# CSCE 478/878 (Fall 2016) Homework 1

Baofeng Zhou, Feiyu Zhu

## I. Problem1

### A.

A hypothesis is that the function C represents that an instance $x$ is labeled as positive when $4 \leq x \leq 10$.

### B.

Version space is a set of hypotheses that consistent with the training set. According to the training set and properties of integer-valued attribute interval, the most general hypothesis is that $C(x)$ is label as positive when $2 \leq x \leq 11$; the most specific hypothesis is that $C(x)$ is label as positive when $5 \leq x \leq 8$. Thus, the lower bound $a$ can be chose from set $a \in \{2, 3, 4, 5\}$, upper bound value $b$ can be chose from set $b \in \{8, 9, 10, 11\}$. As a result the size of version space is $4 \times 4 = 16$.

### C.

Making a query on $x_1$ from $\forall x_1 \in (1, 5) \cap (8, 12)$ could reduce the version space. For example, by making a query about the label of $x_1 = 3$ we can reduce the size of version space. If $x_1 = 3$ is positive, the more specific hypothess extended and the lower bound set reduced to $a \in \{2\}$. Thus the version space reduced. Vice versa, if $x_1 = 3$ is negative, the more general hypothesis is more specified then the version is also reduced.

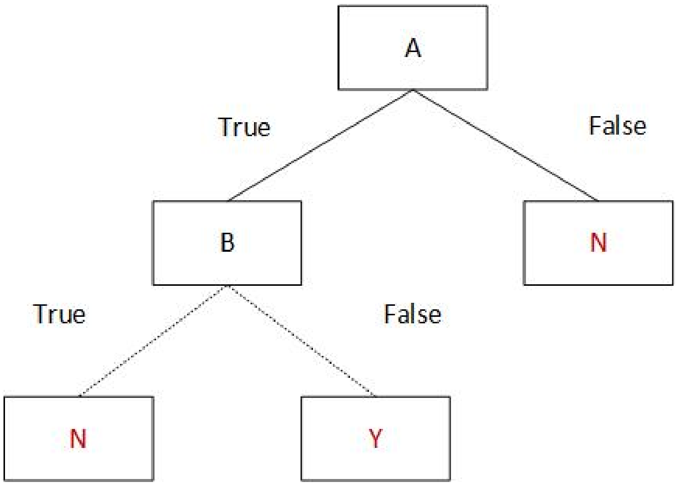
Making a query on $x_2$ from $\forall x_2 \in (-\infty, 1) \cap (5, 8) \cap (12, +\infty)$ will not change the version space. For example, by making a query about the label of $x_2 = 17$ we can guarantee not to change the version space. The answer will be negative according to the condition given by the problem that instance $x$ is labeled as positive if and only if $a \leq x \leq b$.

### D.

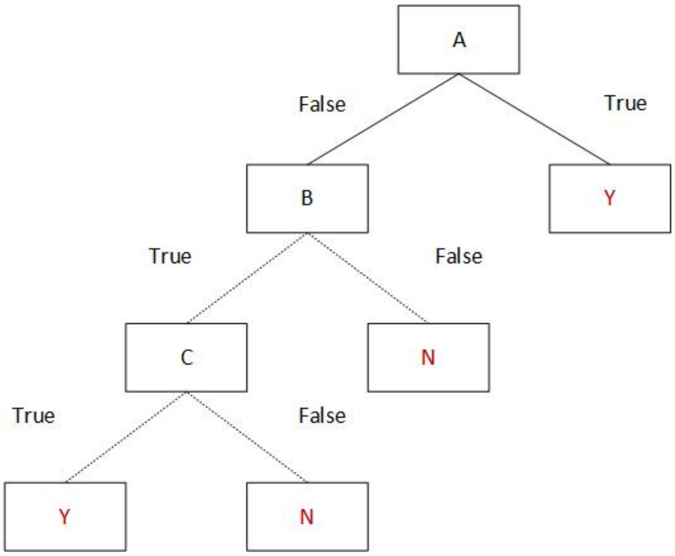
We can do a binary search in region $(x_1, x_2)$ and $(x_2, x_3)$ till we find a positive data, which will reduce the version space size. It costs $log(x_2 - x_1 - 1) + log(x_3 - x_2 - 1)$ queries which is less than $2 * log(x_3)$.

