbitrary density...

Importance Sampling

Importance Sampling lets you sample...

- ...from an (almost) arbitrary density...
- ...that may only be known up to a proportionality constant.

Importance Sampling

Importance Sampling lets you sample...

- ...from an (almost) arbitrary density...
- ...that may only be known up to a proportionality constant.

Recall our particle update formula:

Importance Sampling

Importance Sampling lets you sample...

- ...from an (almost) arbitrary density...
- ...that may only be known up to a proportionality constant.

Recall our particle update formula:

$$\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{1}{\tilde{C}_{t+1} \cdot N_t} \sum_{i=1}^{N_t} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}).$$

Importance Sampling

Importance Sampling lets you sample...

- ...from an (almost) arbitrary density...
- ...that may only be known up to a proportionality constant.

Recall our particle update formula:

$$\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{1}{\tilde{C}_{t+1} \cdot N_t} \sum_{i=1}^{N_t} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}).$$

That's exactly our situation.

Importance Sampling

Importance Sampling lets you sample...

- ...from an (almost) arbitrary density...
- ...that may only be known up to a proportionality constant.

Recall our particle update formula:

$$\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t+1}) = \frac{1}{\tilde{C}_{t+1} \cdot N_t} \sum_{i=1}^{N_t} p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}).$$

That's exactly our situation.

Trick: Give each particle a *weight* in addition to its position.

Importance Sampling in Detail

Suppose we have:

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly
- $\pi(x)$ a different density that is easy to sample from

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly
- $\pi(x)$ a different density that is easy to sample from –the *importance density*

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly
- $\pi(x)$ a different density that is easy to sample from –the *importance density*

Then assign

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly
- $\pi(x)$ a different density that is easy to sample from –the *importance density*

Then assign

$$w^{(i)} \propto \frac{p(x^{(i)})}{\pi(x^{(i)})}$$

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly
- $\pi(x)$ a different density that is easy to sample from –the *importance density*

Then assign and normalize the weights

$$w^{(i)} \propto \frac{p(x^{(i)})}{\pi(x^{(i)})} \qquad \tilde{w}^{(i)} := \frac{w^{(i)}}{\sum_j w^{(j)}}.$$

Importance Sampling in Detail

Suppose we have:

- $p(x) \propto q(x)$ not easy to sample from, q known exactly
- $\pi(x)$ a different density that is easy to sample from –the *importance density*

Then assign and normalize the weights

$$w^{(i)} \propto \frac{p(x^{(i)})}{\pi(x^{(i)})} \qquad \tilde{w}^{(i)} := \frac{w^{(i)}}{\sum_j w^{(j)}}.$$

Any unkown proportionality constant in p or w goes away in the normalization step.

Why and How Importance Sampling Works

Now

$$p(x) \approx \hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}),$$

Why and How Importance Sampling Works

Now

$$p(x) \approx \hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}),$$

assuming the particles $x^{(i)}$ have been drawn using $\pi(x)$.

Why and How Importance Sampling Works

Now

$$p(x) \approx \hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}),$$

assuming the particles $x^{(i)}$ have been drawn using $\pi(x)$. Why?

Why and How Importance Sampling Works

Now

$$p(x) \approx \hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}),$$

assuming the particles $x^{(i)}$ have been drawn using $\pi(x)$.
Why?
A uniformly weighted distribution sampled from $p(x)$ is (approximately) the same as a distribution weighted with $p(x)/\pi(x)$ sampled from $\pi(x)$.

Why and How Importance Sampling Works

Now

$$p(x) \approx \hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}),$$

assuming the particles $x^{(i)}$ have been drawn using $\pi(x)$.
Why?
A uniformly weighted distribution sampled from $p(x)$ is (approximately) the same as a distribution weighted with $p(x)/\pi(x)$ sampled from $\pi(x)$.

$$\int f(x) \frac{p(x)}{\pi(x)} \pi(x) \mathrm{d}x = \int f(x) p(x) \mathrm{d}x.$$

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}).$$

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}).$$

Recall the joint update equation

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}).$$

Recall the joint update equation

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}).$$

Recall the joint update equation

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

This yields the *weight update equation*

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}).$$

Recall the joint update equation

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

This yields the *weight update equation*

$$w_t^{(i)} \propto \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} =$$

Sequential Importance Sampling

Particle Framework + Importance Sampling = ?

To maintain the recursive update property, demand

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \pi(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})\pi(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t}).$$

Recall the joint update equation

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

This yields the *weight update equation*

$$w_t^{(i)} \propto \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{\pi(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} = w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})}.$$

Choice of Importance Density

Choosing a good $\pi(x)$ is extremely important.

Choice of Importance Density

Choosing a good $\pi(x)$ is extremely important.
It can direct particles to interesting parts of the state space–or far away from there.

Choice of Importance Density

Choosing a good $\pi(x)$ is extremely important.
It can direct particles to interesting parts of the state space–or far away from there.

Goal: Keep the weights as uniform as possible.

Choice of Importance Density

Choosing a good $\pi(x)$ is extremely important.
It can direct particles to interesting parts of the state space–or far away from there.

Goal: Keep the weights as uniform as possible.

Optimal choice:

$$\pi_{\text{opt}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}\mathbf{y}_t).$$

Choice of Importance Density

Choosing a good $\pi(x)$ is extremely important.
It can direct particles to interesting parts of the state space–or far away from there.

Goal: Keep the weights as uniform as possible.

Optimal choice:

$$\pi_{\text{opt}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}\mathbf{y}_t).$$

Not usually easy to sample from$\rightarrow$back to original problem.

Choice of Importance Density

Popular choice:

$$\pi_{\text{prior}}(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) := p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)})$$

Choice of Importance Density

Popular choice:

$$\pi_{\text{prior}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) := p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$$

Readily available and often Gaussian→easy choice.

