Minitutorial
Particle and Ensemble Kalman Filters for Data Assimilation and Time Series Analysis

Hans R. Künsch

Seminar for Statistics
ETH Zurich

SIAM-UQ16, Lausanne, April 6

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

Overview I

- State space models 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* given all observations up to time *t* and the quantification of its uncertainty, ideally in the form of a probability distribution. 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 Filtering also provides methods to estimate such parameters
- These topics appear in many talks at this conference

Overview II

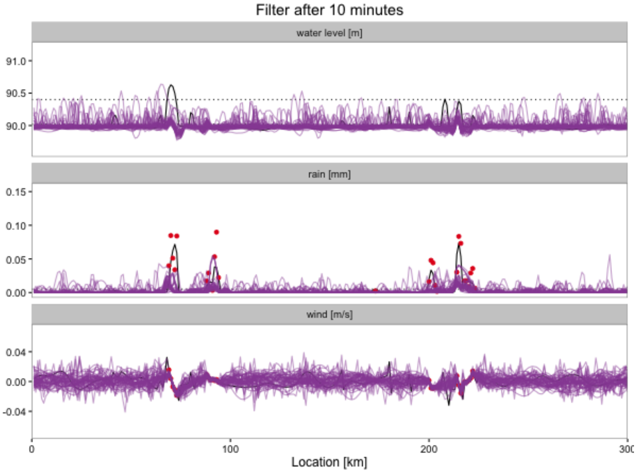
- In this tutorial, I want to introduce the basic concepts for non-specialists and give an outlook on some new and ongoing research
- Statisticians, geophysicists and applied mathematicians have made contributions, often without much exchange of ideas and methods
- It is not possible to cover everything in 2 hours, the selection is my own
- The emphasis here is on the derivation and heuristic properties of algorithms. I do not go into results about asymptotic performance of the algorithms since I think it is often not clear how to do asymptotics which is relevant
- I will take questions in between at the end of each chapter, please don't hesitate to ask

A few success stories

- Particle filters have been extremely successful in tracking problems of image analysis because they can deal with occasional ambiguity
- A version of the Ensemble Kalman filter, called the Local Ensemble Transform Kalman filter is used in operational weather forecasting
- Importance splitting, a method for rare event simulation, can be considered as a particle filter
- Problems with unknown static parameters are much harder. The examples used in papers on particle MCMC are of intermediate complexity, e.g. stochastic volatility or models for GDP with different regimes
- ...

A first example: Cumulus convection

This is a toy model I will return to later. For now, this is just an example for the methods that I will present



Contents

1. Introduction
2. Basics about State Space Models and Filtering
   - State space models
   - Prediction, filtering, data assimilation
   - Monte Carlo filters
3. Particle filters Part I
   - Resampling
4. Ensemble Kalman filters
   - Bayes formula for Gaussian prior and likelihood
   - Perturbed observation EnKF
   - Square root filters
   - Localization of EnKF
5. The Ensemble Kalman Particle Filter
6. Particle Filters, Additional Topics
   - Auxiliary particle filter
   - Particle smoothing
   - General sequential Monte Carlo
7. Parameter estimation

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, Particle filters. Bernoulli 19 (2013).
- K. Law, A. Stuart, K. Zygalakis. Data Assimilation: A Mathematical Introduction. Springer 2015.
- A. Majda, J. Harlim. Filtering Complex Turbulent Systems. Cambridge 2012.

1. **Introduction**

2. **Basics about State Space Models and Filtering**
   - State space models
   - Prediction, filtering, data assimilation
   - Monte Carlo filters

3. **Particle filters Part I**

4. **Ensemble Kalman filters**

5. **The Ensemble Kalman Particle Filter**

6. **Particle Filters, Additional Topics**

7. **Parameter estimation**

State space models

A state space model consists of a dynamical system (X_t) and partial and noisy observations (Y_i) of the state of the system at some discrete time points t_i.

(X_t) contains a complete description of the system and is not fully observable. Its dynamics is given by a differential equation or a Markov process in discrete or continuous time.

Observations Y_i are conditionally independent given the state, and Y_i depends only on X_{t_i}.

Graphical representation of state space models

The dependence between the variables of a state space model can be represented by the following directed acyclic graph

$$\begin{tikzcd}
\dots \arrow[r] & X_{t_{i-1}} \arrow[r] \arrow[d] & X_{t_i} \arrow[r] \arrow[d] & X_{t_{i+1}} \arrow[r] \arrow[d] & \dots \\
\dots & Y_{i-1} & Y_i & Y_{i+1} & \dots
\end{tikzcd}$$

From this graph, conditional independence relations can be deduced, by looking at separation properties after the arrows have been dropped.

In particular, (Y_i) is not a Markov chain, but Y_i is independent of its past given X_{t_i} or $X_{t_{i-1}}$.

Notation and simplifications

X_t takes values in $\mathbb{R}^d$, Y_i takes values in $\mathbb{R}^q$. Distributions are time homogeneous and observation times are $t_i = i$. I write t instead of t_i and use $y_{1:t}$ as shorthand for $(y_1, y_2, \ldots, y_t)$.

Transition kernel of (X_t):

$$M(dx|x') = \mathbb{P}(X_t \in dx|X_{t-1} = x')$$

In the deterministic case, this is a point mass at the value of the solution at time 1 with initial condition x'. Starting distribution at time zero $X_0 \sim M_0(dx)$.

Conditional distribution of observations

$$Y_t|X_t = x \sim g(y|x)dy$$

(existence of a density required, but reference measure arbitrary).

Often p is used as a generic symbol of the density of the variables indicated by its argument.

Basics of data assimilation/filtering

Predicting the state at time $t+1$ based on all observations up to time t is done in two steps: First estimate the state at time t and then use the dynamics of the state given the value at time t. For the first step, combine prediction for time t (computed at time $t-1$) with observations at time t.

In order to quantify uncertainty, consider conditional distributions.
Filter $\pi_t^f =$ conditional distribution of X_t given $Y_{1:t} = y_{1:t}$.
Prediction $\pi_t^p =$ conditional distribution of X_t given $Y_{1:t-1} = y_{1:t-1}$.

In data assimilation, the prediction is sometimes called the background and the filter the analysis.

The observations $y_{1:t}$ are considered fixed and thus usually dropped in the notation.

