Jyotirmoy Sundi(108014811)

Machine Learning assignment 3 Report[a]

3 NEAREST NEIGHBOUR

a) Running with 64 size dataset, we see that for all digit, the 3 nearest neighbours obtained are same as the data sample in test data.

Trace for 64 dataset:

nearest 3 neighbours 1,1,1, - Query Point Digits in Test Data : 1

nearest 3 neighbours 2,2,2, - Query Point Digits in Test Data : 2

nearest 3 neighbours 3,3,3, - Query Point Digits in Test Data : 3

nearest 3 neighbours 4,4,4, - Query Point Digits in Test Data : 4

nearest 3 neighbours 5,5,5, - Query Point Digits in Test Data : 5

nearest 3 neighbours 6,6,6, - Query Point Digits in Test Data : 6

nearest 3 neighbours 7,7,7, - Query Point Digits in Test Data : 7

nearest 3 neighbours 8,8,8, - Query Point Digits in Test Data : 8

nearest 3 neighbours 9,9,9, - Query Point Digits in Test Data : 9

nearest 3 neighbours 0,0,0, - Query Point Digits in Test Data : 0

b) **1024 test case**

Running with 1024 size dataset, we see that for digit 4, the 3 nearest neighbours obtained are:

different from then obtained in 3NN for 64 bit datasize.

Trace for 1024 dataset:

nearest 3 neighbours 1,1,1, - Query Point Digits in Test Data : 1

nearest 3 neighbours 2,2,2, - Query Point Digits in Test Data : 2

nearest 3 neighbours 3,3,3, - Query Point Digits in Test Data : 3

nearest 3 neighbours 4,4,9, - Query Point Digits in Test Data : 4

nearest 3 neighbours 5,5,5, - Query Point Digits in Test Data : 5

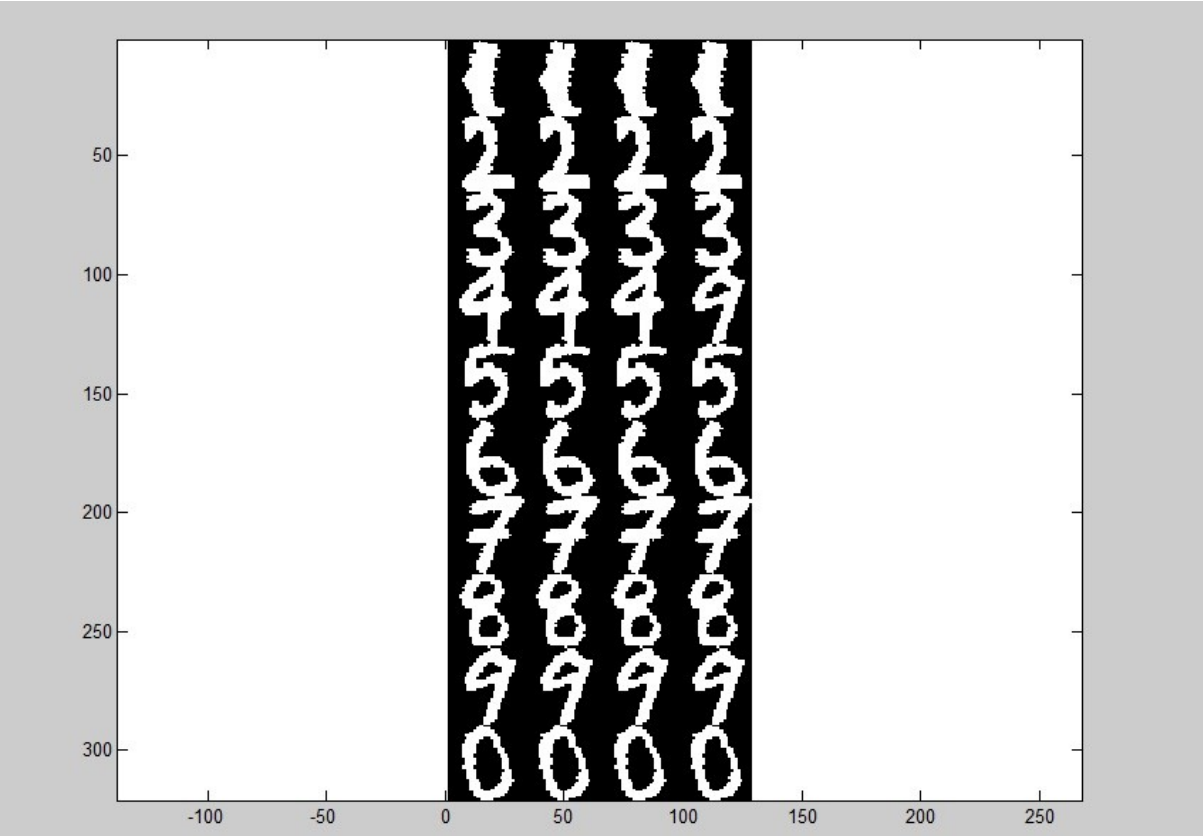
nearest 3 neighbours 6,6,6, - Query Point Digits in Test Data : 6

