ECE4950 HW1

Yi Chen yc2329 https://github.com/agraynel

February 28, 2017

1 Problem1

$S = [9+, 5-]$

$$Entropy(S) = -\frac{9}{14}log(\frac{9}{14}) - \frac{5}{14}log(\frac{5}{14}) = 0.940$$

1.1 Outlook information gain:

$S\_Sunny \leftarrow [2+, 3-]$

$S\_Overcast \leftarrow [4+, 0-]$

$S\_Rain \leftarrow [3+, 2-]$

$Gain(S, Outlook) = Entropy(S) - \sum_{v \in (Sunny, Overcast, Rain)} \frac{|S_v|}{S} Entropy(S_v)$

$= 0.940 + \frac{5}{14} * (\frac{2}{5}log\frac{2}{5} + \frac{3}{5}log\frac{3}{5}) + 0 + \frac{5}{14} * (\frac{2}{5}log\frac{2}{5} + \frac{3}{5}log\frac{3}{5})$

$= 0.940 - \frac{5}{14} * 0.971 - 0 - \frac{5}{14} * 0.971 = 0.940 - 0.694$

$$= 0.246$$

1.2 Temperature information gain:

$$S_Hot \leftarrow [2+, 2-]$$

$$S_Mild \leftarrow [4+, 2-]$$

$$S_Cool \leftarrow [3+, 1-]$$

$$Gain(S, Temperature) = Entropy(S) - \sum_{v \in (Hot, Mild, Cool)} \frac{|S_v|}{S} Entropy(S_v)$$

$$= 0.940 + \frac{4}{14} * (\frac{1}{2} log \frac{1}{2} + \frac{1}{2} log \frac{1}{2}) + \frac{6}{14} * (\frac{2}{3} log \frac{2}{3} + \frac{1}{3} log \frac{1}{3}) + \frac{4}{14} * (\frac{1}{4} log \frac{1}{4} + \frac{3}{4} log \frac{3}{4})$$

$$= 0.940 - \frac{4}{14} * 1 - \frac{6}{14} * 0.918 - \frac{4}{14} * 0.811 = 0.940 - 0.911$$

$$= 0.029$$

2 Problem2

Fig.2.1.