Choice of Importance Density

Popular choice:

$$\pi_{\text{prior}}(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) := p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)})$$

Readily available and often Gaussian→easy choice.
Ignores observations→not always a good choice.

Choice of Importance Density

Popular choice:

$$\pi_{\text{prior}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) := p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$$

Readily available and often Gaussian→easy choice.
Ignores observations→not always a good choice.

Weight update for π_{prior}:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)} = w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}).$$

Choice of Importance Density

Popular choice:

$$\pi_{\text{prior}}(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t) := p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)})$$

Readily available and often Gaussian→easy choice.
Ignores observations→not always a good choice.

Weight update for π_{prior}:

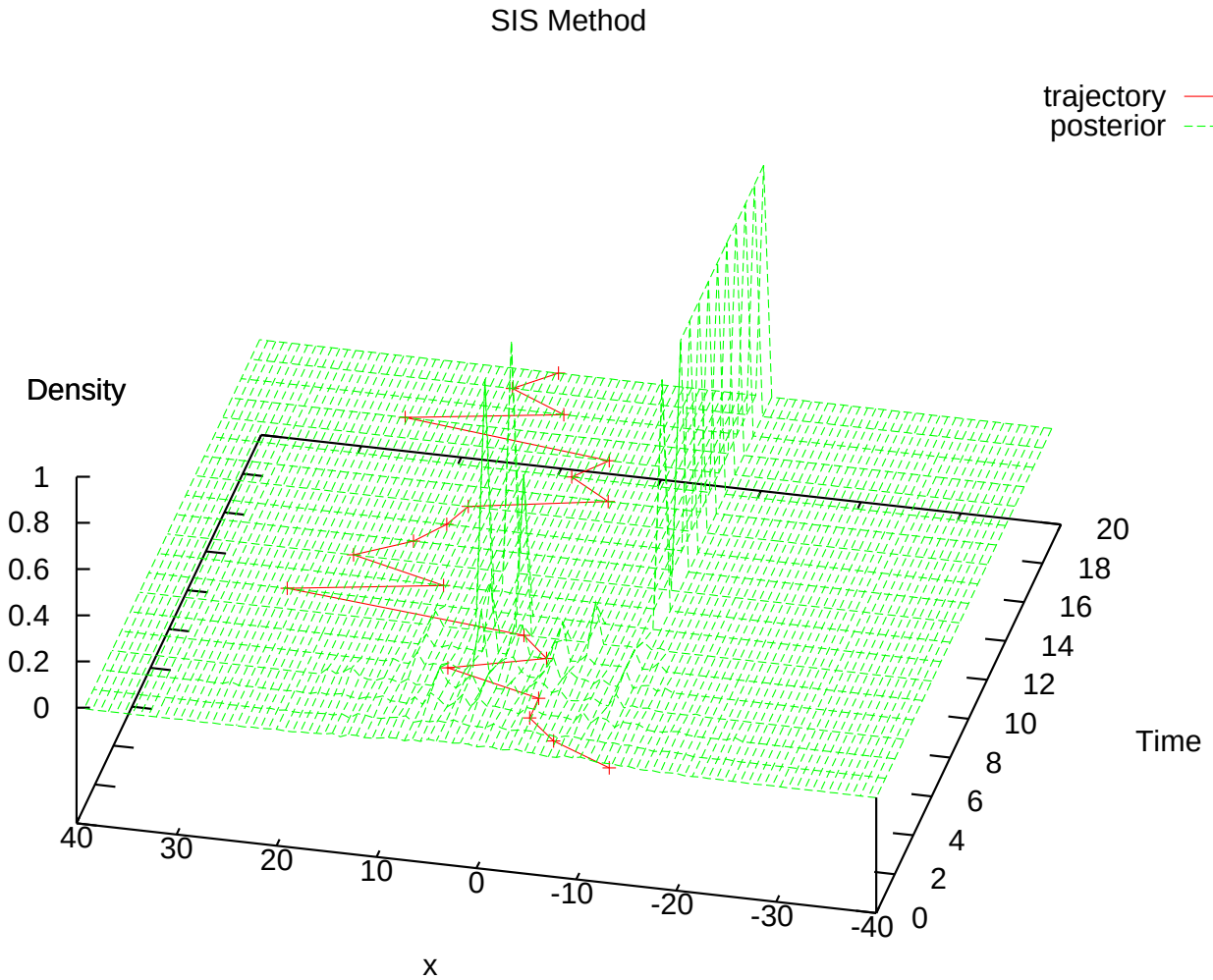
$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{\pi(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{y}_t)} = w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)}).$$

→Actually implementable algorithm.

SIS Implementation


```
def sis(model, n=1000, max_time=20):
    particles = [model.sample_initial() for i in range(n)]
    particle_history = [particles]
    weights = [1/n for i in range(n)]
    weight_history = [weights]

    x = model.sample_initial()
    x_history = [x]

    for t in range(1, max_time):
        x = model.sample_transition(t, x)
        y = model.sample_observation(t, x)
        x_history.append(x)

        evolved_particles = [model.sample_transition(t, xp) for xp
        weights = [weight*model.observation_density(t, y, xtp)
                for weight, xtp in zip(weights, evolved_particles)
        weight_sum = sum(weights)
        weights = [weight/weight_sum for weight in weights]
        particle_history.append(particles)
        weight_history.append(weights)
```

SIS Results



Sequential Importance Resampling

Pure SIS degenerates$\longrightarrow$all weight on one particle.

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight
- Multiply the ones with high weight

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight
- Multiply the ones with high weight
- Rely on process noise to scatter them

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight
- Multiply the ones with high weight
- Rely on process noise to scatter them

How?

