

State space models, filtering and environmental applications

Hans R. Künsch
Seminar für Statistik, D-Math
ETH Zürich

Doctoral Program in Statistics and Applied Probability
Les Diablerets
February 9-12, 2014

Thanks to Markus Hürzeler, Michael Amrein, Marco Frei, Fabio Sigrist, Sylvain Robert and Carlo Albert.

Table of contents

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters

Section 1

Introductory Examples

Postprocessing of numerical weather predictions

Data assimilation for weather prediction

Stochastic reaction networks

Rare event estimation

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters

Numerical precipitation predictions

Numerical predictions of 3-h rainfall in northern Switzerland
(50×100 grid with spacing of 2.2 km), separated by 12 h.

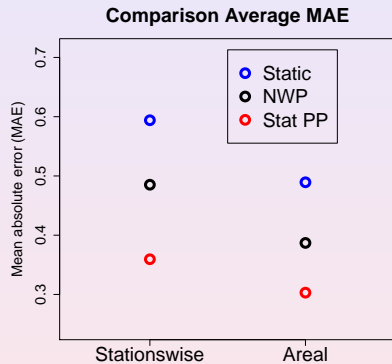
Statistical postprocessing

Numerical weather predictions (MWP) are deterministic with high spatial and temporal resolution. Statistical postprocessing is needed to

- ▶ Correct biases of deterministic forecasts,
- ▶ Quantify uncertainty of deterministic forecasts,
- ▶ Account correctly for spatial and temporal dependence,
- ▶ Obtain predictive distributions which are calibrated and sharp.

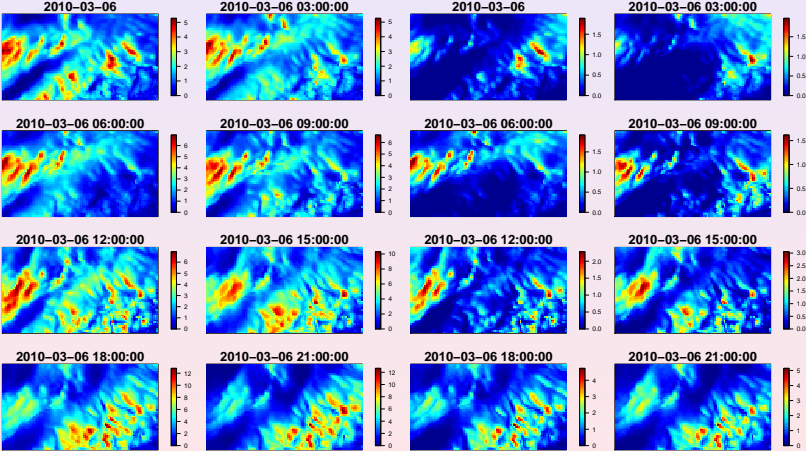
Sigrist et al. (2014) use NWP forecasts as explanatory variables in a statistical space-time model of precipitation. The Kalman filter is an essential tool for fitting this model and for making predictions.

Mean absolute error of 3 forecasts



Effect of postprocessing

NWP forecast (left) and median of predictive distribution (right) for one day in the validation period.



Weather prediction and data assimilation

As noted in the previous example, weather prediction is based on deterministic physical models, formulated as systems of differential equations. Since these equations are sensitive to initial conditions, predictions become unreliable after a few days. One needs to readjust the initial conditions for the next integration cycle frequently, using observations about the state of the atmosphere. In contrast to the previous example, observations are used not only for postprocessing forecasts, but as an input to the next forecast cycle.

Methods which use observations iteratively to estimate the state of a system are called **data assimilation**. In engineering, the same problem is called **filtering**.

Lorenz models

Because any real data assimilation example from atmospheric physics or oceanography is extremely high-dimensional and complex, often simple toy models are used as testbeds. The two most famous are due to Lorenz in 1963 and 1996. The Lorenz 63 model is

$$\frac{d}{dt} \begin{pmatrix} X_t^1 \\ X_t^2 \\ X_t^3 \end{pmatrix} = \begin{pmatrix} 10(X_t^2 - X_t^1) \\ X_t^1(28 - X_t^3) - X_t^2 \\ X_t^1 X_t^2 - \frac{8}{3} X_t^3 \end{pmatrix}.$$

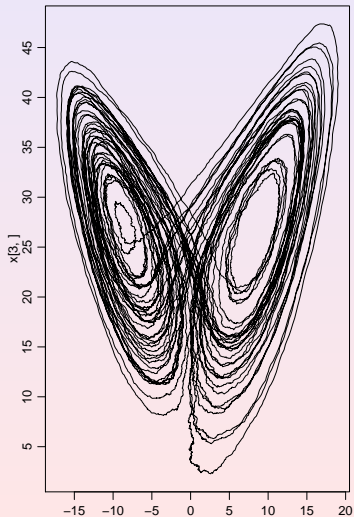
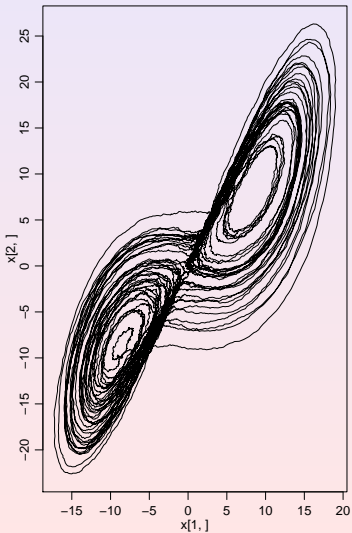
The Lorenz 96 model is

$$\frac{dX_t^k}{dt} = (X_t^{k+1} - X_t^{k-2})X_t^{k-1} - X_t^k + 8, \quad k = 1, \dots, 40, \quad X^k \equiv X^{k+40}.$$

One can also consider stochastic differential equations by adding white noise on the right-hand side to take model deficiencies into account.

The Lorenz 63 model

This model has a famous attractor of fractional dimension. Switches between the two parts seem to occur at random times.

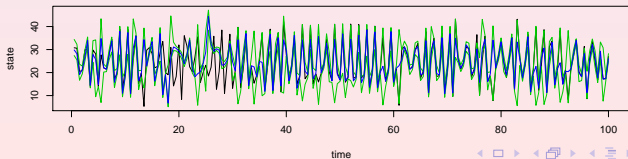
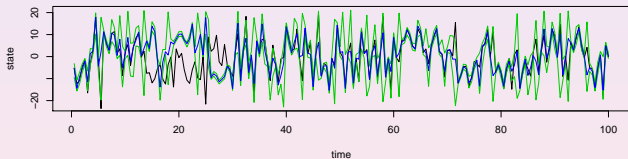
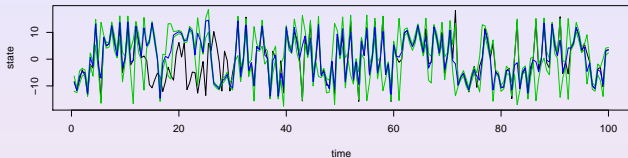


The Lorenz 96 model

This model mimicks large scale motions of a one-dimensional atmosphere.

Particle filter for the Lorenz 63 model

Black: X_t , Blue: Filter median, Green: 10% and 90% quantiles.



Stochastic kinetic reaction networks

Systems biology studies reaction networks with r reactions and p species of molecules. The number of molecules at time t is $x_t = (x_t^1, \dots, x_t^p)$. If reaction i occurs at time t , then x_t changes as follows:

$$x_t^j = x_{t-}^j - v_{ij} + u_{ij} \quad (j = 1, \dots, p).$$

This means that reaction i consumes v_{ij} molecules of type j and produces u_{ij} molecules of type j . Reaction i occurs at the rate

$$\mu_i(t) = \theta_i \prod_{j, v_{ij} \geq 1} \binom{x_t^j}{v_{ij}}$$

provided all $x_t^j \geq v_{ij}$. I.e. the rate of reaction i is proportional to the number of ways the required molecules can be selected.

Stochastic kinetic reaction networks, ctd.

(x_t) is a Markov process in continuous time:

- ▶ At time t , the time until the next reaction is exponential with mean $1/(\mu_1(t) + \dots + \mu_r(t))$.
- ▶ If a reaction occurs, it is type i with probability $\mu_i(t)/(\mu_1(t) + \dots + \mu_r(t))$.
- ▶ The lack of memory of the exponential distribution means that the future depends only on the current state.

The goal is to estimate the rate constants θ_i and the trajectory of the process from noisy measurements of some components $x_{t_i}^j$ at discrete times t_i .

Because of the incomplete and noisy observations, this is an example of a state space model.

A simulated example (Amrein and H.K., 2011)

A system with 3 species and 5 reactions. Two species are observed, the path of the third one is estimated.

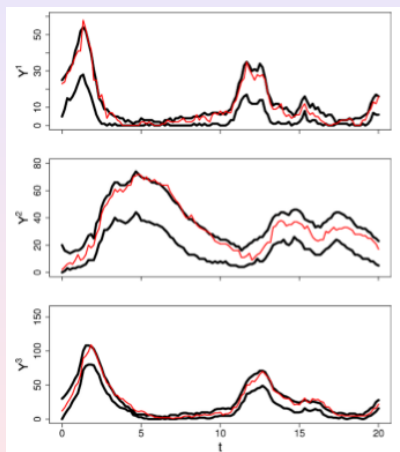


Fig. 5 Pointwise 95% credible bands (indicated by the *thick lines*) of the totally latent components for the Oregonator model in the scenarios S1a (*top*), S2a (*middle*) and S3a (*bottom*), respectively. The true values are shown as *thin line*

Rare event estimation in a queueing network

From Amrein & H.K., ACM Transactions, 21, 2011.

