Project 5

Task1

- The instructions were followed to generate P and B.
- Using which Q and O were generated to be used in following tasks.

Task2

- Estimate p(O|λ)
- To estimate p(O|λ), I used the Forward-Backward Algorithm.
- According to this algorithm, given that we know the parameters of an HMM, P, π and B, and given that we have observed a sequence of observations O, calculate the probability p (O | λ), i.e, calculate the probability that this sequence came from the given HMM.
- The given sequence O = 123312331233. We calculate the probability that this sequence was generated by the HMM λ =P, π, B
- To develop the algorithm, I followed the algorithm in the chapter notes which makes it very easy to translate it to code.
- The algorithm
  - The forward algorithm generates alpha over t.
  - The backward algorithm generates beta over t.
  - We check if the sum(i,N) alpha_t(i)*beta_t(i) remains the same for every t.
- The probability p(O|λ) obtained for the given sequence is 1.54184678e-06
- Since this is a very low probability, this indicates that it is not likely that the given sequence may have been generated from this HMM

Task3

- For this task, I used the Viterbi Algorithm.
- Viterbi is a kind of dynamic programming algorithm that makes use of a dynamic programming. The viterbi algorithm offer an efficient way of finding the single best state sequence
- The algorithm finds the best sequence of states Q, for a given sequence of observations O, given that we know the parameters λ = (P, B, π) of an HMM.
- I used the algorithm for the given sequence of O = 123312331233.
- I followed the algorithm in the chapter notes as well as the example which made it easy to translate it to code.

- I follow the same steps as in Task1 to generate P,B and use that data to generate O.
- My algorithm
    - Initialize an array psi of length of the number of states and initialized to 0.
    - Calculate psi and delta over T,N
    - To calculate path, backtrack using the values of psi.
- I get the most probable path to be [1, 2, 3, 4, 2, 1, 3, 4, 2, 1, 3, 3]
- We might get other paths using Solution in the chapter notes, but viterbi is an optimization algorithm and hence gives the most optimal path.
- We see that the first state is 1 as according to initial probabilities, only state 1 has probability =1, while the rest are 0. Therefore the first state has to be 1, and the obtained path also has the first state as 1.
-

Task4

- I use a multinomial HMM from the hmmlearn package.
- Multinomial HMM is suitable as it takes in discrete input like our data.
- I follow the same steps as in Task1 to generate P,B and use that data to generate O.
- I fit the model for 40 iterations.
- The probabilities obtained from the model are as follows -
    - Initial probabilities
      [4.58257420e-07 1.01291760e-03 9.98986624e-01 8.24953121e-11]

    - Transition probabilities
      [[0.24522132 0.23822122 0.20823566 0.3083218]
       [0.22733326 0.24203643 0.20820936 0.32242095]
       [0.21247813 0.23312974 0.17369796 0.38069417]
       [0.24745146 0.24479664 0.24272784 0.26502406]]

    - Event matrix
      [[0.27094995 0.34412936 0.38492069]
       [0.18360031 0.50603481 0.31036488]
       [0.21048538 0.70731501 0.08219961]
       [0.12041965 0.22558437 0.65399598]]

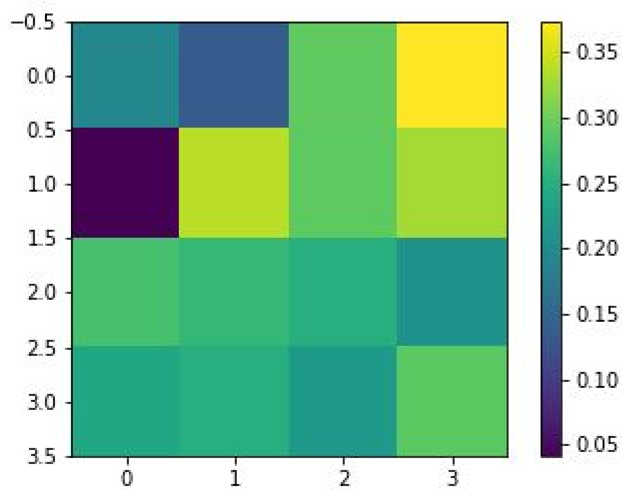
- The original probabilities are -
    - Initial probabilities
      [1,0,0,0]

    - Transition probabilities
      [[0.19547493, 0.13757054, 0.29319991, 0.37375462],