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight
- Multiply the ones with high weight
- Rely on process noise to scatter them

How? Sample N times from

$$\hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}).$$

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight
- Multiply the ones with high weight
- Rely on process noise to scatter them

How? Sample N times from

$$\hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}).$$

Particle descendant counts ξ satisfy

$$\xi \sim \text{Multinomial}(n, \tilde{w}^{(1)}, \ldots, \tilde{w}^{(N)}).$$

Sequential Importance Resampling

Pure SIS degenerates→all weight on one particle.

Idea:

- Kill off particles with small weight
- Multiply the ones with high weight
- Rely on process noise to scatter them

How? Sample N times from

$$\hat{p}(x) := \sum_{i=1}^{N} \tilde{w}^{(i)} \delta(x - x^{(i)}).$$

Particle descendant counts ξ satisfy

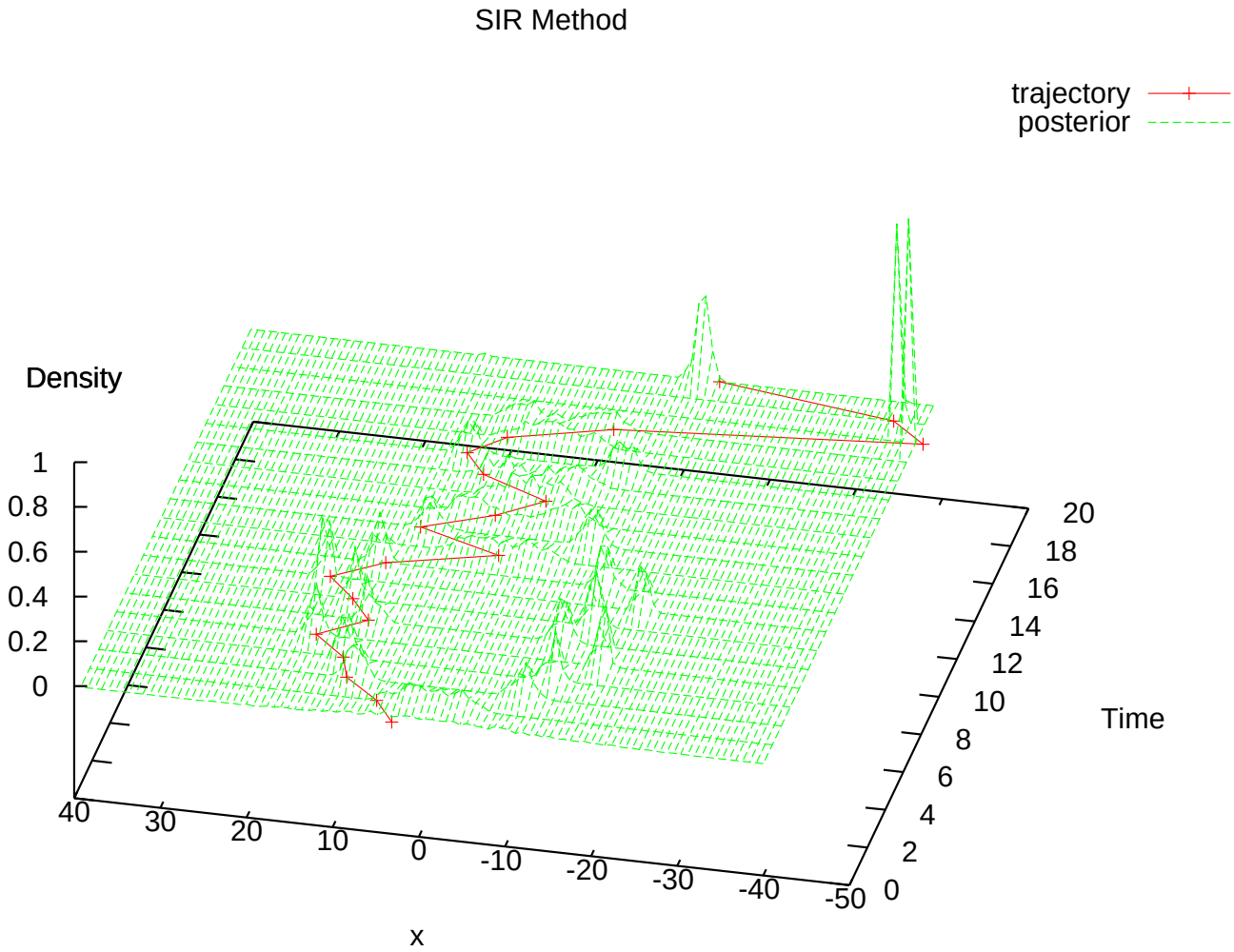
$$\xi \sim \text{Multinomial}(n, \tilde{w}^{(1)}, \ldots, \tilde{w}^{(N)}).$$

→ *Multinomial Resampling*

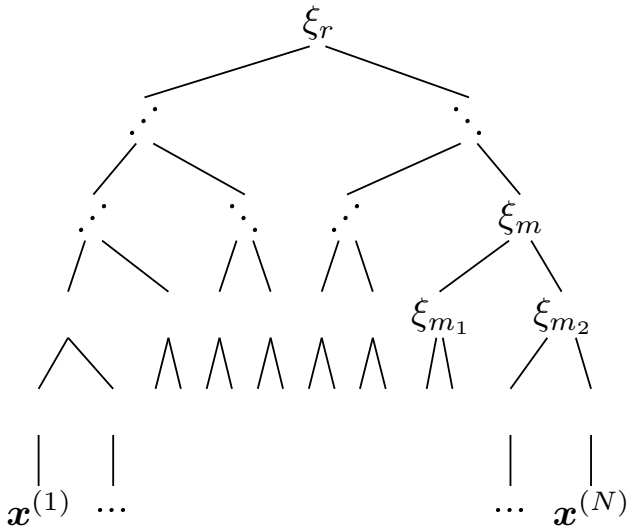
SIR Implementation


```
def sir(model, n=1000, max_time=20):
    particles = [model.sample_initial() for
        i in range(n)]
    particle_history = [particles]

    x = model.sample_initial()
    x_history = [x]

    for t in range(1, max_time):
        x = model.sample_transition(t, x)
        y = model.sample_observation(t, x)
        x_history.append(x)

        evolved_particles = [model.
            sample_transition(t, xp) for xp
            in particles]
        iweights = [model.
            observation_density(t, y, xtp)
            for xtp in evolved_particles]
        particles = sample_discrete(
            evolved_particles, iweights, n)
        particle_history.append(particles)
```