- ▶ Goal: Estimate probabilities of rare events in Markov models, e.g. overflow in 2 queues in series.
- ▶ Arrival at queue 1 according to a Poisson process with rate 1. Service times independent exponential with means ρ_i .
- ▶ State $X_t = (X_t^1, X_t^2)$, where X_t^i = number of customers in queue i at time t .
- ▶ Let $A = \{0, 0\}$, $B = \{(x^1, x^2); x^2 \geq L\}$ and τ_A, τ_B first times of (re)entering A, B . Estimate $\gamma = \mathbb{P}_{(0,0)}(\tau_B < \tau_A)$.
- ▶ If $\rho_1 < 1, \rho_2 < 1, L \gg 1$, γ is small, and direct Monte Carlo fails.

Importance splitting

Alternatives to direct Monte Carlo are importance sampling and importance splitting.

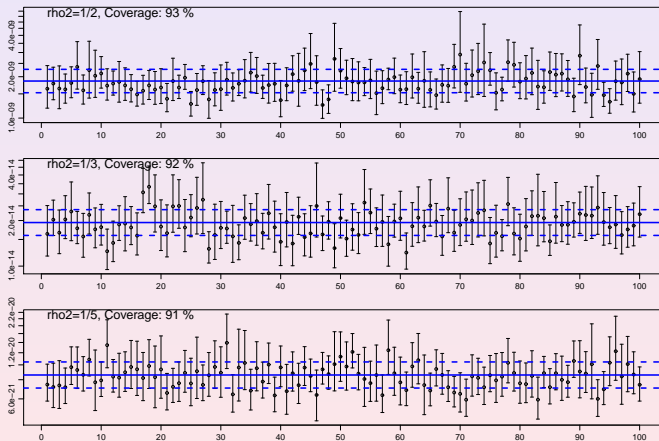
Importance splitting recursively simulates from

$\mathcal{L}(X_{\tau_{B_k}} \mid \tau_{B_k} < \tau_A)$ and estimates $\mathbb{P}(\tau_{B_k} < \tau_A \mid \tau_{B_{k-1}} < \tau_A)$
where $A^c \supset B_1 \supset \dots \supset B_n = B$.

It is a generalization of particle filtering where one wants to approximate conditional distributions of a latent Markov process given an increasing number of instantaneous noisy observations. More generally, one studies recursive simulation from a sequence of different target distributions, so-called sequential Monte Carlo.

Illustration of results

100 replicates of an interval estimate of γ (on log-scale) for $\rho_1 = \frac{1}{2}$ and $\rho_2 = \frac{1}{2}, \frac{1}{3}, \frac{1}{5}$.



Section 2

Introductory Examples

Basics of state space models and filtering

- The state space model

- Filtering and smoothing distributions

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters

State space models

Let X_t be the **state vector** of a system at time t . It is typically not fully observed, but has a simple **Markovian** dynamics. Available are partial **observations** Y_i of X at certain time points t_i , corrupted by independent noise variables.

To simplify notation, assume $t_i = i$ and (X_t) is time-homogeneous with

$$X_t \mid X_{t-1} = x_{t-1} \sim K(dx_t \mid x_{t-1}).$$

(X_t) can be in continuous time and deterministic, i.e. given by a differential equation.

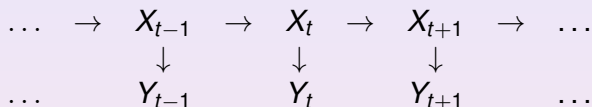
Observations Y_1, Y_2, \dots are **conditionally independent** with

$$Y_t \mid (X_s) \sim g(y \mid x_t) dy.$$

Existence of a density is crucial, but it could be w.r. to a different reference measure.

Graphical representation of state space models

The dependence between the variables of a state space model can be represented as follows



It implies various conditional independencies which we will use later.

An alternative representation of a general state space model is

$$X_t = F(X_{t-1}, U_t), \quad Y_t = H(X_t, V_t)$$

where (U_t) and (V_t) are two independent white noises and F and H are arbitrary functions.

Notation

$X_{s:t}$ is shorthand for $(X_s, X_{s+1}, \dots, X_t)$, similarly $y_{s:t}$.

Define conditional distribution of states in time interval $[s, t]$ given observations up to time r :

$$\pi_{s:t|r}(dx_{s:t} \mid y_{1:r}) := \mathcal{L}(X_{s:t} \mid y_{1:r}).$$

The y 's are fixed and thus are often omitted in $\pi_{s:t|r}$. To simplify, use $\pi_{t|r}$ for the marginal $\pi_{t:t|r}$. Call $\pi_{t|t-1}$ the **prediction** and $\pi_{t|t}$ the **filter** distribution. Distributions with $s < r$ are called **smoothing** distributions.

By abuse of notation, we use P ($P(\cdot \mid \cdot)$) for other (conditional) distributions that will play a role, and $p(\cdot)$ ($p(\cdot \mid \cdot)$) for (conditional) densities. The arguments will indicate which variables are involved.

Gibbs sampler for state space models

If $K(dx_t | x_{t-1}) = k(x_t | x_{t-1})dx_t$, full conditionals are

$$\begin{aligned} p(x_s | y_{1:t}, x_{0:s-1}, x_{s+1:t}) &= p(x_s | y_s, x_{s-1}, x_{s+1}) \\ &\propto g(y_s | x_s)k(x_s | x_{s-1})k(x_{s+1} | x_s). \end{aligned}$$

If the model contains unknown parameters θ :

$$p(\theta | x_{0:t}, y_{1:t}) \propto p(\theta) \prod_{s=1}^t k_\theta(x_s | x_{s-1})g_\theta(y_s | x_s).$$

Hence the Gibbs sampler can be used. But

- ▶ Convergence is in most cases slow: Full conditionals are too tight.
- ▶ Better to update $x_{0:t}$ jointly (or at least in big blocks).
- ▶ If observations become available sequentially, want to update the samples recursively.

Filtering recursions

Prediction and filter distributions can be computed **recursively**:

- ▶ **Propagation** $\pi_{t-1|t-1} \longrightarrow \pi_{t|t-1}$ via conditioning on X_{t-1}

$$\pi_{t|t-1}(dx_t) = \int K(dx_t | x_{t-1})\pi_{t-1|t-1}(dx_{t-1}).$$

- ▶ **Update** $\pi_{t|t-1} \longrightarrow \pi_{t|t}$ via Bayes' formula:

$$\pi_{t|t}(dx_t) \propto \pi_{t|t-1}(dx_t) \times g(y_t | x_t).$$

We start the recursion with $\pi_{0|0}$ = the initial distribution of X_0 .

The denominator in Bayes' formula is

$$\int g(y_t | x_t)\pi_{t|t-1}(dx_t) = p(y_t | y_{1:t-1})$$

Thus the joint density of $Y_{1:t}$ is a byproduct of the filter.

Smoothing

Analogous recursions as for the filter hold for $\pi_{0:t|t}$ with an **extension** instead of a propagation step

$$\begin{aligned}\pi_{0:t|t-1}(dx_{0:t}) &= K(dx_t | x_{t-1})\pi_{0:t-1|t-1}(dx_{0:t-1}) \\ \pi_{0:t|t}(dx_{0:t}) &\propto \pi_{0:t|t-1}(dx_{0:t}) \times g(y_t | x_t).\end{aligned}$$

Alternatively, we can use that given $y_{1:t}$ the state process is still a Markov chain, by rules for conditional independence. If $K(dx_t | x_{t-1}) = k(x_t | x_{t-1})dx_t$, then the **backward transition of the conditional chain** is a modification of the filter distribution:

$$\begin{aligned}P(dx_s | x_{s+1}, y_{1:t}) &= P(dx_s | x_{s+1}, y_{1:s}) \\ &\propto k(x_{s+1} | x_s)\pi_{s|s}(dx_s).\end{aligned}$$

There are also expressions for the forward transition distributions.

Implementations

- ▶ Although the recursions look simple, doing the integrations is in most cases difficult.
- ▶ Closed form solutions exist if X_t takes values in a finite set (Baum-Welch) or in the Gaussian linear case (Kalman) that we will discuss in the next section. These are all practically relevant cases.
- ▶ Analytical approximations exist (Extended Kalman filter, unscented Kalman filter), but they are limited in scope.
- ▶ Numerical approximations are difficult because X_t is often high-dimensional and $\pi_{t|t}$ lives in different parts of the state space for different t 's.
- ▶ Particle and Ensemble Kalman filters are recursive Monte Carlo approximations.

Section 3

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

- Gaussian linear state space models

- Kalman filter and smoother

- Fitting of stochastic advection-diffusion models

Particle filters

Ensemble Kalman filters

Extensions of particle filters

Gaussian linear state space models

A Gaussian linear state space model has the form

$$X_t = FX_{t-1} + U_t, \quad Y_t = HX_t + V_t$$

where (U_t) and (V_t) are independent Gaussian white noises with mean 0 and covariance Q and R , respectively. In other words, we have a partially observed vector autoregression.

If also X_0 is Gaussian, then **all conditional distributions** $\pi_{s:t|r}$ **are again Gaussian**. The Kalman filter and smoother computes required conditional means and covariances recursively. It can be derived from the general recursions above, or directly from basic properties of orthogonal projections.

Kalman filter

Denote mean and covariance of $\pi_{s|t}$ by $\mu_{s|t}$ and $P_{s|t}$. Then

$$\mu_{t|t-1} = F\mu_{t-1|t-1}, \quad P_{t|t-1} = FP_{t-1|t-1}F' + Q$$

and

$$\begin{aligned}\mu_{t|t} &= \mu_{t|t-1} + K_t(y_t - H\mu_{t|t-1}), \\ P_{t|t} &= (I - K_tH)P_{t|t-1} = (I - K_tH)P_{t|t}(I - K_tH)' + K_tRK_t'\end{aligned}$$

where

$$\begin{aligned}K_t &= K(H, P_{t|t-1}, R) = P_{t|t-1}H'(HP_{t|t-1}H' + R)^{-1} \\ &= \text{Cov}(X_t - \mu_{t|t-1}, Y_t - H\mu_{t|t-1}) \text{Cov}(Y_t - H\mu_{t|t-1})^{-1}.\end{aligned}$$

is the so-called **Kalman gain**.

