# State space models, filtering and environmental applications

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

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

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

## Table of contents

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters

## Section 1

Introductory Examples
- Postprocessing of numerical weather predictions
- Data assimilation for weather prediction
- Stochastic reaction networks
- Rare event estimation

Basics of state space models and filtering

Kalman filter and its applications

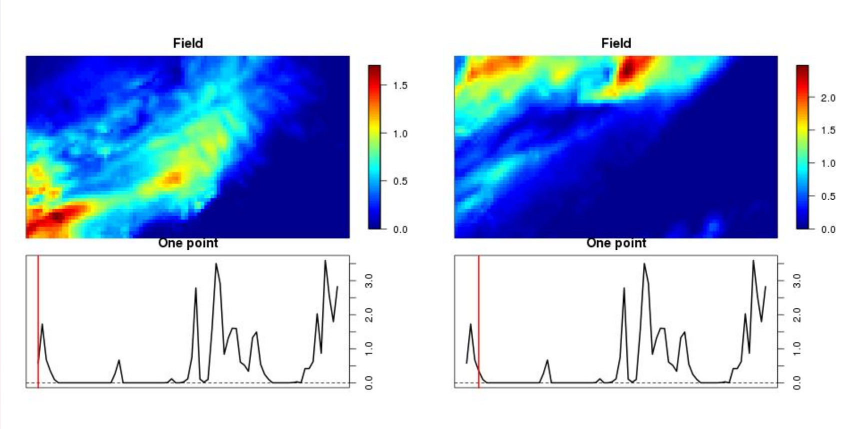
Particle filters

Ensemble Kalman filters

Extensions of particle filters

## Numerical precipitation predictions

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



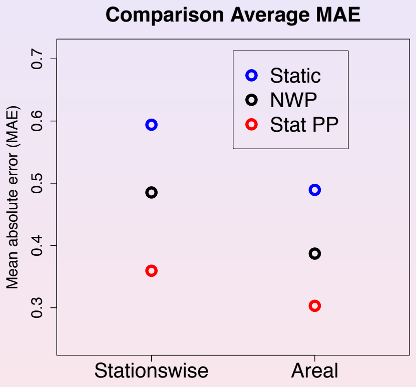
# Statistical postprocessing

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

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

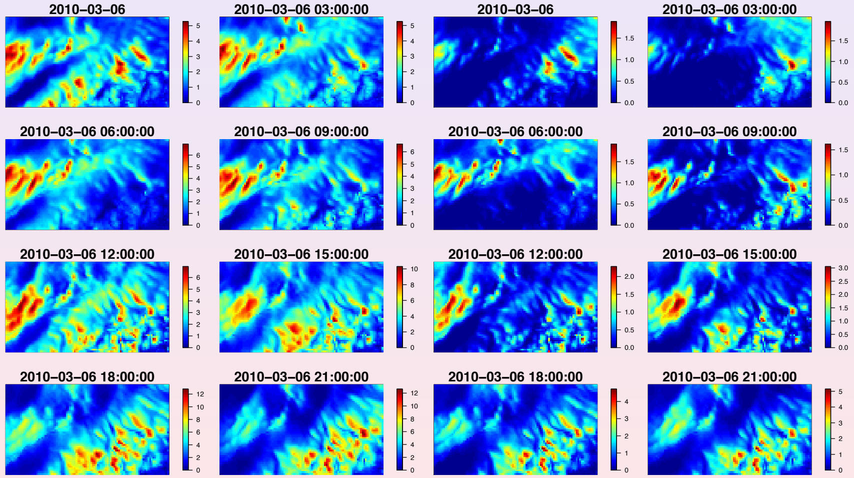
Sigrist et al. (2014) use NWP forecasts as explanatory variables in a statistical space-time model of precipitation. The Kalman filter is an essential tool for fitting this model and for making predictions.