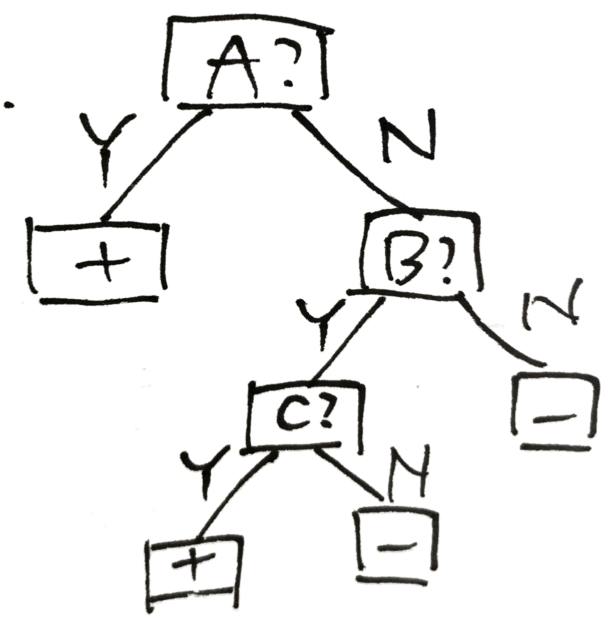


Fig 2.2

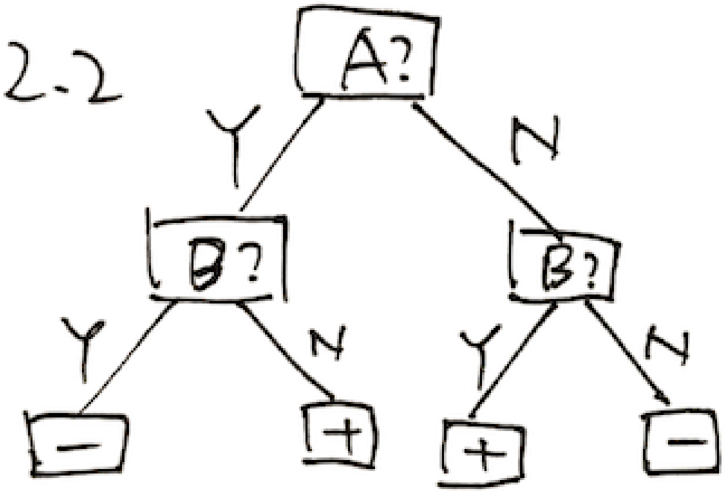
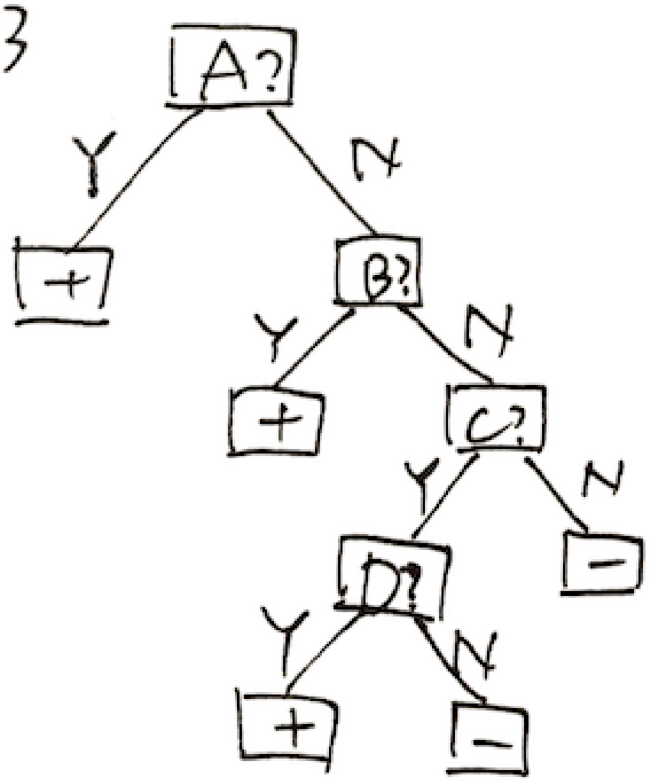


Fig 2.3



3 Problem3

3.1 Entropy of labels

$S = [4+, 5-]$

$$Entropy(S) = \frac{4}{9}log(\frac{4}{9}) + \frac{5}{9}log(\frac{5}{9}) = 0.991$$

3.2 Information gain

3.2.1 Feature1

$$
\begin{aligned}
& S_T \leftarrow [3+, 1-] \\
& S_F \leftarrow [1+, 4-] \\
& Entropy(S_T) = -\frac{2}{3}log(\frac{2}{3}) - \frac{1}{3}log(\frac{1}{3}) = 0.918 \\
& Entropy(S_F) = -\frac{1}{5}log(\frac{1}{5}) - \frac{4}{5}log(\frac{4}{5}) = 0.722 \\
& Gain(Label, Feature1) = Entropy(S) - \sum_{v \in (T,F)} \frac{|S_v|}{S} Entropy(S_v) \\
& \qquad = 0.991 - \frac{4}{9} * 0.811 - \frac{5}{9} * 0.722 = 0.229
\end{aligned}
$$

3.2.2 Feature2

$$
\begin{aligned}
& S_T \leftarrow [2+, 3-] \\
& S_F \leftarrow [2+, 2-] \\
& Entropy(S_T) = -\frac{2}{5}log(\frac{2}{5}) - \frac{3}{5}log(\frac{3}{5}) = 0.971 \\
& Entropy(S_F) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1 \\
& Gain(Label, Feature2) = Entropy(S) - \sum_{v \in (T,F)} \frac{|S_v|}{S} Entropy(S_v) \\
& \qquad = 0.991 - \frac{5}{9} * 0.971 - \frac{4}{9} * 1 = 0.007
\end{aligned}
$$

