ECON 211C: Problem Set 3

Due: Friday, June 2, 2017

Eric M. Aldrich

David Sungho Park

Problem 1

(a)

The coefficient vector for the exact, finite-sample, one-step forecast using five past observations is

$$\boldsymbol{\beta}^{(5,1)} = E[\boldsymbol{X}_t\boldsymbol{X}_t']^{-1}E[\boldsymbol{X}_tY_{t+1}] = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 \\ \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \end{bmatrix}.$$

The autocovariances are computed as

$$\gamma_0 = 17.517,\ \gamma_1 = 15.957,\ \gamma_2 = 12.401,\ \gamma_3 = 8.396,\ \gamma_4 = 5.058,\ \gamma_5 = 3.016.$$

Therefore,

$$\boldsymbol{\beta}^{(5,1)} = \begin{bmatrix} 1.77 & -1.21 & 0.44 & -0.25 & 0.15 \end{bmatrix}'.$$

(b)

Table 1 reports the one-step forecasts for $Y_{101}, Y_{102}, \ldots, Y_{105}$.

Table 1: One-step forecasts using five past observations

$\hat{Y}_{101}$	$\hat{Y}_{102}$	$\hat{Y}_{103}$	$\hat{Y}_{104}$	$\hat{Y}_{105}$
-0.18	-0.31	-0.43	-0.44	-0.37

(c)

The MSE is calculated to be 17.33.

(d)

The coefficient vector for the exact, finite-sample, five-step forecast using five past observations is

$$\boldsymbol{\beta}^{(5,5)} = E[\boldsymbol{X}_t\boldsymbol{X}_t']^{-1}E[\boldsymbol{X}_tY_{t+5}] = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_2 & \gamma_1 & \gamma_0 & \gamma_1 \\ \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & \gamma_0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma_5 \\ \gamma_6 \\ \gamma_7 \\ \gamma_8 \\ \gamma_9 \end{bmatrix}.$$

The autocovariances can be further computed by the formula $\gamma_j = 1.3\gamma_{j-1} - 0.4\gamma_{j-2}$:

$$\gamma_6 = 1.897,\ \ \gamma_7 = 1.260,\ \ \gamma_8 = 0.879,\ \ \gamma_9 = 0.639.$$

Then,

$$\boldsymbol{\beta}^{(5,5)} = \begin{bmatrix} 0.661 & -0.788 & 0.323 & 0.011 & -0.015 \end{bmatrix}'.$$

(e)

An exact, finite-sample, five-step forecast for Y_{105} is -0.1773.

(f)

The MSE is calculated to be 16.913.

(g)

Table 2 reports the $ARMA$ parameters estimated by using the first 100 observations.

Table 2: Estimates of $ARMA$ parameters

$\hat{\phi}_1$	$\hat{\phi}_2$	$\hat{\theta}_1$	$\hat{\theta}_2$	$\hat{\theta}_3$	$\hat{\theta}_4$	$\hat{\theta}_5$
-0.099	0.451	2.016	1.738	1.296	0.565	-0.048

(h)

The autocovariances are estimated to be

$$\hat{\gamma}_0 = 11.706,\ \ \hat{\gamma}_1 = 10.272,\ \ \hat{\gamma}_2 = 6.924,\ \ \hat{\gamma}_3 = 3.000,\ \ \hat{\gamma}_4 = -0.622,\ \ \hat{\gamma}_5 = -3.089.$$

Accordingly the coefficient vector is computed as

$$\hat{\boldsymbol{\beta}}^{(5,1)} = \begin{bmatrix} 0.883 & 0.856 & -1.011 & -0.260 & 0.441 \end{bmatrix}'.$$

(i)

The MSE is calculated to be 16.327.

(j)

The code seems not properly run in `R` for being non stationary. `R` code is attached in the Appendix.

(k)

$$\boldsymbol{\beta}^{(5,5)} = \begin{bmatrix} 0.658 & -0.232 & -0.507 & 0.252 & 0.095 \end{bmatrix}'.$$

(l)

The MSE is calculated to be 17.550.

