An Introduction to Particle Filters

Hansruedi Künsch
Seminar für Statistik, D-Math
ETH Zürich

GMFH-Versammlung, Biel, 15. September 2012

Contents

A few references

- A precursor: Handschin, J. E., and Mayne, D. Q. Intern. J. Control 9, 1969, 547-559.
- The first modern paper about particle filters: Gordon, N. J., Salmond, D. J., and Smith, A. F. M., IEE Proceedings-F 140, 1993, 107-113.
- A collection of short papers by many authors: Doucet, A., de Freitas, N., and Gordon, N. (eds), Sequential Monte Carlo Methods in Practice, Springer, 2001.
- An introductory review chapter: Künsch, H. R., in Complex Stochastic Systems, Barndorff-Nielsen, O.E., Cox, D. R., and Klüppelberg, C. (eds.), Chapman and Hall 2001.
- A modern tutorial: Doucet, A. and Johansen, A. M., in Handbook on Nonlinear Filtering, Crisan, D., and Rozovskii, B. (eds.), Oxford University Press, 2011.

Linear Gaussian state space models

State evolution (in discrete time):

$$X_t = F_t X_{t-1} + V_t$$

Observation equations:

$$Y_t = H_t X_t + W_t.$$

where $V_t \sim \mathcal{N}(0, Q_t)$, $W_t \sim \mathcal{N}(0, R_t)$ uncorrelated white noises, F_t, H_t matrices of appropriate dimensions.

The Kalman filter computes recursively estimates $\hat{x}_{t|t-1}$ and $\hat{x}_{t|t}$ of the state at time t based on observations up to time $t-1$ and t respectively, together with the error covariance matrices $\Sigma_{t|t-1}$ and $\Sigma_{t|t}$.

Kalman filter recursions

Each recursion consists of a **prediction step**

$$\hat{x}_{t|t-1} = F_t \hat{x}_{t-1|t-1}, \quad \Sigma_{t|t-1} = \Sigma_{t-1|t-1} + Q_t,$$

followed by an **update step**

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - H_t \hat{x}_{t|t-1}), \quad \Sigma_{t|t} = \Sigma_{t|t-1} - K_t H_t \Sigma_{t|t-1}$$

where K_t is the Kalman gain.

The estimates have the following two properties

- The estimates are **unbiased** (no systematic error).
- The **error covariance matrix** is **minimal**.

Nonlinear/Non-Gaussian state space models

$$X_t = F_t(X_{t-1}, V_t), \quad Y_t = H_t(X_t, W_t)$$

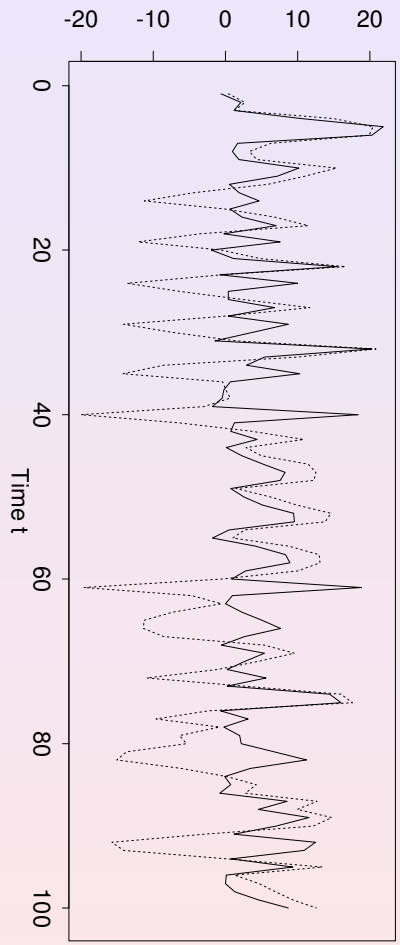
where (V_t) and (W_t) are two independent, white, but not necessarily Gaussian noises, and F_t and H_t are known functions.

As an example, the figure below shows a simulation of the model

$$X_t = \alpha X_{t-1} + \beta \frac{X_{t-1}}{1 + X_{t-1}^2} + \gamma \cos(1.2t) + V_t, \quad Y_t = \frac{X_t^2}{20} + W_t$$

which goes back to Andrade Netto et al., IEEE Trans. Autom. Control (1978).