# Mean absolute error of 3 forecasts



# Effect of postprocessing

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



# Weather prediction and data assimilation

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

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

## Lorenz models

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

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

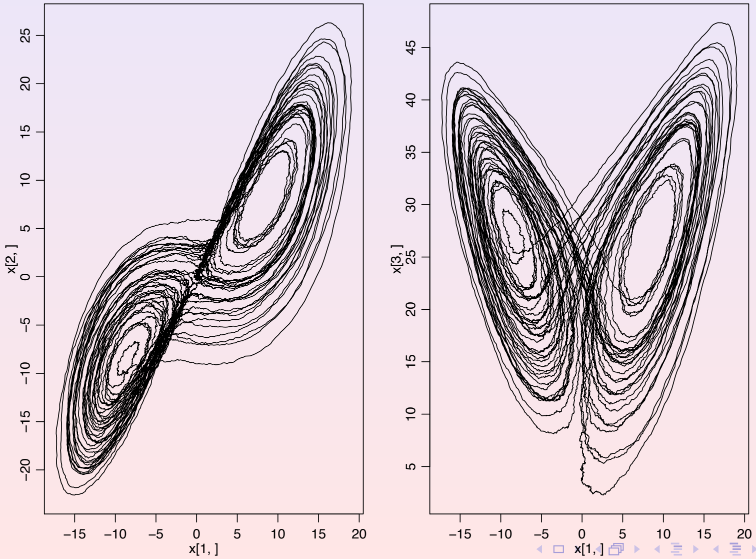
The Lorenz 96 model is

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

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

## The Lorenz 63 model

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



## Data assimilation for the Lorenz 63 model

Still to come: A figure illustrating how the particle filter looses track.

## Stochastic kinetic reaction networks

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

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

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

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

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

## Stochastic kinetic reaction networks, ctd.

$(x_t)$ is a Markov process in continuous time:

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

The goal is to estimate the rate constants $\theta_i$ and the trajectory of the process from noisy measurements of some components $x^j_{t_i}$ at discrete times $t_i$.

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

## A simulated example

Still to come.

## Rare event estimation in a queueing network

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

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

## Importance splitting

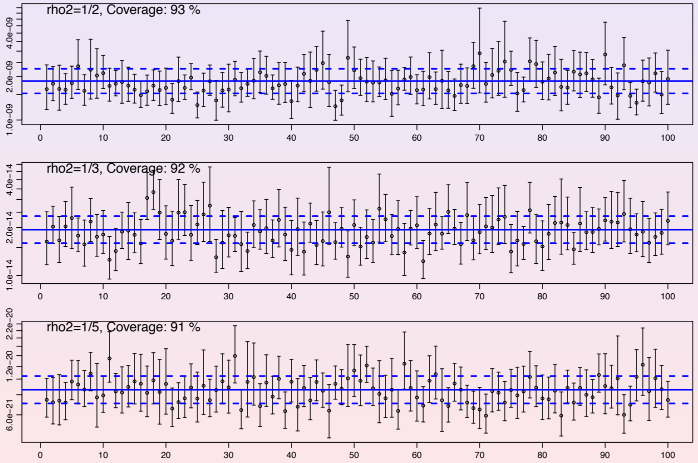
Alternatives to direct Monte Carlo are importance sampling and importance splitting.

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

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

## Illustration of results

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



## Section 2

Introductory Examples

Basics of state space models and filtering

The state space model

Filtering and smoothing distributions

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters

## State space models

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

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

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

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

Observations $Y_1, Y_2, \ldots$ are conditionally independent with

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

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

## Graphical representation of state space models

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

$$\begin{tikzcd}
\ldots \arrow[r] & X_{t-1} \arrow[r] \arrow[d] & X_t \arrow[r] \arrow[d] & X_{t+1} \arrow[r] \arrow[d] & \ldots \\
\ldots & Y_{t-1} & Y_t & Y_{t+1} & \ldots
\end{tikzcd}$$

It implies various conditional independencies which we will use later.

An alternative representation of a general state space model is

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

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

# Notation

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

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

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

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

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

# Gibbs sampler for state space models

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

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

If the model contains unknown parameters, also the density of parameters given $x_{0:t}$ and $y_{1:t}$ is available. Hence the Gibbs sampler can be used. But

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

# Filtering recursions

Prediction and filter distributions can be computed recursively:

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

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

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

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

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

The denominator in Bayes' formula is

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

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

# Smoothing

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

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

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

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

There are also expressions for the forward transition distributions which we will discuss later.

# Implementations

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

# Section 3

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications
Gaussian linear state space models
Kalman filter and smoother
Fitting of stochastic advection-diffusion models

Particle filters

Ensemble Kalman filters

Extensions of particle filters

# Gaussian linear state space models

A Gaussian linear state space model has the form

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

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

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

# Kalman filter

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

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

and

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

where

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

is the so-called Kalman gain.

# Kalman smoother

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

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

This is the basis of the forward-filtering backward simulation algorithm.

It is not difficult to derive also a backward recursion for the mean and covariance of $\pi_{s|t}$.

# Computational complexity

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

- Forward filtering and likelihood evaluation has complexity $O(T(q^2 + d^3))$.
- Backward smoothing has additional complexity $O(Tq^3)$.
- If $q$ and/or $d$ are large, efficient computation becomes crucial.
- Savings are possible if $P_{t-1|t}$ and $P_{t|t}$ are sparse by using sparse Cholesky factorizations.
- We discuss next the use of Kalman filter and smoother for the introductory example of precipitation.

