COM S 5730 Homework 3

1. Please put required code files and report into a compressed file "HW3_FirstName_LastName.zip"
2. Unlimited number of submissions are allowed on Canvas and the latest one will be graded.
3. Due: **Tuesday Oct. 15, 2024 at 11:59pm.**
4. No later submission is accepted.
5. Please read and follow submission instructions. No exception will be made to accommodate incorrectly submitted files/reports.
6. All students are required to typeset their reports using latex. Overleaf (`https://www.overleaf.com/learn/latex/Tutorials`) can be a good start.

---

1. (15 points) You are provided with a training set of examples (see Figure 1). 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.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 1: Table with training examples. Each row corresponds to a single training example. There are four features, namely, outlook, temperature, humidity, and wind. "PlayTennis" is the class label.

Answer

In order to compute the Information Gain of each feature, we need to first compute the entropy of the dataset:

$$H(S) = -\sum_{i=1}^{K} P(S = y_i) \log_2 P(S = y_i)$$

PlayTenis (PT) - Entropy

$$
\begin{aligned}
H(PT) &= -P(PT = Yes)\log_2 P(PT = Yes) - P(PT = No)\log_2 P(PT = No) \\
&= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} \\
&= .4097 + .5305 \\
&= .9402
\end{aligned}
$$

Now, let's calculate the feature entropy and calculate the Information Gain for them.

Outlook (O) - Entropy

Outlook = Rain	**Outlook = Overcast**	**Outlook = Sunny**
$P(Outlook = Rain) = \frac{5}{14}$	$P(Outlook = Overcast) = \frac{4}{14}$	$P(Outlook = Sunny) = \frac{5}{14}$
$Yes : 3$	$Yes : 4$	$Yes : 2$
$No : 2$	$No : 0$	$No : 3$

Outlook = Rain

$$
\begin{aligned}
H(O_R) &= -P(O_R = True)\log_2 P(O_R = True) - P(O_R = False)\log_2 P(O_R = False) \\
&= -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \\
&= .4421 + .5288 \\
&= .9709
\end{aligned}
$$

Outlook = Sunny

$$
\begin{aligned}
H(O_S) &= -P(O_S = True)\log_2 P(O_S = True) - P(O_S = False)\log_2 P(O_S = False) \\
&= -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} \\
&= .5288 + .4421 \\
&= .9709
\end{aligned}
$$

Outlook = Overcast

$$
\begin{aligned}
H(O_O) &= -P(O_O = True)\log_2 P(O_O = True) - P(O_O = False)\log_2 P(O_O = False) \\
&= -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} \\
&= 0
\end{aligned}
$$

Outlook - Weighted Entropy

$$
\begin{aligned}
H(O) &= P(O_R)H(O_R) + P(O_S)H(O_S) + P(O_O)H(O_O) \\
&= (\frac{5}{14})(.9709) + (\frac{5}{14})(.9709) + (\frac{4}{14})(0) \\
&= .6935
\end{aligned}
$$

Outlook - Information Gain (IG)

$$
\begin{aligned}
IG(O) &= H(PT) - H(O) \\
&= .9402 - .6935 \\
&= .2467
\end{aligned}
$$

Temperature (T) - Entropy

Temperature = Cool	**Temperature = Mild**	**Outlook = Hot**
$P(Temperature = Cool) = \frac{4}{14}$	$P(Temperature = Mild) = \frac{6}{14}$	$P(Temperature = Hot) = \frac{4}{14}$
$Yes : 3$	$Yes : 4$	$Yes : 2$
$No : 1$	$No : 2$	$No : 2$

Temperature = Cool

$$
\begin{aligned}
H(T_C) &= -P(T_C = True)\log_2 P(T_C = True) - P(T_C = False)\log_2 P(T_C = False) \\
&= -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} \\
&= .3113 + .5 \\
&= .8113
\end{aligned}
$$

