EVALUATION OF A NEW INCREMENTAL CLASSIFICATION TREE ALGORITHM FOR MINING HIGH SPEED DATA STREAMS

N. Sivakumar[1] and S. Anbu[2]

[1]Research Scholar, Department of Computer Science and Engineering, St. Peter's University, Avadi, Chennai, Tamilnadu, India.
[2]Professor, Department of Computer Science and Engineering, St. Peter's College of Engineering and Technology, Avadi, Chennai, Tamilnadu, India.

ABSTRACT

Abstract—A new model for online machine learning process of 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 principle 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 we proposed this new data stream classification algorithm, where the data can be partitioned into stream of trees. In this algorithm, the new data set 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 paper, we present the experimental results of our new algorithm and prove that our method would eradicate the problems of the existing method.

KEYWORDS

Decision tree, Data mining, Machine learning, incremental decision tree, classification tree, data streams, High speed data streams.

1. INTRODUCTION

Almost of the current computer applications are so designed to produce huge volume of diverse data. Analysis and reorganization of such data require huge computer memory both RAM and ROM than available. At present, new machine learning algorithms are being designed and developed so as to overcome these problems. Even though the incremental learning algorithms are effective in saving time and memory, they cannot be used to process larger data sets.

Recently, the researchers are concentrating on development of new algorithms, which are suitable for processing such large data sets. The algorithms are so designed that they enable 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.

High speed data stream can be finite or infinite. A finite data stream can also be large one but come from finite source. In single pass method, the algorithm scan the data stream, linearly with size of the data. This leads to faster mining of large databases than multiple pass at hand.

DOI:10.5121/mlaij.2016.3302

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.

1.1. Data Stream Classification problem

The classification of data stream is one of the important areas in the in the data mining. Therefore, the effective algorithms are required to take the data from temporary location into permanent record.

There are two typical approaches in the supervised machine learning: 1. Classification: The classification is mainly relevant with 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 level.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 is possible.

2. LITERATURE SURVEY ON CLASSIFICATION METHODS

2.1. Ensemble based classification

This method is based on classic classification algorithms such as Naïve Bayesian classifier. It is also called decision tree model, which increase the quality of data in the output.[2]

2.2. VFDT (Very Fast Decision Tree):

VFDT splits the tree using the current best attribute that satisfies the Hoeffding bound. It has some drawbacks like it ties the attributes and tree grows out of memory.[1]

2.3. On-Demand classification

The mechanism of on-demand classification method consists of two principle components. The first component stores the summarized data the input data streams frequently. The second one continuously uses the summary statistics to execute the classification. The main motivation factor of this method is that it can be defined over a time possibility which depends on the nature of the concept drift and data progress.[3],[4].

2.4. ANNCAD Algorithm

The Adaptive Nearest Neighbor Classification Algorithm for Data stream (ANNCAD) is yet another classification algorithm is used currently. This algorithm is used for multiple decision data representation. The classification starts with seeking the records from nearest neighbor at better levels.[5]

3. THEORY AND MOTIVATION

3.1. Theory

Huge data: Aims at executing large volume of data where the data is recorded and labeled. The algorithm constructs incremental based classification tree.
Single pass: The data is loaded and computed only once in the large database. This can be applied on big data, even for unsecured problems

Execution Process: This can be calculated in this following ways

$$Accuracy = \frac{correct\ case}{total\ case} \quad (1)$$

$$Tree\ size = Number\ of\ rule = number\ of\ leaves \quad (2)$$

$$Speed\ of\ learning = time\ of\ Tree\ size \quad (3)$$

$$Tree\ model = \frac{memory\ cost}{memory\ size} \quad (4)$$

Weak Data: Incomplete data, missing data, noisy data and data with concept drift are considered weak data.

Classifier: Incremental decision tree applies a test and train operation. In this case, if any new data arrives that will travel from root to leaf. At the time of travelling, the node is trained incrementally.