## II. Problem 2

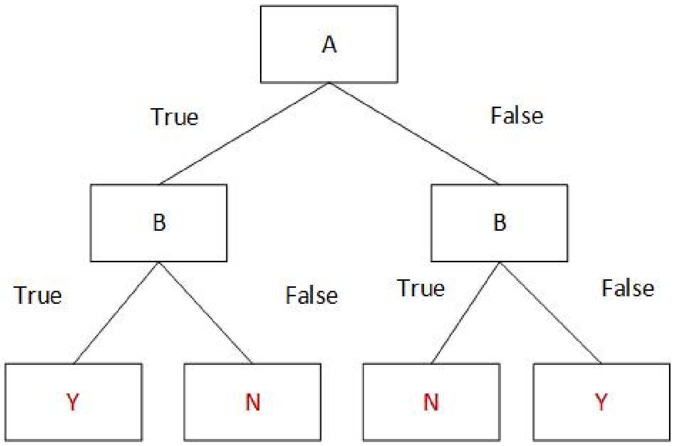
### A. *Decision tree representing the boolean function a.*



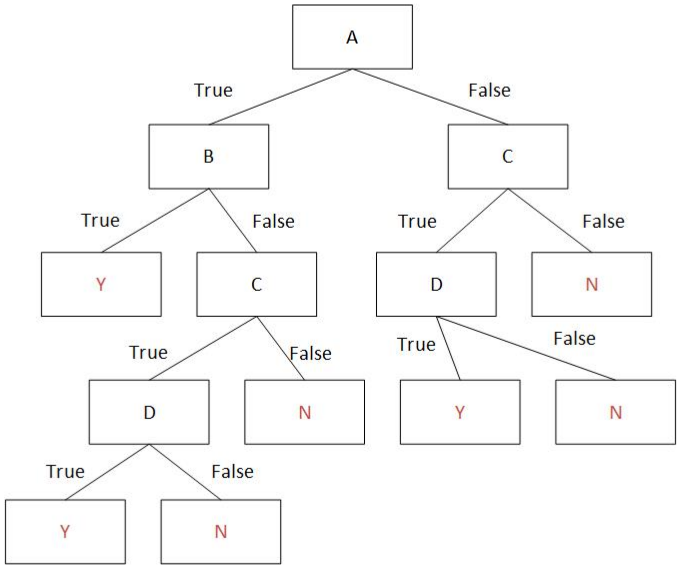
### B. *Decision tree representing the boolean function b.*



### C. Decision tree representing the boolean function $c$.



### D. Decision tree representing the boolean function $d$.



## III. Problem 3

### A.

The entropy of this data set is: $\mathcal{I}_m = -\frac{3}{6}log_2(\frac{3}{6}) - \frac{3}{6}log_2(\frac{3}{6}) = 1$

### B.

The ID3 algorithm will choose $a_1$ for minimum impurity.
The impurity of attribute $a_1$ set is: $\mathcal{I}_m = -\frac{3}{6} \times \frac{2}{3}log_2(\frac{2}{3}) - \frac{3}{6} \times \frac{1}{3}log_2(\frac{1}{3}) = 0.4591$
The impurity of attribute $a_2$ set is: $\mathcal{I}_m = -\frac{4}{6} \times \frac{1}{2}log_2(\frac{1}{2}) - \frac{2}{6} \times \frac{1}{2}log_2(\frac{1}{2}) = 0.5$

## IV. Problem 4

### A. Program Design

We used Python for this assignment for the reason that Python framework[1] provides very good ability on handling arrays and matrices, which play a crucial role in solving decision tree problems. The program we developed implements the ID3 algorithm to classify the target data sets [2].

The tree will be represented by cascaded dictionary. Every node as a key will map to another dictionary containing its corresponding decisions and every decision will map to another dictionary containing the node or leaf node this decision leads to. The hierarchy of the tree is structured visually from the output generated by our program in a graph formed by texts with the root at the left and leaves at the right.

The rules converted from the tree will be stored in three lists. The first list contains all the nodes of each path; The second list contains the edges every node leads to; the third list contains the final decision for every path.

The format of the tree and lists mentioned above will be presented in later figures, we will discuss about them in detail in later sections.

