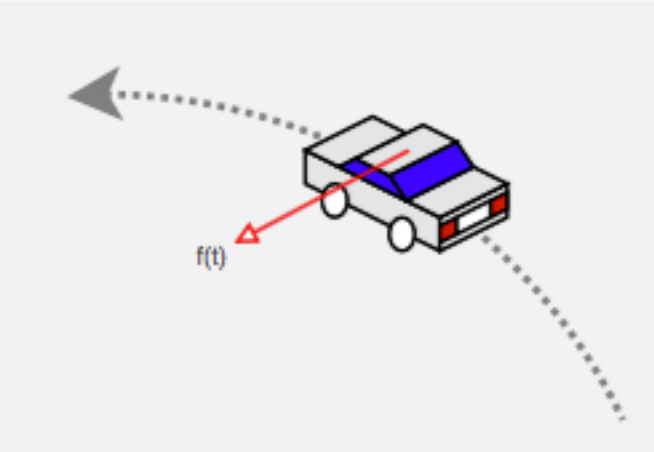
TAMPERE UNIVERSITY OF TECHNOLOGY

Monte Carlo methods for dynamical systems



Henri Nurminen

henri.nurminen@tut.fi

Dept. of Automation Science and Engineering

8.9.2016

This talk is about estimation of a dynamical state with noisy observations

1. Monte Carlo integration & Importance sampling
2. Statistical modeling of dynamical systems
3. Particle filter algorithm
4. Application examples

If you cannot integrate analytically, you can try Monte Carlo

$$\mathrm{E}(g(\mathbf{x})) = \int g(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$$

Solution: Generate N random vector realisations

$$\mathbf{x}^{(i)} \sim p(\mathbf{x})$$

and approximate

$$\mathrm{E}(g(\mathbf{x})) \approx \sum_{i=1}^{N} g(\mathbf{x}^{(i)})$$

What if you cannot generate from $p(\mathbf{x})$ easily?

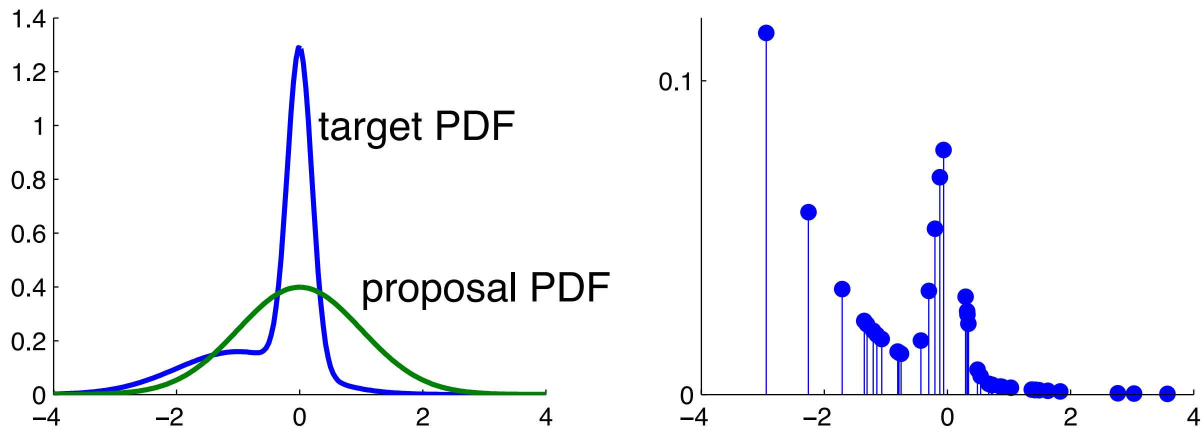
Importance sampling is a more flexible way to form a particle set

$$\mathrm{E}(\mathbf{g}(\mathbf{x})) \approx \sum_{i=1}^{N} \tilde{w}^{(i)} \mathbf{g}(\mathbf{x}^{(i)})$$