3.2. Motivation

The incremental process does not require loading full data in one go; instead, it only required some part of data to train the decision tree. When new data is arrived the count is updated and tests weather the data satisfies the required conditions. If all the conditions are satisfied the tree gets updated incrementally.

4. DESIGNING OF NEW PROPOSED MODEL

The Figure 1 depicts 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 guide to new data sets the frequency of model building and the memory size is adjusted over time.

In the figure 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. These are described in the section VI.

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 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 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 Figure 1 the 'model stability' takes the decision based on drift data and stable data.

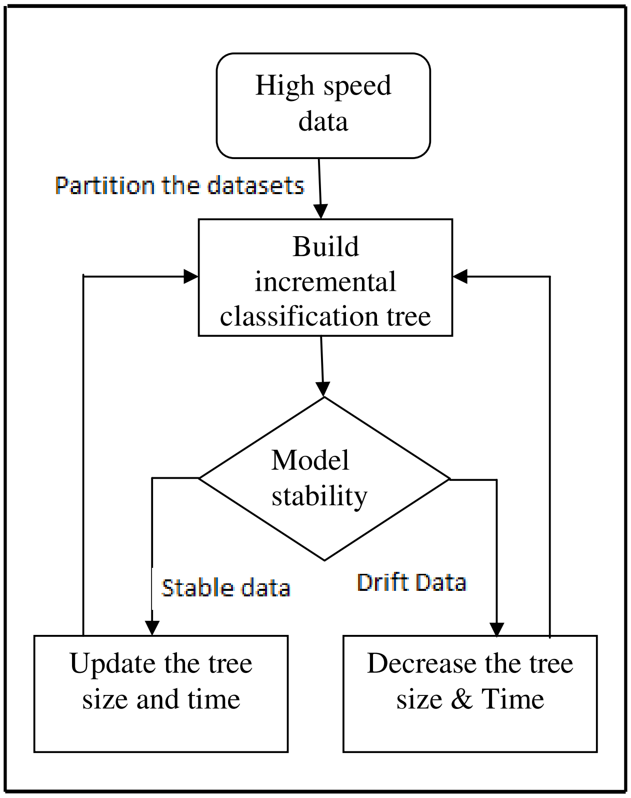

High speed data
Partition the datasets
Build incremental classification tree
Model stability
Stable data
Drift Data
Update the tree size and time
Decrease the tree size & Time

Figure 1: Proposed model

4.1. Advantages of our algorithm

- It makes liberal measure of data than Hoeffding bound used in VFDT.
- We can change the memory size of the model according to the classification error rate.
- Increase in the error rate indicates a high probability of a concept drift. The window size changes according to the value of this increase.
- The algorithm is ready to gain the knowledge for additional data or recent data.
- To train an existing classifier, the algorithm do not access the initial data again and again
- Keep up the knowledge which it has previously learned

- Arrival of new data is possible, while at the same time it greets and accommodates new data that is introduced over the period of time.
- It can learn from streamed data and undergo updating incrementally to get the advantages of newly arrived data.

