Applied Machine Learning (874)

Post-block Assignment 2
Department of Industrial Engineering
Stellenbosch University
24 May 2021

Deadline: 6 June 2021, 23:59
Total: [157]

1 Instructions

1. The focus of this assignment is to test your understanding of the concepts covered in topics 4 and 6.
2. Answer all of the questions below.
3. Where asked to provide all calculations, please make sure that you do. Without these calculations, no marks will be given.
4. Submit your typed answers as a pdf document. Please name this pdf document ???????PBA2.pdf, where you replace the question marks with your student number. Please note that all documents submitted must be pdf. No other formats will be accepted.
5. Please make sure that you do and submit your own work. Plagiarism will not be tolerated.
6. Note that late submissions can not be accepted and that no extensions to the deadline can be provided.

2 Reinforcement Learning [78]:

Complete each of the following assignments:

1. Given the maze in Figure 1 below, implement a Q-learning RL agent to solve the maze. Train the agent for 10 000 episodes using an ϵ-greedy approach. The agent can move left, right, up or down by one block at a time. The reward function is as indicated in the maze. Set $\epsilon = 0.1, \eta = 0.1$, and $\gamma = 0.95$.

   Based on your implementation, answer the following questions:

   (a) How many state-action pairs are there in your Q-table. Show your calculations. (5)

   (b) Plot the training progression in respect of the number of steps per training episode for the duration of the 10 000 training episodes. (10)

   (c) Test your agent employing a greedy policy. Plot the path that the agent has followed. Also complete a table such as the one below for the path followed, stating the state as well as the corresponding state action values. (4)

State	$s_1 = (1, 1)$	$s_2 = (y, x)$	...	$s_T = (6, 12)$
a_1	$Q(s_1, a_1)$	$Q(s_2, a_1)$	...	
a_2	$Q(s_1, a_2)$	$Q(s_2, a_2)$	...	
a_3	$Q(s_1, a_3)$	$Q(s_2, a_3)$	...	
a_4	$Q(s_1, a_4)$	$Q(s_2, a_4)$	...	

   (d) In order to incentivise your agent to move towards the target state faster, implement, as part of your reward function, a punishment of -0.05 that the agent receives every time it enters a state that is not the target state. On the same set of axes, plot the training progression as in (b) above, together with the training progression corresponding to the updated reward function. Comment on the difference (or the lack thereof) in respect of these two training progressions.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	A														
2			X	X	X										
3										X	X	X	X		
4															
5		X	X	X	X	X	X	X	X						
6															
7	X										X			X	
8	X										X				
9											X				
10											X			X	
11		X	X	X	X	X	X				X			X	
12						F								X	
13														X	
14														X	

Starting point = A

Target state = F

$r(s' = F) = 1$

$r(s' \neq F) = 0$

The agent cannot enter blocks marked with X

Figure 1: The maze that the reinforcement learning agent must traverse. The agent cannot enter states marked with X. If the agent attempts to enter these states, the resulting new state is simply the same state that the agent was in before the attempted move (*i.e.* $s' = s$).

State	$s_1 = (1, 1)$	$s_2 = (y, x)$	$\dots$	$s_T = (6, 12)$
a_1	$Q(s_1, a_1)$	$Q(s_2, a_1)$	$\dots$	
a_2	$Q(s_1, a_2)$	$Q(s_2, a_2)$	$\dots$	
a_3	$Q(s_1, a_3)$	$Q(s_2, a_3)$	$\dots$	
a_4	$Q(s_1, a_4)$	$Q(s_2, a_4)$	$\dots$	

(d) cont. Also complete a table such as the one [redacted] for the path followed, stating the state as well as the corresponding state action values. (6)

(e) Change your reward function such that the agent receives a punishment of -1 for every state it enters which is marked by an X. Maintain the punishment of -0.05 whenever the agent enters a state which is not the target state. In this case the agent the episode is also terminated once the agent has entered a state marked by an X. Retrain the agent for 10 000 episodes. On the same set of axes, plot the training progression of the agent as in (b) and (d) above. Is there a noticeable difference between the progressions? (5)

(f) Test the agent of (d) above employing a greedy policy. Plot the path that the agent has followed. Also complete a table such as the one below for the path followed, stating the state as well as the corresponding state action values. Did the agent take a shorter path to the goal state than in (c)? (6)

