Ensemble data assimilation and particle filters

Hans R. Künsch

Seminar for Statistics
ETH Zurich

Geilo Winter School, January 2019

Original parts are based on joint work with Marco Frei and Sylvain Robert

About myself

- I was born in Zurich, went to school in Zurich, studied math at ETH Zurich, did my PhD at ETH Zurich and was a professor at ETH Zurich for over 30 years.
- Between my first degree and my PhD I studied for 2 years in Tokyo, Japan, and I was a postdoc again in Tokyo for 1.5 years.
- My research started in probability theory, then I moved to statistics where I worked in the areas of robust statistics, spatial statistics and time series, and I am interested in environmental applications (soil, aquatic systems, climate and weather).
- My two most cited papers are “The jackknife and the bootstrap for general stationary observations” and “Practical identifiability analysis of large environmental simulation models”
- As a mathematician, I like formulae because they help me to think clearly. I will do my best to explain what a formula wants to say.

Overview I

- I will use the framework of state space models. These are dynamical systems with partial and noisy observations at discrete time points.
- Examples are numerical or stochastic models for weather, earthquakes, flow in porous media, or statistical models in economics, finance, ecology, systems biology, etc..
- Data assimilation or filtering is the estimation of the state of the system at some time t and the quantification of its uncertainty, given all observations up to time t. This is the basis for predicting the system.
- State space models often contain unknown static parameters related to the time evolution of the system or to the measurement process. Methods to estimate such parameters also rely on filtering.

Overview II

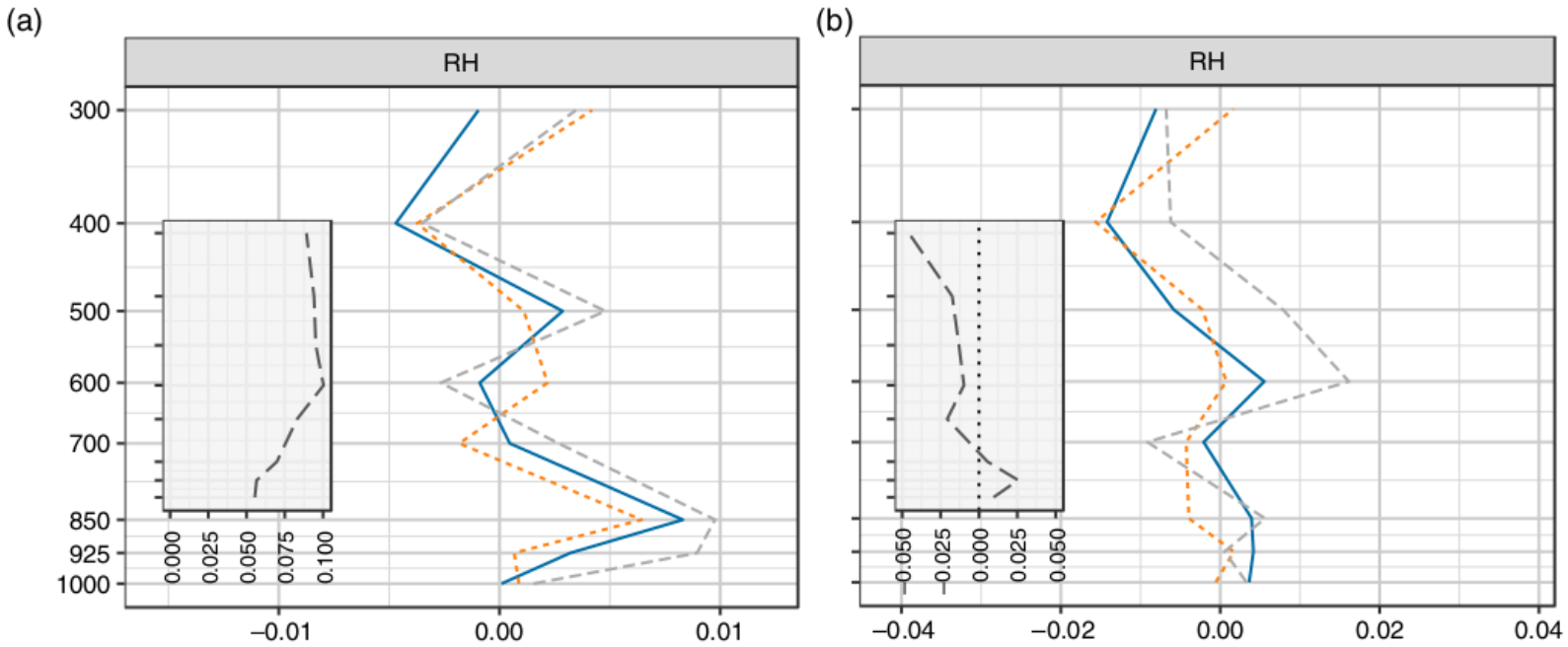
- In these lectures, I want to introduce the basic concepts for non-specialists and give an outlook on some new and ongoing research.
- Statisticians, geophysicists and applied mathematicians have made contributions, often without much exchange of ideas and methods. This has only recently started to change.
- It is not possible to cover everything, the selection is my own.
- I prepared these slides without knowing the exact contents and the notation in the lectures by Remus. I have made some adjustments in the last minutes, but the coordination is not perfect.
- The emphasis here is on the construction and heuristic properties of algorithms.

Achievements and challenges

- Particle filters have been extremely successful in tracking problems and image analysis because they can deal with occasional ambiguity.
- Local Ensemble Transform Kalman filters are regularly used in operational weather forecasting. Only few particle filter based methods are have been tested in weather forecasting experiments with set-ups close to operational conditions.
- Problems with unknown static parameters are much harder. Particle MCMC are promising, but are until now limited to problems of intermediate complexity.
- ...

A cycled experiment in large scale weather prediction

Verification of different 1-h forecasts of relative humidity against all radiosonde measurements over 12 days. Two scores are used, CRPS (left) and bias (right). In the big plot, 3 particle type methods are compared to an Ensemble Kalman method (negative values imply improvements), the small plot shows the score of the Ensemble Kalman method.



State space models

A state space model consists of a dynamical system $(\boldsymbol{X}_t)$ and partial and noisy observations $(\boldsymbol{Y}_i)$ of the state of the system at some discrete time points t_i.

$\boldsymbol{X}_t$ contains a complete description of the system at time t. Its dynamics is given by a differential equation or a Markov process in discrete or continuous time. $\boldsymbol{X}_t$ is however not fully observable.

Observations $\boldsymbol{Y}_i$ are conditionally independent given the state process, and $\boldsymbol{Y}_i$ depends only on $\boldsymbol{X}_{t_i}$.

The goal is to estimate and predict the state of the system sequentially based on observations $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots$. In some applications, one has to estimate also parameters of the dynamics of the system or of the distribution of the observation noise.

Example 1: Lorenz 96

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 were proposed by Ed Lorenz in 1963 and 1996.

The Lorenz 96 model is

$$\frac{dX_{t,k}}{dt} = (X_{t,k-1} - X_{t,k-2})X_{t,k-1} - X_{t,k} + 8, \quad k = 1, \ldots, 40$$

with circular state components: $X_{t,k} \equiv X_{t,k+40}$. It mimicks large scale motions of a one-dimensional atmosphere.

Every second component is observed with additive Gaussian noise.

Example 2: A nonlinear one-dim model

This example was constructed (Andrade-Netto et al.) to illustrate the effects of nonlinear dynamics and observations on filtering. I will use it in the exercise

$$\begin{aligned} X_t &= M_{t-1}(X_{t-1}) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, 10) \\ M_t(x) &= \frac{x}{2} + 25\frac{x}{1+x^2} + 8\cos(1.2 \cdot t) \\ Y_t &= 0.05X_t^2 + \epsilon_t, \quad \epsilon_t \sim N(0, 1). \end{aligned}$$

M_t is strongly expansive near the origin, but contractive for $|x|$ large. Because only the square of the state is observed, the sign must be identified from the dynamics of the state.

Example 3: Fish population

This model is used to analyze data on the abundance of cod in the North Atlantic (Aeberhard et al. 2018, submitted). Here X_t contains the abundances $N_{a,t}$ and the fishing mortality rates $F_{a,t}$ for age classes $a = (\leq 3, 4, \ldots, 9, \geq 10)$ and year t. The observations consist of the number of catches $C_{a,t}$ and indices $I_{a,t}$ from surveys.

The dynamics of the state is based on "biological common sense", $N_{a,t}$ = number of fish alive in age class $a-1$ at the beginning of year $t-1$ that didn't die of natural causes nor got caught during year $t-1$, and simple persistence assumptions, $F_{a,t} = F_{a,t-1}$, both with multiplicative noise.

The catches $C_{a,t}$ are proportional to the number of fish which died during year t (with constant equal to ratio of fishing to total mortality) and the surveys $I_{a,t}$ are proportional to the number of fish alive in year t (with constants equal to so-called catchability coefficients), again with multiplicative noise.

State space models: Operational form

To make the notation easier, I assume that the model is time homogeneous and observation times are $t_i = i$ and I write t instead of t_i. Then a general state space model has the form

$$\boldsymbol{X}_t = M(\boldsymbol{X}_{t-1}, \eta_t), \quad \boldsymbol{Y}_t = H(\boldsymbol{X}_t, \epsilon_t)$$

where the system and observation noise variables η_t and ϵ_t are independent.

This formulation is most useful if you want to simulate the model: Choose an initial value $\boldsymbol{x}_0$, simulate the noise variables and then proceed iteratively. However, for other purposes an equivalent formulation with conditional distributions is prefered.

State space models: Conditional distributions

For the state variables, we have:

$$\boldsymbol{X}_t|\boldsymbol{X}_{t-1} = \boldsymbol{x}_{t-1} \sim m(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})d\boldsymbol{x}_t$$

and this is not changed if additional state variables from the past are known (Markov property). For the observations, we have

$$\boldsymbol{Y}_t|\boldsymbol{X}_t = \boldsymbol{x}_t \sim h(\boldsymbol{y}_t|\boldsymbol{x}_t)d\boldsymbol{y}_t$$

and this conditional distribution is not changed if additional states or observations (past or future) are known.

Passing from one of these two descriptions to the other is difficult in general. If the noise variables are additive, then

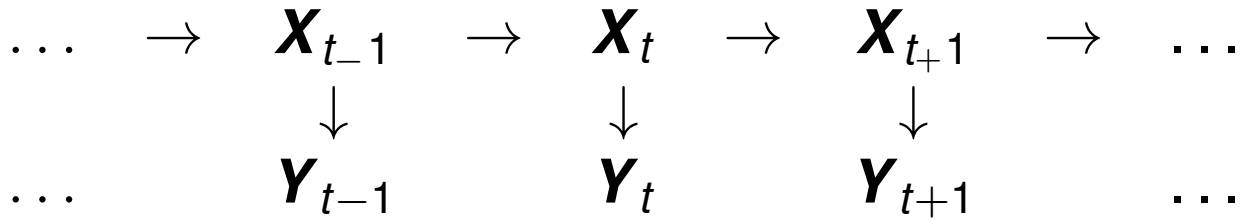
$$m(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = f_\eta(\boldsymbol{x}_t - M(\boldsymbol{x}_{t-1}))$$

and similarly for h. The case of a deterministic state dynamics is included with a Dirac function for f_η.