3.3 Information gain of feature3

Threshold 2.5 has the highest information gain. The information gain with respect to the threshold values 2.5, 3.5, 4.5, 5.5, 6.5, and 7.5 are calculated as follows:

3.3.1 threshold = 2.5

$$S_{<} = 2.5 \leftarrow [1+, 0-]$$

$$S_{>} 2.5 \leftarrow [3+, 5-]$$

$$Entropy(S_{<} = 2.5) = 0$$

$$Entropy(S_{>} 2.5) = -\frac{3}{8} log(\frac{3}{8}) - \frac{5}{8} log(\frac{5}{8}) = 0.954$$

$$Gain(S, threshold = 2.5) = 0.991 - \frac{8}{9} * 954 = 0.143$$

3.3.2 threshold = 3.5

$$S_{<} = 3.5 \leftarrow [1+, 1-]$$

$$S_{>} 3.5 \leftarrow [3+, 4-]$$

$$Entropy(S_{<} = 3.5) = -\frac{1}{2} log(\frac{1}{2}) - \frac{1}{2} log(\frac{1}{2}) = 1$$

$$Entropy(S_{>} 3.5) = -\frac{3}{7} log(\frac{3}{7}) - \frac{4}{7} log(\frac{4}{7}) = 0.985$$

$$Gain(S, threshold = 3.5) = 0.991 - \frac{2}{9} * 1 - \frac{7}{9} * 0.985 = 0.00249$$

3.3.3 threshold = 4.5

$$S_{<} = 4.5 \leftarrow [2+, 1-]$$

$$S_{>} 4.5 \leftarrow [2+, 4-]$$

$$Entropy(S_{<} = 4.5) = -\frac{2}{3} log(\frac{2}{3}) - \frac{1}{3} log(\frac{1}{3}) = 0.918$$

$$Entropy(S_{>} 4.5) = -\frac{2}{3} log(\frac{2}{3}) - \frac{1}{3} log(\frac{1}{3}) = 0.918$$

$$Gain(S, threshold = 4.5) = 0.991 - 0.918 = 0.0727$$

3.3.4 threshold = 5.5

$$S_< = 5.5 \leftarrow [2+, 3-]$$
$$S_> 5.5 \leftarrow [2+, 2-]$$
$$Entropy(S_< = 5.5) = -\frac{2}{5}log(\frac{2}{5}) - \frac{3}{5}log(\frac{3}{5}) = 0.971$$
$$Entropy(S_> 5.5) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1$$
$$Gain(S, threshold = 5.5) = 0.991 - \frac{5}{9} * 0.971 - \frac{4}{9} * 1 = 0.00714$$

3.3.5 threshold = 6.5

$$S_< = 6.5 \leftarrow [3+, 3-]$$
$$S_> 6.5 \leftarrow [1+, 2-]$$
$$Entropy(S_< = 6.5) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1$$
$$Entropy(S_> 6.5) = -\frac{2}{3}log(\frac{2}{3}) - \frac{1}{3}log(\frac{1}{3}) = 0.918$$
$$Gain(S, threshold = 6.5) = 0.991 - \frac{6}{9} * 1 - \frac{3}{9} * 0.918 = 0.0183$$

3.3.6 threshold = 7.5

$$S_< = 7.5 \leftarrow [4+, 4-]$$
$$S_> 7.5 \leftarrow [0+, 1-]$$
$$Entropy(S_< = 7.5) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1$$
$$Entropy(S_> 7.5) = 0$$
$$Gain(S, threshold = 7.5) = 0.991 - \frac{8}{9} * 1 = 0.102$$

3.4 First priority feature

Gain of feature1: 0.229

Gain of feature2: 0.007

Highest gain of feature3 with threshold 2.5: 0.143

Therefore, we choose feature 1.