### B. Dataset Generation

We divide the data set into three subsets. Training data will be used to generate the decision tree; validation data will be used for post-pruning; test data will be used to check the error rate of the decision tree and the error rate of the rules generated from post-pruning. The three subsets will be randomly selected from the data set. The size of each subsets can be set manually in percentage.

We chose the ”wine” data set from the UCI repository [3] as our third data set. This data set contains all numeric values and has to be converted. We used Weka to discretize all the values of each attribute into three groups. After conversion, there are three possible nominal values for each attribute and the converted data set can be used for ID3 algorithm.

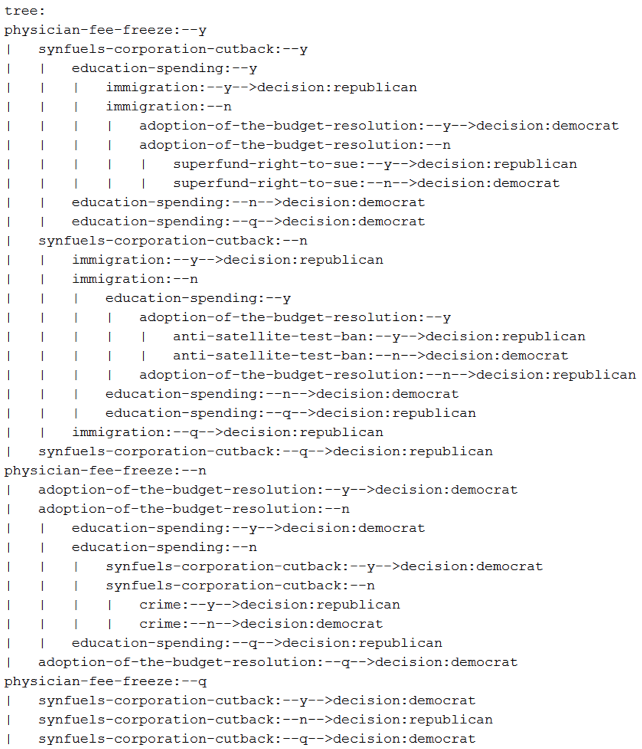
The data obtained from the database will all be preprocessed and converted to csv files. The first row of each file will be the names of attributes and the last column of each file will represent the class.

### C. Result Analysis

The main metric to evaluate the accuracy of our ID3 algorithm is the error rate on test sets. After getting the decision tree from the training data set our program will perform an error evaluation by feeding the training data set into the decision tree and get the errors by comparing the result from decision tree and the original class. To improve the error rate of our algorithm, we adopted postpruning on the decision tree we built. We will discuss the implementation of postpruning in later sections.