Temperature = Mild

$$
\begin{aligned}
H(T_M) &= -P(T_M = True)\log_2 P(T_M = True) - P(T_M = False)\log_2 P(T_M = False) \\
&= -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} \\
&= .3899 + .5283 \\
&= .9182
\end{aligned}
$$

Temperature = Hot

$$
\begin{aligned}
H(T_H) &= -P(T_H = True)\log_2 P(T_H = True) - P(T_H = False)\log_2 P(T_H = False) \\
&= -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} \\
&= .5 + .5 \\
&= 1
\end{aligned}
$$

Temperature - Weighted Entropy

$$
\begin{aligned}
H(T) &= P(T_C)H(T_C) + P(T_M)H(T_M) + P(T_H)H(T_H) \\
&= (\frac{4}{14})(.8113) + (\frac{6}{14})(.9182) + (\frac{4}{14})(1) \\
&= .9109
\end{aligned}
$$

Temperature - Information Gain (IG)

$$
\begin{aligned}
IG(T) &= H(PT) - H(T) \\
&= .9402 - .9109 \\
&= .0293
\end{aligned}
$$

Humidity (H) - Entropy

Humidity = Normal	**Humidity = High**
$P(Humidity = Normal) = \frac{7}{14}$	$P(Humidity = High) = \frac{7}{14}$
$Yes : 6$	$Yes : 3$
$No : 1$	$No : 4$

Humidity = Normal

$$
\begin{aligned}
H(H_N) &= -P(H_N = True)\log_2 P(H_N = True) - P(H_N = False)\log_2 P(H_N = False) \\
&= -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} \\
&= .1907 + .4011 \\
&= .5918
\end{aligned}
$$

Humidity = High

$$
\begin{aligned}
H(H_H) &= -P(H_H = True)\log_2 P(H_H = True) - P(H_H = False)\log_2 P(H_H = False) \\
&= -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} \\
&= .5239 + .4614 \\
&= .9853
\end{aligned}
$$

Humidity - Weighted Entropy

$$
\begin{aligned}
H(H) &= P(H_N)H(H_N) + P(H_H)H(H_H) \\
&= (\frac{7}{14})(.5918) + (\frac{7}{14})(.9853) \\
&= .7885
\end{aligned}
$$

Humidity - Information Gain (IG)

$$IG(H) = H(PT) - H(H)$$
$$= .9402 - .7885$$
$$= .1517$$

Wind (W) - Entropy

Wind = Weak	**Wind = Strong**
$P(Wind = Weak) = \frac{8}{14}$	$P(Wind = Strong) = \frac{6}{14}$
$Yes : 6$	$Yes : 3$
$No : 2$	$No : 3$

Wind = Weak

$$H(W_W) = -P(W_W = True)\log_2 P(W_W = True) - P(W_W = False)\log_2 P(W_W = False)$$
$$= -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$$
$$= .3113 + .5$$
$$= .8113$$

Wind = Strong

$$H(W_S) = -P(W_S = True)\log_2 P(W_S = True) - P(W_S = False)\log_2 P(W_S = False)$$
$$= -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}$$
$$= .5 + .5$$
$$= 1$$

Wind - Weighted Entropy

$$H(W) = P(W_W)H(W_W) + P(W_S)H(W_S)$$
$$= (\frac{8}{14})(.8113) + (\frac{6}{14})(1)$$
$$= .8922$$

Wind - Information Gain (IG)

$$IG(W) = H(PT) - H(W)$$
$$= .9402 - .8922$$
$$= .0480$$

Final Decision

$IG(Outlook) = .247$
$IG(Humidity) = .152$

$IG(Wind) = .048$
$IG(Temperature) = .029$

I'll pick the **Outlook** feature first to split the data as per ID3 decision tree learning algorithm because it has the highest Information Gain among the given features, which is .247.

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.

