MATH5805: Advanced Time Series Analysis
Session 1, 2014
Assignment 2
SOLUTIONS TO SELECTED EXERCISES

Question 1.(D&K Ex. 3.11.1)
Consider the autoregressive moving average, with constant, plus noise model

$$y_t = y_t^* + \epsilon_t, \quad y_t^* = \mu + \phi_1 y_{t-1}^* + \phi_2 y_{t-2}^* + \zeta_t + \theta_1 \zeta_{t-1}$$

for $t = 1, \ldots, n$ where μ is an unknown constant and the disturbances $\epsilon_t \overset{\text{i.i.d}}{\sim} \mathrm{N}(0, \sigma_\epsilon^2)$ and $\zeta_t \overset{\text{i.i.d}}{\sim} \mathrm{N}(0, \sigma_\zeta^2)$ are mutually independent sequences. The autoregressive coefficients ϕ_1, ϕ_2 are restricted such that y_t^* is a stationary process and we assume that $0 < \theta_1 < 1$. Represent this model in the state space form (3.1) and (3.2) of slide 113 of the notes (equation (3.1) of DK).

Question 1 SOLUTION:
The various matrices required to put this into the form of equations (3.1) and (3.2) of slide 113 are:

$$y_t = Z_t \alpha_t + \epsilon_t, \quad \epsilon_t \overset{\text{indep}}{\sim} \mathrm{N}(0, \sigma_\epsilon^2)$$

where $Z_t = [1, 0, 0]$ and the state vector is

$$\alpha_t = \begin{bmatrix} y_t^* \\ \phi_2 y_{t-1}^* + \theta_1 \xi_t \\ \mu \end{bmatrix}$$

and

$$\alpha_{t+1} = T\alpha_t + R\eta_t, \quad \eta_t = \xi_{t+1} \overset{\text{indep}}{\sim} \mathrm{N}(0, \sigma_\eta^2)$$

with

$$T = \begin{bmatrix} \phi_1 & 1 & 1 \\ \phi_2 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 \\ \theta_1 \\ 0 \end{bmatrix}$$

Question 2. (D&K Ex. 3.11.4)

Consider the state space model defined in equations (3.1) and (3.2) of slide 113 of the notes (or equation (3.1) of D&K) extended with regression effects

$$y_t = X_t\beta + Z_t\alpha_t + \epsilon_t, \quad \alpha_{t+1} = W_t\beta + T_t\alpha_t + R_t\eta_t,$$

for $t = 1, \ldots, n$ where X_t and W_t are fixed matrices that (partly) consist of exogenous variables and β is a vector of regression coefficients and matrices and vectors have conformable dimensions.

a) Show that this state space model can be expressed as

$$y_t = X_t^*\beta + Z_t\alpha_t^* + \epsilon_t, \quad \alpha_{t+1}^* = T_t\alpha_t^* + R_t\eta_t,$$

for $t = 1, \ldots, n$.

b) Give and expression for X_t^* in terms of X_t, W_t and the other system matrices.

Question 2 SOLUTION:

Define $\alpha_t^* = \alpha_t + W_t^*\beta$. Substitute this into the alternative observation and state equations of part a) to get:

$$\begin{aligned} y_t &= X_t^*\beta + Z_t\alpha_t^* + \epsilon_t \\ &= X_t^*\beta + Z_tW_t^*\beta + Z_t\alpha_t + \epsilon_t \\ &= (X_t^* + Z_tW_t^*)\beta + Z_t\alpha_t + \epsilon_t \\ &= X_t\beta + Z_t\alpha_t + \epsilon_t \end{aligned}$$

provided $X_t^* = X_t - Z_tW_t^*$ by choice of W_t^*. Also

$$\alpha_{t+1} + W_{t+1}^*\beta = T_t\alpha_t + T_tW_t^*\beta + R_t\eta_t$$

hence

$$\begin{aligned} \alpha_{t+1} &= T_t\alpha_t + (T_tW_t^* - W_{t+1}^*)\beta + R_t\eta_t \\ &= T_t\alpha_t + R_t\eta_t \end{aligned}$$

as required, provided $T_tW_t^* - W_{t+1}^* = W_t$. This is equivalent to requiring

$$W_{t+1}^* = T_tW_t^* - W_t$$

which can be solved recursively.