5. AN INCREMENTAL CLASSIFICATION TREE ALGORITHM

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 →+ve then respond node as‘+’ else‘-‘
        Otherwise
        a.  If current node →respond node then
            Response node → decision node
        Otherwise
        b.  If decision node →attribute
          i.   Make attribute value as low
          ii.  Dispose below the decision node tree
        c.  Update all the decision node from current node
Step 3: Grow the branch if needed
Step 4: If the tree → empty then expand the tree
        Otherwise
Step 5: If the tree → 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 lowest value then
          i.    Restructure the lowest value as root
          ii.   Recursively update the current node
Step 7: If attribute $X_{new}$ is exist in root then
       stop
       Otherwise
          i.    Recursively pull $X_{new}$ to root
          ii.   Transpose $X_{new}$ as root and $X_{old}$ as subtree
End
```

6. AN ERROR REDUCTION PROCESS

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.

1. Cross validation error
2. Generalization error

6.1. Cross validation error

We take all the predicted errors from all *m* stages, we add them and that helps us to calculate the cross validation.

Let the *m* parts be C_1, C_2, C_m. where C_m denotes number of observations in *m*. The n_m is number of observations in part *m*. If *N* is multiple of *m* then

$$n_m = \frac{n}{m} \quad (5)$$

$$CV_{(m)} = \sum_{m=1}^{M} \frac{n_m}{n} MSE_m \quad (6)$$

Where MSE is defined as follows

$$MSE_m = \frac{\sum_{t\in C_m} (y_i - \hat{y}_i)^2}{n_m} \quad (7)$$

Where

- $\hat{y}_i$ is used for observation *i*, obtained from the data.
- The MSE is obtained by fitting the value to the *K-1*and we calculate the error (MSE).
- It is a weighted average formula with nm_n because each of the bends might not be of same size.

The cross-validation error is an average standard error and it gives us the validation estimate.

6.2. Generalization error

This calculates the accuracy an algorithm to predict outcome value for formerly unseen data. This generally 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 following equation.

$$G(\emptyset) = P\big(y \neq \emptyset(x)\big) = E\big(i\big(y \neq \emptyset(x)\big)\big) \quad (8)$$

Where *G* is generalization error, $\emptyset$ is classifier, *I* is the indicator, *(x,y)* is independently distributed according to the *P(x,y)*.

If the value of generalization error is high, the prediction of the tree is expected to be uncertain. However, having low error does not guarantee good prediction for the given incoming data. Generally this error is an optimistic estimate of the predictive error on new data.