For some methods we only need to simulate from the model. However, h must be known and should be a proper density.

Graphical representation of state space models

The conditional independence properties between the variables of a state space model from the previous slide can be represented by the following directed acyclic graph



From this graph, additional conditional independence relations can be deduced. We will see examples in the third lecture when we discuss smoothing.

The graph also shows that $\boldsymbol{Y}_t$ depends on all past observations, i.e. $(\boldsymbol{Y}_t)$ is not a Markov chain.

Basics of data assimilation/filtering

For predicting the state at time t based on all observations up to time $t-1$ we need the prediction density, the conditional density of $\boldsymbol{X}_t$ when $\boldsymbol{Y}_1 = \boldsymbol{y}_1$, $\boldsymbol{Y}_2 = \boldsymbol{y}_2$, ..., $\boldsymbol{Y}_{t-1} = \boldsymbol{y}_{t-1}$, or in shorthand notation $\boldsymbol{Y}_{1:t-1} = \boldsymbol{y}_{1:t-1}$, is known.

For estimating the state at time t based on all observations up to time t we need the filter density, the conditional density of $\boldsymbol{X}_t$ when $\boldsymbol{Y}_{1:t} = \boldsymbol{y}_{1:t}$, is known.

The means (or medians) of these densities give the best estimates, the (co)variance and quantiles quantify uncertainty.

Because these densities appear all the time, I use the special symbols π_t^p and π_t^f for them. Moreover, because the values $\boldsymbol{y}_{1:t-1}$ of the observations are considered fixed, I write $\pi_t^p(\boldsymbol{x}_t)$ instead of $\pi_t^p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$. If there is no confusion, I also drop the subscript t.

Data assimilation uses the terms background instead of prediction and analysis instead of filter density.

Recursions for prediction and filter densities

The prediction and filter densities can be computed recursively:

$$\ldots \to \pi^f_{t-1} \to \pi^p_t \to \pi^f_t \to \ldots$$

$\pi^f_{t-1} \to \pi^p_t$ (Propagation): By the law of total probability (and conditional independence):

$$\pi^p_t(\boldsymbol{x}_t) = \int m(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})\pi^f_{t-1}(\boldsymbol{x}_{t-1})d\boldsymbol{x}_{t-1}$$

$\pi^p_t \to \pi^f_t$ (Update): By Bayes formula with prior π^p and likelihood h:

$$\pi^f_t(\boldsymbol{x}_t) = \frac{\pi^p_t(\boldsymbol{x}_t)h(\boldsymbol{y}_t|\boldsymbol{x}_t)}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})} \propto \pi^p_t(\boldsymbol{x}_t)h(\boldsymbol{y}_t|\boldsymbol{x}_t)$$

where $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \int \pi^p_t(\boldsymbol{x}_t)h(\boldsymbol{y}_t|\boldsymbol{x}_t)d\boldsymbol{x}_t$.

Monte Carlo (Ensemble) filters

Recursions from the previous slides typically cannot be computed analytically or numerically, except in the linear Gaussian case (Kalman filter) or when the state space is finite (Baum-Welch).

Monte Carlo filters approximate π_t^p and π_t^f by samples or ensembles of weighted "particles" $(\boldsymbol{x}_{t,j}^p, w_{t,j}^p)$ and $(\boldsymbol{x}_{t,j}^f, w_{t,j}^f)$ $(j = 1, 2, \ldots, N)$. Then

$$\pi_t^p(\boldsymbol{x}_t) \approx \sum_{j=1}^{N} w_{t,j}^p \, \delta(\boldsymbol{x}_t - \boldsymbol{x}_{t,j}^p)$$

and for any (bounded) test function ψ of the state:

$$\mathbb{E}\left[\psi(\boldsymbol{X}_t)|\boldsymbol{y}_{1:t-1}\right] = \int \psi(\boldsymbol{x}_t)\pi_t^p(\boldsymbol{x}_t)d\boldsymbol{x}_t \approx \sum_{j=1}^{N} \psi(\boldsymbol{x}_{t,j}^p) w_{t,j}^p$$

and similarly for the filter.

Propagation and update for particles

In the propagation step, filter particles move forward according to the dynamics of the state, independently of each other, to become the next prediction particles. Weights do not change:

$$\boldsymbol{x}_{t,j}^{p} \sim m(\boldsymbol{x}_t | \boldsymbol{x}_{t-1,j}^{f}), \quad w_{t,j}^{p} = w_{t-1,j}^{f}$$

The computational burden of this step often limits the size N of the ensemble.

Updating converts the prediction sample into the filter sample by changing the weights and/or the positions of particles. The two main methods are the Particle Filter (PF) and the Ensemble Kalman Filter (EnKF). Since we consider the update for a fixed time point t I drop the subscript t in the following.

Update step for basic particle filter

The particles $\boldsymbol{x}_j^f$ should be drawn from $\pi^f(\boldsymbol{x}) \propto \pi^p(\boldsymbol{x}) h(\boldsymbol{y}|\boldsymbol{x})$. Drawing from the posterior in high dimensions is difficult in general. Here it is worse because there is no formula for π^p, only a sample.

The particle filter uses

$$\pi^p(\boldsymbol{x}) \approx \sum_{j=1}^{N} w_j^p \, \delta(\boldsymbol{x} - \boldsymbol{x}_j^p)$$

Then Bayes formula gives

$$\pi^f(\boldsymbol{x}) \approx \sum_{j=1}^{N} w_j^f \, \delta(\boldsymbol{x} - \boldsymbol{x}_j^p), \quad w_j^f \propto w_j^p \, h(\boldsymbol{y}|\boldsymbol{x}_j^p).$$

Hence the particles don't move, only the weights change, depending on how well a particle fits to the new observation.

Particle filter update: Resampling

Problem: In the iteration weights become quickly unbalanced, and computation is wasted for extremely unlikely time evolutions. In the end, the filter looses track.

Basic remedy to counteract weight unbalance is resampling:

$$\text{Set } \boldsymbol{x}_i^f = \boldsymbol{x}_j^p \text{ with probability } w_j^p, \quad w_i^f = \frac{1}{N}$$

Particles $\boldsymbol{x}_j^p$ with a poor fit to the new observation die, those with a good fit appear $N_j = |\{i; \boldsymbol{x}_i^f = \boldsymbol{x}_j^p\}|$ times.

Resampling creates ties among particles and reduces diversity. If the dynamics of the state is stochastic and particles are propagated independently, some diversity is restored, but one does not know if it represents the true uncertainty at the next prediction. Resampling is only a partial remedy.

Resampling and effective sample size

Resampling introduces an additional Monte Carlo error because w_j^f is replaced by the relative frequency N_j/N. To reduce this, resample only for the next propagation and only if diversity of weights is low.

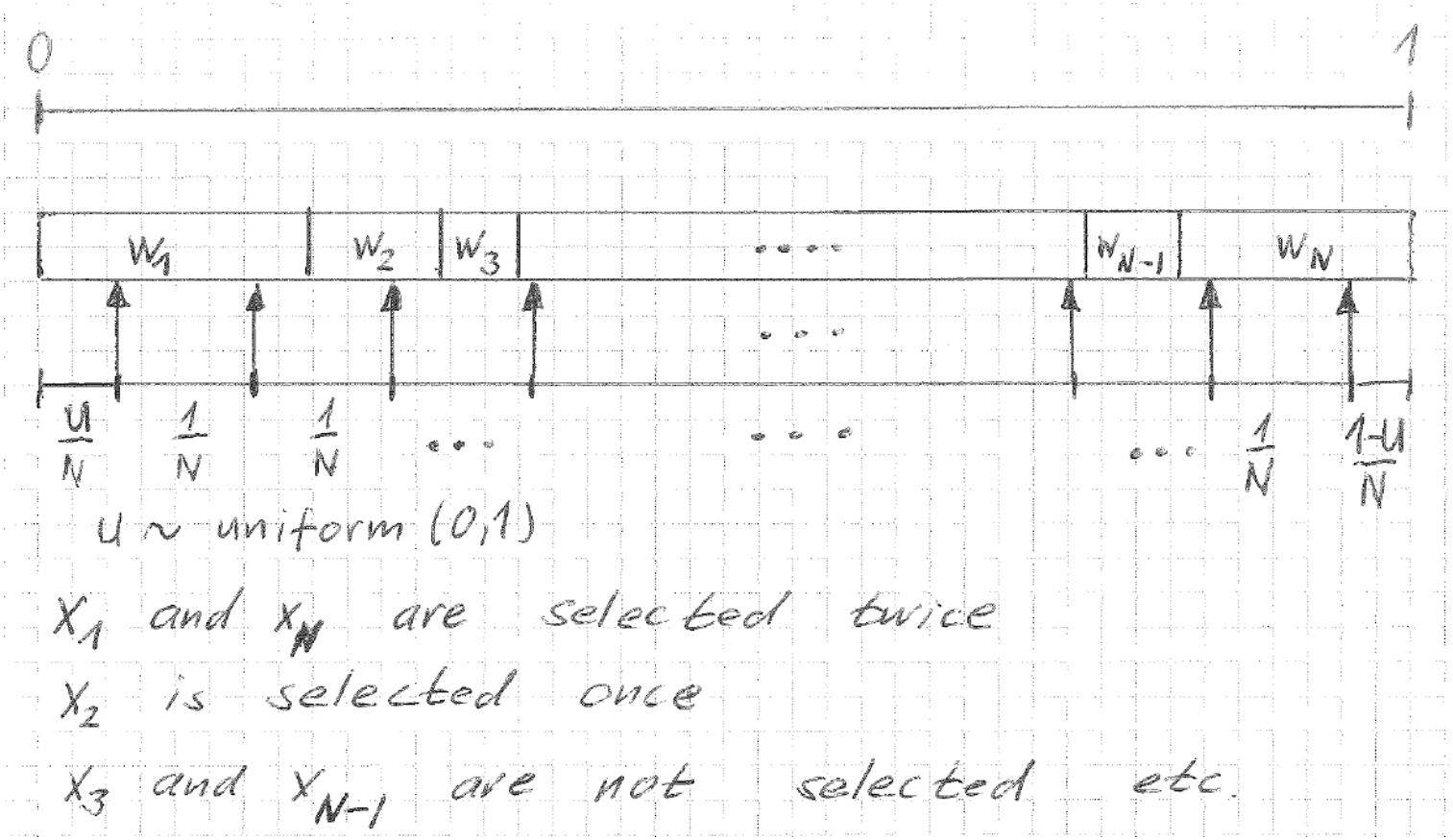
Diversity is usually measured by effective sample size ESS:

$$\mathrm{ESS} = \left(\sum_{j=1}^{N} (w_j^f)^2 \right)^{-1} .$$

$\mathrm{ESS} = 1$ if one $w_j^f = 1$, $\mathrm{ESS} = N$ if all $w_j^f = 1/N$. The definition is based on an approximation of the asymptotic variance of weighted samples (Liu, 1996).

Balanced sampling

Monte Carlo error of resampling can also be reduced by balanced sampling, meaning that $|N_j - Nw_j^f| < 1$. The figure illustrates such a scheme.



Update for the basic EnKF

The Ensemble Kalman filter estimates the mean μ^p and covariance P^p of π^p and assumes that π^p is Gaussian (and hence π^p is determined).

