

# 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 propositional 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:

- $\text{CopyOf}(d, a)$  - Predicate - Disk  $d$  is a copy of album  $a$
- $\text{Owns}(p, d)$  - Predicate - Person  $p$  owns disk  $d$
- $\text{Sings}(p, s, a)$  - Predicate - Album  $a$  includes a recording of song  $s$  sung by person  $p$
- $\text{Wrote}(p, s)$  - 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)  $\text{Wrote}(\textit{Gershwin}, \textit{TheManILove})$
- (b)  $\neg \text{Wrote}(\textit{Gershwin}, \textit{EleanorRigby})$
- (c)  $\text{Wrote}(\textit{Gershwin}, \textit{TheManILove}) \vee \text{Wrote}(\textit{McCartney}, \textit{TheManILove})$
- (d)  $\exists s \text{Wrote}(\textit{Joe}, s)$
- (e)  $\exists d \text{Owns}(\textit{Joe}, \text{CopyOf}(d, \textit{Revolver}))$
- (f)  $\forall s \text{Sings}(\textit{McCartney}, s, \textit{Revolver}) \Rightarrow \text{Wrote}(\textit{McCartney}, s)$
- (g)  $\forall s \text{Sings}(p, s, \textit{Revolver}) \neg \text{Wrote}(\textit{Gershwin}, s)$
- (h)  $\forall s \text{Wrote}(\textit{Gershwin}, s) \Rightarrow \text{Sings}(p, s, a)$

- (i)  $\exists a \forall s \text{Sings}(p, s, a) \text{Wrote}(\text{Joe}, s)$
- (j)  $\exists d, a \text{Owns}(\text{Joe}, \text{CopyOf}(d, a)) \wedge \text{Sings}(\text{BHoliday}, s, a)$
- (k)  $\exists d_i \text{CopyOf}(d_i, a) \forall a \text{Sings}(\text{McCartney}, a, s) \wedge \forall \text{Owns}(\text{Joe}, d)$
- (l)  $\exists d \text{CopyOf}(d, a) \forall a \text{Sings}(\text{BHoliday}, s, a) \wedge \text{Owns}(\text{Joe}, d)$

### Problem 3

In this question, we are looking at the following table of three binary input attributes, 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. For Gini Index, we are looking at the probability distribution of a particular variable being wrong when randomly chosen. From this we can begin to branch our tree into a set of decisions that lead us to a particular predictive result.

$$Gini = 1 - \sum_{i=1}^n p_i^2 \quad (1)$$

...where  $p_i$  is the probability of a given object being classified into a particular class.

So, calculating the Gini index for our given data:

For  $A_1$ , we are looking at the indexes of positive and negative outputs:

$$A_1 = 1, y = 1 \rightarrow \frac{2}{4} = 0.5 \quad (2)$$

$$A_1 = 1, y = 0 \rightarrow \frac{2}{4} = 0.5 \quad (3)$$

...for the Gini calculation of:

$$1 - (0.5^2 + 0.5^2) = 0.5 \quad (4)$$

$$A_1 = 0, y = 1 \rightarrow \frac{0}{1} = 0 \quad (5)$$

$$A_1 = 0, y = 0 \rightarrow \frac{1}{1} = 1 \quad (6)$$

...for the Gini calculation of:

$$1 - (0^2 + 1^2) = 0 \quad (7)$$

and the weighted sum of these Gini indexes to create the Gini index for  $A_1$ :

$$\sum p_i * g_i = (\frac{4}{5} * 0.5 + \frac{1}{5} * 0) = 0.4 \quad (8)$$

Now we continue this for  $A_2$ :

$$A_2 = 1, y = 1 \rightarrow \frac{2}{3} = 0.67 \quad (9)$$

$$A_2 = 1, y = 0 \rightarrow \frac{1}{3} = 0.33 \quad (10)$$

...for the Gini calculation of:

$$1 - (0.67^2 + 0.33^2) = 0.44 \quad (11)$$

$$A_2 = 0, y = 1 \rightarrow \frac{0}{2} = 0 \quad (12)$$

$$A_2 = 0, y = 0 \rightarrow \frac{2}{2} = 1 \quad (13)$$

...for the Gini calculation of:

$$1 - (0^2 + 1^2) = 0 \quad (14)$$

and the weighted sum of these Gini indexes to create the Gini index for  $A_2$ :

$$\sum p_i * g_i = (\frac{3}{5} * 0.44 + \frac{2}{5} * 0) = 0.27 \quad (15)$$

...and finally, following the process for  $A_3$ :

$$A_3 = 1, y = 1 \rightarrow \frac{1}{2} = 0.5 \quad (16)$$

$$A_3 = 1, y = 0 \rightarrow \frac{1}{2} = 0.5 \quad (17)$$

...for the Gini calculation of:

$$1 - (0.5^2 + 0.5^2) = 0.5 \quad (18)$$

$$A_3 = 0, y = 1 \rightarrow \frac{1}{3} = 0.33 \quad (19)$$

$$A_3 = 0, y = 0 \rightarrow \frac{2}{3} = 0.67 \quad (20)$$

...for the Gini calculation of:

$$1 - (0.33^2 + 0.67^2) = 0.44 \quad (21)$$

and the weighted sum of these Gini indexes to create the Gini index for  $A_3$ :

$$\sum p_i * g_i = (\frac{2}{5} * 0.5 + \frac{3}{5} * 0.44) = 0.27 \quad (22)$$

So the Gini indexes for each input is

Input	Gini index
$A_1$	0.4
$A_2$	0.27
$A_3$	0.27

Both  $A_3$  and  $A_2$  have the same Gini index, thus either can be utilized as the root node of our decision tree. For our purposes, we will choose  $A_3$ . From this we repeat the computations above for the positive and negative branches of our  $A_3$  input.

**Positive  $A_3$ :**

Example	$A_1$	$A_2$	$A_3$	Output $y$
$x_2$	1	0	1	0
$x_4$	1	1	1	1

Gini Index of  $A_1 = 1$  when  $A_3 = 1$ :

$$A_1 = 1, y = 1 \rightarrow \frac{1}{2} = 0.5 \quad (23)$$

$$A_1 = 1, y = 0 \rightarrow \frac{1}{2} = 0.5 \quad (24)$$

$$1 - (0.5^2 + 0.5^2) = 0.5 \quad (25)$$

Gini Index of  $A_1 = 0$  when  $A_3 = 1$  is not applicable, as there is no examples anymore of  $A_1 = 0$ . Thus our weighted Gini Index is still 0.5.

Gini Index of  $A_2 = 1$  when  $A_3 = 1$ :

$$A_2 = 1, y = 1 \rightarrow \frac{1}{1} = 1.0 \quad (26)$$

$$A_2 = 1, y = 0 \rightarrow \frac{0}{1} = 0.0 \quad (27)$$

$$1 - (1.0^2 + 0.0^2) = 0.0 \quad (28)$$

Ginin Index of  $A_2 = 0$  when  $A_3 = 1$ :

$$A_2 = 1, y = 1 \rightarrow \frac{0}{1} = 0.0 \quad (29)$$

$$A_2 = 1, y = 0 \rightarrow \frac{1}{1} = 1.0 \quad (30)$$

$$1 - (0^2 + 1.0^2) = 0.0 \quad (31)$$

...for a Gini index, when weighted, to still be 0.0 for  $A_2$ .

Input	Gini index
$A_1$	0.5
$A_2$	0.0

...so we see that we want to use  $A_2$  for the next split as it has the lowest Gini index, and completes this branch of the tree as we have a perfect split.

**Negative  $A_3$ :**

Example	$A_1$	$A_2$	$A_3$	Output $y$
$x_1$	1	0	0	0
$x_3$	0	1	0	0
$x_5$	1	1	0	1

Gini Index of  $A_1 = 1$  when  $A_3 = 0$ :

$$A_1 = 1, y = 1 \rightarrow \frac{1}{2} = 0.5 \quad (32)$$

$$A_1 = 1, y = 0 \rightarrow \frac{1}{2} = 0.5 \quad (33)$$

$$1 - (0.5^2 + 0.5^2) = 0.5 \quad (34)$$

Gini Index of  $A_1 = 0$  when  $A_3 = 0$ :