Kalman smoother

Because $P(dx_s | x_{s+1}, y_{1:t}) \propto k(x_{s+1} | x_s)\pi_{s|s}(dx_s)$, backward transitions are obtained as in the filter update

$$\begin{aligned}\mathbb{E}(X_s | X_{s+1}, y_{1:t}) &= \mathbb{E}(X_s | X_{s+1}, y_{1:s}) \\ &= \mu_{s|s} + K(F, P_{s|s}, Q)^{-1}(X_{s+1} - F\mu_{s|s}),\end{aligned}$$

$$\text{Cov}(X_s | X_{s+1}, y_{1:t}) = (I - K(F, P_{s|s}, Q))P_{s|s}.$$

Thus we can simulate from $\pi_{0:t|t}$ using **forward-filtering**
backward-simulation.

If F, Q, R contain unknown parameters θ , use Metropolis with simultaneous updates of θ and $x_{0:t}$: Propose

$$(x_{0:t}^*, \theta^*) | (x_{0:t}, \theta) \sim q(\theta^* | \theta)p(x_{0:t}^* | y_{1:t}, \theta^*)$$

and accept it with probability

$$\min\left(1, \frac{p(\theta^*)p(y_{1:t} | \theta^*)}{p(\theta)p(y_{1:t} | \theta)}\right).$$

Computational complexity

If $X_t \in \mathbb{R}^q$ and $Y_t \in \mathbb{R}^d$ and we consider T time steps:

- ▶ Forward filtering and likelihood evaluation has complexity $O(T(q^2 + d^3))$.
- ▶ Backward simulation has additional complexity $O(Tq^3)$.
- ▶ If q and/or d are large, efficient computation becomes crucial.
- ▶ Savings are possible if we approximate $P_{t-1|t}$ and $P_{t|t}$ by sparse matrices (\rightarrow sparse Cholesky factorizations).
- ▶ In the introductory example of precipitation forecast, we obtain a state space model where all matrices are block diagonal.

Back to the first example

I now show how the Kalman filter and smoother is used for the statistical postprocessing of numerical weather predictions (NWP).

Rainfall $y(t, \mathbf{s})$ is skewed and has an atom at zero, and thus we cannot use a Gaussian distribution. We assume it depends on a latent “potential rainfall” $w(t, \mathbf{s})$ which has a Gaussian distribution through

$$y(t, \mathbf{s}) = w(t, \mathbf{s})^\lambda \mathbf{1}_{\{w(t, \mathbf{s}) > 0\}}.$$

I will first discuss a class of Gaussian space-time models assuming Gaussian observations. At the end I show how we handle the non-linear dependence of $y(t, \mathbf{s})$ on $w(t, \mathbf{s})$.

Modeling potential rainfall

We assume that the potential rainfall $w(t, \mathbf{s})$ is a Gaussian space-time random field

$$w(t, \mathbf{s}) = \beta_1 y_F(t, \mathbf{s})^{1/\tilde{\lambda}} + \beta_2 \mathbf{1}_{\{y_F(t, \mathbf{s})=0\}} + X(t, \mathbf{s}) + \nu(t, \mathbf{s}),$$

- ▶ $y_F(t, \mathbf{s})$ NWP forecast made at 0:00 UTC of the same day,
- ▶ $X(t, \mathbf{s})$ structured random effect (mean zero),
- ▶ $\nu(t, \mathbf{s})$ white noise in space and time, variance σ_ν^2
- ▶ Unknown parameters: $\beta_1, \beta_2, \lambda > 0, \sigma_\nu^2$ and parameters for X (to be discussed later). $\tilde{\lambda}$ is estimated from a preliminary analysis.

We estimate w and the parameters based on precipitation data from 32 stations that the NWP forecast does not use directly.

Gaussian space-time fields

- ▶ $X(t, \mathbf{s}) =$ Gaussian process on $[0, \infty) \times \mathbb{R}^d$ with mean zero and covariance function C_θ . How to choose C_θ ?
- ▶ General considerations:
 - ▶ Take different nature of time and space into account
 - ▶ Avoid separable models
 - ▶ Find compromise between simplicity and generality
 - ▶ Use parameters which have a clear interpretation
- ▶ Computational tractability:
 - ▶ Evaluation of likelihood has in general the complexity $O((TN)^3)$, T and N number of points in time and space.
- ▶ Adding noise to simple physics based models satisfies the requirements above. Other people have proposed similar approaches. Our contribution: Solution of computational issues.

Stochastic advection-diffusion model

- ▶ Gaussian process defined through **stochastic partial differential equation** (SPDE)

$$\frac{\partial}{\partial t} X(t, \mathbf{s}) = -\boldsymbol{\mu} \cdot \nabla X(t, \mathbf{s}) + \nabla \cdot \boldsymbol{\Sigma} \nabla X(t, \mathbf{s}) - \kappa X(t, \mathbf{s}) + \epsilon(t, \mathbf{s})$$

- ▶ $-\boldsymbol{\mu} \cdot \nabla X(t, \mathbf{s})$ transport, $\boldsymbol{\mu}$ drift vector ("advection")
 - ▶ $\nabla \cdot \boldsymbol{\Sigma} \nabla X(t, \mathbf{s})$ anisotropic diffusion
 - ▶ $-\kappa X(t, \mathbf{s})$ damping
 - ▶ $\epsilon(t, \mathbf{s})$ temporally white noise and spatially colored forcing or source-sink ("convection")
- ▶ Continuous time and space model

Illustration of deterministic PDE

Illustration of SPDE

Elementary solutions of the SPDE

Assume that for a fixed wavenumber $\mathbf{k} \in \mathbb{R}^2$

$$X(0, \mathbf{s}) = \alpha(0) \exp(i\mathbf{k}'\mathbf{s}), \quad \epsilon(t, \mathbf{s}) = \dot{W}(t) \exp(i\mathbf{k}'\mathbf{s})$$

where \dot{W} is white noise (i.e. W is Brownian motion).

Then the solution is

$$X(t, \mathbf{s}) = \left(e^{-h(\mathbf{k})t} \alpha(0) + \int_0^t e^{-h(\mathbf{k})(t-u)} \dot{W}(u) du \right) \exp(i\mathbf{k}'\mathbf{s})$$

where $h(\mathbf{k}) = i\boldsymbol{\mu}'\mathbf{k} + \mathbf{k}'\boldsymbol{\Sigma}\mathbf{k} + \kappa$. Because $\text{Re}(h) > 0$, the solution forgets the initial amplitude $\alpha(0)$, and we have for large t a Gaussian amplitude which is stationary in time.

By linearity of the SPDE, sums of such solutions are again solutions.

Stationary solutions of the SPDE

If $X(0, \mathbf{s})$ and $\epsilon(t, \mathbf{s})$ are stationary in space, by the Cramér representation

$$X(0, \mathbf{s}) = \int \exp(i\mathbf{k}'\mathbf{s})\alpha(0, d\mathbf{k}), \quad \epsilon(t, \mathbf{s}) = \int \exp(i\mathbf{k}'\mathbf{s})\dot{W}(t, d\mathbf{k}),$$

with $\mathbb{E} \left(\dot{W}(t, d\mathbf{k}) \overline{\dot{W}(u, d\mathbf{j})} \right) = \delta_{t,u} \delta_{\mathbf{k},\mathbf{j}} f_{\epsilon}(\mathbf{k}) d\mathbf{k}$, and similarly for $\alpha(0, d\mathbf{k})$.

Then the solution has the representation

$$X(t, \mathbf{s}) = \int \exp(i\mathbf{k}'\mathbf{s})\alpha(t, d\mathbf{k}),$$
$$\alpha(t, d\mathbf{k}) = e^{-h(\mathbf{k})t}\alpha(0, d\mathbf{k}) + \int_0^t e^{-h(\mathbf{k})(t-u)}\dot{W}(u, d\mathbf{k})du.$$

Amplitudes develop independently for different wavenumbers.

Discretizing time and space

Assume we are interested in X at a finite set of locations and times. Truncating wavenumbers to a finite set, we have

$$\mathbf{X}(t_j) = \mathbf{\Phi}\alpha(t_j), \quad \alpha(t_j) = \mathbf{G}\alpha(t_{j-1}) + \mathcal{N}(0, \mathbf{Q}).$$

Here

- ▶ $\mathbf{\Phi}$ has elements $\exp(i\mathbf{k}'_{\ell}\mathbf{s}_j)$.
- ▶ \mathbf{G} and \mathbf{Q} are diagonal.
- ▶ \mathbf{G} depends on the time step $t_j - t_{j-1}$ and on the parameters μ , Σ and κ of the SPDE.
- ▶ \mathbf{Q} depends on the time step $t_j - t_{j-1}$ and on the assumed spatial spectral density f_{ϵ} of $\epsilon(t, \cdot)$. We use the Matérn density with smoothness parameter 1.
- ▶ The time discretization of $\alpha(t)$ is exact, the only approximation is the truncation of wavenumbers.

Fitting with linear Gaussian observations

Adding observation noise to \mathbf{X} , we obtain a Gaussian linear state space model with states $\alpha(t_j)$ and observations \mathbf{w}_j :

$$\alpha(t_j) = \mathbf{G}\alpha(t_{j-1}) + \mathcal{N}(0, \mathbf{Q}), \quad \mathbf{w}_j = \mathbf{\Phi}\alpha(t_j) + \mathcal{N}(0, \sigma_\nu^2 \mathbf{I}).$$

If we use sin and cos instead of the complex exponential, \mathbf{G} becomes 2×2 block diagonal.