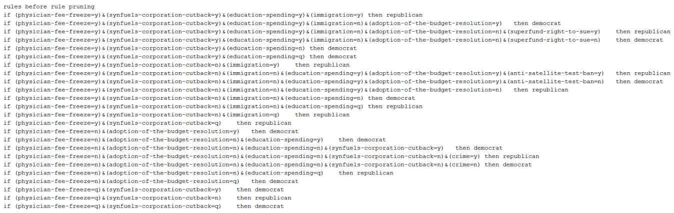
*1) Congressional Voting Data:*

*a) Size of the decision tree:* Of the total data set, we randomly picked 20% of the data set for test set, 16% for validation set, and 64% for training set. Since there are 435 instances in the data set, the test set and validation set will be large enough for sufficient testing and validation. The tree produced for data set ”Congressional Voting Data” has the shape shown in Fig. 1 (The leaf nodes that has NULL decision have been removed). The tree has 6 levels and 24 valid leaf nodes.



```
tree:
physician-fee-freeze:--y
|   synfuels-corporation-cutback:--y
|   |   education-spending:--y
|   |   |   immigration:--y-->decision:republican
|   |   |   immigration:--n
|   |   |   |   adoption-of-the-budget-resolution:--y-->decision:democrat
|   |   |   |   adoption-of-the-budget-resolution:--n
|   |   |   |   |   superfund-right-to-sue:--y-->decision:republican
|   |   |   |   |   superfund-right-to-sue:--n-->decision:democrat
|   |   education-spending:--n-->decision:democrat
|   |   education-spending:--q-->decision:democrat
|   synfuels-corporation-cutback:--n
|   |   immigration:--y-->decision:republican
|   |   immigration:--n
|   |   |   education-spending:--y
|   |   |   |   adoption-of-the-budget-resolution:--y
|   |   |   |   |   anti-satellite-test-ban:--y-->decision:republican
|   |   |   |   |   anti-satellite-test-ban:--n-->decision:democrat
|   |   |   |   adoption-of-the-budget-resolution:--n-->decision:republican
|   |   |   education-spending:--n-->decision:democrat
|   |   |   education-spending:--q-->decision:republican
|   |   immigration:--q-->decision:republican
|   synfuels-corporation-cutback:--q-->decision:republican
physician-fee-freeze:--n
|   adoption-of-the-budget-resolution:--y-->decision:democrat
|   adoption-of-the-budget-resolution:--n
|   |   education-spending:--y-->decision:democrat
|   |   education-spending:--n
|   |   |   synfuels-corporation-cutback:--y-->decision:democrat
|   |   |   synfuels-corporation-cutback:--n
|   |   |   |   crime:--y-->decision:republican
|   |   |   |   crime:--n-->decision:democrat
|   |   education-spending:--q-->decision:republican
|   adoption-of-the-budget-resolution:--q-->decision:democrat
physician-fee-freeze:--q
|   synfuels-corporation-cutback:--y-->decision:democrat
|   synfuels-corporation-cutback:--n-->decision:republican
|   synfuels-corporation-cutback:--q-->decision:democrat
```

Fig. 1: Tree for data set ”Congressional Voting Data”



```
rules before rule pruning
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=y) then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=n)&(adoption-of-the-budget-resolution=y)   then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=n)&(adoption-of-the-budget-resolution=n)&(superfund-right-to-sue=y)    then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=n)&(adoption-of-the-budget-resolution=n)&(superfund-right-to-sue=n)    then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=n) then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=q) then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=y)    then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=y)&(adoption-of-the-budget-resolution=y)&(anti-satellite-test-ban=y)   then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=y)&(adoption-of-the-budget-resolution=y)&(anti-satellite-test-ban=n)   then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=y)&(adoption-of-the-budget-resolution=n)   then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=n) then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=q) then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=q)    then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=q)    then republican
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=y)   then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=y)    then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=n)&(synfuels-corporation-cutback=y)   then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=n)&(synfuels-corporation-cutback=n)&(crime=y) then republican
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=n)&(synfuels-corporation-cutback=n)&(crime=n) then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=q)    then republican
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=q)   then democrat
if (physician-fee-freeze=q)&(synfuels-corporation-cutback=y)    then democrat
if (physician-fee-freeze=q)&(synfuels-corporation-cutback=n)    then republican
if (physician-fee-freeze=q)&(synfuels-corporation-cutback=q)    then democrat
```

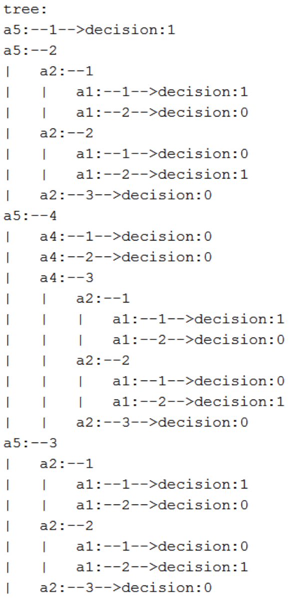
Fig. 2: Rules for data set ”Congressional Voting Data” before post-pruning

*b) Overfitting problem:* The overfitting can occur, as the accuracy rate for training data is 100% and there are NULL leaf nodes, which means that some cases that couldn’t be determined by this tree. Through later tests in Problem 5, we can see that there exist rules that can generalize better. The overfitting could be reduced by post-pruning.

The rules before pruning are listed in Fig. 2. The error rate on test set before rule pruning is 0.12643678160919541.