importance weights

$$\text{where } \mathbf{x}^{(i)} \sim \pi \text{ and } \tilde{w}^{(i)} = \frac{p(\mathbf{x}^{(i)})}{N\pi(\mathbf{x}^{(i)})}$$

proposal



proof: $$\mathrm{E}(\mathbf{g}(\mathbf{x})) = \int \mathbf{g}(\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x} = \int \mathbf{g}(\mathbf{x}) \frac{p(\mathbf{x})}{\pi(\mathbf{x})} \pi(\mathbf{x}) \mathrm{d}\mathbf{x}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{g}(\mathbf{x}^{(i)}) \frac{p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)})}, \quad \mathbf{x}^{(i)} \sim \pi(\mathbf{x})$$

Bayes rule: knowledge comes from prior information and measurements

likelihood *prior*

state (unknown)

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})\,p(\mathbf{x})}{p(\mathbf{y})}$$

measurement (known)

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})\,p(\mathbf{x})$$

"Posterior is prior times measurement likelihood."

Importance sampling can be used in Bayesian inference

$$\mathrm{E}(g(\mathbf{x})) \approx \sum_{i=1}^{N} w^{(i)} g(\mathbf{x}^{(i)})$$

where $\mathbf{x}^{(i)} \sim p(\mathbf{x})$

and $\tilde{w}^{(i)} = p(\mathbf{y}|\mathbf{x}^{(i)})$ normalized to $w^{(i)} = \frac{\tilde{w}^{(i)}}{\sum_{i=1}^{N} \tilde{w}^{(i)}}$

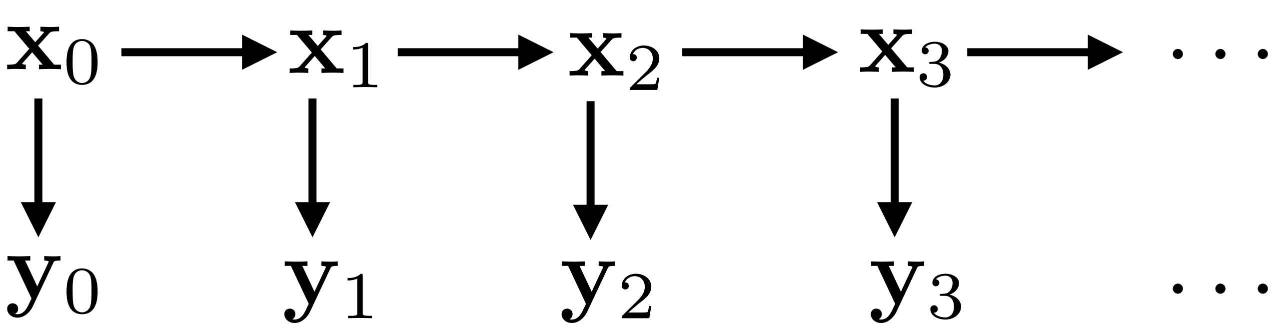
Generate random samples from prior, give more weight to samples that support the obtained measurement.

State-space model describes dynamical systems with noisy measurements

$$\mathbf{x}_k = f_k(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \qquad \mathbf{w}_{k-1} \sim D_k^{\mathbf{w}}$$

$$\mathbf{y}_k = h_k(\mathbf{x}_k) + \mathbf{v}_k \qquad \mathbf{v}_k \sim D_k^{\mathbf{v}}$$

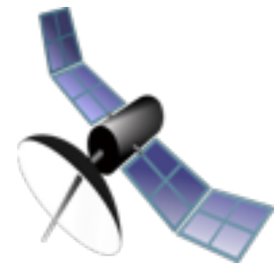
- $\mathbf{y}_k$: measurement (observed)
- $\mathbf{x}_k$: state
- $\mathbf{v}_k$: measurement noise
- $\mathbf{w}_{k-1}$: process noise

