CHAPTER 6
EVALUATION OF A NEW INCREMENTAL CLASSIFICATION TREE ALGORITHM

This chapter presents the detailed description of the various approaches towards an Evaluation of a New Incremental Classification Tree [ICT] algorithm for mining high-speed data streams. It also presents the results being produced by different methods proposed. The method has been implemented and evaluated for their performance in various metrics.

Almost all the current computer applications are so designed to produce the huge volume of diverse data. Analysis and reorganization of such data require a huge computer memory than available. At present, new machine learning algorithms are being designed and developed to overcome these problems. Even though the incremental learning algorithms are effective in saving time and memory, they cannot be used to process larger datasets. Recently, the researchers are concentrating on the development of new algorithms, which are suitable for processing such large datasets. The algorithms are so designed that they enable the computer to learn from a single pass, require less memory usage and update incrementally at a later stage. Further, they should perform the data mining task at any time.

A high-speed data stream can be finite or infinite. A finite data stream can also be the large one but they appear from the finite source. In single pass method, the algorithm scans the data stream, linearly with the size of the data. This leads to faster mining of large databases than multiple passes at hand. In contrast to finite dataset the infinite dataset, come from known but endless sources, which comprise of
continuous data. The algorithm to process infinite dataset must be able to read the data quickly. If the algorithm is slow in processing the data that would result in loss of vital data.

6.1 Existing Methodology

The classification of the data stream is one of the important areas in the data mining. Therefore, the effective algorithms are required to take the data from temporary location into the permanent record. There are two typical approaches to the supervised machine learning; they are classification and regression.

6.1.1 Classification

The classification is mainly relevant with the class attribute as dependent variables; this can be divided into two levels: building and testing. The building model level could be used to estimate an output from learning algorithm and the testing model level estimate the quality of building model.

6.1.2 Regression

Regression is relevant with numerical attributes as its output. The different methods such as neural networks, decision tree, rule-based etc. are used for the classification. These methods are contrived to build classification model where distinct passes on the stored data are possible.

6.2 Proposed Methodology

6.2.1 Incremental Learning Tree

The incremental process does not require full loading data in one go; instead, it only requires some part of data to train the decision tree. When new data arrives, the count is updated and tests whether the data satisfies the required conditions. If all the conditions are satisfied the tree gets updated incrementally.
Figure 6.1 depicts the schematic representation of the proposed model where the system re-builds the classification model with the most recent data. By using the error rate as the guide to new data sets the frequency of model building and the memory size is adjusted over time. In this, the high-speed data is split into data sets using info-fuzzy techniques for classification tree model. It uses information theory to calculate the memory size. The main idea to change the memory size is based on the classification error rate. Each level of the tree represents only one attribute except the root node layer. The nodes represent different values of the attribute.

The process of constructing tree is determined if the split of an attribute would decrease the entropy or not. The measure used is mutual conditional information that assesses the dependency between the current input attribute under examination and the output attribute. At each stage, the algorithm chooses an attribute with the maximum mutual information and adds a layer. Each node represents the different value of this attribute. The iteration stops when there is no increase in the common information for any of the remaining attributes that have not been measured in the tree. In figure 6.1 the ‘model stability’ takes different decisions based on input data type. The drift data decrease the value of the tree while stable data is added incrementally upon arrival of each dataset of the stream. In this, the ‘model stability' takes the decision based on drift data and stable data.

A new model for online machine learning process of a high-speed data stream is proposed, to minimize the severe restrictions associated with the existing computer learning algorithms. Most of the existing models have three principal steps. In the first step, the system would create a model incrementally. In the second step, the time taken by the examples to complete a prescribed procedure with their arrival speed is computed. In the third and final step of the model the size of memory required for computation is predicted in advance. To overcome these restrictions, it is proposed a new data stream classification algorithm, where the data can be partitioned into the
stream of trees. In this algorithm, the new dataset can be updated with the existing tree. This algorithm, called incremental classification tree algorithm, is proved to be an excellent solution for processing larger data streams. In this research work, it presents the experimental results of the new algorithm proving that this method would eradicate the problems of the existing method.

