MA461 Assignment 2

Clodagh Murray

2/15/2022

Question: What is the probability that the autoimmune condition is active in month 6?

Forward Algorithm

```
# Given:
Symbols <- c("low", "medium", "high")
state = c("active", "dormant")
observations <- c("low", "high", "medium", "low", "high", "high", "high", "medium")
emission_probs <- matrix(c(0.7,0.2,0.1,0.2,0.3,0.5), nrow =2, byrow=T)
dimnames(emission_probs) <- list(state, Symbols)
emission_probs
```

```
##         low medium high
## active  0.7    0.2  0.1
## dormant 0.2    0.3  0.5
```

```
TPM <- matrix(c(1/2, 1/2, 1/5, 4/5), nrow = 2, byrow=T)
dimnames(TPM) <- list(state,state)
TPM
```

```
##         active dormant
## active     0.5     0.5
## dormant    0.2     0.8
```

```
# define stationary vector for starting state
pi = eigen(t(TPM))$vectors[,1]
stat = pi/sum(pi)
stat #[Active, Dormant]
```

```
## [1] 0.2857143 0.7142857
```

```
# define empty matrix for alpha
alpha = matrix(nrow = 2, ncol=8)
colnames(alpha) <- observations
rownames(alpha) <- state

alpha
```

```
##         low high medium low high high high medium
## active   NA   NA     NA  NA   NA   NA   NA     NA
## dormant  NA   NA     NA  NA   NA   NA   NA     NA
```

```
# Numerics for function
nObservations <- length(observations)
nStates <- length(state)

# Initialize column 1 of alpha using stationary distribution
for(i in 1:nStates){
  alpha[i,1] = stat[i]*emission_probs[i][1]
}
alpha
```

```
##               low high medium low high high high medium
## active  0.2000000   NA     NA  NA   NA   NA   NA     NA
## dormant 0.1428571   NA     NA  NA   NA   NA   NA     NA
```

```
# Forward algorithm

for(i in 2:nObservations){
    if (colnames(alpha)[i] == "low"){
      m=1} # m will be used to point at emission prob entries for each iteration
      else if (colnames(alpha)[i] == "medium"){
      m=2}
      else{
      m=3}
  # Loop over all entries of alpha
  for (a in 1:2){
    alpha[a,i] = emission_probs[a,m] *
      sum((TPM[1,a]*alpha[1,i-1]),(TPM[2,a]*alpha[2,i-1]))
    }
}

alpha
```

```
##               low       high      medium     low      high       high
## active  0.2000000 0.01285714 0.005571429 0.00582 0.0003906 0.00008847
## dormant 0.1428571 0.10714286 0.027642857 0.00498 0.0034470 0.00147645
##                 high       medium
## active  0.0000339525 2.790315e-05
## dormant 0.0006126975 1.521403e-04
```

Backward Algorithm

```
# Initialize beta

beta =  matrix(nrow=2, ncol=8)
colnames(beta) <- observations
rownames(beta) <- state
```

```
beta[,8]=1 # final column = 1
beta
```

```
##         low high medium low high high high medium
## active   NA   NA     NA  NA   NA   NA   NA      1
## dormant  NA   NA     NA  NA   NA   NA   NA      1
```

```
# Backwards algorithm
for(o in (nObservations-1):1){ # filling in from column 7:1
      if (colnames(alpha)[o+1] == "low"){
      m=1    # point to emission prob entries for next obs in sequence (o+1)
      }
    else if (colnames(alpha)[o+1] == "medium"){
      m=2
      }
    else{
      m=3
      }
    for(k in 1:nStates){
    beta[k,o] = sum(beta[,o+1]*TPM[k,]*emission_probs[1:2,m])
  }
}
beta
```

```
##                  low        high      medium         low     high   high high
## active  0.0004491686 0.001453365 0.006828188 0.01378125 0.033375 0.0825 0.25
## dormant 0.0006314679 0.001506002 0.005136975 0.02004750 0.048450 0.1170 0.28
##         medium
## active       1
## dormant      1
```

Forward-Backward Algorithm

Forward-backward algorithm for entry 6 in active state i.e probability that the autoimmune condition is active in month 6

Equation: Fm(i)Bm(i)/P(x)

```
fb6 <- (alpha[1,][6]*beta[1,][6])/((alpha[1,][6]*beta[1,][6])+(alpha[2,][6]*beta[2,][6]))
fb6
```

```
##       high
## 0.04053897
```

Verify Using HMM package

```
library(HMM)
```

```r
observed <- c("low", "high", "medium", "low", "high", "high", "high", "medium")
Symbols <- c("low", "medium", "high")
States = c("active", "dormant")
transprobs <- matrix(c(1/2, 1/2, 1/5, 4/5), nrow = 2, byrow=T)

pi = eigen(t(transprobs))
pi = pi$vectors[,1]
pi = pi/sum(pi)

emmissionprobs <- matrix(c(0.7,0.2,0.1,0.2,0.3,0.5), nrow =2, byrow=T)
hmm <- initHMM(c("active", "dormant"), c("low", "medium", "high"), startProbs =
                 pi, matrix(c(1/2, 1/2, 1/5, 4/5), nrow = 2, byrow=T),
               matrix(c(0.7,0.2,0.1,0.2,0.3,0.5), nrow =2, byrow=T) )

forward <- forward(hmm, observation = observed)
forward <- exp(forward)
forward
```

```
##          index
## states            1          2           3       4         5          6
##   active  0.2000000 0.01285714 0.005571429 0.00582 0.0003906 0.00008847
##   dormant 0.1428571 0.10714286 0.027642857 0.00498 0.0034470 0.00147645
##          index
## states               7            8
##   active  0.0000339525 2.790315e-05
##   dormant 0.0006126975 1.521403e-04
```

```r
backward <- backward(hmm, observation = observed)
backward <- exp(backward)

fb_6 <- (forward[1,][6]*backward[1,][6])/
  ((forward[1,][6]*backward[1,][6])+(forward[2,][6]*backward[2,][6]))
fb_6
```

```
##          6
## 0.04053897
```

```r
post <- posterior(hmm, observed) # built in forward-backward algorithm function
```

```r
post[1,6]
```

```
## [1] 0.04053897
```