IoT Analytics

Project 5 – Hidden Markov Model

Name - Aditya Kadole

Student ID – 200321563

The objective of this task was to solve the three problems associated with the Hidden Markov Model that is Evaluating, Decoding and Learning of the model.

Task 1: Data Set Generation

So, the task here was to generate set of observations by simulating the hidden markov model consisting of 4 states and 3 objects i.e., S = {1, 2, 3, 4} and v = {1, 2, 3} respectively. For the given set we had the initial vector pi = {1, 0, 0, 0} and we had to generate the one step transition probabilities 'p' and event matrix 'b' which also called as the emission probabilities.

So, we had to generate the set of observations using the one step probabilities p_{ij} and probabilities $b_i(k)$ with the help of pseudo random number generator using seed which was our Student ID.

Generation of p_{ij} and $b_i(k)$probabilities

So, for generating the probabilities I used random generator which produces a matrix of 4X4 and then divided the number of each row by the summation of that row. Similarly, I generated the probabilities for $b_i(k)$.

So, here we generated two matrices p_{ij} and $b_i(k)$. Then we followed the steps as shown in the pdf to generate the set of 1000 observations.

We can see the generated set of observations in the code.

P_{ij} probabilities matrix is given as

```
[[0.0494 0.3172 0.4317 0.2017]
 [0.2444 0.2085 0.1325 0.4146]
 [0.0691 0.2647 0.2867 0.3795]
 [0.303  0.2302 0.3753 0.0915]]
```

$B_i(k)$ event matrix is given as

```
[[0.1515 0.435  0.4136]
 [0.348  0.2562 0.3958]
 [0.2515 0.4804 0.2681]
 [0.7535 0.2202 0.0262]]
```