If $h = \mathcal{N}(H\boldsymbol{x}, R)$ and $\pi^p = \mathcal{N}(\mu^p, P^p)$, then by Bayes formula $\pi^f = \mathcal{N}(\mu^f, P^f)$ where

$$\mu^f = \mu^p + K(\boldsymbol{y} - H\mu^p), \quad P^f = P^p - KHP^p$$

and K is the Kalman gain

$$K = P^p H^T (HP^p H^T + R)^{-1}$$

The Ensemble Kalman filter plugs the estimated values of μ^p and P^p into these formulae and then transforms the particles $\boldsymbol{x}_i^p$ into particles $\boldsymbol{x}_i^f$ with the desired first two moments.

Stochastic EnKF

Let $\widehat{K}$ be the Kalman gain with estimated prediction covariance $\widehat{P}^p$ (the observation error covariance R is assumed to be known).

The stochastic version of the EnKF generates N artificial observation errors $\epsilon_i \sim \mathcal{N}(0, R)$ and sets

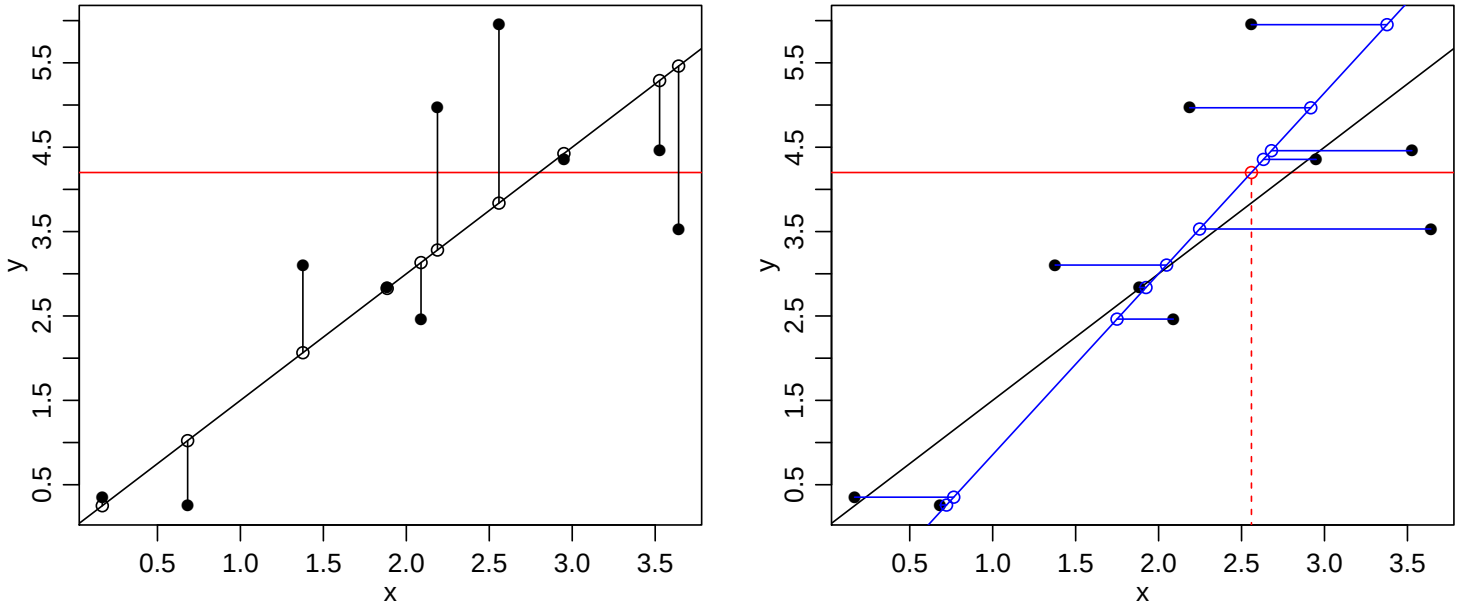
$$\boldsymbol{x}_i^f = \boldsymbol{x}_i^p + \widehat{K}(\boldsymbol{y} - H\boldsymbol{x}_i^p - \epsilon_i)$$

The move of particle $\boldsymbol{x}_i^p$ depends on how much the actual observation $\boldsymbol{y}$ differs from an artificial observation $\boldsymbol{y}_i^p = H\boldsymbol{x}_i^p + \epsilon_i$, assuming that the state equals $\boldsymbol{x}_i^p$. One can check that the filter ensemble has the desired first two moments.

There is a regression interpretation of this: One regresses $\boldsymbol{y}_i^p$ on $\boldsymbol{x}_i^p$ The estimated regression line applied to the actual observation y gives the filter mean $\bar{\boldsymbol{x}}^f$, the residuals quantify uncertainty.

Regression interpretation of EnKF

Left: Forward regression line $y = Hx$ and points (x_i^p, y_j^p). Red line is at actual observation y.
Right: Inverse regression line $x = \bar{x}^p + \hat{K}(y - \bar{y}^p)$. Dotted red line is at $\bar{x}^f$, horizontal residuals equal $x_i^f - \bar{x}^f$.



The square root EnKF

The square root version of the EnKF computes $\bar{\boldsymbol{x}}^f$ and sets

$$\boldsymbol{x}_i^f = \bar{\boldsymbol{x}}^f + (\Delta \boldsymbol{X}^p) W$$

where the matrix $\Delta \boldsymbol{X}^p = (\boldsymbol{x}_1^p - \bar{\boldsymbol{x}}^p | ... | \boldsymbol{x}_N^p - \bar{\boldsymbol{x}}^p)$ and W satisfies a certain matrix equation.

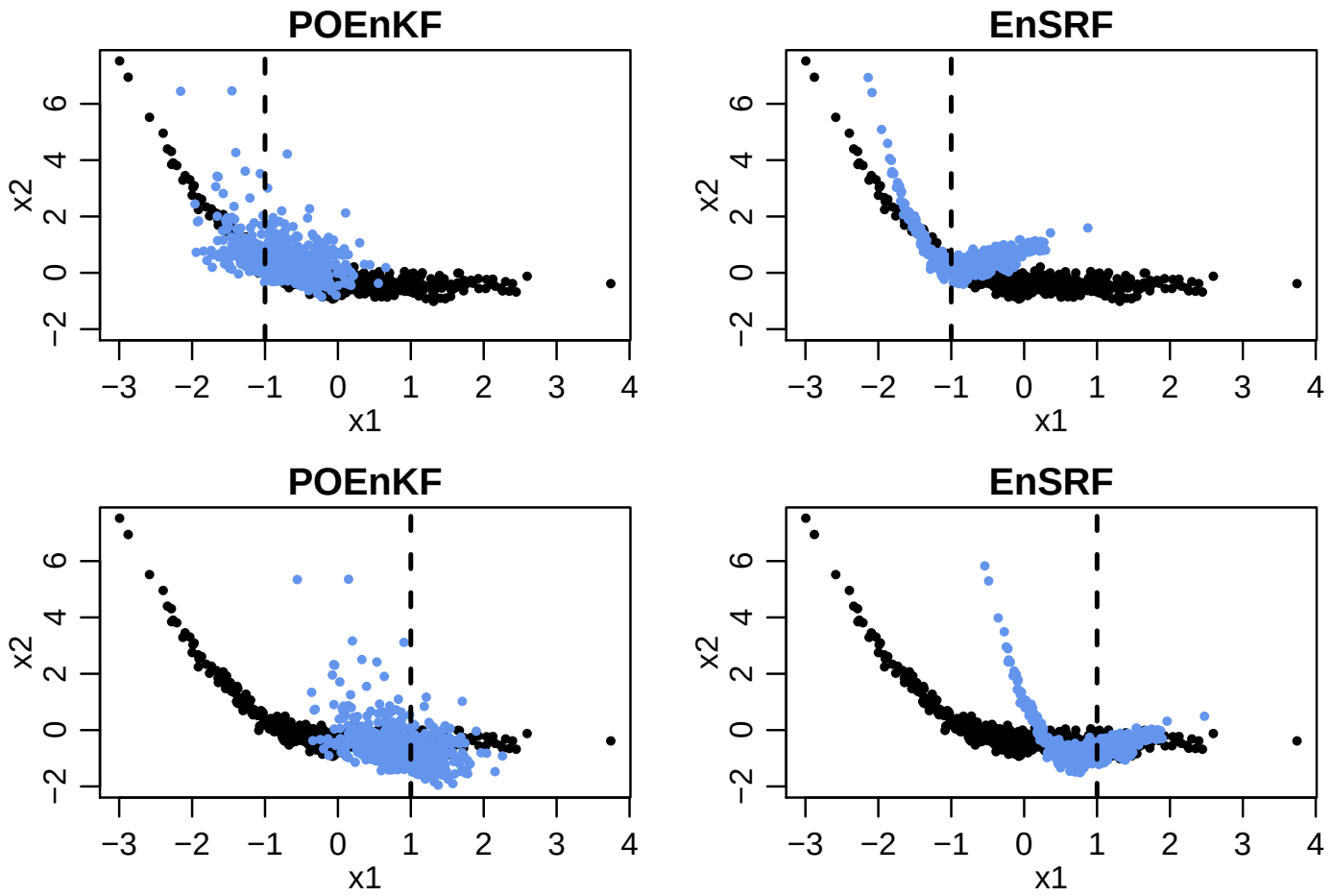
In both versions, changing the observation changes the location of the ensemble, but not the spread nor the shape of the point cloud. For a non-Gaussian prior or a non-Gaussian likelihood, the shape and the spread of the posterior typically changes if the observed value changes.

Particle vs. Ensemble Kalman filter

- PF originated in statistics (Gordon et al., 1993), EnKF in geophysics (Evensen, 1994).
- PF uses weighting and resampling. It works for arbitrary observation densities h.
- PF is consistent under very weak assumptions, but degenerates easily, in particular in high dimensions.
- EnKF moves the particles towards the observations. The algorithm (essentially) assumes additive observation error with constant variance.
- EnKF is consistent only if observation is a linear function of the state plus independent Gaussian errors and if π^p is Gaussian. However, a simple modification makes it extremely robust in practice.