We can estimate θ and α using Metropolis with simultaneous updates as long as the set of wavenumbers and the set of locations with observations are not too big. Estimates of $X(t_j, \mathbf{s})$ follow from estimates of α .

Fitting with linear Gaussian observations, ctd.

If \mathbf{w} is on an $n \times n$ grid with spacing Δ , we can use wavenumbers on the grid with spacing $2\pi/(\Delta n)$. The Kalman filter and smoother decouple into 2-dimensional subproblems because Φ is then orthogonal:

$$\Phi' \mathbf{w}_i = \alpha(t_i) + \mathcal{N}(0, \sigma_v^2 \mathbf{I}),$$

and FFT makes evaluation of $\Phi' \mathbf{w}_i$ fast.

Nonlinear and non-Gaussian observations

In the precipitation example, iterate between updates of (\mathbf{w}, λ) given $\alpha, \mathbf{y}, \sigma_{\nu}^2$ and updates of (θ, α) given \mathbf{w} . For the former, propose first a new λ . If accepted, modify \mathbf{w} deterministically at sites where $\mathbf{y} > 0$. At other sites, sample w from a truncated Gaussian distribution.

Similar procedure is possible for binary data in a probit model. Otherwise, need to use iterative methods with Gaussian approximations of $p(y(t, \mathbf{s}) | X(t, \mathbf{s}))$.

Section 4

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

- The basic particle filter

- Balanced sampling

- Auxiliary particle filter

- Likelihood estimation

- Importance splitting as a particle filter

Ensemble Kalman filters

Extensions of particle filters

Particle Filter

Remember the filtering recursions:

- ▶ Propagation: $\pi_{t|t-1}(dx_t) = \int K(dx_t | x_{t-1})\pi_{t-1|t-1}(dx_{t-1})$.
- ▶ Update: $\pi_{t|t}(dx_t) \propto \pi_{t|t-1}(dx_t) \times g(y_t | x_t)$.

Recursive Monte Carlo implementation: If $(x_{t-1|t-1}^j; j = 1, \dots, N)$ is an (approximate) sample from $\pi_{t-1|t-1}$, construct an (approximate) sample from $\pi_{t|t}$ by

- ▶ **Propagation:** Generate $(x_{t|t-1}^j)$ by simulating the Markov process with initial conditions $X_{t-1} = x_{t-1|t-1}^j$, independently for different j 's.
- ▶ **Update:** Obtain $(x_{t|t}^j)$ by resampling from $(x_{t|t-1}^j)$ with probabilities proportional to $g(y_t | x_{t|t-1}^j)$.

The $x_{t|t-1}^j$'s and $x_{t|t}^j$'s are called **particles** which move in space and have offsprings. They interact via the number of offsprings.

The update step

- ▶ Update can be split into a weighting step

$$\pi_{t|t}(dx_t) \approx \sum_{j=1}^N w_t^j \Delta_{x_{t|t-1}^j}(dx_t), \quad w_t^j \propto g(y_t | x_{t|t-1}^j).$$

(importance sampling), followed by resampling.

- ▶ Without resampling, weights would be updated by multiplication and become very unbalanced.
- ▶ Resampling introduces additional noise but it avoids waisting simulation efforts in the next propagation step with unlikely starting values.
- ▶ Hence it is better to use the weighted approximation of $\pi_{t|t}$ and to resample at the beginning of the propagation step.

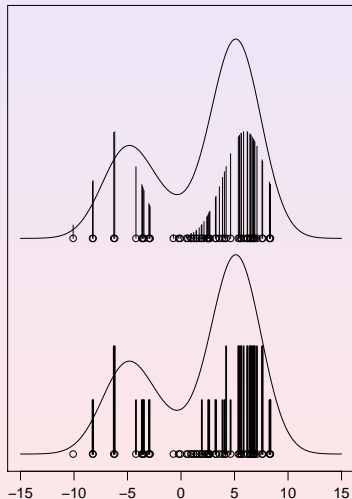
Ties due to resampling

- ▶ Resampling introduces ties among $x_{t|t}^j$. These are broken in the next propagation step (called “rejuvenation”) if state dynamics is stochastic.
- ▶ If some components of state change deterministically, can break ties by adding noise (compensated by shrinkage to the mean).

A single step of the particle filter

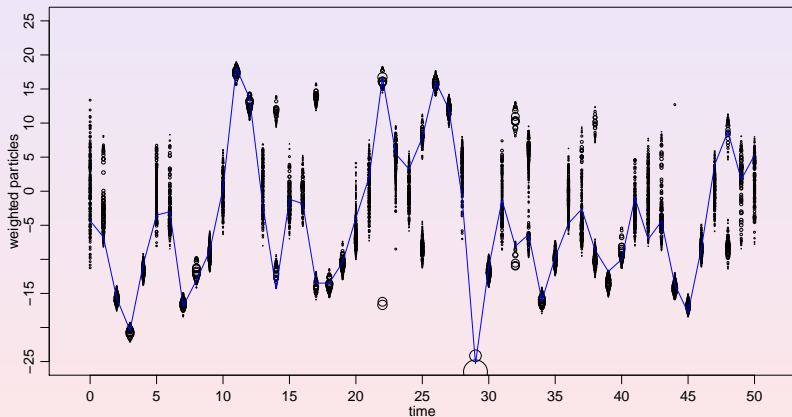
Left: Propagation (only few arrows shown).

Right: Reweighting and resampling



Several steps of the particle filter

Blue: X_t nonlinear AR, $Y_t = X_t^2 + V_t$. Black: Particle filter with area of circles \propto weight. .



Effective sample size (ESS)

If we have a weighted sample (X^j, w^j) of size N constructed by importance sampling, to how many unweighted sample values does this correspond ? Liu et al. gave the answer

$$\text{ESS} = \left(\sum_{j=1}^N (w^j)^2 \right)^{-1} .$$

This is correct in the two extreme cases $w^j = 1/N$ for all j and $w^j = 1$ for one j . The general case follows from an approximation of the variance of $\sum_j w^j \psi(X^j)$.

It has been suggested to resample only if ESS is below a certain threshold. Note however that in a recursive setting, ESS is presumably too optimistic since the propagation step does not fully compensate the loss of precision due to weighting and resampling at time $t - 1$.

Balanced sampling

For resampling, can use any scheme s.th. $x_{t|t-1}^j$ is chosen on average Nw_t^j times. A sampling scheme is called balanced if $x_{t|t-1}^j$ is chosen $\lceil Nw_t^j \rceil$ or $\lceil Nw_t^j \rceil + 1$ times.

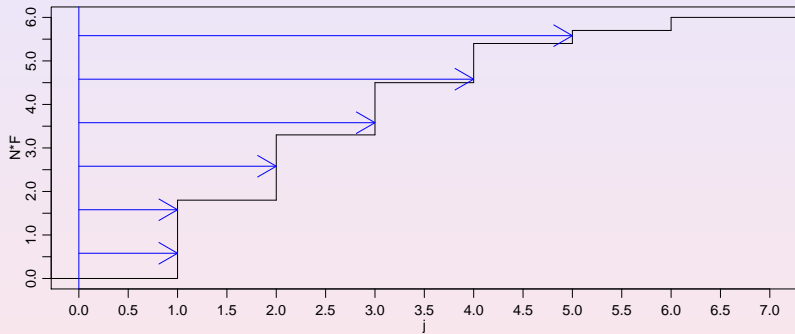
There are many balanced sampling schemes. The simplest such scheme is

$$x_{t|t}^j = x_{t|t-1}^{l_j}, \quad l_j = F_t^{-1}((j - U)/N)$$

where U is uniform(0, 1) and F_t is the cdf of the discrete distribution on $\{1, 2, \dots, N\}$ with weights w_t^j , see figure.

This is hard to analyze theoretically because no limit theory applies. So-called tree sampling is an alternative.

Illustration of balanced sampling



Auxiliary particle filter

Idea: Propagate particles with a transition $Q \neq K$ to bring them closer to y_t and adjust with suitable weights which are more equal

- ▶ Generate $x_t^{*j} \sim q(x_t | x_{t-1}^j|_{t-1}, y_t) dx_t$, independently for different j 's. (Change of notation since the x_t^{*j} are no longer a sample from $\pi_{t|t-1}$).
- ▶ Resample the particles x_t^{*j} with probabilities

$$w_t^j \propto \tilde{w}(x_{t-1}^j|_{t-1}, x_t^{*j}, y_t) = \frac{g(y_t | x_t^{*j})k(x_t^{*j} | x_{t-1}^j|_{t-1})}{q(x_t^{*j} | x_{t-1}^j|_{t-1}, y_t)}.$$

- ▶ More generally, replace ratio k/q by Radon-Nikodym derivative provided $K \ll Q$.

Auxiliary particle filter ctd.

- ▶ Weights w_t^j depend on x_t^{*j} and x_{t-1}^j .
- ▶ $((x_{t-1}^j, x_t^{*j}), w_t^j)$ is a weighted sample from $\pi_{t-1:t|t}$.
- ▶ Variance of weights \tilde{w} is minimal for

$$q(x_t | x_{t-1}, y_t) = p(x_t | x_{t-1}, y_t) = \frac{g(y_t | x_t)k(x_t | x_{t-1})}{p(y_t | x_{t-1})}.$$

In this case, $w_t^j \propto p(y_t | x_{t-1}^j)$, and **resampling before propagation** increases diversity.

- ▶ Also in other cases one can add a resampling step with probabilities $\tau(x_{t-1}^j, y_t)$ before propagation. Then weights after propagation are

$$w_t^j \propto \tilde{w}(x_{t-1}^j, x_t^{*j}, y_t) = \frac{g(y_t | x_t^{*j})k(x_t^{*j} | x_{t-1}^j)}{\tau(x_{t-1}^j, y_t)q(x_t^{*j} | x_{t-1}^j, y_t)}.$$

Auxiliary particle filter, ctd.