Simulation from a nonlinear state space model



Other examples

Stochastic volatility in finance:
Observations = log returns of an asset, state = unobservable variances of log returns:

$$X_t = \nu + \alpha X_{t-1} + V_t, \quad Y_t = \exp(X_t/2)\, W_t.$$

If we use $\log|Y_t|$, we obtain a linear state space model with a strongly non-Gaussian noise, but this is not really a simplification as will be discussed below.

Others: Tracking problems (“bearings only”), Computer vision, Stochastic kinetic models in biology, Meteorology, etc.

Structure of general state space models

State evolution $X_t = F_t(X_{t-1}, V_t)$ defines a Markov process in discrete time (past values X_s for $s < t - 1$ are irrelevant).

If $Y_t = H_t(X_t, W_t)$ with W_t white noise, then observations depend only on the state at same time and are conditionally independent given the states.

Shortcomings of the (extended) Kalman filter

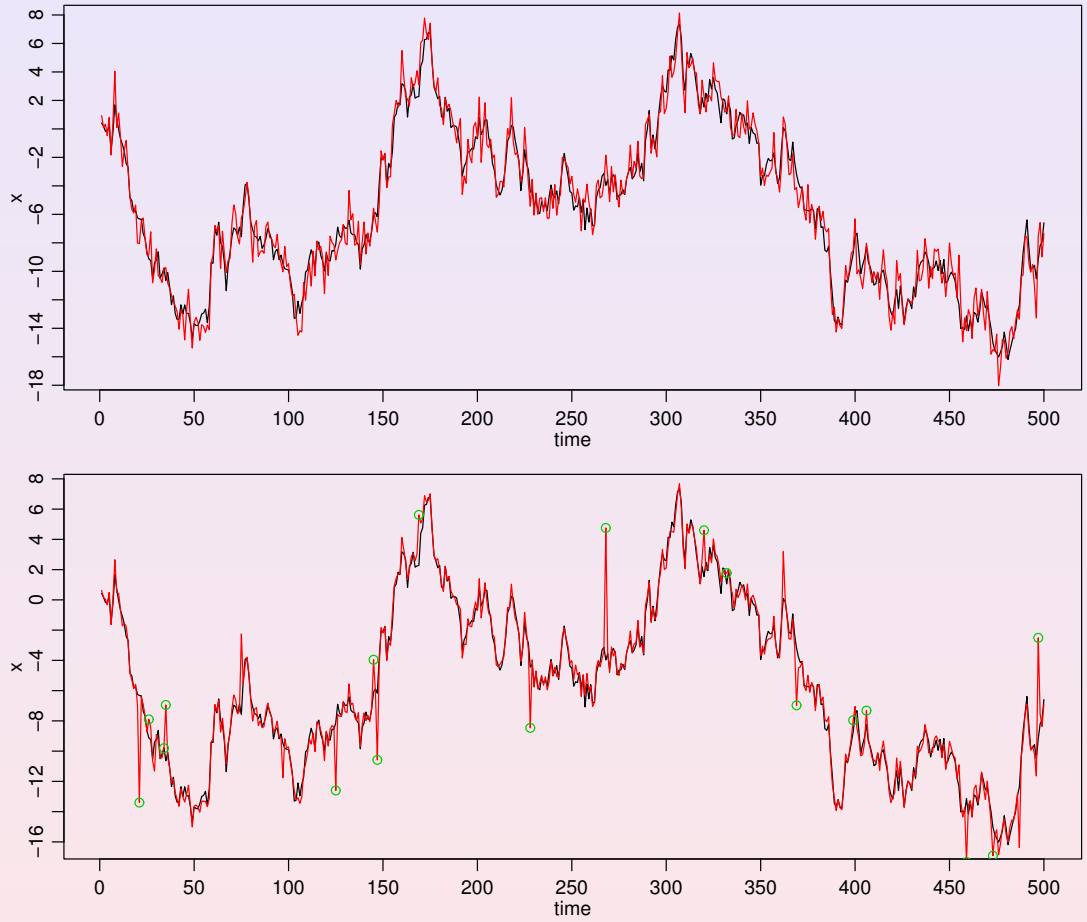
As in the linear Gaussian case, the aim is to compute estimates of the state at time t based on observations up to time $t-1$ or t, together with some uncertainty quantification.

