Assignment 1

Adriaan Louw (53031377)

April 16, 2018

Contents

1 Question 1

The book *Introduction to Machine Learning* by Nils J, Nilsson can be found at http://robotics.stanford.edu/people/nilsson/MLBOOK.pdf and is 1.855 MB.

The book *A first encounter with Machine Learning* by Max Welling can be found at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.441.6238&rep=rep1&type=pdf and is 416 KB.

2 Question 2

We can represent some learning functions in machine learning as Boolean function. We can define a Boolean function as a function of the form

$$f(x_1, x_2, x_3, ...x_n) \tag{1}$$

Which maps a n-tuple of (0,1) values to 0,1. 0,1 can also be expressed as false,true.

There are 3 basic types of operations can be performed in Boolean functions. Firstly we have the "and" operation which uses the connective "." as in

$$x_1.z_2 \tag{2}$$

Which only returns true if x_1 and x_2 is true. Then there is the "or" operation represented as a "+"

$$x_1 + x_2 \tag{3}$$

Where it returns true if x_1 is true or if x_2 or both are true. Thirdly we have the negation operation indicated by a $^-$ as in

$$\bar{x_1} \tag{4}$$

This operation returns true if x_1 is false and false if x_1 is true.

The . and + operations are commutative

$$\begin{aligned} x_1.x_2 &= x_2.x_1 \\ x_1 + x_2 &= x_2 + x_1 \end{aligned} \tag{5}$$

and associative

$$\begin{aligned} x_1.x_2(x_3) &= x_1(x_2.x_3) \\ x_1 + x_2 + (x_3) &= x_1 + (x_2 + x_3) \end{aligned} \tag{6}$$

To commute between . and + we use DeMorgan's laws

$$\begin{aligned} \overline{x_1.x_2} &= \overline{x_1} + \overline{x_2} \\ \overline{x_1 + x_2} &= \overline{x_1}.\overline{x_2} \end{aligned} \tag{7}$$

Boolean functions can be broken down into various sub-classes. The first subclass is called *terms*. We can write these as $k_1k_2k_3...k_n$ where k_i are literals. An example would be the following term of size 4, $x_2.x3.\overline{x_6}.x_{12}$. This is also called a conjunctive literal as in all the terms are separated by the and operation.

Secondly we have *clauses* or . A clause is a function where the literals are separated by the or function. As in $k_1 + k_2 + ... + k_n$. An example would be $x_1 + \overline{5} + x_8$.

3 Question 3

4 Question 4

5 Question 5

6 Question 6

Entropy is used to calculate a decision tree in the ID3 algorithm. For a collection of Samples (S):

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{8}$$

If we assume we only have 2 outcomes, then we let the number of positive outcomes by p and the number of negative outcomes be n. Then we can write Equation 8 as

$$Entropy(p,n) = -\frac{p}{p+n}log_2\frac{p}{p+n} - \frac{n}{p+n}log_2\frac{n}{p+n} \tag{9}$$

More specifically we compare the gain of different attributes of the sample set. Then create the tree based on the attributes with the greatest gain

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{10}$$

We need to determine $Entropy(S)$

$$\begin{aligned} Entropy(p,n) &= -\frac{p}{p+n}log_2\frac{p}{p+n} - \frac{n}{p+n}log_2\frac{n}{p+n} \\ &= -\frac{3}{3+1}log_2\frac{3}{3+1} - \frac{1}{3+1}log_2\frac{1}{3+1} \\ &= \end{aligned} \tag{11}$$

Now we need to determine the gain for the 6 attributes to determine which one will be the root node.

For Sky we have sunny $p_1 = 3, n_1 = 0)$ and rainy $p_1 = 0, n_1 = 1)$

$$\begin{aligned} Gain(S,Sky) = &Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \\ =&?? - \frac{1}{4}Entropy(S_{Sunny}) - Entropy(S_{Rainy}) \\ =&?? - \frac{1}{4}Entropy(3,0) - Entropy(0,1) \\ =&?? - \frac{3}{4}(-\frac{3}{3}log_2\frac{3}{3} - \frac{0}{3}log_2\frac{0}{3}) - \frac{1}{4}(-\frac{0}{1}log_2\frac{0}{1} - \frac{1}{1}log_2\frac{1}{1}) \\ =& \end{aligned} \tag{12}$$

References