$$A_1 = 1, y = 1 \rightarrow \frac{0}{1} = 0 \quad (35)$$

$$A_1 = 1, y = 0 \rightarrow \frac{1}{1} = 1.0 \quad (36)$$

$$1 - (1.0^2 + 0.0^2) = 0.0 \quad (37)$$

...for the weighted sum of a Gini index of  $A_1$  when  $A_3 = 0$ :

$$\sum p_i * g_i = (\frac{2}{3} * 0.5 + \frac{1}{3} * 0) = 0.33 \quad (38)$$

Moving onto the Gini Index of  $A_2$  when  $A_3 = 0$ , we first explore when  $A_2 = 1$ :

$$A_2 = 1, y = 1 \rightarrow \frac{1}{2} = 0.5 \quad (39)$$

$$A_2 = 1, y = 0 \rightarrow \frac{1}{2} = 0.5 \quad (40)$$

$$1 - (0.5^2 + 0.5^2) = 0.5 \quad (41)$$

Looking at  $A_2 = 0$  when  $A_3 = 0$ :

$$A_2 = 1, y = 1 \rightarrow \frac{0}{1} = 0.0 \quad (42)$$

$$A_2 = 1, y = 0 \rightarrow \frac{1}{1} = 1.0 \quad (43)$$

$$1 - (1.0^2 + 0.0^2) = 0.0 \quad (44)$$

...and our weight index for  $A_2$  when  $A_3 = 0$  is:

$$\sum p_i * g_i = (\frac{2}{3} * 0.5 + \frac{1}{3} * 0) = 0.33 \quad (45)$$

for our collected Gini indexes being:

Input	Gini index
$A_1$	0.33
$A_2$	0.33

...thus making it inconclusive which of  $A_1$  or  $A_2$  we could branch off of. We shall choose  $A_1$  for our purposes. One more branch to go!

This next branch doesn't need much in the way of calculations; to save time, we can just observe that when  $A_3 = 0$  and  $A_1 = 1$ ,  $A_2 = y$ . When  $A_3 = 0$  and  $A_1 = 0$ ,  $y = 0$ .

Thus we exhaustively formed the following tree:

**b.**

Now we utilize Information Gain to create a decision tree for this data. Here we will be calculating the amount of entropy for each branch.

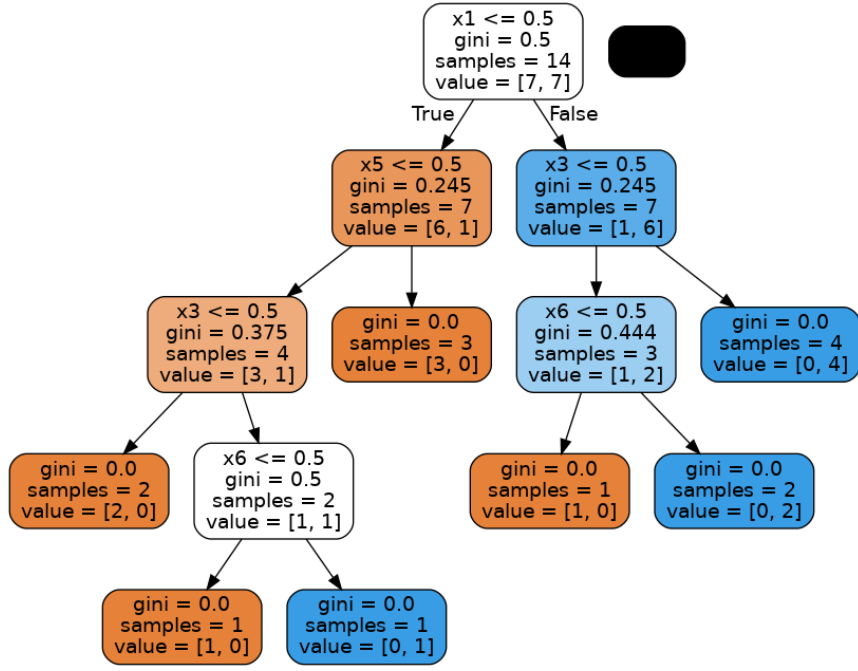
## Problem 4

In this section, we consider the following data 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.

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



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

$$T = 8x_1 + 0x_2 + 4x_3 - x_4 - 54x_5 + 3x_6 - 4 \quad (46)$$