Instead of sampling approximately from $\pi_{t-1:t|t}$, we can also look at sampling from the approximate filter density

$$\hat{\pi}_{t|t}(x_t) \propto \frac{1}{N} \sum_{j=1}^N g(y_t | x_t) k(x_t | x_{t-1}^j |_{t-1}).$$

Importance sampling of $x_{t,*}$ alone leads to an $O(N^2)$ algorithm since computing a single weight involves a sum over N terms. Importance sampling of a mixture index l together with $x_{t,*}$ avoids this problem: If we use a proposal

$$\mathbb{P}(l = j) \propto \tau(x_{t-1}^j |_{t-1}, y_t), \quad p(x_{t,*} | l = j) = q(x_{t,*} | x_{t-1}^j |_{t-1}, y_t)$$

then the importance weights are indeed proportional to

$$\frac{g(y_t | x_t^{*j}) k(x_t^{*j} | x_{t-1}^j |_{t-1})}{\tau(x_{t-1}^j |_{t-1}, y_t) q(x_t^{*j} | x_{t-1}^j |_{t-1}, y_t)}.$$

Implementing the auxiliary particle filter

The optimal choices $\tau(x_{t-1|t-1}^j, y_t) = p(y_t | x_{t-1|t-1}^j)$ and $q(x_t | x_{t-1}, y_t) = p(x_t | x_{t-1}, y_t)$ are usually not available explicitly, so need an approximation. Simplest choice by Gaussian approximation

$$\begin{aligned} & \log k(x_t | x_{t-1}) + \log g(y_t | x_t) \\ \approx & c(x_{t-1}, y_t) + b(x_{t-1}, y_t)'x_t + \frac{1}{2}x_t' A(x_{t-1}, y_t)x_t, \end{aligned}$$

obtained by using e.g. a Taylor approximation around the $\arg \max_{x_t}$ of $k(x_t | x_{t-1})g(y_t | x_t)$.

Likelihood estimation

Remember that the the likelihood increment $p(y_t | y_{1:t-1})$ is the denominator in Bayes formula in the update step,

$$\int g(y_t | x_t) \pi_{t|t-1}(x_t) dx_t = \int g(y_t | x_t) \pi_{t-1:t|t-1}(x_t, x_{t-1}) dx_t dx_{t-1}.$$

Hence we have the estimate

$$\hat{p}(y_t | y_{1:t-1}) = \frac{1}{N} \sum_{j=1}^N g(y_t | x_{t|t-1}^j)$$

which we compute when normalizing the weights.

If we use the auxiliary particle filter, we have similarly

$$\hat{p}(y_t | y_{1:t-1}) = \frac{1}{N} \sum_{j=1}^N \tilde{w}(x_{t-1|t-1}^j, x_t^{*j}, y_t).$$

Likelihood estimation is unbiased

In general the particle filter does not produce unbiased estimates of $\int \psi(x_t) \pi_{t|t}(dx_t)$. But we have

Lemma If x_0^j is sampled from the initial distribution of X_0 and if resampling is unbiased, then

$$\mathbb{E}\left(\prod_{t=1}^n \hat{p}(y_t | y_{1:t-1})\right) = \prod_{t=1}^n p(y_t | y_{1:t-1}) = p(y_{1:n}),$$

but in general $\mathbb{E}(\hat{p}(y_t | y_{1:t-1})) \neq p(y_t | y_{1:t-1})$.

This will be used in rare event estimation and in particle MCMC.

Properties of particle filter

- ▶ Under weak conditions, it is consistent (as $N \rightarrow \infty$) and a CLT holds.
- ▶ Under strong conditions, can also show that consistency is uniform in t and variance in CLT is uniformly bounded in t .
- ▶ Respects constraints $\psi(x(t)) \equiv 0$.
- ▶ Weights degenerate quickly as dimensions grow (see Bickel et al. for a theoretical explanation). Particle filter can lose track easily.
- ▶ Auxiliary particle filter can bring substantial improvements, but not applicable if density k does not exist or is not available analytically.

Importance splitting as a particle filter

- ▶ Let (X_t) be a Markov process on some set E , $A, B \subset E$ $A \cap B = \emptyset$. We want to estimate $\gamma = \mathbb{P}_{x_0}(\tau_B < \tau_A)$ where τ_A and τ_B are first times of (re)entering A, B and $x_0 \notin B$.
- ▶ Importance splitting chooses $A^c \supset B_1 \supset \dots \supset B_n = B$ and simulates recursively from $\mathcal{L}(X_{\tau_{B_k}} \mid \tau_{B_k} < \tau_A)$.
- ▶ If we consider the state process $X'_k = X_{\tau_{B_k}}$ and assume that “observations” $Y_k = 1_{[\tau_{B_k} < \tau_A]}$ all have the value 1, this becomes a filtering problem.
- ▶ Moreover

$$\gamma = \prod_{k=1}^n \mathbb{P}(\tau_{B_k} < \tau_A \mid \tau_{B_{k-1}} < \tau_A) = \prod_{k=1}^n \mathbb{P}(Y_k = 1 \mid Y_{1:k-1} \equiv 1).$$

Therefore estimation of the “likelihood” gives the desired estimate of γ , and it is unbiased !

Implementing importance splitting

- ▶ A particle with $\tau_{B_k} > \tau_A$ gets weight 0 and is thus killed. Resampling inflates the number R_k of surviving particles at step k back to N . Impossible if $R_k = 0$, so then $\hat{\gamma} := 0$.
- ▶ Observations are not made available sequentially, so simulation effort can vary in each step. It is better to control precision by propagating particles until a fixed value ≥ 2 for R_k is achieved. Then N_{k-1} becomes random.
- ▶ Unbiasedness of $\hat{\gamma}$ continues to hold if we use the UMVU estimator for the negative binomial

$$\hat{\mathbb{P}}(\tau_{B_k} < \tau_A \mid \tau_{B_{k-1}} < \tau_A) = \frac{R_k - 1}{N_{k-1} - 1}.$$

- ▶ In our paper we also address the choices of n , B_k , R_k by a two-step procedure and construct confidence intervals for γ by running particle filters in parallel.

Section 5

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

- Basic idea

- Square root and perturbed observation versions

- Fine tuning

- Bridging the two methods

Extensions of particle filters

Basic idea of Ensemble Kalman filter (Evensen, 1994)

Assume arbitrary state dynamics, but Gaussian-linear observations

$$Y_t = HX_t + \mathcal{N}(0, R).$$

Prediction step is the same as in the particle filter, but **update** assumes (wrongly) that $\pi_{t|t-1} = \mathcal{N}(\mu_{t|t-1}, P_{t|t-1})$.

Then – as in the Kalman filter – $\pi_{t|t} = \mathcal{N}(\mu_{t|t}, P_{t|t})$ where

$$\begin{aligned}\mu_{t|t} &= \mu_{t|t-1} + K(H, P_{t|t-1}, R)(y_t - H\mu_{t|t-1}) \\ P_{t|t} &= (I - K(H, P_{t|t-1}, R)H)P_{t|t-1}.\end{aligned}$$

Ensemble Kalman filter estimates $\mu_{t|t-1}$ and $P_{t|t-1}$ and converts $(x_{t|t-1}^j)$ into a sample with the correct first and second moment.

Square root and stochastic version

The Ensemble Kalman filter comes in two versions:

- ▶ **Stochastic:** Perturbed observation EnKF

$$x_{t|t}^j = x_{t|t-1}^j + \hat{K}_t(y_t + \varepsilon_t^j - Hx_{t|t-1}^j), \quad \varepsilon_t^j \sim \mathcal{N}(0, R)$$

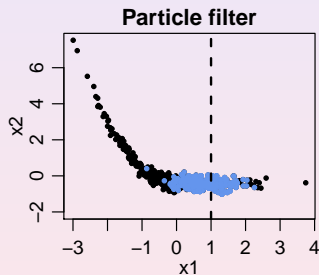
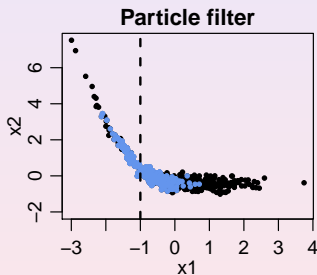
where $\hat{K}_t = K(H, \hat{P}_{t|t-1}, R)$. This is the update for the mean with artificial noise added to the observation.

- ▶ **Deterministic:** Ensemble square root filter. It applies an affine linear transformation to $(x_{t|t-1}^j)$.

Particle filter for banana-shaped prediction

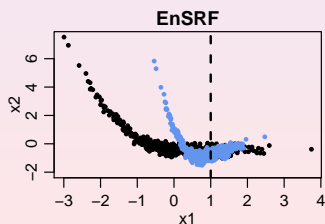
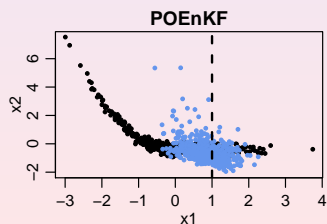
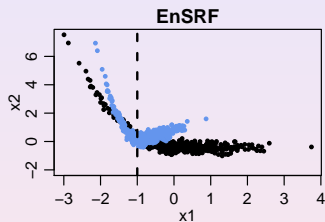
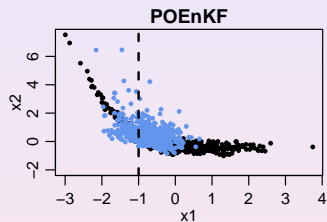
A single observation $y = x_1 + \mathcal{N}(0, 0.5^2) \in \{-1, 1\}$.

Banana-shaped prediction sample (black). Particle filter update (blue).



