Stochastic Simulation

Autumn Semester 2024

Prof. Fabio Nobile *Assistants: Matteo Raviola, Eliott Van Dieren*

Project - 8

Submission deadline: 16 January 2025

Particle Filters

The goal of the project is to implement different numerical algorithms for data assimilation both in the framework of a linear and non-linear dynamical system. A systematic integration of data into mathematical models has been one of the central research challenges for the computational sciences in the twenty-first century. When the underlying mathematical model is a dynamical system and the data can be ordered in time, we refer to this process as data assimilation. In this project we will focus on the problem of discrete filtering, i.e. data is obtained as an observation of some quantities related to the true state of the system at some discrete time instants. The question is how to integrate this data into the model to improve the prediction of future evolution. Some applications include weather forecasts and oceanography. A proper mathematical description of this problem is done by means of Bayesian inference, and we refer the reader to [4]. We give hereafter a brief and intuitive description of the discrete filtering problem.

1 Theoretical background

Given a sequence of times $0 = t_0 < t_1 < \cdots < t_J$, we want to identify the state of the system $u_0, u_1, \ldots, u_J \in \mathbb{R}^d$ at those times. We adopt a Bayesian approach, considering $u_0, \ldots, u_J$ as random whose prior distribution is governed by a stochastic dynamical system

$$u_{j+1} \sim \mathbb{P}(\cdot|u_j) \qquad j \geq 0, \tag{1}$$

$$u_0 \sim \rho_0, \tag{2}$$

with $\mathbb{P}$ being a Markov transition density function. Hereafter, we assume that the probability distribution of each state u_j has a density which we call ρ_j. Equations (1)-(2) fully characterize the prior density ρ_j of each state u_j. Suppose now that some noisy observations of the system are available at time t_j in the form

$$y_j \sim \nu(\cdot \,|\, u_j). \tag{3}$$

A common model is

$$y_j = h(u_j) + \eta_j, \tag{4}$$

with $h \in C^0(\mathbb{R}^d; \mathbb{R}^k)$, $k \leq d$, an observation operator and $\eta_j \overset{\text{iid}}{\sim} \tilde{\nu}$ an additive random measurement error. In this case one has $\nu(y_j \,|\, u_j) = \tilde{\nu}(y_j - h(u_j))$. The goal of filtering is to "correct" the prior density ρ_j of u_j with the knowledge of the observations $Y_j := \{y_i\}_{i=1}^j$, i.e. to obtain the *conditional* density $\pi_j(u_j) := \mathbb{P}(u_j \,|\, Y_j)$, also known as the *filtering distribution*[1]. Interestingly, the filtering distribution can be computed recursively using a two-step updating procedure. Let us introduce

[1] With some abuse of notation, $\mathbb{P}(u_j \,|\, Y_j)$ denotes the conditional probability density function of u_j given Y_j.

$\hat{\pi}_j := \mathbb{P}(u_j \,|\, Y_{j-1})$, then we have

$$\begin{aligned}\hat{\pi}_{j+1} &= \mathbb{P}(u_{j+1} \,|\, Y_j) \\ &= \int \mathbb{P}(u_{j+1} \,|\, Y_j, u_j)\mathbb{P}(u_j \,|\, Y_j) du_j \\ &= \int \mathbb{P}(u_{j+1} \,|\, u_j)\pi_j(u_j) du_j,\end{aligned} \tag{5}$$

which constitutes the *prediction step.* To obtain the final filtering distribution π_{j+1}, we have

$$\begin{aligned}\pi_{j+1}(u_{j+1}) &= \mathbb{P}(u_{j+1} \,|\, Y_{j+1}) \\ &= \frac{\mathbb{P}(u_{j+1}, y_{j+1} \,|\, Y_j)}{\mathbb{P}(y_{j+1} \,|\, Y_j)} \\ &= \frac{\mathbb{P}(y_{j+1} \,|\, Y_j, u_{j+1})\mathbb{P}(u_{j+1} \,|\, Y_j)}{\mathbb{P}(y_{j+1} \,|\, Y_j)} \\ &= \frac{1}{Z}\nu(y_{j+1} \,|\, u_{j+1}), \qquad Z = \mathbb{P}(y_{j+1} \,|\, Y_j),\end{aligned} \tag{6}$$

