Quiz 2 - 32376881

Dhun Sheth

2023-12-13

Question 1

Probability would be 0.6 because of the Markov property (MC is memory less and next state only depends on current state).

Question 2

```r
P <- matrix(c(0, .5, 0.5,
              0.5, 0, 0.5,
              0.5, 0.5, 0), nrow = 3, byrow = TRUE)
print(P)
```

```
##      [,1] [,2] [,3]
## [1,]  0.0  0.5  0.5
## [2,]  0.5  0.0  0.5
## [3,]  0.5  0.5  0.0
```

```r
print(P[1,1])
```

a)

```
## [1] 0
```

$P(x_1 = 1|x_0 = 1) = 0$

```r
P_3 <- P%*%P%*%P
print(P_3)
```

b)

```
##       [,1]  [,2]  [,3]
## [1,] 0.250 0.375 0.375
## [2,] 0.375 0.250 0.375
## [3,] 0.375 0.375 0.250
```

$P(x_3 = 1|x_0 = 1) = 0.25$

c) States 1 and 3 communicate because you can go from state 1 directly to state 3 with probability 0.5 and you can also go directly from state 3 to state 1 with probability 0.5, so states 1 and 3 do communicate with each other.

d) Because all states in S communicate with each other, the state space S is irreducible.

```
A <- t(P) - diag(rep(1,3))
A <- rbind(A, rep(1,3))
RHS <- c(rep(0,3), 1)
stationary_dist <- qr.solve(A, RHS) # no longer a square system

print(stationary_dist)
```

e)

```
## [1] 0.3333333 0.3333333 0.3333333
```

```
set.seed(12345)
Ntransitions <- 100000
X <- numeric(Ntransitions)
current.state <- 1
for (t in 1:Ntransitions) {
  current.state <- sample(1:3, size = 1, prob = P[current.state, ])
  X[t] <- current.state
}

simulated_dist <- table(X)/Ntransitions
print(simulated_dist)
```

f)

```
## X
##       1       2       3
## 0.33437 0.33213 0.33350
```

The stationary distribution derived through simulation is very similar to the calculated stationary distribution found in part e.

Question 3

Run below before conintue

```
P <- matrix(c(1, 8, 7, 2, 0, 3, 7, 2, 0), nrow=3)/10
PE <- matrix(c(1, 2, 3, 1, 2, 1, 8, 6, 4,  0, 0, 2), nrow=3)/10
observed_y <- c(2, 4, 3, 3, 3, 3, 3, 3, 2, 3, 1, 3, 2, 3, 1, 3, 3, 4, 1)
```

```r
#your solution here

hmm <- initHMM(c("1","2","3"), c("1","2", "3","4"),
               transProbs= P,
               emissionProbs= PE)
observations <- as.character(observed_y)
simDataFit <- viterbiTraining(hmm, observations)
print("Transition Matrix")
```

a)

```
## [1] "Transition Matrix"
```

```r
new_P <- simDataFit$hmm$transProbs
print(simDataFit$hmm$transProbs)
```

```
##     to
## from    1    2 3
##    1 0.00 0.00 1
##    2 1.00 0.00 0
##    3 0.75 0.25 0
```

```r
print("Emission Matrix")
```

```
## [1] "Emission Matrix"
```

```r
new_PE <- simDataFit$hmm$emissionProbs
print(simDataFit$hmm$emissionProbs)
```

```
##       symbols
## states         1         2         3    4
##      1 0.1111111 0.1111111 0.7777778 0.00
##      2 0.0000000 0.5000000 0.5000000 0.00
##      3 0.2500000 0.1250000 0.3750000 0.25
```

```r
A <- t(new_P) - diag(rep(1,3))
A <- rbind(A, rep(1,3))
RHS <- c(rep(0,3), 1)
stationary_dist <- qr.solve(A, RHS) # no longer a square system

print(stationary_dist)
```

b)

```
##         1         2         3
## 0.4444444 0.1111111 0.4444444
```

Used the updated transition matrix (P) from part 3a to find the stationary distribution.

c) The hidden markov chain is in state 3 with probability 0.44 based on the stationary distribution and based on the new emission probabilities $P(Y = 4|X = 3) = 0.25$, thus the long run probability would be $0.25 * 0.44 = 0.1111111$

d) If we are given Y1=1 then based on the updated emission probabilities X would have been 3 because for X={1,2} Y cannot be in state 4. Then if we are looking for probability that Y2=4 given Y1=4 (X0=3) then it would mean X would have to equal 3 again for Y2=4, however, based on the updated transition probabilities if X is in state 3 it must go to either state 1 or 2, however, if X1={1,2} then Y2 cannot equal 4 therefore $P(Y_2 = 4|Y_1 = 4) = 0$.