SIR Results

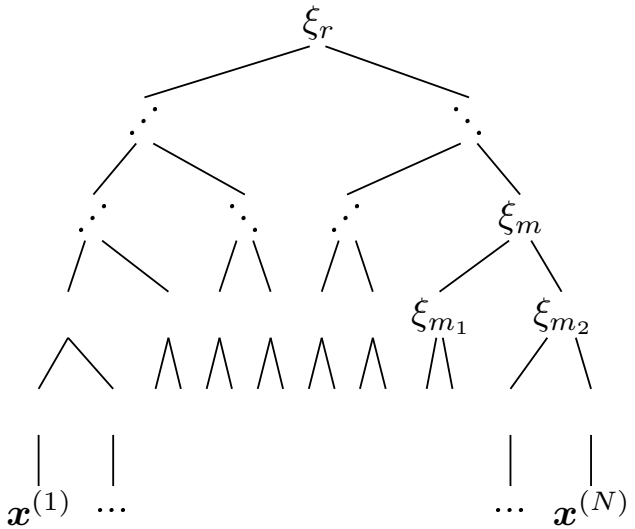


Binary-Tree-based Resampling



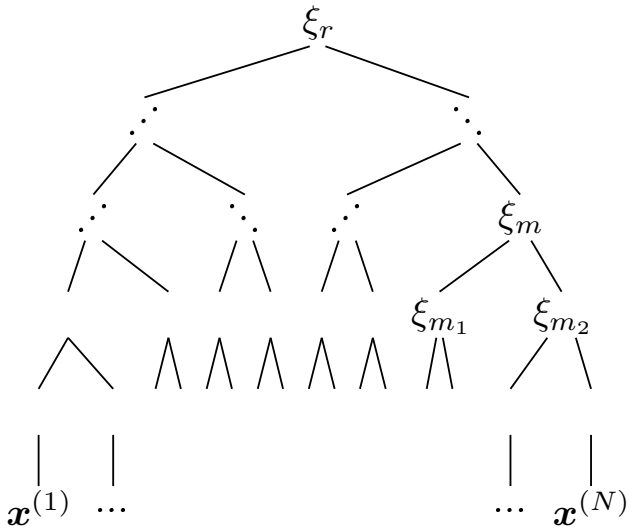
Construct random variables $\xi_m \in \{0, \ldots, N\}$ for each node m,

Binary-Tree-based Resampling



Construct random variables $\xi_m \in \{0, \ldots, N\}$ for each node m, with $\xi_m = \xi_{m_1} + \xi_{m_2}$

Binary-Tree-based Resampling



Construct random variables $\xi_m \in \{0, \ldots, N\}$ for each node m, with $\xi_m = \xi_{m_1} + \xi_{m_2}$ such that $E[\xi_m] = w_m$ and

$$\xi_m = \begin{cases} \lfloor w_m \rfloor & \text{with probability } 1 - w_m - \lfloor w_m \rfloor, \\ \lfloor w_m \rfloor + 1 & \text{with probability } \quad w_m - \lfloor w_m \rfloor. \end{cases}$$

A Convergence Result

Theorem (D. Crisan, 2001)

Let the transition kernel $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ *have the* Feller property, *that is for any bounded and continuous* $f : \mathbb{R}^n \to \mathbb{R}$, *the map*

$$\mathbf{x}_{t-1} \mapsto \int f(x) p(\mathrm{d}\mathbf{x}_t|\mathbf{x}_{t-1})$$

must be bounded and continuous. Then as the number of particles $N \to \infty$, *the approximated posterior densities of the SIR method with multinomial resampling*

$$\hat{p}^N(\mathbf{x}_t|\mathbf{y}_{0:t}) \to p(\mathbf{x}_t|\mathbf{y}_{0:t})$$

p-almost surely.

Outline

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

$$\int_A^B f(x)\mathrm{d}x \approx \sum_{i=1}^{N} f(x^{(i)})w^{(i)},$$

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

$$\int_A^B f(x)\mathrm{d}x \approx \sum_{i=1}^{N} f(x^{(i)})w^{(i)},$$

with some quadrature weights $w^{(i)}$ and points $x^{(i)}$.

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

$$\int_A^B f(x)\mathrm{d}x \approx \sum_{i=1}^{N} f(x^{(i)})w^{(i)},$$

with some quadrature weights $w^{(i)}$ and points $x^{(i)}$.

Consider a probability density function p supported on (A, B).

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

$$\int_A^B f(x)\mathrm{d}x \approx \sum_{i=1}^N f(x^{(i)})w^{(i)},$$

with some quadrature weights $w^{(i)}$ and points $x^{(i)}$.

Consider a probability density function p supported on (A, B).

$$E[f(x)] = \int_A^B f(x)p(x)\mathrm{d}x \approx \sum_{i=1}^N f(x^{(i)})p(x^{(i)})w^{(i)}.$$

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

$$\int_A^B f(x)\mathrm{d}x \approx \sum_{i=1}^N f(x^{(i)})w^{(i)},$$

with some quadrature weights $w^{(i)}$ and points $x^{(i)}$.

