

Computer lab C

Instructions

- The lab is assumed to be done in groups.
- Create a report to the lab solutions in PDF.
- Be concise and do not include unnecessary printouts and figures produced by the software and not required in the assignments.
- **Include all your codes as an appendix into your report.**
- A typical lab report should 2-4 pages of text plus some number of figures plus appendix with codes.
- The group lab report should be submitted via LISAM before the deadline specified in LISAM.
- **Use 12345 as a random seed everywhere where the result of the simulation differs with the run unless stated otherwise.**

Implementation of Kalman filter

Assignment 1

In table 1 a script for generation of data from simulation of the following state space model and implementation of the Kalman filter on the data is given.

$$\begin{aligned}\mathbf{z}_t &= A_{t-1}\mathbf{z}_{t-1} + e_t, \\ \mathbf{x}_t &= C_t\mathbf{z}_t + \nu_t, \\ \nu_t &\sim N(0, R_t), \\ e_t &\sim N(0, Q_t).\end{aligned}$$

- Write down the expression for the state space model that is being simulated.
- Run this script and compare the filtering results with a moving average smoother of order 5.
- Also, compare the filtering outcome when R in the filter is 10 times smaller than its actual value while Q in the filter is 10 times larger than its actual value. How does the filtering outcome vary?
- Now compare the filtering outcome when R in the filter is 10 times larger than its actual value while Q in the filter is 10 times smaller than its actual value. How does the filtering outcome vary?
- Implement your own Kalman filter and replace `ksmooth0` function with your script.
- How do you interpret the Kalman gain?

In Table 2 the Kalman filtering algorithm is given for reference.

Table 1: An R script for the Kalman filtering, smoothing, and prediction

```

# generate data
set.seed(1); num = 50
w = rnorm(num+1,0,1); v = rnorm(num,0,1)
mu = cumsum(w) # state: mu[0], mu[1],..., mu[50]
y = mu[-1] + v # obs: y[1],..., y[50]
# filter and smooth (Ksmooth0 does both)
ks = Ksmooth0(num, y, A=1, mu0=0, Sigma0=1, Phi=1, cQ=1, cR=1)
# start figure
par(mfrow=c(3,1)); Time = 1:num
plot(Time, mu[-1], main='Predict', ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xp)
lines(ks$xp+2*sqrt(ks$Pp), lty=2, col=4)
lines(ks$xp-2*sqrt(ks$Pp), lty=2, col=4)
plot(Time, mu[-1], main='Filter', ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xf)
lines(ks$xf+2*sqrt(ks$Pf), lty=2, col=4)
lines(ks$xf-2*sqrt(ks$Pf), lty=2, col=4)
plot(Time, mu[-1], main='Smooth', ylim=c(-5,10))
lines(Time,y,col="green")
lines(ks$xs)
lines(ks$xs+2*sqrt(ks$Ps), lty=2, col=4)
lines(ks$xs-2*sqrt(ks$Ps), lty=2, col=4)
mu[1]; ks$x0n; sqrt(ks$P0n) # initial value info

```

Table 2: Kalman filtering recursion

```

1: Inputs:  $A_t, C_t, Q_t, R_t, m_0, P_0$  and  $\mathbf{x}_{1:T}$ .
   initialization
2:  $m_{1|0} \leftarrow m_0, P_{1|0} \leftarrow P_0$ 
3: for  $t = 1$  to  $T$  do
   observation update step
4:  $K_t \leftarrow P_{t|t-1} C_t^T (C_t P_{t|t-1} C_t^T + R_t)^{-1}$ 
5:  $m_{t|t} \leftarrow m_{t|t-1} + K_t (\mathbf{x}_t - C_t m_{t|t-1})$ 
6:  $P_{t|t} \leftarrow (I - K_t C_t) P_{t|t-1}$ 
   prediction step
7:  $m_{t+1|t} \leftarrow A_t m_{t|t}$ 
8:  $P_{t+1|t} \leftarrow A_t P_{t|t} A_t^T + Q_{t+1}$ 
9: end for
10: Outputs:  $m_{t|t}, P_{t|t}$  for  $t = 1 : T$ 

```
