International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-2, December 2012

Implementation of Decision Tree

Aashoo Bais, Kavita Deshmukh, Manish Shrivastava

Abstract— Data mining is rich field of algorithms and data structures to arrange negotiate and navigate the information from the different source of data. There are various kind of mining approaches are developed and implemented to get the knowledge from the raw data. The application of this knowledge is used to enhance the research, organizational growth and others. The data and its complexity is increases day by day in an explosive manner, and due to these complexity there are is a need to discover patterns and knowledge from the large data set. The conventional algorithm that are used to mine the patterns from data are becomes less effective due to the complexity of data. Due to this required to introduce some performance study and improvements over the conventional model to get efficient and effective data modeling technique. In this paper we introduce a modification over the traditional algorithm ID3 and C4.5 to make capable the algorithms to work with large dataset with higher performance. Here we provide the implementation, performance analysis and conclusion after implementation of the work.

Keywords- data mining, modification, large datasets, performance issues, implementation, performance analysis.

I. INTRODUCTION

Data mining sometimes also called knowledge discovery is the process of analyzing data from different outlooks and summarizing it into useful information. Information that can be used to increase revenue, cuts costs, or both, Data mining allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Due to growth of computer applications and computer automation the field and application of data mining also grows with this growth. These domains may be medical research, software engineering, information and scientific calculations, organizational growth, sales and marketing and other progressive domain of business and economy. In these fields we are know that a large amount of data is available for analysis and discovery of knowledge for different aspects. During the implementation of the data mining application we see that the size of data set and data type of the data set is effect the performance of the system. Mean to say when we introduce a large data set over traditional algorithms the performance of the classification is decrease effectively.

Manuscript published on 30 December 2012.
***** Correspondence Author (s)
Aashoo Bais, MTech Student, Department Of Information Technology Lnct, Bhopal, (M.P.) India.
Kavita Deshmukh, Ast. Professor, Department Of Information Technology Lnct, Bhopal, (M.P.) India.
Manish Shrivastava, Head Of Department, Department Of Information Technology Lnct, Bhopal, (M.P.) India.

Thus our main aim is to design a decision tree algorithm that works with a hug data set and less affected with the size and type of the data.

In this section we provide the introduction of decision tree application and the remaining paper contains the literature survey, review, design and implementation of the data mining application, additionally here we include the experimental results, conclusion and relative future work.

II. BACKGROUND

A. *Related Work*

In the domain of data mining there are various tools, techniques and methodologies and developed and implemented. But there is too little literature and work is available for the large dataset classification or clustering. Due to study of research papers and articles we found the below given facts that are important to discuss:

1. Our main goal is achieve high performance results with decision tree algorithm.
2. The performance of decision trees are varies according to the size and type of data set.
3. Decision trees are transparent model of data mining and navigation and prediction is based on the training set.
4. If we make changes over training data each and every time new model is formed.
5. There are too few words done under missing fields or attributes.

B. *Advantages and disadvantages*

Decision trees offer advantages over other methods of analyzing alternatives. They are: [1]

- **Graphic**: You can represent decision alternatives, possible outcomes, and chance events schematically. The visual approach is particularly helpful in comprehending sequential decisions and outcome dependencies
- **Efficient:** You can quickly express complex alternatives clearly. You can easily modify a decision tree as new information becomes available. Set up a decision tree to compare how changing input values affect various decision alternatives. Standard decision tree notation is easy to adopt.
- **Revealing:** You can compare competing alternatives—even without complete information—in terms of risk and probable value. The Expected Value (EV) term combines relative investment costs, anticipated payoffs, and uncertainties into a single numerical value. The EV reveals the overall merits of competing alternatives.
- **Complementary:** You can use decision trees in conjunction with other project management tools. For example, the decision tree method can help evaluate project schedules.

Limitations: Among the major disadvantages of a decision tree analysis is its inherent limitations. The major limitations include: [2]

- Inadequacy in applying regression and predicting continuous values
- Possibility of spurious relationships
- Unsuitability for estimation of tasks to predict values of a continuous attribute
- Difficulty in representing functions such as parity or exponential size
- Possibility of duplication with the same sub-tree on different paths
- Limited to one output per attribute, and inability to represent tests that refer to two or more different objects

An understanding of the pros and cons of a decision tree analysis reveals that decision tree disadvantages negate much of the advantages, especially in large and complex trees, inhibiting its widespread application as a decision-making tool.