Linear non-Gaussian case: Kalman filter estimates can be used and they are BLUE (best linear unbiased), but nonlinear estimates can be much better.

Nonlinear case: Need to linearize state and observation equations (extended Kalman filter). Simple, but no control on approximation error.

In general, full information about X_t based on observations is obtained from conditional distributions. In particular, best estimate = conditional expectation. In the linear, Gaussian case, conditional distributions are again Gaussian, therefore only conditional mean and covariance needed. Extended Kalman filter says nothing about shape of conditional distribution.

Two observation noises with the same variance



Transition densities for states and observations

Transition densities for the state

$$a_t(x_{t-1}, x_t) = \frac{P(X_t \in x_t + dx_t \mid X_{t-1} = x_{t-1})}{dx_t}$$

can be deduced from F_t and the densities of V_t, but formula is in general complicated. In case of additive noise, $X_t = F_t(X_{t-1}) + V_t$, we have

$$a_t(x_{t-1}, x_t) = f_{V_t}(x_t - F_t(x_{t-1})).$$

Similarly, observation densities

$$b_t(x_t, y_t) = \frac{P(Y_t \in y_t + dy_t \mid X_t = x_t)}{dy_t}$$

can in principle be computed from H_t and the density of W_t.

Filter recursions

Denote the conditional density of X_t given the observations from time 1 to time s by $f_{t|s}(x_t|y_1^s)$. As in the Kalman filter, we can compute $f_{t|t-1}$ from $f_{t-1|t-1}$ by a **prediction step**

$$f_{t|t-1}(x_t|y_1^{t-1}) = \int f_{t-1|t-1}(x_{t-1}|y_1^{t-1}) a_t(x_{t-1}, x_t) dx_{t-1}$$

and $f_{t|t}$ from $f_{t|t-1}$ by an **update step**

$$f_{t|t}(x_t|y_1^t) = \frac{b_t(x_t, y_t) f_{t|t-1}(x_t|y_1^{t-1})}{\int b_t(x_t', y_t) f_{t|t-1}(x_t'|y_1^{t-1}) dx_t'}.$$

The prediction step follows from the law of total probability, and the update step from Bayes formula.

Combined prediction and update step

The denominator in the update step is simply a normalization which turns the numerator into a probability density. Using the symbol $\propto$ for "is proportional" we can write the two steps in the short form

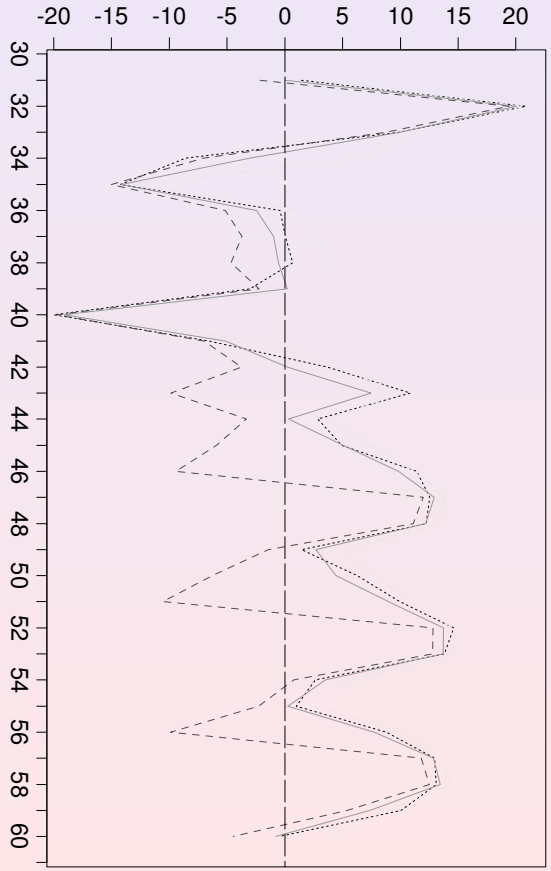
$$f_{t|t}(x_t|y_1^t) \propto b_t(x_t, y_t) \int f_{t-1|t-1}(x_{t-1}|y_1^{t-1}) a_t(x_{t-1}, x_t) dx_{t-1}.$$

Looks simple, but is difficult to use in practice, except in the linear Gaussian case or when the states are discrete.