# Back to the first example

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

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

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

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

# Modeling potential rainfall

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

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

- $y_F(t, \boldsymbol{s})$ NWP forecast made at 0:00 UTC of the same day,
- $X(t, \boldsymbol{s})$ structured random effect (mean zero),
- $\nu(t, \boldsymbol{s})$ nugget effect,
- Unknown parameters: $\beta_1$, $\beta_2$, $\lambda > 0$, nugget variance $\sigma_\nu^2$, parameters for $X$ (to be discussed later).

We use precipitation data from 32 stations that the NWP forecast does not use directly.

## Gaussian space-time fields

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

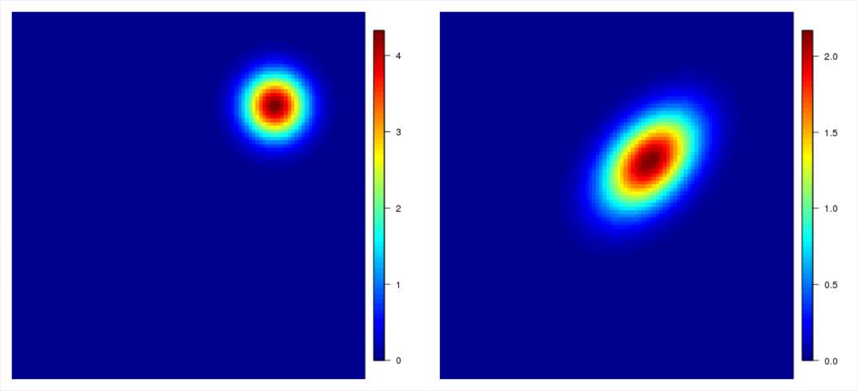
## Stochastic advection-diffusion model

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

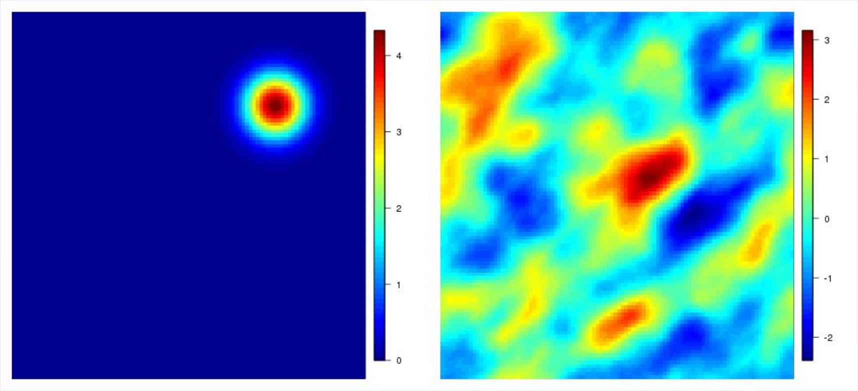
$$\frac{\partial}{\partial t}X(t,\mathbf{s}) = -\boldsymbol\mu\cdot\nabla X(t,\mathbf{s}) + \nabla\cdot\boldsymbol\Sigma\nabla X(t,\mathbf{s}) - \kappa X(t,\mathbf{s}) + \epsilon(t,\mathbf{s})$$

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

## Illustration of deterministic PDE



## Illustration of SPDE



## Elementary solutions of the SPDE

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

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

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

Then the solution is

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

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

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

## Stationary solutions of the SPDE

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

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

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

Then the solution has the representation

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

Amplitudes develop independently for differen wavenumbers.

## Discretizing time and space

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

$$\boldsymbol{X}(t_i) = \boldsymbol{\Phi}\boldsymbol{\alpha}(t_i), \quad \boldsymbol{\alpha}(t_i) = \boldsymbol{G}\boldsymbol{\alpha}(t_{i-1}) + \mathcal{N}(0,\boldsymbol{Q}).$$

Here

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

## Fitting with linear Gaussian observations

Adding a nugget to $\boldsymbol{X}$, we obtain a Gaussian linear state space model with states $\boldsymbol{\alpha}(t_i)$ and observations $\boldsymbol{w}_i$:

$$\boldsymbol{\alpha}(t_i) = \boldsymbol{G}\boldsymbol{\alpha}(t_{i-1}) + \mathcal{N}(0,\boldsymbol{Q}), \quad \boldsymbol{w}_i = \boldsymbol{\Phi}\boldsymbol{\alpha}(t_i) + \mathcal{N}(0,\sigma_\nu^2\boldsymbol{I}).$$

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

If also parameters are unknown, do simultaneous updates of parameters and states: Propose

$$(\boldsymbol{\alpha}^*, \theta^*) \mid \boldsymbol{\alpha}, \theta \sim q(\theta^* \mid \theta)p(\boldsymbol{\alpha}^* \mid \boldsymbol{w}, \theta^*)d\theta^* d\alpha^*$$

and accept it with probability $\min(1, p(\theta^* \mid \boldsymbol{w})/p(\theta \mid \boldsymbol{w}))$.