The 1000 generated observations are shown below

```
[3, 1, 1, 2, 2, 2, 1, 2, 1, 1, 1, 3, 2, 3, 2, 1, 3, 2, 1, 1, 3, 1, 2, 3, 1, 2, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 3,
3, 1, 2, 1, 1, 3, 2, 2, 2, 1, 2, 3, 1, 2, 1, 1, 3, 1, 1, 1, 1, 2, 3, 2, 1, 2, 1, 1, 1, 3, 1, 1, 2, 1, 1, 3, 1, 1,
1, 1, 2, 2, 2, 3, 3, 1, 1, 2, 1, 1, 2, 3, 1, 3, 3, 1, 2, 3, 2, 3, 2, 1, 2, 3, 3, 1, 1, 1, 2, 1, 2, 1, 2, 1, 3, 3,
2, 2, 1, 3, 2, 2, 3, 1, 2, 1, 3, 2, 2, 1, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 1, 3, 1, 1, 1, 2, 2, 2, 3, 2, 2, 1, 2, 3,
3, 1, 3, 1, 3, 1, 2, 1, 1, 1, 1, 1, 1, 3, 1, 2, 3, 1, 2, 1, 3, 2, 3, 2, 3, 3, 1, 3, 1, 2, 1, 2, 1, 2, 1, 3, 1, 2,
1, 1, 2, 1, 2, 1, 2, 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 3, 2, 2, 1, 2, 3, 1, 1, 3, 1, 2, 1, 1, 1, 3, 1, 3, 1,
1, 1, 2, 1, 1, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 3, 1, 2, 2, 2, 1, 2, 1, 3, 1, 1, 2, 1, 3, 1, 3, 1, 2, 1, 3, 1, 2,
1, 1, 1, 1, 1, 3, 3, 1, 3, 1, 3, 3, 1, 1, 1, 3, 2, 3, 1, 1, 1, 2, 1, 1, 3, 1, 2, 1, 2, 3, 2, 1, 2, 1, 2, 1, 3, 1,
2, 1, 1, 1, 3, 2, 2, 2, 2, 3, 2, 3, 1, 2, 1, 3, 2, 2, 1, 1, 1, 1, 3, 1, 3, 2, 1, 2, 2, 1, 1, 3, 3, 1, 1, 2, 1, 3,
3, 1, 1, 2, 3, 1, 1, 3, 1, 3, 2, 2, 2, 2, 1, 1, 2, 2, 3, 3, 1, 1, 1, 1, 3, 1, 3, 1, 2, 2, 1, 3, 1, 3, 3, 3, 2, 3,
3, 1, 3, 1, 2, 1, 2, 1, 1, 1, 1, 3, 1, 2, 1, 2, 1, 2, 2, 2, 2, 1, 1, 3, 1, 1, 1, 3, 1, 2, 1, 3, 2, 2, 3, 1, 2, 1,
2, 1, 1, 3, 1, 2, 2, 3, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1, 3, 3, 3, 1, 2, 3, 1, 2, 1, 2, 1, 2, 3, 3, 1,
1, 1, 2, 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 1, 2, 1, 1, 1, 1, 3, 1, 2, 3, 1, 1,
1, 2, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 2, 2, 2, 1, 3, 2, 2, 1, 3, 2, 1, 1, 2, 1, 2, 2,
3, 2, 1, 3, 1, 1, 3, 1, 3, 2, 1, 1, 2, 3, 1, 1, 2, 1, 2, 1, 3, 1, 2, 1, 1, 2, 2, 2, 3, 1, 1, 3, 1, 2, 1, 2, 1, 3,
2, 1, 3, 1, 1, 3, 1, 1, 3, 1, 2, 2, 1, 3, 1, 3, 2, 2, 1, 2, 1, 2, 1, 3, 2, 1, 1, 1, 3, 1, 2, 1, 2, 1, 1, 1, 2, 2,
2, 1, 2, 2, 3, 2, 1, 1, 1, 1, 1, 1, 2, 3, 2, 2, 2, 1, 2, 1, 1, 3, 3, 2, 1, 2, 1, 1, 3, 2, 1, 1, 1, 2, 1, 3, 1, 1,
1, 3, 1, 1, 2, 2, 3, 1, 1, 1, 2, 1, 3, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 3, 1, 2, 3, 1, 3, 1, 3, 2, 1, 2, 1, 3,
3, 2, 3, 3, 1, 2, 1, 2, 1, 2, 3, 1, 2, 1, 2, 2, 1, 1, 3, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 3, 1, 1, 1, 2, 2, 1, 2, 1,
1, 2, 1, 3, 2, 1, 3, 1, 1, 3, 1, 1, 3, 2, 1, 1, 3, 1, 2, 1, 2, 1, 3, 2, 1, 2, 1, 2, 2, 2, 1, 3, 1, 3, 1, 1, 1, 3,
1, 3, 1, 2, 2, 1, 2, 2, 1, 1, 2, 1, 3, 1, 3, 1, 2, 3, 1, 1, 3, 2, 2, 1, 2, 1, 1, 2, 2, 1, 1, 2, 3, 1, 1, 1, 1, 2,
1, 2, 2, 3, 1, 3, 1, 3, 1, 1, 1, 1, 1, 1, 2, 1, 1, 3, 1, 3, 1, 2, 2, 2, 3, 2, 3, 2, 2, 1, 1, 3, 1, 1, 2, 2, 1, 2,
1, 2, 1, 2, 1, 2, 1, 2, 1, 3, 2, 3, 1, 3, 2, 2, 1, 2, 1, 2, 1, 2, 3, 1, 3, 1, 3, 1, 1, 2, 1, 2, 1, 2, 1, 3, 1, 3,
2, 3, 1, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 1, 3, 1, 1, 2, 2, 2, 1, 2, 3, 1, 1, 1, 3, 3, 3, 3, 1, 2, 1, 1, 3, 2, 3, 1,
1, 3, 1, 1, 1, 2, 3, 1, 2, 1, 1, 2, 2, 3, 1, 2, 1, 3, 1, 1, 3, 3, 1, 1, 3, 1, 2, 2, 1, 1, 2, 1, 2, 2, 1, 3, 1, 2,
1, 3, 1, 3, 1, 2, 2, 2, 1, 1, 3, 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 1, 2, 3, 1, 2, 1, 3, 2, 1, 3, 3, 3, 1, 3, 2, 3, 2,
1, 1, 3, 3, 3, 2, 1, 3, 3, 1, 2, 1, 3]
```

Task 2: Estimate p (O / λ)

In this task we had to calculate the probability that the sequence of observations came from the hidden markov model

Given set of observation O – 123312331233

So, we made use of the forward and backward algorithm to calculate the probability. We calculated the likelihood that is the probability using the forward algorithm and the later compared the obtained probability with the backward algorithm by multiplying the first row and first column of alpha matrix with first row and first column of beta matrix which can be seen below

Since, there are only 4 states the probability of these observations in the hidden markov model is less and as the length or the size of the observation increases the probability keeps on decreasing

Here, we obtained probability of the observation directly using the forward algorithm. And the backward algorithm output is required for training the HMM.