Numerical integration is feasible for one- or two-dimensional states, but not in general. In the last 20 years, Monte Carlo methods have been developed for this purpose.

Filter distributions for a nonlinear system

Non-linear model of slide 7. The lines are the true state and the 10%-, 50%- and 90%-quantile of the filter distribution approximated by Monte Carlo.



Basics of Monte Carlo

X is a d-dimensional random vector with joint density f, $h : \mathbb{R}^d \to \mathbb{R}$. Monte Carlo methods use the law of large numbers to estimate

$$\mathbb{E}(h(X)) = \int_{\mathbb{R}^d} h(x) f(x) dx \approx \frac{1}{N} \sum_{i=1}^{N} h(x_i)$$

where x_i are independent realizations of X.

$+$: Often easy to use, accuracy does not depend on d.
$-$: Error decreases as $1/\sqrt{N}$ (rather slow).

Implementation on a computer: Based on a stream of uniform pseudo-random numbers ("The generation of random numbers is too important to be left to chance").

Generating random vectors with given f

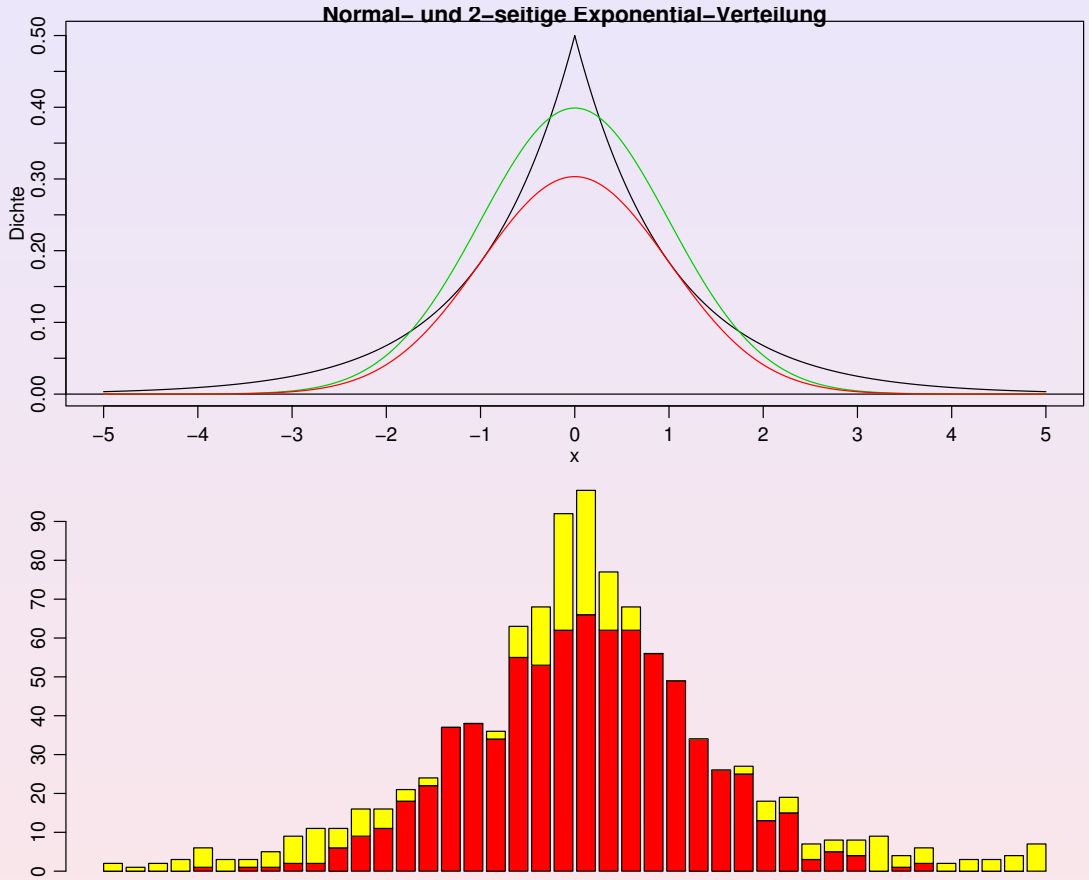
There are many methods to generate random vectors with a given density f (the "target") from uniform random numbers. Some do it directly, but more flexibility is gained by first using a wrong density g (the "proposal") to generate values and by correcting later.