This is feasible as long the set of wavenumbers and the set of locations with observations are not too big. Sampling $X$ on a finer grid can be done at the end.

## Fitting with linear Gaussian observations, ctd.

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

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

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

If observations are available on only parts of the grid, we can impute the missing values in an iterative procedure.

## Nonlinear and non-Gaussian observations

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

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

## Section 4

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

The basic particle filter<br>
Balanced sampling<br>
Auxiliary particle filter<br>
Likelihood estimation<br>
Importance splitting as a particle filter

Ensemble Kalman filters

Extensions of particle filters

## Particle Filter

Remember the filtering recursions:

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

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

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

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

## Why resampling ?

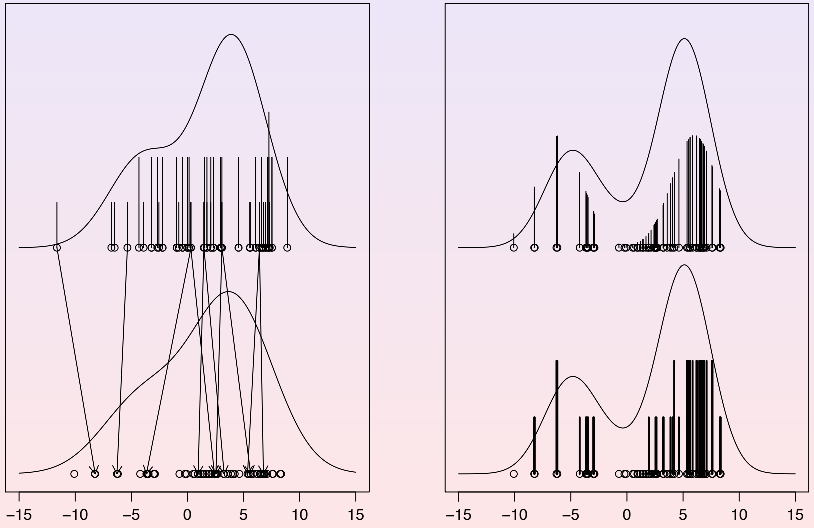
- Update step can be split into a weighting and a resampling step. This gives a weighted approximation of

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

- Resampling introduces additional noise, but it avoids waisting simulation efforts in the next propagation step with unlikely starting values.
- Hence it is better to use the weighted approximation of $\pi_{t|t}$ and to resample at the beginning of the propagation step.
- In particular, resampling introduces ties among $x^j_{t|t}$. These are broken in the next propagation step (called “rejuvenation”) if state dynamics is stochastic.
- If some components of state change deterministically, can break ties by adding noise (compensated by shrinkage to the mean).

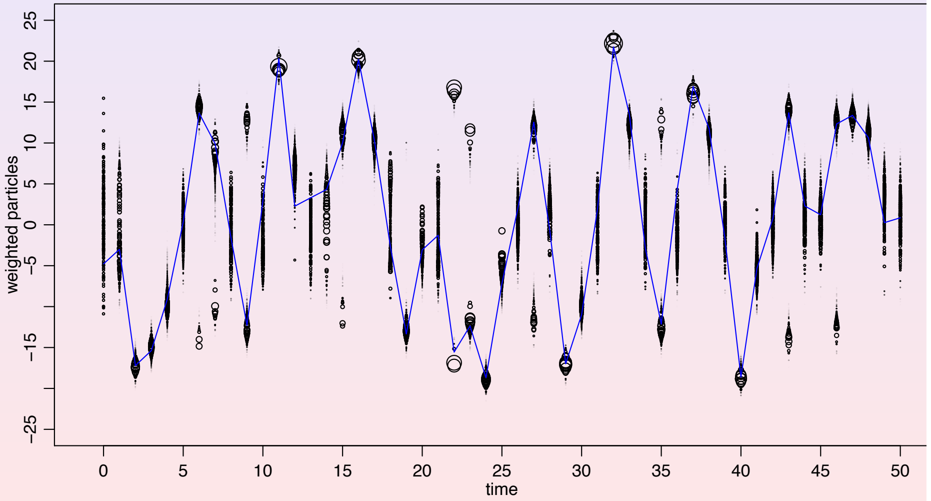
## A single step of the particle filter

Left: Propagation (only few arrows shown).
Right: Reweighting and resampling



## Several steps of the particle filter

Area of circles $\propto$ weight. $Y_t = X_t^2$ plus noise, Blue=true (unobserved) $X_t$.



