

Chapter 6 - Solutions to exercises

Exercises: 1,4,5

Exercise 1

a)

We have that X is a continuous random variable with cumulative distribution function c.d.f. F , and that $U = F(X)$ so $X = F^{-1}(U)$. Then

$$\begin{aligned}\Pr(U < u) &= \Pr(F(X) < u) \\ &= \Pr(X < F^{-1}(u)) \\ &= u,\end{aligned}$$

which is the c.d.f. for a uniform variable on $[0; 1]$.

b)

$$\Pr(X < x) = \Pr(F^{-1}(U) < x) = \Pr(U < F(x)) = F(x).$$

c)

i. For an exponential variable X we have $F(x) = 1 - e^{-\lambda x}$. The inverse c.d.f. is therefore

$$\begin{aligned}y &= 1 - e^{-\lambda x} \\ e^{-\lambda x} &= 1 - y \\ x &= -\frac{1}{\lambda} \log(1 - y) \\ \Leftrightarrow F^{-1}(y) &= -\frac{1}{\lambda} \log(1 - y), \quad y \in [0; 1].\end{aligned}$$

ii. See Figure 1 for numerical verification.

d)

$$\begin{aligned}Z &= \Phi^{-1}(F(x)) \\ Z &= \Phi^{-1}(U) \\ \Phi(Z) &= U\end{aligned}$$

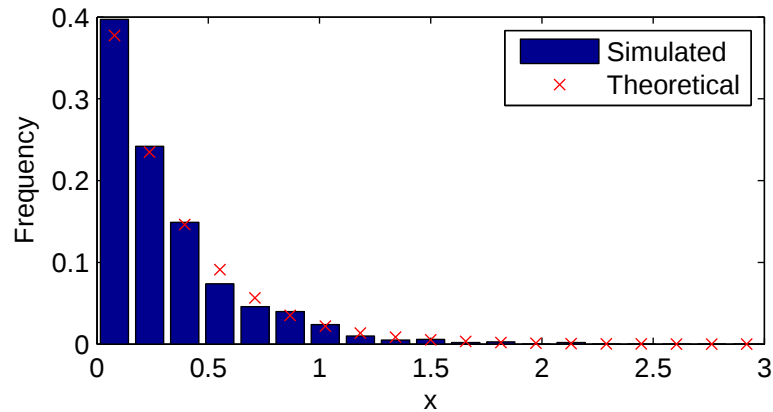


Figure 1: Histogram of sample for the exponential distribution ($\lambda = 3$).

Exercise 4

```
# Chapter 6, R-code for exercise 4, mwp 2/2-2011
source("A2.txt")
statdist <- function(gamma){
  m = dim(gamma)[1]
  matrix(1,1,m) %*% solve(diag(1,m) - gamma + matrix(1,m,m))
}

m = 2
gamma = rbind(c(0.6,0.4),c(0.4,0.6))
delta = statdist(gamma)
lambda1 = c(2,5) # for t=1:100
lambda2 = c(2,7) # for t=101:120

n1 = 100
n2 = 20

x1 <- pois.HMM.generate_sample(n1,m,lambda1,gamma,delta)
x2 <- pois.HMM.generate_sample(n2,m,lambda2,gamma,delta)

x = c(x1,x2)

res <- pois.HMM.EM(x[1:80],m,lambda1,gamma,delta,maxiter=1000,tol=1e-6)

pois.HMM.forecast.pseudo <- function(x,m,lambda,gamma,
  delta=NULL,xrange=NULL,H=1,...)
{
  if(is.null(delta))
    delta<-solve(t(diag(m)-gamma+1),rep(1,m))
  if(is.null(xrange))
    xrange<-qpois(0.001,min(lambda)):
      qpois(0.999,max(lambda))
  n <- length(x)
```

```

allprobs <- outer(x,lambda,dpois)
allprobs <- ifelse(!is.na(allprobs),allprobs,1)
foo      <- delta*allprobs[1,]
sumfoo   <- sum(foo)
lscale   <- log(sumfoo)
foo      <- foo/sumfoo

N        <- length(xrange)
allprobs2<- t(outer(xrange,lambda,dpois))
fdists   <- matrix(NA,nrow=n,ncol=N)
foo      <- foo%*%gamma
xi       <- foo
fdists[1,] <- xi %*% allprobs2
ul <- uh <- rep(NA,n)

for (i in 2:n)
{
  # New stuff
  cumdist <- cumsum(fdists[i-1,])
  ul[i]   <- cumdist[x[i]+1]
  uh[i]   <- cumdist[x[i]+2]
  # Old stuff
  foo     <- foo%*%gamma*allprobs[i,]
  sumfoo  <- sum(foo)
  lscale  <- lscale+log(sumfoo)
  foo     <- foo/sumfoo
  # New stuff
  foo     <- foo%*%gamma
  xi      <- foo
  fdists[i,] <- xi %*% allprobs2
}
um       <- 0.5*(ul+uh)
npsr     <- qnorm(rbind(ul,um,uh))
#list(xrange=xrange,fdists=fdists)
npsr
}

npsr <- pois.HMM.forecast.pseudo(x[81:120],m,
                                res$lambda,res$gamma,delta,xrange=NULL,H=1)

postscript('npsr.ps')
plot(npsr[2,1:39], ylim = c(-4, 4), type = "n",ylab = "pseudo-residual",xlab = "index")
abline(h = outer(c(1.96, 2.58), c(-1, 1)))
segments(x0=1:length(x), y0=npsr[1,], y1 = npsr[3,])
dev.off()

