Machine Learning

BITS F-464

Assignment #1

* * * * * * * *

Decision Tree, Random Forest

&

Boosting Techniques

* * * * * * * *

Submitted to:

Dr. N.L Bhanumurthy

(Dept. of Computer Science/ Information System)

* * * * * * * *

By

Bhavya Mishra : 2013B3A7657H

Mohit Jain : 2013B3A7675H

Manaswini Pandey : 2013B3A7719H

Ridam Jain : 2013B5A7841H

Decision Tree: -

Decision tree learning is done through the implementation of **ID3** (Decision Tree Learning) algorithm that works like a greedy algorithm to choose the best possible hypothesis among the set of hypothesis. It does so by selecting a feature that minimizes the entropy or maximizes the entropy gain at every step of feature selection and hence come up with an learned structure that can be used later to test the new hypothesis for results.

Implementaiton Details :

1. Reading Files and pre-processing:

Input is taken in the form of a .data/.txt file where each line corresponds to one of the data instance. Each line is earlier encoded numerically in the form a vector< vector <int > > namely encoded_input. Each vector in encoded_input contains a vector of integers of size 14(number of attributes), as per their values mapped to a category in the dataset provided and their answers stored in another vector <int> namely target_vectors. For example,

{ 39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K }

is encoded as **{39,6,77516,0,13,2,8,4,0,1,2174,0,40,0}** and their target_vector having value **0 (<=50).** All the variables values of the categories are mapped to a dictionary namely **attribute_list** in the code.

2. Missing values and continuous variables :

Modes (maximum occurring value) are calculated for all the attributes corresponds to **discrete attributes** and **medians** are calculated for all the **continuous attributes**. All the missing values for discrete values are replaced with their modes, while continuous values are replaced via their medians.

3. Algorithm used and Testing :

ID3 algorithm has been implemented with each node in the tree having a three variables **(int flag, int attribute, vector <treenode *> attr_types)** , where flag represents if answer is provided for the hypothesis at that node, attribute represents, if node is not the final node, then what attribute is to be checked and attr_types represents the value of the attribute. At every node, for **continuous data, median split** has been considered whereas for discrete nodes, their values are taken as it is. Further details of the implementation are described in comments as well.

For testing, each node, is first processed, similar to how input nodes are processed and encoded as well. Later on, is passed through the tree structure and hence final results are compared with the structure.

The **results and sensitivity analysis** are described in later sections.

Random Forest: -

Random forest is a set of decision tree developed in the previous algorithm. While developing each tree, **total input size by 3 (33% of the training dataset)** which are selected randomly is used to develop each tree with at each node, **5 attributes selected randomly** are used to consider the split.

Detailed description of the algorithm is given in the code file as well. For concluding the results, majority of the decision output from individual trees are used and output is declared. Sensitivity analysis and results are described in the next section.

Boosted Forest: -

Ada boosting technique is used for boosting the given set of trees in the random forest for gaining higher accuracy. The 20 super classifier are used in the algorithm out of the bowl of total classifiers and results were obtained. The details of the implementation are described in the beginning of the code. The algorithm used can be found through this this link **(http://www.inf.fu-berlin.de/inst/ag-ki/adaboost4.pdf)**

Results:

The details of the results are provided below for the input file as "adult.data" and test file as "adult.txt" (first line of test file is ignored)

- **Decision Tree:** Each node in the tree contains the value of the attribute used, the results (flag) and also the pointers to the child nodes with median split used for continuous variables.

- **Random Forest:** A total of 100 trees are created, with 5 attributes selected randomly at each node(not selected before at higher levels in the tree) for considering the information gain and the split.

- **Boosted Tree:** 20 Trees are selected from the above created forest (100 trees) and boosted using the ada boost algorithm.

The results for the time required and accuracy are provided below:

Results:

Technique	Accuracy	Time (sec)
ID3	72.2499	0.927
Random Forest(100 trees)	74.1171	3.113
Boosted Trees(100 trees-20 boosted trees)	76.7836	5.811

As can be seen, the accuracy over the steps in algorithm increases from 72% to 76% but time required to compute also increased.

Further Comparison has been done in random forest for number of trees and time required.

Random forest: Number of trees vs Accuracy vs Time taken:

Number of trees	Accuracy	Time (sec)
100	74.1171	3.8
150	75.0691	5.529
200	74.1232	7.195
250	74.2779	8.866
300	74.7997	11.156
350	75.4315	12.592
400	74.8726	14.135
450	74.332	16.385
500	74.504	17.686

As the number of trees increased, the accuracy increased but later becomes stagnant. This can be accompanied by the fact that as the number of trees increased, the number of duplications increased and hence accuracy doesn't increase. Though time increases linearly with number of attributes.

Boosted Forest: Number of trees vs Accuracy vs Time taken:

Forest of the Boosted trees (20 super classifiers per forest) are compared with the number of total trees in the forest and time taken to make each of the forest.

Number of trees	Accuracy	Time (sec)
100	76.3774	5.672
150	76.3835	8.718
200	76.3712	11.779
250	76.3835	14.786
300	76.4142	20.087
350	76.4142	21.777
400	76.449	27.856
450	76.4049	30.708
500	76.398	32.229

Boosted forest: Number of Boosted trees vs Accuracy vs Time taken:

For a set of 150 trees, different number of boosted trees are selected and accuracy is compared for different instances. The results are shown in the table below.

Number of Boosted Classifiers	Accuracy	Time (sec)
4	75.8246	7.046
8	76.2424	7.335
12	76.3405	8.019
16	76.3712	8.146
20	76.3835	8.641
24	76.3835	9.139
28	76.3712	9.516
32	76.3774	9.659

As per the observed results, number of we can conclude that as the number of trees increases, accuracy will increase and reach upto a saturation level. Similar trend has been observed with the other variables compared above.

* * * * * * * * * * * * * * * * *

In order to run the files for different data sets (fname and testfile), or different number of trees (NOT) or super classifiers (NOSC) (in case of random forest and boosted forest), the data can be changed in the macro section of the .cpp files provided.

ID3 : ID3.cpp

Random Forest : Random_Forest.cpp

Boosted Forest : Boosted_forest.cpp

A separate file Partially_random_forest.cpp is present which is a varied form of a random forest algorithm where at initial step, 5 attributes are selected for splits and at every split only the previously selected but not previously used attributes are used.

* * * * * * * * * * * * * * * * *