nearest 3 neighbours 7,7,7, - Query Point Digits in Test Data : 7

nearest 3 neighbours 8,8,8, - Query Point Digits in Test Data : 8

nearest 3 neighbours 9,9,9, - Query Point Digits in Test Data : 9

nearest 3 neighbours 0,0,0, - Query Point Digits in Test Data : 0



Please change the location of the input file from where the data is to be read. The code can be executed in eclipse by importing the file to a project and executing directly. Any standard java compilation and execution will do.

To see the images. execute plotknn.m , please provide the location of the file in which writeknn.txt is created. writeknn.txt contains the calculated data(by java program) for the 3NN model.

Files:
KNN.java
plotknn.m
writeknn.txt(this file is loaded by plotknn.m to produce the 10*4 matrix)

2. Decision Trees

a. DecisionTree using Info Gain:

The decision tree is printed in terms of equivalent set of rules as learned from the dataset.The attribute VISIBILITY is chosen in depth 1 as it has the highest information gain out of all the atributes . Similarly the attributes which has the highest info gain in successive levels is chosen for partitioning.

Tree using info gain(Trace)

```
if( VISIBILITY == "yes") {
        if( ERROR == "XL") {
                AUTO = "1";
        } else if( ERROR == "LX") {
                AUTO = "1";
        } else if( ERROR == "MM") {
                if( STABILItY == "stab") {
                        if( SIGN == "pp") {
                                if( MAGNITUDE == "Medium") {
                                        AUTO = "2";
                                } else if( MAGNITUDE == "Strong") {
                                        if( WIND == "head") {
                                                AUTO = "1";
                                        } else if( WIND == "tail") {
                                                        AUTO = "2";
                                        }
                                } else if( MAGNITUDE == "OutOfRange") {
                                        AUTO = "1";
                                } else    if( MAGNITUDE == "Low") {
                                        AUTO = "2";
                                }
                        } else if( SIGN == "nn") {
                                AUTO = "1";
                        }
                } else if( STABILItY == "xstab") {
                        AUTO = "1";
                }
        } else if( ERROR == "SS") {
                if( STABILItY == "stab") {
                        if( MAGNITUDE == "Medium") {
                                AUTO = "2";
                        } else if( MAGNITUDE == "Strong") {
                                AUTO = "2";
                        } else if( MAGNITUDE == "OutOfRange") {
```

```
                        AUTO = "1";
                    } else if( MAGNITUDE == "Low") {
                        AUTO = "2";
                    }
                } else  if( STABILItY == "xstab") {
                    AUTO = "1";
                }
        }
} else if( VISIBILITY == "no") {
                AUTO = "2";
}
```

