COM S 573: Machine Learning
Homework #3

1. (15 points) You are provided with a training set of examples. Which feature will you pick first to split the data as per the ID3 decision tree learning algorithm? Show all your work: compute the information gain for all the four attributes and pick the best one.

Sol: We have 14 data points and 4 features i.e. Outlook, Temperature, Humidity, and Wind.

$$\begin{aligned}\text{Total Entropy} &= E((Y=9)/14, (N=5)/14)\\ &= -9/14 * log_2 9/14 - 5/14 * log_2 5/14\\ &= 0.409 + 0.53 = 0.939\end{aligned}$$

The Information Gain (IG) for Outlook is Total Entropy - Weighted Entropy of feature Outlook
Entropy of branch sunny is equal to the entropy of branch rain i.e.

$$\begin{aligned}E(2/5, 3/5) &= -2/5 * log_2 2/5 - 3/5 * log_2 3/5\\ &= 0.528 + 0.442\\ &= 0.97\\ &= E(3/5, 2/5)\end{aligned}$$

$$\begin{aligned}\text{IG}_{\text{outlook}} &= 0.939 - 5/14 * E(2/5, 3/5) - 5/14 * E(3/5, 2/5)\\ &= 0.939 - 5/14 * 0.97 - 5/14 * 0.97\\ &= 0.249 \text{ bits}\end{aligned}$$

The Information Gain (IG) for Temperature is Total Entropy - Weighted Entropy of feature Temperature
The entropy of branch Hot is

$$\begin{aligned}E(2/4, 2/4) &= -2/4 * log_2 2/4 - 2/4 * log_2 2/4\\ &= 0.5 + 0.5\\ &= 1\end{aligned}$$

The entropy of branch Mild is

$$\begin{aligned}E(4/6, 2/6) &= -4/6 * log_2 4/6 - 2/6 * log_2 2/6\\ &= (0.117 + 0.159)/log_{10} 2\\ &= 0.916\end{aligned}$$

The entropy of branch Cool is

$$\begin{aligned}E(3/4, 1/4) &= -3/4 * log_2 3/4 - 1/4 * log_2 1/4\\ &= (0.093 + 0.15)/log_{10} 2\\ &= 0.807\end{aligned}$$

$$\begin{aligned}\text{IG}_{\text{Temperature}} &= 0.939 - 4/14 * E(2/4, 2/4) - 6/14 * E(4/6, 2/6) - 4/14 * E(3/4, 1/4) \\ &= 0.939 - 4/14 * 1 - 6/14 * 0.916 - 4/14 * 0.807 \\ &= 0.031 \text{ bits}\end{aligned}$$

The Information Gain (IG) for Humidity is Total Entropy - Weighted Entropy of feature Humidity
The entropy of branch High is

$$\begin{aligned}E(3/7, 4/7) &= -3/7 * log_2 3/7 - 4/7 * log_2 4/7 \\ &= (0.158 + 0.139)/log_{10} 2 \\ &= 0.987\end{aligned}$$

The entropy of branch Normal is

$$\begin{aligned}E(6/7, 1/7) &= -6/7 * log_2 6/7 - 1/7 * log_2 1/7 \\ &= (0.057 + 0.12)/log_{10} 2 \\ &= 0.588\end{aligned}$$

$$\begin{aligned}\text{IG}_{\text{Humidity}} &= 0.939 - 7/14 * E(3/7, 4/7) - 7/14 * E(6/7, 1/7) \\ &= 0.939 - 7/14 * 0.587 - 7/14 * 0.588 \\ &= 0.152 \text{ bits}\end{aligned}$$

The Information Gain (IG) for Wind is Total Entropy - Weighted Entropy of feature Wind
The entropy of branch Weak is

$$\begin{aligned}E(6/8, 2/8) &= -6/8 * log_2 6/8 - 2/8 * log_2 2/8 \\ &= (0.094 + 0.15)/log_{10} 2 \\ &= 0.81\end{aligned}$$

The entropy of branch Strong is

$$E(3/6, 3/6) = 1$$

$$\begin{aligned}\text{IG}_{\text{Wind}} &= 0.939 - 8/14 * E(6/8, 2/8) - 6/14 * E(3/6, 3/6) \\ &= 0.939 - 8/14 * 0.81 - 6/14 * 1 \\ &= 0.048 \text{ bits}\end{aligned}$$

Outlook has the more IG. Therefore, the best feature is outlook.

2. (15 points) We know that we can convert any decision tree into a set of if-then rules, where there is one rule per leaf node. Suppose you are given a set of rules $R = \{r_1, r_2, \ldots, r_k\}$, where r_i corresponds to the i^{th} rule. These rules are valid and complete, which means there is no conflicting rules. You can always obtain a prediction based on these rules. Is it possible to convert

the rule set R into an equivalent decision tree? Explain your construction or give a counterexample.

Sol: If each rule contains a root node and leaf node, and if there are no conflict rules we can construct a tree directly from the rules. Because each rule has a root node where the tree starts and ends at a leaf node as one path. If we go through all rules then paths will be created from a root node to a leaf node. Since the root node is the best feature node it will be common for all rules which construct a decision tree. If the rules do not have the best feature node we can add any rule from the best feature node to the node given in that particular rule and complete the path to a leaf node. This way we can always construct a decision tree from the rule set when there are no conflict rules. A logical AND operation is involved in each rule.