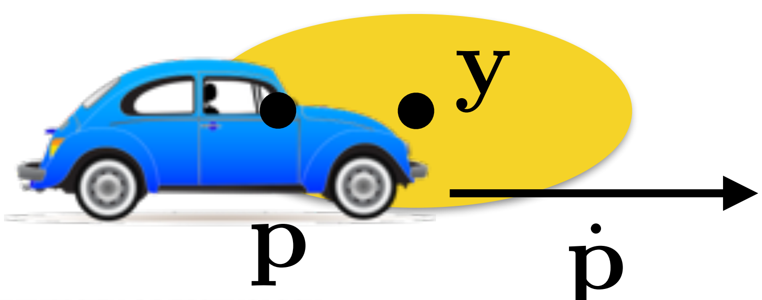


Example: Almost-constant-velocity model with position measurements

$$\begin{bmatrix} \mathbf{p}_k \\ \dot{\mathbf{p}}_k \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{k-1} + \Delta \dot{\mathbf{p}}_{k-1} \\ \dot{\mathbf{p}}_{k-1} \end{bmatrix} + \mathbf{w}_{k-1}$$

$$\mathbf{y}_k = \mathbf{p}_k + \mathbf{v}_k$$





Bayesian filter solves the filtering distribution

Problem: Find state **x** given the noisy measurements **y.**

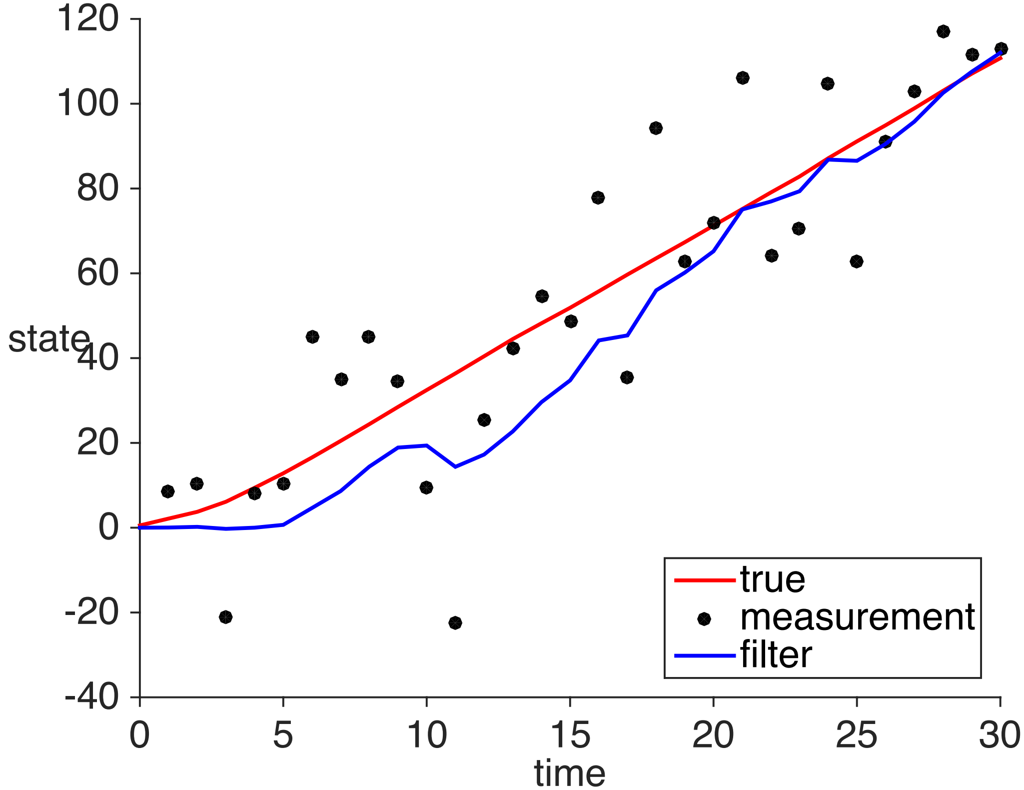
Given $p(x_0)$, $p(x_k|x_{k-1})$, $p(y_k|x_k)$, find $p(x_k|y_{1:k})$.

