Report Lab2

Milda Poceviciute

27 September 2018

Question 1

Based on the scenario described in the lab instruction, the hidden Markov model (HMM) is:

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

```
##   Sector 1       0.2      0.2      0.2      0.0      0.0      0.0      0.0
##   Sector 2       0.2      0.2      0.2      0.2      0.0      0.0      0.0
##   Sector 3       0.2      0.2      0.2      0.2      0.2      0.0      0.0
##   Sector 4       0.0      0.2      0.2      0.2      0.2      0.2      0.0
##   Sector 5       0.0      0.0      0.2      0.2      0.2      0.2      0.2
##   Sector 6       0.0      0.0      0.0      0.2      0.2      0.2      0.2
##   Sector 7       0.0      0.0      0.0      0.0      0.2      0.2      0.2
##   Sector 8       0.0      0.0      0.0      0.0      0.0      0.2      0.2
##   Sector 9       0.2      0.0      0.0      0.0      0.0      0.0      0.2
##   Sector 10      0.2      0.2      0.0      0.0      0.0      0.0      0.0
##            symbols
## states      Sector 8 Sector 9 Sector 10
##   Sector 1       0.0      0.2       0.2
##   Sector 2       0.0      0.0       0.2
##   Sector 3       0.0      0.0       0.0
##   Sector 4       0.0      0.0       0.0
##   Sector 5       0.0      0.0       0.0
##   Sector 6       0.2      0.0       0.0
##   Sector 7       0.2      0.2       0.0
##   Sector 8       0.2      0.2       0.2
##   Sector 9       0.2      0.2       0.2
##   Sector 10      0.2      0.2       0.2
```

Question 2

In order to simulate from the HMM model from Q1, the function simHMM is used. The simulated Z^i and corresponding X^i are:

```
## $states
##   [1] "Sector 9"  "Sector 9"  "Sector 9"  "Sector 9"  "Sector 10"
##   [6] "Sector 1"  "Sector 2"  "Sector 2"  "Sector 2"  "Sector 2"
##  [11] "Sector 3"  "Sector 3"  "Sector 4"  "Sector 4"  "Sector 4"
##  [16] "Sector 4"  "Sector 4"  "Sector 4"  "Sector 4"  "Sector 5"
##  [21] "Sector 6"  "Sector 6"  "Sector 7"  "Sector 8"  "Sector 9"
##  [26] "Sector 10" "Sector 10" "Sector 10" "Sector 1"  "Sector 2"
##  [31] "Sector 2"  "Sector 3"  "Sector 3"  "Sector 4"  "Sector 4"
##  [36] "Sector 4"  "Sector 5"  "Sector 5"  "Sector 5"  "Sector 6"
##  [41] "Sector 7"  "Sector 7"  "Sector 8"  "Sector 9"  "Sector 10"
##  [46] "Sector 1"  "Sector 2"  "Sector 3"  "Sector 3"  "Sector 4"
##  [51] "Sector 5"  "Sector 5"  "Sector 6"  "Sector 6"  "Sector 7"
##  [56] "Sector 7"  "Sector 8"  "Sector 8"  "Sector 8"  "Sector 8"
##  [61] "Sector 9"  "Sector 10" "Sector 10" "Sector 10" "Sector 10"
##  [66] "Sector 1"  "Sector 1"  "Sector 2"  "Sector 2"  "Sector 2"
##  [71] "Sector 2"  "Sector 2"  "Sector 3"  "Sector 3"  "Sector 3"
##  [76] "Sector 4"  "Sector 5"  "Sector 5"  "Sector 5"  "Sector 6"
##  [81] "Sector 7"  "Sector 8"  "Sector 8"  "Sector 8"  "Sector 8"
##  [86] "Sector 8"  "Sector 9"  "Sector 9"  "Sector 9"  "Sector 10"
##  [91] "Sector 10" "Sector 10" "Sector 1"  "Sector 1"  "Sector 1"
##  [96] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 2"  "Sector 3"
##
## $observation
##   [1] "Sector 7"  "Sector 10" "Sector 8"  "Sector 10" "Sector 2"
##   [6] "Sector 3"  "Sector 10" "Sector 3"  "Sector 4"  "Sector 4"
##  [11] "Sector 5"  "Sector 4"  "Sector 2"  "Sector 3"  "Sector 2"
```

```
##  [16] "Sector 6"  "Sector 6"  "Sector 5"  "Sector 4"  "Sector 3"
##  [21] "Sector 5"  "Sector 5"  "Sector 8"  "Sector 9"  "Sector 10"
##  [26] "Sector 9"  "Sector 9"  "Sector 10" "Sector 2"  "Sector 10"
##  [31] "Sector 2"  "Sector 5"  "Sector 3"  "Sector 2"  "Sector 6"
##  [36] "Sector 6"  "Sector 4"  "Sector 7"  "Sector 7"  "Sector 6"
##  [41] "Sector 5"  "Sector 9"  "Sector 7"  "Sector 10" "Sector 10"
##  [46] "Sector 3"  "Sector 3"  "Sector 1"  "Sector 3"  "Sector 3"
##  [51] "Sector 6"  "Sector 5"  "Sector 4"  "Sector 7"  "Sector 7"
##  [56] "Sector 9"  "Sector 9"  "Sector 10" "Sector 6"  "Sector 9"
##  [61] "Sector 10" "Sector 2"  "Sector 9"  "Sector 9"  "Sector 8"
##  [66] "Sector 1"  "Sector 3"  "Sector 2"  "Sector 3"  "Sector 4"
##  [71] "Sector 3"  "Sector 2"  "Sector 5"  "Sector 4"  "Sector 4"
##  [76] "Sector 2"  "Sector 4"  "Sector 6"  "Sector 4"  "Sector 6"
##  [81] "Sector 8"  "Sector 10" "Sector 8"  "Sector 7"  "Sector 6"
##  [86] "Sector 6"  "Sector 7"  "Sector 8"  "Sector 9"  "Sector 10"
##  [91] "Sector 1"  "Sector 9"  "Sector 2"  "Sector 2"  "Sector 3"
##  [96] "Sector 9"  "Sector 2"  "Sector 10" "Sector 4"  "Sector 1"
```