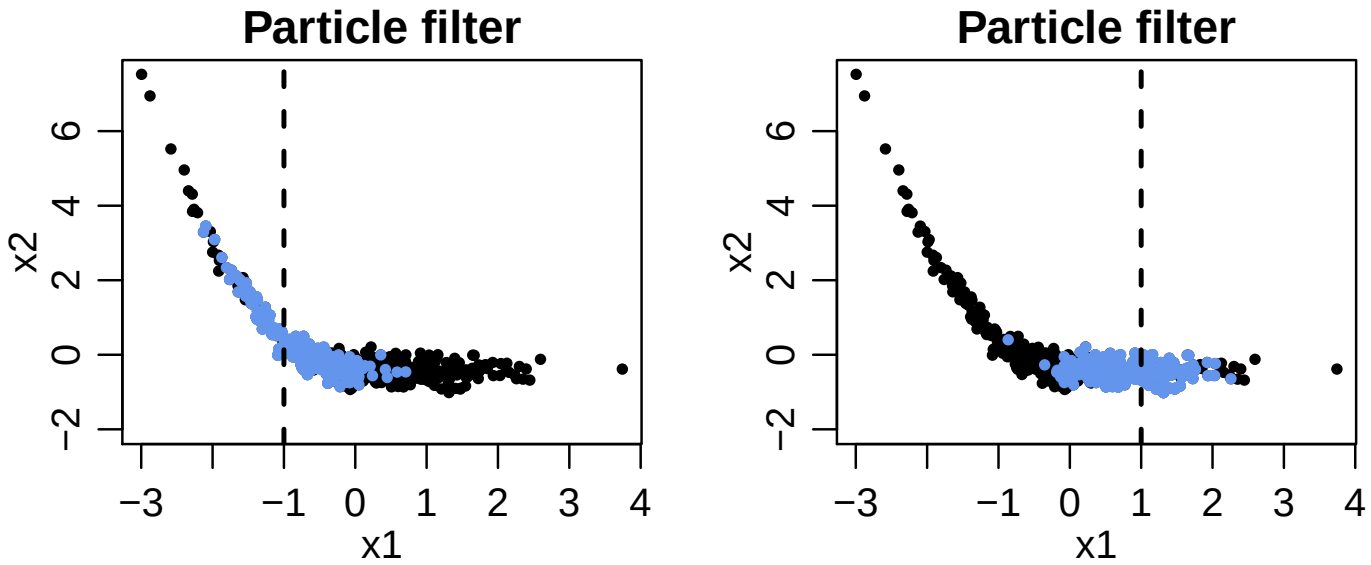
EnkF for banana-shaped prediction

Black: prediction ensemble (2-d). Observation $Y \sim \mathcal{N}(x_1, 0.5^2)$.
Blue: EnKF updates for two values $y = \pm 1$. Left: stochastic (perturbed observations), Right: deterministic (square-root).



Particle filter for banana-shaped prediction

Particle filter update for the same situation.



Shape and spread of true π^f depend here on y. The EnKF allows only the mean of π^f to depend on y.

Covariance tapering

The EnKF needs to estimate the prediction covariance P^p. The simple empirical covariance

$$\widehat{P}^p = \frac{1}{N-1}\sum_i (\boldsymbol{x}_i^p - \overline{\boldsymbol{x}^p})(\boldsymbol{x}_i^p - \overline{\boldsymbol{x}^p})^T = \frac{1}{N-1}(\Delta \boldsymbol{X}^p)(\Delta \boldsymbol{X}^P)^T$$

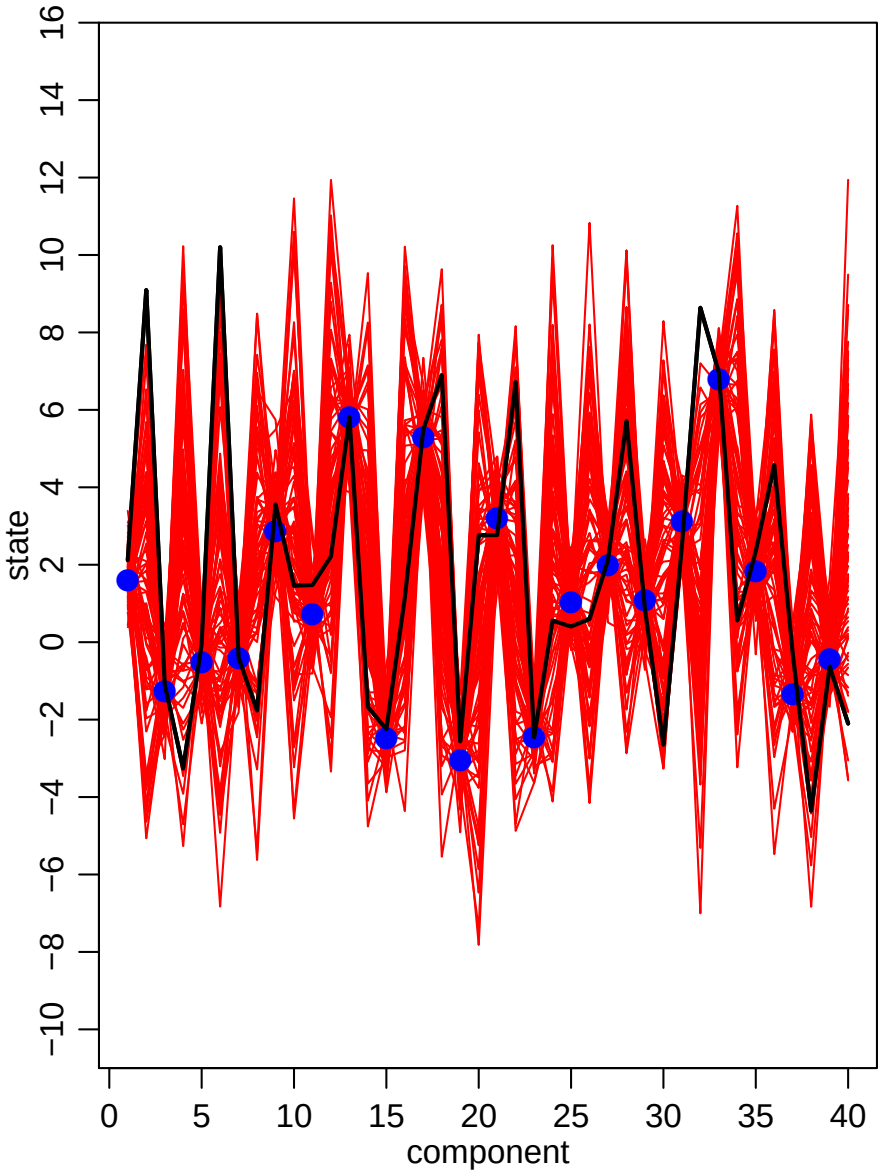
is unstable in typical examples and causes breakdown of the EnKF.

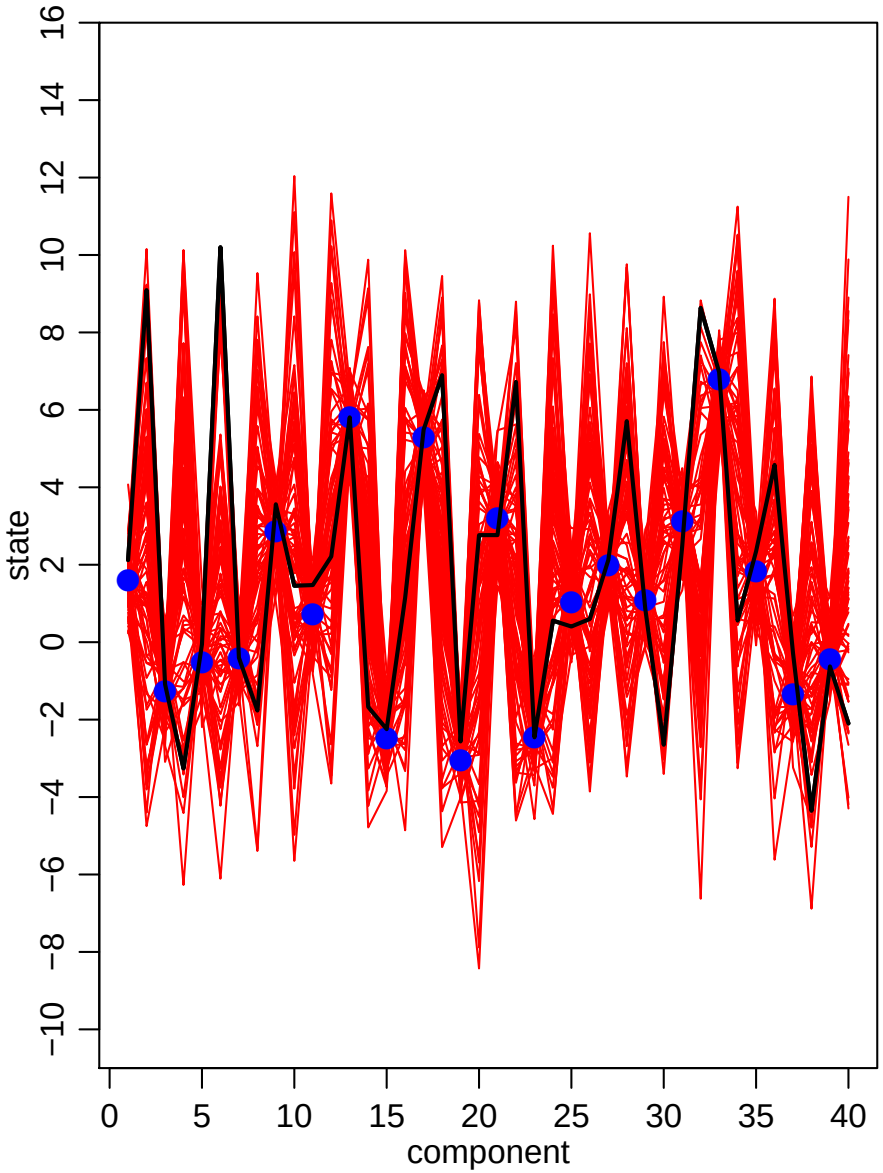
The standard way to regularize the estimate is to multiply $\widehat{P}^p$ elementwise by a banded correlation matrix ρ (called tapering). If components of $\boldsymbol{x}$ correspond to different spatial locations, this is justified by lack of correlation at large distances. If ρ is positive definite, then so is the regularized estimate.

Tapering the EnKF in the Lorenz 96 model

Filter ensemble (red) and truth (black) in cycles $1, 2, \ldots 20$.

Range of taper = 10 (left) and =5 (right).