Solution: Use the recursion

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1})\, p(x_{k-1} \mid y_{1:k-1})\, \mathrm{d}x_{k-1}$$

$$p(x_k \mid y_{1:k}) \propto p(y_k \mid x_k)\, p(x_k \mid y_{1:k-1})$$

Bayesian filtering smooths the estimated trajectory



Side note: state model can be a discretised continuous-time model

Stochastic differential equation

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\,\mathrm{d}t + \mathbf{g}(\mathbf{x}, t)\,\mathrm{d}\beta$$

Brownian motion random element

The source of randomness can be:

- the ground truth is not known exactly
- nature (quantum mechanics)

Bayesian filter is generally untractable

Kalman filter: the analytic solution for linear-Gaussian case

$$
\begin{aligned}
x_{k|k-1} &= F_k x_{k-1|k-1} \\
P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_{k-1} \\
K_k &= P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1} \\
x_{k|k} &= x_{k|k-1} + K_k (y_k - H_k x_{k|k-1}) \\
P_{k|k} &= (I - K_k H_k) P_{k|k-1}
\end{aligned}
$$

For almost any other model, one has to approximate.

Solution: Sequential Importance Sampling (SIR)

SIS gives a Monte Carlo approximation of the posterior of a dynamical system

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0) \qquad w_0^{(i)} = \frac{1}{N}$$

initialization

for $k = 1, 2, \ldots$

$$\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$$

"guessing" new state

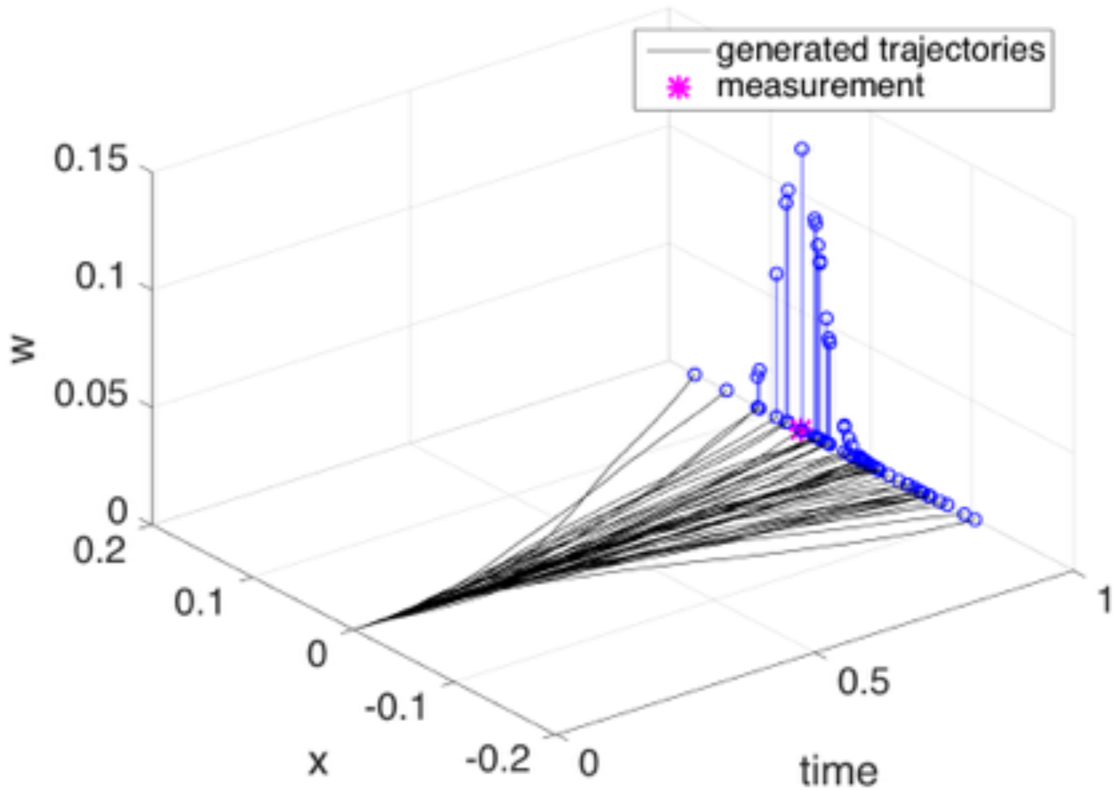
$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \cdot p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \qquad w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N} \tilde{w}_k^{(i)}}$$

