CMPT 310 - Artificial Intelligence Survey

Assignment 4

Due date: April 12, 2021 J. Classen
10 marks March 30, 2021

Important Note: Students must work individually on this, and other CMPT 310, assignments. You may not discuss the specific questions in this assignment, nor their solutions with any other student. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the general concepts involved in the questions in the context of completely different problems. If you are in doubt as to what constitutes acceptable discussion, please ask!

Question 1 (4 marks)

The table on the right contains (fictitious) examples of holiday trips. Relevant attributes are the destination country of the trip, the season during which the trip took place, the type of trip, and its length in weeks. The target attribute is how much fun the trip was.

example	attributes				goal
	country	season	type	weeks	fun
1	Italy	summer	repose	2	much
2	Italy	winter	sports	1	much
3	Austria	winter	culture	1	little
4	Austria	winter	repose	3	little
5	Austria	winter	sports	1	much
6	Spain	summer	repose	3	much
7	Spain	summer	sports	2	much
8	Spain	winter	repose	2	little

(You may assume that all possible values of the attributes are already mentioned in the examples.)

(a) Generate a decision tree from these examples using the Decision-Tree-Learning algorithm in order to predict the expected fun for arbitrary holiday trips. Determine the best attribute for each test by means of computing information gains. Please show and explain all steps.

(b) From the examples, find two decision lists (DL) predicting the expected fun. The first list should comply with (i), the second with (ii).

(i) The tests of the DL contain as few literals as possible (e.g. only one, if possible).

(ii) The DL consists of as few tests as possible (e.g. only one, if possible).

Question 2 (3 marks)

The table on the right contains (again fictitious) examples of observations about movies. Attributes represent the genre of the movie, the number of celebrity main actors, the amount of marketing that was done, the production costs, and whether the movie was well received by critics or not.

example	attributes				goal
	genre	actors	marketing	cost	reception
1	SciFi	2	much	high	good
2	Comedy	3	little	low	bad
3	Comedy	1	little	medium	bad
4	Drama	2	much	low	good
5	Drama	1	little	high	good
6	SciFi	3	much	low	good
7	Comedy	1	little	high	good
8	SciFi	2	little	low	bad

In this question, a perceptron shall be trained using the examples in the table. To this end, in part (a), the attributes have to be transformed into numeric inputs first. Then, in part (b), Neural-Network-Learning can be applied.

(a) For encoding the examples use the eight inputs I_{D}, I_{C}, I_{S}, $I_{\#}$, I_{M}, I_{h}, I_{m}, I_{l}, where the attributes “genre”, “marketing”, “cost” are encoded by the following schemes:

genre	I_{D}	I_{C}	I_{S}
Drama	1	0	0
Comedy	0	1	0
SciFi	0	0	1

marketing	I_{M}
much	1
little	0

cost	I_{h}	I_{m}	I_{l}
high	1	0	0
medium	0	1	0
low	0	0	1

The values of the attribute “actors” can directly serve as input $I_{\#}$ since they are numeric. For the goal attribute “reception”, 1 represents “good” and 0 represents “bad”.

Set up the following table of transformed examples where T denotes the correct output.

example	I_{D}	I_{C}	I_{S}	$I_{\#}$	I_{M}	I_{h}	I_{m}	I_{l}	T
1	0	0	1	2	1	1	0	0	1
2									
⋮									
8									

For which of the attributes “genre”, “actors”, “marketing”, “cost”, and “reception” did we use local encoding, and for which ones or distributed encoding?

(b) Use the transformed examples of part (a) to train a perceptron that has eight inputs, namely I_{D}, I_{C}, I_{S}, $I_{\#}$, I_{M}, I_{h}, I_{m}, I_{l} with corresponding weights W_{D}, W_{C}, W_{S}, $W_{\#}$, W_{M}, W_{h}, W_{m}, W_{l}. Let the activation function be $step_0$, the learning rate be 2, and the weights initialized by +1. (For the given examples, these settings yield a fast convergence.)

As a trace of the Neural-Network-Learning algorithm applied to the examples of part (a), set up the following table where O denotes the perceptron output and $E = T - O$ is the error. You must include the output and the error for every example but you may omit weights whose values are not changed. In the table below, the first half of epoch 1 is already entered.

	example	O	E	W_{D}	W_{C}	W_{S}	$W_{\#}$	W_{M}	W_{h}	W_{m}	W_{l}
initial.				+1	+1	+1	+1	+1	+1	+1	+1
epoch 1	1	1	0								
	2	1	−1		−1		−5				−1
	3	0	0								
	4	0	+1	+3			−1	+3			+1
	5										
	6										
	7										
	8										
epoch 2	1										
	⋮										
	8										
epoch 3	⋮										

[Hint: If you did not make a mistake — neither here nor in part (a) — it should turn out that all examples are correctly predicted in epoch 3. Thus use this hint to verify your solution (as a necessary (but not sufficient) condition).]

You are invited to solve this question by implementing the perceptron learning algorithm and applying the program to the given examples. Of course, you can solve it “by hand”, too.

Question 3 (3 marks)

(a) Construct a feed-forward neural network that has at most one hidden layer (i. e., it is a perceptron or a feed-forward 2-layer network) and represents the 3-ary Boolean function

$$f_1(x_1, x_2, x_3) = (x_1 \equiv (x_2 \wedge x_3))$$

using only step functions as activation functions.[1]

(b) Can the following Boolean function be represented by a *perceptron*, using only step functions? If yes, present one, if no, explain why this is the case.

$$f_2(x_1, x_2, x_3) = (x_1 \wedge x_2 \wedge x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge \neg x_3))$$

[1] $\mathit{step}_t(x) = 0$, if $x < t$; $\mathit{step}_t(x) = 1$, if $x \geq t$.