which is known as the *analysis step.* In conclusion, given $\pi_0 = \rho_0$, we compute recursively π_{j+1}, $j = 0, \ldots, J-1$, as

$$\hat{\pi}_{j+1}(\cdot) = \int \mathbb{P}(\cdot \,|\, u_j)\pi_j(u_j) du_j \qquad \text{(Prediction)} \tag{7}$$

$$\pi_{j+1}(\cdot) = \frac{1}{Z}\nu\left(y_{j+1} \,|\cdot\right)\hat{\pi}_{j+1}(\cdot) \qquad \text{(Analysis).} \tag{8}$$

In the case of a linear system (1), additive noise as in (4) with a linear observation operator h and Gaussian noise, Gaussian initial condition, the conditional distribution is Gaussian for any j and can be written explicitly, leading to the so-called *Kalman filter.* In the nonlinear or non-Gaussian case, several methods have been proposed in the literature to approximate the conditional distribution π_j, such as the so-called *Ensemble Kalman Filter* (EnKF) [2, 5] and the bootstrap particle filter [6].

Kalman Filter

The *Kalman Filter* is a widely used method for estimating the state of a linear dynamical system with linear observations and Gaussian noise:

$$u_{j+1} = A_j u_j + b_j, \qquad b_j \sim \mathcal{N}(m_j, C_j), \qquad u_0 \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{9}$$

$$y_j = H u_j + \eta_j, \qquad \eta_j \sim \mathcal{N}(0, \Gamma). \tag{10}$$

In this case, the filtering distributions π_j and $\hat{\pi}_j$ both remain Gaussian at each time step:

$$\pi_j = \mathcal{N}(\mu_j, \Sigma_j), \quad \hat{\pi}_j = \mathcal{N}(\hat{\mu}_j, \hat{\Sigma}_j), \tag{11}$$

and can be characterized as:

- **Prediction step:**

$$\begin{aligned}\hat{\mu}_{j+1} &= A_j \mu_j + m_j, \\ \hat{\Sigma}_{j+1} &= \Sigma_j A_j^\top + C_j.\end{aligned}$$

- **Analysis step:**

$$\begin{aligned}\mu_{j+1} &= \hat{\mu}_{j+1} + K_{j+1} d_{j+1}, \\ \Sigma_{j+1} &= A_j \Sigma_j A_j^\top + C_j,\end{aligned}$$

with

$$d_{j+1} = y_{j+1} - H\hat{\mu}_{j+1}, \quad S_{j+1} = H\hat{\Sigma}_{j+1}H^\top + \Gamma, \quad K_{j+1} = \hat{\Sigma}_{j+1}H^\top S_{j+1}^{-1}.$$

We denote by d_{j+1} the innovation residual, S_{j+1} the innovation covariance and K_{j+1} the Kalman gain. The resulting estimates are optimal under the assumptions of linearity and Gaussianity, making the Kalman Filter efficient and accurate in such contexts.

Ensemble Kalman Filter (EnKF)

The *Ensemble Kalman Filter (EnKF)* provides an approximation to the Kalman filter by approximating the distribution π_j with an ensemble of particles $\{u_j^{(i)}\}_{i=1}^N$. Namely, if we assume equal weights between the particles, we obtain $\pi_j^N := \frac{1}{N}\sum_{i=1}^N \delta_{u_j^{(i)}} \approx \pi_j$. The particles are propagated according to the underlying dynamics and are then updated collectively based on observed data. This approach extends to the nonlinear setting by evolving the particles according to the nonlinear dynamics (1) and assimilating general observations (3). The EnKF proceeds as follows:

- **Prediction:** Each particle $u_j^{(i)}$ is advanced to the next time step by sampling from the transition kernel $\mathbb{P}(\cdot \mid u_j^{(i)})$, thus obtaining the forecast ensemble $\{\hat{u}_{j+1}^{(i)}\}_{i=1}^N$.
- **Analysis:** Given an observation y_{j+1}, the forecast particles are adjusted by a Kalman-like gain computed from the ensemble covariance, bringing each member closer to the observation. That is, we sample $y_{j+1}^{(i)} \sim \nu(\cdot \mid \hat{u}_{j+1}^{(i)})$ for $i = 1, \dots, N$ and obtain

$$u_{j+1}^{(i)} = \hat{u}_{j+1}^{(i)} + \bar{K}_{j+1}\left(y_{j+1} - y_{j+1}^{(i)}\right), \quad i = 1, \dots, N.$$

  where $\bar{K}_{j+1}$ solves $\bar{K}_{j+1}\tilde{Y}_{j+1}\tilde{Y}_{j+1}^\top = \tilde{U}_{j+1}\tilde{Y}_{j+1}^\top$, where, denoting $\mathbb{1} = [1, \dots, 1]^\top$,

$$\begin{aligned}\tilde{Y}_{j+1} &= \left[y_{j+1}^{(1)}, \dots, y_{j+1}^{(N)}\right]\left(I_N - \frac{1}{N}\mathbb{1}\mathbb{1}^\top\right), \\ \tilde{U}_{j+1} &= \left[\hat{u}_{j+1}^{(1)}, \dots, \hat{u}_{j+1}^{(N)}\right]\left(I_N - \frac{1}{N}\mathbb{1}\mathbb{1}^\top\right).\end{aligned}$$

The EnKF thus provides a practical way to approximate π_j even for high-dimensional or nonlinear problems, where the exact Kalman filter is not feasible. We refer to [5] for further details about the aforementioned computations.

Bootstrap Particle Filter

The *Bootstrap Particle Filter*, or sequential importance sampling filter, is a method for estimating π_j in nonlinear, non-Gaussian systems by approximating the filtering distribution with a set of weighted particles $\{u_j^{(i)}, w_j^{(i)}\}_{i=1}^N$. Each particle represents a potential state, and its weight $w_j^{(i)}$ indicates its likelihood given the observations. In what follows, π_j^N denotes the empirical measure $\pi_j^N = \sum_{i=1}^N w_j^{(i)}\delta_{u_j^{(i)}}$ The procedure is as follows:

- **Resampling:** Draw independently N new particles $\tilde{u}_j^{(i)} \overset{\text{iid}}{\sim} \pi_j^N$, $i = 1, \dots, N$, with $\pi_0^N = \rho_0$.
- **Prediction:** Each particle $\tilde{u}_j^{(i)}$ is propagated forward by sampling from $\mathbb{P}(\cdot \mid \tilde{u}_j^{(i)})$, yielding forecasted particles $\hat{u}_{j+1}^{(i)}$.

- **Analysis:** After observing y_{j+1}, we set $u_{j+1}^{(i)} = \hat{u}_{j+1}^{(i)}$ and update each particle's weight according to the likelihood of the observation, i.e., $w_{j+1}^{(i)} \propto \nu\left(y_{j+1} \,\middle|\, u_{j+1}^{(i)}\right)$, $i = 1, \dots, N$, with $\sum_i w_{j+1}^{(i)} = 1$. Lastly, we update the distribution π_j^N for the next iteration as

$$\pi_{j+1}^N = \sum_{i=1}^N w_{j+1}^{(i)} \delta_{u_{j+1}^{(i)}}.$$

The Bootstrap Particle Filter offers flexibility for general, nonlinear, non-Gaussian systems, but it may require a large number of particles to achieve accurate results in high-dimensional systems.

2 Goals of the project

The objective of this project is to get acquainted with the filtering problem and, notably, the Kalman filter, Ensemble Kalman filter, and bootstrap particle filter approaches [3, 4, 5, 6].