Consider a probability density function p supported on (A, B).

$$E[f(x)] = \int_A^B f(x)p(x)\mathrm{d}x \approx \sum_{i=1}^N f(x^{(i)})p(x^{(i)})w^{(i)}.$$

So view the quadrature points $x^{(i)}$ as the particles

Gauss-Legendre Quadrature

Idea: Cleverly reinterpret Gaussian Quadrature as a particle approximation.

Recall:

$$\int_A^B f(x)\mathrm{d}x \approx \sum_{i=1}^N f(x^{(i)})w^{(i)},$$

with some quadrature weights $w^{(i)}$ and points $x^{(i)}$.

Consider a probability density function p supported on (A, B).

$$E[f(x)] = \int_A^B f(x)p(x)\mathrm{d}x \approx \sum_{i=1}^N f(x^{(i)})p(x^{(i)})w^{(i)}.$$

So view the quadrature points $x^{(i)}$ as the particles and

$$\tilde{p}(x^{(i)}) := p(x^{(i)})w^{(i)}$$

as the weights.

Construction of the Quadrature Filter

Recall the Update Formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{C_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

Construction of the Quadrature Filter

Recall the Update Formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{C_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

with

$$C_t = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

Construction of the Quadrature Filter

Recall the Update Formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{C_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

with

$$C_t = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

$$\tilde{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) := \frac{1}{\hat{C}_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t) w_t^{(i)} \underbrace{\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})}_{\text{prediction}}.$$

Construction of the Quadrature Filter

Recall the Update Formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{C_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

with

$$C_t = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

$$\tilde{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) := \frac{1}{\hat{C}_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)w_t^{(i)} \underbrace{\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})}_{\text{prediction}}.$$

Compute this by finding

$$q^{(i)} := p(\mathbf{y}_t|\mathbf{x}_t)w_t^{(i)}\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})$$

Construction of the Quadrature Filter

Recall the Update Formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{C_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

with

$$C_t = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

$$\tilde{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) := \frac{1}{\hat{C}_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t) w_t^{(i)} \underbrace{\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})}_{\text{prediction}}.$$

Compute this by finding

$$\begin{aligned} q^{(i)} &:= p(\mathbf{y}_t|\mathbf{x}_t) w_t^{(i)} \hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) \\ \hat{C}_t &= \sum_{i=1}^{N} q^{(i)}, \end{aligned}$$

Construction of the Quadrature Filter

Recall the Update Formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{1}{C_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$$

with

$$C_t = \int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})\mathrm{d}\mathbf{x}_t.$$

$$\tilde{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) := \frac{1}{\hat{C}_t} \cdot p(\mathbf{y}_t|\mathbf{x}_t) w_t^{(i)} \underbrace{\hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1})}_{\text{prediction}}.$$

Compute this by finding

$$\begin{aligned} q^{(i)} &:= p(\mathbf{y}_t|\mathbf{x}_t) w_t^{(i)} \hat{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t-1}) \\ \hat{C}_t &= \sum_{i=1}^{N} q^{(i)}, \qquad \tilde{p}(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) = \frac{q^{(i)}}{\hat{C}_t}. \end{aligned}$$

Construction of the Filter: Prediction

Recall the original prediction formula:

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_t$$

Construction of the Filter: Prediction

Recall the original prediction formula:

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_t$$

Approximate prediction formula:

$$\hat{p}(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t-1}) = \sum_{j=1}^{N} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)})\tilde{p}(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t})$$

Construction of the Filter: Prediction

Recall the original prediction formula:

$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_t$$

Approximate prediction formula:

$$\hat{p}(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t-1}) = \sum_{j=1}^{N} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)})\tilde{p}(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t})$$

$\tilde{p}(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t})$ already contains quadrature weights.

Construction of the Filter: Prediction

Recall the original prediction formula:

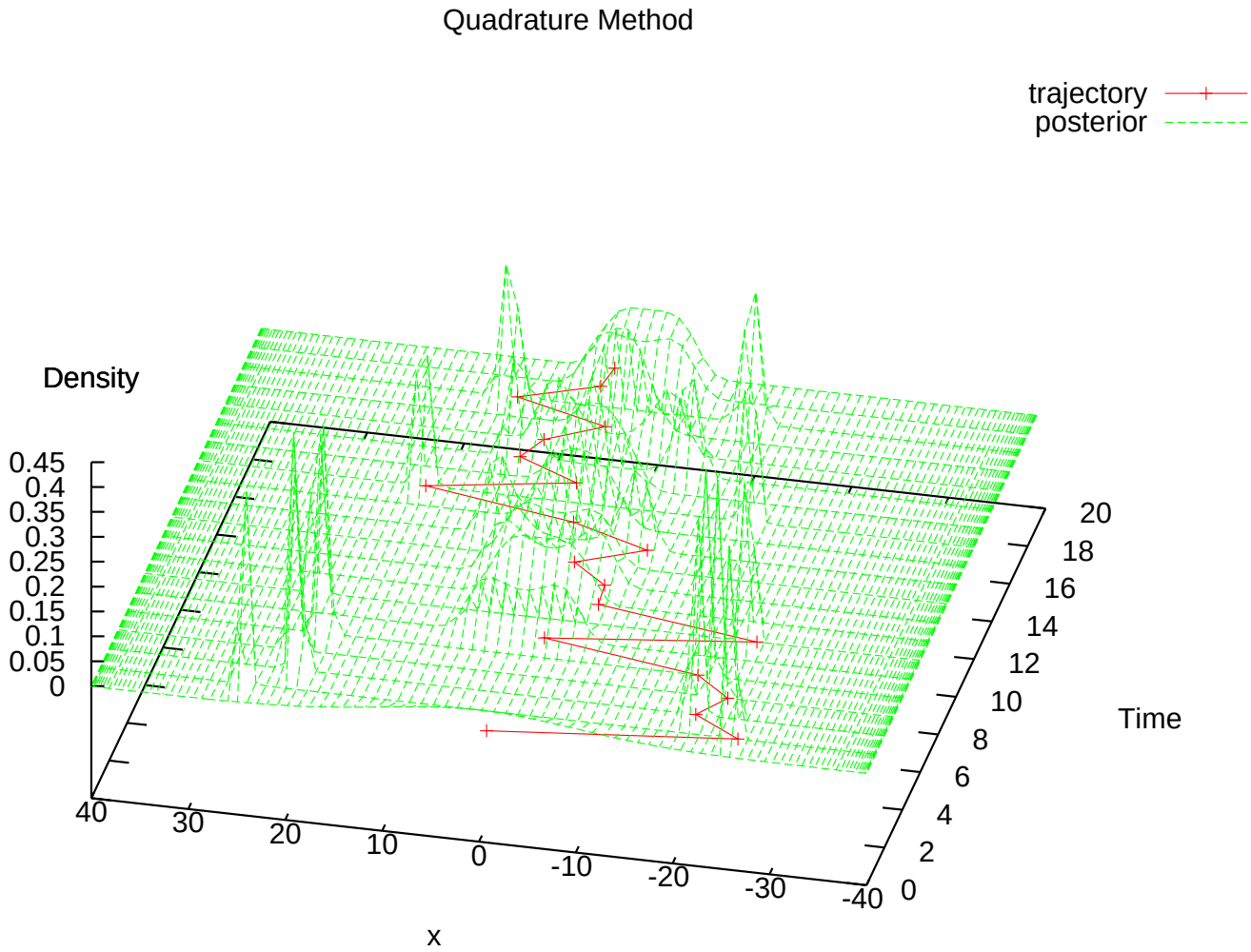
$$p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t})\mathrm{d}\mathbf{x}_t$$