Question 3

Definitions of the filtering and smoothing probability distributions are as follows:

Filtering: $p(Z^t|x^{0:t}) = \frac{\alpha(Z^t)}{\sum_{z^t} \alpha(z^t)}$

Smoothing: $p(Z^t|x^{0:t}) = \frac{\alpha(Z^t)\beta(Z^t)}{\sum_{z^t} \alpha(z^t)\beta(z^t)}$

In the HMM package, the $log(\alpha)$ is returned by the *forward* while the $log(\beta)$ is returned by the *backward* functions.

```
## Filtering probabilities:
```

```
##            index
## states        1   2         3         4         5          6
##   Sector 1  0.0 0.0 0.0000000 0.1666667 0.3846154 0.58974359
##   Sector 2  0.0 0.0 0.0000000 0.0000000 0.1153846 0.33333333
##   Sector 3  0.0 0.0 0.0000000 0.0000000 0.0000000 0.07692308
##   Sector 4  0.0 0.0 0.0000000 0.0000000 0.0000000 0.00000000
##   Sector 5  0.2 0.0 0.0000000 0.0000000 0.0000000 0.00000000
##   Sector 6  0.2 0.0 0.0000000 0.0000000 0.0000000 0.00000000
##   Sector 7  0.2 0.0 0.0000000 0.0000000 0.0000000 0.00000000
##   Sector 8  0.2 0.4 0.2222222 0.1111111 0.0000000 0.00000000
##   Sector 9  0.2 0.4 0.4444444 0.3333333 0.0000000 0.00000000
##   Sector 10 0.0 0.2 0.3333333 0.3888889 0.5000000 0.00000000
```

```
## Smoothing probabilities:
```

```
##            index
## states              1         2          3         4          5         6
##   Sector 1  0.0000000 0.0000000 0.00000000 0.2003536 0.50540765 0.7888311
##   Sector 2  0.0000000 0.0000000 0.00000000 0.0000000 0.04873128 0.2111689
##   Sector 3  0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.0000000
##   Sector 4  0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.0000000
##   Sector 5  0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.0000000
##   Sector 6  0.0000000 0.0000000 0.00000000 0.0000000 0.00000000 0.0000000
##   Sector 7  0.1534318 0.0000000 0.00000000 0.0000000 0.00000000 0.0000000
```

```
##   Sector 8  0.4241889 0.3068636 0.06859401 0.0000000 0.00000000 0.0000000
##   Sector 9  0.4223794 0.5415141 0.47653910 0.2057820 0.00000000 0.0000000
##   Sector 10 0.0000000 0.1516223 0.45486689 0.5938644 0.44586107 0.0000000
```