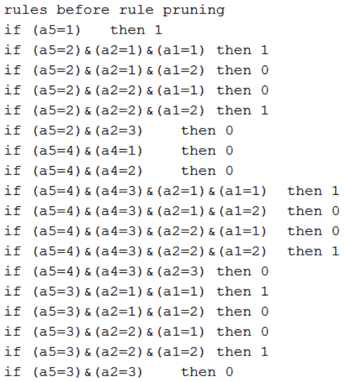
*2) Monks1 Data:*

*a) Size of the decision tree:* We used the same ratio for each set as that for dataset1 (20% for test set, 16% for validation set, and 64% for training set). The tree produced for data set ”Monks” has the shape in Fig. 3. The tree has 4 levels and 18 valid leaf nodes.



```
tree:
a5:--1-->decision:1
a5:--2
|   a2:--1
|   |   a1:--1-->decision:1
|   |   a1:--2-->decision:0
|   a2:--2
|   |   a1:--1-->decision:0
|   |   a1:--2-->decision:1
|   a2:--3-->decision:0
a5:--4
|   a4:--1-->decision:0
|   a4:--2-->decision:0
|   a4:--3
|   |   a2:--1
|   |   |   a1:--1-->decision:1
|   |   |   a1:--2-->decision:0
|   |   a2:--2
|   |   |   a1:--1-->decision:0
|   |   |   a1:--2-->decision:1
|   |   a2:--3-->decision:0
a5:--3
|   a2:--1
|   |   a1:--1-->decision:1
|   |   a1:--2-->decision:0
|   a2:--2
|   |   a1:--1-->decision:0
|   |   a1:--2-->decision:1
|   a2:--3-->decision:0
```

Fig. 3: Tree for data set ”Monks”



```
rules before rule pruning
if (a5=1)   then 1
if (a5=2)&(a2=1)&(a1=1) then 1
if (a5=2)&(a2=1)&(a1=2) then 0
if (a5=2)&(a2=2)&(a1=1) then 0
if (a5=2)&(a2=2)&(a1=2) then 1
if (a5=2)&(a2=3)    then 0
if (a5=4)&(a4=1)    then 0
if (a5=4)&(a4=2)    then 0
if (a5=4)&(a4=3)&(a2=1)&(a1=1)  then 1
if (a5=4)&(a4=3)&(a2=1)&(a1=2)  then 0
if (a5=4)&(a4=3)&(a2=2)&(a1=1)  then 0
if (a5=4)&(a4=3)&(a2=2)&(a1=2)  then 1
if (a5=4)&(a4=3)&(a2=3) then 0
if (a5=3)&(a2=1)&(a1=1) then 1
if (a5=3)&(a2=1)&(a1=2) then 0
if (a5=3)&(a2=2)&(a1=1) then 0
if (a5=3)&(a2=2)&(a1=2) then 1
if (a5=3)&(a2=3)    then 0
```

Fig. 4: Rules for data set ”Monks” before post-pruning

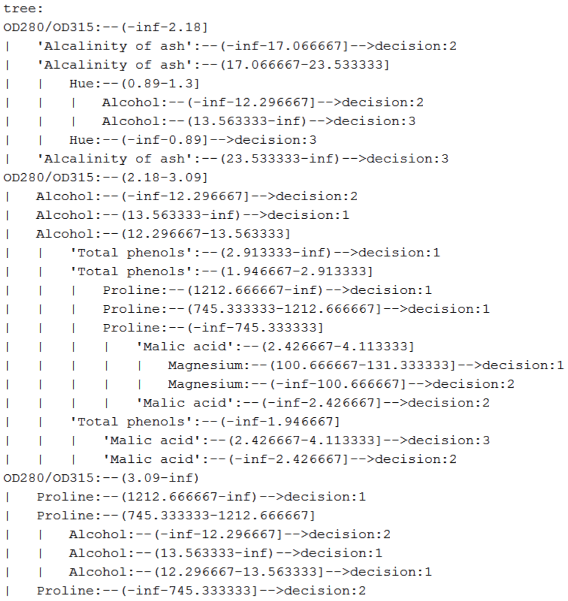
*b) Overfitting problem:* The overfitting can occur, although the tree generated is relatively small. Through later tests, we can see that the overfitting could be reduced by post-pruning.

The rules before pruning are listed in Fig.4. The error rate on test set before rule pruning is 0.02702702702702703.

