# Chapter 3 - Solutions to exercises

Exercises: 2,5,7,8

## Exercise 2

```
# Chapter 3, R-code for exercise 2, mwp 27/1-2011
statdist <- function(gamma){
  m = dim(gamma)[1]
  matrix(1,1,m) %*% solve(diag(1,m) - gamma + matrix(1,m,m))
}

gamma = rbind(c(0.9,0.1),c(0.2,0.8))
lambda = c(1,5)
m = length(lambda)
data = c(2,8,6,3,6,1,0,0,4,7)
T = length(data)+1
del <- statdist(gamma)

alpha = matrix(0,T,m)
alpha[1,] = del

# a) Compute the alpha vector
for(i in 2:T){
  P = rbind(c(dpois(data[i-1],lambda[1]),0),c(0,dpois(data[i-1],lambda[2])))
  alpha[i,] = alpha[i-1,] %*% gamma %*% P
}
L1 = log(sum(alpha[T,]))

# b) Compute the phi vector
phi = matrix(0,T,m)
w = matrix(0,T,1)
for(i in 1:T){
  w[i] = sum(alpha[i,])
  phi[i,] = alpha[i,] / w[i]
}
L2 = log(w[T])

# Alternative recursion for computing the likelhood
phi = matrix(0,T,m)
psi = matrix(0,T,1)
psi[1] = sum(alpha[1,])
phi[1,] = alpha[1,] / psi[1]
for(i in 2:T){
  P = rbind(c(dpois(data[i-1],lambda[1]),0),c(0,dpois(data[i-1],lambda[2])))
  foo = phi[i-1,] %*% gamma %*% P
  psi[i] = sum(foo)
  phi[i,] = foo / psi[i]
}
L3 = sum(log(psi))
```

# Exercise 5

## a)

Let $I_t$ be the indicator function for the joint event $(C_{t-1} = i, C_t = j)$. The probability of this indicator is therefore

$$\begin{aligned}\Pr(I_t) &= \Pr(C_{t-1} = i, C_t = j)\\ &= \Pr(C_t = j | C_{t-1} = i)\Pr(C_{t-1} = i)\\ &= \gamma_{ij}\delta_i.\end{aligned}$$

Since for indicator variables $\mathrm{E}(I_t) = \Pr(I_t)$ we have that the expected number of transitions, $K$, from state $i$ to $j$ in a series of $T$ observations is

$$\mathrm{E}(K) = \sum_{t=2}^{T} \mathrm{E}(I_t) = \sum_{t=2}^{T} \Pr(I_t) = \sum_{t=2}^{T} \gamma_{ij}\delta_i = (T-1)\gamma_{ij}\delta_i.$$

## b)

For $\delta_3 = 0.152$ and $T = 107$ we have

$$\begin{aligned} & (107-1)0.152\gamma_{31} &<& \quad 1\\ \Leftrightarrow \quad & \gamma_{31} &<& \quad 0.062.\end{aligned}$$

## Exercise 7

We have an $m$-state Poisson-HMM with transition probabilities $\gamma_{ij}$ and Poisson intensities $\lambda_i$ (the natural parameters). These are transformed into the working parameters $\tau_{ij}$ and $\eta_i$ using the transformation stated in Section 3.3.1 in Zucchini09.

### a)

Recall that $\tau_{ij}$ are only defined for $i \neq j$, cf. the matrix on p. 48 in Zucchini09. Then for all $i, j$ we have

$$\begin{aligned}
\frac{\partial}{\partial \tau_{ij}} \gamma_{ij} &= \frac{\partial}{\partial \tau_{ij}} \left( \frac{\rho_{ij}}{\sum_k \rho_{ik}} \right) \\
&= \frac{\partial}{\partial \tau_{ij}} \left( \frac{\exp(\tau_{ij})}{\sum_k \exp(\tau_{ik})} \right) \\
&= \frac{\frac{\partial}{\partial \tau_{ij}} [\exp(\tau_{ij})] \sum_k \exp(\tau_{ik}) - \exp(\tau_{ij}) \frac{\partial}{\partial \tau_{ij}} \left[\sum_k \exp(\tau_{ik})\right]}{\left[\sum_k \exp(\tau_{ik})\right]^2} \\
&= \frac{\exp(\tau_{ij})}{\sum_k \exp(\tau_{ik})} - \frac{\exp(\tau_{ij})^2}{\left[\sum_k \exp(\tau_{ik})\right]^2} \\
&= \gamma_{ij}(1 - \gamma_{ij}).
\end{aligned}$$

For $j \neq l$

$$\begin{aligned}
\frac{\partial}{\partial \tau_{il}} \gamma_{ij} &= \frac{\partial}{\partial \tau_{il}} \left( \frac{\exp(\tau_{ij})}{\sum_k \exp(\tau_{ik})} \right) \\
&= \frac{\frac{\partial}{\partial \tau_{il}} [\exp(\tau_{ij})] \sum_k \exp(\tau_{ik}) - \exp(\tau_{ij}) \frac{\partial}{\partial \tau_{il}} \left[\sum_k \exp(\tau_{ik})\right]}{\left[\sum_k \exp(\tau_{ik})\right]^2} \\
&= \frac{-\exp(\tau_{ij}) \exp(\tau_{il})}{\left[\sum_k \exp(\tau_{ik})\right]^2} \\
&= -\gamma_{ij}\gamma_{il}.
\end{aligned}$$