The Viterbi algorithm in HMM package is used to find the most probable path:

```
##   [1] "Sector 8"  "Sector 9"  "Sector 10" "Sector 1"  "Sector 1"
##   [6] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 2"  "Sector 2"
##  [11] "Sector 3"  "Sector 3"  "Sector 3"  "Sector 3"  "Sector 3"
##  [16] "Sector 4"  "Sector 4"  "Sector 4"  "Sector 4"  "Sector 5"
##  [21] "Sector 6"  "Sector 7"  "Sector 8"  "Sector 9"  "Sector 10"
##  [26] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 1"  "Sector 1"
##  [31] "Sector 2"  "Sector 3"  "Sector 3"  "Sector 3"  "Sector 4"
##  [36] "Sector 4"  "Sector 4"  "Sector 5"  "Sector 5"  "Sector 6"
##  [41] "Sector 7"  "Sector 8"  "Sector 9"  "Sector 10" "Sector 1"
##  [46] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 2"  "Sector 3"
##  [51] "Sector 4"  "Sector 4"  "Sector 4"  "Sector 5"  "Sector 6"
##  [56] "Sector 7"  "Sector 7"  "Sector 8"  "Sector 8"  "Sector 8"
##  [61] "Sector 9"  "Sector 10" "Sector 10" "Sector 10" "Sector 10"
##  [66] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 1"  "Sector 2"
##  [71] "Sector 2"  "Sector 2"  "Sector 3"  "Sector 3"  "Sector 3"
##  [76] "Sector 3"  "Sector 3"  "Sector 4"  "Sector 5"  "Sector 6"
##  [81] "Sector 7"  "Sector 8"  "Sector 8"  "Sector 8"  "Sector 8"
##  [86] "Sector 8"  "Sector 9"  "Sector 10" "Sector 1"  "Sector 1"
##  [91] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 1"  "Sector 1"
##  [96] "Sector 1"  "Sector 1"  "Sector 1"  "Sector 2"  "Sector 2"
```

Question 4

```
## Accuracy of Filtered probability distribution:
```

```
## $accuracy
## TRUE
## 0.52
##
## $table
## compare
## FALSE  TRUE
##    48    52
```

```
## Accuracy of Smoothed probability distribution:
```

```
## $accuracy
## TRUE
## 0.74
##
## $table
## compare
## FALSE  TRUE
##    26    74
```

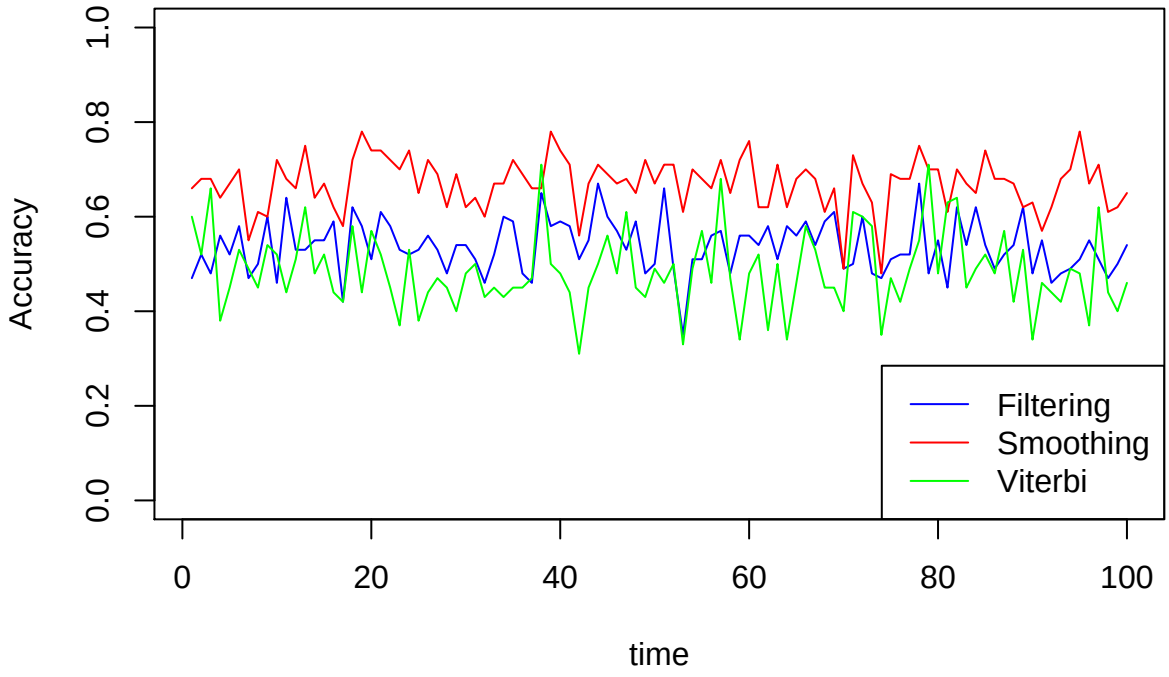
```
## Accuracy of the Viterbi algorithm:
```

```
## $accuracy
## TRUE
## 0.56
##
```

```
## $table
## compare
## FALSE  TRUE
##    44    56
```