State	$s_1 = (1, 1)$	$s_2 = (y, x)$	$\dots$	$s_T = (6, 12)$
a_1	$Q(s_1, a_1)$	$Q(s_2, a_1)$	$\dots$	
a_2	$Q(s_1, a_2)$	$Q(s_2, a_2)$	$\dots$	
a_3	$Q(s_1, a_3)$	$Q(s_2, a_3)$	$\dots$	
a_4	$Q(s_1, a_4)$	$Q(s_2, a_4)$	$\dots$	

2. Train a SARSA agent to solve the maze in Figure 1. Use the original reward function as in 1. (a)

above. Employ the ϵ-greedy method during training for 10 000 episodes. Set $\epsilon = 0.1, \eta = 0.1$, and $\gamma = 0.95$. Answer the questions below:

(a) Plot the training progression of the reinforcement learning agent over the 10 000 episodes in respect of the number of time steps per episode. (5)

(b) Test your agent employing a greedy policy. Plot the path that the agent has followed. Also complete a table such as the one below for the path followed, stating the state as well as the corresponding state action values. (4)

State	$s_1 = (1, 1)$	$s_2 = (y, x)$	...	$s_T = (6, 12)$
a_1	$Q(s_1, a_1)$	$Q(s_2, a_1)$	...	
a_2	$Q(s_1, a_2)$	$Q(s_2, a_2)$	...	
a_3	$Q(s_1, a_3)$	$Q(s_2, a_3)$	...	
a_4	$Q(s_1, a_4)$	$Q(s_2, a_4)$	...	

3. Comment on the effectiveness of the Q-learning and SARSA agent when solving the maze. Focus particularly on the time until convergence during training, and the final routes followed during testing. What are key differences between these algorithms. Are these differences reflected in the results achieved? (5)

4. Given the typically long episodes at the start of the training procedure. Suggest a method that may be employed in order to improve the speed at which the agent learns, in which there is no change to the reward function used in the training procedure. Clearly state all steps in this procedure. The resulting approach should still be table-based. (5)

5. Suppose the maze were significantly larger and function approximation has to be performed. Define the type and structure of an artificial neural network that you would use in conjunction with deep Q-learning. Specify clearly what the inputs and outputs of the network would be. (4)

6. Given the maze in Figure 2 below. Train a Q-learning reinforcement learning agent to solve the maze, while evading the opponent. The goal of the opponent is to catch the agent (not to also reach the target state). Assume that the opponent knows where the agent is located, moves greedily towards the agent. Assume that the episode is also terminated when the opponent and the agent are located in the same block (*i.e.* the opponent has caught the agent). The movement of the opponent is governed as follows.

- Before the agent takes its step, determine the x- and y differences between the agent and the opponent.
- Determine the Manhattan distance between the agent and the opponent.
- Take the allowable (based on the blocks marked X in the maze) step that results in the smallest resulting Manhattan distance between the agent and the opponent. Note that the Manhattan distance does not have to take into account the path restrictions of all the blocks marked X.
- Once the opponent has moved, the agent can determine the Manhattan distance between itself and the opponent, and then take its own next step.

Train the agent for 10 000 episodes using an ϵ-greedy approach. Set $\epsilon = 0.1, \eta = 0.1$, and $\gamma = 0.95$. Answer the questions below:

(a) Describe in detail how you incorporated the information on the movement of the opponent in addition to the agents location into your implementation of the Q-table. (4)

(b) Plot the training progression of the reinforcement learning agent over the 10 000 episodes in respect of the number of time steps per episode. Indicate on the plot episodes when the agent was caught by the opponent. (10)

(c) Test your agent using a greedy policy. Plot the path that both the agent and the opponent have followed. (5)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	A														
2			X	X	X										
3										X	X	X	X		
4															
5		X	X	X	X	X	X	X	X						
6															
7	X										X			X	
8	X										X				
9											X				O
10											X			X	
11		X	X	X	X	X	X				X			X	
12						F								X	
13														X	
14														X	

Starting point = A
Target state = F
$r(s' = F) = 1$
$r(s' \neq F) = -0.01$

The agent cannot enter blocks marked with X
Opponent = O

Figure 2: The maze that the reinforcement learning agent must traverse. The agent cannot enter states marked with X. If the agent attempts to enter these states, the resulting new state is simply the same state that the agent was in before the attempted move (*i.e.* $s' = s$). The opponent starts at the location marked O.