Approximate prediction formula:

$$\hat{p}(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t-1}) = \sum_{j=1}^{N} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)})\tilde{p}(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t})$$

$\tilde{p}(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t})$ already contains quadrature weights.
Note: $O(N^2)$ complexity.

Quadrature Filter Results



Features of the Quadrature Filter

- Deterministic, no sampling.

Features of the Quadrature Filter

- Deterministic, no sampling.
- Well-suited to comparing likelihoods

$$p(\mathbf{y}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \prod_{t=1}^{T} C_t$$

of different models.

Features of the Quadrature Filter

- Deterministic, no sampling.
- Well-suited to comparing likelihoods

$$p(\mathbf{y}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \prod_{t=1}^{T} C_t$$

  of different models.
- Suffers heavily with high-dimensional statespaces: $O(N^2)$, $N = m^d$. (SIS: $O(N)$, SIR: $O(N \log N)$)

Features of the Quadrature Filter

- Deterministic, no sampling.
- Well-suited to comparing likelihoods

$$p(\mathbf{y}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \prod_{t=1}^{T} C_t$$

  of different models.
- Suffers heavily with high-dimensional statespaces: $O(N^2)$, $N = m^d$. (SIS: $O(N)$, SIR: $O(N \log N)$)
- Requires far fewer particles than SMC for comparable accuracy.

Features of the Quadrature Filter

- Deterministic, no sampling.
- Well-suited to comparing likelihoods

$$p(\mathbf{y}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \prod_{t=1}^{T} C_t$$

  of different models.
- Suffers heavily with high-dimensional statespaces: $O(N^2)$, $N = m^d$. (SIS: $O(N)$, SIR: $O(N \log N)$)
- Requires far fewer particles than SMC for comparable accuracy.
- Explicit error estimates available.

Error Estimates: Setup

Error in the log-likelihood of the observation:

$$\delta_t := \log\{\hat{p}(\mathbf{y}_{1:t})\} - \log\{p(\mathbf{y}_{1:t})\} = \sum_{s=1}^{t} \log \hat{C}_s - \sum_{s=1}^{t} \log C_s.$$

Error Estimates: Setup

Error in the log-likelihood of the observation:

$$\delta_t := \log\{\hat{p}(\mathbf{y}_{1:t})\} - \log\{p(\mathbf{y}_{1:t})\} = \sum_{s=1}^{t} \log \hat{C}_s - \sum_{s=1}^{t} \log C_s.$$

Prediction error indicator:

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) := \exp(\delta_t)\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).$$

Error Estimates: Setup

Error in the log-likelihood of the observation:

$$\delta_t := \log\{\hat{p}(\mathbf{y}_{1:t})\} - \log\{p(\mathbf{y}_{1:t})\} = \sum_{s=1}^{t} \log \hat{C}_s - \sum_{s=1}^{t} \log C_s.$$

Prediction error indicator:

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) := \exp(\delta_t)\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).$$

Indicator ignores error in normalization $\hat{C}_t$

Error Estimates: Setup

Error in the log-likelihood of the observation:

$$\delta_t := \log\{\hat{p}(\mathbf{y}_{1:t})\} - \log\{p(\mathbf{y}_{1:t})\} = \sum_{s=1}^{t} \log \hat{C}_s - \sum_{s=1}^{t} \log C_s.$$

Prediction error indicator:

$$\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) := \exp(\delta_t)\hat{p}(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).$$

Indicator ignores error in normalization $\hat{C}_t \longrightarrow exp(\delta_t)$ necessary.