From the accuracy rates above, I conclude that the Smoothing probability distribution yields the highest accuracy.

Question 5

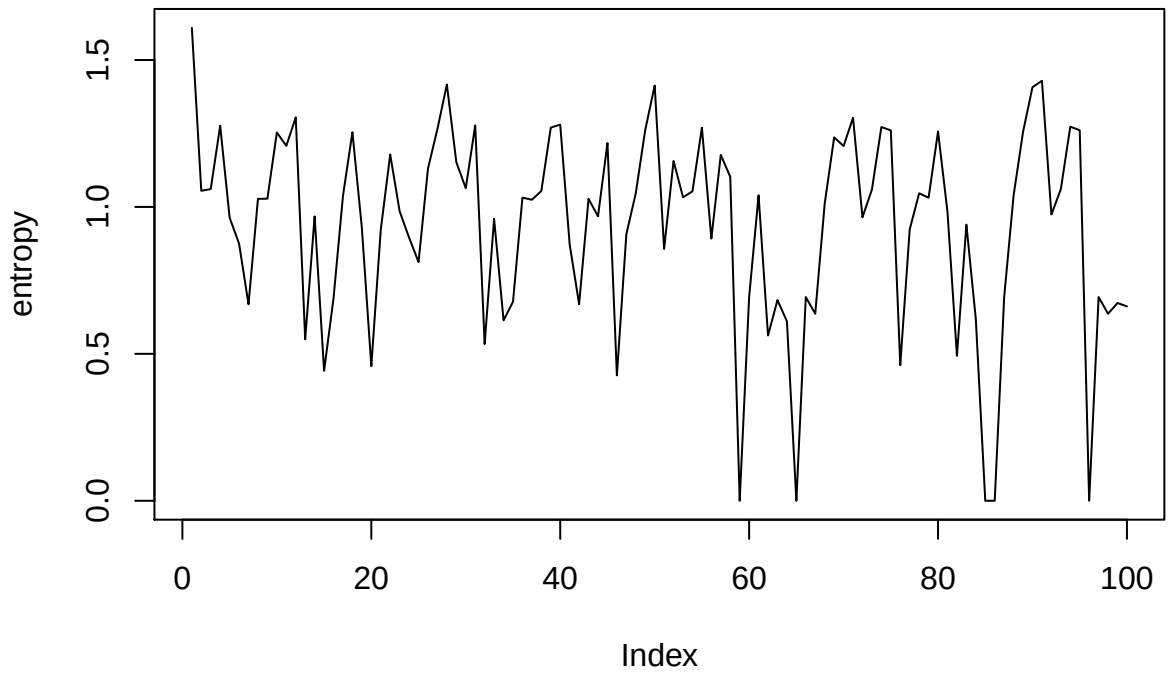


From the plot above, I conclude that the path estimated by the Smoothing probability distribution performs better (has higher accuracy) in most cases. Viterbi and Filtering distribution seem to be fluctuating similarly, hence it is not possible to conclude which, if any, is more accurate.

