

# Educational Data Mining by Using Neural Network

Nitya Upadhyay  
RITM  
Lucknow, India

**Abstract:** At the present time, the amount of data in educational database is increasing day by day. These data enclose the concealed information that can lift the student's performance. Among all classification algorithms, decision tree is most algorithm. Decision tree provides the more correct and relevant results which can be beneficial in improvement of learning outcomes of a student. The ID3, C4.5 and CART decision tree algorithms are already implemented on the data of students to anticipate their accomplishment. All three classification algorithm have a limitation that they all are used only for small So, for large database we are using a new algorithm i.e. SPRINT which removes all the memory restriction and accuracy arrives in other algorithms. It is fast and scalable than others because it can be implemented in both serial and parallel fashion good data replacement and load balancing. In this paper, we are representing a new SPRINT decision tree algorithm which will used to solve the problems of classification in educational data system.

Key words: Educational Data mining, Classification, WEKA

## 1. INTRODUCTION:

Data mining is an emergent and rising area of research and development, both in academic as well as in business. It is also called knowledge discovery in database (KDD) and is an emerging methodology used in educational field to get the required data and to find the hidden relationships helpful in decision making. It is basically a process of analysing data from different perspectives and summarizing it into useful information (ramachandram, 2010). Now a day, large quantities of data is being accumulated. Data mining can be used in various applications like banking, telecommunication industry, DNA analysis, Retail industry etc.

**Educational Data Mining:** It is concerned with developing methods for exploring the unique types of data that come from educational database and by using data mining techniques; we can predict student's academic performance and their behaviour towards education (yadav, 2012). As we know, large amount of data is stored in educational database; data mining is the process of discovering interesting knowledge from these large amounts of data stored in database, data warehouse or other information repositories:

- Regression
- Artificial intelligence
- Neural networks
- Decision trees
- Genetic algorithm
- Association rules etc.

These techniques allow the users to analyse data from different dimensions, categorize it and summarized the relationship, identified during the mining process (yadav, 2012). **Classification** is one of the most useful data mining techniques used for performance improvement in education sector. It is based on predefined knowledge of the objects used in grouping similar data objects together (baradhvaj, 2011). Classification has been identified as an important problem in the emerging field of data mining. It maps data into predefined groups of classes (kumar, 2011). Classification is an important problem in data mining. It has been studied extensively by the machine learning community as a possible solution to the knowledge acquisition or knowledge extraction problem. The input to the classifier construction algorithm is a training set of records, each of which is tagged with a class label. A set of attribute values defined each record. Attributes with discrete domains are referred to as categorical, while those with ordered domains are referred to as numeric. The goal is to induce a model or description for each class in terms of the attribute. The model is then used by the classifier to classify future records whose classes are unknown.

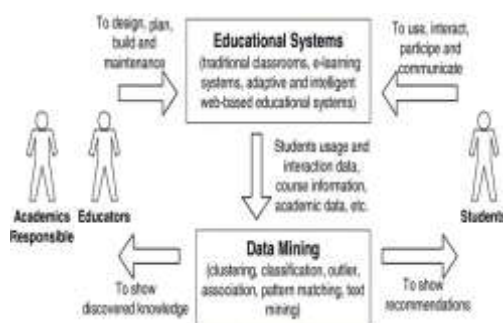


Figure 1.1- The cycle of applying data mining in educational system

Various algorithms and techniques are used for knowledge discovery from databases. These are as follows:-

- Classification
- Clustering

## 2. LITERATURE SURVEY:

A number of data mining techniques have already been done on educational data mining to improve the performance of students like Regression, Genetic algorithm, Bays classification, k-means clustering, associate rules, prediction etc. Data mining techniques can be used in educational field to enhance our understanding of learning process to focus on identifying, extracting and evaluating variables related to the learning process of students.

Decision tree algorithm can be implemented in a serial or parallel fashion based on the volume of data, memory space

available on the computer resource and scalability of the algorithm. The C4.5, ID3, CART decision tree algorithms are already applied on the data of students to predict their performance. But these are useful for only that data set whose training data set is small. These algorithms are explained below:-

#### • ID3

**Iterative Dichotomiser 3** is a decision tree algorithm introduced in 1986 by Quinlan Ross. It is based on Hunt's algorithm. ID3 uses information gain measure to choose the splitting attribute. It only accepts categorical attributes in building a tree model. It does not give accurate result when there is noise and it is serially implemented. Thus an intensive pre-processing of data is carried out before building a decision tree model with ID3 (verma, 2012). To find an optimal way to classify a learning set, what we need to do is to minimize the questions asked.

#### • C4.5

It is an improvement of ID3 algorithm developed by Quinlan Ross in 1993. It is based on Hunt's algorithm and also like ID3, it is serially implemented. Pruning takes place in C4.5 by replacing the internal node with a leaf node thereby reducing the error rate. It accepts both continuous and categorical attributes in building the decision tree. It has an enhanced method of tree pruning that reduces misclassification errors due to noise and too many details in the training data set. Like ID3 the data is sorted at every node of the tree in order to determine the best splitting attribute. It uses gain ratio impurity method to evaluate the splitting attribute (baradhwaj, 2011).

#### • CART

It stands for **classification and regression trees** and was introduced by Breiman in 1984. It builds both classifications and regression trees. The classification tree construction by CART is based on binary splitting of the attributes. It is also based on Hunt's algorithm and can be implemented serially. It uses gini index splitting measure in selecting the splitting attribute. CART is unique from other Hunt's based algorithm as it is also used for regression analysis with the help of the regression trees (baradhwaj, 2011). The regression analysis feature is used in forecasting a dependent variable given a set of predictor variables over a given period of time. It uses many single-variable splitting criteria like gini index, sym gini etc and one multi-variable in determining the best split point and data is stored at every node to determine the best splitting point. The linear combination splitting criteria is used during regression analysis.

#### • SLIQ

It stands for **supervised learning in ques**. It was introduced by Mehta et al (1996). It is fast scalable decision tree algorithm that can be implemented in serial and parallel pattern. It is not based on HUNT'S Algorithm for decision tree classification. It partitions a training data set recursively using breadth-first greedy strategy that is integrated with pre-sorting technique during the tree building phase. The first technique used in SLIQ is to implement a scheme that eliminates the need to sort the data at each node of the decision tree. In building a decision tree model SLIQ handles both numeric and categorical attributes (Rissanen, 2010). Sorting of data is required to find the split for numeric attributes.

#### • PUBLIC

It stands for pruning and building integrated in classification. Public is a decision tree classifier that during the growing phase, first determines if a node will be pruned during the following pruning phase, and stops expanding such nodes. Hence, PUBLIC integrates the pruning phase into the building phase instead of performing them one after the other. Traditional decision tree classifiers such as ID3, C4.5 and CART generally construct a decision tree in two distinct phases. In the first building phase, a decision tree is first built by repeatedly scanning database, while in the second pruning phase, nodes in the built tree are pruned to improve accuracy and prevent over fitting (Rastogi, 2000).

#### • Rainforest

It provides a framework for fast decision tree constructions of large datasets. In this algorithm, we have a unifying framework for decision tree classifiers that separates the scalability aspects of algorithms for constructing a decision tree from the central features that determine the quality of the tree. This generic algorithm is easy to instantiate with specific algorithms from the literature (including C4.5, CART, CHAID, ID3 and extensions, SLIQ, Sprint and QUEST).

Rainforest is a general framework which is used to close the gap between the limitations to main memory datasets of algorithms in the machine learning and statistics literature and the scalability requirements of a data mining environment (Gehrke, 2010).

#### • SPRINT algorithm

It stands for **Scalable Parallelizable Induction of decision tree** algorithm. It was introduced by Shafer et al in 1996. It is fast, scalable decision tree classifier. It is not based on Hunt's algorithm in constructing the decision tree, rather it partitions the training data set recursively using breadth-first greedy technique until each partition belong to the same leaf node or class. It can be implemented in both serial and parallel pattern for good data placement and load balancing (baradhwaj, 2011).

Sprint algorithm is designed to be easily parallelized, allowing many processors to work together to build a single consistent model. This parallelization exhibits excellent scalability to the users.

It provides excellent speedup, size up and scale up properties. The combination of these properties or characteristics makes Sprint an ideal tool for data mining.

#### Algorithm:-

- Partition (data S)
- If (all points in S are of the same class) then
- Return;
- For each attribute A do evaluate splits on attribute A;
- Use best split found to partition S into S1 & S2;
- Partition (S1);
- Partition (S2);
- Initial call: partition (Training data)

There are 2 major issues that have critical performance implications in the tree-growth phase:

1. How to find split points that define node tests.
2. Having chosen a split point, how to partition the data.

It uses two data structure: attribute list and histogram which is not memory resident making sprint suitable for large data sets, thus it removes all the data memory restrictions on data.

It handles both continuous and categorical attributes. Data structures of SPRINT are explained below:-

**Attribute list** - SPRINT initially creates an attribute list for each attribute in the data. Entries in these lists, which we call attribute records, consist of an attribute value, a class label and the index of the record from which these values were obtained. Initial list for continuous attributes are sorted by attribute value once when first created.

- **Histograms** – Two histograms are associated with each decision-tree node that is under consideration for splitting. These histograms denoted as  $C_{below}$  which maintain data that has been processed and  $C_{above}$  which maintain data that hasn't been processed. Categorical attributes also have a histogram associated with a node.

However, only one histogram is needed and it contains the class distribution for each value of the given attribute. We call this histogram a count matrix. SPRINT has also been designed to be easily parallelized. Measurements of this parallel implementation on a shared-nothing IBM POWER parallel system SP2. SPRINT has excellent scale up, speedup and size up properties. The combination of these characteristics makes SPRINT an ideal tool for data mining (Shafer).

### 3. PRESENT WORK:

Decision tree classification algorithm can be implemented in a serial or parallel fashion based on the volume of data, memory space available on the computer resource and scalability of the algorithm. The main disadvantages of serial decision tree algorithm (ID3, C4.5 and CART) are low classification accuracy when the training data is large. This problem is solved by SPRINT decision tree algorithm. In serial implementation of SPRINT, the training data set is recursively partitioned using breadth-first technique.

In this research work, the dataset of 300 students have been taking from B.tech. (Mechanical Engineering) by considering the input parameters as: - name, reg. no., their open elective

**Table 3.1: Example of attribute list of dataset**

Marks	Grade	Rid
72	Good	0
83	Good	1
78	Good	2
91	Good	3
65	Average	4
52	Average	5
43	Average	6

Table 3.2: Dataset after applying pre-sorting  
**After Pre-sorting:**

subject in 4<sup>th</sup> sem., midterm marks, end term marks, choice of Open elective subject, polling should be there? Yes or no, suggestion regarding polling: - if yes then why and if no then why? There are 9 OE subjects in B.tech. (ME) and because of limited sheets, most of the students do not get their own choice of subject. It could be effect on their performance in exam. So the output would come out to be how students are performing according to the choice of their preference.

### Objectives of Problem:

The objectives of the present investigation are framed so as to assist the low academic achievers in higher education and they are:-

- Identification of the choice of students in polling system which affects a student's Performance during academic career.
- Validation of the developed model for higher education students studying in various universities or institutions.
- Prediction of student's performance in their final exam.

In my proposed work, I am implementing SPRINT decision tree algorithm for improved classification accuracy and reduce misclassification errors and execution time. I have explained this algorithm and then apply serial implementation on it to find out the desired results. I am comparing it with other existing algorithms to find out which will be more efficient in terms of the accurately predicting the outcome of the student and time taken to derive the tree.

### Data structures:

#### 1. Attribute lists:

The initial list created from the testing set are associated with the root of the classification tree. As the tree is grown and nodes are split to create new children, the attribute lists belonging to each node are partitioned and associated with the children. The example of the attribute list is:

In sprint algorithm, Sorting of data is required to find the split for numeric attributes. It uses gini-splitting index for evaluate split. Sprint only sort data once at the beginning of the tree building phase by using different data structure. Each node has its own attribute list and to find the best split point for a node, we scan each of the node's attribute lists and evaluate splits based on that attribute.

**Histogram:** - Histograms are used to capture the class distribution of the attribute records at each node.

#### ➤ Performing the Split:

When the best split point has been found for a node, we execute the split by creating child nodes and dividing the attribute records between them. We can perform this by splitting the node's list into two as shown in figure 4. In our example, the attribute used in the winning split point is Marks. After this, we scan the list and apply the split test on it. Then we move the records to two new attribute list i.e. one for each new child. We have no test that we can apply to the attribute values for the remaining attribute lists of the node to decide how to divide the records. To solve this problem, we work with rids (Shafer).

As we partition the list of the splitting attribute i.e. marks, we insert rids of each record into a hash table to notify that the record was moved in which child. We can scan the list of the remaining attributes and probe the hash table after collected rids.

The output then tells us with which child to place the record. Splitting process is done in more than one step, if the hash table is large for memory.

➤ **Finding split points:**

During the process of making decision tree, the goal at each node is to determine the split point that best divides the dataset belonging to that node. The value of a split point depends upon how well it separates the classes. Many splitting have been proposed in the past to evaluate the goodness of the split. We need some function which can measure which questions provide the most balanced splitting. The information gain metric is such a function.

- **Measuring impurity:** - we have a data table that contains attributes and class of that attribute, we can measure homogeneity or heterogeneity of the table based on the classes. We can say that a table is pure or homogenous if it contains only a single class. If it contains several classes, then the table is impure or homogenous. There are so many indices to measure degree of impurity. Most common indices are entropy, gin index and classification error.

$$\text{Entropy} = \sum_j -p_j \log p_j$$

Entropy of a pure table is zero because the probability is 1 and  $\log(1) = 0$ . Entropy reaches maximum value when all classes in the table have equal probability. For a data set S

$$\text{Gini Index} = 1 - \sum p_j^2$$

In the above formula,  $P_j$  is the relative frequency of class  $j$  in  $S$ . If a split divides  $S$  into two subsets  $S_1$  and  $S_2$ , the index of the divided data Gini split( $S$ ) is given by the following formula:

$$\text{Gini split}(S) = n_1/n \text{ gini}(S_1) + n_2/n \text{ gini}(S_2)$$

The advantage of this index is that its calculation requires only the distribution of the class values in each of the partitions. To find the best split point for a node, we scan each of the node's attribute lists and evaluate splits based on that attribute.

The attribute containing the split point with the lowest value for the Gini index is then used to split the node. Gini index of a pure table consist of single class is zero because the probability is 1 and  $1-1^2=0$ . Similar to entropy, gini index also reaches maximum, value when all classes in the table have equal probability.

$$\text{Classification error} = 1 - \max \{P_j\}$$

Similar to entropy and Gini index, classification error index of a pure table is zero because the

probability is 1 and  $1-\max(1) = 0$ . The value of classification error index is always between 0 and 1. In fact the maximum Gini index for a given number of classes is always equal to the maximum of classification error index because for a number of classes  $n$ , we set probability is equal to  $p=1/N$ .

- **Splitting criteria:**

To determine the best attribute for a particular node in the tree we use the measure called information gain. The information gain,  $\text{gain}(S, A)$  of an attribute  $A$ , relative to a collection of examples  $S$ , is defined as

$$\text{Gain ratio} = \frac{\text{Gain}(S, A)}{\text{Split Information}}$$

The process of selecting a new attribute and partitioning the dataset is now repeated for each non terminal descendant node. Attributes that have been incorporated higher in the tree are excluded, so that any given attribute can appear at most once along any path.

#### 4. RESULTS:

The proposed SPRINT decision tree algorithm is implemented in WEKA tool. It contains a collection of visualization tools and algorithms for data analysis and predictive modelling, together with graphical user interfaces for easy access to this functionality. In this, data can be imported in any format like CSV, Arff, binary etc. data can also read from URL or database using SQL. There are various models for classifiers like Naïve Bayes, Decision Trees etc. We have used classifiers for our experiment purpose. In this, the classify panel allows the user to apply classification SPRINT decision tree and other existing algorithms to the data set estimate the accuracy of the resulting model.



Figure 4.1: Preview after data set imported in Weka

In figure 4.1, Red colour implies that these attributes belong to option A, Blue colour implies that these attributes belong to option B and the green colour means that these attributes belong to option C.

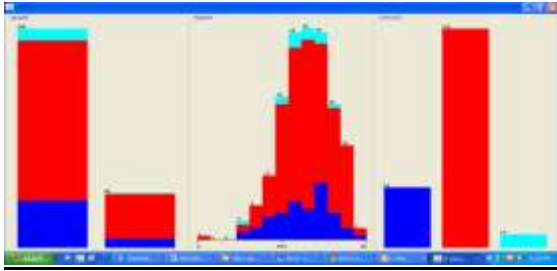


Figure 4.2: Visualizing all Attributes used in URL Classification

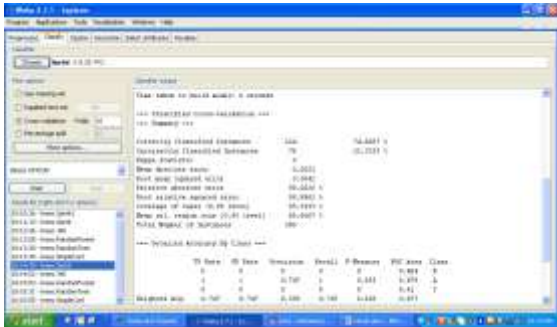


Figure 4.3: Classification by SPRINT Decision tree

Figure 4.3 shows the comparison among all attributes on parameters like accuracy, true positive rate and false positive rate. The definitions of these terms are explained below:-

- **Accuracy:** The accuracy is the proportion of total number of predictions that were correct.
- **True Positive Rate:** The true positive rate (TP) is the proportion of examples which are classified as class x, among all examples which truly have class x, i.e. how much part of the class are captured. It is equivalent to recall.
- **False positive Rate:** The false positive rate (FN) is the proportion of examples which are classified as class X, but belong to a different class, among all examples which are not of class X.
- **Precision:** It is the proportion of examples which truly have class x among all those which are classified as class X.
- **F-Measure:** It is a combined measure for precision and recall defined by the following formula: -

$$F\text{-Measure} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

#### 4.1 COMPARISON:

The following table 1 shows the comparison between the working of different decision algorithms on the basis of different parameters.

Table 4.1:- Parameter Comparison of Decision tree algorithms

ALGORITHMS	ID3 & C4.5	CART	SPRINT
Measure	Entropy info-gain	Gini diversity index	Gini Index
Procedure	Top-down decision tree construction	Constructs binary decision tree	Decision tree construction in a breadth first manner
Pruning	Pre-pruning using a single pass algorithm	Post-pruning based on cost-complexity measure	Post-pruning based on MDL principle

## 4.2 OUTPUT

The three decision trees as examples of predictive models obtained from the data set of 300 students by three machine learning algorithms: C4.5 decision tree algorithm, random tree algorithm and the new SPRINT decision tree algorithm. Table 4.2 shows the simulation result of each algorithm. From this table, we can see that a SPRINT algorithm has highest accuracy of 74.6667% compared to other algorithms. It also shows the time complexity in seconds of various classifiers to build the model for training data. By this experimental comparison, it is clear that SPRINT is the best algorithm among four as it is more accurate and less time consuming.

Algorithm	Correctly classified instances	Incorrectly classified instances	Execution Time (sec)	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
SPRINT	74.6667%	25.3333%	0	0.2651%	0.3641%	99.2622%	99.9962%
J48	74.6667%	25.3333%	0.06	0.2651%	0.3641%	99.2622%	99.9962%
Random forest	67.666%	32.3333%	0.16	0.2714%	0.4123%	101.6639 %	113.2245%
Random tree	64.6667%	35.3333%	0.02	0.271%	0.4179%	101.4793 %	114.7645%

The result can vary according to the machine on which we are analysing our experiment. This is due to the specifications of the machine like processor, RAM, ROM and its operating system. However it will not affect the accuracy of the algorithm used.

## 5. CONCLUSION:

The efficiency of all the decision tree algorithms can be analysed based on their accuracy and time taken to derive the tree. The main disadvantages of serial decision tree algorithm (ID3, C4.5 and CART) are low classification accuracy when the training data is large. This problem is solved by SPRINT decision tree algorithm. SPRINT removes all the memory restriction and accuracy problem which comes in other existing algorithms. It is fast and scalable than others because it can be implemented in both serial and parallel fashion for good data placement and load balancing.

In this work, SPRINT decision tree algorithm has been applied on the dataset of 300 students for predicting their performance in exam on the basis of their choice in polling system. This result help us to find that the students who are opted their own choice of subject are giving better results than others.

## 6. REFERENCES:

[1] Brijsh Kumar bhardwaj and Saurabh Pal “Data mining: a prediction for performance improvement using classification”, International journal of computer science an information security, vol. 9, no. 4, 2011.

[2] C.Romero and S.Ventra “Educational data mining: A survey from 1995 to 2005”, 2006 Elsevier ltd. All rights reserved. [www.elsevier.com/locate/eswa](http://www.elsevier.com/locate/eswa)

[3] Dorina kabakchieva,” Student performance prediction by using data mining classification algorithms”, International journal of computer science and management research, Vol 1 issue 4 November 2012

[4] Devi Prasad bhukya and S. Ramachandram,“Decision tree induction- An Approach for data classification using AVL –Tree”, International journal of computer and electrical engineering, Vol. 2, no. 4, August,2010.

[5] John shafer, Rakesh agrawal, Manish Mehta “SPRINT: A scalable parallel classifier for data mining” IBM Almaden Center, 650 Harry road, San Jose, CA 95120.