Project LINMA1731: Tracking a ship at see

Philémon Beghin, Jehum Cho

08/03/2023

1 Part 1: Acceleration Noise Estimation

In this project, we would like to navigate a boat on the sea. Sea breeze and waves make it more difficult to track the boat than it would be on a lake. As the first part of our project, we want to estimate the random acceleration noise that is caused by these uncontrollable factors. We assume that this noise at time step k is a random variable w_k following a Gumbel distribution. This distribution is characterized by two parameters: a location parameter μ, and a scale parameter $\beta > 0$. The cumulative distribution function (CDF) of a Gumbel distribution with the parameters μ and β is given as:

$$F(x; \mu, \beta) = e^{-e^{-(x-\mu)/\beta}}. \tag{1}$$

The PDF of the distribution is given as:

$$f(x; \mu, \beta) = \frac{1}{\beta} e^{-(z+e^{-z})}, \tag{2}$$

where $z = (x - \mu)/\beta$.

Let us further assume that the parameters are location invariant and time invariant. In other words, $\forall\ k, w(k)$ have the same parameters. The purpose of the first part of our project is to estimate these two parameters from empirical observations.

First, let us assume that the scale parameter is known as $\beta = 1$. Let $w_i, i = 1, \cdots, N$ be the realizations from the distribution $\mathbf{Gumbel}(\mu, 1)$.

a) Derive the maximum likelihood estimator of μ based on these N observations.

b) Show that this estimator is consistent. (Hint: Use the weak law of large numbers. Use the Gamma function for the integration.)

Now, let us assume that the two parameters μ, β are both unknown. Let $W_i, i = 1, \cdots, N$ be the realizations from $\mathbf{Gumbel}(\mu, \beta)$.

c) Explain theoretically how to jointly derive maximum likelihood estimators of the parameters of the distribution based on N observations.

d) Generate N samples from $\mathbf{Gumbel}(1, 2)$. Compute the ML estimators $\hat{\mu}_N, \hat{\beta}_N$ with different sizes of N using the methods you explained in c). Repeat the experiments M times (e.g. 500 times). Plot the results and comment on it.

e) It is known that the Cramér-Rao bounds for the parameters are as follows:

$$\sigma^2_{\hat{\mu}_N} \geq \frac{6\beta^2}{N\pi^2}(1 + \pi^2/6 + \gamma^2 - 2\gamma), \tag{3}$$

where γ is the Euler's constant ($\gamma \simeq 0.5772$),

$$\sigma^2_{\hat{\beta}_N} \geq \frac{6\beta^2}{N\pi^2}. \tag{4}$$

Show numerically that the ML estimators are (asymptotically) efficient, i.e. the empirical variances converges to the Cramér-Rao lower bounds as N increases, by conducting similar experiments as in d). (Try sufficiently high number of experiments M, for instance $M = 10^4$.)

f) Approximation to a Normal Distribution. The mean and the variance of the Gumbel distribution $\mathbf{Gumbel}(\mu, \beta)$ are given as follows:

$$\mathbf{Mean} = \mu + \beta\gamma, \tag{5}$$

$$\mathbf{Variance} = \frac{\pi^2}{6}\beta^2, \tag{6}$$

where γ is Euler's constant. It is known that Gumbel distributions resemble Normal distributions with the same mean and variance, even though its right tail is fatter than that of Normal distribution. This information will be useful for the second part of the project where we assume that the error terms in the system equations follow a Normal distribution for Kalman Filter implementation. Check this fact by comparing the plots of these two distributions with different parameters.

2 Part 2: Kalman Filter and Particle Filter Comparison

2.1 Context

The purpose of the second part of the project is to track a boat sailing on the see using a filtering algorithm. The true 2D position of the ship at time k is given by s_k. However, you only have access to noisy observations Y_k such that

$$Y_k = s_k + v_k$$

where $v_k \sim \mathcal{N}(0, \sigma_v^2)$ is a white noise representing a GPS positioning error. The ship velocity and acceleration at time k are respectively $\dot{s}_k$ and a_k. They can be obtained from the equations of uniformly accelerated motion:

$$\begin{aligned} s_{k+1} &= s_k + \dot{s}_k dt + \frac{1}{2} a_k dt^2 \\ \dot{s}_{k+1} &= \dot{s}_k + a_k dt \end{aligned}$$