The **accept/reject method** corrects by a "yes or no" decision: A proposed value x is accepted with probability

$$\frac{f(x)/g(x)}{\max_x f(x)/g(x)}.$$

Values that are not accepted are thrown away.

Illustration of the accept/reject method



Importance sampling method

It keeps all proposed N values $x_1, \ldots x_N$, but assigns different weights:

$$\mathbb{E}(h(X)) \approx \sum_{i=1}^{N} w_i h(X_i), \quad w_i = \frac{f(x_i)/g(x_i)}{\sum_j f(x_j)/g(x_j)}.$$

If we want N values with equal weights, we can **"resample"**, that is draw another sample from the proposed values $(x_1, \ldots x_N)$ with probabilities $(w_1, \ldots w_N)$. Some x_k's are then chosen repeatedly, others never.

Both methods require that the proposal g is in some sense close to the target f in order to work well.

The idea of the particle filter

The idea of the particle filter is to construct a sequences of values $(x_{t,1}, x_{t,2}, \ldots x_{t,N})$ with weights $(w_{t,1}, w_{t,2}, \ldots w_{t,N})$ in such a way that

$$\mathbb{E}(h(X_t)|y_1^t) = \int h(x_t) f_{t|t}(x_t|y_1^t) dx_t \approx \sum_{i=1}^{N} h(x_{t,i}) w_{t,i}.$$

The values $x_{t,k}$ are called "particles".

For a computationally efficient algorithm, want to modify $x_{t,i}$ and $w_{t,i}$ slightly when t increases by one.

Sequential importance sampling

We can approximate the integral in the filter recursion with the particles at time $t-1$:

$$f_{t|t-1}(x_t|y_1^{t-1}) \approx f_{t|t-1}^N(x_t|y_1^{t-1}) = \sum_{k=1}^N w_{t-1,k} a_t(x_{t-1,k}, x_t).$$

Hence if $x_{t,k} \sim a_t(x_{t-1,k}, x_t)$, the weighted sample $(x_{t,k}, w_{t-1,k})$ approximates $f_{t|t-1}$. Furthermore

$$f_{t|t}(x_t|y_1^t) \approx f_{t|t}^N(x_t|y_1^t) \propto b_t(x_t, y_t) f_{t|t-1}^N(x_t|y_1^{t-1}).$$

Hence if $w_{t,k} \propto w_{t-1,k} b_t(x_{t,k}, y_t)$, the weighted sample $(x_{t,k}, w_{t,k})$ approximates $f_{t|t}$.

Summary: we alternate between propagation and reweighting steps. This is a sequential implementation of importance sampling with proposal = marginal distribution of states (without observations).