## Effective sample size (ESS)

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

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

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

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

## Balanced sampling

For resampling, can use any scheme s.th. $x^j_{t|t-1}$ is chosen on average $Nw^j_t$ times. A sampling scheme is called balanced if $x^j_{t|t-1}$ is chosen $\left\lfloor Nw^j_t \right\rfloor$ or $\left\lfloor Nw^j_t \right\rfloor + 1$ times.

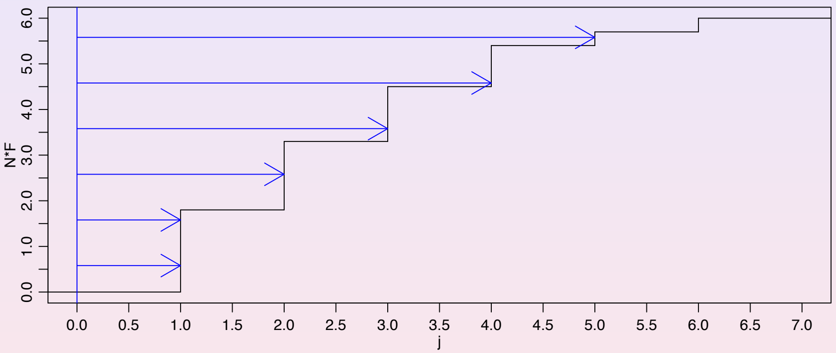
There are many balanced sampling schemes. The simplest such scheme is

$$x^j_{t|t} = x^{I_j}_{t|t-1}, \quad I_j = F_t^{-1}((j-U)/N)$$

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

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

## Illustration of balanced sampling



## Auxiliary particle filter

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

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

$$w^j_t \propto \tilde{w}(x^j_{t-1|t-1}, x^j_{t,*}, y_t) = \frac{g(y_t \mid x^j_{t,*})k(x^j_{t,*} \mid x^j_{t-1|t-1})}{q(x^j_{t,*} \mid x^j_{t-1|t-1}, y_t)}.$$

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

## Auxiliary particle filter ctd.

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

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

  In this case, $w^j_t \propto p(y_t \mid x^j_{t-1|t-1})$, and resampling before propagation increases diversity.
- Also in other cases one can add a resampling step with probabilities $\tau(x^j_{t-1|t-1}, y_t)$ before propagation. Then weights after propagation are

$$w^j_t \propto \tilde{w}(x^j_{t-1|t-1}, x^j_{t,*}, y_t) = \frac{g(y_t \mid x^j_{t,*})k(x^j_{t,*} \mid x^j_{t-1|t-1})}{\tau(x^j_{t-1|t-1}, y_t)q(x^j_{t,*} \mid x^j_{t-1|t-1}, y_t)}.$$

# Auxiliary particle filter, ctd.

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

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

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

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

then the importance weights are indeed proportional to

$$\frac{g(y_t \mid x^j_{t,*})k(x^j_{t,*} \mid x^j_{t-1|t-1})}{\tau(x^j_{t-1|t-1}, y_t)q(x^j_{t,*} \mid x^j_{t-1|t-1}, y_t)}.$$

# Implementing the auxiliary particle filter

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

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

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

# Likelihood estimation

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

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

Hence we have the estimate

$$\widehat{p}(y_t \mid y_{t-1}) = \frac{1}{N}\sum_{j=1}^{N} g(y_t \mid x^j_{t|t-1})$$

which we compute when normalizing the weights.

If we use the auxiliary particle filter, we have similarly

$$\widehat{p}(y_t \mid y_{t-1}) = \frac{1}{N}\sum_{j=1}^{N} \tilde{w}(x^j_{t-1|t-1}, x^j_{t,*}, y_t).$$

# Likelihood estimation is unbiased

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

**Lemma** If $x^j_0$ is sampled from the initial distribution of $X_0$ and if resampling is unbiased, then

$$\mathbb{E}\Big(\prod_{t=1}^{n} \widehat{p}(y_t \mid y_{1:t-1})\Big) = \prod_{t=1}^{n} p(y_t \mid y_{1:t-1}) = p(y_{1:n}),$$

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

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

## Properties of particle filter

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

## Importance splitting as a particle filter

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

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

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

## Implementing importance splitting

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

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

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

## Section 5

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters
Basic idea
Square root and perturbed observation versions
Fine tuning
Bridging the two methods

Extensions of particle filters

## Basic idea of Ensemble Kalman filter

Assume Gaussian-linear observations

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

Prediction step as for particle filter, but update as if $\pi_{t|t-1}$ is Gaussian with mean $\mu_{t|t-1}$ and covariance $P_{t|t-1}$.