Recursions for prediction and filter

Prediction follows from the filter one time step earlier

$$\pi_t^p(dx_t) = \int \pi_{t-1}^f(dx_{t-1}) M(dx_t | x_{t-1})$$

Conversely, the filter can be computed from the prediction at the same time, using Bayes' formula and $p(y_t | x_t, y_{1:t-1}) = g(y_t | x_t)$:

$$\pi_t^f(dx_t) = \frac{\pi_t^p(dx_t) g(y_t | x_t)}{p(y_t | y_{1:t-1})} \propto \pi_t^p(dx_t) g(y_t | x_t)$$

(Remember that $\pi^f(dx_t) = \pi^f(dx_t | y_{1:t})$ and $\pi^p(dx_t) = \pi^p(dx_t | y_{1:t-1})$). Prediction is used as the prior, it contains all the relevant information from earlier observations $y_{1:t-1}$.

These two steps are applied recursively. They are called propagation and update.

Monte Carlo 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).

Moreover, the transition distribution $M(dx|x')$ is often not in closed form, but for given starting point x' one can draw from it. In the case of differential equations, this means computing the solution at time 1, starting at x'.

Monte Carlo filters approximate π_t^p and π_t^f by samples or ensembles of weighted "particles" $(x_t^{p,j}, \alpha_t^{p,j})$ and $(x_t^{f,j}, \alpha_t^{f,j})$ $(j = 1, 2, \ldots, N)$. I.e. for any (bounded) $\psi : \mathbb{R}^d \to \mathbb{R}$

$$\mathbb{E}\left[\psi(X_t)|y_{1:t-1}\right] = \int \psi(x)\pi_t^p(dx) \approx \sum_{j=1}^{N} \psi(x_t^{p,j})\alpha_t^{p,j}$$

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. They become the next prediction particles. Weights do not change:

$$x_t^{p,j} \sim M(dx|x_{t-1}^{f,j}), \quad \alpha_t^{p,j} = \alpha_{t-1}^{f,j}$$

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 particles. The two main methods are the Particle Filter (PF) and the Ensemble Kalman Filter (EnKF).

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 g
- 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 $\pi_{t|t-1}$ is Gaussian. However, it is extremely robust in practice

1. **Introduction**
2. **Basics about State Space Models and Filtering**
3. **Particle filters Part I**
    - Resampling
4. **Ensemble Kalman filters**
5. **The Ensemble Kalman Particle Filter**
6. **Particle Filters, Additional Topics**
7. **Parameter estimation**

Particle filter update: Reweighting

We consider only one step of the recursion and drop the time index t.

Sequential importance sampling sets $x^{f,j} = x^{p,j}$ and updates only the weights

$$\alpha^{f,j} \propto \alpha^{p,j} g(y|x^{p,j})$$

Particles with a good fit to the new observation have a high weight.

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

Particle filter update: Resampling

Basic remedy to counteract weight unbalance is resampling:
Take an unweighted filter sample containing the j-th prediction particle $x^{p,j}$ N_j times where

$$\mathbb{E}\left[N_j\right] = N\,\alpha^{f,j}, \quad \sum_j N_j = N$$

Particles with a poor fit to the new observation die, those with an excellent fit have children.

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 of the next prediction. Resampling is only a partial remedy.

Resampling and effective sample size

Resampling also introduces an additional Monte Carlo error because $\alpha^{f,j}$ is replaced by N_j/N. To reduce this, resample only if diversity of weights is low, as measured by effective sample size ESS:

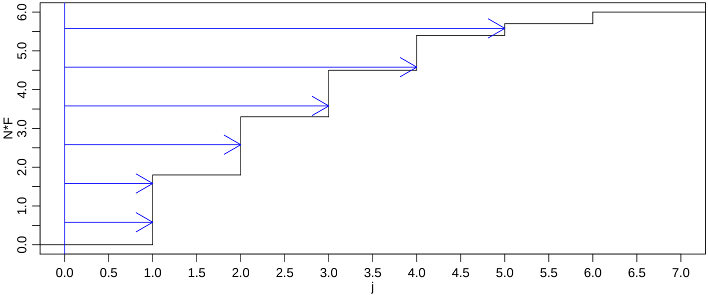
$$\mathrm{ESS} = \left(\sum_{j=1}^{N} (\alpha^{f,j})^2\right)^{-1}.$$

$\mathrm{ESS} = 1$ if one $\alpha^{(f,j)} = 1$, $\mathrm{ESS} = N$ if all $\alpha^{(f,j)} = 1/N$. 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 $|N_j - N\alpha^{f,j}| < 1$.

There are many balanced sampling schemes. The simplest one is illustrated in the following figure. Steps have height $N\alpha^{f,j}$

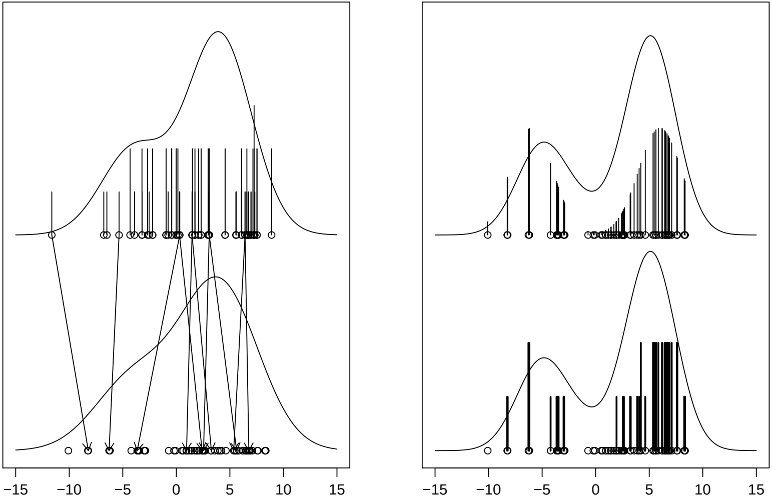


This is hard to analyze theoretically because no limit theory applies. So-called tree sampling is an alternative.

A single step of the particle filter

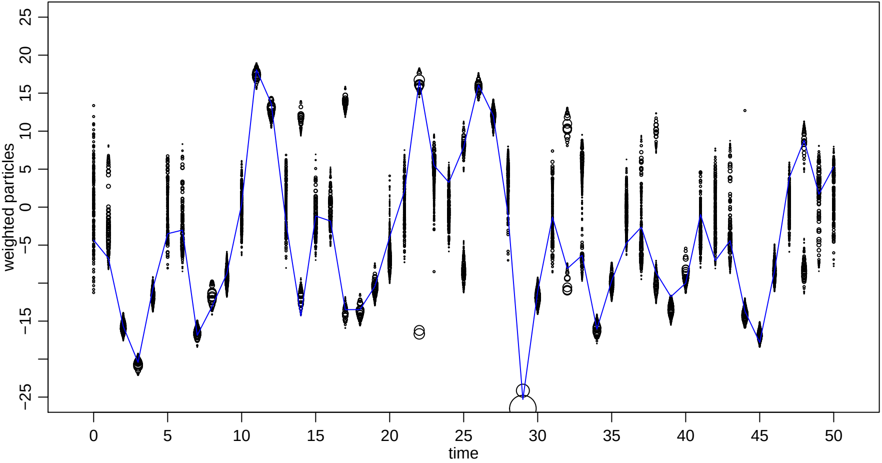
Left: Propagation (only few arrows shown).