```

See Figure 2 for the monitoring of the 40 subsequent observations and their normal forecast pseudo-residuals.

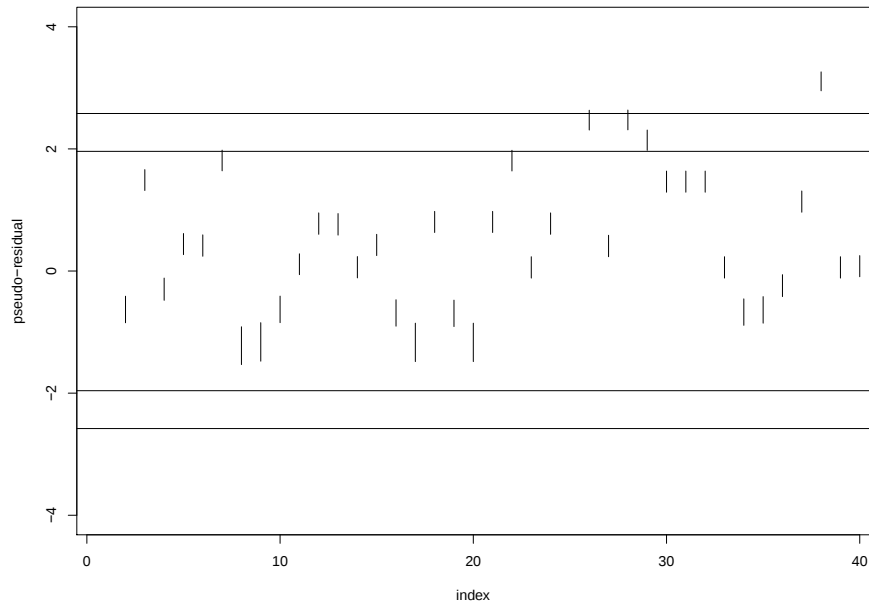


Figure 2: Normal forecast pseudo-residuals for the simulated data in exercise 4. After time 20 (corresponding to data point number 100) there is a tendency to observe larger residual values indicating that the underlying model has changed.