Ensemble Kalman filter for banana-shaped prediction

A single observation $y = x_1 + \mathcal{N}(0, 0.5^2) \in \{-1, 1\}$. Prediction sample (black). Two versions of the ensemble Kalman filter update (blue). (left: stochastic, right: deterministic).



Comparison of deterministic and stochastic version

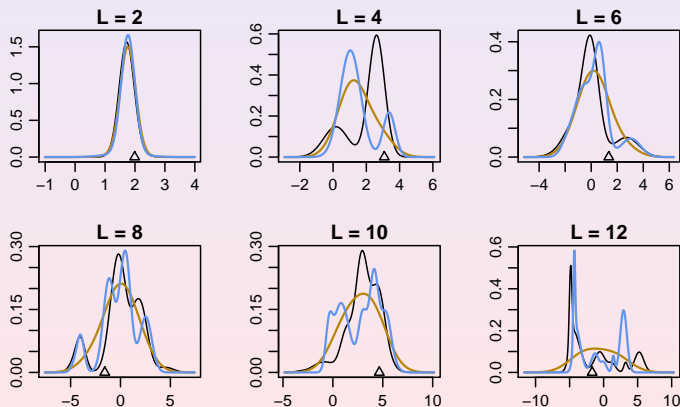
Suppose that $\pi_{t|t-1}$ is not Gaussian. What is the limit of the empirical distribution of $(x_{t|t}^j)$ for $N \rightarrow \infty$? Because prediction sample is modified, the limit distribution is not Gaussian.

Some analysis is possible when $\pi_{t|t-1}$ is a Gaussian mixture. Limit distribution exists for both versions, but is different from the correct update according to Bayes formula, and depends on which version is used.

Typically (see next slide) the stochastic version is too close to a Gaussian distribution, the deterministic version often has peaks at wrong places.

Comparing the two versions: Examples

A single observation $y = x + \mathcal{N}(0, 1)$. The prediction distribution is a Gaussian mixture with L components and composite variance equal to one. True update distribution (black), EnSRF (blue), POEnKF (golden).



Properties of the EnKF

Both versions:

- + With some tuning: Works well also for $N \ll \dim(y)$.
- Not consistent unless $\pi_{t|t-1}$ is Gaussian. Observation y influences only location, but not scale or shape of the updated ensemble. Does not respect constraints $\psi(x(t)) \equiv 0$.

Stochastic:

- + Robust in highly non-linear non-Gaussian systems.
- Loss of skill due to additional Monte Carlo error in update.

Deterministic:

- + No Monte Carlo error in update formula.
- Susceptible to outliers in the prediction ensemble that arise in highly non-linear systems.

Fine tuning

Essentially, two different sources of error may impact filter performance:

- ▶ **Systematic errors**
 - ▶ Misspecified dynamics
 - ▶ Misspecified observation mechanisms (representation errors, wrong error covariances)
 - ▶ Non-Gaussian prediction distributions
- ▶ **Monte Carlo error** due to limited ensemble size. Affects in particular estimation of covariance matrix $P_{t|t-1}$.

Both sources of error typically lead to a too small spread of the updated ensemble, i.e., the uncertainty is underestimated.

Covariance inflation

Basic idea: Artificially increase the spread of the prediction ensemble by a factor $\delta > 1$:

$$x_{t|t-1}^j \mapsto \tilde{x}_{t|t-1}^j = \bar{x}_{t|t-1} + \delta \cdot (x_{t|t-1}^j - \bar{x}_{t|t-1}). \quad (j = 1, \dots, N)$$

Prediction mean is unchanged, but sample covariance is inflated: $\widehat{\text{Cov}}(\tilde{x}_{t|t-1}) = \delta^2 \cdot \widehat{\text{Cov}}(x_{t|t-1})$. Can be used also for modifications of the method that we will discuss below.

Inflation factor δ may be spatially varying (a different factor for each coordinate). δ is typically chosen such that the filter is correctly calibrated, i.e., in the long run the observations should behave as if taken from the prediction or updated ensemble.

Localization

The components of x and y have a spatial interpretation in many applications. An observation y should have low impact on spatially distant state variables.

Two paradigms:

- ▶ Covariance tapering (“covariance filtering, background localization”)
- ▶ Local updates (“observation localization, localization in grid space”)

Localized prediction covariance does not give local updates.

Covariance tapering

Basic idea: Replace the sample covariance matrix $\hat{P}_{t|t-1}$ of the prediction ensemble by a regularized estimate that has a smaller Monte Carlo error.

If components of x correspond to spatial location, regularize by assuming that separated state variables should be uncorrelated (called tapering):

$$\hat{P}_{t|t-1} \mapsto C \circ \hat{P}_{t|t-1},$$

where C is a sparse correlation matrix and \circ denotes entry-wise multiplication.

Taper C is usually constructed via a compactly supported correlation function.

Bridging the two methods: Overview

- ▶ Many methods in the literature attempt to combine advantages of Ensemble Kalman and particle filter.
- ▶ In our experience, these methods are difficult to implement: Either the description of the algorithm is not clear, or it has many tuning parameters which are hard to choose, or we could not reproduce the results.
- ▶ We have proposed two generalizations of the stochastic EnKF, called XEnKF and EnKPF. Both rely on an exact update formula for Gaussian mixtures and have one tuning constant.
- ▶ XEnKF is based on clustering of the prediction particles. Tuning parameter is the number of clusters.
- ▶ In this talk, focus on the EnKPF.

Updates of Gaussian mixtures

If $\pi_{t|t-1}$ is a Gaussian mixture, $\pi_{t|t}$ is again a Gaussian mixture:

$$\pi_{t|t-1} = \sum_{\ell=1}^L \alpha_{t|t-1}^{\ell} \mathcal{N}(\mu_{t|t-1}^{\ell}, \mathbf{P}_{t|t-1}^{\ell}) \Rightarrow \pi_{t|t} = \sum_{\ell=1}^L \alpha_{t|t}^{\ell} \mathcal{N}(\mu_{t|t}^{\ell}, \mathbf{P}_{t|t}^{\ell})$$

where $(\mu_{t|t-1}^{\ell}, \mathbf{P}_{t|t-1}^{\ell}) \rightarrow (\mu_{t|t}^{\ell}, \mathbf{P}_{t|t}^{\ell})$ as in the Kalman filter and

$$\alpha_{t|t}^{\ell} \propto \alpha_{t|t-1}^{\ell} \times \varphi(\mathbf{y} | \mathbf{H} \mu_{t|t-1}^{\ell}, \mathbf{H} \mathbf{P}_{t|t-1}^{\ell} \mathbf{H}' + \mathbf{R})$$

(φ denotes the Gaussian density).

“Particle filter for the weights, Kalman filter for means and covariances”.

Bridging between EnKF and PF: EnKPF

We can apply Bayes formula in two steps (**progressive correction**)

$$\pi_{t|t}(dx_t) \propto \pi_{t|t-1}(dx_t)g(y_t | x_t)^\gamma g(y_t | x_t)^{1-\gamma} \quad (0 \leq \gamma \leq 1).$$

Step 1: From $\pi_{t|t-1}(dx_t)$ to $\pi_{t|t}^\gamma(dx_t) \propto \pi_{t|t-1}(dx_t)g(y_t | x_t)^\gamma$ by Ensemble Kalman Filter: Simply replace R by R/γ or $\hat{P}_{t|t-1}$ by $\gamma\hat{P}_{t|t-1}$ in the Kalman gain.

Step 2: From $\pi_{t|t}^\gamma(dx_t)$ to $\pi_{t|t}(dx) \propto \pi_{t|t}^\gamma(dx_t)g(y_t | x_t)^{1-\gamma}$ by particle filter.

The essential trick is to do both steps analytically, and to sample only at the end. See next slide.

EnKF as a Gaussian mixture

The EnKF in step 1 produces updates

$$x_{t|t}^j = x_{t|t-1}^j + \hat{K}_t^\gamma (y_t + \varepsilon^j - Hx_{t|t-1}^j)$$

where $\hat{K}_t^\gamma = K(H, \gamma \hat{P}_{t|t-1}, R)$. This can be considered as a balanced sample from the Gaussian mixture

$$\hat{\pi}_{t|t}^\gamma = \frac{1}{N} \sum_{j=1}^N \mathcal{N}(x_{t|t-1}^j + \hat{K}_t^\gamma (y - Hx_{t|t-1}^j), \frac{1}{\gamma} \hat{K}_t^\gamma R \hat{K}_t^{\gamma'})$$

There is no need to draw the sample !

Step 2 converts $\hat{\pi}_{t|t}^\gamma$ into another Gaussian mixture from which we sample to get $(x_{t|t}^j)$.

Choice of tuning parameter γ

The exponent γ trades off systematic and Monte Carlo errors:

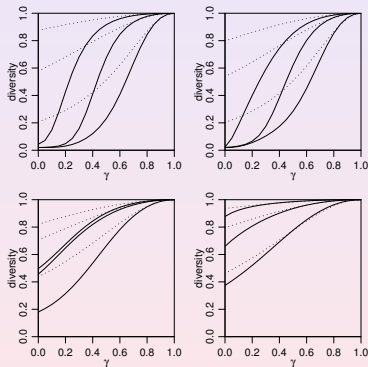
- ▶ $\gamma = 0$ corresponds to particle filter, $\gamma = 1$ to Ensemble Kalman filter, with continuous interpolation in between.
- ▶ A small γ means small bias but large Monte Carlo variance.
- ▶ A large γ means large bias but small Monte Carlo variance.

Need to find an interesting compromise!

Ansatz: Choose γ to achieve a given sampling diversity, i.e., the mixture weights in step 2 should not be too different from uniform weights.