Right: Reweighting and resampling



A simple example of particle filtering

Blue: $X_t =$ nonlinear AR, $Y_t = X_t^2$ plus noise (not shown). Black: Particle filter with area of circles $\propto$ weight. π_t^f is often bimodal.



Particle filter fails in high dimensions

In high dimensions, filter ensemble is often grossly overconfident, because too few prediction particles survive. Analysed theoretically by Bickel et al. (2008).

Auxiliary particle filtes is another method to reduce the loss of diversity, see below. However, it does not resolve the problem in high dimensions, and it requires that we can write down the transition *M*.

1. **Introduction**

2. **Basics about State Space Models and Filtering**

3. **Particle filters Part I**

4. **Ensemble Kalman filters**
   - Bayes formula for Gaussian prior and likelihood
   - Perturbed observation EnKF
   - Square root filters
   - Localization of EnKF

5. **The Ensemble Kalman Particle Filter**

6. **Particle Filters, Additional Topics**

7. **Parameter estimation**

Ensemble Kalman filter update

Based on the following standard result:

If $X \sim \mathcal{N}(\mu^p, P^p)$ and $Y|X = x \sim \mathcal{N}(Hx, R)$ then $X|Y = y \sim \mathcal{N}(\mu^f, P^f)$ where

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

where K is the Kalman gain

$$\begin{aligned} K &= Cov(X, Y)Cov(Y, Y)^{-1} = P^p H^T (HP^p H^T + R)^{-1} \\ &= Cov(X, Z)Cov(Z, Z)^{-1} R^{-1/2}, \quad Z = R^{-1/2} Y \end{aligned}$$

EnKF estimates μ^p and P^p from $(x^{p,j})$ and K, μ^f and P^f by plug-in: $\hat{K}$ denotes estimated K etc.

Finally $(x^{p,j})$ is converted into a sample with mean $\hat{\mu}^f$ and covariance $\hat{P}^f$. There are different algorithms to achieve this.

Perturbed observation Ensemble Kalman filter

This version is stochastic. Draw independent $\varepsilon^j \sim N(0, R)$ and set

$$x^{f,j} = x^{p,j} + \hat{K}(y + \varepsilon^j - Hx^{p,j})$$

Mean update of each particle with a perturbed observation.

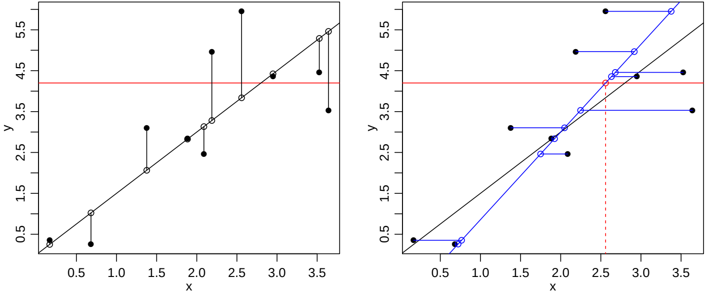
Can be considered as a balanced sample from

$$\frac{1}{N}\sum_{j=1}^{N}\mathcal{N}(x^{p,j} + \hat{K}(y - Hx^{p,j}), \hat{K}R\hat{K}^T)$$

The pairs $(x^{p,j}, y^{p,j} = Hx^{p.j} - \varepsilon^j)$ are a sample of the joint prediction distribution of (X, Y). The method regresses $x^{p,j}$ on $y^{p,j}$. The estimated regression line applied to the actual observation y gives the filter mean, the residuals represent the spread.

Regression interpretation of EnKF

Left: Forward regression line $y = Hx$ and points $(x^{p,j}, y^{p,j})$. Red line indicates actual observation y.
Right: Inverse regression line $x = \bar{x}^p + \hat{K}(y - \bar{y}^p)$.



Square root filters

This version is deterministic. First update the mean

$$\bar{x}^f = \bar{x}^p + \hat{K}(y - H\bar{x}^p)$$

and transform the residuals $x^{p,j} - \bar{x}^p$ linearly so that the empirical covariance is $\hat{P}^f$. This can be achieved by pre- or post-multiplication.

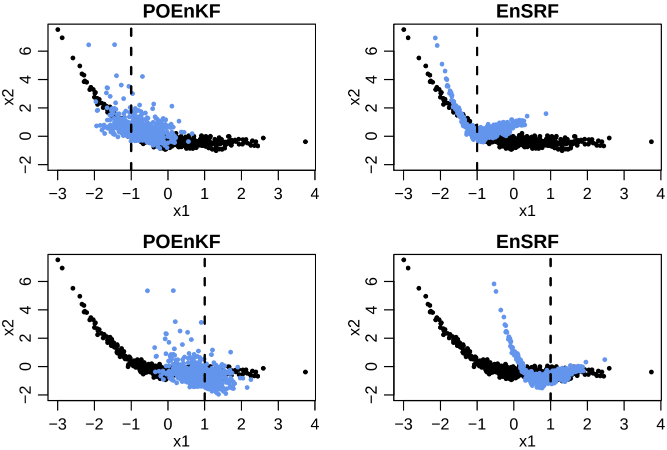
Let $\tilde{X}^p$ be the $d \times N$ matrix with j-th column equal to $x^{p,j} - \bar{x}^p$, and similarly $\tilde{X}^f$. Then the two possibilities are

$$\tilde{X}^f = A\tilde{X}^p, \quad \tilde{X}^f = \tilde{X}^p W$$

Equations for A ($d \times d$) and W ($N \times N$) are straightforward to write down.

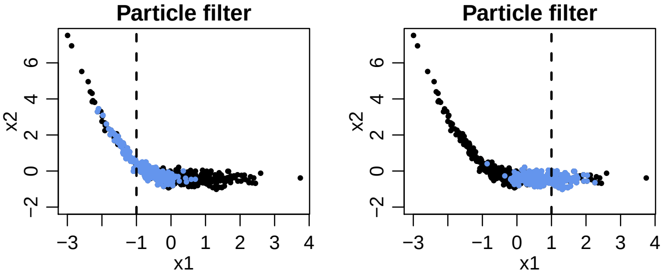
EnkF for banana-shaped prediction

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.