C. Literature Survey

In this section of our paper we include the previously derived test and experiment and conclusion that are found after implementation. In previous work we are already perform experiments with ID3 and SLIQ for getting the information about the decision tree and we compare their results with different performance parameters. Like accuracy, memory used model build time and others and we found the following facts.

1. Preprocessing is required to get the higher performance of the system.
2. In our study we found that the different system performance parameters are depends upon some factors like data set size and data set data types.
3. Some performance factors are independent from different above performance criteria.

III. IMPLEMENTATION

In this section we include the implementation and algorithm design for the new tree. This tree implementation is based on the above given facts. And there primary steps are given below.

In straight words we can say that required to design such a decision tree that provide the higher performance and consumes less resources like building time and memory uses. for that purpose we create a model that is preprocess the data first and then consumes with C4.5 decision tree.

C4.5 decision tree is an algorithm that is derived from the ID3 algorithm and their performance of the system is higher than ID3.

For the preprocessing we just do one thing that changes may be reflects the changes over tree building structure and their performance too.

Preprocessing:

1. Read all data from data set
2. Create groups of data for one target class. For example we have a data set with three classes A, B and C. then we creates three groups.
3. Apply one group data to design a tree and after that with B and in same way with C.

Changes in splitting function: to achieve better performance we make a small change over the C4.5 algorithm. C45 decision tree the pseudo code for that algorithm is described below.

INPUT: Experimental data set D which is showed by discrete value attributes.
OUTPUT: A decision tree T which is created by giving experimental dataset.
Create the node N;
If instance is belong to the same class
Then return node N as the leaf node and marked with CLASS C;
IF attribute List is null, THEN
Return the node N as the leaf node and signed with the most common CLASS;
Selecting the attribute with highest information gain in the attribute List, and signing the test_attribute;
Signing the node N as the test_attribute;
FOR the known value of each test_attribute to divide the samples;
Generating a new branch which is fit for the test_attribute= ai from node N;
Suppose that Ci is the set of test_attribute = ai in the samples;
IF Ci is null THEN
Adding a leaf node and signed with the most common CLASS;
ELSE we will add a leaf node return by the Generate_decision_tree.

IV. ALGORITHM PROCESSING

Input Data:

The data is raw in nature and found in unformatted way. But to work with the data model required to format data first this process also called the data preprocessing. Data pre-processing includes the different phases to achieve a well formatted and arranged data. Moreover it after processing the data can be categorized in to three main parts.

1. Data set with only numerical values
2. Data set with nominal values
3. Data set with both nominal and numerical values.

In this paper for use the data we use the ARFF data format that is available online to use for experiments of machine learning. ARFF also abbreviated as attribute relationship file format.

The Header of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types.

```
% 1. Title: Iris Plants Database
  %
  % 2. Sources:
  %      (a) Creator: R.A. Fisher
  %      (b) Donor: Michael Marshall
(MARSHALL%PLU@io.arc.nasa.gov)
  %      (c) Date: July, 1988
  %
  @RELATION iris
  @ATTRIBUTE sepallength  NUMERIC
  @ATTRIBUTE sepalwidth   NUMERIC
  @ATTRIBUTE petallength
```


Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

NUMERIC
@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

The **Data** of the ARFF file looks like the following:

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa

Out of algorithm:

The above data is processed with help of our proposed system and generate the tree in the below given manner.

middle-middle-square = b
| top-left-square = b
| | bottom-right-square = b: Nothing
| | bottom-right-square = o: negative
| | bottom-right-square = x: positive
| top-left-square = o
| | bottom-right-square = b: negative
| | bottom-right-square = o: negative
| | bottom-right-square = x
| | | top-right-square = b
| | | | bottom-left-square = b: Nothing
| | | | bottom-left-square = o: negative
| | | | bottom-left-square = x: positive
| | | top-right-square = o
| | | | top-middle-square = b: positive
| | | | top-middle-square = o: negative
| | | | top-middle-square = x

V. USER NAVIGATION

In this section we include the user interaction to the system, and the options available to navigate the system. The fig 1 shows the implementation screen of the system proposed and the selected algorithm is named as rain forest. There is a provision available for input data set the button labeled with start is contains the processing of selected algorithm in below dropdown box. And the results and performance parameters are provided in the right side of the screen. A text box given to input the number of folds which is used during the performance analysis of the algorithm. The left side screen provides the tree generated using algorithm.