weighting & normalisation

$$\mathrm{E}(g(\mathbf{x}_k)) \approx \sum_{i=1}^{N} w_k^{(i)} g(\mathbf{x}_k^{(i)})$$

point estimate

SIS means “guessing” the trajectory and weighting based on measurement



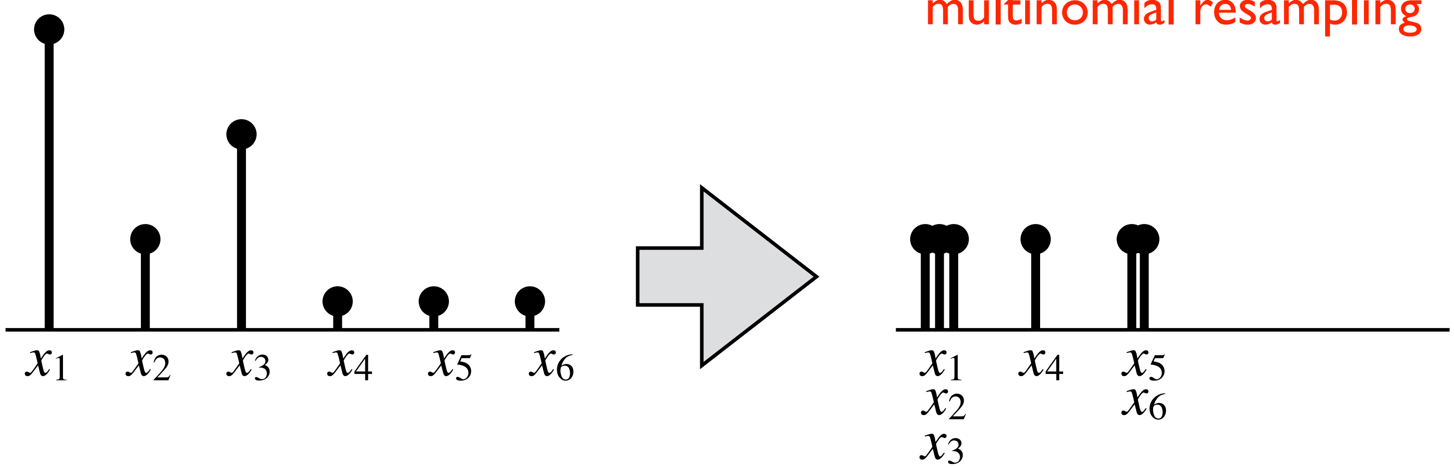
Resampling is needed, otherwise the weights degenerate (i.e. all but one go to zero)

let Π_k be the distribution on $1{:}N$ with probabilities $w_k^{(1:N)}$

draw $j_1, \ldots, j_N$ from Π_k — categorical distribution

replace $\mathbf{x}_k^{(1:N)} \leftarrow \mathbf{x}_k^{(j_{1:N})}$ and $w_k^{(1:N)} \leftarrow \frac{1}{N}$

multinomial resampling



Need not be done at every step, can be done e.g. when

$$\frac{1}{\sum_{i=1}^{N}(w_k^{(i)})^2} \leq 0.1 \cdot N$$

Sequential Importance Resampling (SIR)