Then $\pi_{t|t}$ is again Gaussian with mean

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

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

## Square root and stochastic version

The Ensemble Kalman filter comes in two versions:

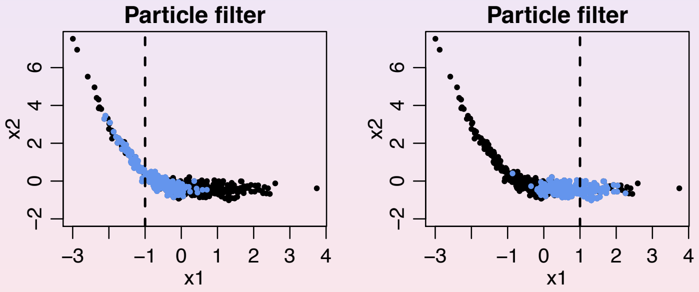
- Stochastic: Perturbed observation EnKF

$$x^j_{t|t} = x^j_{t|t-1} + \widehat{K}_t(y_t + \varepsilon^j_t - Hx^j_{t|t-1}), \quad \varepsilon^j_t \sim \mathcal{N}(0, R)$$

  where $\widehat{K}_t = K(H, \widehat{P}_{t|t-1}, R)$. This is the update for the mean with artificial noise added to the observation.
- Deterministic: Ensemble square root filter. It applies an affine linear transformation to $(x^j_{t|t-1})$.

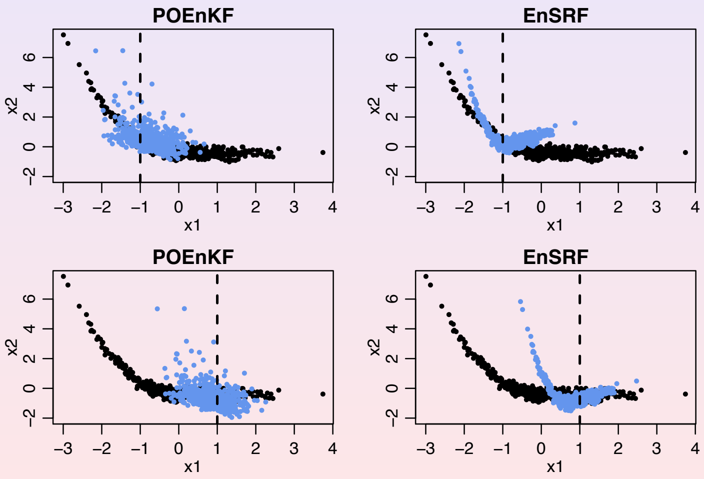
## Particle filter for banana-shaped prediction

A single observation $y = x_1 + \mathcal{N}(0, 0.5^2) \in \{-1, 1\}$. Banana-shaped prediction sample (black). Particle filter update (blue).



## Ensemble Kalman filter for banana-shaped prediction

A single observation $y = x_1 + \mathcal{N}(0, 0.5^2) \in \{-1, 1\}$. Banana-shaped prediction sample (black). Ensemble Kalman filter update (blue).



# Comparison of deterministic and stochastic version

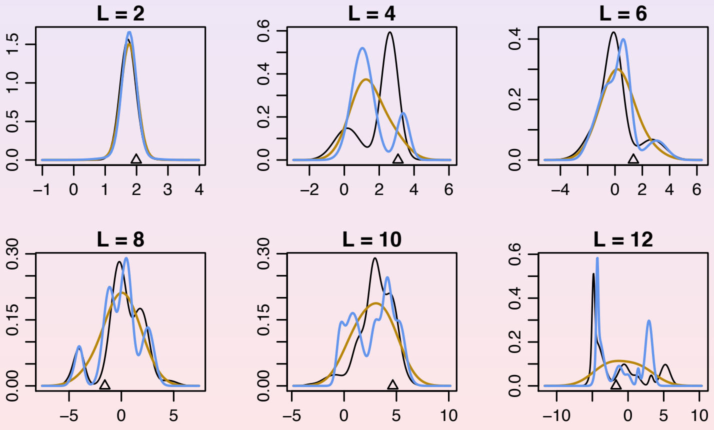
Suppose that $\pi_{t|t-1}$ is not Gaussian. What can we say about the limiting distribution as $N \to \infty$? Because prediction sample is modified, the limit distribution is not Gaussian.