Exercise 5

```
# Chapter 6, R-code for exercise 5, mwp 3/2-2011
source("A1.txt")
source("A2.txt")

statdist <- function(gamma){
  m = dim(gamma)[1]
  matrix(1,1,m) %*% solve(diag(1,m) - gamma + matrix(1,m,m))
}

x <- read.table("soap.txt")
x <- data.matrix(x)
x <- as.vector(x)

# case m=1
m1= 1
lambda1= c(5)
gamma1= 1
delta1= 1
res1 <- pois.HMM.mle(x,m1,lambda1,gamma1)

# case m=2
m2= 2
lambda2= c(3,7)
```

```

gamma2= rbind(c(0.9,0.1),c(0.1,0.9))
delta2= statdist(gamma2)
res2 <- pois.HMM.mle(x,m2,lambda2,gamma2)

# case m=3
m3 = 3
lambda3 = c(3,5,7)
gamma3 = rbind(c(0.9,0.05,0.05),c(0.05,0.9,0.05),c(0.05,0.05,0.9))
delta3 = statdist(gamma3)
res3 <- pois.HMM.mle(x,m3,lambda3,gamma3)

# case m=4
m4 = 4
lambda4 = c(1,3,7,14)
gamma4 = rbind(c(0.85,0.05,0.05,0.05),c(0.05,0.85,0.05,0.05),
               c(0.05,0.05,0.85,0.05),c(0.05,0.05,0.05,0.85))
delta4 = statdist(gamma4)
res4 <- pois.HMM.mle(x,m4,lambda4,gamma4)

# Calculate pseudo residuals
npr1 <- pois.HMM.pseudo_residuals(x,m1,lambda1,gamma1,as.vector(delta1))
npr2 <- pois.HMM.pseudo_residuals(x,m2,lambda2,gamma2,as.vector(delta2))
npr3 <- pois.HMM.pseudo_residuals(x,m3,lambda3,gamma3,as.vector(delta3))
npr4 <- pois.HMM.pseudo_residuals(x,m4,lambda4,gamma4,as.vector(delta4))

# PLOTTING
postscript('diag.ps')
par(mfrow=c(4,4))
# residual plots
plot(npr1[,2,], ylim = c(-4, 4), type = "n",ylab = "",xlab = "")
abline(h = outer(c(1.96, 2.58), c(-1, 1)))
segments(x0=1:length(x), y0=npr1[,1,], y1 = npr1[,3,])
plot(npr2[,2,], ylim = c(-4, 4), type = "n",ylab = "",xlab = "")
abline(h = outer(c(1.96, 2.58), c(-1, 1)))
segments(x0=1:length(x), y0=npr2[,1,], y1 = npr2[,3,])
plot(npr3[,2,], ylim = c(-4, 4), type = "n",ylab = "",xlab = "")
abline(h = outer(c(1.96, 2.58), c(-1, 1)))
segments(x0=1:length(x), y0=npr3[,1,], y1 = npr3[,3,])
plot(npr4[,2,], ylim = c(-4, 4), type = "n",ylab = "",xlab = "")
abline(h = outer(c(1.96, 2.58), c(-1, 1)))
segments(x0=1:length(x), y0=npr4[,1,], y1 = npr4[,3,])
# QQ plots
tmp <- qqnorm(npr1[,2,], type = "n", xlab="",ylab = "",main="",
              xlim = c(-4, 4),ylim = c(-4, 4))
segments(x0=tmp$x, y0=npr1[,1,], y1 = npr1[,3,])
abline(0,1)
tmp <- qqnorm(npr2[,2,], type = "n", xlab="",ylab = "",main="",
              xlim = c(-4, 4),ylim = c(-4, 4))
segments(x0=tmp$x, y0=npr2[,1,], y1 = npr2[,3,])
abline(0,1)
tmp <- qqnorm(npr3[,2,], type = "n", xlab="",ylab = "",main="",
              xlim = c(-4, 4),ylim = c(-4, 4))
segments(x0=tmp$x, y0=npr3[,1,], y1 = npr3[,3,])
abline(0,1)

```

```
tmp <- qqnorm(npr4[2,], type = "n", xlab="", ylab = "", main="",
              xlim = c(-4, 4), ylim = c(-4, 4))
segments(x0=tmp$x, y0=npr4[1,], y1 = npr4[3,])
abline(0,1)
# histogram
hist(npr1[2,], main = "", xlab = "", ylab = "", xlim = c(-4, 4),
      ylim = c(0, 0.6), freq=FALSE)
hist(npr2[2,], main = "", xlab = "", ylab = "", xlim = c(-4, 4),
      ylim = c(0, 0.6), freq=FALSE)
hist(npr3[2,], main = "", xlab = "", ylab = "", xlim = c(-4, 4),
      ylim = c(0, 0.6), freq=FALSE)
hist(npr4[2,], main = "", xlab = "", ylab = "", xlim = c(-4, 4),
      ylim = c(0, 0.6), freq=FALSE)
# Auto correlation functions
acf(npr1[2,], main = "", xlab = "", ylab = "")
acf(npr2[2,], main = "", xlab = "", ylab = "")
acf(npr3[2,], main = "", xlab = "", ylab = "")
acf(npr4[2,], main = "", xlab = "", ylab = "")
dev.off()
```

a)

From the AIC below it seems that a 3-state HMM is the best choice, however with the BIC it seems that a 2-state HMM is superior. For the AIC one might even consider a 4-state HMM since the AIC values are close for the 3 and 4-state HMM. We will make our final conclusion in b).

```
> res1$AIC 1425.546
> res2$AIC 1245.337
> res3$AIC 1239.043 <--
> res4$AIC 1240.372

> res1$BIC 1429.035
> res2$BIC 1259.292 <--
> res3$BIC 1270.444
> res4$BIC 1296.195
```

b)

See Figure 3 for a diagnostic plot. Based on this plot the 1-state HMM is definitely not appropriate. The plots for the 2 and 3-state HMMs look somewhat alright apart from the QQ-plot where the upper quantiles are badly represented. The plots pertaining to the 4-state HMM all look healthy and therefore argues in favor of this model. So, even though the AIC and BIC suggested using 2 or 3-state HMMs we choose the 4-state HMM as the final model since its AIC and BIC values were only marginally higher than the competing models.

