

Hidden Markov Lab2

Aman Kumar Nayak

9/20/2020

Task

Model the behavior of a robot that walks around a ring. The ring is divided into 10 sectors. At any given time point, the robot is in one of the sectors and decides with equal probability to stay in that sector or move to the next sector. You do not have direct observation of the robot. However, the robot is equipped with a tracking device that you can access. The device is not very accurate though: If the robot is in the sector i , then the device will report that the robot is in the sectors $[i - 2, i + 2]$ with equal probability.

Question 1

```
#Building Hidden Markov Model
library(HMM)

States = 1:10
Symbols = 1:10

#Since
#the robot is in one of the sectors and decides with equal probability to stay in that sector or move to
#we get probability of staying as 0.5 and probability of moving as 0.5

transProbs = 0.5*diag(10)
for(i in 1:ncol(transProbs))
{
  if(i != ncol(transProbs))
  {
    transProbs[i+1 , i] = 0.5
  }else{
    transProbs[1 , i] = 0.5
  }
}

colnames(transProbs) = States
rownames(transProbs) = States

#emissionProbs
#If the robot is in the sector i, then the device will report that the robot is in the sectors [i - 2,
#so robot can be in any sector i with probability of 0.2 (5/10)

emissionProbs = 0.2 * diag(10)
n = ncol(emissionProbs)
for(i in 1:n){
```

```

if(i-1 == 0) {
  Col_Indx = c((n-1) , n , i , i+1 , i+2)
}else if(i-1 == 1){
  Col_Indx = c( n , i-1 ,i , i+1 , i+2)
}else if(i+1 == n){
  Col_Indx = c( i-2 ,i-1 ,i , i+1 , 1)
}else if(i+1 > n) {
  Col_Indx = c( i-2 ,i-1 ,i , 1 , 2)
}else{
  Col_Indx = c( i-2 , i-1 ,i , i+1 , i+2)
}
emissionProbs[i , Col_Indx] = 0.2
}#for

```

#Hidden Markov Model

```

hmm = HMM::initHMM(States , Symbols , transProbs = transProbs , emissionProbs = emissionProbs)

hmm

```

```

## $States
## [1] 1 2 3 4 5 6 7 8 9 10
##
## $Symbols
## [1] 1 2 3 4 5 6 7 8 9 10
##
## $startProbs
## 1 2 3 4 5 6 7 8 9 10
## 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
##
## $transProbs
## to
## from 1 2 3 4 5 6 7 8 9 10
## 1 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5
## 2 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## 3 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## 4 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0 0.0
## 5 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0 0.0
## 6 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0 0.0
## 7 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0 0.0
## 8 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0 0.0
## 9 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5 0.0
## 10 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.5 0.5
##
## $emissionProbs
## symbols
## states 1 2 3 4 5 6 7 8 9 10
## 1 0.2 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.2 0.2
## 2 0.2 0.2 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.2
## 3 0.2 0.2 0.2 0.2 0.2 0.0 0.0 0.0 0.0 0.0
## 4 0.0 0.2 0.2 0.2 0.2 0.2 0.0 0.0 0.0 0.0
## 5 0.0 0.0 0.2 0.2 0.2 0.2 0.2 0.0 0.0 0.0
## 6 0.0 0.0 0.0 0.2 0.2 0.2 0.2 0.2 0.0 0.0

```

```
##      7  0.0 0.0 0.0 0.0 0.0 0.2 0.2 0.2 0.2 0.2 0.0
##      8  0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.2 0.2 0.2 0.2
##      9  0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.2 0.2 0.2
##     10 0.2 0.2 0.0 0.0 0.0 0.0 0.0 0.0 0.2 0.2 0.2
```

Question 2

```
suppressWarnings(RNGversion("3.5.1"))
set.seed(12345)
nSim = 100
simulateHmm = simHMM(hmm , nSim)
```

```
cat("Simulated Values")
```

```
## Simulated Values
```

```
cat("\n")
```