Assumptions for the EnKF

- The theoretical basis for the EnKF assumes a Gaussian prediction distribution and linear Gaussian observations
- To apply the method, fewer assumptions are needed. For the square root filter, $Y = H(X) + \varepsilon$ where ε is independent of X is sufficient
- The approach which regresses $x^{p,j}$ on $y^{p,j}$ can be applied for any conditional distribution of $Y|X$
- However, in these more general settings it is not clear how much the filter sample $(x^{f,j})$ differs from a sample of the true filter distribution
- There are results saying that if the distributions of $Y|X$ and $X|Y$ are both of the form “arbitrary mean plus independent errors”, then the joint distribution must be Gaussian

Data assimilation for seismicity

This is a joint project with Ylona van Dinther and Andreas Fichtner from ETH Zurich

- The model used is a seismo-thermal-mechanical model, applied to a one-dimensional laboratory experiment
- The seismo-part of the model has been developed by Ylona van Dinther in her PhD thesis
- So far, we use a perfect model framework, i.e. the “observations” come from a simulation of the model, not from an experiment
- We use the standard Ensemble Kalman Filter, there is no new methodology for the assimilation part until now
- I will show some slides by Ylona to give an idea of the model, the experiment and the results

EnKF in ensemble space

Assume $N \leq d$. The ensemble space is the $(N-1)$-dimensional hyperplane in $\mathbb{R}^d$ spanned by the particles $x^{p,j}$.

If $\hat{P}^p$ is the empirical covariance $(N-1)^{-1}\tilde{X}^p(\tilde{X}^p)^T$, the filter particles $x^{f,j}$ of the perturbed observation EnKF and of the post-multiplication square root filter are both in ensemble space. This is true because

$$\hat{K} = \hat{P}^p \cdots = \frac{1}{N-1}\tilde{X}^p \cdots$$

This is a possible reason for the stability of the EnKF: The directions of main expansion of the system are preserved during the update.

However, if $N \leq d$, the empirical covariance is not a good estimate of the true covariance. Regularized estimates (e.g. by shrinking off-diagonal elements) have better statistical properties, but then the update is in general not in the ensemble space any more.

Localization of EnKF

In many applications, $d \gg N$ (e.g. in weather prediction, $d \approx 10^8$, $N < 100$). Components of X and Y are typically related to locations in space.

Update is local if each component of $x^{f,j}$ is only influenced by components of y whose locations are close. Localization of updates are essential for stability.

There are two paradigms for localization

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

Observation localization

Update each component of $x^{p,j}$ individually, using only observations nearby, and concatenate (glue together) these updates. This concatenation can introduce discontinuities in the filter particles.

To keep this discontinuity small, use the same artificial noise values ϵ^j in the stochastic version. In the square root version choose the multipliers A or W in a continuous way.

One can further reduce the discontinuities introduced by localization by using some observations only partially. Assume R is diagonal. The update which uses only the subset y_V can be computed by setting $R_{ii} = \infty$ for all locations $i \notin V$. To use an observation only partially, one thus inflates R_{ii} by a factor in $(1, \infty)$.

Background localization

In observation localization , each component of x was updated only once, but each component of y was used several times. Here, each component of y is used only once, but each component of x is updated several times.

Regularize by elementwise multiplication of $\hat{P}^p$ with a compactly supported correlation function C (tapering). This sets correlations at large distance to zero. If also H is local, updating with one component of y then affects only a few components of x.

If R is diagonal, we can update serially, using one component of y after the other. However, after each update $\hat{P}^b$ should be replaced by $\hat{P}^f$ which has a larger range of correlations, destroying locality eventually. To preserve locality, need to neglect this increase in the range of correlations, e.g. by using always the same taper C.

Covariance inflation

So far we have assumed that there is no error in the propagation step. In practice, this is not the case: The model for state evolution usually is a simplification and there are errors in the numerical solution of differential equations.

The simplest solution is covariance inflation: Increase the spread of the prediction ensemble by a factor $\delta > 1$:

$$x^{p,j} \mapsto \tilde{x}^{j,p} = \bar{x}^p + \delta \cdot (x^{j,p} - \bar{x}^p)$$

Inflation factor δ may be spatially varying (a different factor for each coordinate) and is often chosen adaptively. Ideally, covariance inflation should ensure 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.

1. Introduction
2. Basics about State Space Models and Filtering
3. Particle filters Part I
4. Ensemble Kalman filters
5. The Ensemble Kalman Particle Filter
6. Particle Filters, Additional Topics
7. Parameter estimation

Bridging the Particle and the Ensemble Kalman filter

With localization, EnKF works surprisingly well in many real large scale applications. For instance, a localized squared root version called Local Ensemble Transform Kalman filter (LETKF) is the state of the art in weather prediction.

Can we explain this success by some theory? Can we improve the ability of the EnKF to deal with non-Gaussian features in π^p without loosing the stability?

In the following I discuss the EnKPF (Frei and K., 2013). It adresses the second question by progressive update

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

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

There are many other proposals to combine PF and EnKF.

Implementing the EnKPF

Both steps of the EnKPF can be done exactly. The first step gives the Gaussian mixture:

$$\pi^{f,\gamma} = \frac{1}{N}\sum_{j=1}^{N}\mathcal{N}(x^{p,j} + \hat{K}^{\gamma}(y - Hx^{p,j}), \hat{K}^{\gamma}R(\hat{K}^{\gamma})^{T})$$

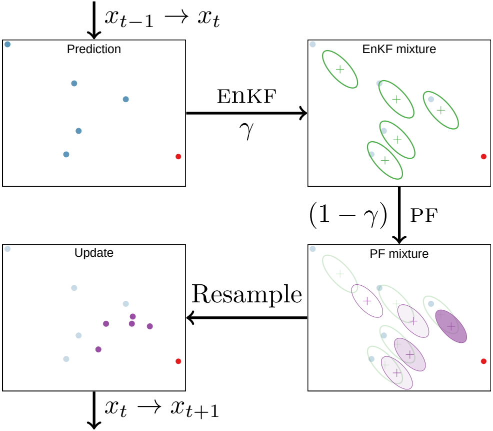
where $\hat{K}^{\gamma} =$ Kalman gain with $\gamma\hat{P}^{p}$. The second step gives a another Gaussian mixture:

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

from which we sample.

In order to capture non-Gaussian features of the prediction sample $(x^{p.j})$, choose γ as small as possible while keeping a minimal amount of diversity (measured e.g. by ESS of $(\alpha^{\gamma,j})$).

