CS534 - HW 1

Keith Chester

Due date: June 12 2022

Problem 1

In problem 1, we are tasked with creating a recursive and linear time agent for following propositonal logic statements. The work associated with this problem can be found in *problem1.py*.

Problem 2

In this problem we are exploring a first-order logical knowledge base and writing out logical expressions utilizing it. The knowledge base is represented as:

- CopyOf(d, a) - Predicate - Disk d is a copy of album a
- Owns(p, d) - Predicate - Person p owns disk d
- Sings(p, s, a) - Predicate - Album a inludes a recording of song s sung by person p
- Wrote(p, s) - Person p wrote song s

We are also injecting the following constants:

- $McCartney$ - a person
- $Gershwin$ - a person
- $BHoliday$ - a person
- Joe - a person
- $EleanorRigby$ - a song
- $TheManILove$ - a song
- $Revolver$ - an album

Within this, we express the following statements using first-order logic:

(a) $\text{Wrote}(Gershwin, TheManILove)$

(b) $\neg\,\text{Wrote}(Gershwin, EleanorRigby)$

(c) $\text{Wrote}(Gershwin, TheManILove) \vee \text{Wrote}(McCartney, TheManILove)$

(d) $\exists s\, \text{Wrote}(Joe, s)$

(e) $\exists d\, \text{Owns}(Joe, CopyOf(d, Revolver))$

(f) $\forall s\, \text{Sings}(McCartney, s, Revolver) \Rightarrow \text{Wrote}(McCartney, s)$

(g) $\forall s\, \text{Sings}(p, s, Revolver) \neg\, \text{Wrote}(Gershwin, s)$

(h) $\forall s\, \text{Wrote}(Gershwin, s) \Rightarrow \text{Sings}(p, s, a)$

(i) $\exists a \forall s\, \text{Sings}(p, s, a)\, \text{Wrote}(Joe, s)$

(j) $\exists d, a\, \text{Owns}(Joe, \text{CopyOf}(d, a)) \wedge \text{Sings}(BHoliday, s, a)$

(k) $\exists d_i\, \text{CopyOf}(d_i, a) \forall a\, \text{Sings}(McCartney, a, s) \wedge \forall\, \text{Owns}(Joe, d)$

(l) $\exists d\, \text{CopyOf}(d, a) \forall a\, \text{Sings}(BHoliday, s, a) \wedge \text{Owns}(Joe, d)$

Problem 3

In this queston, we are looking at the following table of three binary input atributes, and a singular binary output:

Example	A_1	A_2	A_3	Output y
x_1	1	0	0	0
x_2	1	0	1	0
x_3	0	1	0	0
x_4	1	1	1	1
x_5	1	1	0	1

a.

Using the Gini Index, we aim to create a decision tree for this data.

b.

Now we utilize Information Gain to create a decision tree for this data.

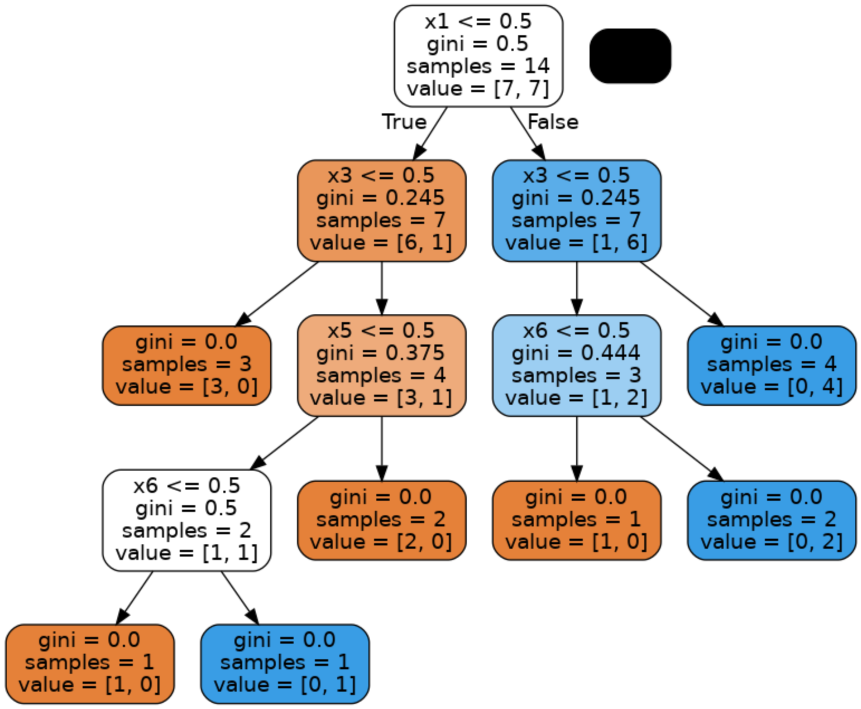
Problem 4

In this section, we consider the following dta input with six inputs and a singular target output:

Example	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14
x_1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
x_2	0	0	0	1	1	0	0	1	1	0	1	0	1	1
x_3	1	1	1	0	1	0	0	1	1	0	0	0	1	1
x_4	0	1	0	0	1	0	0	1	0	1	1	1	0	1
x_5	0	0	1	1	0	1	1	0	1	1	0	0	1	0
x_6	0	0	0	1	0	1	0	1	1	0	1	1	1	0
T	1	1	1	1	1	1	0	1	0	0	0	0	0	0

When we run our code found in *problem4.py*, we find that we can train both a perceptron and decision tree on this data. If given unseen data of $A_{15} :< 1, 1, 0, 0, 1, 1 >$ with result $T_{15} = 1$, we find that both accurately predict this result.

Whne we generate the decision tree, we can output its resulting branches, seen as follows:



Our code that utilizes the perceptron makes a perceptron makes a perceptron enacting the following linear equation:

$$T = 6x_1 + 0x_2 + 3x_3 - 2x_4 - 4x_5 + 4x_6 - 4 \tag{1}$$