2.a Linear model

Let us first consider the one dimensional linear model

$$\begin{aligned} u_{j+1} &= 0.9u_j + \xi_j, \quad u_0 \sim \mathcal{N}(0,1) \\ y_j &= 1.3u_j + \eta_j, \end{aligned} \tag{12}$$

where $\xi_j \sim \mathcal{N}(0, 0.5)$ and $\eta_j \sim \mathcal{N}(0, 0.1)$.

a.1 Simulate one realization of the process (12) for $J = 2000$ steps. Record the obtained values $Y_J = \{y_1, \dots, y_J\}$, which will be your observations. Likewise, record the obtained values $U_J = \{u_1, \dots, u_J\}$, which will be your "Truth" (true trajectory that you would like to recover from your noisy observations).

a.2 Implement the Kalman filter. Compare the result with the true trajectory computed in a.1 (you can plot the mean ± standard deviation as a function of j and compare it with the true solution). A naive filter for this problem is obtained by ignoring the noise in the observations and taking simply $\hat{u}_j = y_j/1.3$. Add to the previous plot also the naive filter and comment on the results. Is the Kalman filter performing better than the naive filter? Compute the integrated RMSE defined as

$$\frac{1}{J}\sum_{j=1}^J \mathbb{E}\left[\left(u_j^{\mathrm{KF}} - u_j\right)^2\right].$$

a.3 Implement the EnKF and the bootstrap particle filter using N particles. Produce similar plots as in the previous point for different values of N. What do you observe? Plot the integrated RMSE for both methods as a function of N and compare it with that of the Kalman filter.

a.4 In particle filter, the effective sample size at step j is defined as

$$\mathrm{ESS}_j = \frac{1}{N\sum_i^N (w_j^{(i)})^2},$$

where $w_j^{(i)}$ are the particles' weights. Compute and plot ESS as a function of j for different values of N. How does the plot change if you do not perform the resampling step in the algorithm?

2.b Nonlinear model

Let us consider the following stochastic volatility model (this example is borrowed from [7])

$$\begin{aligned} u_{j+1}|(u_j = u) &\sim \mathcal{N}(\phi u, \sigma^2) \\ y_j|(u_j = u) &\sim \mathcal{N}(0, \beta^2 e^u) \end{aligned} \tag{13}$$

In this model, u_j denotes the latent volatility of the price of some financial asset and y_j the observed scaled log-return of the asset. The file `synthetic_data.csv` (on Moodle) contains the values of $J = 500$ observations $Y_J = \{y_j\}_{j=1}^{500}$.

b.1 Consider the values $\phi = 0.98, \sigma^2 = 0.16, \beta^2 = 0.7$. Implement the bootstrap particle filter and the EnKF and compare the results.

b.2 Suppose now that ϕ is unknown (whereas $\sigma^2 = 0.16$ and $\beta^2 = 0.70$ are known) and we want to estimate it from the observations. The likelihood of ϕ given the observations Y_J can be written as

$$l(\phi \,|\, Y_J) = \mathbb{P}(Y_J \,|\, \phi) = \prod_{j=1}^{J} \mathbb{P}(y_j \,|\, Y_{j-1}, \phi) = \prod_{j=1}^{J} \int \mathbb{P}(y_j \,|\, \hat{u}_j, \phi)\hat{\pi}_j(\hat{u}_j \,|\, \phi) d\hat{u}_j,$$

where the notation $\hat{\pi}_j(\hat{u}_j \,|\, \phi)$ emphasizes the fact that ϕ is used in the forward model (13). A bootstrap particle filter estimator for $l(\phi \,|\, Y_J)$ is then

$$\hat{l}(\phi \,|\, Y_J) = \prod_{j=1}^{J} \frac{1}{N} \sum_{k=1}^{N} \nu\left(y_j \,|\, \hat{u}_j^{(k)}\right).$$