This is consistent with the fact that Smoothing probability distribution includes all data points from the past and future: that is, when Z^t is estimated where $0 < t < T$, the observations $x^{0:T}$ are considered. The filtering probability distribution only includes the past data points, i.e. $x^{0:t}$. The access to additional data makes Smoothing probability distribution more accurate.

The Viterbi algorithm finds the path that has highest probability but also is plausible, while Smoothing and Filtering produces the paths that have highest probability (but they might not be plausible). That's why Smoothing may show better accuracy than the Viterbi algorithm.

Question 6



The plot above shows how Shannon entropy is changing as I add more data. The lower the Shannon entropy, the more information that particular distribution contains. In other words, the more sure it is where the robot is at that time point. However, there is no visible tendency for the entropy measurement to decrease as t increases. Hence, I conclude that having more data observed does not make the estimation of the robot's position more accurate.

Question 7

In this question I am estimating the probability $P(Z^{t+1}|X^{1:t})$.

It is given by the probability distribution of the robot's position at time t multiplied by the transition matrix:

```
## The probability distribution of the latent variable at time t+1 is:
```

```
##          [,1]
##  [1,] 0.0000
##  [2,] 0.1875
##  [3,] 0.5000
##  [4,] 0.3125
##  [5,] 0.0000
##  [6,] 0.0000
##  [7,] 0.0000
##  [8,] 0.0000
##  [9,] 0.0000
## [10,] 0.0000
```

Hence, it is most likely that the robot would be standing at the sector 3.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
library(HMM)
library(entropy)
set.seed(12345)
# Create states
grid_states <- c(paste0("Sector ",1:10))
Readings <- grid_states

# Building states probabilities
Tprobs1 <- c(rep.int(0.5,2),rep.int(0,8))
Tprobs2 <- c(0,rep.int(0.5,2),rep.int(0,7))
Tprobs3 <- c(rep.int(0,2),rep.int(0.5,2),rep.int(0,6))
Tprobs4 <- c(rep.int(0,3),rep.int(0.5,2),rep.int(0,5))
Tprobs5 <- c(rep.int(0,4),rep.int(0.5,2),rep.int(0,4))
Tprobs6 <- c(rep.int(0,5),rep.int(0.5,2),rep.int(0,3))
Tprobs7 <- c(rep.int(0,6),rep.int(0.5,2),rep.int(0,2))
Tprobs8 <- c(rep.int(0,7),rep.int(0.5,2),rep.int(0,1))
Tprobs9 <- c(rep.int(0,8),rep.int(0.5,2))
Tprobs10 <- c(0.5,rep.int(0,8),0.5)

transProbs <- matrix(c(Tprobs1,Tprobs2,Tprobs3,Tprobs4,Tprobs5,Tprobs6,Tprobs7,Tprobs8,Tprobs9,Tprobs10)

# Building Observed readings probabilities
probs1 <- c(rep.int(0.2,3),rep.int(0,5),rep.int(0.2,2))
probs2 <- c(rep.int(0.2,4),rep.int(0,5),rep.int(0.2,1))
probs3 <- c(rep.int(0.2,5),rep.int(0,5))
probs4 <- c(0,rep.int(0.2,5),rep.int(0,4))
probs5 <- c(0,0,rep.int(0.2,5),rep.int(0,3))
probs6 <- c(rep.int(0,3),rep.int(0.2,5),rep.int(0,2))
probs7 <- c(rep.int(0,4),rep.int(0.2,5),rep.int(0,1))
probs8 <- c(rep.int(0,5),rep.int(0.2,5))
probs9 <- c(0.2,rep.int(0,5),rep.int(0.2,4))
probs10 <- c(rep.int(0.2,2),rep.int(0,5),rep.int(0.2,3))

emissProbs <- matrix(c(probs1,probs2,probs3,probs4,probs5,probs6,probs7,probs8,probs9,probs10),ncol=10,n

# Initiate HMM

Robot_hmm <- initHMM(States=grid_states, Symbols=Readings, startProbs=rep.int(0.1,10), transProbs=transl