For $i \neq k$

$$\frac{\partial}{\partial \tau_{kl}} \gamma_{ij} = 0,$$

since $\gamma_{ij}$ are transformed row-wise i.e. parameters in other rows ($i \neq k$) are not influenced by row $k$, and therefore have zero partial derivative with respect to this row. Of course this can be verified using the above approach. For all $i$

$$\frac{\partial}{\partial \eta_i} \lambda_i = \frac{\partial}{\partial \eta_i} \exp(\eta_i) = \exp(\eta_i) = \lambda_i.$$

**b)**

We consider the case $m = 3$. We then have the following vector of free natural parameters

$$\boldsymbol{\theta} = (\gamma_{12}, \gamma_{13}, \gamma_{21}, \gamma_{23}, \gamma_{31}, \gamma_{32}, \lambda_1, \lambda_2, \lambda_3).$$

These are transformed to the working parameters

$$\boldsymbol{\phi} = (\tau_{12}, \tau_{13}, \tau_{21}, \tau_{23}, \tau_{31}, \tau_{32}, \eta_1, \eta_2, \eta_3).$$

Then, following the results in a) we get

$$\mathbf{M} = \begin{pmatrix} \gamma_{12}(1-\gamma_{12}) & -\gamma_{13}\gamma_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\gamma_{13}\gamma_{12}) & \gamma_{13}(1-\gamma_{13}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_{21}(1-\gamma_{21}) & -\gamma_{23}\gamma_{21} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma_{23}\gamma_{21}) & \gamma_{23}(1-\gamma_{23}) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_{31}(1-\gamma_{31}) & -\gamma_{32}\gamma_{31} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\gamma_{32}\gamma_{31}) & \gamma_{32}(1-\gamma_{32}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_3 \end{pmatrix}.$$

# Exercise 8

```
# Chapter 3, R-code for exercise 8, mwp 28/1-2011
pois.HMM.pn2pw <- function(m,lambda,gamma)
{
 tlambda <- log(lambda)
 tgamma  <- NULL
 if(m>1)
   {
   foo   <- log(gamma/diag(gamma))
   tgamma<- as.vector(foo[!diag(m)])
   }
 parvect <- c(tlambda,tgamma)
 parvect
}


pois.HMM.pw2pn <- function(m,parvect)
{
 epar   <- exp(parvect)
 lambda <- epar[1:m]
 gamma  <- diag(m)
 if(m>1)
   {
   gamma[!gamma] <- epar[(m+1):(m*m)]
   gamma         <- gamma/apply(gamma,1,sum)
   }
 delta  <- solve(t(diag(m)-gamma+1),rep(1,m))
 list(lambda=lambda,gamma=gamma,delta=delta)
}


pois.HMM.mllk <- function(parvect,x,m,...)
{
 if(m==1) return(-sum(dpois(x,exp(parvect),log=TRUE)))
 #n          <- length(x)                              # <--- OLD
 n = dim(x)[1]                                         # <--- NEW
 pn         <- pois.HMM.pw2pn(m,parvect)
 #allprobs   <- outer(x,pn$lambda,dpois)                # <--- OLD
 apfoo.low   <- outer(x[,1]-1,pn$lambda,ppois)          # <--- NEW
 apfoo.high  <- outer(x[,2],pn$lambda,ppois)            # <--- NEW
 allprobs <- apfoo.high - apfoo.low
 allprobs   <- ifelse(!is.na(allprobs),allprobs,1)
 lscale     <- 0
 foo        <- pn$delta
 for (i in 1:n)
   {
   foo    <- foo%*%pn$gamma*allprobs[i,]
   sumfoo <- sum(foo)
   lscale <- lscale+log(sumfoo)
   foo    <- foo/sumfoo
   }
 mllk       <- -lscale
 mllk
```

```
}


pois.HMM.mle <- function(x,m,lambda0,gamma0,...)
{
 parvect0  <- pois.HMM.pn2pw(m,lambda0,gamma0)
 mod       <- nlm(pois.HMM.mllk,parvect0,x=x,m=m)
 pn        <- pois.HMM.pw2pn(m,mod$estimate)
 mllk      <- mod$minimum
 np        <- length(parvect0)
 AIC       <- 2*(mllk+np)
 n         <- sum(!is.na(x))
 BIC       <- 2*mllk+np*log(n)
 list(lambda=pn$lambda,gamma=pn$gamma,delta=pn$delta,
              code=mod$code,mllk=mllk,AIC=AIC,BIC=BIC)
}

# Script for testing the function
gamma = rbind(c(0.9,0.1),c(0.2,0.8))
lambda = c(1,5)
m = length(lambda)
x = t(rbind(c(2,8,6,3,6,1,0,0,4,7),c(3,15,10,3,8,3,2,5,Inf,25)))

parvect <- pois.HMM.pn2pw(m,lambda,gamma)
mllk <- pois.HMM.mllk(parvect,x,m)
```