Error Propagation Recursion

A nice bit of algebra (see notes) shows

$$\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
= \quad & C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \underline{\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})} \\
& + \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).
\end{aligned}$$

Error Propagation Recursion

A nice bit of algebra (see notes) shows

$$\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
= \quad & C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \underline{\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})} \\
& + \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).
\end{aligned}$$

Split error into

Error Propagation Recursion

A nice bit of algebra (see notes) shows

$$\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
= \quad & C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \underline{\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})} \\
& + \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).
\end{aligned}$$

Split error into

- Method error (First line)

Error Propagation Recursion

A nice bit of algebra (see notes) shows

$$\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
= \quad & C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \underline{\varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})} \\
& + \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).
\end{aligned}$$

Split error into

- Method error (First line)
- Quadrature error (Second line) on *exact probabilities*

Error Estimates: Assumptions

- *Approximation Assumption*: Assume we have enough particles so that

Error Estimates: Assumptions

- *Approximation Assumption*: Assume we have enough particles so that

$$|\sum_j w^{(j)} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)}) p(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})|$$

$$\leqslant \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}).$$

Error Estimates: Assumptions

- *Approximation Assumption*: Assume we have enough particles so that

$$|\sum_j w^{(j)} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)}) p(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})|$$

$$\leqslant \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}).$$

Note: Assumption is on quadrature of *exact* probabilities.

Error Estimates: Assumptions

- *Approximation Assumption*: Assume we have enough particles so that

$$|\sum_j w^{(j)} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)}) p(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})|$$

$$\leqslant \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}).$$

  Note: Assumption is on quadrature of *exact* probabilities.

- *Induction Assumption*:

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon,$$

Error Estimates: Assumptions

- *Approximation Assumption*: Assume we have enough particles so that

$$|\sum_j w^{(j)} p(\mathbf{x}_{t+1}^{(i)}|\mathbf{x}_t^{(j)}) p(\mathbf{x}_t^{(j)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})|$$

$$\leqslant \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}).$$

  Note: Assumption is on quadrature of *exact* probabilities.

- *Induction Assumption*:

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon,$$

  where r_t is the relative error at time t, i.e. the smallest positive real such that

$$|\varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})| \leqslant r_t p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}).$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\leqslant \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + |\sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})|
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\leqslant \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + |\sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) - p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})| \\
\overset{\text{AA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\overset{\text{AA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\overset{\text{AA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} \varepsilon(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}) \\
\overset{\text{IA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} r_{t-1} p(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\overset{\text{IA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} r_{t-1} p(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\overset{\text{IA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} r_{t-1} p(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}) \\
= \quad & |r_{t-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\overset{\text{IA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} r_{t-1} p(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}) \\
= \quad & |r_{t-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}) \\
\overset{\text{AA}}{\leqslant} \quad & r_{t-1} |p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})(1+\varepsilon)| + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}).
\end{aligned}
$$

Proof of Error Bound

$$
\begin{aligned}
& \varepsilon(\mathbf{x}_{t+1}|\mathbf{y}_{1:t}) \\
\overset{\text{IA}}{\leqslant} \quad & |C_t^{-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) p(\mathbf{y}_t|\mathbf{x}_t^{(i)}) w_t^{(i)} r_{t-1} p(\mathbf{x}_t|\mathbf{y}_{1:t-1})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}) \\
= \quad & |r_{t-1} \sum_i p(\mathbf{x}_{t+1}|\mathbf{x}_t^{(i)}) w_t^{(i)} p(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t})| \\
& + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}) \\
\overset{\text{AA}}{\leqslant} \quad & r_{t-1} |p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t})(1+\varepsilon)| + \varepsilon p(\mathbf{x}_{t+1}^{(i)}|\mathbf{y}_{1:t}). \\
\Rightarrow \quad & r_t \leqslant r_{t-1}(1+\varepsilon) + \varepsilon.
\end{aligned}
$$

Error Estimates: Summary

We showed

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon.$$

Error Estimates: Summary

We showed

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon.$$

This implies

$$r_t \leqslant (1+\varepsilon)^t - 1.$$

Error Estimates: Summary

We showed

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon.$$

This implies

$$r_t \leqslant (1+\varepsilon)^t - 1.$$

$\longrightarrow$ Potentially *exponential* error growth.

Error Estimates: Summary

We showed

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon.$$

This implies

$$r_t \leqslant (1+\varepsilon)^t - 1.$$

$\longrightarrow$ Potentially *exponential* error growth.

A similar proof shows

$$\delta_t \leqslant (t+1)\log(1+\varepsilon')$$

Error Estimates: Summary

We showed

$$r_{t+1} \leqslant r_t(1+\varepsilon) + \varepsilon.$$

This implies

$$r_t \leqslant (1+\varepsilon)^t - 1.$$

$\longrightarrow$ Potentially *exponential* error growth.

A similar proof shows

$$\delta_t \leqslant (t+1)\log(1+\varepsilon')$$

$\longrightarrow$ *Linear* error growth in the *log*-likelihood.

Outline

Continuous-Time Nonlinear Filtering

Have:

- A *Markov diffusion process*

$$\mathrm{d}x^i(t) = b^i(\mathbf{x}(t))\mathrm{d}t + \sigma^{ij}(\mathbf{x}(t))\mathrm{d}W^j(t)$$

where $\mathbf{x} = (x^1, \ldots, x^d)$ and $\mathbf{x}(0) = \mathbf{x}_0$.

Continuous-Time Nonlinear Filtering

Have:

- A *Markov diffusion process*

$$\mathrm{d}x^i(t) = b^i(\mathbf{x}(t))\mathrm{d}t + \sigma^{ij}(\mathbf{x}(t))\mathrm{d}W^j(t)$$

  where $\mathbf{x} = (x^1, \ldots, x^d)$ and $\mathbf{x}(0) = \mathbf{x}_0$.

- An *observation*

$$\mathbf{y}(t) = \int_0^t \mathbf{h}(\mathbf{x}(s))\mathrm{d}s + \mathbf{V}(t).$$

Continuous-Time Nonlinear Filtering

Have:

- A *Markov diffusion process*

$$\mathrm{d}x^i(t) = b^i(\mathbf{x}(t))\mathrm{d}t + \sigma^{ij}(\mathbf{x}(t))\mathrm{d}W^j(t)$$

  where $\mathbf{x} = (x^1, \ldots, x^d)$ and $\mathbf{x}(0) = \mathbf{x}_0$.