![Figure 6.1 Block diagram of proposed ICT](image)

In figure 6.1, the high-speed data is split into data sets using info-fuzzy techniques for classification tree model. It uses information theory to calculate the memory size. The main idea to change the memory size is based on the classification error rate. Each level of the tree represents only one attribute except the root node layer. The nodes represent different values of the attribute. These concepts are implemented in section 6.2.2 as an evaluation of new ICT (Incremental Classification Tree) algorithm.
6.2.2 ICT Algorithm

Declare
   Respond node (or leaf node)
   Current node
   Decision node (or non-leaf node)
   Tree, subtree, root, node
   Attribute; $X_{new}, X_{old}$

Begin

Step 1 For each node; update $+$Ve and $-$Ve values

Step 2 If the node $\rightarrow +ve$ then respond node as‘+‘ else‘-‘

Otherwise

   a. If current node $\rightarrow$ respond node then
      Response node $\rightarrow$ decision node
   Otherwise

   b. If decision node $\rightarrow$ attribute
      i. Make attribute value as low
      ii. Dispose below the decision node tree
      iii. Update all the decision node from current node

Step 3 Grow the branch if needed

Step 4 If the tree $\rightarrow$ empty then expand the tree
   Otherwise

Step 5 If the tree $\rightarrow$ un expandable then do
   i. add the instances
   ii. expand it one level by testing
   iii. update the $-$ve instances during training

Step 6 If current node has the lowest value then
   i. Restructure the lowest value as root
   ii. Recursively update the current node
6.3 An Error Reduction Process

The different methods have been discussed for the efficient classification of ensembles. They have been evaluated for different parameters as follows: The overall error rate of the process is calculated by measuring the difference between the error rate during the training at one hand and the error rate during the model validation at the other hand. The following errors are calculated.

- Cross validation error
- Generalization error

6.3.1 Cross validation error

All the predicted errors from all $m$ stages are taken and added to calculate the cross-validation. Let the $m$ parts be $C_1$, $C_2$, $C_m$, where $C_m$ denotes the number of observations in $m$. The $n_m$ is the number of observations in part $m$. If $N$ is multiple of $m$ then

$$n_m = \frac{n}{m} \quad (6.1)$$

$$CV_m = \sum_{m=1}^{M} \frac{n_m}{n}MSE_m \quad (6.2)$$

Where MSE is defined as follows

$$MSE = E_m = \frac{\sum_{i=1}^{C_m} (y_i - \hat{y}_i)^2}{n_m} \quad (6.3)$$

Where

\(\hat{y}_i\) is used for observation $i$, obtained from the data.

Step 7 If attribute $X_{new}$ exists in root then

stop

Otherwise

i. Recursively pull $X_{new}$ to root

ii. Transpose $X_{new}$ as root and $X_{old}$ as subtree
- The MSE is obtained by fitting the value to the $K-1$ and calculates the error (MSE).
- It is a weighted average formula with $nm_n$ because each of the bends might not be of the same size.
- The cross-validation error is an average standard error and it gives us the validation estimate.

6.3.2 Generalization error

This calculates the accuracy of an algorithm to predict the outcome value for formerly unseen data. This minimizes the calculation time and avoids overfit. Overfitting occurs in case of complex datasets, with too many parameters to capture. Generalization error can be calculated using the equation. $G(\hat{\Theta})=P(y \neq \hat{\Theta}(x)) = E(i\neq y\neq \hat{\Theta}(x))$. Where $G$ is generalization error, $\hat{\Theta}$ is the classifier, $(x,y)$ is independently distributed according to the $P(x,y)$. Suppose if the generalization error rate is high then the prediction of the tree is expected to be uncertain. However, having low error rate does not guarantee good prediction for the given incoming data. Generally, this error is an optimistic estimate of the predictive error on new data.

6.4 Results and Discussion

All the methods are implemented in MATLAB-R2014B software with higher configuration system. The training data sets are used as bagged ensemble problems which can be generated an artificial dataset with 20 predictors. Consequently, various transactions such as independent test, loss test, validation tests and error reports are generated and their outputs are stored.

The new incremental classification tree based on ensemble classification algorithm has been implemented and produced effective results. The different data sets were taken to analyze the stable data and drift data. Based on the result it has been stored incrementally.
The figure 6.2 shows the snapshot of the incremental tree being generated. The tree has been generated in an incremental manner where each leaf has been generated based on the conditions and values of x.

Figure 6.2 Creating an Incremental Classification Tree

The Figure 6.3 shows the depth is achieved by the proposed algorithm in classification tree generation.

Figure 6.3 Tree depth diagram
Figure 6.4 Near-Optimal Trees

Figure 6.4 shows the snapshot of near optimal tree from the previous one generated. At each level the tree represents only one attribute that except the root node layer. The nodes represent different values of the attribute.

6.5 Summary

This chapter shows the evaluation of new Incremental Classification Tree (ICT) algorithm that has been demonstrated to be an important and novel method to mine the high speed streamed data and validate the new data thus predicted. The performance of the new algorithm is also compared with an existing popular algorithm. The results obtained revealed that this algorithm performs much faster than the existing algorithm.