Hidden Markov Models

Rodrigo Escobar
raeb1g08
22841997

Task 1

The first part of this coursework was about generation a sequence of numbers based on die rolls, where one of the dices was common and the second one was a loaded one. Using the Hidden Markov models as example, the recreation of such sequence based on probability was done.
The excersice requested the sequence for each two 2 different models with different probability, one favouring the even numbers in the loaded dice and the other favouring the odd numbers[Figure 1].

In both cases the probability was a random number that decided the first state of each one of the rolls. Another random number for the probability decided if the number was even or odd in the case of the loaded dice. If the probability favoured the even, the even number to be rolled by the dice was selected randomly and "fair" among the even numbers.
Both sequences contained numbers from 1 to 6 on them. The sequence in the matrix `my_roll_one` was generated by the first model, which favours even probabilities. While the one in `my_roll_two` was generated by the second model.

Task 2

The second part of this coursework demanded to use the Forward algorithm to calculate the probability that the sequences of each one of the two rolls previously created (`my_roll_one` and `my_roll_two`) plus three more rolls provided in the specifications (`roll_one`, `roll_two` and `roll_three`) were generated by each one of the models.
The forward algorithm takes in consideration all the probabilities that the value in the sequence is taken from the state being consider.
For the first value of the sequence it was implied that there were 50% chance of stating with each role. Therefore the probability for this value was $0.5 * probability\_for\_the\_value\_in\_the\_state$. The next values take in consideration the probabilities in the values before them and it builts up until reaching the end. In this specific case the probabilities in each state were got by following the formula:

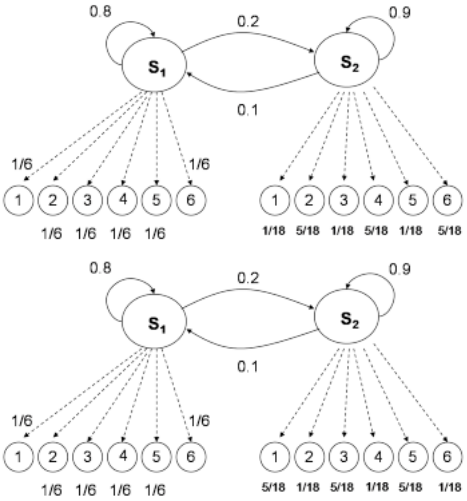

Figure 1: Die models

	1	2	3	4	5	6	7	8	9	10
1	1.2134e-71	3.9289e-86	2.3029e-77	2.1308e-86	6.4621e-86	3.9342e-86	4.8577e-71	1.9314e-83	5.9500e-72	5.9033e-71

Figure 2: Task two results

$(probability\_for\_the\_past\_value\_state\_one * transition * probability\_for\_the\_value\_in\_the\_state\_two) +$

$(probability\_for\_the\_past\_state * transition * probability\_for\_the\_value\_in\_the\_state)$
At the end of the sequence the probabilities from both states were added and this gave as result the probability used for the model. This algorithm was used on both model for each roll sequence, giving as a result 10 values which are stored in the task_two matrix [Figure 2]. A graphical representation for these values is shown in Figure 3.

In the results, it can be seen that `my_roll_one` and `roll_one` are more likely to be generated by the first model while `my_roll_two`, `roll_two`, `roll_three` are more likely to be generated by the second model.

Task 3

For task 3 of this coursework it was needed to get the most probable state for each value on each model for every roll. This is five rolls, two states and a hundred values.

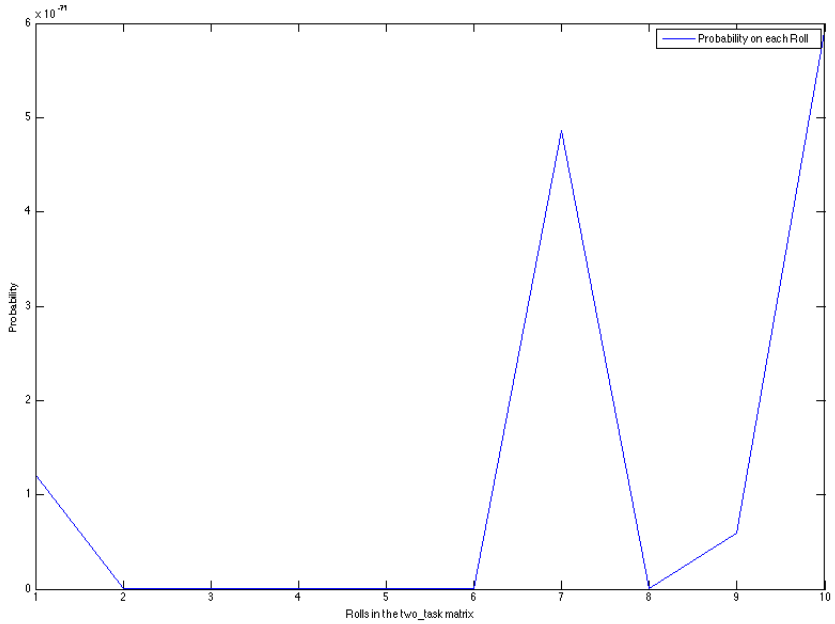

Figure 3: Probabilities for each roll with each model

The backwards algorithm was used to compliment the results of the forward algorithm for each state and make the results accuarate. The backwards algorithm works very similar to the forward algorithm which the exception that it starts at the end of the sequence with stating probability of 100% instead of 50% and it finishes at the start of the sequence. However the probabilities are not taken from the past values in the sequence as in the forward algorithm but rather in from the next in line (if we consider the next as we would in the forward algorithm)

The results favour the fair dice of each model in the most of the sequences for each model, excepting the one in `my_roll_one` and `roll_one` for the even model, which values turned out to be more likely to be generated by the unfair dice .