Problem 2

(a)

The $VAR(2)$ model can be specified as

$$\boldsymbol{Y}_t = \boldsymbol{c} + \boldsymbol{\Phi}_1 \boldsymbol{Y}_{t-1} + \boldsymbol{\Phi}_2 \boldsymbol{Y}_{t-1} + \boldsymbol{\varepsilon}_t$$

$$\iff \begin{bmatrix} Y_{1t} \\ Y_{2t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} \phi_{11}^1 & \phi_{12}^1 \\ \phi_{21}^1 & \phi_{22}^1 \end{bmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \phi_{11}^2 & \phi_{12}^2 \\ \phi_{21}^2 & \phi_{22}^2 \end{bmatrix} \begin{bmatrix} Y_{1,t-2} \\ Y_{2,t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{bmatrix}$$

$$\iff \begin{pmatrix} Y_{1t} = c_1 + \phi_{11}^1 Y_{1,t-1} + \phi_{12}^1 Y_{2,t-1} + \phi_{11}^2 Y_{1,t-2} + \phi_{12}^2 Y_{2,t-2} + \varepsilon_{1t} \\ Y_{2t} = c_2 + \phi_{21}^1 Y_{1,t-1} + \phi_{22}^1 Y_{2,t-1} + \phi_{21}^2 Y_{1,t-2} + \phi_{22}^2 Y_{2,t-2} + \varepsilon_{2t} \end{pmatrix},$$

where Y_{1t} is returns and Y_{2t} is order flow. With the given data for the EUR/USD exchange rate on 13 Nov 2013, the coefficients are estimated as

$$\boldsymbol{c} = \begin{bmatrix} 8.057e-06 \\ 4.902e+00 \end{bmatrix} \quad \boldsymbol{\Phi}_1 = \begin{bmatrix} -5.967e-03 & -1.195e-07 \\ 3.619e+04 & -1.079e-01 \end{bmatrix} \quad \boldsymbol{\Phi}_2 = \begin{bmatrix} 6.708e-02 & -4.342e-08 \\ 2.830e+04 & 5.419e-02 \end{bmatrix}.$$

(b)

$$\begin{bmatrix} Y_{1t} \\ Y_{2t} \\ Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \phi_{11}^1 & \phi_{12}^1 & \phi_{11}^2 & \phi_{12}^2 \\ \phi_{21}^1 & \phi_{22}^1 & \phi_{21}^2 & \phi_{22}^2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{1,t-2} \\ Y_{2,t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \\ 0 \\ 0 \end{bmatrix}$$

$$\implies \begin{bmatrix} Y_{1t} \\ Y_{2t} \\ Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} = \begin{bmatrix} 8.057e-06 \\ 4.902e+00 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -5.967e-03 & -1.195e-07 & 3.619e+04 & -1.079e-01 \\ 6.708e-02 & -4.342e-08 & 2.830e+04 & 5.419e-02 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \\ Y_{1,t-2} \\ Y_{2,t-2} \end{bmatrix}$$

(c)

The matrix is computed as

$$T = \begin{bmatrix} 1.303374e-06 & -1.000000e+00 \\ 1.000000e+00 & 1.303374e-06 \end{bmatrix}.$$