3.5 Gini impurity measure

$$Gini(S) = 1 - (\frac{4}{9})^2 - (\frac{5}{9})^2 = \frac{40}{81}$$

3.5.1 gini feature 1

$$S_T \leftarrow [3+, 1-]$$

$$S_F \leftarrow [1+, 4-]$$

$$Gini(S_T) = 1 - (\frac{3}{4})^2 - (\frac{1}{4})^2 = \frac{3}{8}$$

$$Gini(S_F) = 1 - (\frac{1}{5})^2 - (\frac{4}{5})^2 = \frac{8}{25}$$

$$Gini(S, Feature1) = \frac{40}{81} - \frac{3}{8} * \frac{4}{9} - \frac{8}{25} * \frac{5}{9} = 0.149$$

3.5.2 gini feature 2

$$S_T \leftarrow [2+, 3-]$$

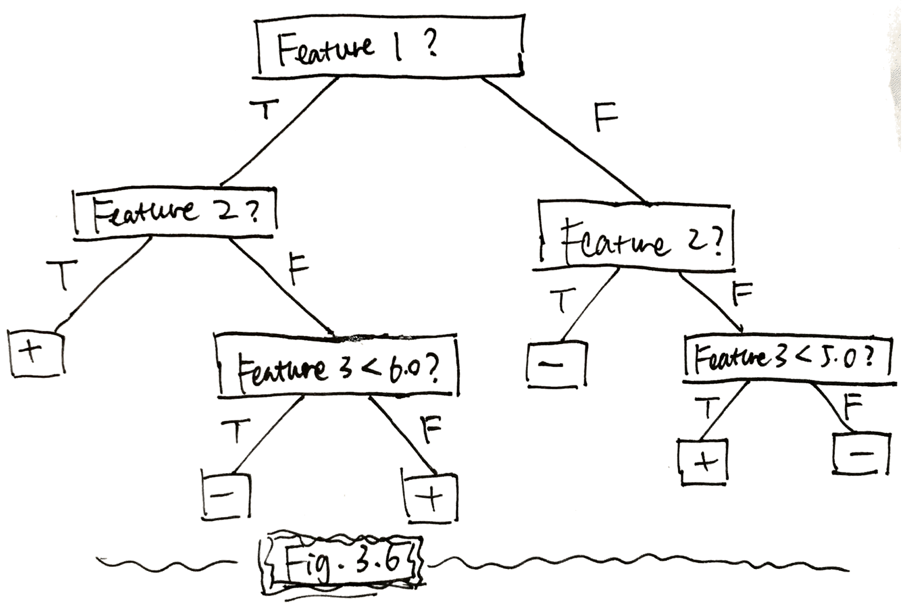
$$S_F \leftarrow [2+, 2-]$$

$$Gini(S_T) = 1 - (\frac{2}{5})^2 - (\frac{3}{5})^2 = \frac{12}{25}$$

$$Gini(S_F) = 1 - (\frac{1}{2})^2 - (\frac{1}{2})^2 = \frac{1}{2}$$

$$Gini(S, Feature1) = \frac{40}{81} - \frac{12}{25} * \frac{5}{9} - \frac{1}{2} * \frac{4}{9} = 0.0494$$