Diversity as a function of γ

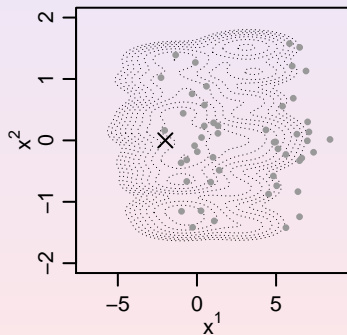
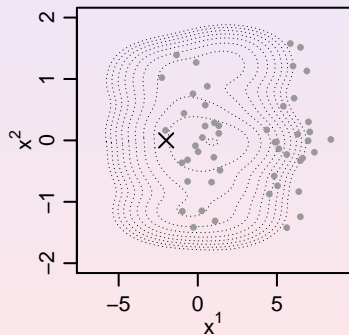
Left/right: Two different values of γ . Two priors: Gaussian (top) and bimodal (bottom). State dimension: 10, 50, 250.



Single update for bimodal prior I

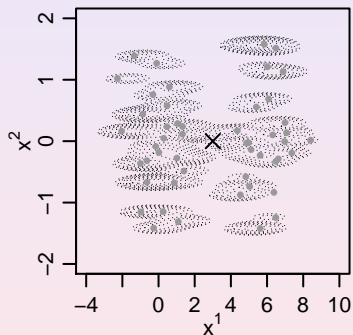
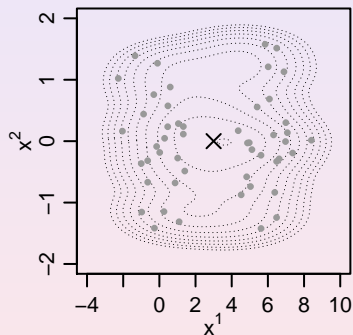
Left: EnKF, Right: EnKPF, diversity $\approx 40\%$.

Dots: Prior sample, Dotted: Contours of underlying update density.



Single update for bimodal prior II

As before, but with observation leading to a bimodal posterior.



Example: Lorenz 96 model: RMSE for 2000 cycles

Every 2nd component observed, $N = 400$.

		10%	median	mean	90%
XEnKF	$L = 2$	0.49	0.70	0.78	1.16
	$L = 2$, mod. weights	0.50	0.70	0.77	1.10
	$L = 5$	0.46	0.70	1.01	2.20
	$L = 5$, mod. weights	0.44	0.63	0.71	1.04
EnKPF	25%–50% diversity	0.49	0.70	0.78	1.16
EnKF	(stochastic version)	0.56	0.81	0.87	1.25

mod. weights: heavy tailed instead of Gaussian likelihood was used.

Summary and Outlook

- ▶ Particle filter fails in high-dimensional problems, EnKF works much better, although its success is still a bit of a mystery.
- ▶ In particular, the errors due to nonlinear state evolution are difficult to assess.
- ▶ Finding algorithms which relax the assumption of a Gaussian prediction distribution remains a challenge. How well do our proposals meet that challenge ?
- ▶ We intend to find out in a large scale weather prediction scheme where the Ensemble Kalman Filter is used.

Section 6

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters

- Particle smoothing

- Parameter estimation

- General sequential Monte Carlo

Smoothing by filtering of paths

We can implement the recursion

$$\begin{aligned}\pi_{0:t|t-1}(dx_{0:t}) &= K(dx_t | x_{t-1})\pi_{0:t-1|t-1}(dx_{0:t-1}) \\ \pi_{0:t|t}(dx_{0:t}) &\propto \pi_{0:t|t-1}(dx_{0:t}) \times g(y_t | x_t).\end{aligned}$$

by a particle filter that generates samples of paths $(x_{0:t|t}^j)$:
Attach the propagated particle to the current path

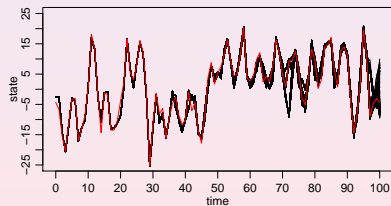
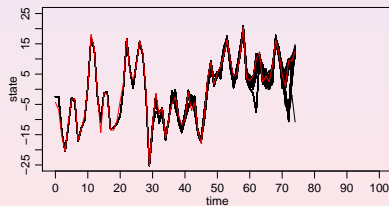
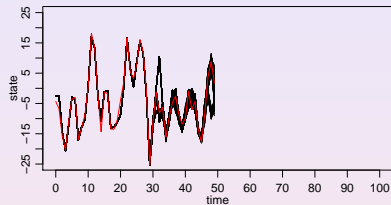
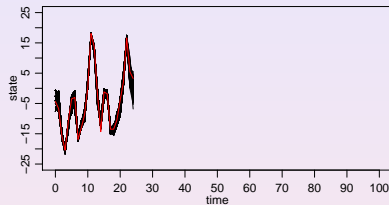
$$x_{0:t|t-1}^j = (x_{0:t-1|t-1}^j, x_{t|t-1}^j)$$

and resample $(x_{0:t|t-1}^j)$ with probabilities $\propto g(y_t | x_{t|t-1}^j)$.

But then $(x_{s|t}^j)$ degenerates quickly to a single value for any fixed s since this component is not rejuvenated. So this method is useless for uncertainty quantification, but it still can generate a reasonable path from $\pi_{0:t|t}$.

Particle filter for $\pi_{0:t|t}$

Black: Filter samples. Red: True state.



Forward filtering, backward smoothing

We can combine the formula

$$\begin{aligned} P(dx_s \mid x_{s+1}, y_{1:t}) &= P(dx_s \mid x_{s+1}, y_{1:s}) \\ &\propto k(x_{s+1} \mid x_s) \pi_{s|s}(dx_s). \end{aligned}$$

with the particle filter approximations of $\pi_{s|s}$ to generate recursively an approximate sample from $\pi_{0:t|t}$:

$$\mathbb{P}(x_{s|t}^j = x_{s|s}^i \mid x_{s+1|t}^j) = \frac{k(x_{s+1|t}^j \mid x_{s|s}^i)}{\sum_{\ell=1}^N k(x_{s+1|t}^j \mid x_{s|s}^\ell)}.$$

If $\pi_{s|t}$ is much more concentrated than $\pi_{s|s}$, diversity is too low. Moreover, sampling from this discrete distribution by inversion leads to an overall complexity $O(N^2)$.

Smoothing by accept/reject

We can get rid of both problems by using the approximation

$$\pi_{s|s} \propto g(y_s | x_s) \sum_{j=1}^N k(x_s | x_{s-1|s-1}^j)$$

instead of the empirical distribution of $(x_{s|s}^j)$. Then

$$p(x_{s|t}^j | x_{s+1|t}^j, y_{1:s}) \propto k(x_{s+1|t}^j | x_s) g(y_s | x_s) \sum_{i=1}^N k(x_s | x_{s-1|s-1}^i).$$

Since we need only one draw from this distribution, we have to use the accept/reject method: Propose an index l^* and a value x_s^* as in the auxiliary particle filter and accept x_s^* as $x_{s|t}^j$ with the appropriate probability.

Smoothing using the two filter formula

A different approach is based on the formula

$$\pi_{s|t} \propto \pi_{s|s-1} p(y_{s:t} | x_s).$$

The second factor is unknown, but it satisfies the recursion

$$\begin{aligned} p(y_{s+1:t} | x_s) &= \int p(y_{s+1:t} | x_{s+1}) k(x_{s+1} | x_s) dx_{s+1} \\ p(y_{s:t} | x_s) &= g(y_s | x_s) p(y_{s+1:t} | x_s). \end{aligned}$$

This has the same structure as the filter relations, except that $p(y_{s:r} | x_s)$ is not a probability density in x_s (hence there is no normalization in the update step).

Approximating the 2-filter formula

We can turn this into a recursion for densities by setting

$$\tilde{\pi}_{s|s}(dx_s) = \frac{p(y_{s:t} | x_s) \gamma_s(x_s)}{\int p(y_{s:t} | x_s) \gamma_s(x_s) dx_s}$$

where γ_s is a density s.th. the denominator is finite.

Then we start with a sample $(\tilde{x}_{t|t}^j)$ from $\tilde{\pi}_{t|t} \propto g(y_t | x_t) \gamma_t$ and construct recursively approximate samples $(\tilde{x}_{s|s}^j)$ from $\tilde{\pi}_{s|s}$ by propagation, weighting and resampling.

In this way we obtain an approximation of $\pi_{s|t}$ with density proportional to

$$\underbrace{\sum_{i=1}^N k(x_s | x_{s-1|s-1}^i) g(y_s | x_s)}_{\approx \pi_{s|s}} \underbrace{\sum_{j=1}^N \frac{k(\tilde{x}_{s+1|s+1}^j | x_s)}{\gamma_{s+1}(\tilde{x}_{s+1|s+1}^j)}}_{\approx p(y_{s+1:t} | x_s)}.$$

An $O(N)$ smoothing algorithm

The density on the previous slide is a mixture of N^2 components. Like in the auxiliary particle filter we generate a weighted sample by sampling first the two indices of the mixture and then x_s from an approximation to the chosen component and finally by weighting.

If we try to use an approximately optimal choice of the two indices, we end up again by an $O(N^2)$ algorithm. However, we obtain an $O(N)$ algorithm if we choose the two indices independently. The price we pay is a potential loss of effective sample size.

Fearnhead, Wyncoll and Tawn, *Biometrika* (2010).

Parameters as components of the state

We discuss next the estimation of parameters θ in the state transition K and/or the observation density g .

The easiest method is to include θ as a deterministic component of the state:

$$K(d\theta_t, dx_t \mid \theta_{t-1}, x_{t-1}) = \Delta_{\theta_{t-1}}(d\theta_t)K(dx_t \mid x_{t-1}, \theta_{t-1})$$

where Δ is a point mass. The particle filter degenerates quickly because there is no rejuvenation of the θ -component.