Appendix: R Code

```
# Some functions
## Function to generate coefficient vector
betaGen = function(gammaVec0,m,s){
  gammaMat = matrix(0,m,m)
  for(i in 1:m){
    for(j in 1:m){
      gammaMat[i,j] = gammaVec0[abs(i-j)+1]
    }
  }
  beta = solve(gammaMat) %*% gammaVec0[(1+s):(s+m)]
  return(beta)
}

## Function to simulate ARMA(p,q)
ARMAsim = function(phi,theta,mu,sigma,nObs,nSim){
  p = length(phi)
  q = length(theta)-1
  eps = matrix(0,nObs+q,nSim)
  for(j in 1:nSim){
    for(i in 1:(nObs+q)){
      eps[i,j] = rnorm(1,mu,sigma)
    }
  }
  ySim = matrix(0,nObs+q,nSim)
  for(j in 1:nSim){
    for(i in (1+q):(nObs+q)){
      ySim[i,j] = phi %*% ySim[(i-1):(i-p),j] + theta %*% eps[i:(i-q),j]
    }
  }
  ySim = ySim[-(1:q),]
  return(ySim)
}

## Function to generate exact, finite-sample, s-step forecasts
EFSforecast = function(Y,beta,tStart,tEnd,s){
  yHat = Y
  m = length(beta)
  nSim = ncol(Y)
  if(is.null(nSim)==TRUE){
    for(t in tStart:tEnd){
      yHat[t] = yHat[(t-s):(t-s-m+1)] %*% beta
    }
  }
  else{
    for(j in 1:nSim){
      for(t in tStart:tEnd){
        yHat[t,j] = yHat[(t-s):(t-s-m+1),j] %*% beta
      }
    }
  }
  return(yHat)
}
```

```
## Function to calculate MSE
MSEcal = function(Y,yHat,t){
  nObs = nrow(Y)
  nSim = ncol(Y)
  sqErr = rep(0,nSim)
  for(j in 1:nSim){
    sqErr[j] = (yHat[t,j]-Y[t,j])^2
  }
  MSE = mean(sqErr)
  return(MSE)
}

##################################### Question 1 (a) #####################################
# Theoretical autocovariances (obtained in PS1: Q3)
gammaVec0_a = c(17.5170, 15.9570, 12.4010, 8.3985, 5.0576, 3.0155)

# Generate beta
beta51_a = betaGen(gammaVec0_a,5,1)

##################################### Question 1 (b) #####################################
# Simulation: 105 obs x 1 time
## Parameters
phi = c(1.3, -0.4)
theta = c(1, 0.7, 0, 0.1, -0.5, -0.2)
mu = 0
sigma = 1
nObs = 105
nSim = 1
## Simulate using "ARMAsim" function
Y_b = ARMAsim(phi,theta,mu,sigma,nObs,nSim)

# One-step forecasts of Y_t for t = 101:105
yHat_b = EFSforecast(Y_b,beta51_a,101,105,1)

# Export to LaTeX
library(xtable)
xtable(t(matrix(yHat_b[101:105])),align = "cccccc")

##################################### Question 1 (c) #####################################
# Simulation: 105 obs x 1000 times
## Parameters
phi = c(1.3, -0.4)
theta = c(1, 0.7, 0, 0.1, -0.5, -0.2)
mu = 0
sigma = 1
nObs = 105
nSim = 1000
## Simulate using "ARMAsim" function
Y_c = ARMAsim(phi,theta,mu,sigma,nObs,nSim)

# One-step forecasts of Y_t for t = 101:105 (x 1000 times)
yHat_c = EFSforecast(Y_c,beta51_a,101,105,1)
```

```
# Calculate MSE using "MSEcal" function
MSE_c = MSEcal(Y_c,yHat_c,105)

##################################### Question 1 (d) #####################################
# Extending gammaVec
gammaVec0_d = gammaVec0_a
for(g in 6:9){
  gammaVec0_d = c(gammaVec0_d,1.3*gammaVec0_d[g] -0.4*gammaVec0_d[g-1])
}

# Generate beta55
beta55_d = betaGen(gammaVec0_d,5,5)

##################################### Question 1 (e) #####################################
# A single five-step forecast of Y_105
yHat_e = EFSforecast(Y_b,beta55_d,105,105,5)
yHat_e[105]

##################################### Question 1 (f) #####################################
# Five-step forecast of Y_105 (x 1000 times)
yHat_f = EFSforecast(Y_c,beta55_d,105,105,5)
yHat_f[105,]

# Calculate MSE using "MSEcal"
MSE_f = MSEcal(Y_c,yHat_f,105)

##################################### Question 1 (g) #####################################
# ARMA estimation
arima(Y_b,order = c(2,0,5),include.mean=FALSE)$coef

# Export to LaTeX
ARMAest_g = arima(Y_c[1:100,1],order = c(2,0,5),include.mean=FALSE)$coef
xtable(t(matrix(ARMAest_g)),digit=3)