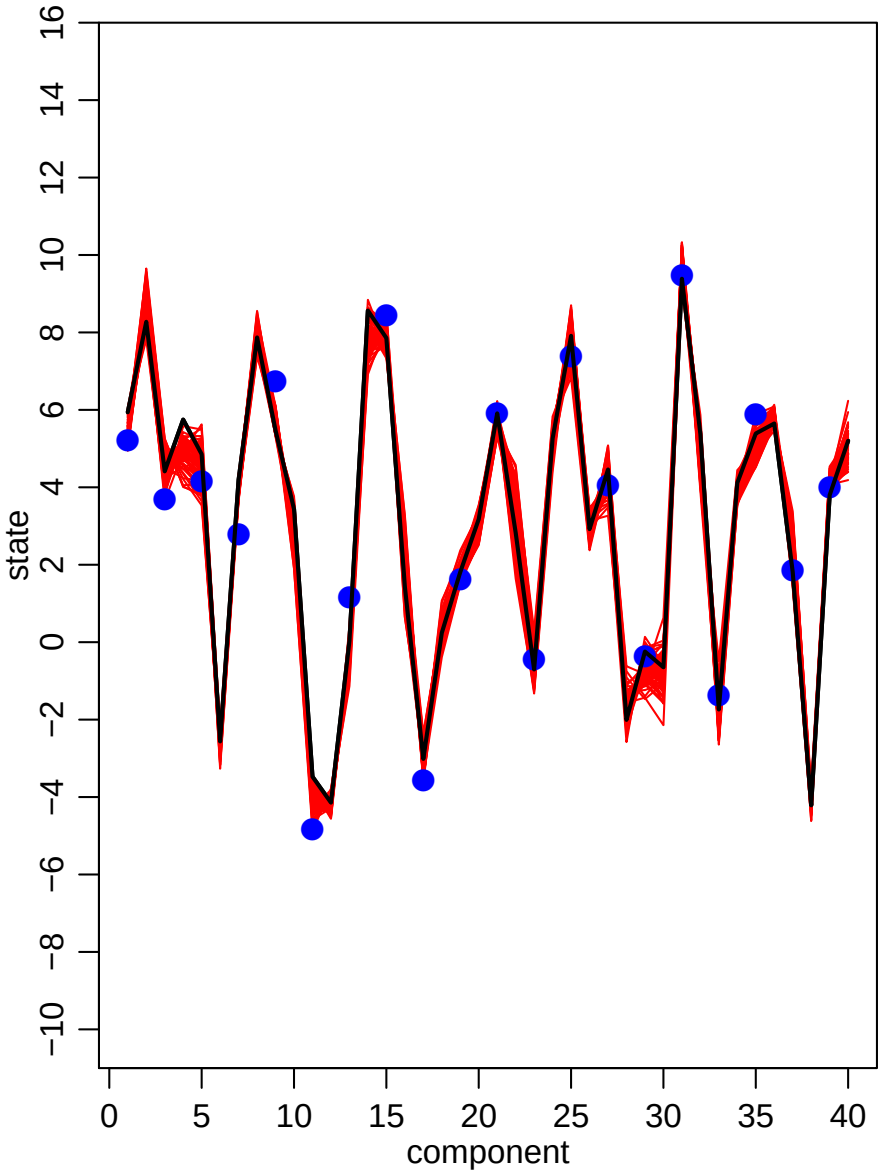


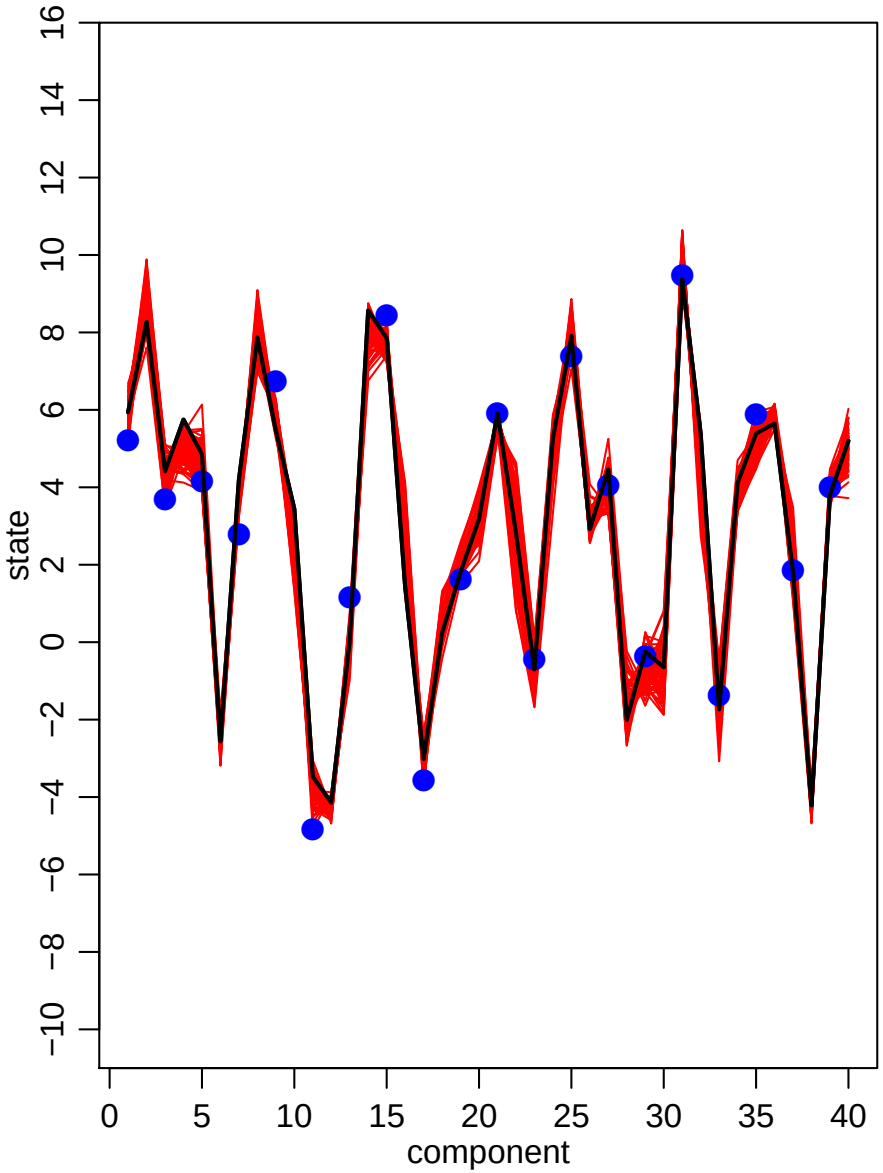


Tapering the EnKF in the Lorenz 96 model, ctd.

Filter ensemble (red) and truth (black) in cycles $21, 31, \ldots 201$.

Range of taper = 10 (left) and = 5 (right).





Breakdown of the particle filter

Bickel et al. (2008) give a theoretical explanation why the particle filter breaks down in high dimensions:

Typically, $\log h(\boldsymbol{y}|\boldsymbol{x}_j^p)$ is approximately normal with some mean $\mu_q = O(q)$ and standard deviation $\sigma_q = O(\sqrt{q})$. E.g. if q components of $\boldsymbol{x}$ are observed with i.i.d. normal observations errors:

$$\log h(\boldsymbol{y}|\boldsymbol{x}_j^p) = -\frac{1}{2\sigma^2}\sum_{\alpha=1}^{q}(y_\alpha - x_{\alpha,j}^p)^2$$

Hence the ratio of the two largest weights behaves like the ratio of the two largest values $(Z_{(N)}, Z_{(N-1)})$ in a sample of N lognormal random variables. By standard results from extreme value theory:

$$\frac{Z_{(N-1)}}{Z_{(N)}} \sim \exp\left(-\frac{\sigma_q}{\sqrt{2\log N}}E\right)$$

where E is a standard exponential random variable.

Hence the maximal weight converges to 1 if $N = o(\exp(q))$.

Reweighting and propagation in reverse order

If the transition density is known analytically, we can do the propagation step in closed form:

$$\begin{aligned} \pi_t^p(\boldsymbol{x}_t) &\approx \sum_j w_{t-1,j}^f \, m(\boldsymbol{x}_t|\boldsymbol{x}_{t-1,j}^f) \\ \pi_t^f(\boldsymbol{x}_t) &\propto \sum_j w_{t-1,j}^f \, \underbrace{m(\boldsymbol{x}_t|\boldsymbol{x}_{t-1,j}^f) h(\boldsymbol{y}_t|\boldsymbol{x}_t)}_{=p(\boldsymbol{x}_t,\boldsymbol{y}_t|\boldsymbol{x}_{t-1,j}^f)} \end{aligned}$$

(p is the generic symbol for a conditional density whose arguments indicate which random variables are involved.)

As $p(\boldsymbol{x}_t, \boldsymbol{y}_t|\boldsymbol{x}_{t-1,j}^f) = p(\boldsymbol{y}_t|\boldsymbol{x}_{t-1,j}^f) p(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{x}_{t-1,j}^f)$, we can also write

$$\pi_t^f(\boldsymbol{x}_t) \approx \sum_j \tilde{w}_{t,j}^f \, p(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{x}_{t-1,j}^f), \quad \tilde{w}_{t,j}^f \propto w_{t-1,j}^f \, p(\boldsymbol{y}_t|\boldsymbol{x}_{t-1,j}^f)$$

Sampling from this approximate filter density

$\sum_j \tilde{w}^f_{t,j}\, p(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{x}^f_{t-1,j})$ is a mixture. To sample from it, we can choose first a mixture component (with probabilities given by the weights). If component j is chosen, we propagate $\boldsymbol{x}^f_{t-1,j}$ using $p(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{x}^f_{t-1,j})$. That is we modify the true dynamics (given by $m(\boldsymbol{x}_t|\boldsymbol{x}^f_{t-1})$) in order to steer the particle towards the new observation.

Then $w^f_{t,j} = \frac{1}{N}$, i.e no weighting of filter particles needed. But choosing a mixture component is equivalent to resampling the filter particles at time $t-1$. The only difference to the standard particle filter is that weights are now proportional to $p(\boldsymbol{y}_t|\boldsymbol{x}^f_{t-1,j})$ instead of $p(\boldsymbol{y}_t|\boldsymbol{x}^p_{t,j})$. The gain is substantial only if the dynamics forgets initial condition quickly.

Exact sampling for nonlinear dynamics

Exact sampling is possible if $m(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(M(\boldsymbol{x}_{t-1}), Q)$ and if $h(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t; H\boldsymbol{x}_t, R)$. By the same computation as for the Kalman filter

$$p(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{x}^f_{t-1,j}) = \mathcal{N}(M(\boldsymbol{x}^f_{t-1,j}) + K_t(\boldsymbol{y}_t - HM(\boldsymbol{x}^f_{t-1,i})), Q - K_tHQ)$$

where K_t is the Kalman gain computed with Q as prediction covariance. Moreover

$$p(\boldsymbol{y}_t|\boldsymbol{x}^f_{t-1,j}) = \mathcal{N}(HM(\boldsymbol{x}^f_{t-1,j}), HQH^T + R).$$

The update looks like an EnKF update, but the variances differ, and there is a resampling step involved in selecting the particles at time $t-1$.

Proposal distributions for propagation

If $p(\boldsymbol{x}_t|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t)$ is not tractable, we can still steer the particle in the propagation step towards the new observation $\boldsymbol{y}_t$ using a "proposal distribution" $q(\boldsymbol{x}_t|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t)$ and correct by weighting:

$$\boldsymbol{x}^f_{t,j} \sim q(\boldsymbol{x}_t|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t), \quad w^f_{t,j} \propto \frac{m(\boldsymbol{x}^f_{t,j}|\boldsymbol{x}^f_{t-1,j})h(\boldsymbol{y}_t|\boldsymbol{x}^f_{t,j})}{q(\boldsymbol{x}^f_{t,j}|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t)}$$

In addition, we can use the new observations $\boldsymbol{y}_t$ also for selecting the starting particle (i.e. the mixture component): If we use probabilties $r_{t-1,j}(\boldsymbol{y}_t)$ instead of $w^f_{t-1,j}$ the weights at time t become

$$w^f_{t,j} \propto \frac{w^f_{t-1,j}m(\boldsymbol{x}^f_{t,j}|\boldsymbol{x}^f_{t-1,j})h(\boldsymbol{y}_t|\boldsymbol{x}^f_{t,j})}{r_{t-1,j}(\boldsymbol{y}_t)q(\boldsymbol{x}^f_{t,j}|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t)}$$

This idea is due to Pitt and Shephard who called it the "auxiliary particle filter" (the index of the mixture is an auxiliary variable).

Does the auxiliary PF solve the problem of filter collapse?

One can show that the "optimal" auxiliary particle filter (in the sense of minimzing the expected L_2-distance from uniform weights) takes $q(\boldsymbol{x}_t|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t) = p(\boldsymbol{x}_t|\boldsymbol{x}^f_{t-1,j}, \boldsymbol{y}_t)$ and then the weights are proportional to $p(\boldsymbol{y}_t|\boldsymbol{x}^f_{t-1,j})$.

