Assignment 4: Kalman filtering and maksimum likelihood estimation

Instructions: The assignment is to be handed in via DTU Learn "FeedbackFruits" latest at March 15th at 23:59. You are allowed to hand in in groups of 1 to 4 persons. You must hand in a single pdf file presenting the results using text, math, tables and plots, only include code in the report where instructed to. Presenting values, e.g. of parameters, by copying from code output is ok. Arrange the report in sections and subsections according to the questions in this document. Indicate your student numbers on the report.

This document introduces the assignment. There are some reference in footnotes to scripts from exercises in Week 10 and 11, if you didn't do those exercises, it's a good way to get started, see under Week 10 here and Week 11 here. The scripts are in R, however they can relatively easy be coded similarly in other languages.

Data is in the `assignment4_2024.zip` file.

Introduction

Due to climate change the level of rain is increasing, both in frequency and intensity. Therefore much effort (and money!) is spend on improving handling of water, especially in urban aread. Modelling and control is used increasing to optimize the rainwater management. Models of the "rainfall-runofff", which is the process from the rainwater hits the surface until in enters collection bassins, from where it is let slowly into the sewer, are needed.

In this exercise you will set up a stochastic state-space model for modelling the rainfall-runoff. You will get data from four rain events[1] sampled every minute. The data is in files, one for each rain event.

Each event has two time series:

- u_t is the measured incomming rain scaled from units mm/min to match the area of approximately 1 km^2. The values given is in units 100 m^3/min.
- y_t is the measured water level in a bassin collecting the rain water of the area. The unit is 100 m^3.

A stochastic state-space model

$$\begin{aligned}\mathbf{X}_t &= \mathbf{A}\mathbf{X}_{t-1} + \mathbf{B}u_{t-1} + \mathbf{G}(\mathbf{X_{t-1}})\mathbf{e}_{1,t}\\ Y_t &= \mathbf{C}\mathbf{X}_t + e_{2,t}\end{aligned}$$

is used, where

- $\{\mathbf{e}_{1,t}\}$ is multi-variate and $\{e_{2,t}\}$ single-variate. Both are uncorrelated normally distributed white noise processes.

Note, that the model has "state-dependent" system noise (the $\mathbf{G}(\mathbf{X_{t-1}})$), which actually makes it non-linear, however it not complicated to implement as explained later.

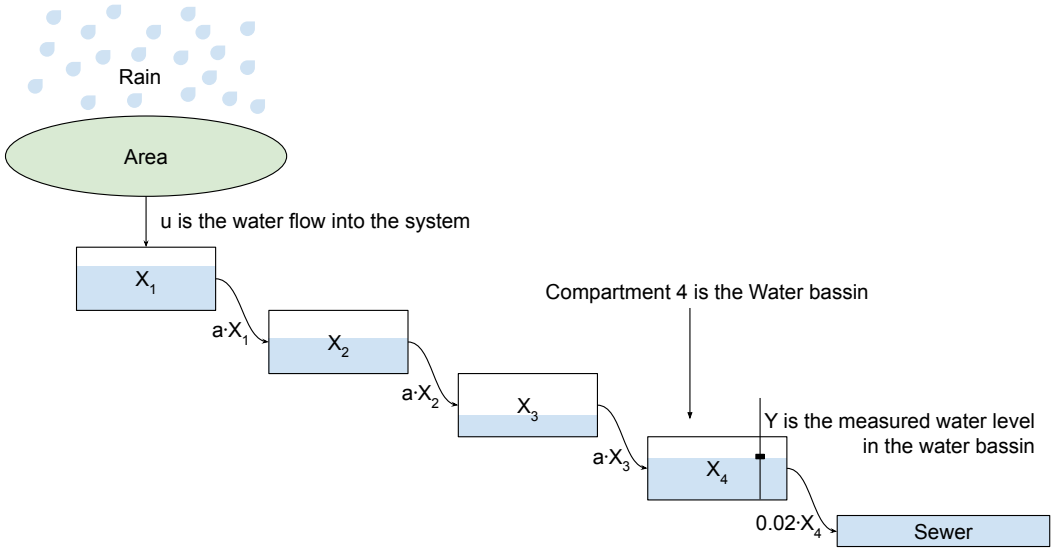
The particular model used have four states, hence $\mathbf{X}_t = [X_{1,t}\ X_{2,t}\ X_{3,t}\ X_{4,t}]^T$, and the transition matrix

$$\mathbf{A} = \begin{bmatrix} 1-a & 0 & 0 & 0 \\ a & 1-a & 0 & 0 \\ 0 & a & 1-a & 0 \\ 0 & 0 & a & 0.98 \end{bmatrix}$$

where a is the transition rate, which determines how fast the rain water runs from one state to the next.

Actually, such a model is called a "compartment model", because the rain enters into a compartment from which it flows through several compartments until it enters the final compartment – the bassin where the water level is measured. The model is illustrated by this diagram

[1] Note that all the data is simulated, hence not from real measurements, and simpler than modelling a real case.



The input u_{t-1} is simply the rain in the area and it enters through the matrix $\mathbf{B} = [1\ 0\ 0\ 0]^T$, thus directly into the first state X_1, i.e. the first compartment.