##################################### Question 1 (h) #####################################
# Estimated ARMA parameters
phi_est_h = arima(Y_c[1:100,1],order = c(2,0,5),include.mean=FALSE)$coef[1:2]
theta_est_h = c(1,arima(Y_c[1:100,1],order = c(2,0,5),include.mean=FALSE)$coef[3:7])

# Estimated autocovariances
gammaVec0_est_h = ARMAacf(phi_est_h,theta_est_h,lag.max=5)*var(Y_c[1:100,1])

# Generate beta
beta51_h = betaGen(gammaVec0_est_h,5,1)
xtable(t(matrix(beta51_h)),digit=3)

##################################### Question 1 (i) #####################################
# Simulation: 105 obs x 1000 times
## Estimated Parameters
phi_est_h = arima(Y_c[1:100,1],order = c(2,0,5),include.mean=FALSE)$coef[1:2]
theta_est_h = c(1,arima(Y_c[1:100,1],order = c(2,0,5),include.mean=FALSE)$coef[3:7])
mu = 0
sigma = 1
nObs = 105
```

```
nSim = 1000
## Simulate using "ARMAsim" function
Y_i = ARMAsim(phi_est_h,theta_est_h,mu,sigma,nObs,nSim)

# One-step forecasts of Y_t for t = 101:105 (x 1000 times)
yHat_i = EFSforecast(Y_i,beta51_h,101,105,1)

# Calculate MSE
MSE_i = MSEcal(Y_i,yHat_i,105)

##################################### Question 1 (j) #####################################
# Attempting several different Y_j's
j = 70
phi_est_j = arima(Y_c[1:100,j],order = c(2,0,5),include.mean=FALSE)$coef[1:2]
theta_est_j = c(1,arima(Y_c[1:100,j],order = c(2,0,5),include.mean=FALSE)$coef[3:7])
mu = 0
sigma = 1
nObs = 105
nSim = 1000
Y_j = ARMAsim(phi_est_j,theta_est_j,mu,sigma,nObs,nSim)

# One-step forecasts of Y_t for t = 101:105 (x 1000 times) (updating estimates)
yHat_j = Y_j
for(j in 1:nSim){
  for(s in 101:105){
    phi = arima(yHat_j[1:(s-1),j],order = c(2,0,5),include.mean=FALSE)$coef[1:2]
    theta = c(1,arima(yHat_j[1:(s-1),j],order = c(2,0,5),include.mean=FALSE)$coef[3:7])
    gammaVec0 = ARMAacf(phi,theta,lag.max=5)*var(yHat_j[(s-1),j])
    beta51 = betaGen(gammaVec0,5,1)
    yHat_j[s,j] = EFSforecast(yHat_j[(s-100):(s-1),j],beta51,s,s,1)[s]
  }
}

# Calculate MSE
MSE_j = MSEcal(Y_j,yHat_j,105)

##################################### Question 1 (k) #####################################
# Extending gammaVec
gammaVec0_k = gammaVec0_est_h
for(g in 6:9){
  gammaVec0_k = c(gammaVec0_k,1.3*gammaVec0_k[g] -0.4*gammaVec0_k[g-1])
}

# Generate beta55
beta55_k = betaGen(gammaVec0_k,5,5)
xtable(t(matrix(beta55_k)),digit=3)

##################################### Question 1 (l) #####################################
# Five-step forecast of Y_105 (x 1000 times)
yHat_l = EFSforecast(Y_c,beta55_k,105,105,5)
yHat_l[105,]

# Calculate MSE using "MSEcal"
MSE_l = MSEcal(Y_c,yHat_l,105)
```

```
#################################### Question 2 (a) ######################################
Y1 = read.csv("ps3Dat.csv")$Returns
Y2 = read.csv("ps3Dat.csv")$OrderFlow

# Estimation
t = length(Y1)
eq1Est = lm(Y1[3:t] ~ Y1[2:(t-1)] + Y2[2:(t-1)] + Y1[1:(t-2)] + Y2[1:(t-2)])
eq2Est = lm(Y2[3:t] ~ Y1[2:(t-1)] + Y2[2:(t-1)] + Y1[1:(t-2)] + Y2[1:(t-2)])
eq1Est_c = signif(summary(lm(Y1[3:t] ~ Y1[2:(t-1)] + Y2[2:(t-1)] + Y1[1:(t-2)]
    + Y2[1:(t-2)]))$coefficients[,1],4)
eq2Est_c = signif(summary(lm(Y2[3:t] ~ Y1[2:(t-1)] + Y2[2:(t-1)] + Y1[1:(t-2)]
    + Y2[1:(t-2)]))$coefficients[,1],4)

CCC = matrix(c(eq1Est_c[1],eq2Est_c[1]),2,1)
Phi1 = matrix(c(eq1Est_c[2],eq1Est_c[3],eq2Est_c[2],eq2Est_c[3]),2,2,byrow=TRUE)
Phi2 = matrix(c(eq1Est_c[4],eq1Est_c[5],eq2Est_c[4],eq2Est_c[5]),2,2,byrow=TRUE)

# Covariance matrix
w11 = sum(eq1Est$resid^2)/383
w22 = sum(eq2Est$resid^2)/383
w12 = sum(eq1Est$resid*eq2Est$resid)/383
w21 = sum(eq2Est$resid*eq1Est$resid)/383
omega = matrix(c(w11,w21,w12,w22),2,2)

#################################### Question 2 (c) ######################################
Tmat = eigen(omega)$vectors
xtable(Tmat,digits=3)
```