CSC411 A1

Zhihong Wang 1002095207

September 2018

Q1:

(a). Determine $E[Z]$ and $V[Z]$, $Z = (X - Y)^2$:

The independent random variables X and Y are uniform distributed on $[0, 1]$, so we can get:
$E[X] = E[Y] = \frac{1}{2}(0 + 1) = \frac{1}{2}$,
$V[X] = V[Y] = \frac{1}{12}(1 - 0)^2 = \frac{1}{12}$

Also, we know $Z = (X - Y)^2$, so:
$E[Z] = E[(X - Y)^2] = E[X^2 - 2XY + Y^2] = E[X^2] - 2E[X]E[Y] + E[Y^2]$

By definition, we know $V[X] = E[X^2] - (E[X])^2$, so we can get:
$E[X^2] = V[X] + (E[X])^2 = \frac{1}{12} + \frac{1}{4} = \frac{1}{3}$
$E[Y^2] = V[Y] + (E[Y])^2 = \frac{1}{12} + \frac{1}{4} = \frac{1}{3}$

$E[Z] = E[X^2] - 2E[X]E[Y] + E[Y^2] = \frac{1}{3} - 2 \times \frac{1}{4} + \frac{1}{3} = \frac{1}{6}$

$$\begin{aligned} E[Z^2] &= E[X^4 - 4X^3Y + 6X^2Y^2 - 4XY^3 + Y^4] \\ &= E[X^4] - 4E[X^3Y] + 6E[X^2Y^2] - 4E[XY^3] + E[Y^4] \end{aligned}$$

Note:
$E[X^3] = E[Y^3] = \int_0^1 X^3 f(x)dx = \int_0^1 X^3 dx = [\frac{1}{4}X^4]_0^1 = \frac{1}{4}$
$E[X^4] = E[Y^4] = \int_0^1 X^4 f(x)dx = \int_0^1 X^4 dx = [\frac{1}{5}X^5]_0^1 = \frac{1}{5}$

$$\begin{aligned} E[Z^2] &= E[X^4] - 4E[X^3Y] + 6E[X^2Y^2] - 4E[XY^3] + E[Y^4] \\ &= E[X^4] - 4E[X^3]E[Y] + 6E[X^2]E[Y^2] - 4E[X]E[Y^3] + E[Y^4] \\ &= \frac{1}{5} - 4 \times \frac{1}{4} \times \frac{1}{2} + 6 \times \frac{1}{3} \times \frac{1}{3} - 4 \times \frac{1}{2} \times \frac{1}{4} + \frac{1}{5} \\ &= \frac{1}{15} \end{aligned}$$

$V[Z] = E[Z^2] - (E[Z])^2 = \frac{1}{15} - (\frac{1}{6})^2 = \frac{7}{180}$

Therefore, $E[Z] = \frac{1}{6}, V[Z] = \frac{7}{180}$

(b). Determine $E[R]$ and $V[R]$, $R = Z_1 + ... + Z_d$:

Because independent random variables $X_1...X_d$ and $Y_1...Y_d$ are uniform distributed on $[0, 1]$, $Z_i = (X_i - Y_i)^2$
So we can get $E[R] = E[Z_1 + ... + Z_d] = E[Z_1] + ... + E[Z_d], E[Z_1] = ... = E[Z_d]$
Based on (a), we also know $E[Z_1] = ... = E[Z_d] = \frac{1}{6}$
So $E[R] = \frac{1}{6} \times d = \frac{d}{6}$

Because $Z_1...Z_d$ are also independent, the co-variance between Z_s are all equal to zero.
So $V[R] = V[Z_1 + ... + Z_d] = V[Z_1] + ... + V[Z_d], V[Z_1] = ... = V[Z_d]$
Based on (a), we also know $V[Z_1] = ... = V[Z_d] = \frac{7}{180}$
So $V[R] = \frac{7}{180} \times d = \frac{7d}{180}$