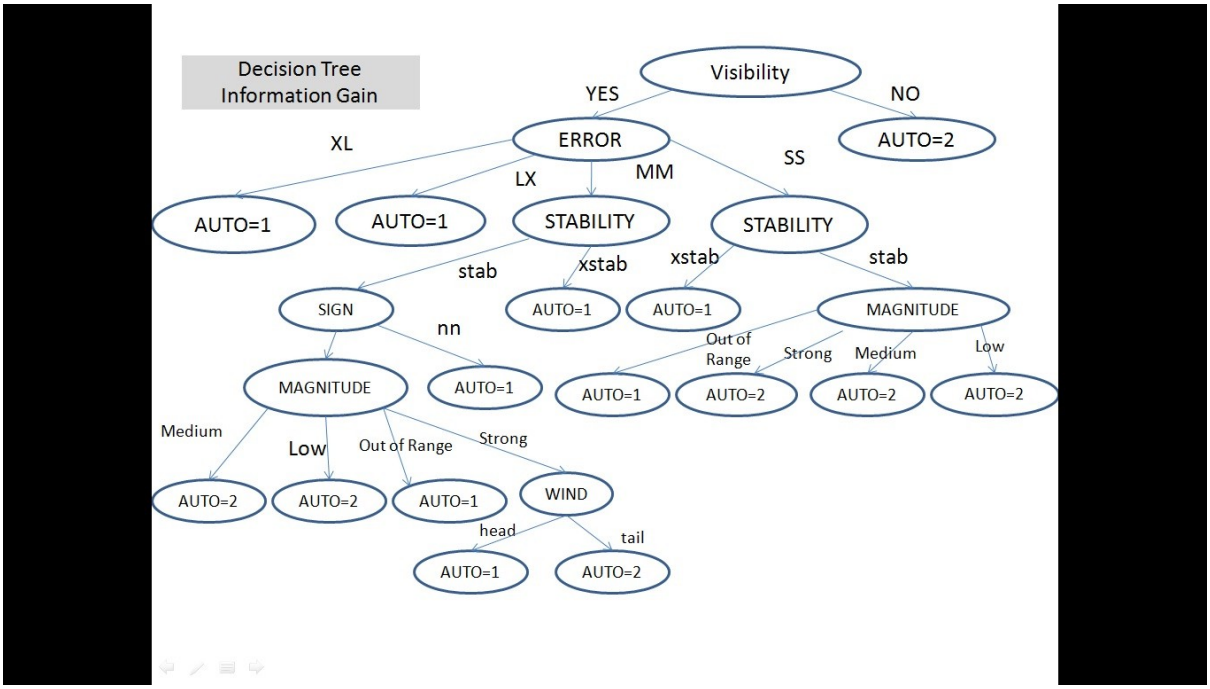


Figure : InfoGainDtree is the tree formed by using information gain for choosing the attributes.

b. DecisionTree using Gain Ratio:

The decision tree is printed in terms of equivalent set of rules as learned from the dataset.
It is seen that the decision tree is different form the one obtained using information gain. It is because the gain ratio discourages the selection of an attribute which has many uniformly distributed values. In this case it selects stability in depth 2 instead of ERROR when using Info Gain as the selection criteria.

(GainRatio is defined as = Information Gain/SplitInformtaion,
SplitInformtaion+=-|Si|/|S|LOG(Si/S) (i=1 to c)
At each step the parameter with the highest gain ratio is selected.