One can avoid this by adding noise to $\theta_{t|t-1}^j$, possibly combined with shrinkage to the mean. But we need to let the variance of the noise go to zero in order that $(\theta_{t|t}^j)$ approximates $p(\theta \mid y_{1:t})$.

Particle MCMC

In the linear Gaussian case, we used simultaneous updates of parameters and states: We proposed

$$(x_{0:t}^*, \theta^*) \mid (x_{0:t}, \theta) \sim q(\theta^* \mid \theta) d\theta^* \pi_{0:t|t, \theta^*}(dx_{0:t}^*)$$

and accepted it with probability $\min(1, p(\theta^* \mid y_{1:t})/p(\theta \mid y_{1:t}))$.

This cannot be used in general because we cannot draw true realizations from $\pi_{0:t|t, \theta}$, and for the acceptance probability we need $p(y_{1:t} \mid \theta)$ which we cannot compute exactly.

Andrieu et al. JRSS B (2010) showed that using particle approximations at both instances still gives a consistent algorithm.

The particle MCMC algorithm

Assume we have currently the values $(\theta, x_{0:t})$ for parameters and states and an approximation $\hat{p}(y_{1:t} | \theta)$. Then we do

- ▶ Choose a new θ^* according to $q(\theta^* | \theta)$.
- ▶ Use a particle filter to generate a sample $(x_{0:t|t}^j)$ from $\pi_{0:t|t, \theta^*}$ and an approximation $\hat{p}(y_{1:t} | \theta^*)$.
- ▶ Choose $x_{0:t}^*$ uniformly from $(x_{0:t|t}^j)$.
- ▶ Accept $(\theta^*, x_{0,t}^*)$ with probability

$$\min \left(1, \frac{p(\theta^*) \hat{p}(y_{1:t} | \theta^*)}{p(\theta) \hat{p}(y_{1:t} | \theta)} \right).$$

Sampling from moving targets

In many applications outside from state space models, one also wants to sample not from one distribution π , but from a sequence of related distributions π_t , $t = 0, 1, \dots, n$.

In particular, if the target π is difficult to sample, one can start with a simpler distribution π_0 and construct an approximating sequence such that $\pi_n = \pi$. Examples are importance splitting as discussed in the Introduction, or simulated tempering where

$$\pi_t(dx) \propto \left(\frac{d\pi}{d\pi_0}(x) \right)^{\beta_t} \pi_0(dx) \quad (0 = \beta_0 < \beta_1 < \dots < \beta_n = 1).$$

To simplify the notation, assume that all π_t have densities which are known up to a normalizing constant.

Sequential sampling

Generalizing the particle filter, we want to approximate π_t by a sequence of particles (x_t^j) which evolve by propagation and resampling.

If in the propagation step $x_t^{*j} \sim q_t(x_t | x_{t-1}^j) dx_t$ (independently for different j 's), then resampling must be with probabilities

$$w_t^j \propto \tilde{w}_t(x_t^{*j}) = \frac{\pi_t(x_t^{*j})}{\int \pi_{t-1}(x_{t-1}) q_t(x_t^{*j} | x_{t-1}) dx_{t-1}}.$$

But typically, the integral in the denominator cannot be computed analytically. One exception is when q_t leaves π_{t-1} invariant, e.g. a Metropolis-Hastings transition. But then we move from π_{t-1} to π_t only by importance resampling, propagation just does some rejuvenation by breaking ties from the resampling step.

Resampling pairs of particles

- ▶ We cannot compute the density of (x_t^{*j}) , but the density of (x_{t-1}^j, x_t^{*j}) is known: $\pi_{t-1}(x_{t-1})q_t(x_t | x_{t-1})$.
- ▶ Importance resampling can convert (x_{t-1}^j, x_t^{*j}) to a sample from a distribution with second marginal π_t .
- ▶ Such distributions have the density $\pi_t(x_t)r_{t-1}(x_{t-1} | x_t)$ where r_{t-1} is an arbitrary transition density.
- ▶ By marginalization, we can choose any “backward transition” r_{t-1} and resample (x_t^{*j}) with probabilities

$$w_t^j \propto \tilde{w}_t(x_{t-1}^j, x_t^{*j}) = \frac{\pi_t(x_t^{*j})r_{t-1}(x_{t-1}^j | x_t^{*j})}{\pi_{t-1}(x_{t-1}^j)q_t(x_t^{*j} | x_{t-1}^j)}.$$

Choice of the transitions

We are free to choose forward and backward transitions q_t and r_{t-1} . For given q_t , the variance of \tilde{w}_t is minimal if

$$r_{t-1}(x_{t-1} | x_t) = \frac{\pi_{t-1}(x_{t-1})q_t(x_t | x_{t-1})}{\int \pi_{t-1}(x_{t-1})q_t(x_t | x_{t-1})dx_{t-1}},$$

bringing us back to the problem at the start. Still, one can approximate the optimal choice by something which does not involve an integral.

E.g. approximating $\log \pi_{t-1}(x_{t-1}) + \log q_t(x_t | x_{t-1})$ by a quadratic expression in x_{t-1} , will lead to a Gaussian backward density.

The case of the particle filter

For the particle filter $\pi_t = \pi_{t|t}$ for which we do not have an explicit expression up to a normalizing constant. But if we choose the pair target density $\pi_t(x_t)r_{t-1}(x_{t-1} | x_t)$ to be

$$\pi_{t-1:t|t}(x_t, x_{t-1}) = \frac{\pi_{t-1}(x_{t-1})k(x_t | x_{t-1})g(y_t | x_t)}{p(y_t | y_{1:t-1})},$$

the weights are explicit up to a normalizing constant

$$\tilde{w}_t(x_{t-1}, x_t) \propto \frac{k(x_t | x_{t-1})g(y_t | x_t)}{q(x_t | x_{t-1}, y_t)}.$$

As stated earlier, the optimal choice of q is $p(x_t | x_{t-1}, y_t)$.

Sequential Monte Carlo squared

Chopin, Jacob and Papaspiliopoulos, JRSS B (2013).

Sequential Monte Carlo can be used to sample from a posterior $\pi(\theta) = p(\theta|y_{1:n})$: Simply take $\pi_t = p(\theta|y_{1:t})$ and use a forward transition q_t which leaves π_{t-1} invariant. Reweighting is then by

$$w_t^j \propto \frac{p(\theta_t^j | y_{1:t})}{p(\theta_t^j | y_{1:t-1})} = p(y_t | y_{1:t-1}, \theta_t^j).$$

In state space models, $p(y_t | y_{1:t-1}, \theta)$ is not available analytically. The idea of (SMC)² is to approximate it by running particle filters for each θ_t^j . Again, this gives only approximate weights, but they show that the method is still consistent.

Approximate Bayesian computation (ABC)

- ▶ Models where computing the likelihood $p(y | \theta)$ is much harder than to simulate from it occur frequently.
- ▶ Examples: State space models, models in genetics, other models with many latent variables.
- ▶ Basic ABC “algorithm”: Draw a parameter θ from the prior and an observation from the corresponding likelihood. If **simulated observation $y_{\text{sim}} \neq$ actual observation y_{obs}** , reject the parameter and repeat. The accepted parameters are then a sample from the posterior.
- ▶ ABC tries to obtain a practical algorithm by relaxing the condition for an exact match with the actual observation.

Relaxing the exact match condition

- ▶ Keep the draws with $d(y_{\text{sim}}, y_{\text{obs}}) \leq \varepsilon$ where d is a metric on the space of observations.
- ▶ Reduce dimension by choosing a statistic (feature) S and keep the draws with $d(S(y_{\text{sim}}), S(y_{\text{obs}})) \leq \varepsilon$.
- ▶ Use sequential Monte Carlo to sample from

$$\pi_t(y, \theta) \propto p(\theta)p(y | \theta) \exp(-d(S(y), S(y_{\text{obs}}))/\varepsilon_t)$$

for large ε_0 and small ε_n .

- ▶ Because $p(y | \theta)$ is unknown, the choice of forward transitions is limited: Have to take Metropolis-Hastings with proposal $q(\theta^* | \theta)p(y^* | \theta^*)$.
- ▶ Albert et al. propose to make only Metropolis-Hastings transitions, without any importance weighting and resampling (as in simulated annealing). Use the set of particles to derive a good cooling schedule (ε_t), based on ideas from non-equilibrium thermodynamics.

Summary

- ▶ State space models have an unobserved Markovian state process and conditionally independent observations. They are used in many applications.
- ▶ Linear Gaussian state space models with unknown parameters can be fitted with the Kalman filter and smoother. Some extensions to nonlinear/non-Gaussian observations are possible.
- ▶ The particle filter samples from the current state given observations and parameters. The auxiliary particle filter can reduce the problem of sample depletion.
- ▶ Ensemble Kalman filter is an alternative to the particle filter in high dimensions.
- ▶ There are more complex methods to extend particle filtering for sampling all unobserved variables and the parameters.
- ▶ Particle filter ideas can be used to sample from a complicated distribution using successive approximations.

A few references

- ▶ C. Andrieu, A. Doucet and R. Holenstein. Particle Markov chain Monte Carlo methods, JRSS B 72 (2010).
- ▶ A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In Handbook on Nonlinear Filtering, Oxford, 2011.
- ▶ G. Evensen. Data Assimilation: The Ensemble Kalman Filter, Springer, 2007.
- ▶ M. Frei and H. R. Künsch. Bridging the ensemble Kalman and particle filter, Biometrika 100 (2013).
- ▶ H. R. Künsch State space and hidden Markov models. In Complex Stochastic Systems, CRC Press (2001).
- ▶ H. R. Künsch, Particle filters. Bernoulli 19 (2013).
- ▶ F. Sigrist, H. R. Künsch and W. Stahel. SPDE based modeling of large space-time data sets. JRSS B (2014), in press.