This remains true even if we propagate $\boldsymbol{x}^f_{t-1,j}$ with a proposal that uses also the other particles $(\boldsymbol{x}^f_{t-1,i}; i \neq j)$

Van Leeuwen suggested to choose a target weight and then place the filter particles such that the weight of all or most particles are equal to the target weight. This is an appealing idea, but it means that the distribution of the filter particles differs from the target. It seems difficult to me to say anything about how large this error is.

Bridging the Particle and the Ensemble Kalman filter

Can we combine the EnKF with a PF in such a way that the method inherits robustness from the EnKF and the ability to deal with non-Gaussian features in π^p from the PF?

There are many proposals for such a combination. The EnKPF (Frei and K., 2013) applies Bayes formula in two steps

$$\pi^p(\boldsymbol{x}) \xrightarrow{\text{EnKF}} \pi^{f,\gamma}(\boldsymbol{x}) \propto \pi^p(\boldsymbol{x}) h(\boldsymbol{y}|\boldsymbol{x})^\gamma \xrightarrow{\text{PF}} \pi^f(\boldsymbol{x}) \propto \pi^{f,\gamma}(\boldsymbol{x}) h(\boldsymbol{y}|\boldsymbol{x})^{1-\gamma}.$$

This interpolates continuously between PF ($\gamma = 0$) and EnKF ($\gamma = 1$).

Implementing the EnKPF

Both steps of the EnKPF can be done exactly. For the first step we need to give another interpretation of the EnKF. Remember the update for the stochastic EnKF:

$$\boldsymbol{x}_i^f = \boldsymbol{x}_i^p + \widehat{K}(\boldsymbol{y} - H\boldsymbol{x}_i^p - \epsilon_i), \quad \epsilon_i \sim \mathcal{N}(0, R)$$

This means that $\boldsymbol{x}_i^f \sim \mathcal{N}(\boldsymbol{x}_i^p + \widehat{K}(\boldsymbol{y} - H\boldsymbol{x}_i^p), \widehat{K}R\widehat{K}^T)$, i.e. the filter ensemble is a balanced sample from

$$\pi^{f,EnKF} = \frac{1}{N}\sum_{j=1}^{N} \mathcal{N}(\boldsymbol{x}_i^p + \widehat{K}(\boldsymbol{y} - H\boldsymbol{x}_i^p), \widehat{K}R\widehat{K}^T)$$

If we do a partial update with $h(\boldsymbol{y}|\boldsymbol{x})^\gamma \propto \mathcal{N}(H\boldsymbol{x}, R/\gamma)$, the same formula holds for $\pi^{f,\gamma}$ provided we compute the Kalman gain with R/γ (i.e. with less weight on the observation).

Implementing the EnKPF, ctd.

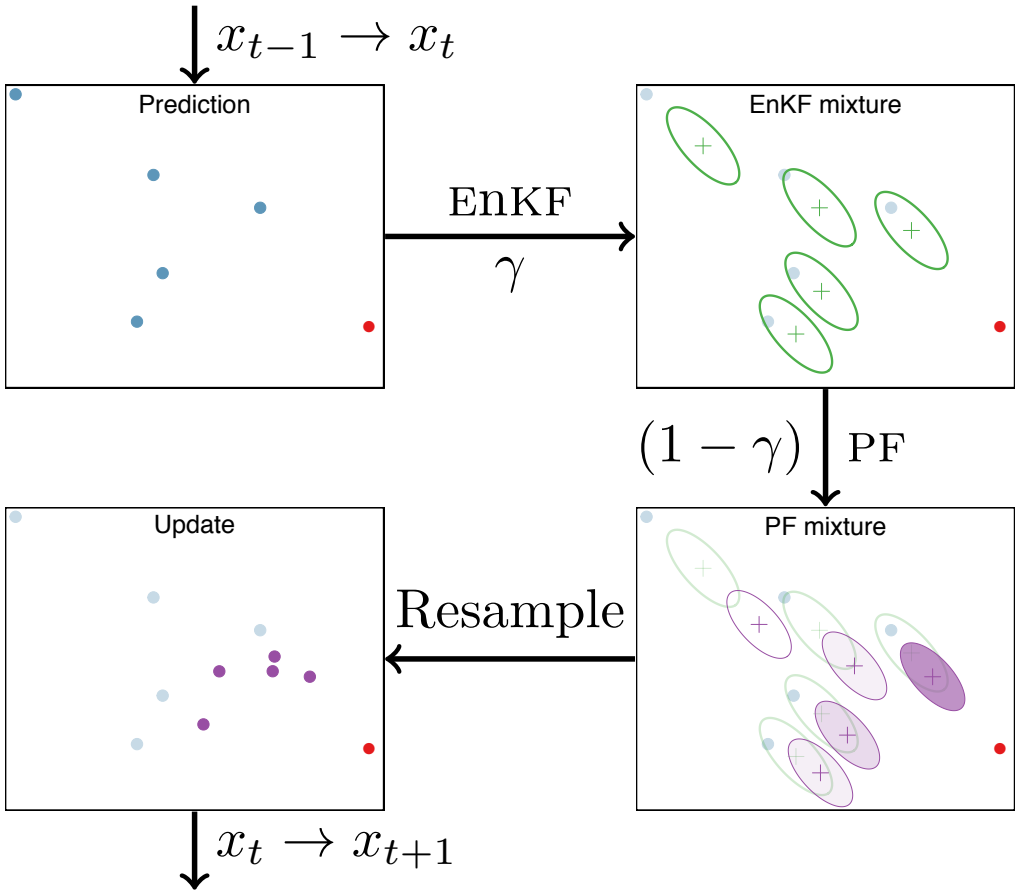
Using Bayes formula, one can show that the second step gives a another Gaussian mixture:

$$\pi^f = \sum_{j=1}^{N} w^{\gamma,j} \mathcal{N}(\mu^{\gamma,j}, \hat{P}^\gamma)$$

(I skip the formulae for $w^{\gamma,j}$, $\mu^{\gamma,j}$, $\hat{P}^\gamma$)

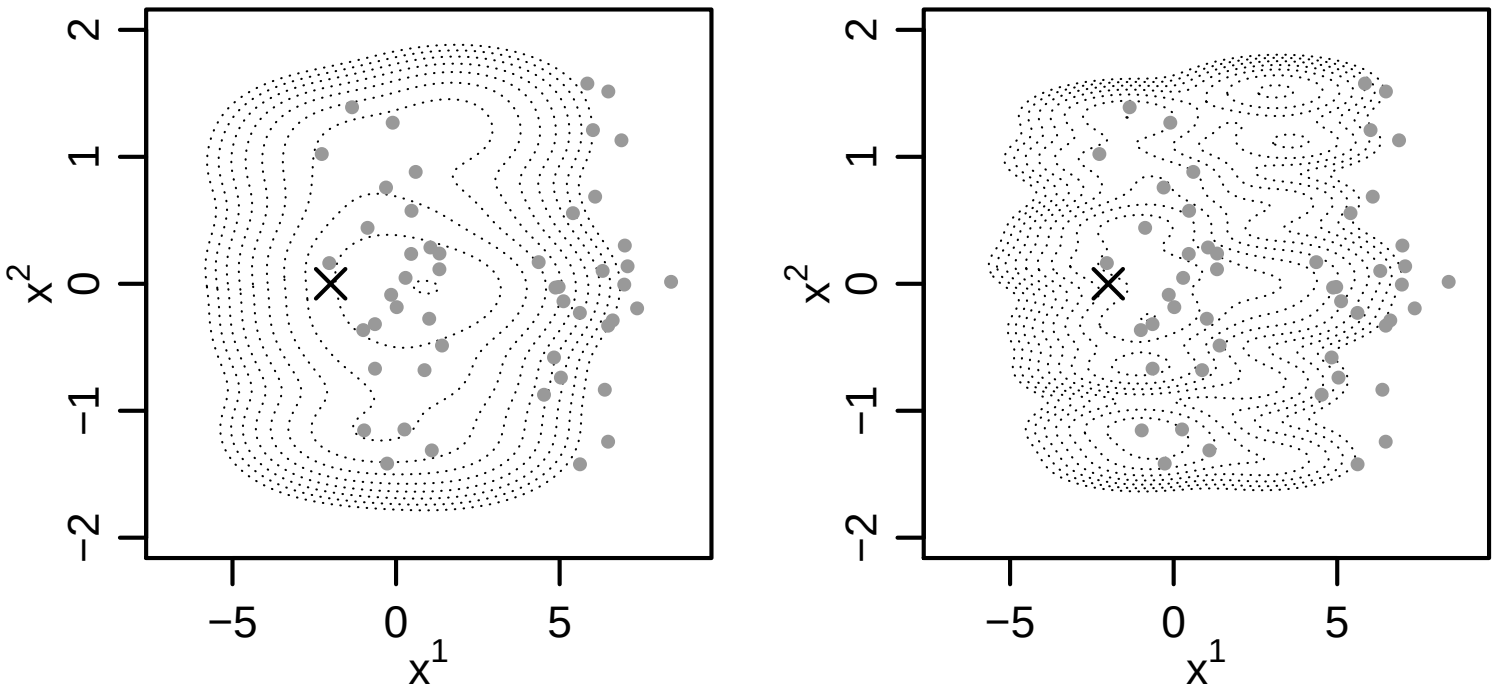
We obtain the filter ensemble by drawing from this mixture. Choosing the mixture component is like resampling, so the same mixture component can be chosen several times. But even then no two filter particles are the same. γ should be such that sufficiently many mixture components are chosen at least once.