Tree using Gain Ratio(Trace)

```
if( VISIBILITY == "yes") {
        if( STABILITY == "stab") {
                if( ERROR == "XL") {
```

```
                        AUTO = "1";
                } else if( ERROR == "LX") {
                        AUTO = "1";
                } else if( ERROR == "MM") {
                        if( SIGN == "pp") {
                                if( MAGNITUDE == "Medium") {
                                        AUTO = "2";
                                } else if( MAGNITUDE == "Strong") {
                                        if( WIND == "head") {
                                                AUTO = "1";
                                        } else if( WIND == "tail") {
                                                AUTO = "2";
                                        }
                                } else    if( MAGNITUDE == "OutOfRange") {
                                                AUTO = "1";
                                } else if( MAGNITUDE == "Low") {
                                        AUTO = "2";
                                }
                        } else if( SIGN == "nn") {
                                AUTO = "1";
                        }
                } else if( ERROR == "SS") {
                        if( MAGNITUDE == "Medium") {
                                AUTO = "2";
                        } else if( MAGNITUDE == "Strong") {
                                AUTO = "2";
                        } else if( MAGNITUDE == "OutOfRange") {
                                AUTO = "1";
                        } else    if( MAGNITUDE == "Low") {
                                AUTO = "2";
                        }
                }
        } else if( STABILITY == "xstab") {
                AUTO = "1";
        }
} else if( VISIBILITY == "no") {
        AUTO = "2";
}
```

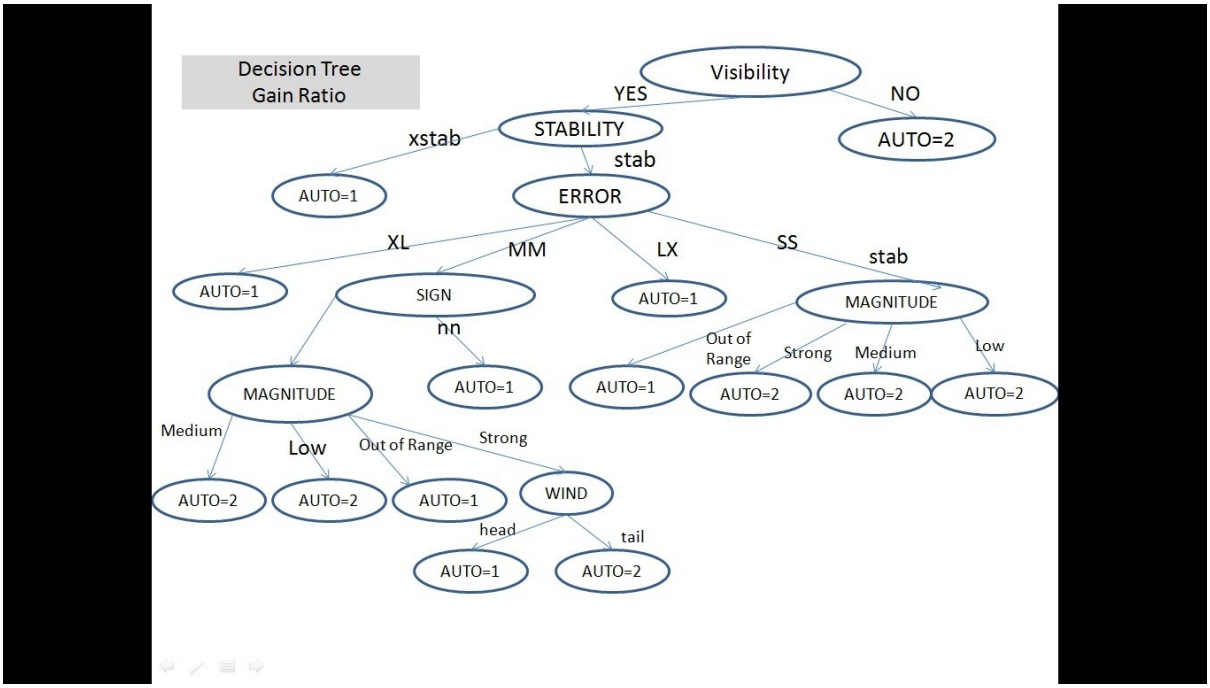


Figure : GainRatioDtree is the tree formed by using gain ratio for choosing the attributes

How to run the files:

Please change the location of the dataset file(shuttle_ext_unique.dat) and execute the program. The code can be executed in eclipse by importing the file to any project and executing directly. Any standard java compilation and execution will do.

Files:

`DecTreeUsingInfoGain.java`
DecTreeUsingGainRatio.java

Thanks.
Jyotirmoy Sundi
.