3.6 Construct decision tree gives correct answers

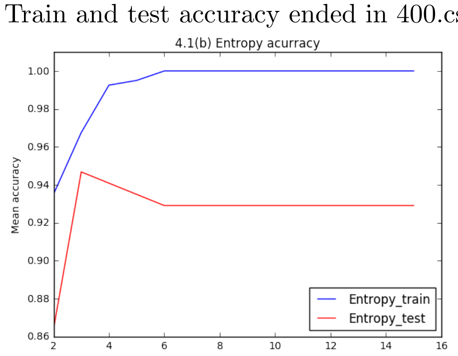


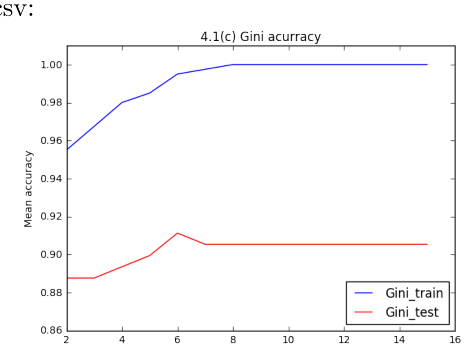
4 Problem4

Codes are in the part of appendix and zip file ended in *.ipynb.

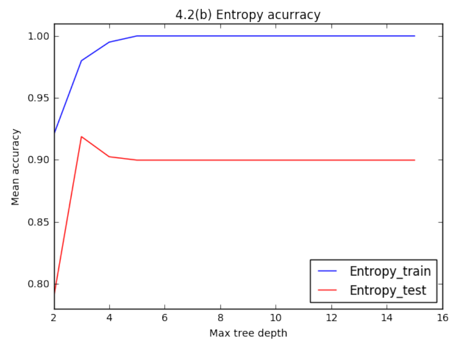
Entropy criterion is on the left, gini criterion is on the right.

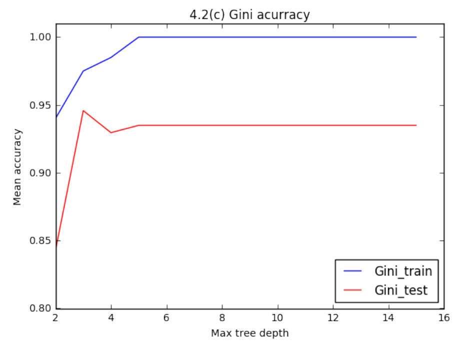
Train and test accuracy ended in 400.csv:





Train and test accuracy ended in 400.csv:





4.1 Two plots in data sets -400.csv

For the first plot, criterion is entropy, the tree depth corresponding to highest test accuracy is 3. And the second plot, whose criterion is gini, its tree depth corresponding to highest test accuracy is 6.
Gini's depth is larger than entropy's.
The criterion of entropy(information gain) is purer and more accurate in splitting the nodes apart. Thus, it takes fewer steps to get the best accuracy. And when the depth is larger, overfitting occurs, and the accuracy declines.

4.2 Two plots for gini impurity index

For gini impurity index, the first plot reading -400.csv, its tree depth corresponding to highest test accuracy is 6. And the second plot, who reads -200.csv, its tree depth corresponding to highest test accuracy is 3.
The second plot's depth is smaller than the first plot.
Because the first has 400 training examples, while the second has 200 examples. Thus, it takes more steps to reach the best result.

4.3 Two datasets

For entropy: dataset -400.csv has a higher test accuracy. Its highest accuracy is higher, and its average accuracy is also higher than -200.csv.
The reason might be, training data of -400.csv has more examples, thus it is

more accurate in predicting the test data.

For gini: dataset -200.csv has a higher test accuracy.

Maybe there are many training examples, thus overfitting occurs.