In conclusion we set $X_t^* = X_t - Z_tW_t^*$ where W_t^* are obtained for each t by solving the last recursively to get $W_1^*, W_2^*, \ldots, W_t^*, \ldots$.

Question 3. (D&K Ex. 4.14.1)
Refer to Lemma 4.1 on slide 130. Using the notation there let

$$\Sigma_* = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma'_{xy} & \Sigma_{yy} \end{bmatrix}.$$

a) Verify

$$\Sigma_* = \begin{bmatrix} I & \Sigma_{xy}\Sigma_{yy}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma'_{xy} & 0 \\ 0 & \Sigma_{yy} \end{bmatrix} \begin{bmatrix} I & 0 \\ \Sigma_{yy}^{-1}\Sigma'_{xy} & I \end{bmatrix}$$

.

b) Hence verify

$$\Sigma_*^{-1} = \begin{bmatrix} I & 0 \\ -\Sigma_{yy}^{-1}\Sigma'_{xy} & I \end{bmatrix} \begin{bmatrix} (\Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma'_{xy})^{-1} & 0 \\ 0 & \Sigma_{yy}^{-1} \end{bmatrix} \begin{bmatrix} I & -\Sigma_{xy}\Sigma_{yy}^{-1} \\ 0 & I \end{bmatrix}$$

.

c) Using the joint density

$$p(x, y) = \text{constant} \times \exp\left[-\frac{1}{2}\begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}' \Sigma_*^{-1} \begin{pmatrix} x - \mu_x \\ y - \mu_y \end{pmatrix}\right]$$

obtain $p(x|y)$ and hence prove Lemma 4.1.

Question 4.

a) (D&K Ex. 4.14.3) Suppose that the fixed vector λ is regarded as an estimate of a random vector x whose mean vector is μ. Show that the minimum mean square error matrix $E[(\lambda - x)(\lambda - x)']$ is obtained when $\lambda = \mu$.

b) (D&K Ex. 4.14.4) In the joint distribution, not necessarily normal, of x and y suppose that $\bar{x} = \beta + \gamma y$ is an estimate of x given y with mean squared error $\mathrm{MSE}(\bar{x}) = E[(\bar{x} - x)(\bar{x} - x)']$. Use the details of the proof of Lemma 4.2 and part a) of this question show that the minimum mean square error matrix is obtained when $\bar{x} = \hat{x}$ where $\hat{x}$ and $\mathrm{MSE}(\hat{x}) = \mathrm{Var}(x - \hat{x})$ are as given in Lemma 4.2.

Question 5.

This exercise requires you to use the **KFAS** R-package. Locate the file `SeatBelts.R` and using the **KFAS** manual (Package 'KFAS' December 7, 2013) for reference as needed, perform the following tasks.

a) Explain what each step of the code in `SeatBelts.R` is doing. You can annotate the file with brief comments and submit as part of your answer to this question.

b) Write out using the state space formulation of Week 4 notes the model being fit. Briefly describe the key model elements and the features of the Seat Belt series being modelled by them.

c) Report the estimated values of all unknown parameters in the model and their standard errors. You may need to use the delta-method to do this.

d) Provide an estimate of the impact of the intervention (seat belt law enactment).

e) Create additional R-code to:

- Perform residual analysis as described in Week 3 Notes.
- Forecast 24 months into the future from the end of the available data with 68% prediction intervals.
- Simulate 4 samples of the signal conditional on the observed series (provide these in a four panel plot.

Include the code in an expanded '`SeatBelts.R`' and included a printout of this with any output and graphs required with your answers.