Illustration of EnKPF



$x_{t-1} \to x_t$

Prediction

EnKF

γ

EnKF mixture

$(1-\gamma)$

PF

PF mixture

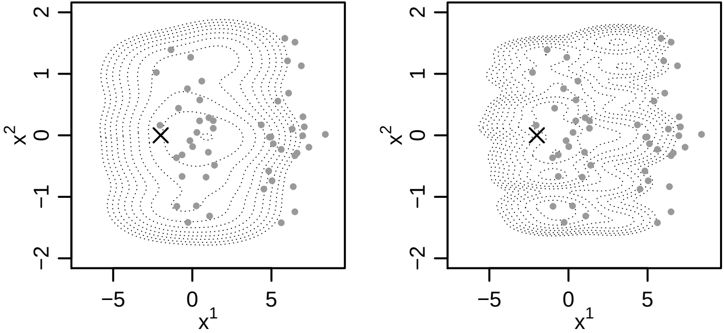
Resample

Update

$x_t \to x_{t+1}$

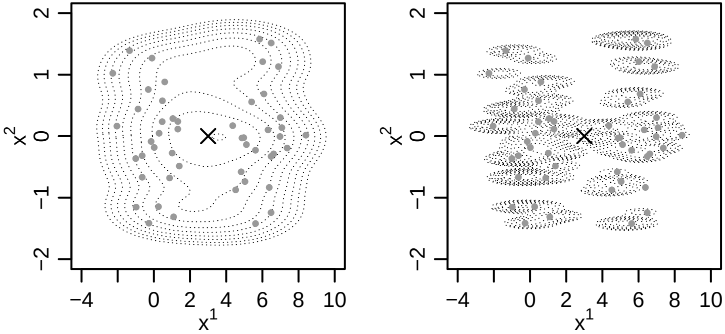
Single update for bimodal prior I

Left: EnKF, Right: EnKPF, diversity ≈ 40%.
Dots: Prior sample, Dotted: Contours of underlying filter density.



Single update for bimodal prior II

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



The EnKPF in ensemble space

The *j*-th particle in the EnKPF is

$$x^{f,j} = \mu^{\gamma,I(j)} + \eta^{\gamma,j}$$

where $I(1), \dots, I(N)$ are the resampling indices and $\eta^{\gamma,j} \sim \mathcal{N}(0, \hat{P}^{\gamma})$. By the same argument as for the EnKF, one can see that $x^{f,j}$ is in ensemble space if $\hat{P}^{p}$ is the empirical covariance:

- $\mu^{\gamma,j}$ is the result of two EnKF updates
- Resampling just deletes some of these points
- $\eta^{\gamma,j}$ can be generated by adding two independent $\mathcal{N}(0, R)$ variables, multiplied by two Kalman gains

A natural question is whether there is a square root version of the EnKPF, i.e. an update of the form

$$x^{f,j} = \mu^{\gamma,I(j)} + \tilde{X}^{p} W$$

This leads to a Ricatti equation for W (Robert and K., 2016).

Localizing the EnKPF

Resampling does not lend itself to localization, unless the resampling probabilities are global.

If we use observation localization, resampling probabilities change continuously from one location to the next. Resampling then occasionally leads to concatenating updates that come from two different prediction particles which can lead to large discontinuities.

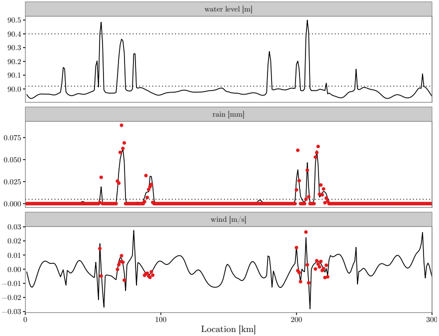
With background localization, we can concatenate updates in two separated regions such that second moments are still correct. Details in Robert and K. (2016)

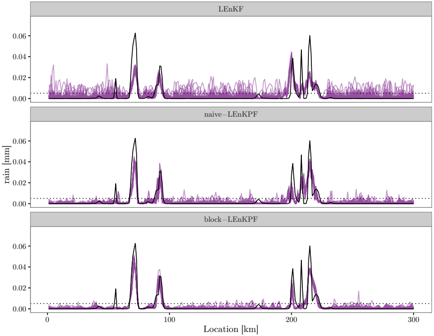
The modified shallow water equation

This is a one-dimensional model resembling cumulus convection, proposed by Würsch and Craig (2014). It has three variables, the height of the cloud, the velocity and rain water. The first two obey a shallow water equation with a modified geopotential and there is a conservation equation for rain water. Clouds with moist air raise, and if a certain height is reached, rain starts and the cloud slowly disappears. Random perturbation of the velocity are added which produce new clouds. Height is unobserved, and wind is only observed where there is rain.

In a cooperation with MeteoSwiss and the German Weather Service, Sylvain Robert and I are currently testing 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. Results are not yet available.

A modified shallow water equation

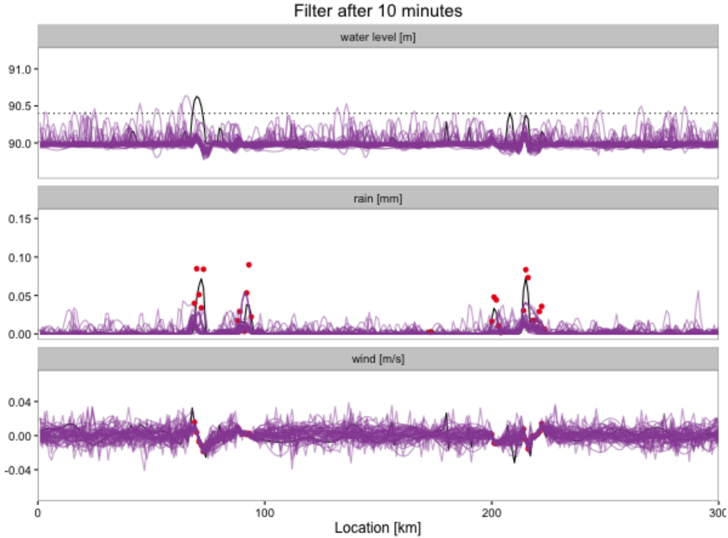




A solution of the equation at a fixed time with observations in red.