5 Appendix

5.1 Code for 4.1

```
#4.1 read 400 files

import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score

Xtrn400 = pd.read_csv('X-trn-400.csv', header = None)
Xtst400 = pd.read_csv('X-tst-400.csv', header = None)
Ytrn400 = pd.read_csv('Y-trn-400.csv', header = None)
Ytst400 = pd.read_csv('Y-tst-400.csv', header = None)

entropy_400_train = []
entropy_400_test = []
gini_400_train = []
gini_400_test = []

for i in range(2, 16):
    clf_entropy = DecisionTreeClassifier(criterion = '
        entropy', max_depth= i, random_state = 0)
    clf_entropy = clf_entropy.fit(Xtrn400, Ytrn400)
    entropy_test = clf_entropy.predict(Xtst400)
```

```
    entropy_400_train.append(clf_entropy.score(Xtrn400, Ytrn400))
    entropy_400_test.append(accuracy_score(Ytst400, entropy_test))

    clf_gini = DecisionTreeClassifier(criterion = 'gini', max_depth= i, random_state = 0)
    clf_gini = clf_gini.fit(Xtrn400, Ytrn400)
    gini_test = clf_gini.predict(Xtst400)
    gini_400_train.append(clf_gini.score(Xtrn400, Ytrn400))
    gini_400_test.append(accuracy_score(Ytst400, gini_test))

# Plotting decision regions

plt.figure(figsize=(15, 5))
plt.subplot(121)
plt.plot(range(2, 16), entropy_400_train, c='blue', label='Entropy_train')
plt.plot(range(2, 16), entropy_400_test, c='red', label='Entropy_test')
plt.legend(loc=4)
plt.ylim(0.86, 1.01)
plt.ylabel('Mean␣accuracy')
plt.xlabel('Max␣tree␣depth')
plt.title('4.1(b)␣Entropy␣acurracy')

plt.subplot(122)
```

```
plt.plot(range(2, 16), gini_400_train, c='blue', label='Gini train')
plt.plot(range(2, 16), gini_400_test, c='red', label='Gini test')
plt.legend(loc=4)
plt.ylim(0.86, 1.01)
plt.ylabel('Mean accuracy')
plt.xlabel('Max tree depth')
plt.title('4.1(c) Gini acurracy')

plt.show()
```

5.2 Code for 4.2

```
#4.2 read 200 files

import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score

Xtrn200 = pd.read_csv('X-trn-200.csv', header = None)
Xtst200 = pd.read_csv('X-tst-200.csv', header = None)
Ytrn200 = pd.read_csv('Y-trn-200.csv', header = None)
Ytst200 = pd.read_csv('Y-tst-200.csv', header = None)

entropy_200_train = []
entropy_200_test = []
gini_200_train = []
```

```
gini_200_test = []

for i in range(2, 16):
    clf_entropy = DecisionTreeClassifier(criterion = 'entropy', max_depth= i, random_state = 0)
    clf_entropy = clf_entropy.fit(Xtrn200, Ytrn200)
    entropy_test = clf_entropy.predict(Xtst200)
    entropy_200_train.append(clf_entropy.score(Xtrn200, Ytrn200))
    entropy_200_test.append(accuracy_score(Ytst200, entropy_test))

    clf_gini = DecisionTreeClassifier(criterion = 'gini', max_depth= i, random_state = 0)
    clf_gini = clf_gini.fit(Xtrn200, Ytrn200)
    gini_test = clf_gini.predict(Xtst200)
    gini_200_train.append(clf_gini.score(Xtrn200, Ytrn200))
    gini_200_test.append(accuracy_score(Ytst200, gini_test))

# Plotting decision regions

plt.figure(figsize=(15, 5))
plt.subplot(121)
plt.plot(range(2, 16), entropy_200_train, c='blue', label='Entropy_train')
plt.plot(range(2, 16), entropy_200_test, c='red', label='Entropy_test')
plt.legend(loc=4)
```

```
plt.ylim(0.78, 1.01)
plt.ylabel('Mean accuracy')
plt.xlabel('Max tree depth')
plt.title('4.2(b) Entropy acurracy')

plt.subplot(122)
plt.plot(range(2, 16), gini_200_train, c='blue', label='
    Gini_train')
plt.plot(range(2, 16), gini_200_test, c='red', label='
    Gini_test')
plt.legend(loc=4)
plt.ylim(0.8, 1.01)
plt.ylabel('Mean accuracy')
plt.xlabel('Max tree depth')
plt.title('4.2(c) Gini acurracy')

plt.show()
```