DS8004 - Assignment 2

Najlis, Bernardo

February 21, 2017

Hidden Markov Models - Forward Algorithm Implementation

You will submit a piece of code (in language of your choice java/python/c#/R) that implements Forward Algorithm for Hidden Markov Models.

Hint: The best description of the algorithm I could find is in Section III of this paper.

Lawrence R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77 (2), p. 257–286, February 1989. 10.1109/5.18626

Deliverable:

A. A script that requires three inputs:

1) State Transition Matrix (with state names)
2) Emission Matrix (with symbol names)
3) Initial State Distribution

B. It requires a sequence of observations vector

C. It output probability of the observations vector given the HMM (specified in A)

1) Using Exhaustive search
2) Using Forward Algorithm

Resolution

Forward Algorithm

First we will create the state transition matrix, emission matrix and initial state distribution variables.

```
states <- c("rain", "sun")
states_probabilities <- c(0.4, 0.2, 0.6, 0.8)
symbols <- c("happy", "grumpy")
emission_probabilities <- c(0.4, 0.9, 0.6, 0.1)
initial_probabilities <- c(0.5, 0.5)
observations = c("happy", "grumpy", "happy")

state_transition_matrix <- matrix(states_probabilities, nrow=length(states), ncol=length(states), dimnar
emission_matrix <- matrix(emission_probabilities, nrow=length(states), ncol=length(symbols), dimnames=li
initial_state_distribution <- matrix(initial_probabilities, nrow=1, ncol=length(states), dimnames=list(
```

Let's look at the values on these input matrices:

```
state_transition_matrix
```

```
##      rain sun
## rain  0.4 0.6
## sun   0.2 0.8
```

```
emission_matrix
```

```
##      happy grumpy
## rain   0.4    0.6
## sun    0.9    0.1
```

```
initial_state_distribution
```

```
##                      rain sun
## intial probabilities  0.5 0.5
```

```
fwd_algo_alpha = array(0, c(length(states), length(observations)), dimnames = list(states, index = 1:le

for (state_num in 1:length(states)) {
  fwd_algo_alpha[state_num,1] <- initial_state_distribution[state_num] * emission_matrix[state_num,obse
}

# every next time
for (obs_num in 2:length(observations)) { # for each observation
  for (state_num in 1:length(states)) { # for each state
    temp <- 0
    for (prev_state_num in 1:length(states)){ # prior state for each current state
      temp <- temp + state_transition_matrix[prev_state_num,state_num] * fwd_algo_alpha[prev_state_num,
    }
    fwd_algo_alpha[state_num,obs_num] <- temp * emission_matrix[state_num,observations[obs_num]] # upda
  }
}
```

Building an HMM model using the "HMM" package to compare...

```
test_hmm <- initHMM(States = states,
                    Symbols = symbols,
                    startProbs = initial_probabilities,
                    transProbs = state_transition_matrix,
                    emissionProbs = emission_matrix)
test_hmm
```

```
## $States
## [1] "rain" "sun"
##
## $Symbols
## [1] "happy"  "grumpy"
##
## $startProbs
## rain  sun
##  0.5  0.5
##
## $transProbs
##       to
## from   rain sun
##   rain  0.4 0.6
##   sun   0.2 0.8
##
```

```
## $emissionProbs
##       symbols
## states happy grumpy
##   rain   0.4    0.6
##   sun    0.9    0.1
```

```r
logfwdprobs <- forward(test_hmm, observations)
```

These are the forward probabilities based on the HMM package:

```r
print(exp(logfwdprobs))
```

```
##       index
## states    1     2       3
##   rain 0.20 0.102 0.02016
##   sun  0.45 0.048 0.08964
```

These are the forward probabilities based on the implemented forward algorithm:

```r
print(fwd_algo_alpha)
```

```
##       index
##           1     2       3
##   rain 0.20 0.102 0.02016
##   sun  0.45 0.048 0.08964
```