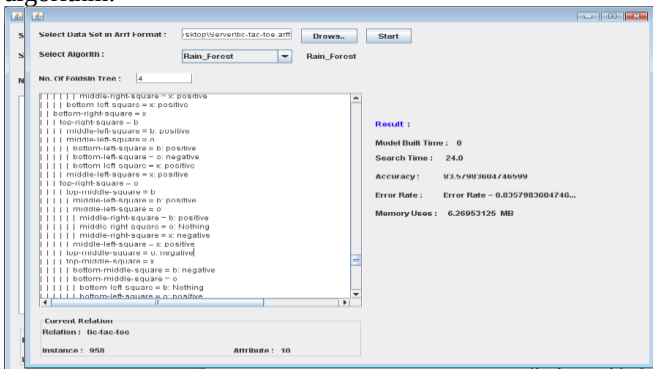

Fig 1. Shows implementation screen

And in last group box contains the formal information related to the selected data set.

A. Results Evaluation:

For the processing of results we use the N cross fold validation process to evaluate the results. In this process we produce the training data in a random sequence and evaluate the correct outputs and incorrect classified outputs. Suppose we select 100 input for the purpose of evaluation and the output of this is collected in the true counts and false count, now if we use 4 folds to test then all four cycle we get the results then total evaluation of input pattern is 400, and if we get 350 correct values and 50 wrong or incorrectly classified values then accuracy of the model is 350/4 = 87.5 and the error rate is 100- accuracy thus error rate is 12.5. or 50/4.

B. User defined classes and responsiblties

This section includes the classes and methods generated by us to implement this using our java code. Java is an open source framework and provides a rich and secure environment to develop the java applications. The implementation of different classes and methods are listed in two levels.

1. User Defined classes: listing and implementation of all classes is complex enough thus here we provide only 5 classes that are used

Class name	Responsibility
Form1	That is swing GUI used to provide user interface and the provision of selecting input data and process them
Rain_Forest	That is java class used to implement our concept of decision tree
NCrossValidation	This class is used to evaluate the data model build using provided data
ID3	Here we again implement ID3 for compare the performance of tree designed
SLIQ	This class contains the methods and functions to work with SLIQ.

2. Method and signature: this section includes the used functions and methods that are implemented for process the given data.

Method	Signature
MakeTree()	This function is used to mount tree structure from supplied data
EvaluateModel	This is a function implemented under the NCrossValidation class and used to evaluate error rate and accuracy
DoClassification()	This function is written under Form1 and used to create a sequence for call other implemented classes and functions
ComputeInfoGain()	Used by class Id3 to calculate the information gain
IsNumeric()	This method is used get the data type of the selected data set

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

VI. RESULTS

After implementation of our algorithm here we provide the performance analysis and comparative performance of designed system.

Accuracy: as we discussed in previous sections we use the N cross validation technique to evaluate results the below given table id comparative results of accuracy.

Data set Size	ID3	SLIQ	Rain Forest
100	72.31	64.59	67.26
300	75.47	79.23	78.33
800	68.18	73.56	74.51
1500	72.38	78.29	77.03
2500	77.69	74.45	73.25

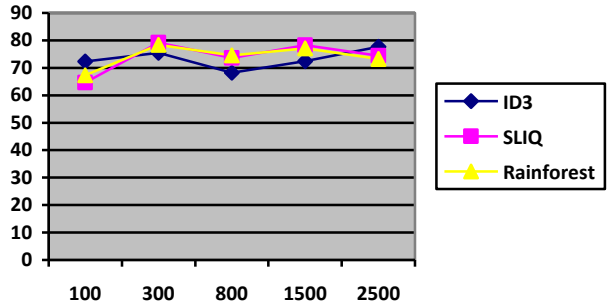

Fig 2 shows the accuracy of the system

The above graph shows the accuracy of all three systems and we found that most of the time it provides the higher performance results thus we can say that our system perform well than conventional algorithm.