Illustration of EnKPF



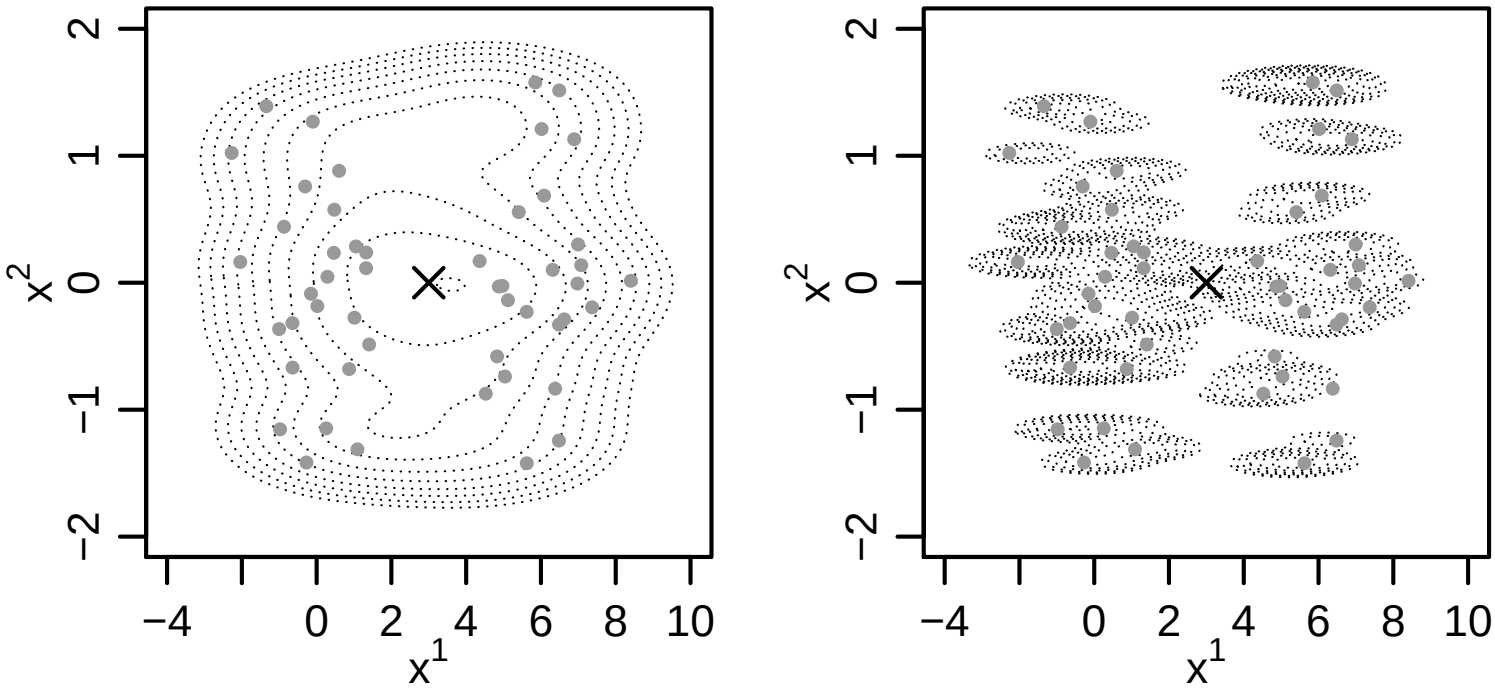
Single update for bimodal prior I

Left: EnKF. Right: EnKPF with γ s. th. diversity $\approx 40\%$.
Dots: Prior sample, Dotted: Contours of the Gaussian mixture from which the filter ensemble will be drawn.



Single update for bimodal prior II

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



Localization of filter updates

In many applications, components of $\boldsymbol{x}$ and $\boldsymbol{y}$ are related to positions in space. Then assume that H is local (its entries are zero unless locations are close) and that R is diagonal.

Then intuitively, the update of each component of $\boldsymbol{x}$ should depend only on components of $\boldsymbol{y}$ that are close, and each component of $\boldsymbol{y}$ should influence only the update of components of $\boldsymbol{x}$ close-by. This is also necessary for efficient computation (parallelization).

This is easier to achieve for EnKF than for PF. I discuss EnKF first.

Localization of EnKF

Covariance tapering does not imply that updates are local. Even if $HP^pH^T + R$ is sparse, its inverse is usually dense. Then also the Kalman gain is dense.

Localization can be enforced by updating each component of $\boldsymbol{x}$ once, using only observations nearby, or by assimilating each component of $\boldsymbol{y}$ once, updating only components of $\boldsymbol{y}$ nearby. In practice, both methods seem to give similar results. The first method is prefered since the available observations can change at each iteration.

For both methods, care is needed to ensure that the update does not create artificial discontinuities in the filter particles. This can be done.

Localization of PF

For the PF (and any other method that uses resampling), the problem of artificial discontinuities in the filter particles is more severe: Unless the resampling probabilities are the same at all locations (which contradicts localization), there are locations next to each other where at least one filter particle originates from different prediction particles.

You can reduce this problem to some extent by permuting the indices of the particles and by coupling the resampling at neighboring locations.

For the EnKPF we have found a good solution (we believe) for enforcing locality, but only when we use each observation component once.

Filtering in the setup used by MeteoSwiss

Sylvain Robert and I tested a localized EnKPF in a high resolution numerical weather prediction model (COSMO 2). This is a particular challenge since the complexity of the code limits the methods we can use for filtering.

We don't have access to the whole vectors $\boldsymbol{y}$ and $\boldsymbol{x}_i^p$. The system updates the state variables in non-overlapping blocks and the result of the assimilation must be in the form of a $N \times N$ matrix W (where $N = 40$) for each block such that

$$\boldsymbol{X}^f = \boldsymbol{X}^p W$$

Here $\boldsymbol{X}^p = (\boldsymbol{x}_1^p | \boldsymbol{x}_2^p | ... | \boldsymbol{x}_N^p)$ where each $\boldsymbol{x}_j^p$ contains the components of the state in that block. At the center of the block, the particles $\boldsymbol{x}_j^f$ are used whereas for other grid points the weight matrices are interpolated and then used for updating.

Input for the update

In order to compute the a matrix W, we are given

$$(H\Delta \boldsymbol{X}^p)^T R^{-1}(\boldsymbol{y} - H\bar{\boldsymbol{x}}^p)$$

where $\Delta \boldsymbol{X}^p = (\boldsymbol{x}_1^p - \bar{\boldsymbol{x}}^p | ... | \boldsymbol{x}_N^p - \bar{\boldsymbol{x}}^p)$ and the spectral decomposition of the matrix

$$(H\Delta \boldsymbol{X}^p)^T R^{-1}(H\Delta \boldsymbol{x}^p).$$

First we had to adapt the EnKPF so that it can be computed with these inputs.

Illustration of EnKF update at one particular time

The plot shows the values W_{11} and W_{12} in the update formula

$$\boldsymbol{x}_1^f - \bar{\boldsymbol{x}}^f = W_{11}(\boldsymbol{x}_1^p - \bar{\boldsymbol{x}}^p) + W_{12}(\boldsymbol{x}_2^p - \bar{\boldsymbol{x}}^p) + ...$$

for all locations.

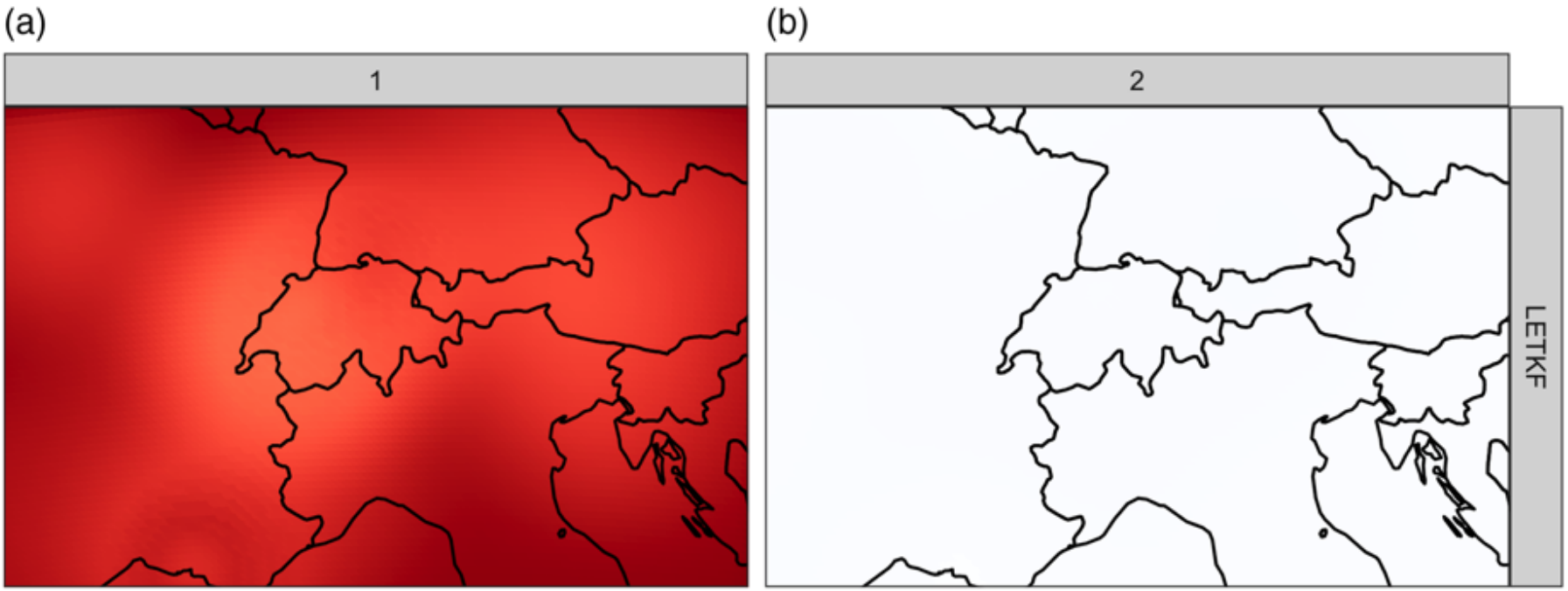
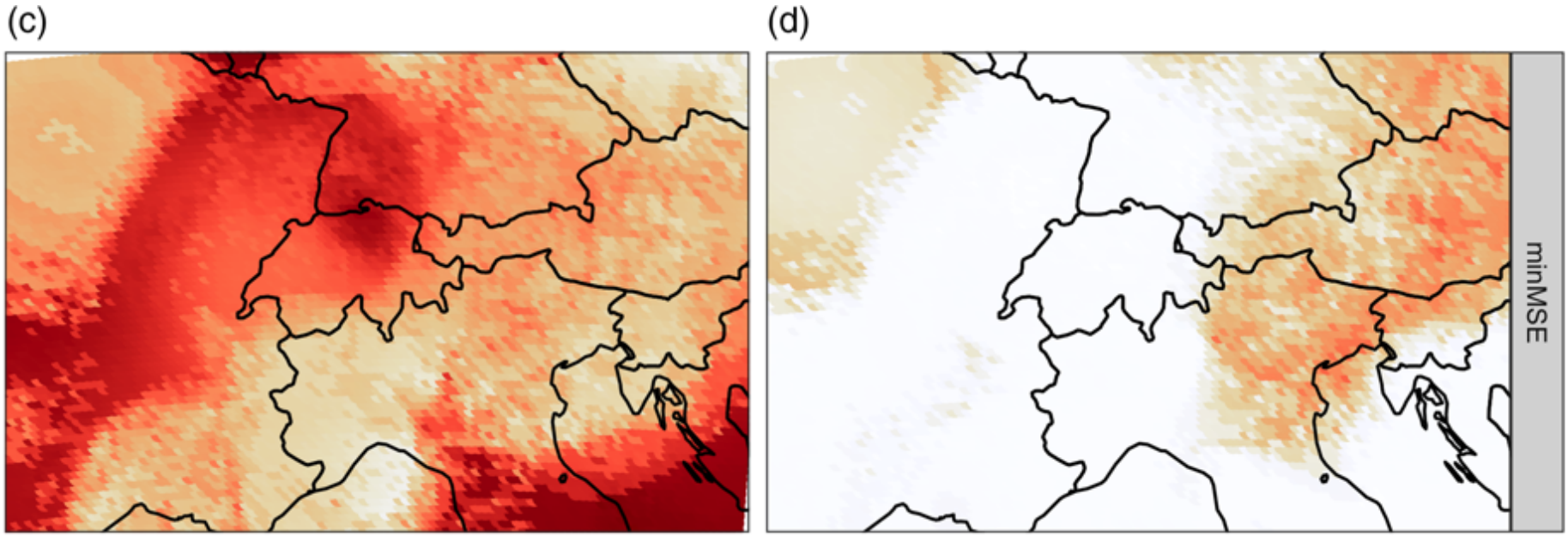


Illustration of EnKPF update at the same time

Same plot for the EnKPF:



Overview of smoothing

Smoothing means to estimate the state at some/all times before (or equal to) t, using all observations up to time t. I will discuss ensemble methods where the distribution of $\boldsymbol{X}_{u:s}$ given $\boldsymbol{Y}_{1:t} = \boldsymbol{y}_{1:t}$ is approximated by a (possibbly weighted) sample $(x^t_{s:u,1}, ..., x^t_{s:u,N})$ for $0 \leq s \leq u \leq t$. So the superscript tells which observations are available, subscripts give time points of states to be estimated and the particle number.