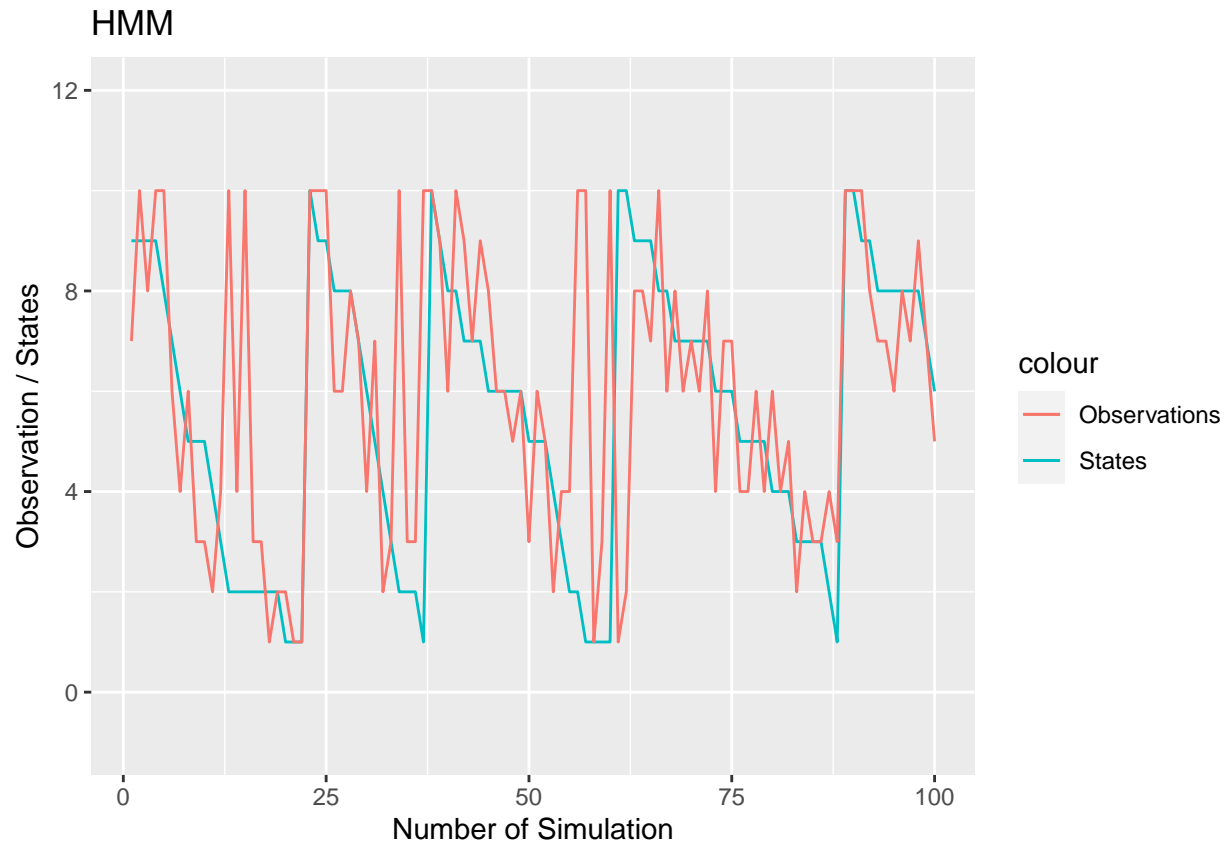
```
simulateHmm
```

```
## $states
##  [1]  9  9  9  9  8  7  6  5  5  5  4  3  2  2  2  2  2  2  1  1  1 10  9  9
## [26]  8  8  8  7  6  5  4  3  2  2  2  1 10  9  8  8  7  7  7  6  6  6  6  5
## [51]  5  5  4  3  2  2  1  1  1  1 10 10  9  9  9  8  8  7  7  7  7  6  6  6
## [76]  5  5  5  5  4  4  4  3  3  3  3  2  1 10 10  9  9  8  8  8  8  8  7  6
##
## $observation
##  [1]  7 10  8 10 10  6  4  6  3  3  2  4 10  4 10  3  3  1  2  2  1  1 10 10 10
## [26]  6  6  8  7  4  7  2  3 10  3  3 10 10  9  6 10  9  7  9  8  6  6  5  6  3
## [51]  6  5  2  4  4 10 10  1  3 10  1  2  8  8  7 10  6  8  6  7  6  8  4  7  7
## [76]  4  4  6  4  6  4  5  2  4  3  3  4  3 10 10 10  8  7  7  6  8  7  9  7  5
```

```
cat("\n")
```

```
library(ggplot2)

ggplot() +
  geom_line(aes(x = 1 : nSim , y = simulateHmm$states , color = "States"))+
  geom_line(aes(x = 1 : nSim , y = simulateHmm$observation , color = "Observations"))+
  #geom_line(aes(x = simulateHmm$states , y = simulateHmm$observation , color = "Observations"))+
  ylim(-1 , 12)+
  xlab("Number of Simulation")+
  ylab("Observation / States")+
  ggtitle("HMM")
```