*3) Wine Data:*

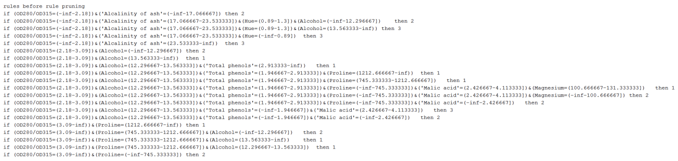
*a) Size of the decision tree:* For this dataset, we still used 20% for test set, 16% for validation set, and 64% for training set. The tree produced for data set ”Wine” has the shape in Fig. 5. The tree has 6 levels and 20 valid leaf nodes.

*b) Overfitting problem:* The overfitting can occur in this tree, as the accuracy rate for training data is 100% and there are NULL leaf nodes, which means that some cases that couldn’t be determined by this tree. Through later tests, we can see that there exist rules that can generalize better. The overfitting could be reduced by post-pruning.



```
tree:
OD280/OD315:--(-inf-2.18]
|   'Alcalinity of ash':--(-inf-17.066667]-->decision:2
|   'Alcalinity of ash':--(17.066667-23.533333]
|   |   Hue:--(0.89-1.3]
|   |   |   Alcohol:--(-inf-12.296667]-->decision:2
|   |   |   Alcohol:--(13.563333-inf)-->decision:3
|   |   Hue:--(-inf-0.89]-->decision:3
|   'Alcalinity of ash':--(23.533333-inf)-->decision:3
OD280/OD315:--(2.18-3.09]
|   Alcohol:--(-inf-12.296667]-->decision:2
|   Alcohol:--(13.563333-inf)-->decision:1
|   Alcohol:--(12.296667-13.563333]
|   |   'Total phenols':--(2.913333-inf)-->decision:1
|   |   'Total phenols':--(1.946667-2.913333]
|   |   |   Proline:--(1212.666667-inf)-->decision:1
|   |   |   Proline:--(745.333333-1212.666667]-->decision:1
|   |   |   Proline:--(-inf-745.333333]
|   |   |   |   'Malic acid':--(2.426667-4.113333]
|   |   |   |   |   Magnesium:--(100.666667-131.333333]-->decision:1
|   |   |   |   |   Magnesium:--(-inf-100.666667]-->decision:2
|   |   |   |   'Malic acid':--(-inf-2.426667]-->decision:2
|   |   'Total phenols':--(-inf-1.946667]
|   |   |   'Malic acid':--(2.426667-4.113333]-->decision:3
|   |   |   'Malic acid':--(-inf-2.426667]-->decision:2
OD280/OD315:--(3.09-inf)
|   Proline:--(1212.666667-inf)-->decision:1
|   Proline:--(745.333333-1212.666667]
|   |   Alcohol:--(-inf-12.296667]-->decision:2
|   |   Alcohol:--(13.563333-inf)-->decision:1
|   |   Alcohol:--(12.296667-13.563333]-->decision:1
|   Proline:--(-inf-745.333333]-->decision:2
```

Fig. 5: Tree for data set ”Wine”



```
rules before rule pruning
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(-inf-17.066667]) then 2
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(17.066667-23.533333])&(Hue=(0.89-1.3])&(Alcohol=(-inf-12.296667])    then 2
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(17.066667-23.533333])&(Hue=(0.89-1.3])&(Alcohol=(13.563333-inf)) then 3
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(17.066667-23.533333])&(Hue=(-inf-0.89])  then 3
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(23.533333-inf))  then 3
if (OD280/OD315=(2.18-3.09])&(Alcohol=(-inf-12.296667]) then 2
if (OD280/OD315=(2.18-3.09])&(Alcohol=(13.563333-inf))  then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(2.913333-inf))   then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(1212.666667-inf))  then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(745.333333-1212.666667])   then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(-inf-745.333333])&('Malic acid'=(2.426667-4.113333])&(Magnesium=(100.666667-131.333333])   then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(-inf-745.333333])&('Malic acid'=(2.426667-4.113333])&(Magnesium=(-inf-100.666667]) then 2
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(-inf-745.333333])&('Malic acid'=(-inf-2.426667])   then 2
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(-inf-1.946667])&('Malic acid'=(2.426667-4.113333])   then 3
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(-inf-1.946667])&('Malic acid'=(-inf-2.426667])   then 2
if (OD280/OD315=(3.09-inf))&(Proline=(1212.666667-inf)) then 1
if (OD280/OD315=(3.09-inf))&(Proline=(745.333333-1212.666667])&(Alcohol=(-inf-12.296667])   then 2
if (OD280/OD315=(3.09-inf))&(Proline=(745.333333-1212.666667])&(Alcohol=(13.563333-inf))    then 1
if (OD280/OD315=(3.09-inf))&(Proline=(745.333333-1212.666667])&(Alcohol=(12.296667-13.563333])  then 1
if (OD280/OD315=(3.09-inf))&(Proline=(-inf-745.333333]) then 2
```