   Answer

   Given the set of rules are valid and complete, it's always possible to convert the set R into an equivalent decision tree. The construction involves using the conditions of each rule as branches from the root node to the leaf node, ensuring that the decision tree provides the same output as the given set of rules.

   Suppose we have the following rules:

   Rule 1: If $A = 1$ and $B = 2$, then $Class = 0$.
   Rule 2: If $A = 1$ and $B = 3$, then $Class = 1$.
   Rule 3: If $A = 2$, then $Class = 2$.

   Then, we have the following tree:

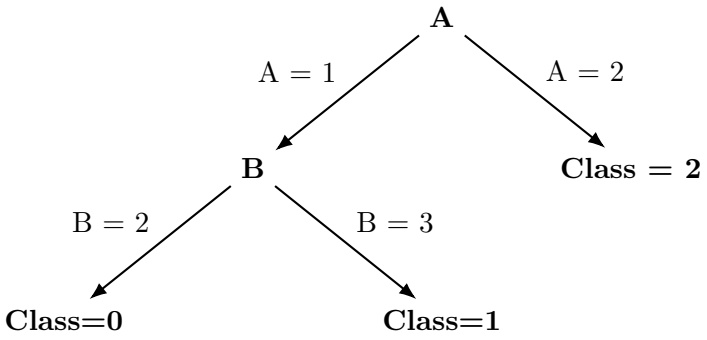

Figure 2: Decision Tree Representation

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 $f(E[X]) \leq E[f(X)]$.

   Answer

- If we denote:

$$y_i = x_i - z_i$$

- The inequality becomes:

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

Note: $f(x) = x^2$, which is a **convex function**.

- Consider y_i as samples of a random variable Y, with **equal probability**. That means the probability associated to each y_i is $p_i = \frac{1}{d}$
  - Expected value of Y:

$$E[Y] = \sum_{i=1}^{d} p_i y_i = \frac{1}{d} \sum_{i=1}^{d} y_i$$

  - Expected value of Y^2:

$$E[Y^2] = \sum_{i=1}^{d} p_i y_i^2 = \frac{1}{d} \sum_{i=1}^{d} y_i^2$$

- Jensen's Inequality with $f(x) = x^2$ function.

$$(E[Y])^2 \leq E[Y^2]$$

- Substitute the expected values.

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

- Multiply both sides by d, we get:

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

- Substitute y_i back to $x_i - z_i$.

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

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

This brings us back to the original inequality, thereby proving that the inequality holds for the given points x and z in high-dimensional space.

(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?

Answer

By using Jensen's inequality, we can quickly estimate a lower bound on the distance between points. This allows us to skip over points that are obviously too far away, which means we don't have to calculate the full distance for every single point. By reducing these unnecessary calculations, we make the nearest neighbor search much faster, especially in high-dimensional spaces.

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.

```
============================================================
Validate accuracy on tree without pruning ========> 0.8881
Validate accuracy on tree with pruning ===========> 0.6895
Test accuracy on tree without pruning ============> 0.8757
Test accuracy on tree with pruning ===============> 0.6908
Tree size without pruning ========================>    288
Tree size with pruning ===========================>      4
Tree depth without pruning =======================>      7
Tree depth with pruning ==========================>      2
============================================================
```

Accuracy Comparison

Both validation and test accuracy decreased after pruning (validation: 0.8881 to 0.6895, test: 0.8757 to 0.6908). This suggests that the pruned tree may be under-fitting compared to the non-pruned version, as it has lost predictive power on both datasets.

Tree Complexity

The pruned tree has significantly reduced size (from 288 nodes to 4 nodes) and depth (from 7 to 2). This reduction in complexity makes the model simpler, which resulted in the under-fitting observed in this experiment.

Generalization vs. Fit

Although pruning generally aims to reduce over-fitting and improve generalization, in this case, it appears that pruning caused under-fitting, resulting in lower performance on both the validation and test datasets.