There are 3 main cases:
Marginal smoothing: $s = u < t$;
Fixed lag smoothing: $s = t - L, u = t$;
Full smoothing:$s = 0, u = t$.

Full smoothing also solves the other two problems, but the other two might be easier to do. There are many methods, I will only mention the simplest ones.

Full smoothing by extending the state space

Define an extended state at time t by $\boldsymbol{Z}_t = \boldsymbol{X}_{0:t}$. $(\boldsymbol{Z}_t, \boldsymbol{Y}_t)$ is still a state space model: For the dynamics, we generate $\boldsymbol{X}_t$ by the original model and attach it to $\boldsymbol{Z}_t$, and the observation $\boldsymbol{Y}_t$ depends only on the last component of $\boldsymbol{Z}_t$.

Applying the particle filter to this new state gives the following algorithm:

- Propagate: $\boldsymbol{x}^t_{0:t-1,i} = \boldsymbol{x}^{t-1}_{0:t-1,i}$, $\boldsymbol{x}^t_{t,i} \sim m(\boldsymbol{x}_t | \boldsymbol{x}^{t-1}_{t-1,i})$.
- Reweight: $w^t_i \propto h(\boldsymbol{y}_t | \boldsymbol{x}^t_{t,i})$.
- Resample: Transform the weighted sample $(\boldsymbol{x}^t_{0:t,i}, w^t_i)$ into an unweighted sample.

Problems: Storage and sample depletion. The number of different particles $\boldsymbol{x}^t_{s,i}$ at a fixed time s can only decrease if t increases. It can work well for fixed lag smoothing when the particle filter works.

Forward filtering, backward smoothing

This is based on the fact, that conditionally on $\boldsymbol{Y}_{1:t}$, $\boldsymbol{X}_{0:t}$ is a time inhomogeneous Markov chain with backward transition densities

$$p(\boldsymbol{x}_s|\boldsymbol{x}_{s+1}, \boldsymbol{y}_{1:t}) = p(\boldsymbol{x}_s|\boldsymbol{x}_{s+1}, \boldsymbol{y}_{1:s}) \propto m(\boldsymbol{x}_{s+1}|\boldsymbol{x}_s)\pi_s^f(\boldsymbol{x}_s)$$

Hence if we have stored the particle filter approximations, we can draw one full smoothing particle by the following algorithm:

- Choose $\boldsymbol{x}_t^t = \boldsymbol{x}_{t,j}^f$ with probability $w_{t,j}^f$
- For $s = t-1, t-2, ..., 0$ choose $\boldsymbol{x}_s^t = \boldsymbol{x}_{s,j}^f$ with probability proportional to $m(\boldsymbol{x}_{s+1}^t|\boldsymbol{x}_{s,j}^f)w_{s,j}^f$

One draw from the full smoothing density has complexity $O(tN)$ because in the second step we need to normalize the weights. Hence constructing a smoothing ensemble of size N has complexity $O(tN^2)$.

Marginal forward-backward smoothing

The result from the previous slide implies

$$p(\boldsymbol{x}_s|\boldsymbol{y}_{1:t}) = \int \underbrace{p(\boldsymbol{x}_s|\boldsymbol{x}_{s+1}, \boldsymbol{y}_{1:s})}_{\propto m(\boldsymbol{x}_{s+1}|\boldsymbol{x}_s)\pi_s^f(\boldsymbol{x}_s)} p(\boldsymbol{x}_{s+1}|y_{1:t}) d\boldsymbol{x}_{s+1}$$

Hence we can obtain a weighted marginal smoothing ensemble that has the same particles as the filter ensemble, but different weights:

$$w_{t,i}^t = w_{t,i}^f; \quad w_{s,i}^t = \sum_{j=1}^N w_{s+1,j}^t \frac{m(\boldsymbol{x}_{s+1,j}^f|\boldsymbol{x}_{s,i}^f) w_{s,i}^f}{\sum_k m(\boldsymbol{x}_{s+1,j}^f|\boldsymbol{x}_{s,k}^f) w_{s,k}^f}$$

This avoids the Monte Carlo error in the backward step without increasing the complexity.

Other smoothers

A different particle smoother is based on the so-called two-filter formula. It combines two particle filters, one forward and one backward in time.

Both forward-backward and two-filter smoothers can be modified to have $O(tN)$ complexity, see the references in the review by Paul Fearnhead and myself.

Finally, there are Ensemble Kalman smoothers based on either extending the state or on forward-backward Kalman filter recursions.

Parameter estimation: Basics

We discuss the estimation of static parameters θ (λ in Remus' notation) that are present in the state density m and/or the observation density h.

The simplest methods include θ in the state, but this often does not use the information in the data efficiently. It has however the advantage that it is easily updated if new data arrive.

When parameter are estimated once all data are available, then statisticians prefer to use either maximum likelihood or Bayesian methods.

Maximum likelihood and Bayesian estimation

The maximum likelihood estimator is defined as

$$\arg\max_{\theta} p_\theta(\boldsymbol{y}_{1:t})$$

Bayesian methods put a prior p_0 on θ and draw samples from the posterior

$$p(\theta|\boldsymbol{y}_{1:T}) \propto p_0(\theta)p_\theta(\boldsymbol{y}_{1:T})$$

The basic difficulty for both methods is that the likelihood $p_\theta(\boldsymbol{y}_{1:t})$ is intractable (it is a high-dimensional integral). Maximum likelihood applies stochastic versions of the EM (Expectation-Maximization)-algorithm, Bayesian methods so-called Particle MCMC (Markov chain Monte Carlo).

Parameter uncertainty in filtering

Bayesian methods have the advantage that uncertainty about the parameter can be taken into account in filtering or prediction by integrating over the unknown parameter:

$$\pi_t^f = \int p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \theta) p(\theta|\boldsymbol{y}_{1:t}) d\theta = \int p(\boldsymbol{x}_t, \theta|\boldsymbol{y}_{1:t}) d\theta$$

and similarly for π_t^p. If one samples jointly the state and the parameter, this integral is trivial.

Parameters included in the state: PF

The easiest method is to include θ as a deterministic component of the state. Then we initialize the filter with $\theta_0^{f,j} \sim p_0(\theta)d\theta$ and the update becomes

$$\theta_{t,j}^p = \theta_{t-1,j}^f, \quad \boldsymbol{x}_{t,j}^p \sim m_{\theta_{t-1,j}^f}(\boldsymbol{x}|\boldsymbol{x}_{t-1,j}^f)$$

The particle filter degenerates quickly because the diversity of the θ-component cannot be recovered in the propagation step. One can avoid this by adding noise to $\theta_{t,j}^f$, preferably combined with shrinkage to the mean. But variance of the noise must go to zero in order that $(\theta_t^{f,j})$ approximates the posterior $p(\theta|\boldsymbol{y}_{1:t})$.

Parameters included in the state: EnKF

For parameters θ in the transition density m, the EnKF obtains information about θ through correlations of θ and the state in the prediction distribution. This can be weak compared to information in the likelihood.

For measurement equations of the form $y = H(x, \theta) + \mathcal{N}(0, R)$ with known R, information is obtained through correlations of θ with $H(x, \theta)$.

When the error covariance R depends on θ, a modification of the EnKF is needed (Frei and K., 2013).

The likelihood $p_\theta(\boldsymbol{y}_{1:t})$

The joint density of both the states $\boldsymbol{x}_{0:t}$ and the observations $\boldsymbol{y}_{1:t}$ available can be written down:

$$p_\theta(\boldsymbol{x}_{0:t}, \boldsymbol{y}_{1:t}) = p_\theta(\boldsymbol{x}_0) \prod_{s=1}^{t} m_\theta(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}) g_\theta(\boldsymbol{y}_s|\boldsymbol{x}_s)$$

The joint density of the observations alone is obtained by integrating the states out:

$$p_\theta(\boldsymbol{y}_{1:t}) = \int p_\theta(\boldsymbol{x}_{0:t}, \boldsymbol{y}_{1:t}) d\boldsymbol{x}_{0:t}$$

The integral can be approximated by a 2nd-order Taylor approximatio of $\log p_\theta(\boldsymbol{x}_{0:t}, \boldsymbol{y}_{1:t})$ around its maximum. We chose this approach in the project on the abundance of fish mentioned at the beginning.

The likelihood $p_\theta(\boldsymbol{y}_{1:t})$, ctd.

A different approach starts with the identity

$$p_\theta(\boldsymbol{y}_{1:t}) = \prod_{s=1}^{t} p_\theta(\boldsymbol{y}_s|\boldsymbol{y}_{1:s-1})$$

The factors on the right are the normalizing constants in Bayes formula for the update step

$$\pi^f_{\theta,s}(\boldsymbol{x}_s) = \frac{\pi^p_{\theta,s}(\boldsymbol{x}_s)h_\theta(\boldsymbol{y}_s|\boldsymbol{x}_s)}{p_\theta(\boldsymbol{y}_s|\boldsymbol{y}_{1:s-1})}$$

It can be estimated by running a particle filter with parameter θ and computing the average of the weights $h_\theta(\boldsymbol{y}_s|\boldsymbol{x}^p_{s,i})$.

A basic result shows that this estimate is unbiased for every fixed $\boldsymbol{y}_{1:T}$ and every fixed N, i.e. on average over all possible particles the estimated likelihood is equal to the true likelihood.

Particle MCMC for state space models

The standard Metropolis-Hastings algorithm to sample from the posterior $p(\theta|\boldsymbol{y}_{1:T})$ runs as follows: Given the current value θ, propose a new value

$$\theta' \sim q(\theta'|\theta)d\theta'$$

and accept it with probability

$$a(\theta, \theta') = \min\left(1, \frac{p_0(\theta')p_{\theta'}(\boldsymbol{y}_{1:T})q(\theta|\theta')}{p_0(\theta)p_\theta(\boldsymbol{y}_{1:T})q(\theta'|\theta)}\right)$$

Otherwise keep the current value θ.

As the likelihood is not available, one runs a particle filter with parameter θ', computes the unbiased estimate from the previous slide and plugs it into the formula for the acceptance probability.

Andrieu & Roberts (2009), Andrieu et al. (2010) have shown that the stationary distribution of this Markov chain is still the exact posterior.

Summary and Conclusion

- State space models provide a unified framework for state prediction and filtering in complex systems.
- In many applications, the only way to approximate prediction and filtering distributions is by Monte Carlo.
- Monte Carlo methods iterate between propagation and updating. The update step is more difficult, with particle filter and ensemble Kalman filter as the two basic methods.
- Ensemble Kalman filter works well also in high dimensions, provided we localize the update.
- Particle filters are more general, but they degenerate quickly in high dimensions. Modifications to improve the performance of particle filters and hybrid methods have been proposed, but it is still open whether they are able to beat the EnKF in geophysical applications.
- Smoothing algorithms and estimation of static parameters are other areas of active research.

Thank you for your attention!