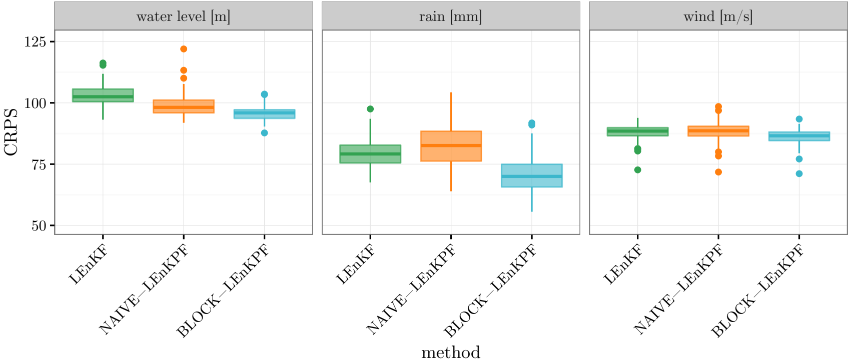
Rain component of ensemble members at the same time. Top: localized EnKF, Middle and Bottom: 2 versions of localized EnKPF.

Filter iterations



Calibration and sharpness of filters

Continuous ranked probability score (CRPS) for 3 ensemble filters in a simulation with many iterations.



1. **Introduction**

2. **Basics about State Space Models and Filtering**

3. **Particle filters Part I**

4. **Ensemble Kalman filters**

5. **The Ensemble Kalman Particle Filter**

6. **Particle Filters, Additional Topics**
   - Auxiliary particle filter
   - Particle smoothing
   - General sequential Monte Carlo

7. **Parameter estimation**

Further conditional distributions

In the following we need $\pi_{s:t|u}$ = conditional distribution of $X_{s:t}$ given $Y_{1:u} = y_{1:u}$. Compared with previous notation $\pi_t^p = \pi_{t|t-1}$ and $\pi_t^f = \pi_{t|t}$.

By Bayes' formula

$$\pi_{0:t|t}(dx_{0:t}) \propto M_0(dx_0) \prod_{s=1}^{t} M(dx_s|x_{s-1})g(y_s|x_s)$$

or in recursive form

$$\pi_{0:t|t}(dx_{0:t}) \propto \pi_{0:t-1|t-1}(dx_{0:t-1})M(dx_t|x_{t-1})g(y_t|x_t)$$

This is essentially the same recursion as for π_t^f, except that we have an extension instead of propagation.

All other $\pi_{s:t|u}$ follow in principle by marginalization. In one of the subsections, we will see more efficient ways to compute e.g. the so-called smoothing distribution $\pi_{t|T}$.

Auxiliary particle filter

This is an idea by Pitt and Shephard (1999) to improve the balance of weights. The new observation y_t is already taken into account in the propagation step to generate particles with better fit to y_t.

The state transition M must be known. For simplicity I assume here that it has a density denoted by m. Then also all $\pi_{s:t|u}$ have a density.

In the auxiliary particle filter, filter particles $(x_{t-1}^{f,j})$ by a transition Q with density q which can (and should) depend on y_t. Assume for simplicity that weights $\alpha_t^{f,j} \equiv \frac{1}{N}$.

Weights for propagated particles

If we want the propagated particles $(x_t^{f,j})$ to approximate π_t^f, we must weight them by

$$\begin{aligned}\alpha_t^{f,j} &= \frac{\pi_t^f(x_t^{f,j})}{\int \pi_{t-1}^f(x')q(x_t^{f,j}|x')dx'} \\ &\propto \frac{g(y_t|x_t^{f,j}) \int \pi_{t-1}^f(x')m(x_t^{f,j}|x')dx'}{\int \pi_{t-1}^f(x')q(x_t^{f,j}|x')dx'}\end{aligned}$$

Because of the integrals, these weights cannot be computed.

If we replace the integrals by averages using $(x_{t-1}^{f,j})$, computation is possible, but with complexity $O(N^2)$.

Pairs of particles

A better solution considers the pairs of particles $(x_{t-1}^{f,j}, x_t^{f,j})$ which have the density

$$\pi_{t-1}^f(x_{t-1})q(x_t|x_{t-1}).$$

If we want them to approximate $\pi_{t-1:t|t}$, the conditional distribution of $X_{t-1:t}$ given $y_{1:t}$, we must use the weights

$$\alpha_{t-1:t}^{f,j} \propto \frac{m(x_t^{f,j}|x_{t-1}^{f,j})g(y_t|x_t^{f,j})}{q(x_t^{f,j}|x_{t-1}^{f,j})}$$

In particular the weighted sample $(x_t^{f,j}, \alpha_{t-1:t}^{f,j})$ approximates then π_t^f, but the weights now depend on both $x_{t-1}^{f,j}$ and $x_t^{f,j}$.

This idea of considering the distribution of interest as the marginal of some distribution on a product space which is easier to sample plays an important role in modern Monte Carlo.

Choice of the propagation q

One can show that the choice of q which minimizes the variance of the weights is the conditional distribution of X_t given (x_{t-1}, y_t), i.e.

$$\begin{aligned} q(x|y_t, x_{t-1}) &= \frac{m(x|x_{t-1})g(y_t|x)}{p(y_t|x_{t-1})} \\ p(y_t|x_{t-1}) &= \int m(x|x_{t-1})g(y_t|x)dx \end{aligned}$$

Then the weights depend only on $x_{t-1}^{f,j}$:

$$\alpha_{t-1:t}^{f,j} \propto p(y_t|x_{t-1}^{f,j})$$

One can then gain diversity by weighting and resampling the particles $(x_{t-1}^{f,j})$ before instead of after propagation.

In the standard particle filter, the weights are proportional to the likelihood of x_t given y_t whereas now they are proportional to the likelihood of x_{t-1}. Because this time delay looses information, the weights are more equal, but in general the gain is not huge.

Approximating the ideal transition

The ideal choice for q and the likelihood $p(y_t|x_{t-1})$ which is needed for the weights are usually not available explicitely. Often we can use a Gaussian approximation

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

obtained by using e.g. a Taylor approximation around $\arg\max_x m(x|x_{t-1}) g(y_t|x)$.

The auxiliary particle filter algorithm

The idea of weighting and resampling the particles before propagation can be used in general. Hence we choose a weight function w and a transition density q which depend also on y_t.

Given the weighted filter sample $(x_{t-1}^{f,j}, \alpha_{t-1}^{f,j})$ at time $t-1$, the algorithm does the following

- Resample $(x_{t-1}^{f,j})$ with probabilities proportional to $w(x_{t-1}^{f,j})\alpha_{t-1}^{f,j}$. Let the j-th resampled particle be $x_{t-1}^{f,I(j)}$.
- Propagate resampled particles with transition q: $x_t^{f,j} \sim q(x|x_{t-1}^{f,I(j)})dx$, independently for different indices j.
- Compute new weights

$$\alpha_t^{f,j} \propto \frac{g(y_t|x_t^{f,j})m(x_t^{f,j}|x_{t-1}^{f,I(j)})}{w(x_{t-1}^{f,j})\alpha_{t-1}^{f,j}q(x_t^{f,j}|x_{t-1}^{f,I(j)})}$$

Overview of smoothing methods

Smoothing is concerned with conditional distributions of past state values. I will discuss the following methods

- Smoothing by filtering of paths
- Forward filtering, backward smoothing
- Applications of the two-filter formula
- Ensemble Kalman smoothers

All methods require the storage of past filter samples.