Take a reasonably coarse grid for ϕ between 0 and 1, and use the bootstrap particle filter to estimate the likelihood of ϕ given the data for each of these values of ϕ. Compute 10 estimates for each grid point and plot all of them in one figure as, e.g., boxplots (ϕ on the horizontal axis, log-likelihood estimates on the vertical axis).

b.3 Define the estimator

$$\hat{l}_{j:J} = \prod_{l=j}^{J} \frac{1}{N} \sum_{k=1}^{N} \nu\left(y_j \,|\, \hat{u}_j^{(k)}\right),$$

and let A_j be the σ-algebra generated by all random variables used up to step j, e.g. for resampling particles at steps $k < j$ and advancing particles from step $k < j$ to step $k+1$. Notice that, conditional on A_j, the particles $\hat{u}_{j+1}^{(k)}$ are i.i.d. (they are drawn independently from the empirical distribution $\sum_k w_j^{(k)} \delta_{u_j^{(k)}}$ and evolved independently). Prove by backward induction (from $j = J$ to $j = 1$) that

$$\mathbb{E}\left[\hat{l}_{j:J} \,|\, A_{j-1}\right] = \sum_{i=1}^{N} \mathbb{P}(y_j, \ldots, y_J \,|\, u_{j-1}^{(i)}, \phi) w_{j-1}^{(i)}.$$

Conclude that $\mathbb{E}[\hat{l}_J(\phi \,|\, Y_J)] = \mathbb{P}(Y_J \,|\, \phi)$, i.e. the bootstrap particle filter likelihood estimator is unbiased.

b.4 Assume now instead that the variance parameters σ^2 and β^2 are unknown (whereas $\phi = 0.985$ is known). We consider a Bayesian setting and place inverse Gamma priors on the parameters

$$\sigma^2 \sim IG(0.01, 0.01), \qquad \beta^2 \sim IG(0.01, 0.01) \tag{14}$$

where IG is the inverse Gamma distribution[2]. Implement the particle Metropolis-Hastings (see [1]) algorithm to compute the posterior distribution $\mathbb{P}(\sigma^2, \beta^2 \,|\, Y_J)$, using a Gaussian random walk proposal. That is, run a Random walk Metropolis-Hastings algorithm to sample from the posterior distribution $\mathbb{P}(\sigma^2, \beta^2 \,|\, Y_J)$ where for each candidate state (σ^2, β^2) the likelihood is estimated by a particle filter. Monitor the convergence of the Markov chain and plot the marginal distributions of σ^2 and β^2 (histograms of empirical CDF). Assess the influence of the number of particles on the result.

References

[1] Johan Dahlin and Thomas B Schön, *Getting started with particle Metropolis-Hastings for inference in nonlinear dynamical models*, Journal of Statistical Software **88** (2019), 1–41.

[2] Geir Evensen, *Data assimilation*, second ed., Springer-Verlag, Berlin, 2009, The ensemble Kalman filter. MR 2555209

[3] A. Iglesias, K. Law, and A. Stuart, *Ensemble Kalman methods for inverse problems*, Inverse Problems **29** (2013), no. 4.

[4] K. Law, A. Stuart, and K. Zygalakis, *Data assimilation*, Springer, 2015.

[5] Michael Roth, Gustaf Hendeby, Carsten Fritsche, and Fredrik Gustafsson, *The ensemble kalman filter: a signal processing perspective*, EURASIP Journal on Advances in Signal Processing **2017** (2017), 1–16.

[6] Daniel Sanz-Alonso, Andrew Stuart, and Armeen Taeb, *Inverse problems and data assimilation*, London Mathematical Society Student Texts, Cambridge University Press, 2023.

[7] Thomas B. Schön and Fredrik Lindsten, *Sequential monte carlo methods*, Course at Uppsala University, 2019.

[2] We recall that the inverse Gamma distribution with parameters (a, b) is given by

$$IG(x; a, b) = \frac{b^a}{\Gamma(a)} \left(\frac{1}{x}\right)^{1+a} e^{(-b/x)}$$