Robot_hmm

Robot_path <- simHMM(Robot_hmm,100)
Robot_path
robot_probs <- function(hmm,xs){
    # Alpha and Beta
    forward_probs <- forward(hmm = hmm, observation = xs)
    backward_probs <- backward(hmm = hmm, observation = xs)

    # Normalising the probabilities
    e_forw <- exp(forward_probs)
```

```
    filter_prob <- prop.table(e_forw,2)

    e_back <- exp(backward_probs)
    smooth_prob <- prop.table(e_forw*e_back,2)

    return(list(filter_prob = filter_prob, smooth_prob = smooth_prob))
}

path_finder <- function(line){
    # select the path with the highest probability
    m <- max(line)
    list_m <- which(line == m)

    if (length(list_m) != 1){
        # if some hav ethe same highest prob, randomly select one of them
        index <- sample(list_m,1)
    } else {
        index <- list_m
    }
    return(grid_states[index])
}


result3 <- robot_probs(hmm=Robot_hmm,xs=Robot_path$observation)
cat("Filtering probabilities:")
cat("\n")
result3$filter_prob[,1:6]
cat("Smoothing probabilities:")
cat("\n")
result3$smooth_prob[,1:6]
cat("\n")
# The most likely path calculated by the Viterbi Algorithm
viterbi_path <- viterbi(hmm = Robot_hmm, observation = Robot_path$observation)
viterbi_path
accuracy <- function(pathA, trueP){
    compare <- pathA==trueP
    t <- table(compare)
    accur <- t[2]/(sum(t))
    return(list(accuracy=accur,table=t))
}

# Path calculated by hand based on the forward probabilities
filter_path <- apply(result3$filter_prob,2,path_finder)
# Path calculated by hand based on the backward probabilities
smooth_path <- apply(result3$smooth_prob,2,path_finder)
True_path <- Robot_path$states

result_f <- accuracy(filter_path,True_path)
result_v <- accuracy(viterbi_path,True_path)
result_s <- accuracy(smooth_path,True_path)

#get the most probable path
cat("Accuracy of Filtered probability distribution:")
```

```r
cat("\n")
result_f
cat("Accuracy of Smoothed probability distribution:")
cat("\n")
result_s
cat("Accuracy of the Viterbi algorithm:")
cat("\n")
result_v

sim_probs <- function(hmm,sampleN){

    simulation <- simHMM(hmm=hmm,length=sampleN)
    result_probs <- robot_probs(hmm=hmm,xs=simulation$observation )

    # Path calculated by hand based on the forward probabilities
    filter_path2 <- apply(result_probs$filter_prob,2,path_finder)
    # Path calculated by hand based on the backward probabilities
    smooth_path2 <- apply(result_probs$smooth_prob,2,path_finder)
    True_path2 <- simulation$states
    # The most likely path calculated by the Viterbi Algorithm
    viterbi_path2 <- viterbi(hmm = hmm, observation = simulation$observation)

    result_f2 <- accuracy(filter_path2,True_path2)
    result_v2 <- accuracy(viterbi_path2,True_path2)
    result_s2 <- accuracy(smooth_path2,True_path2)

    return(list(result_f=result_f2$accuracy,result_v=result_v2$accuracy,result_s=result_s2$accuracy))
}


question5 <- lapply(rep.int(100,100),sim_probs,hmm=Robot_hmm)

# Extract inner lists for plotting
filter5 <- unname(unlist(lapply(question5, `[`, 1)))
smooth5 <- unname(unlist(lapply(question5, `[`, 3)))
viterbi5 <- unname(unlist(lapply(question5, `[`, 2)))

plot(x=1:100, y = filter5, col="blue",xlab="time",ylab="Accuracy", type="l",ylim = c(0,1))
lines(x=1:100,y=smooth5,col="red")
lines(x=1:100,y=viterbi5,col="green")
legend("bottomright",c("Filtering", "Smoothing", "Viterbi"), col = c("blue","red","green"), lty = c(1,


entropy <- apply(result3$filter_prob,2,entropy.empirical)

plot(entropy, type="l")

Z100 <- matrix(rep(result3$filter_prob[,100],10),nrow=10,ncol=10)
Z_101 <- t(transProbs) %*% Z100[,1]
most_prob_pos <- which.max(Z_101)
cat("The probability distribution of the latent variable at time t+1 is:")
cat("\n")
Z_101
```