7. EXPERIMENTAL RESULTS

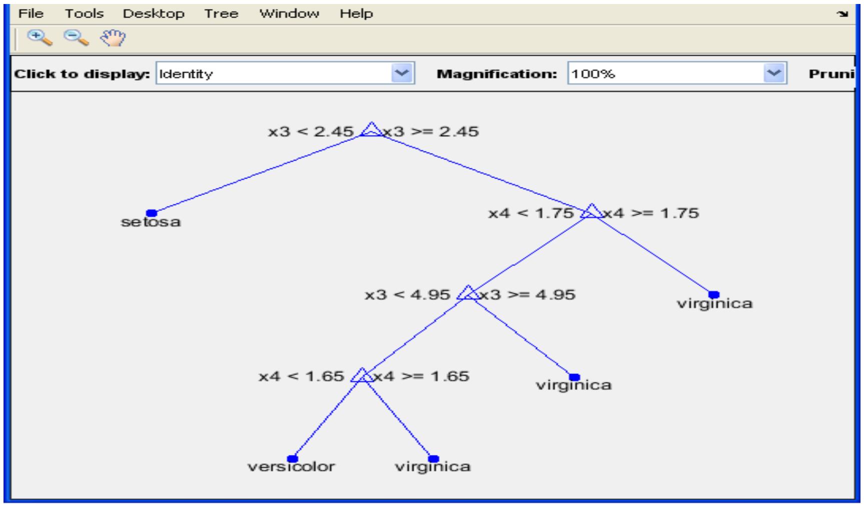
The experimental results given below were obtained using MatLab software and using a computer with high-end system configuration. All examples were used for testing the model before train them. The learning algorithm presented in this paper is proved to be better. Our algorithm can be controlled to execute the maximum tree size and maximum number of data sets could be allowed in our model. Our experiments furnished good tree classification accuracy, better learning speed and low memory usage for the same data stream as depicted in Figure 5.

7.1. Datasets

For our experiment we used two data sets namely *ionosphere* and *fisheriris*. These datasets are composed of more than 20000 examples as described below. In our experiments we used only the part of examples of the original data sets to perform our experiments fast.

7.2. Creating an Incremental classification tree

Figure 2 shows the graphic description of a classification tree.



Click to display: Identity

Magnification: 100%

Figure 2: Creating an incremental classification tree

In Figure 2, the tree is designed to classify data based on 2 predictors; they are x1 and x2. The prediction begins at the top node, represented by a triangle (Δ). If the first decision x1 is less than 0.5 the tree follows left side branch as a result the tree classifies the data as type 0. If, x1 is greater than 0.5, then the right side branch is followed to the lower-right triangle node. Subsequently, the tree checks if value of x2 is less than 0.5. If so, it follows again the left side branch and the tree sorts out the data as type 0. If not, then follow the right side branch and it sorts out the data as type 1.

7.3. Select an appropriate Tree Depth.

Figure 3 shows the cross authorized classification trees for the known datasets with minimum leaf occupation. This example demonstrates the way to control and select the depth of a decision tree. The best leaf size lies between 20 and 50 observations per leaf.

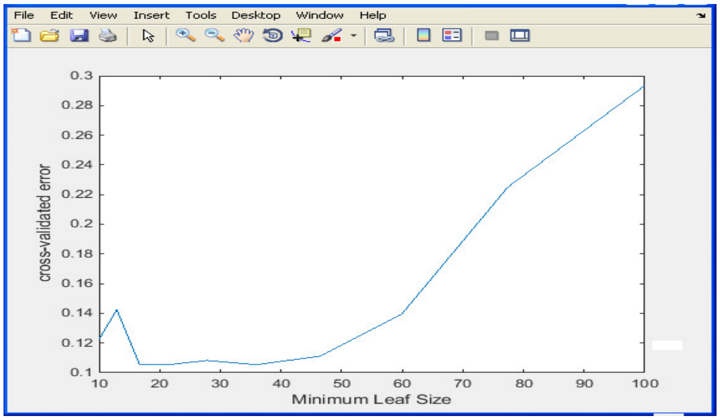
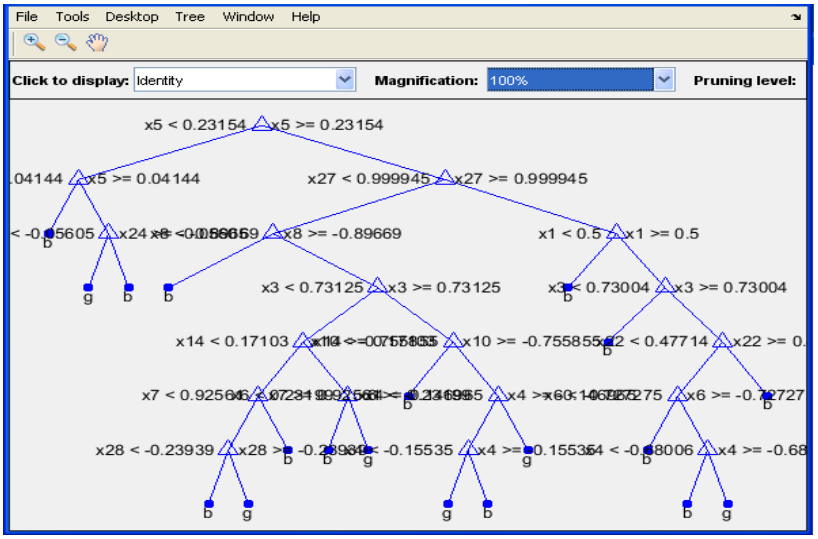


Figure 3: Tree depth diagram

7.4. Comparison between default tree and optimal tree



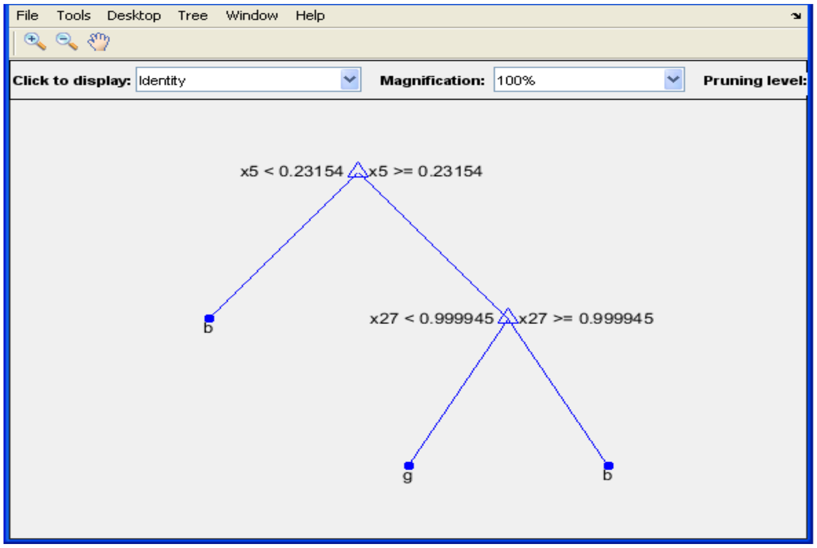
Click to display: Identity

Magnification: 100%

Pruning level:

Figure 4: Default tree

The default tree and optimum tree are depicted in Figure 4 and Figure 5 respectively. The near-optimal tree has at least 40 observations for each leaf, while the default tree has 10 observations for each parent node and 1 observation for each leaf node. A comparison between default tree and near-optimal tree shows that the default tree is larger in size than the near-optimal tree and it gives different accuracy than the cross validated error rate.