- An *observation*

$$\mathbf{y}(t) = \int_0^t \mathbf{h}(\mathbf{x}(s))\mathrm{d}s + \mathbf{V}(t).$$

All coefficients are assumed smooth,

Continuous-Time Nonlinear Filtering

Have:

- A *Markov diffusion process*

$$\mathrm{d}x^i(t) = b^i(\mathbf{x}(t))\mathrm{d}t + \sigma^{ij}(\mathbf{x}(t))\mathrm{d}W^j(t)$$

  where $\mathbf{x} = (x^1, \ldots, x^d)$ and $\mathbf{x}(0) = \mathbf{x}_0$.

- An *observation*

$$\mathbf{y}(t) = \int_0^t \mathbf{h}(\mathbf{x}(s))\mathrm{d}s + \mathbf{V}(t).$$

All coefficients are assumed smooth, and $\mathbf{x}_0$ and the Wiener processes $\mathbf{W}$ and $\mathbf{V}$ are assumed to be independent.

The Optimal Continuous-Time Filter

The optimal filter (= best mean-square estimate) for $f(\mathbf{x}(t))$ is

The Optimal Continuous-Time Filter

The optimal filter (= best mean-square estimate) for $f(\mathbf{x}(t))$ is

$$\hat{f}(\mathbf{x}(t)) = \frac{\int_{\mathbb{R}^d} f(\mathbf{x}) u(t, \mathbf{x}) \mathrm{d}\mathbf{x}}{\int_{\mathbb{R}^d} u(t, \mathbf{x}) \mathrm{d}\mathbf{x}},$$

The Optimal Continuous-Time Filter

The optimal filter (= best mean-square estimate) for $f(\mathbf{x}(t))$ is

$$\hat{f}(\mathbf{x}(t)) = \frac{\int_{\mathbb{R}^d} f(\mathbf{x}) u(t, \mathbf{x}) \mathrm{d}\mathbf{x}}{\int_{\mathbb{R}^d} u(t, \mathbf{x}) \mathrm{d}\mathbf{x}},$$

where u is the *unnormalized filtering density*.

The Optimal Continuous-Time Filter

The optimal filter (= best mean-square estimate) for $f(\mathbf{x}(t))$ is

$$\hat{f}(\mathbf{x}(t)) = \frac{\int_{\mathbb{R}^d} f(\mathbf{x}) u(t, \mathbf{x}) \mathrm{d}\mathbf{x}}{\int_{\mathbb{R}^d} u(t, \mathbf{x}) \mathrm{d}\mathbf{x}},$$

where u is the *unnormalized filtering density*.
u obeys the *Zakai SPDE*

$$\mathrm{d}u(t, \mathbf{x}) = \mathcal{L}^* u(t, \mathbf{x}) \mathrm{d}t + h(\mathbf{x}) u(t, \mathbf{x}) \mathrm{d}\mathbf{y}(t),$$

The Optimal Continuous-Time Filter

The optimal filter (= best mean-square estimate) for $f(\mathbf{x}(t))$ is

$$\hat{f}(\mathbf{x}(t)) = \frac{\int_{\mathbb{R}^d} f(\mathbf{x}) u(t, \mathbf{x}) \mathrm{d}\mathbf{x}}{\int_{\mathbb{R}^d} u(t, \mathbf{x}) \mathrm{d}\mathbf{x}},$$

where u is the *unnormalized filtering density*.
u obeys the *Zakai SPDE*

$$\mathrm{d}u(t, \mathbf{x}) = \mathcal{L}^* u(t, \mathbf{x}) \mathrm{d}t + h(\mathbf{x}) u(t, \mathbf{x}) \mathrm{d}\mathbf{y}(t),$$

where

$$\mathcal{L}^* u := \frac{1}{2} \frac{\partial^2}{\partial x_i x_j} ((\sigma\sigma^*)^{ij} u) - \frac{\partial}{\partial x_i} (b^i u).$$

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

$$u(t,\mathbf{x}) = \sum_\alpha \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t,\mathbf{x}) \xi_\alpha(\mathbf{y}).$$

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

$$u(t,\mathbf{x}) = \sum_\alpha \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t,\mathbf{x}) \xi_\alpha(\mathbf{y}).$$

- Coefficients φ_α satisfy a deterministic system of PDEs.

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

$$u(t,\mathbf{x}) = \sum_\alpha \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t,\mathbf{x}) \xi_\alpha(\mathbf{y}).$$

- Coefficients φ_α satisfy a deterministic system of PDEs.
- The scheme separates

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

$$u(t,\mathbf{x}) = \sum_\alpha \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t,\mathbf{x}) \xi_\alpha(\mathbf{y}).$$

- Coefficients φ_α satisfy a deterministic system of PDEs.
- The scheme separates
  - dependency on process parameters (φ_α) from

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

$$u(t,\mathbf{x}) = \sum_\alpha \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t,\mathbf{x}) \xi_\alpha(\mathbf{y}).$$

- Coefficients φ_α satisfy a deterministic system of PDEs.
- The scheme separates
  - dependency on process parameters (φ_α) from
  - dependency on observation (ξ_α).

One possible Solution Method

Sergey Lototsky's *spectral separating scheme* expresses u as an expansion into *Wick polynomials* ξ_α:

$$u(t,\mathbf{x}) = \sum_\alpha \frac{1}{\sqrt{\alpha!}} \varphi_\alpha(t,\mathbf{x}) \xi_\alpha(\mathbf{y}).$$

- Coefficients φ_α satisfy a deterministic system of PDEs.
- The scheme separates
  - dependency on process parameters (φ_α) from
  - dependency on observation (ξ_α).
- φ_α can be precomputed.

Questions?

?