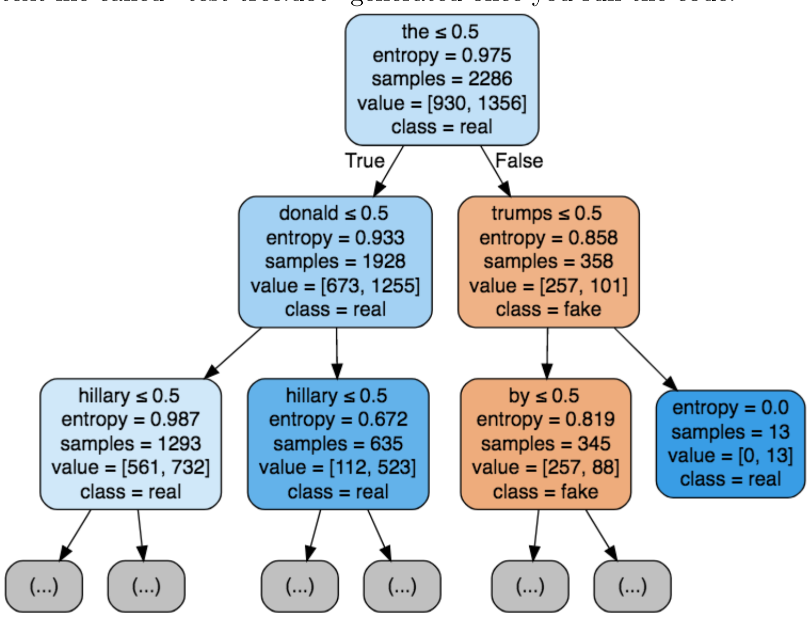
Q2:

(b). Using function "select model" trains the decision tree classifier using 5 different values of max depth, as well as two different split criteria ("gini", "entropy"), evaluates the performance of each one on the validation set, and prints the resulting accuracy of each model.

```
Depth:  3  Criteria:  gini  Accuracy:  0.718367346939
Depth:  3  Criteria:  entropy  Accuracy:  0.718367346939
Depth:  10  Criteria:  gini  Accuracy:  0.734693877551
Depth:  10  Criteria:  entropy  Accuracy:  0.724489795918
Depth:  20  Criteria:  gini  Accuracy:  0.740816326531
Depth:  20  Criteria:  entropy  Accuracy:  0.769387755102
Depth:  50  Criteria:  gini  Accuracy:  0.783673469388
Depth:  50  Criteria:  entropy  Accuracy:  0.759183673469
Depth:  100  Criteria:  gini  Accuracy:  0.765306122449
Depth:  100  Criteria:  entropy  Accuracy:  0.789795918367

Best accuracy:  0.789795918367
Best depth from [3, 10, 20, 50, 100]:  100
Best criteria from ['gini', 'entropy']:  entropy
```

(c). Extract and visualize the first two layers of the tree. You can find a display text file called "test tree.dot" generated once you run the code.



(d). Report the outputs of "compute information gain" function for the topmost split from the previous part, and for several other keywords.