Memory used: memory used of our designed system is sometimes higher than both systems.

Data set Size	ID3	SLIQ	Rain Forest
100	33268	32329	32832
300	32749	33425	32789
800	34831	34968	35828
1500	34974	33627	34829
2500	35221	35282	34834

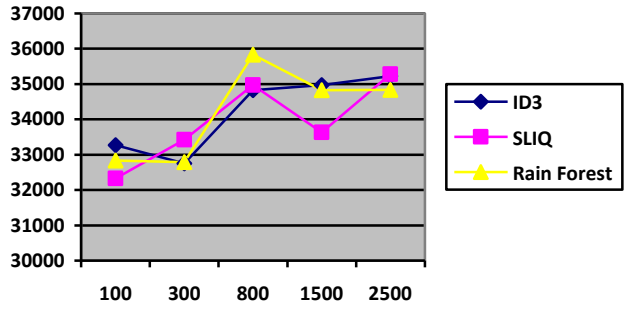

Fig 3 shows the memory uses of the system.

Memory is an expensive and a major factor in performance parameters the normal algorithms and our designed algorithm consumes similar memory.

Build time is defined as time required to prepare data model using provided data below given table and graph .

shows the model proposed consume less time to traditional models.

Data set Size	ID3	SLIQ	Rain Forest
100	2.54	3.21	2.26
300	4.27	4.23	3.34
800	6.19	6.54	5.1
1500	13.84	12.25	12.03
2500	17.90	17.52	16.57

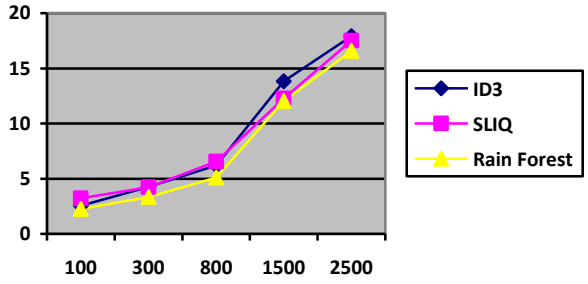

Fig 4 shows the build time

Search time of the system is too little values and it is remain constant in about all times, thus there are no need discuss more additionally one other factor named error rate is just opposite of the accuracy thus their graphical representations is not required.

VI. CONCLUSION AND FUTURE WORK

By the experimental results we found that the following conclusion for the system.

1. The implementation of decision tree is successfully complete
2. Designed modifications is less time consuming for large data set
3. Perform better than previously designed model.

But this implementation is not complete solve our problem to obtain the tree with higher performance, thus required to study in more depth and exploration of new dimensions are required to enhance decision tree model.

REFERENCES

[1] A Primer for Decision-making Professionals, By Rafael Olivas 2007, Rev. 5, 04/05/07.

[2] Disadvantages to Using Decision Trees written by: N Nayab • edited by: Jean Scheid • updated: 2/9/2011

[3] Rain Forest-A Framework for Fast Decision Tree Construction of Large Datasets Johannes Gehrk e Raghu Rama Krishnan Venkatesh Ganti Department of Computer Sciences, University of Wisconsin-Madison johannes,raghu,vganti @cs.wisc.edu

[4] S.P. Curram and J. Mingers. Neural networks, decision tree induction and discriminant analysis: an empirical comparison. Journal of the Operational Research Society, 45:440–450, 1994.

[5] Knowledge Extraction and Data Mining for the Competitive Electricity Auction Market M.-P. Cheong, Student Member, IEEE, G. B. Sheblé, Fellow, IEEE, and D. Berleant, Senior Member, IEEE

[6] Ian H. Witten and Eibe Frank, Data Mining: Practical machine learning tools with Java implementations,, San Fransisco: Morgan Kaufmann, 2000

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)