Smoothing by filtering of paths

We can implement the recursion

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

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

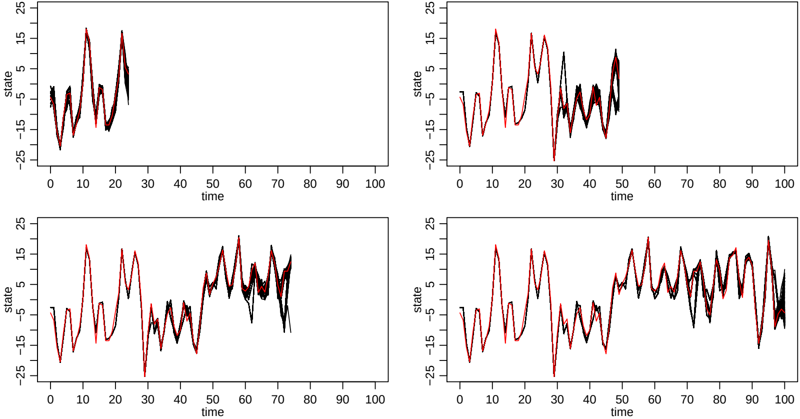
$$x^j_{0:t|t-1} = (x^j_{0:t-1|t-1}, x^{p,j}_t)$$

and resample $(x^j_{0:t|t-1})$ with probabilities $\propto g(y_t|x^{p,j}_t)$.

Then $(x^j_{s|t})$ 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 $\pi_{0:t|t}$

Nonlinear AR-model for X_t, $Y_t = X_t^2$ plus noise.
Black: Filter samples. Red: True state.



Forward filtering, backward smoothing

Assume state transition $M(dx_t|x_{t-1})$ has a density $m(x_t|x_{t-1})$. Under $\pi_{0:t|t}$, $(X_t, X_{t-1}, \ldots, X_0)$ is Markov chain with backward transition densities

$$p(x_s|x_{s+1}, y_{1:t}) = p(x_s|x_{s+1}, y_{1:s}) \propto m(x_{s+1}|x_s)\pi_s^f(x_s)$$

Hence we can generate recursively an approximate sample from $\pi_{0:t|t}$ by weighting and resampling the particle filter approximation of π_s^f

$$\mathbb{P}(x_{s|t}^j = x_s^{f,k}|x_{s+1|t}^j) = \frac{m(x_{s+1|t}^j|x_s^{f,k})}{\sum_{\ell=1}^N m(x_{s+1|t}^j|x_s^{f,\ell})}.$$

For different values of $x_{s+1|t}^j$ we have different resampling probabilities, so a naive implementation has complexity $O(N^2)$. Moreover, many ties occur if spread of $\pi_{s|t}$ is much lower than spread of $\pi_{s|s}$.

Smoothing by accept/reject

For an absolutely continuous approximation of $p(x_s|x_{s+1}, y_{1:t})$, we go back to the filter sample at time $s-1$, using

$$\pi_s^f(x_s) \propto g(y_s|x_s) \sum_{k=1}^{N} m(x_s|x_{s-1}^{f,k})$$

Then we draw $x_{s|t}^j$ from

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

Since we need only one draw from this distribution, we have to use the accept/reject method instead of importance sampling.

Smoothing using the two filter formula

Another approach is based on the formula

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

The second factor is unknown, but it satisfies the recursion

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

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

Monte Carlo approximation of $p(y_{s:t}|x_s)$

If $\int p(y_{s:t}|x_s)dx_s < \infty$, weighted particles $(\tilde{x}_s^j, \tilde{\alpha}_s^j)$ can be constructed recursively (starting at $s = t$ and going backward) such that

$$\int \psi(x_s)p(y_{s:t}|x_s)dx_s \approx \sum_{j=1}^{N} \psi(\tilde{x}_s^j)\tilde{\alpha}_s^j$$

for bounded $\psi : \mathbb{R}^d \to \mathbb{R}$. For the general case, need to introduce an additional weight function.

Then we obtain an approximation of $\pi_{s|t}$ by the density proportional to

$$\underbrace{\sum_{i=1}^{N} m(x_s|x_{s-1}^{f,i})}_{\approx \pi_s^p} g(y_s|x_s) \underbrace{\sum_{j=1}^{N} m(\tilde{x}_{s+1}^j|x_s)\tilde{\alpha}_{s+1}^j}_{\approx p(y_{s+1:t}|x_s)}$$

$O(N)$ smoothing algorithms

The density on the previous slide is a mixture of N^2 components. The same ideas as in auxiliary particle filter can be used to sample from it. Algorithms that have a low variance of weights have however typically complexity $O(N^2)$. The same is true for the accept/reject method before.

Suggestions for an efficient $O(N)$ algorithm are in Fearnhead et al. (2010).

Smoothing by EnKF

We can also apply the EnKF to the whole path $x_{0:t}$. We set $x^j_{0:t|t-1} = (x^j_{0:t-1|t-1}, x^{p,j}_t)$ and then apply the updates

$$x^j_{s|t} = x^j_{s|t-1} + \hat{K}_{st}(y_t + \varepsilon^j_t - Hx^j_{t|t-1})$$

for $s = t, t-1, \ldots$ where $\hat{K}_{st}$ is an estimate of $Cov(X_s, Y_t|y_{1:t-1})Cov(Y_t, Y_t|y_{1:t-1})^{-1}$

Alternatively, one can use an ensemble implementation of the forward/backward method for linear Gaussian state space models. It is

$$x^j_{s|t} = x^{f,\tau(j)}_s + \hat{S}_s(x^j_{s+1|t} - x^{p,\tau(j)}_{s+1})$$

where τ is a permutation of the indices and $\hat{S}_s$ is an estimate of $Cov(X_s, X_{s+1}|y_{1:s})(P^p_{s+1})^{-1}$. (Cosme et al., 2012; Frei, 2013).

Sequential Monte Carlo: Sampling from moving targets

Particle filter ideas can also be used in other problems where one wants to sample from a sequence of related distributions π_t, $t = 0, 1, \ldots, n$.

An important class of such problems occurs if one approximates a difficult target distribution π by a sequence of simpler approximating distributions $\pi_0, \pi_1, \ldots, \pi_n = \pi$, e.g. by simulated tempering

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

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

$$\alpha_t^j \propto \frac{\pi_t(x_t^j)}{\int \pi_{t-1}(x_{t-1})q_t(x_t^j|x_{t-1})dx_{t-1}}.$$

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 weighting and resampling. Propagation just does some rejuvenation by breaking ties from the previous resampling step.