Question 3

Discard the hidden states from the sample obtained above. Use the remaining observations to compute the filtered and smoothed probability distributions for each of the 100 time points. Compute also the most probable path.

Filtered and Smoothed Probability

Now in order to calculate filtered probability, I am using `forward(hmm, observation)`

In order to calculate, smoothed probabilities, we can use `posterior(hmm, observation)` where The posterior probability of being in a state X at time k can be computed from the forward and backward probabilities.

Now most probable path is calculated considering that bot can move from State X to State $X \pm 1$

```
filteredProb = HMM::forward(hmm , simulateHmm$observation)
smoothedProb = HMM::posterior(hmm , simulateHmm$observation)
```

References

1. Package “HMM”

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
#Building Hidden Markov Model
library(HMM)
```

```

States = 1:10
Symbols = 1:10

#Since
#the robot is in one of the sectors and decides with equal probability to stay in that sector or move to another sector
#we get probability of staying as 0.5 and probability of moving as 0.5

transProbs = 0.5*diag(10)
for(i in 1:ncol(transProbs))
{
  if(i != ncol(transProbs))
  {
    transProbs[i+1 , i] = 0.5
  }else{
    transProbs[1 , i] = 0.5
  }
}

colnames(transProbs) = States
rownames(transProbs) = States

#emissionProbs
#If the robot is in the sector i, then the device will report that the robot is in the sectors [i - 2, i - 1, i, i + 1, i + 2]
#so robot can be in any sector i with probability of 0.2 (5/10)

emissionProbs = 0.2 * diag(10)
n = ncol(emissionProbs)
for(i in 1:n){
  if(i-1 == 0) {
    Col_Indx = c((n-1) , n , i , i+1 , i+2)
  }else if(i-1 == 1){
    Col_Indx = c( n , i-1 , i , i+1 , i+2)
  }else if(i+1 == n){
    Col_Indx = c( i-2 , i-1 , i , i+1 , 1)
  }else if(i+1 > n) {
    Col_Indx = c( i-2 , i-1 , i , 1 , 2)
  }else{
    Col_Indx = c( i-2 , i-1 , i , i+1 , i+2)
  }
  emissionProbs[i , Col_Indx] = 0.2
}#for

#Hidden Markov Model

hmm = HMM::initHMM(States , Symbols , transProbs = transProbs , emissionProbs = emissionProbs)

hmm

suppressWarnings(RNGversion("3.5.1"))
set.seed(12345)
nSim = 100

```

```

simulateHmm = simHMM(hmm , nSim)

cat("Simulated Values")
cat("\n")
simulateHmm
cat("\n")
library(ggplot2)

ggplot() +
  geom_line(aes(x = 1 : nSim , y = simulateHmm$states , color = "States"))+
  geom_line(aes(x = 1 : nSim , y = simulateHmm$observation , color = "Observations"))+
  #geom_line(aes(x = simulateHmm$states , y = simulateHmm$observation , color = "Observations"))+
  ylim(-1 , 12)+
  xlab("Number of Simulation")+
  ylab("Observation / States")+
  ggtitle("HMM")

filteredProb = HMM::forward(hmm , simulateHmm$observation)

smoothedProb = HMM::posterior(hmm , simulateHmm$observation)

```