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

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

# Comparing the two versions: Examples

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



# Properties of the EnKF

Both versions:

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

Stochastic:

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

Deterministic:

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

# Fine tuning

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

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

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

## Covariance inflation

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

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

This leaves the prediction mean unchanged, but inflates the sample covariance: $\widehat{\mathrm{Cov}}(\tilde{x}_{t|t-1}) = \delta^2 \cdot \widehat{\mathrm{Cov}}(x_{t|t-1})$. It can be used for XEnKF and EnKPF.

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

## Localization

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

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

## Covariance tapering

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

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

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

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

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

## Bridging the two methods: Overview

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

## Updates of Gaussian mixtures

Explicit updates are possible if $\pi_{t|t-1}$ is a Gaussian mixture (GM). $\pi_{t|t}$ is again a Gaussian mixture:

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

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

$$\alpha^{\ell}_{t|t} \propto \alpha^{\ell}_{t|t-1} \times \varphi(y | H\mu^{\ell}_{t|t-1}, HP^{\ell}_{t|t-1}H' + R)$$

($\varphi$ denotes the Gaussian density). "Particle filter for the weights, Kalman filter for mean and variance".

## Bridging between EnKF and PF: EnKPF

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

$$\pi_{t|t}(dx) \propto \pi_{t|t-1}(dx) p(y|x)^{\gamma} p(y|x)^{1-\gamma} \quad (0 \le \gamma \le 1).$$

Step 1: From $\pi_{t|t-1}(dx)$ to $\pi^{\gamma}_{t|t}(dx) \propto \pi_{t|t-1}(dx) p(y|x)^{\gamma}$ by PoEnKF:
Simply replace $R$ by $R/\gamma$ or $\hat{P}_{t|t-1}$ by $\gamma \hat{P}_{t|t-1}$ in the Kalman gain.

Step 2: From $\pi^{\gamma}_{t|t}(dx)$ to $\pi_{t|t}(dx) \propto \pi^{\gamma}_{t|t}(dx) p(y|x)^{1-\gamma}$ by particle filter.

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

## EnKF as a Gaussian mixture

We can consider $x^{j}_{t|t} = x^{j}_{t|t-1} + \hat{K}(y + \varepsilon^{j} - Hx^{j}_{t|t-1})$ $(j = 1, 2, \dots, N)$ as a balanced sample from the Gaussian mixture

$$\hat{\pi}_{t|t} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{N}(x^{j}_{p} + \hat{K}(y - Hx^{j}_{p}), \hat{K} R \hat{K}').$$

Hence the EnKF in step 1 without sampling gives

$$\hat{\pi}^{\gamma}_{t|t} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{N}(x^{j}_{p} + K(\gamma \hat{P}_{t|t-1})(y - Hx^{j}_{p}), \frac{1}{\gamma} K(\gamma \hat{P}_{t|t-1}) R K(\gamma \hat{P}_{t|t-1}))').$$

Bayes formula converts $\hat{\pi}^{\gamma}_{t|t}$ in the second step into another Gaussian mixture from which we sample to get $(x^{j}_{t|t})$.

## Choice of tuning parameter $\gamma$

The exponent $\gamma$ trades off systematic and Monte Carlo errors:

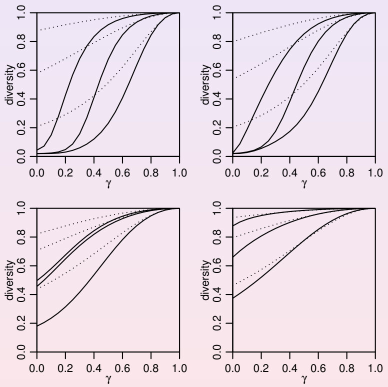
- $\gamma = 0$ corresponds to PF, $\gamma = 1$ corresponds to EnKF, with continuous interpolation in between.
- A small $\gamma$ means small bias but large Monte Carlo variance.
- A large $\gamma$ means large bias but small Monte Carlo variance.

Need to find an interesting compromise!