The probability that the above sequence of observation came from HMM: **2.574717509143524e-07**

```
The probability that the above sequence of observation came from HMM:
2.574717509143524e-07
```

Though we can verify the probability using the alpha and beta matrix

Alpha Matrix

```
[[1.51500000e-01 0.00000000e+00 0.00000000e+00 0.00000000e+00]
 [3.25558350e-03 1.23118960e-02 3.14193850e-02 6.72877251e-03]
 [3.05226492e-03 5.32959584e-03 3.90621904e-03 4.79473167e-04]
 [7.72826165e-04 1.27596028e-03 8.91082226e-04 1.14011244e-04]
 [6.75904362e-05 2.69105975e-04 2.01440305e-04 7.78734641e-04]
 [1.38758296e-04 7.94563936e-05 1.99292681e-04 6.00937126e-05]
 [2.40935738e-05 5.03327520e-05 4.02472553e-05 3.72198029e-06]
 [7.19680234e-06 1.17343256e-05 8.04461738e-06 1.08316112e-06]
 [6.22282023e-07 2.47364890e-06 1.85470181e-06 7.13465639e-06]
 [1.27248867e-06 7.29268464e-07 1.82829536e-06 5.52210848e-07]
 [2.21172289e-07 4.61801470e-07 3.69275160e-07 3.41485835e-08]
 [6.60329460e-08 1.07677230e-07 7.38229232e-08 9.93865190e-09]]
```

Beta Matrix

```
[[1.69948350e-06 1.68687716e-06 1.63482421e-06 1.88057250e-06]
 [5.14524215e-06 4.87039639e-06 4.33703708e-06 6.61192422e-06]
 [2.35335867e-05 1.85322896e-05 1.89893415e-05 2.64767036e-05]
 [7.21695051e-05 9.01564360e-05 9.03223896e-05 5.41750263e-05]
 [1.85207303e-04 1.83836959e-04 1.78164293e-04 2.04938122e-04]
 [5.60698881e-04 5.30793907e-04 4.72622905e-04 7.20622379e-04]
 [2.56532401e-03 2.01903685e-03 2.06978415e-03 2.88478565e-03]
 [7.85473341e-03 9.83162109e-03 9.84426049e-03 5.89204560e-03]
 [2.01576232e-02 1.99559695e-02 1.93321569e-02 2.23849008e-02]
 [6.14992139e-02 5.72604169e-02 5.18252602e-02 7.73338652e-02]
 [2.67002910e-01 2.29993910e-01 2.20155190e-01 3.19449190e-01]
 [1.00000000e+00 1.00000000e+00 1.00000000e+00 1.00000000e+00]]
```

So, by multiplying $\alpha[1][1] * \beta[1][1]$ we get = 1.515e-01 X 1.69948e-06 = **2.574717e-07** which is equal to the probability that we obtained using forward algorithm.

From the probability we can say that the value is too low as the number of observations increases.

Task 3: Estimate the most probable sequence Q

In this task we are going to obtain the most probable sequence Q using the given set of observations which is O – 123312331233

So, for this task we made use of the Viterbi algorithm. It uses dynamic programming and obtains the best sequence for the given set of observations.

The sequence obtained is – [1, 3, 2, 2, 4, 1, 3, 2, 4, 3, 2, 1]

```
 The most probable sequence for given observation:
[1. 3. 2. 2. 4. 1. 3. 2. 4. 3. 2. 1.]
```

The optimum path is as shown above which is different to the one used initially. Hence, this can be chosen as the optimum path.

Task 4: Train HMM

In this task we don't know the parameters of HMM but we will calculate the parameters using the 1000 generated observations assuming that there are 4 states and 3 objects.

The states are given by S = {1, 2, 3, 4}and the objects are v = {1, 2, 3}

We made use of the 1000 observations that we generated to get the alpha and beta values. By this the probability was too low in the range 10^-100 above so we had to discard certain observations and hence we considered only 12 observations of the 1000 observations generated and the obtained matrices for P_{ij} and $B_i(k)$ is given below and is compared with the original matrices which were used in the generation of observations in task 1.

Original P_{ij}	Estimated P_{ij}
[[0.0494 0.3172 0.4317 0.2017] [0.2444 0.2085 0.1325 0.4146] [0.0691 0.2647 0.2867 0.3795] [0.303 0.2302 0.3753 0.0915]]	[[0.06472255 0.35360359 0.4437647 0.13790915] [0.31397927 0.27408712 0.13125866 0.28067495] [0.11011892 0.38796011 0.30609893 0.19582203] [0.37704213 0.19379013 0.3744699 0.05469784]]
Original $B_i(k)$	**Estimated $B_i(k)$**
[[0.1515 0.435 0.4136] [0.348 0.2562 0.3958] [0.2515 0.4804 0.2681] [0.7535 0.2202 0.0262]]	[[0.37158924 0.22727952 0.40113124] [0.12675451 0.18454462 0.68870087] [0.09019864 0.40885756 0.50094381] [0.64284487 0.2627333 0.09442184]]