**Talha Chaudhry**

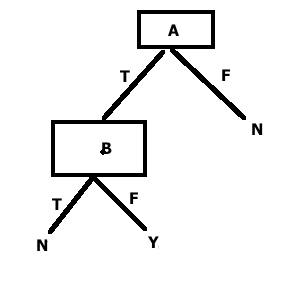
**COMP 4745**

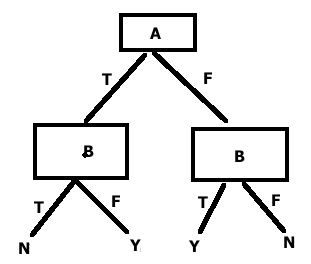
**HW 1**

**09/23/2019**

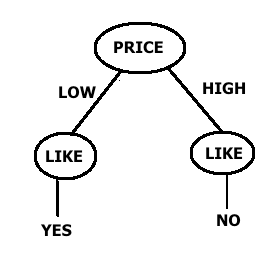
1. Decision Trees

* A ∧ ¬B

****

* A XOR B****

1. Decision Tree for Restaurant like or dislike:

****

1. The statement is false. In a decision tree, it is entirely possible, in fact common, that an attribute, or feature, maybe repeated as a node. For example, in Q1 A XOR B, the node B is repeated. In fact, there could be as many as 2n nodes for a dataset with n features.
2. Precision and Recall

Precision = true positives (correctly classified as faces)

true positives + false positives (incorrectly labeled as faces)

= = ½

Recall = true positives

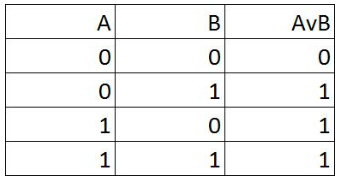
true positives + false negatives (faces incorrectly labeled as not)

= ) = 3/5

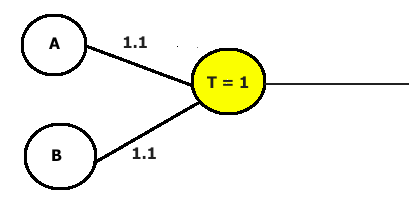
1. For a k-fold validation, the classifier should be trained k times.
2. We cannot conclude that A1 is a better algorithm than A2 simply because it has a lower training error. A lower training error could mean that A1 is “overfitting” the training dataset compared to A2, which probably signifies that it would not generalize well. Therefore, lower training error alone cannot determine whether an algorithm is better than another.
3. Perceptrons

* A ∨ B

Let’s look at the table:



Looking at the table, the perceptron should “fire”, i.e. give output as 1, when either or both of A & B is equal to 1. Therefore, the individual weights should be higher than the threshold. This can be achieved by setting individual weights, WA and WB, to 1.1, and the threshold can be θ = 1.

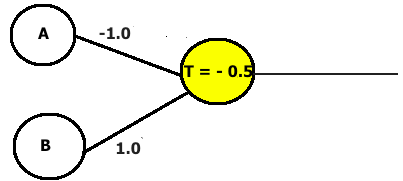


* ¬A ∨ B

Let’s look at the table:

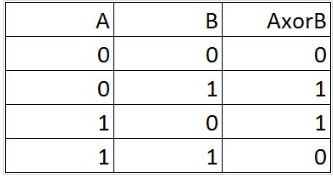
|  |  |  |
| --- | --- | --- |
| ¬A | B | ¬A ∨ B |
| 1 | 0 | 1 |
| 1 | 1 | 1 |
| 0 | 0 | 0 |
| 0 | 1 | 1 |

Looking at this table, the perceptron should not fire when A = 1 and B = 0. A way to assign weights to generate a perceptron to be is WA = -1, and WB = 1. The threshold is set at θ = -0.5. When A = 0, i.e. ¬A = 1, whatever B is the perceptron will fire because 0,1 > -0.5. When A = 1, i.e. ¬A = 0, the perceptron will not fire when B = 0 because -1 < -0.5, and fire otherwise.

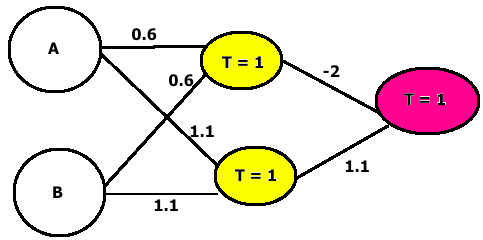


* A XOR B

Let’s look at the table:

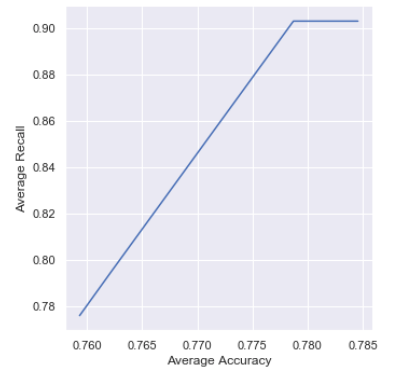


XOR cannot be portrayed by a single layer, will require two layers. The perceptron should fire precisely when either A = 1 or B = 1. Thus, weights should be assigned such that in the second layer the A,B =1 and A,B=0 do not cross the threshold.



If both A = B = 1, then the upper neuron takes precedence over the lower, and the perceptron will not fire because of the -2 weight in the next layer.

1. It is not necessary that the perceptron will have a 100% accuracy on all possible test data sets given that it has 100% accuracy on a training dataset. A counter example would suffice. Suppose that in a linearly separable dataset there are two distinct variables, X1 lies between 0 and 1 on the real line, and X2 lies between 1 and 2. Now suppose that the perceptron trains on a subset of the data, which will also be linearly separable, with the highest value of X1 being 0.9. It trains to have a ‘separator’ at 0.91-so 100% accuracy. However, in the test data set X1 takes values between 0.91 and 0.99, the perceptron will not have 100% accuracy.
2. Decisions trees can be good for real-time prediction. However, it would depend on the type of data. Less tree depth and fewer node spread would be essential. In some cases, training the tree might be expensive and time consuming, and real-time prediction could be much more difficult.
3. The following is the graph of average precision vs average recall of 5-fold decision tree cross validation at different pruning levels.



As it can be observed both accuracy and recall improve with higher pruning parameters, which is to be expected. Also, the feature most often showing up at the top of the decision trees (root node) was ‘Sex’. In fact, being male improved chances of survival considerably.