This is the test body:

```python
if __name__ == '__main__':
    training, training_marker, validation, validation_marker, test, test_marker, vocabulary, \
    training_set_without_vectorized, total_data_with_splited_words, marker = load_data()

    select_model()

    print("\nTopmost split IG of handout tree data: ")
    IG_computor(1101, 1778, 890, 1778, 211, 0)

    print("\nTopmost split IG of Q2.(c): ")
    IG_computor(930, 1356, 673, 1255, 257, 101)

    print("==========================================================================================")

    keyword = "trump"
    print(
        "\nCompute the information gain by input total data without vectorized (if you want, "
        "you can use different data set, as long as they are not vectorized), "
        "the labels corresponding to the data,  and the keyword:", keyword)
    compute_information_gain(total_data_with_splited_words, marker, keyword)

    print(
        "\nCompute the information gain by input training data without vectorized (if you want, "
        "you can use different data set, as long as they are not vectorized), "
        "the labels corresponding to the data,  and the keyword:", keyword)
    compute_information_gain(training_set_without_vectorized, training_marker, keyword)

    print("==========================================================================================")

    keyword = "donald"
    print(
        "\nCompute the information gain by input total data without vectorized (if you want, "
        "you can use different data set, as long as they are not vectorized), "
        "the labels corresponding to the data,  and the keyword:", keyword)
    compute_information_gain(total_data_with_splited_words, marker, keyword)

    print(
        "\nCompute the information gain by input training data without vectorized (if you want, "
        "you can use different data set, as long as they are not vectorized), "
        "the labels corresponding to the data,  and the keyword:", keyword)
    compute_information_gain(training_set_without_vectorized, training_marker, keyword)

    print("==========================================================================================")

    keyword = "hillary"
    print(
        "\nCompute the information gain by input total data without vectorized (if you want, "
        "you can use different data set, as long as they are not vectorized), "
        "the labels corresponding to the data,  and the keyword:", keyword)
    compute_information_gain(total_data_with_splited_words, marker, keyword)

    print(
        "\nCompute the information gain by input training data without vectorized (if you want, "
        "you can use different data set, as long as they are not vectorized), "
        "the labels corresponding to the data,  and the keyword:", keyword)
    compute_information_gain(training_set_without_vectorized, training_marker, keyword)
```

(1). Topmost split IG of the tree data from the handout (1101, 1778, 890, 1778, 211, 0):

```
Topmost split IG of handout tree data:
H(y) =  0.9597363555454765  H(y|x1) =  0.9185455064660029  H(y|x2) =  -0.0
IG(y|x) =  0.10851043986249786
```

(2). Topmost split IG of Q2.(c) data (930, 1356, 673, 1255, 257, 101):

```
Topmost split IG of Q2.(c):
H(y) =  0.9748027561984685  H(y|x1) =  0.9332314976591101  H(y|x2) =  0.8583273166476142
IG(y|x) =  0.05330165959015257
```

(3). Topmost split IG when keyword is ”trump”:

```
================================================================================

Compute the information gain by input total data without vectorized (if you want, you can use different data set, as long as they are not vectorized), the labels
  corresponding to the data,  and the keyword: trump
H(y) =  0.9694262018413933  H(y|x1) =  0.9834291788698206  H(y|x2) =  0.348157663597672
IG(y|x) =  0.03365211214538655

Compute the information gain by input training data without vectorized (if you want, you can use different data set, as long as they are not vectorized), the labels
  corresponding to the data,  and the keyword: trump
H(y) =  0.9674182354913282  H(y|x1) =  0.9818262744730264  H(y|x2) =  0.34438910560526487
IG(y|x) =  0.033274268925732764
================================================================================
```

(4). Topmost split IG when keyword is ”donald”:

```
================================================================================

Compute the information gain by input total data without vectorized (if you want, you can use different data set, as long as they are not vectorized), the labels
  corresponding to the data,  and the keyword: donald
H(y) =  0.9694262018413933  H(y|x1) =  0.7526269564427315  H(y|x2) =  0.9992761831031816
IG(y|x) =  0.04989943188990653

Compute the information gain by input training data without vectorized (if you want, you can use different data set, as long as they are not vectorized), the labels
  corresponding to the data,  and the keyword: donald
H(y) =  0.9674182354913282  H(y|x1) =  0.7394432672886269  H(y|x2) =  0.9994785353169415
IG(y|x) =  0.05450437163963051
================================================================================
```

(5). Topmost split IG when the keyword is ”hillary”:

```
================================================================================

Compute the information gain by input total data without vectorized (if you want, you can use different data set, as long as they are not vectorized), the labels
  corresponding to the data,  and the keyword: hillary
H(y) =  0.9694262018413933  H(y|x1) =  0.5787946246321198  H(y|x2) =  0.951650393435414
IG(y|x) =  0.03764013901583019

Compute the information gain by input training data without vectorized (if you want, you can use different data set, as long as they are not vectorized), the labels
  corresponding to the data,  and the keyword: hillary
H(y) =  0.9674182354913282  H(y|x1) =  0.6532642567060225  H(y|x2) =  0.9507464722913228
IG(y|x) =  0.032157495332398534
```