Click to display: Identity

Magnification: 100%

Pruning level:

Figure 5: Near-Optimal tree

It may also be noted that the near-optimal tree (Figure 5) is much smaller and gives a much higher generalization error. Yet it gives similar accuracy for cross-validated data (Figure 3).

8. CONCLUSION

In this paper the incremental classification tree algorithm has been demonstrated to be an important and novel method to mine the high speed streamed data and validate the new data thus predicted. Also, performance of our new algorithm was compared with an existing popular algorithm. The results are explained in detail in section 7. The results obtained revealed that our algorithm performs much faster than the existing algorithm.

REFERENCES

[1]. Domingo's P., and Hulten G. 'Mining high-speed data streams', in Proc. of 6th ACM SIGKDD International conference on Knowledge Discovery and Data Mining (KDD'00), ACM, New York, NY, USA, pp. 71-80, 2000.

[2]. Lazaridis I., and Mehrotra S, 'Capturing sensor-generated time series with quality guarantees'. In Proceedings of the 19th International Conference on Data Engineering. 2003.

[3]. AslamJ., Butler Z., Constantin F., CrespiV., CybenkoG. and RusD., 'Tracking a Moving Object with a Binary Sensor Network'. In Proceedings of ACM SenSy, 2003.

[4]. Babcock B., and Olston C. 'Distributed top-k monitoring'. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 2003.

[5]. Heinzelman,W.R, Chandrakasan.A & Balakrishnan.H 'Energy-Efficient Communication Protocol for Wireless Micro sensor Networks'. In Proceedings of the 33rd Hawaii Intl. Conf. on System Sciences, Volume 8, 2000.

[6]. Hang Yang, 'Solving Problems of Imperfect Data Streams by Incremental Decision Trees', journal of emerging technologies in web intelligence, vol. 5, no. 3, August 2013.

[7]. Prerana Gupta, Amit Thakkar, Amit Ganatra, ‘Comprehensive study on techniques of Incremental learning with decision trees for streamed data’, Journal of emerging technologies in web intelligence, vol. 5, NO. 3, August 2013.

[8]. Pallavi Kulkarni and Roshani Ade, ‘Incremental learning from Unbalanced data with concept class, Concept drift and missing features: A REVIEW’, International Journal of Data Mining & Knowledge Management Process, Vol.4, No.6, November 2014.

[9]. Junhui Wang and Xiaotong Shen, ‘Estimation of Generalization error: Random and Fixed inputs’, Statistica Sinica 16, 569-588, 2006.

[10]. Ponkiya Parita, Purnima Singh, ‘A Review on Tree Based Incremental Classification’, Ponkiya Parita et al, International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 306-309, 2014.