Fig. 6: Rules for data set ”Wine” before post-pruning

The rules before pruning are listed in Fig. 6. The error rate on test set before rule pruning is 0.16666666666666666.

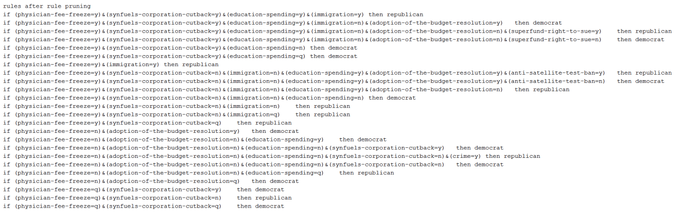
## V. Problem 5

### A. Program Design

For the post-pruning, we will use a trial and error method on the full trees we got from Problem 4. During the postpruning, we first delete one node tentatively from the rules, and check the error rate of the new set of rules. If the new error rate is lower than the original error rate, we will delete the node for good. We will continue this process for every node of the new rules until the error rate couldn't be lowered any more.

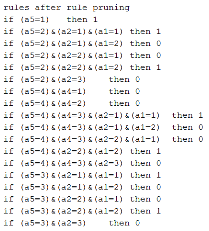
### B. Dataset Generation

The training set, validation set and test set will remain the same as generated randomly in Problem 4. A list contains indices from 1 to the length of the data set will be created first; then *x*% of these indices will be chosen randomly as the indices for test set; in the remaining indices, *y*% percent will be chosen randomly for validation set; there will be (1-*x*%)*(1-*y*%) of indices left for training set. The assignment for these three sets are different for each run.



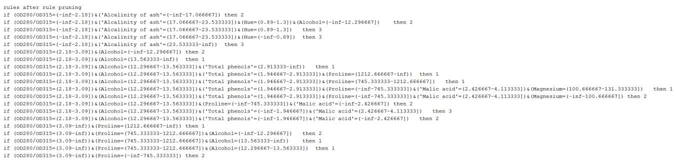
```
rules after rule pruning
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=y) then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=n)&(adoption-of-the-budget-resolution=y)   then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=n)&(adoption-of-the-budget-resolution=n)&(superfund-right-to-sue=y)    then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=y)&(immigration=n)&(adoption-of-the-budget-resolution=n)&(superfund-right-to-sue=n)    then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=n) then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=y)&(education-spending=q) then democrat
if (physician-fee-freeze=y)&(immigration=y) then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=y)&(adoption-of-the-budget-resolution=y)&(anti-satellite-test-ban=y)   then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=y)&(adoption-of-the-budget-resolution=y)&(anti-satellite-test-ban=n)   then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=y)&(adoption-of-the-budget-resolution=n)   then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)&(education-spending=n) then democrat
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=n)    then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=n)&(immigration=q)    then republican
if (physician-fee-freeze=y)&(synfuels-corporation-cutback=q)    then republican
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=y)   then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=y)    then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=n)&(synfuels-corporation-cutback=y)   then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=n)&(synfuels-corporation-cutback=n)&(crime=y) then republican
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=n)&(synfuels-corporation-cutback=n)   then democrat
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=n)&(education-spending=q)    then republican
if (physician-fee-freeze=n)&(adoption-of-the-budget-resolution=q)   then democrat
if (physician-fee-freeze=q)&(synfuels-corporation-cutback=y)    then democrat
if (physician-fee-freeze=q)&(synfuels-corporation-cutback=n)    then republican
if (physician-fee-freeze=q)&(synfuels-corporation-cutback=q)    then democrat
```