3 Probability-based learning [79]:

Complete each of the following assignments:

1. Consider you have been given a dataset of 1 500 news articles which on which sentiment analysis is performed, which have been classified as positive or negative. There are 1 100 negative articles in the dataset, and 400 positive articles in the dataset. The tables below give the number of documents from each sentiment class which contain a selection of words. Answer the questions below and show all your working clearly.

Words contained in positive articles						
danger	is	getaway	lawyer	critic	crazy	powerful
235	612	12	135	89	180	375
Words contained in negative articles						
danger	is	getaway	lawyer	critic	crazy	powerful
412	637	122	48	102	99	357

   (a) What target level will a Naive Bayes model predict for the following document: "danger is powerful"? (4)

   (b) What target level will a Naive Bayes model predict for the following document: "powerful lawyer is crazy"? (6)

2. Consider the "SMSSpamCollection.txt" dataset. This dataset contains 5 574 SMS messages which have been classified as *spam* or *ham* (*i.e.* not spam). Build a Naïve Bayes classifier that is capable of classifying the SMS messages as spam or not. In order to generate the prior probabilities and likelihoods employ the same logic as was used in exercise 6 of Section 6.7 of the prescribed textbook (*i.e.* generate word counts and associated classes). Then implement the Naïve Bayes classifier. You do not have to remove any special characters or reduce the total number of words.

Hint: In order to perform the word counts in Python consider the `CountVectorizer` function from the Scikit-learn library. If you are working in R, consider the `CountVectorizer` function from the superml v0.5.3 package.

Detail the process followed, from importing and preparing the data, through the building of the Naïve Bayes model in your final answer. Finally, assess the performance of the classifier on the training set, and provide the resulting confusion matrix in your final answer. (10)

3. The dataset "GaussianMix.csv" contains values drawn from a probability distribution that is a mixture of four Gaussian distributions. Use Expectation-Maximisation in order to determine the parameters π_k, μ_k, and σ_k of the underlying $k = 4$ distributions. Use the following initial values

k	$\pi_k(0)$	$\mu_k(0)$	$\sigma_k(0)$
1	1/4	4	1
2	1/4	5	1
3	1/4	6	1
4	1/4	7	1

In your final answer, plot the mixture of Gaussians fitted to the data, and a histogram of the raw data on the same set of axes. Also state the values for all fitted parameters clearly. You may assume that convergence has been achieved when the adjustments in the parameter values are smaller than 0.001. (16)

4. The dataset "Cluster.csv" contains coordinate pairs that need to be grouped together into three separate clusters. Use expectation-maximisation in order to perform the clustering. Each coordinate pair has randomly been assigned to one of the three clusters (Column Cluster(0)). Use these random assignments in order to calculate the initial likelihood and prior probabilities. State all assumptions and any parameter settings clearly.

   Plot the initial cluster allocation as provided. On a separate set of axes plot the final clustering allocation as learnt during the expectation-maximisation process described on slides 191–195. (14)

5. These days, smokers are increasingly rare. While more common than smoking, getting the common cold is also rare, especially given the increased level of protection based on mandatory mask wearing. Lung disease is also rare, but smokers have a higher probability of getting lung disease. A cough is often experience by patients with either lung disease or the common cold. Chest pain and shortness of breath are common symptoms of lung disease, while people with the common cold often experience fever. Answer the following questions based strictly on the information given above.

   (a) Define the topology of a Bayesian network that encodes the causal relationships described above. (7)

   (b) The table below lists a set of instances from the house alarm domain. Using the data in this table, create the conditional probability tables (CPTs) for the network you created in part (a) of this question. (14)

   (c) What value will the Bayesian network predict for Cough given that the patient has Lung Disease, but does not have a cold and is not a smoker. (8)

ID	Smokes	Cold	Lung Disease	Fever	Shortness of Breath	Chest Pain	Cough
1	True	False	True	True	True	False	True
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	True
4	False	False	False	False	False	False	False
5	True	False	False	False	False	False	True
6	False	False	False	False	False	False	False
7	False	True	False	False	False	False	True
8	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False
10	False	False	False	True	False	False	True
11	False	False	False	False	False	False	True
12	False	False	False	False	False	False	False
13	True	False	False	False	False	False	False
14	True	False	False	False	False	False	False
15	False	False	False	False	False	False	True
16	False	False	False	True	False	False	True
17	False	False	True	False	False	True	True
18	False	True	False	True	True	False	False
19	False	False	False	False	False	False	True
20	False	False	False	True	False	False	False