The system noise is state-dependent such that the *variance* of the process scales with the states, hence

$$\mathbf{G}(\mathbf{X_{t-1}}) = \begin{bmatrix} \sqrt{|X_{1,t-1}|} & 0 & 0 & 0 \\ 0 & \sqrt{|X_{2,t-1}|} & 0 & 0 \\ 0 & 0 & \sqrt{|X_{3,t-1}|} & 0 \\ 0 & 0 & 0 & \sqrt{|X_{4,t-1}|} \end{bmatrix}$$

The reason to take the square root is exactly to have the variance (and *not* the standard deviation) that scales with the level. The absolute value is just to avoid chrashing when states are close to zero, they can become negative.

The variance of the white noise process is modelled with the same level for each state

$$\mathrm{V}(\mathbf{e}_{1,t}) = \mathbf{\Sigma}_1 = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_1^2 & 0 & 0 \\ 0 & 0 & \sigma_1^2 & 0 \\ 0 & 0 & 0 & \sigma_1^2 \end{bmatrix}$$

Finally, $\mathbf{C} = [0\ 0\ 0\ 1]^T$ makes the last state the model output, which thus is the measured water level in the bassin – it has observation noise with variance $\mathrm{V}(e_{2,t}) = \sigma_2^2$ added to it.

The three parameters of the model, which can be varied to change the dynamics and variation through the system, are:

- $a \in [0.01, 0.1]$ the transition rate.
- $\sigma_1^2 \in [0.001, 1]$ the variance of the system noise.
- $\sigma_2^2 \in [0.1, 5]$ the variance of the observation noise.

The intervals should be set as the lower and upper bound when carrying out estimation of the three parameters.

The sampling interval is always one minute. Lastly, the initial state-vector is always set to zero, i.e. $\mathbf{X}_0 = [0\ 0\ 0\ 0]$.

1 Plot the data

Plot the rain input and bassin water level of each the four rain events. Consider scaling the input by 10 to have a nice single window plot. Using the same range on the time axis for all the plots provides a better overview, and same range for y-axis on all plots.

Answer the following:

1.1. Load the data and include the plots in the report.

1.2. Comment on the data. What relations are obvious? dynamics, variance, etc.

2 Simulate using the model

The first modelling task is to simulate the bassin water level for Rain Event 1 using the model. Write in the model to iteratively step to the next time step (i.e. to the next minute). To implement the scaling of the system noise with the states, set the square rooted state vector in the diagonal of a 4x4 matrix and multiply it to the standard normal random number vector (i.e. draw random values in an independent vector of length 4). Remember, the state-noise is independent between the states, i.e. only values in the diagonal.

2.1. Write the code to simulate the model and include it in Appendix 1 of the report, make it concise, but still with useful comments[2]. You must save both the simulated states and the output in each iteration.

2.2. Load the first rain event (i.e `"data/rain1.csv"`) and simulate the bassin water level with the model. Use some values of the three parameters that you find appropriate, include them in the report and plot the input, with the simulated states and the output.

2.3. Comment on the plotted simulation explaining how the model works.

2.4. Vary the three parameters one-by-one: for each, make one or two plots including multiple simulations (and the input and output), and comment what how the parameter influence the system dynamics and variation.

3 Simulate to find best starting values for estimation

When estimating parameters with maximum likelihood it is always a nice challenge to find good starting values for the parameters for the optimization. Set up code to make multiple simulations for each rain event and plot them together the input and measured output. One plot for each rain event.

3.1. Simulate with the same parameters for all rain events. Change them until you find the parameter values (same set for all events) where the simulations seems to "capture" the dynamics and variation on the bassin water level quite ok for all four events. Provide the values you have selected.

3.2. Include the plots of the simulations and measurements, and comment on your considerations leading you to decide the parameter values.

4 Maximum likelihood estimation with the Kalman filter

Now time has finally come to implement the model in the Kalman filter. It's straight forward, just write in the state-space model with the same matrices, NOTE however that the state-dependent system noise is now multiplied to the system noise variance Σ^{XX}, thus directly scaling the variance and therefore must be *without taking the square root* (but still the absolute value).

[2] You can seek inspiration from the simulation code given for the Week 11 exercise here.

4.1. Implement the model in the Kalman filter for calculating the likelihood of a rain event[3]. Again, make it nice and concise and include the code in Appendix 2 in the report.

4.2. Calculate the likelihood for the Rain Event 1. Calculate it for the appropriate set of parameter values for the three parameters you found, as well as for a lower and an upper bounds given in the introduction, make sure that the likelihood evaluates to a real number. Present the calculated likelihoods and comment. Note that the initial value of the state-vector is always zero, and the state-covariance should be set a little bit high only having values in the diagonal, e.g. 1000 in the diagonal.

4.3. Estimate the parameters for each of the rain events. Present them and comment on them, are they reasonable? How is the variation between the rain events?

4.4. Finally, do a model validation by simulating with the estimated parameters for each event. Are the simulations capturing the dynamics and variation of the system?

5 Further work

There are plenty of interesting things to do! However, we don't expect you to do more, however, if you feel like it, why not play around and do a bit more! It won't be evaluated (in the peer-review, and thus not count in the score), but if you do it, we are interested to see your results and ideas! ;)

For example

- Do a model validation by analyzing the one-step residuals from the Kalman filter. Are they white noise?
- Try approximating confidence intervals of the parameters, are they reasonable?
- Join the likelihoods from all events and estimate the parameters, do you get better estimates? Better prediction?
- Do some cross-validation, fit on one event and predict the other. Are we over-fitting etc.?
- Do a multi-step prediction with the Kalman filter and see if they are good, and what about the prediction intervals (i.e. use Σ^{YY}, are they capturing the measured output?
- More good ideas? Play around, you can only risk to learn something...while having fun!

[3] You can get inspiration from code given for the Week 11 exercise here.