Fig. 7: Rules for data set ”Congressional Voting Data” after post-pruning



```
rules after rule pruning
if (a5=1)   then 1
if (a5=2)&(a2=1)&(a1=1) then 1
if (a5=2)&(a2=1)&(a1=2) then 0
if (a5=2)&(a2=2)&(a1=1) then 0
if (a5=2)&(a2=2)&(a1=2) then 1
if (a5=2)&(a2=3)    then 0
if (a5=4)&(a4=1)    then 0
if (a5=4)&(a4=2)    then 0
if (a5=4)&(a4=3)&(a2=1)&(a1=1)  then 1
if (a5=4)&(a4=3)&(a2=1)&(a1=2)  then 0
if (a5=4)&(a4=3)&(a2=2)&(a1=1)  then 0
if (a5=4)&(a2=2)&(a1=2) then 1
if (a5=4)&(a4=3)&(a2=3) then 0
if (a5=3)&(a2=1)&(a1=1) then 1
if (a5=3)&(a2=1)&(a1=2) then 0
if (a5=3)&(a2=2)&(a1=1) then 0
if (a5=3)&(a2=2)&(a1=2) then 1
if (a5=3)&(a2=3)    then 0
```

Fig. 8: Rules for data set ”Monks” after post-pruning



```
rules after rule pruning
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(-inf-17.066667]) then 2
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(17.066667-23.533333])&(Hue=(0.89-1.3])&(Alcohol=(-inf-12.296667])    then 2
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(17.066667-23.533333])&(Hue=(0.89-1.3])   then 3
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(17.066667-23.533333])&(Hue=(-inf-0.89])  then 3
if (OD280/OD315=(-inf-2.18])&('Alcalinity of ash'=(23.533333-inf))  then 3
if (OD280/OD315=(2.18-3.09])&(Alcohol=(-inf-12.296667]) then 2
if (OD280/OD315=(2.18-3.09])&(Alcohol=(13.563333-inf))  then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(2.913333-inf))   then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(1212.666667-inf))  then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(745.333333-1212.666667])   then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(-inf-745.333333])&('Malic acid'=(2.426667-4.113333])&(Magnesium=(100.666667-131.333333])   then 1
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(1.946667-2.913333])&(Proline=(-inf-745.333333])&('Malic acid'=(2.426667-4.113333])&(Magnesium=(-inf-100.666667]) then 2
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&(Proline=(-inf-745.333333])&('Malic acid'=(-inf-2.426667]) then 2
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(-inf-1.946667])&('Malic acid'=(2.426667-4.113333])   then 3
if (OD280/OD315=(2.18-3.09])&(Alcohol=(12.296667-13.563333])&('Total phenols'=(-inf-1.946667])&('Malic acid'=(-inf-2.426667])   then 2
if (OD280/OD315=(3.09-inf))&(Proline=(1212.666667-inf)) then 1
if (OD280/OD315=(3.09-inf))&(Proline=(745.333333-1212.666667])&(Alcohol=(-inf-12.296667])   then 2
if (OD280/OD315=(3.09-inf))&(Proline=(745.333333-1212.666667])&(Alcohol=(13.563333-inf))    then 1
if (OD280/OD315=(3.09-inf))&(Proline=(745.333333-1212.666667])&(Alcohol=(12.296667-13.563333])  then 1
if (OD280/OD315=(3.09-inf))&(Proline=(-inf-745.333333]) then 2
```

Fig. 9: Rules for data set 3 after post-pruning

### C. Result Analysis

*1) Congressional Voting Data:* We used validation set for post-pruning, the rules generated is listed in Fig. 7. The error rate after rule pruning is 0.09195402298850575. We can see a slight improvement of the error rate by 3%. For the attributes with ”Unknown disposition”, we treat them as a legitimate attribute.

*2) Monks1 Data:* The rules generated is listed in Fig. 8. The error rate after rule pruning is 0.0. Compared with the result before pruning, the error rate improved by 2.7%.

*3) Wine Data:* The rules generated is listed in Fig. 9. The error rate after rule pruning is 0.1388888888888889. Compared with the result before pruning, the error rate improved by 2.8%.

## References

[1] G. Van Rossum *et al.*, “Python documentation,” *Webová stránk a http://www. python. org/doc*, 2005.

[2] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[3] M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml