International Journal of Pure and Applied Mathematics
Volume 116 No. 22 2017, 19-29
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Classification Using Decision Tree Approach towards Information Retrieval Keywords Techniques and a Data Mining Implementation Using WEKA Data Set

[1]K.F. Bindhia, [2]Yellepeddi Vijayalakshmi, [3]P. Manimegalai and
[4]Suvanam Sasidhar Babu
[1]Dept. of Computer Science,
Bharathiar University, Coimbatore, India.
[2]Dept. of Computer Science and Engineering,
Karpagam University, Coimbatore, India.
[3]Dept. of Computer Science and Engineering,
Karpagam University, Coimbatore, India.
[4]Dept. of Computer Science and Engineering,
SNGCE, Kadayiruppu, Ernakulam Dt., India.

Abstract

Data Mining is an extraction tool for analyzing and retrieving hidden predictive information from large amount of data. The detected patterns give new subsets of data. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future trends. When the target values are used as discrete values, then we use classification tree. Decision tree classification with Waikato Environment for Knowledge Analysis (WEKA) is the simplest way to mining information from huge database. My paper includes the process of WEKA analysis by taking a data set as an example, step by step process of WEKA execution of that data set on different tree algorithms, selection of attributes to be mined and comparison with Knowledge Extraction and Evolutionary Learning. The following classification tree algorithms (AD Tree, Decision stump, NB Tree J48, Random forest, CART,) are used by WEKA for prediction. By comparing the accuracy and correctly classified attributes suitable decision can be figure doubt.

Key Words: Decision tree, WEKA, dataset, attribute, giniindex, entropy, attribute, split criteria, classification.

1. Introduction

Data mining (DM) give emphasis on mining large amount of data [1]. It applies machine Learning and statistical methods in order to discover hidden information hence it is known to be knowledge mining. It's also knowledge extraction, data/pattern analysis, data dredging. As a rule, the Knowledge Discovery from Data KDD process involves the following steps: data cleaning, data Integration, data selection, transformation, data mining, pattern evaluation and knowledge presentation. Data mining functionalities are used to specify the kind of pattern to be found. Classification is a process of finding a model that describes and distinguishes data classes and concepts in order to predict the class of objects whose class label is unknown. The derived model should be represented in decision tree or neural networks.

The Classification process involves following steps:

- Create training dataset.
- Identify class attribute and classes.
- Identify useful attributes for classification (Relevance analysis).
- Learn a model using training examples in Training set.
- Use the model to classify the unknown data samples.

This paper presents the analysis of various decision tree classification algorithms [11] using WEKA [4]. In section 2 decision approach and the splitting method is specified as the tree expands on attribute. In section 3 the measures to select the best attribute is discussed. In section 4 the traditional decision tree method is pointed. In section 4, WEKA has been discussed, different decision tree algorithms for classification have been compared. Section5 and 6 presents implementation and results of the analysis. Section7representsconcludingremarks.

2. Decision Tree

Decision Tree induction is the learning from class labeled training tuples. In decision tree nodes represent the input values, the edges will point to all the possible moves, thus from node to leaf through the edge its giving the target values from which we can create classification to predict. This learning approach is to recursively divide the training data into buckets of homogeneous members through the most discriminative dividing criteria. The construction of tree does not require domain knowledge. During decision tree construction attribute selection measures are used o select the attribute that best partitions the tuple into distinct classes [1]. The measurement will be the entropy or gini index of the bucket. Each internal node denotes a test on a predictive attribute and each branch denotes an attribute value. A leaf node represents predicted classes or class distributions [8]. An unlabeled object is classified by starting at the topmost

(root) node of the tree, then travel sing the tree, based on the values of the predictive attributes in this object.
Discrete a1, a2,…aj split point (< or>)

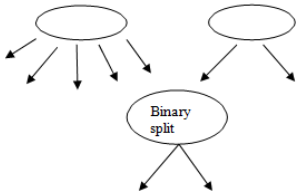


Figure 1: Recursive Algorithm for Building Decision Tree

Decision Tree implementations differ primarily along these axes:

1) The splitting criterion (i.e., how "variance" is calculated)
2) Whether it builds models for regression (continuous variables, e.g., a score) as well as classification (discrete variables, e.g., a class label).
3) Technique to eliminate/reduce over-fitting.
4) Whether it can handle incomplete data.

3. Attribute Selection Measures

To select the best split of attributes selection of attributes depends on the type and way to split. It can be discrete valued, continuous values and binary split. Two important measures are information gain or gain ratio. And gini index Information gain is the difference between the original information rrequirements. Let *pi* be the probability that an arbitrary tuple in D belongs to class Ci, it is estimated by

$$|Ci,D| / |D|$$

Expected information (entropy) needed to classify a tuple in D: *Info* (*D*) =

$$Info\ (D) =$$

Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|}$$

Information gained by branching on attribute A
$Gain(A)=Info(D)-InfoA(D)$

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$$

It gives the expected information required to classify a tuple from D based on partitioning by attribute A. The gain ratio is defined as

Gain Ratio (A) = Gain (A)/ Split Info A (D).

The attribute with the highest gain ratio is selected as the splitting attribute [1].

4. Traditional Method

During late 1970s Ross Quinlan developed decision tree algorithm for building decision trees based on concept learning. It was a bench mark for newer supervised learning algorithms. This uses a greedy approach in which tree are constructed in top –down recursive divide and conquer manner. A typical algorithm for building decision trees is given in figure 1. The algorithm begins with the original set X as the root node. for each unused attribute of the set X and calculates the information gain (IG). The formulas needed to calculate information gain along with the formula for calculating information gain is given above. The algorithm then chooses to split on the feature that has the highest information gain [11].

Function Build DecisionTree (Data,Lbels)

```
If all labels are same Then
    Return Leafnode for that label
Else
    Calculate Information Gain of all the features
    Choose the feature with highest information gain for splitting
    Left =  BuildDecisionTree(data withf=0,labelwithf=0)
    Right=  BuildDecisionTree(data withf=1,labelwithf=1)
    Return Tree(f,Left,Right)
Endif EndFunction
```

The Set X is then split by the feature obtained in the previous step to produce the subset of data depending on the value of feature. Partitioning stops on anyone of the following terminating condition like all of the tuples in partition D belong to same class or there were no remaining attributes on which it can be further partitioned and there are no tuples for a given branch that is D is empty.

5. Weka

The University of Waikato in New Zealand developed WEKA (Waikato Environment for Knowledge Analysis)[10] data mining software . Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka is an open source data mining tool it supports data mining algorithms and bagging and boosting. Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. Machine learning (ML) techniques and their application to real-world data mining problems can be done using weka. WEKA would not only afford a toolbox of learning algorithms, but also a framework inside which researchers could implement new algorithms without having to be concerned with supporting infrastructure for data manipulation and scheme evaluation. WEKA is open source software issued under General Public

License [5]. The data file normally used by Weka is in ARFF file for-mat, which consists of special tags to indicate different things in the data file foremost: attribute names, attribute types, and attribute values and the data.

The GUI allows us to try out different data preparation, transformation and modeling algorithms on data set. It allows running different algorithms in batch and compares the result. The buttons can be used to start the following applications; it's shown in Figure 2:

Explorer: It is the main graphical interface in WEKA for knowledge flow. It allows you to process large dataset an incremental manner. Once a dataset has been loaded, one of the other panels in the Explorer can be used to perform further analysis.

Experimenter: uses one classifier, one or more datasets, does classification or regression, then after cross validation or random split run the experiment evaluate and output the result Knowledge Flow: It presents dataflow .it handle data incrementally using classifier and updates on an instance by instance base.

Simple CLI: it's a text based command-line interface that allows direct execution of WEKA commands.



Figure 2: Weka GUI Chooser

6. Methods and Results

Various decision tree algorithms are used in classification. Different classes of tree classifiers in weka are given in table 1.

Table 1: Decision Tree Algorithm

Class	Description
ADTree	Alternating decision tree.
BFTree	Class for building a best- first decision tree classifier.

Decision Stump	Class for building and using a decision stump.
J48	Class for generating a pruned or unpruned C4.5 decision tree.
J48graft	Class for generating a grafted (pruned or unpruned) C4.5 decision tree.
LMT	Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves.
NBTree	Class for generating a decision tree with naive Bayes classifiers at the leaves.
Random Forest	Class for constructing a forest of random trees.
Random Tree	Class for constructing a tree that considers K randomly chosen attributes at each node.
Simple Cart	Class implementing minimal cost-complexity pruning.
User Classifier	Interactively classify through visual means.

The following table shows some of the decision tree algorithms which we choose to study. These algorithms consider binary, continuous or categorical data.

AD tree works on preconditions and input conditions to predict the outcome.J48 consider missing values. Decision stump check rules for branching.

CART and random forest are classification and regression based tree algorithms which handle numerical and categorical value, it also considers missing values. Table 2 shows detail NB tree works on naïve bayes classification procedure, Which create subset for all attributes an d create branches.

Table 2: Decision Tree Algorithm Characteristics

Decision tree	Split	Criteria	Branching
AD Tree	Multi way	Entropy	Precondition, condition and score 0and 1
BF Tree	Binary	Entropy or Gini index	Best first selection, maximum impurity reduction
Decision Stump	Binary	Entropy	1Rule generated decision
J48	Multi way, predictive model	Entropy	Cross validation, tree pruning un pruning, generate rules
NB Tree	Multi way	Entropy	Use naïve bayes classification
Random forest	Ensembl e method	Gini index	Random tree
Simple CART	Binary tree	Gini index	Classification and regression, extend to RF

Dataset

Here I am using the credit card German, its information needed for a credit card company to identify the profitable customers by analyzing the different branches on attributes .they want to find the most profitable customers for them. They are those customers whose pay the credit card repayments without due. And it can be analyzed from the most accurately separated count of positive attributes. the table shows the credit g data set with 20 attributes,1000 instances and 4 classes. The figure 3 below shows the features through WEKA

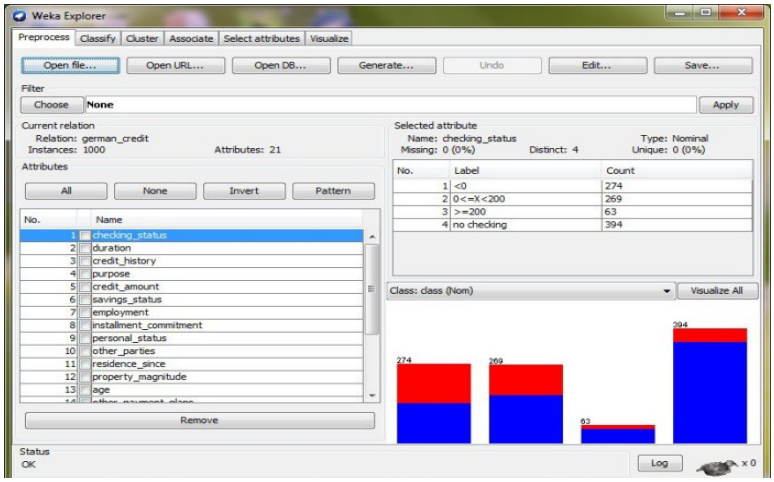


Figure 3: Weka Credit-g Data Set

Execution in WEKA

The following steps are needed to do a performance analysis through weka.

- Choose a data file, if it's in Excel then convert to attribute file format (arff).
- Preprocess the data ,use filter option to select or filter the attributes
- Take explorer option.
- Classify using different decision tree algorithm, as here we are focusing on that
- Compare the result for various decision trees, here we are considering the following decision tree algorithms.
- Visualize the data using tree and with different result parameter.

Result

Experiments were conducted under the framework of Weka to study the various kinds of Classification decision Algorithms on credit datasets. Here we compared various results measured by percentage accuracy. The environmental variables are same for each algorithm and dataset. Various parameters like TP rate, FP rate, precision, recall, time taken etc. TP rate is the true positive rate and the FP rate is the false alarming rate.

The ratio of predicted positive instances that were correct to the total number of false positive and true positive is precision. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database.

$$\text{Precision}=TP/TP+FP\times 100\% \quad (1)$$

$$\text{Recall} = TP/\ TP+FN\times 100\% \quad (2)$$

Where, TP, TN, FP, and FN are as represented in the confusion matrix in. The details of result is represented in Table 3 and Table 4.

Table 3: Results of CREDIT- g Data Set in Weka

Decision tree	Correctly classified instance	Incorrectly classified instance	Time taken	Relative absolute error
AD Tree	72.4%	27.6%	0.16	92.4%
BF Tree	73.3%	26.7%	1.04	78%
Decision Stump	70.0%	30.05	0.02	89%
J48	70.5%	29.5%	0.08	82%
NB Tree	75.3%	24.7%	0.62	74%
Random forest	73.6%	26.4%	0.07	81%
Simple CART	73.9%	26.1%	0.77	78%

Table 4: Results of Credit- g Data Set in Weka

Decision tree	TP	FP	Precision	Recall	F measure	ROC area
AD Tree	.724	.444	.709	.724	.713	.736
BF Tree	.733	.436	.718	.733	.722	.678
Decision Stump	.700	.7	.49	.7	.576	.654
J48	.705	.475	.687	.705	.692	.639
NB Tree	.753	.382	.744	.753	.747	.763
Random forest	.736	.454	.718	.736	.720	.746
Simple CART	.739	.424	.725	.739	.729	.713

7. Conclusion

As we studied the different decision tree algorithms we can came to conclusion that for credit data set NB tree is best suited for decision making as its giving 75.3% of correctly classified instance in 0.062 seconds referred in Figure 4.1000 instances were covered under 21 attributes. The confusion matrix and precision figure are given in figure below. In the same way we can analyze huge amount of data and any data set. In future we can develop a GUI for accepting or collecting raw data and analyzing and the important attributes can be classified using the tree diagram so that we can predict on data which may be a center point o take key decision. We can select the best classifier by analyzing and comparing the result

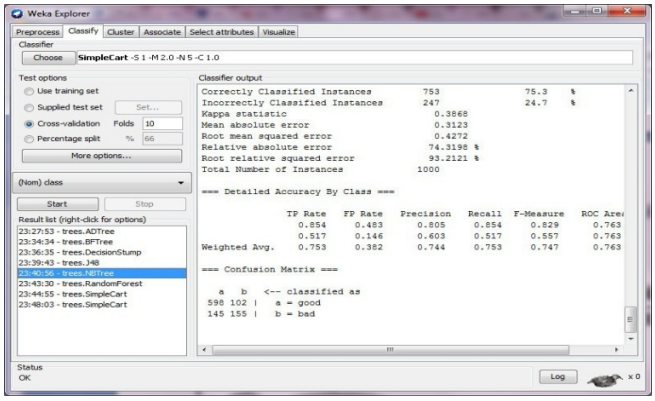


Figure 4: NB Tree Data in Weka

Acknowledgment

I would like to express my deepest thanks to all those who provided me the possibility to complete this paper. A special gratefulness gives to my guide, Dr.Suvanam Sasidhar Babu, Research Supervisor, Sree Narayana Gurukulam College of Engineering, whose contribution in stimulating suggestions and encouragement helped meto coordinate my work especially in writing this paper. Furthermore I would also like to acknowledge with much appreciation the crucial role of my family & friends, who gave the full effort in achieving the goal. I have to gratitude the guidance given by all for permission to use all the necessary equipment to complete the task. Last but not least, many thanks go to the god to giving me strength and courage to complete this paper.

References

[1] Daniel T. Larose, Data Mining Methods and Models, John Wiley & Sons, INC Publication, Hoboken, New Jersey (2006).

[2] Xindog Wu, Vipin Kumar, Top 10 Algorithms in Data Mining, Knowledge and Information Systems 14(1) (2008), 1-37.

[3] Andrew Secker, Matthew N. Davies, An Experimental Comparison of Classification Algorithms for the Hierarchical Prediction of Protein Function, Expert Update (the BCSSGAI) Magazine 9(3) (2007), 17-22.

[4] Han J., Kamber M., Data Mining: Concepts and Techniques, Morgan Kaufmann (2001).

[5] Ryan Potter, Comparison of Classification Algorithms Applied to Breast Cancer Diagnosis and Prognosis, Wiley Expert Systems 24(1) (2007), 17-31.

[6] Yoav Freund, Llew Mason, The Alternative Decision Tree Learning Algorithm, International Conference on Machine Learning (1999), 124-133.

[7] Singhal S., Jena M., A study on WEKA tool for data preprocessing, classification and clustering, International Journal of Innovative Technology and Exploring Engineering 2(6) (2013), 250-253.

[8] Peng W., Chen J., Zhou H., An Implementation Of ID3-Decision Tree Learning Algorithm, School of Computer Science & Engineering, University of New South Wales, Sydney, Australia.

[9] Wikipedia contributors, C4.5_algorithm, Wikipedia, The Free Encyclopedia, Wikimedia Foundation (2015).

[10] Wikipedia contributors, Random_tree, Wikipedia, The Free Encyclopedia, Wikimedia Foundation (2014).

[11] Osmar R.Z., Introduction to Data Mining, CMPUT690 Principles of Knowledge Discovery in Databases (1999).

[12] Gholap J., Performance tuning of J48 algorithm for prediction of soil fertility, Asian Journal of Computer Science and Information Technology 2(8) (2012).

[13] Anshul Goyal, Performance Comparison of Naïve Bayes and J48 Classification Algorithms, International Journal of Applied Engineering Research 7(11) (2012).

[14] Provost F., Fawcett T., Kohavi R., The case against accuracy estimation for comparing classifiers, 5th Int. In Conference on Machine Learning, San Francisco, Kaufman Morgan (1998).

[15] http://weka.sourceforge.net/doc.stable/weka/classifiers/trees/package-summary.html

[16] http://www.cs.ccsu.edu/~markov/weka-tutorial.pdf

[17] http://research.cs.queensu.ca/home/cisc333/tutorial/We ka.html https://www.youtube.com/results?search_query=weka

[18] http://www.ripublication.com/ijaer.htm..