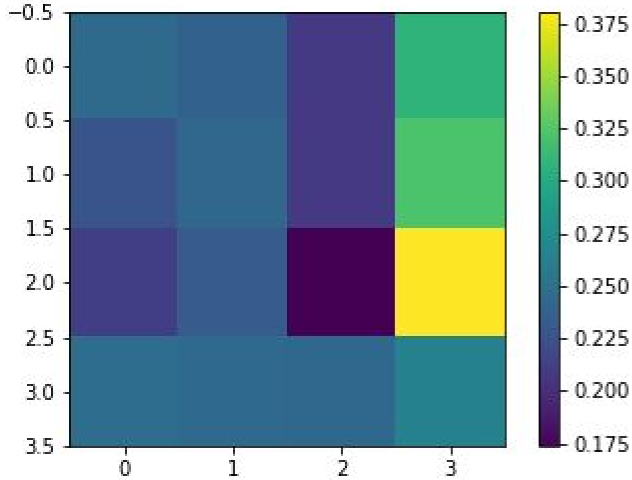
[0.04133207, 0.33883134, 0.29073149, 0.3291051],
[0.27688667, 0.26228274, 0.25067505, 0.21015553],
[0.23890892, 0.24938778, 0.22122013, 0.29048318]]

- Event matrix
  [[0.14493102, 0.28696169, 0.56810729],
  [0.0595772 , 0.5480175 , 0.39240531],
  [0.17631078, 0.42892038, 0.39476884],
  [0.33184461, 0.30245334, 0.36570205]]

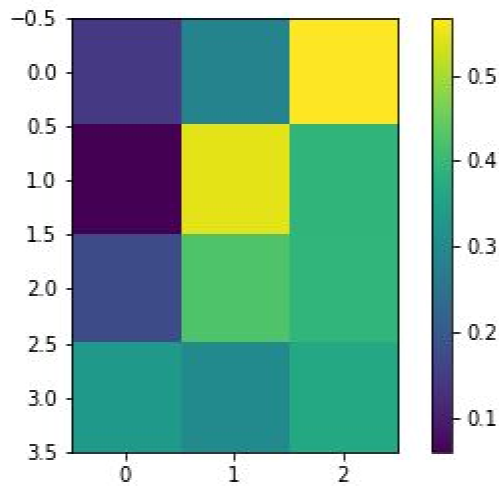
- I observed that the probabilities changed rapidly with change in number of iterations and for each run of the program, which is why I set the random state.
- I picked one set of probabilities for comparison with the generated data.
- Comparing transition probabilities, we see that there is a major difference between the original and the model generated values.
- This could be for various reasons. The error might be due to the randomness involved in generating our initial data.
- It might also be due to improper hyperparameter tuning of the HMM solver.
- The same applies to the event matrix as we as pi.
- The x axis indicate columns and row 0 starts from the top left corner and the color bar indicating the matrix values, are on the right.
  - P



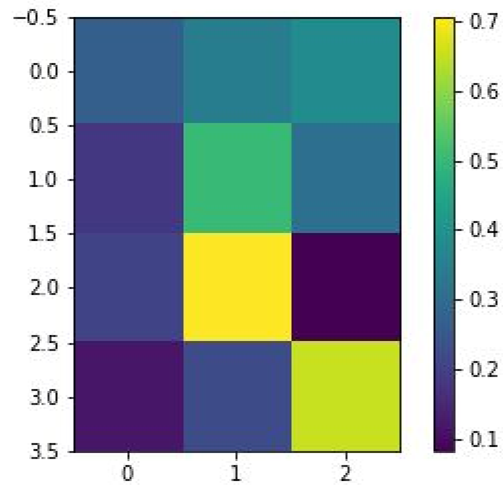
  - Model generated transition values



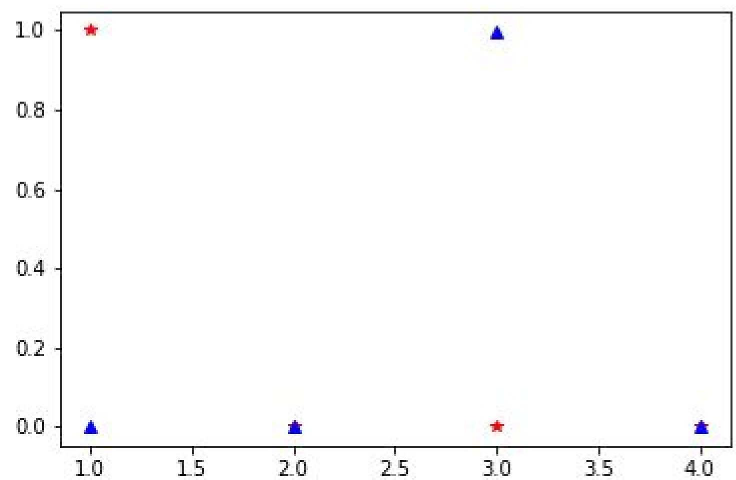
-
- B



- Model generated event matrix



- ○
- ○ Pi



- ○
- ○ The red stars indicate the given values whereas the blue triangles indicate the the model generated values. All of these values vary a lot with respect to n-iterations as well as random state.

- The p-values were not available in this package.