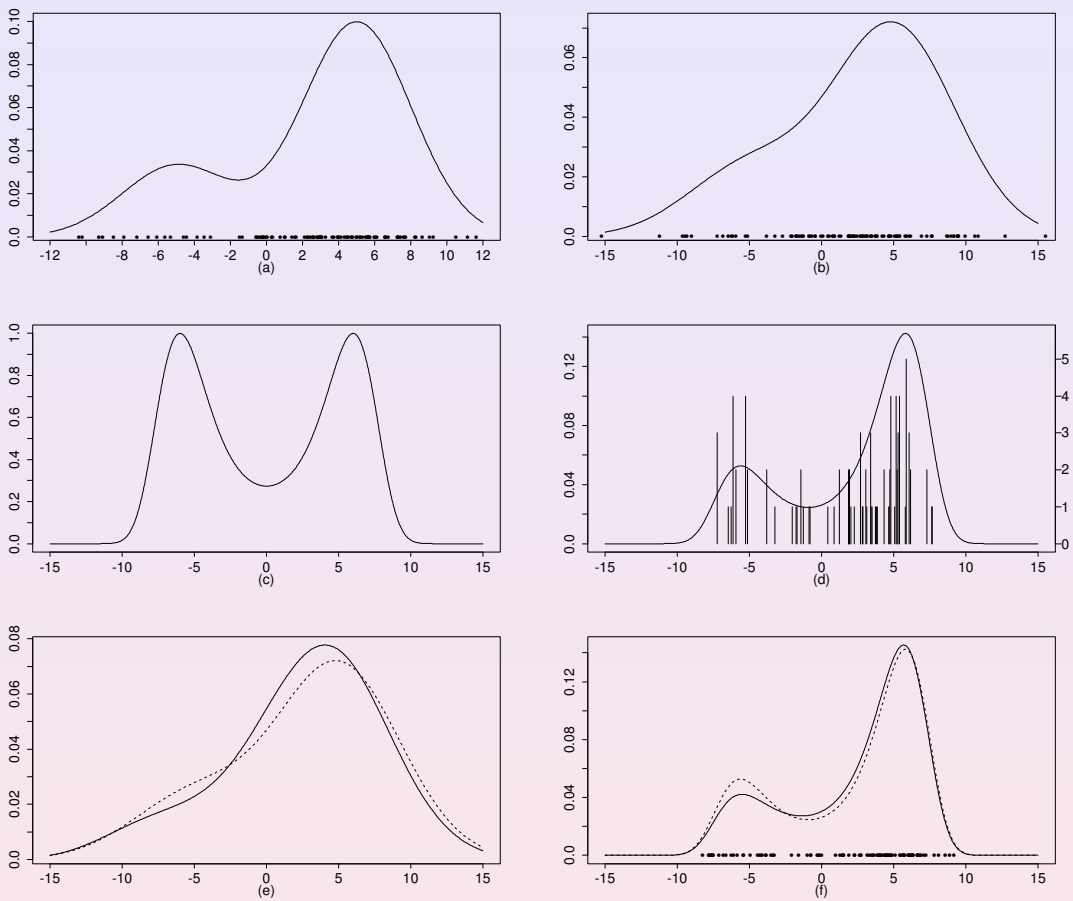
Unbalanced weights and resampling

Sequential importance sampling has the following problems: Most particles are in regions where the filter density is small and have small weights. Propagation of these particles is a waste of effort.

Introduce resampling to make the weights equal. Particles with high weights are chosen many times, those with low weights disappear. Propagate particles at the same position independtly of each other to create diversity.

Summary: In the particle filter, we alternate between the three steps propagation, reweighting and resampling

Illustration of particle filtering



(a): $f_{t-1|t-1}$ and $(x_{t-1,k})$, (b): $f_{t|t-1}$ and $(x_{t-1,k})$,
(c): $b_t(x_t, y_t)$, (d): $f_{t|t}$ and an importance sample of (b),
(e): $f^N_{t|t-1}$, (f): $f^N_{t|t}$ and an exact sample from it.

Tricks for improvement

Resampling introduces additional Monte Carlo error. Can minimize this by using so-called **balanced sampling**: Choose particle k N_k times where $|N_k - Nw_k| < 1$ and $\mathbb{E}(N_k) = Nw_k$.

In order to make weights in the reweighting step more similar, one can propagate not according to $a_t(x_{t-1,k}, x_t)$ but according to a different **proposal depending on the new observation** y_t, $g_t(x_{t-1,k}, x_t)$. This is corrected with importance weights

$$\frac{w_{t-1,K} a_t(x_{t-1,k}, x_{t,k})}{g_t(x_{t-1,k}, x_{t,k}) b_t(x_{t,k}, y_t)}.$$

Convergence and sensitivity to initial conditions

At time t, we sample from the approximate filter density $f^N_{t|t}$.
There is a random sampling error at time t plus a systematic error which is the consequence of errors at previous time steps.
A natural question is: Do these errors remain bounded or do they increase over time ?
This question is equivalent to the following: Do differences in initializing the filter recursion wash out, or do they continue for ever ?
It turns out that this is a really hard question: A positive answer needs strong conditions on a_t (but virtually no condition for b_t).
On the other hand, there are no examples where the negative answer is known to be true.

Extensions and challenges

- Smoothing instead of filtering: Computing conditional distributions of X_t (or of $(X_0, X_1, \ldots, X_T)$ given (y_1^T).
- Filtering if the state transition and/or observation densities have unknown parameters.
- Filtering in high dimensions with a small number of particles (Ensemble Kalman filter = particle filter with a different update method).

This figure shows the smoothing distribution for the non-linear model of slide 7.