where dt is the time increment. The acceleration a_k is composed of a manoeuvring acceleration u_k (which we control, it is the input of the system) and a random acceleration $w_k \sim$ **Gumbel**(μ, β) that you studied in the first part (this random acceleration is caused by the see breeze and waves, for instance). It follows

$$a_k = u_k + w_k.$$

The state X_k of the system at time k is defined as the position and speed of the ship

$$X_k = \begin{pmatrix} s_k \\ \dot{s}_k \end{pmatrix}.$$

Hence, the state equation is

$$\begin{pmatrix} s_{k+1} \\ \dot{s}_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & dt \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_k \\ \dot{s}_k \end{pmatrix} + \begin{pmatrix} \frac{1}{2} dt^2 \\ dt \end{pmatrix} u_k + \begin{pmatrix} \frac{1}{2} dt^2 \\ dt \end{pmatrix} w_k$$

while the observation equation is

$$Y_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} s_k \\ \dot{s}_k \end{pmatrix} + v_k.$$

Developing the system using the position $s_k = (x_k, y_k)$ and the velocity $\dot{s}_k = (\dot{x}_k, \dot{y}_k)$, we finally write:

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{pmatrix} + \begin{pmatrix} \frac{1}{2} dt^2 & 0 \\ 0 & \frac{1}{2} dt^2 \\ dt & 0 \\ 0 & dt \end{pmatrix} u_{k,(2\times 1)} + \begin{pmatrix} \frac{1}{2} dt^2 & 0 \\ 0 & \frac{1}{2} dt^2 \\ dt & 0 \\ 0 & dt \end{pmatrix} w_{k,(2\times 1)}$$

and

$$Y_k = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{pmatrix} + v_{k,(2\times 1)}$$

The derivation of the model has been inspired from [1].

2.2 Questions

1. **Kalman Filter.** First implement a Kalman filter to estimate the position of the boat at sea (i.e. the (x, y) coordinates). You are provided with the true position of the system, the noisy observations and the input of the system. Use a time step $dt = 0.1$ sec and a number of iterations $N = 100$. Then comment on the performances of the Kalman filter. You can compute the norm of the error on the partial state s_k at each time step or use the Mean Square Error (MSE) as a global performance estimator. The MSE is given by

$$MSE = \frac{1}{N}\sum_{k=1}^{N} ||s_k - \hat{s}_k||_2^2$$

   where s_k is the true position of the boat at time k and $\hat{s}_k$ is the estimated position you derived with the filter. For references, cfr. Slides PAA06&07.

   Note that for the implementation of the Kalman filter derived in the lecture notes, we assume that w_k is a zero-mean white noise that follows $w_k \sim \mathcal{N}(0, Q)$. In this project, we assume that $w_k \sim$ **Gumbel**(μ, β). Therefore, for the implementation of the Kalman filter (not the particle!), you will have to slightly modify the prediction step by adding the "shifting" term $G\begin{pmatrix}\mu+\beta\gamma\\ \mu+\beta\gamma\end{pmatrix}$, where μ and β are the Gumbel distribution parameters and γ is the Euler's constant ($\gamma \approx 0.5772$). Hence, the prediction equation becomes

$$\hat{X}_{k|k-1} := A\hat{X}_{k-1|k-1} + Bu_k + G\begin{pmatrix}\mu+\beta\gamma\\ \mu+\beta\gamma\end{pmatrix}.$$

   We still consider that the realisations of the random acceleration are independent from each other, but do not forget that the variance of the Gumbel distribution is not the same as the Gaussian distribution. Hence, the covariance matrix Q is given by $Q = \frac{\pi^2}{6}\beta^2 GG^T$.

2. **Particle Filter.** Next, implement a particle filter to estimate the position of the boat. Use the same noisy observations and input as in the previous question, then apply the filter to estimate the true positions. Start with a time step $dt = 0.1$ sec, a number of particles $N_p = 1000$ and a number of iterations $N = 100$. Then, comment on the performances of the particle filter by changing the number of particles. You can also use the MSE or the norm of the error at each time step. For references, cfr. Slide PAA04.

   In the particle filter algorithm, you have to compute weights at each time iteration and resample them. There are many possibilities for both the generation of the weights and the resampling. Here we ask you to proceed as in the Particle Filter example script covered during the course. In other words, to compute the weight vector ω at each iteration i, use a normal output noise $\omega = f(x)$, where $x = y_{i+1} - \tilde{x}_{i+1}$ is the vector of the difference between the observation y_{i+1} and the prediction of the particles (before resampling) and f is the probability density function of the standard normal distribution (i.e. with parameters $\mu = 0$ and $\sigma_v^2 = 1$). For resampling, return a new random list of N_p particles whose probability density is given by their weight computed previously. The function `random.choices()` might help you.

3. **Comparison.** Compare and comment the performances of both filters with respect to each other regarding interesting features.

Practical details related to the implementation.

1. Your implementation must be done in Python. Write your code in a single file `Project_1731_Part2.py`. Make sure it is commented and clear enough to be understood at first sight.
2. You are provided with three files: `True_Data.txt`, `Observations.txt` and `Input.txt`, which contain respectively the true position and velocity of the ship X_{true} for 100 time iterations, the noisy observations Y and the input u of the system. Use these data to assess the performances of your filters. The function `numpy.loadtxt()` will help you.
3. For the observation noise, consider that $v_k \sim \mathcal{N}(0, \sigma_v^2 = 1)$.
4. For the random acceleration, consider that $w_k \sim \textbf{Gumbel}(\mu = 1, \beta = 1)$.
5. In both filters, you will need to compute a first estimate X_0 of the initial state. For the Kalman filter, consider that $X_0 = (s_0, \dot{s}_0)$ such that $s_0 \sim \mathcal{N}(10, 50)$ and $\dot{s}_0 \sim \mathcal{N}(10, 10)$. For the particle filter implementation, first generate N_p particles with the same initial distribution and then average them to compute X_0.
6. For the Kalman filter implementation, you need to know the initial covariance matrix P_0 between the state variables. Use

$$P_0 = \begin{pmatrix} P_{0,pos} & 0 & 0 & 0 \\ 0 & P_{0,pos} & 0 & 0 \\ 0 & 0 & P_{0,vel} & 0 \\ 0 & 0 & 0 & P_{0,vel} \end{pmatrix},$$

   with $P_{0,pos} = 50$ and $P_{0,vel} = 10$.
7. Finally, due to the randomness of the process we studied, you might better generalise your results for a great amount of experiences, e.g. $M = 100$ experiments (or even more).
8. For your information, the initial conditions set in `True_Data.txt` are $(x_0, y_0, \dot{x}_0, \dot{y}_0) = (0, 0, 0.1, 0.1)$ and the input acceleration $u_k = (u_{k,x}, u_{k,y})$ has been generated such that $u_{k,x} \sim \mathcal{N}(3, 50)$ and $u_{k,y} \sim \mathcal{N}(1, 50)$. However, this information should not appear in your implementation.
9. To save your figures, use the function `plt.savefig()` rather than screenshots.

Practical details

Modality	The project is carried out by groups of two students. If you need to work alone or do not know anybody to work with, please contact us to find an arrangement. Each group must register on Moodle by Friday 17 March 2023, 18.15 pm.
Supervision	Office hours from week 7 to week 13: one hour of office hour on Tuesday 3 pm at the Euler building (room A.010) by Philémon Beghin and Friday 2 pm on Teams by Jehum Cho. For the office hours on Teams, we recommend you to send a clear description of your questions during the office hour (not earlier or later), and wait for the teaching assistant to respond you back. Please note that there can be several groups waiting for one teaching assistant, so that you might experience some delay.
Report	English is strongly recommended. But the course is French friendly, hence French is allowed without penalty. However, reports in French, if any, will have to go through a different grading procedure. The goal is not to evaluate your English skills and we will therefore not pay attention to the quality of the language, but to the scientific quality of your report instead. Maximum 10 pages for the total of both parts (e.g. 5 pages each).
Deadlines	Part I : Friday 28 April 2023 at 18.15 pm on Moodle. The report (in pdf format) and the code (in Python .py or .ipynb) will be submitted together in a zip file named `LINMA1731_2023_Project_Part1_NAME1_NAME2.zip`. Part II : Friday 19 May 2023 at 18.15 pm on Moodle. The report (in pdf format) and the code (in Python .py or .ipynb) will be submitted together in a zip file named `LINMA1731_2023_Project_Part2_NAME1_NAME2.zip`.

References

[1] Wanjin Xu et al, *Ship tracking based on the fusion of Kalman filter and particle filter*, J. Phys.: Conf. Ser. 2113 012017, 2021