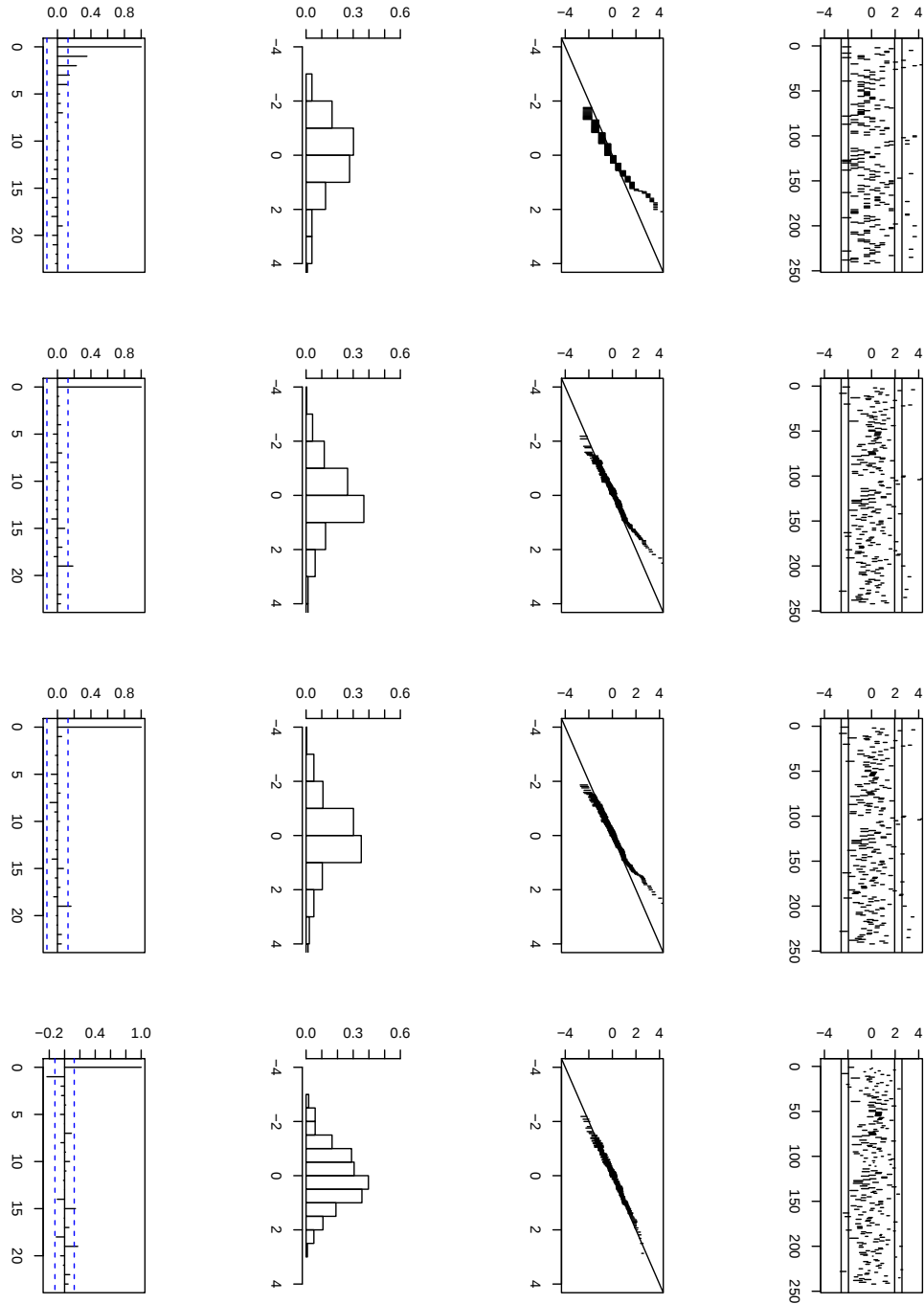


Figure 3: Diagnostic plot for the soap data based on normal ordinary pseudo-residuals similar to Figure 6.5 in Zucchini. Rows are: simple plot of residuals, QQ-plot of residuals, histogram of residuals (should be approximately normal-shaped), autocorrelation function of residuals. Columns are: HMM with 1, 2, 3 and 4 states respectively.