3. (20 points) Suppose $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$ and $\boldsymbol{z} = [z_1, z_2, \ldots, z_d]$ be two points in a high-dimensional space (i.e., d is very large).

   (a) (10 points) Try to prove the following, where the right-hand side quantity represent the standard Euclidean distance.

$$\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d} x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d} z_i\right)^2 \leq \sum_{i=1}^{d}(x_i - z_i)^2$$

   Hint: Use Jensen's inequality – If X is a random variable and f is a convex function, then $f(E[X]) \leq E[f(X)]$.

   Sol:

$$\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d}(x_i - z_i)\right)^2 \leq \sum_{i=1}^{d}(x_i - z_i)^2$$

   Multiply both sides by 1/d.

$$\frac{1}{d}\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d}(x_i - z_i)\right)^2 \leq \frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2$$

$$\left(\frac{1}{\sqrt{d}}\frac{1}{\sqrt{d}}\sum_{i=1}^{d}(x_i - z_i)\right)^2 \leq \frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2$$

$$\left(\frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)\right)^2 \leq \frac{1}{d}\sum_{i=1}^{d}(x_i - z_i)^2$$

   let $x_i - z_i = y_i$

$$\left(\frac{1}{d}\sum_{i=1}^{d} y_i\right)^2 \leq \frac{1}{d}\sum_{i=1}^{d} y_i^2$$

$$f(E[Y]) = \left(\frac{1}{d}\sum_{i=1}^{d} y_i\right)^2 \leq \frac{1}{d}\sum_{i=1}^{d} y_i^2 = E[f(Y)]$$

   Here f is the element-wise squaring operation on Y.

(b) (10 points) We know that the computation of nearest neighbors is very expensive in the high-dimensional space. Discuss how we can make use of the above property to make the nearest neighbors computation efficient?

Sol: The lower bound of true distance has only one squaring operation. Therefore, the lower bound is always computationally efficient. To show that the above inequality helps in reducing the computational complexity of K-nearest neighbours, we follow the steps given below:
• Step 1: find out the K random distances from the point X to point Z (one of the N training examples) using original distance formula $\sum_{i=1}^{d}(x_i - z_i)^2$ then get the maximum distance of those K distances.

• Step 2: Now find out the distance from X to Z, which is not in the K distances set, using the lower bound distance formula $\left(\frac{1}{sqrtd}\sum_{i=1}^{d}(x_i - z_i)\right)^2$. If the new distance is greater than the maximum distance then using the inequality true distance also greater than the maximum one which means the new distance does not lie in the K nearest distance set.

• Step 3: If the new distance is less than the maximum distance we are not certain about true distance is greater than the maximum distance. In that case we find out the true distance for that point and compare with the maximum distance. If true distance is greater than maximum we ignore that distance otherwise we keep that distance in the K nearest distance set and throw the maximum distance.
We repeat the step 2 and 3 with all the remaining examples. Therefore, the above inequality is useful to reduce computational burden in finding out the K nearest neighbours around X.

4. (50 points) **Car Evaluation**: You will build a Car Evaluation classifier. This classifier will be used to classify the condition of a car.

   The data: car_evaluation.csv. This is the data consisting of car evaluations.

   (a) (20 points) Implement the ID3 decision tree learning algorithm that we discussed in the class. The key step in the decision tree learning is choosing the next feature to split on. Implement the information gain heuristic for selecting the next feature. Please see lecture notes or `https://en.wikipedia.org/wiki/ID3_algorithm` for more details.

   (b) (20 points) Implement the decision tree pruning algorithm discussed in the class (via validation data).

   (c) (10 points) Compute the accuracy of decision tree and pruned decision tree on validation examples and testing examples. List your observations by comparing the performance of decision tree with and without pruning.
   Sol: On validation data pruned tree performance improved but on test data there is slight decrease in performance for pruned tree compared to with out pruned.

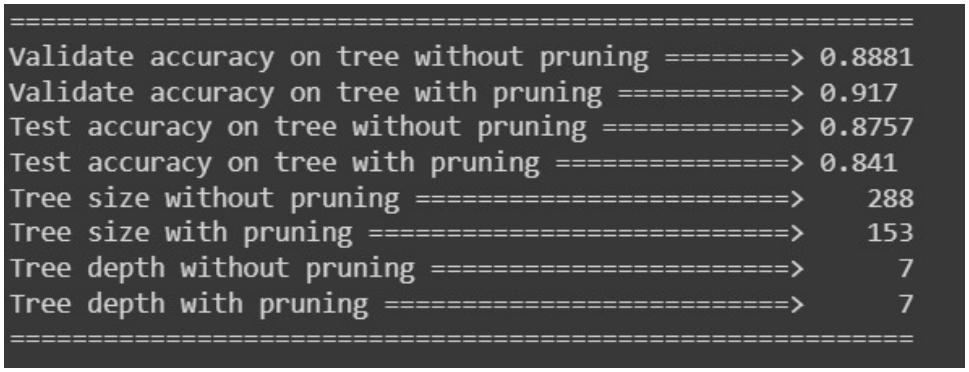

Figure 1: Accuracy table