Resampling pairs of particles

As in the auxiliary particle filter, we consider the pairs (x_{t-1}^j, x_t^j) which have density

$$\pi_{t-1}(x_{t-1})q_t(x_t|x_{t-1})$$

and convert them by weighting and resampling to have a distribution with second marginal π_t.

Densities with second marginal π_t have the form

$$\pi_t(x_t)r_{t-1}(x_{t-1}|x_t)$$

where r_{t-1} is an arbitrary "backward" transition density. Resampling of (x_t^j) is done with probabilities

$$\alpha_t^j \propto \frac{\pi_t(x_t^j)r_{t-1}(x_{t-1}^j|x_t^j)}{\pi_{t-1}(x_{t-1}^j)q_t(x_t^j|x_{t-1}^j)}.$$

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 (α_t^j) 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

In the particle filter $\pi_t = \pi_t^f$, and we do not have an explicit expression up to a normalizing constant. But if we choose r_{t-1} implicitly by

$$\begin{aligned} \pi_t(x_t) r_{t-1}(x_{t-1}|x_t) &= \pi_{t-1:t|t}(x_{t-1}, x_t) \\ &\propto \pi_{t-1}(x_{t-1}) p(x_t|x_{t-1}) g(y_t|x_t) \end{aligned}$$

the resampling probabilities can be computed because the unknown π_{t-1} cancels.

1. **Introduction**

2. **Basics about State Space Models and Filtering**

3. **Particle filters Part I**

4. **Ensemble Kalman filters**

5. **The Ensemble Kalman Particle Filter**

6. **Particle Filters, Additional Topics**

7. **Parameter estimation**

Parameter estimation

We discuss the estimation of static parameters θ that are present in the state transition M and/or the observation density g.

In the Bayesian framework we put a prior p_0 on θ and want to approximately sample from the posterior

$$p(\theta|y_{1:T}) \propto p_0(\theta)p(y_{1:T}|\theta)$$

For filtering or prediction, we need to take the uncertainty about θ into account

$$\pi_t^f = \int p(x_t|y_{1:t},\theta)p(\theta|y_{1:t})d\theta = \int p(x_t,\theta|y_{1:t})d\theta$$

and similarly for π_t^p. Typically, one tries to sample jointly the state and the parameter.

Parameters included in the state: PF

The easiest method is to include θ as a deterministic component of the state: $\theta_0^{f,j} \sim p_0(\theta)d\theta$ and

$$\theta_t^{p,j} = \theta_{t-1}^{f,j}, \quad x_t^{p,j} \sim M(dx|x_{t-1}^{f,j}, \theta_{t-1}^{f,j})$$

The particle filter degenerates quickly because there is no rejuvenation of the θ-component. One can avoid this by adding noise to $\theta_t^{f,j}$, 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 $p(\theta|y_{1:t})$.

Parameters included in the state: EnKF

For parameters θ in the transition kernel 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).

MCMC for state space models

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

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

and accept it with probability

$$a(\theta,\theta') = \min\left(1, \frac{p_0(\theta')p(y_{1:T}|\theta')q(\theta|\theta')}{p_0(\theta)p(y_{1:T}|\theta)q(\theta'|\theta)}\right)$$

Otherwise keep the current value θ.

As the number of iterations goes to infinity, the values obtained in this way are approximately distributed according to the posterior, under weak conditions on the proposal density q.

However, the likelihood $p(y_{1:T}|\theta')$ is not tractable and thus the acceptance probability cannot be computed

Particle MCMC

The likelihood

$$p(y_{1:T}|\theta) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}, \theta)$$

is not available, but it can be estimated by running a particle filter with parameter θ and computing the average of the weights.

A basic result shows that this estimate is unbiased for every fixed $y_{1:T}$ Note: Contrary to some claims, the estimate of each factor is not unbiased in general.

Andrieu & Roberts (2009), Andrieu et al. (2010) have shown that if one replaces the likelihood $p(y_{1:T}|\theta)$ by an unbiased non-negative estimate $\hat{p}(y_{1:T}|\theta)$ in the formula for the acceptance probability, the algorithm still produces samples from the posterior when the number of iterations tends to infinity. The variance of the estimate $\hat{p}(y_{1:T}|\theta)$ does not have to go to zero, the number of particles is fixed and arbitrary.

Unbiased estimates for the likelihood are sufficient

Write $\hat{p}(y_{1:T}|\theta) = \hat{p}(y_{1:T}|\theta, U)$ where U are the random variables used to generate the likelihood estimates. W.l.o.g., assume $U \sim p(u)du$, independent of θ. Then

$$\int \hat{p}(y_{1:T}|\theta, u)p(u)du = p(y_{1:T}|\theta)$$

Therefore the joint density

$$\frac{p_0(\theta)\hat{p}(y_{1:T}|\theta, u)p(u)}{p(y_{1:T})}$$

has marginal $p(\theta|y_{1:T})$. It suffices to sample from this density and to keep only the component θ.

Metropolis Hastings with proposal $(\theta', U') \sim q(\theta'|\theta)d\theta' p(u')du'$ gives the acceptance probability

$$a(\theta, \theta') = \min\left(1, \frac{p_0(\theta')\hat{p}(y_{1:T}|\theta', u')q(\theta|\theta')}{p_0(\theta)\hat{p}(y_{1:T}|\theta, u)q(\theta'|\theta)}\right)$$

Tuning particle MCMC

Particle MCMC is computer-intensive because each time a new value θ' is proposed, we need to run a new particle filter.

For implementation, one has to choose the number N of particles and the proposal q for θ. The latter is usually taken as a Gaussian random walk with some variance Σ. N determines the computational cost and N and Σ together the mixing of the chain and thus the accuracy of the approximation of the posterior.

Some analysis is possible if one assumes that $\log \hat{p}(y_{1:T}|\theta)$ is normal with mean $\log p(y_{1:T}|\theta) - \sigma^2/(2N)$ and variance σ^2/N (the bias of log likelihood makes the likelihood unbiased) and that the asymptotic variance σ^2 is independent of θ.

For details, see recent work of Doucet et al. (2015), Sherlock et al. (2015), Nemeth et al. (2016).

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.
- Particle filter is more general, but degenerates quickly in high dimensions.
- Ensemble Kalman filter works well also in high dimensions, but we do not understand the reasons for this.
- Estimation of static parameters and improving the ensemble Kalman filter in high dimensions are the main challenges.

Thank you for your attention!