Question 5 R-CODE for SOLUTION:

```
# Setting up SeatBelt models using KFAS.

# Seatbelts data
data(Seatbelts)
par(mfrow=c(3,1))
ts.plot(log(Seatbelts[,"drivers"]),ylab="",main="log UK driver killed or seriously injured")
ts.plot(log(Seatbelts[,"PetrolPrice"]),ylab="",main="log Petrol Price")
ts.plot(Seatbelts[,"law"],ylab="",main="Seat Belt Law Change")

# Create State Space Model for response log(drivers) as a regression on log(PetrolPrice)
# and intervention step for change in seat belt law ('law') plus a trignometric
# seasonal term (using pairs of cos and sin - see DK equation (3.8)) plus a random walk
# local level and observations noise.

model<-SSModel(log(drivers)~SSMtrend(1,Q=list(NA))+
               SSMseasonal(period=12,sea.type='trigonometric',Q=NA)+
               log(PetrolPrice)+law,data=Seatbelts,H=NA)

# As trigonometric seasonal contains several disturbances which are all
```

```
# identically distributed, default behaviour of fitSSM is not enough,
# as we have constrained Q. We can either provide our own
# model updating function with fitSSM, or just use optim directly:

# option 1:
ownupdatefn<-function(pars,model,...){
  model$H[]<-exp(pars[1])
  diag(model$Q[,,1])<-exp(c(pars[2],rep(pars[3],11)))
  model #for option 2, replace this with -logLik(model) and call optim directly
}

fit<-fitSSM(inits=log(c(var(log(Seatbelts[,'drivers'])),0.001,0.0001)),
            model=model,updatefn=ownupdatefn,method='BFGS', hessian=TRUE)

# Estimated parameters are obtained by
fit$optim.out$par
# covariance natrix of these can be obtained by (note optim minimises!! so no need
# to change the sign of the Hessian)
solve(fit$optim.out$hessian)
# giving
#[,1]         [,2]         [,3]
#[1,]  0.02308874 -0.041751042 -0.061092132
#[2,] -0.04175104  0.365913435 -0.009827622
#[3,] -0.06109213 -0.009827622  1.459500791
# standard errors for pars are the square roots of diagonal elements:
results<-cbind(fit$optim.out$par,diag(solve(fit$optim.out$hessian))^0.5)
colnames(results)<-c("par. est.", "SE")
print(results)

# giving:
#par. est.        SE
#[1,]  -5.576387 0.1519498
#[2,]  -8.225728 0.6049078
#[3,] -13.665147 1.2080980
# convert to variances by reversing log transform:
exp(fit$optim.out$par)
#[1] 3.786220e-03 2.676774e-04 1.162257e-06

# Already Converted to model variances in:
fitH<-fit$model$H
print(fitH) # observation noise variance estimated as 0.00378622

#[1,] 0.00378622

fitQ<-diag(fit$model$Q[1:2,1:2,1])
print(fitQ)
```

```
#[1] 2.676774e-04 1.162257e-06

# Standard errors for the variances needs the delta method applied to basic parameters.

# Regression term estimates are in the smoothed state vector

out<-KFS(fit$model,smoothing=c('state','signal'))
# this performs Kalman Filtering and Smoothing and produces smoothed estimates of
# states alpha_t and signal theta_t.
out
# This last line gives the estimates of states at beginning and end
# of the time period. The first two elements of the estimated state vector
# are the regression coefficients for log(PetrolPrice) and sealt belt law change
# 'law'. These are time invariant.
# Smoothed values of the first and last states and their standard errors:
#  alphahat_1  alphahat_192  se_1       se_192
#log(PetrolPrice)  -0.291403   -0.291403      0.098335   0.098318
#law               -0.237737   -0.237737      0.046317   0.046317