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

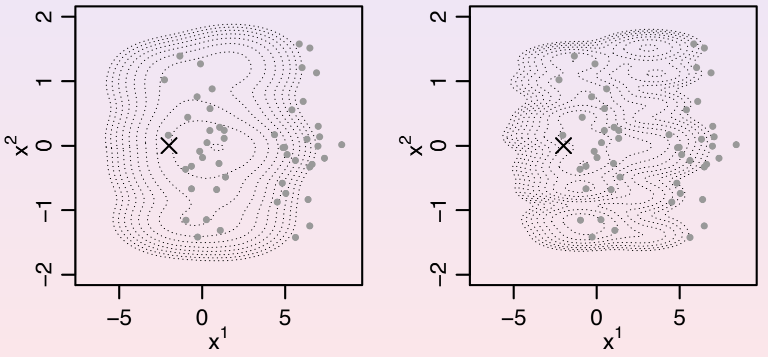
## Diversity as a function of $\gamma$

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



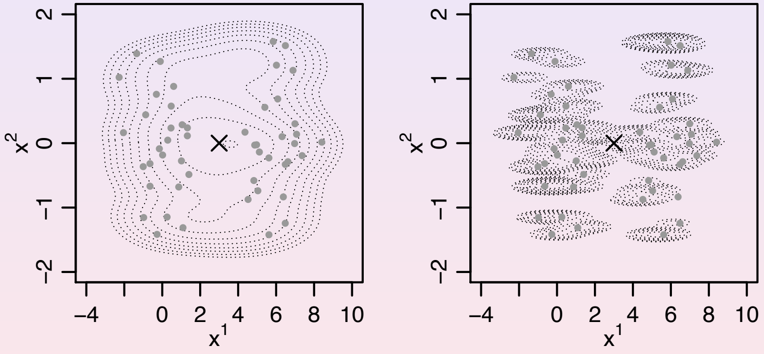
## Single update for bimodal prior I

Left: EnKF, Right: EnKPF, diversity $\approx$ 40%.
Dots: Prior sample, Dotted: Contours of underlying update density.



## Single update for bimodal prior II

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



## Section 6

Introductory Examples

Basics of state space models and filtering

Kalman filter and its applications

Particle filters

Ensemble Kalman filters

Extensions of particle filters
- Particle smoothing
- Parameter estimation
- General sequential Monte Carlo

## Smoothing by filtering of paths

We can implement the recursion

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

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

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

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

But 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 $\pi_{0:t|t}$.

## Forward filtering, backward smoothing

We can combine the formula

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

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

$$\mathbb{P}(x^j_{s|t} = x^i_{s|s} \mid x^j_{s+1|t}) = \frac{k(x^j_{s+1|t} \mid x^i_{s|s})}{\sum_{\ell=1}^N k(x^j_{s+1|t} \mid x^\ell_{s|s})}.$$

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

## Smoothing by accept/reject

We can get rid of both problems by using the approximation

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

instead of the empirical distribution of $(x^j_{t|t})$. This means that we sample $x^j_{s|t}$ given $x^j_{s+1|t}$ from the density which is proportional to

$$k(x^j_{s+1|t} \mid x_s) g(y_s \mid x_s) \sum_{i=1}^N k(x_s \mid x^i_{s-1|s-1}).$$

Since we need only one draw from this distribution, we have to use the accept/reject method: Propose an index $I_*$ and a value $x_{s,*}$ according to $\tau(i)q(x_s \mid x^i_{s|s})$ and accept $x_{s,*}$ as $x^j_{s|t}$ with the appropriate probability.

## Smoothing using the two filter formula

A different approach is based on the formula

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

The second factor is unknown, but it satisfies the recursion

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

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

## Approximating the 2-filter formula

We can turn this into a recursion for densities by setting

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

where $\gamma_s$ is a density s.th. the denominator is finite.

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

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

$$\sum_{i=1}^{N} k(x_s \mid x^i_{s-1|s-1}) g(y_s \mid x_s) \sum_{j=1}^{N} \frac{k(\tilde{x}^j_{s+1|s+1} \mid x_s)}{\gamma_{s+1}(\tilde{x}^j_{s+1|s+1})}.$$

## An $O(N)$ smoothing algorithm

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

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

## Parameters as components of the state

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

The easiest method is to include $\theta$ as a deterministic component of the state:

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

where $\Delta$ is a point mass. The particle filter degenerates quickly because there is no rejuvenation of the $\theta$-component.

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

## Particle MCMC

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

$$(x^*_{0:t}, \theta^*) \mid (x_{0:t}, \theta) \sim q(\theta^* \mid \theta) d\theta^* P(dx_{0:t} \mid y_{1:t}, \theta^*)$$

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

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

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

## Sampling from moving targets

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

In particular, if the target $\pi$ is difficult to sample, one can start with a simpler distribution $\pi_0$ and construct an approximating sequence such that $\pi_n = \pi$. Examples are importance splitting as discussed in the Introduction, or simulated tempering where

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

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

## Sequential sampling

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

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

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

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

## Resampling pairs of particles

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

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

## Choice of the transitions

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

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

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

# The case of the particle filter

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

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

the weights are explicit up to a normalizing constant

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

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