SIR gives a Monte Carlo approximation of the posterior of a dynamical system

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0) \qquad w_0^{(i)} = \frac{1}{N}$$

initialise

for $k = 1, 2, \ldots$

$$\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)})$$

"guess" new state

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \cdot p(\mathbf{y}_k | \mathbf{x}_k^{(i)}) \qquad w_k^{(i)} = \frac{\tilde{w}_k^{(i)}}{\sum_{i=1}^{N} \tilde{w}_k^{(i)}}$$

weight & normalise

$$\mathrm{E}(g(\mathbf{x}_k)) \approx \sum_{i=1}^{N} w_k^{(i)} g(\mathbf{x}_k^{(i)})$$

point estimate

resampling

only best guesses survive

Common terminology

particle ≜ one Monte Carlo sample of the current state / of the trajectory

particle filter ≜ SIR

Example 1: Indoor location becomes more accurate by downweighting wall-collided particles

Example 2: Aircraft localisation with altitude measurements

https://www.youtube.com/watch?v=aUkBa1zMKv4

Demo by A. Svensson, Uppsala University, 2013.

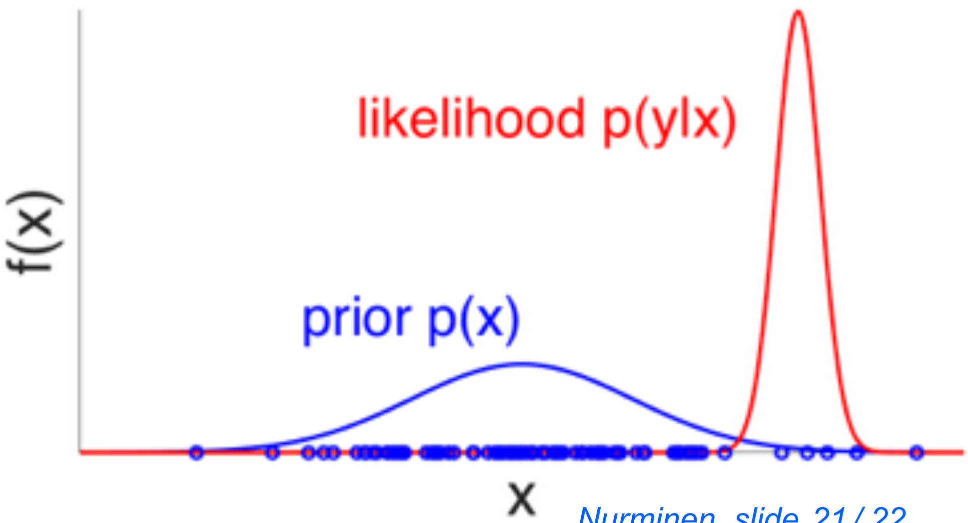
Particle methods are flexible but computationally expensive

- ✦ Very flexible: work with almost any model.
- ✦ Mathematically rigorous: $N \to \infty$ convergence proved. Good reference solution even for real-time stuff.
- ✦ Give full probability distributions, not only point estimates.
- ✦ Easy to code and often easy to understand.

– Computationally heavy for some models
– Do not work with constant parameters
– Curse of dimensionality

Particle filter has lots of extensions and active research directions

- particle smoothers (estimate the whole trajectory)
- Rao-Blackwellized particle filters (do some dimensions analytically, more efficiency)
- Fighting particle degeneracy:
  - more efficient importance distributions
  - particle flow filters
  - …



Literature

[1] Gordon N.J., Salmond D.J., and Smith A.F.M. 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE-Proceedings-F 140: 107–113.

[2] Doucet A., Godsill S., Andrieu C. 2000. On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing, vol. 10.

[3] Ristic B., Arulampalam S., Gordon N. 2004. Beyond the Kalman Filter. Artech House Publishers.

[4] Särkkä S. 2013. Bayesian Filtering and Smoothing. Cambridge University Press.