# Thus the log(petrol price) is estimated to have a negative coefficient of -0.291403 with
# standard error of 0.098335 (thus is highly significant) and law has an estimated negative
# coefficient of -0.237737 with standard error 0.046317 again highly significant.
# Since we are modelling log(drivers) the impact of a one unit increase in petrol prices has
# approximately an exp(-.291403) = 0.75 decrease on drivers killed (3/4 of a driver) and
# the level of deaths is about  =79% of the level before the law. Thus the law is associated
# with a reduction of about 21% in deaths.

# plot of local level components and seasonals can be obtained from out$alphahat
# components 3 to 14 as for example in this plot:
par(mfrow=c(2,2))
ts.plot(out$alphahat[,3],ylab="level")
ts.plot(out$alphahat[,4],ylab="trig_1")
ts.plot(out$alphahat[,5],ylab="trig_1*")
ts.plot(out$alphahat[,6],ylab="trig_2")
# the seasonal component of the signal can be stripped out as
seasonal<-out$alphahat[,-c(1,3)]%*%model$Z[1,4:14,1]
# needs to be fixed
# ditto getting other components, regression and local level..
# TO COME.

par(mfrow=c(1,1))
ts.plot(cbind(out$model$y,out$muhat),lty=1:2,col=1:2,
main='Observations and smoothed signal with and without seasonal component')
legend('bottomleft',
       legend=c('Observations', 'Smoothed signal'),
                    col=c(1,2,4), lty=c(1,2,1))
```

```
# Residual Analysis:

# can use standardsised residuals function 'rstandard':

residuals<-rstandard(out,type="pearson")
par(mfrow=c(2,2))
ts.plot(residuals,type="h")
hist(residuals,prob=TRUE, main="Distibution of Residuals")
lines(density(residuals),col="red")
qqnorm(residuals)
qqline(residuals, col="red")
acf(residuals)

# Forecasting
# Assume law change stays constant at value 1 (no further changes in laws over forecast hori
# Assume petrol prices stay constant at last value (roughly average for past three year or s
# local level and seasonals will take care of themselves.

logdrivers_extend<-c(log(Seatbelts[,"drivers"]),rep(NA,24))
ndata<-dim(Seatbelts)[1]
logPetrolPrice_extend<-c(log(Seatbelts[,"PetrolPrice"]),rep(log(Seatbelts[,"PetrolPrice"])[
law_extend<-c(Seatbelts[,"law"],rep(1,24))

par(mfrow=c(3,1))
ts.plot(logdrivers_extend,ylab="",main="log UK driver killed or seriously injured")
ts.plot(logPetrolPrice_extend,ylab="",main="log Petrol Price")
ts.plot(law_extend,ylab="",main="Seat Belt Law Change")


model_extend<-SSModel(logdrivers_extend~SSMtrend(1,Q=list(fitQ[1]))+
                 SSMseasonal(period=12,sea.type='trigonometric',Q=fitQ[2])+
                 logPetrolPrice_extend+law_extend,H=fit$model$H)

model_extend.FS<-KFS(model_extend,smoothing=c('state','signal'),simplify = FALSE)

# Plot observation forecasts 24 months ahead:
par(mfrow=c(1,1))
logSeatbelts.fore<-ts(c(log(Seatbelts[,"drivers"]),model_extend.FS$muhat[(ndata+1):(ndata+24
temp<-cbind(logSeatbelts.fore,
            logSeatbelts.fore+qnorm(0.84)*model_extend.FS$F[1,(ndata+1):(ndata+24)]^0.5,
            logSeatbelts.fore-qnorm(0.84)*model_extend.FS$F[1,(ndata+1):(ndata+24)]^0.5)
ts.plot(ts(temp[,1],start=1),col=c(1,1,1),lwd=c(1,2,1),lty=c(1,0,0),
main="UK Road Deaths and Injuries Forecast")
lines(ts(temp[(ndata+1):(ndata+24),2],start=ndata+1),col="red")
lines(ts(temp[(ndata+1):(ndata+24),3],start=ndata+1),col="blue")
```

```
# Simulation
# Recall from above: out<-KFS(fit$model,smoothing=c('state','signal'))

# use out$muhat for smoothed signal.
simulation<-simulateSSM(fit$model,
                         type = c("signal"),
                         filtered = FALSE, nsim = 4, antithetics = FALSE,
                         conditional = TRUE)
par(mfrow=c(2,2))
ylims<-range(c(out$muhat,simulation))
for (rep in 1:4){
ts.plot(ts(out$muhat, ,start=1969, frequency=12),
        ylim=ylims, lwd=2, col=2, ylab="")
points(ts(simulation[,1,rep],start=1969, frequency=12), pch="*")
title(paste("Signal Simulation ",as.character(rep)))

}

sims<-simulateSSM()
```