CAREER POINT UNIVERSITY HAMIRPUR

CPUH-Research Journal: **2015***, 1(2),* 28-32
ISSN (Online): 2455-6076
http://www.cpuh.in/academics/academic_journals.php

A Survey on Hoeffding Tree Stream Data Classification Algorithms

Arvind Kumar[1*], Parminder Kaur[2] and Pratibha Sharma[3]

[1, 2 & 3]*Department of Computer Science and Engineering,*

National Institute of Technology, Hamirpur-177005, India

[*] *Correspondance E-mail: arvind.pillania@gmail.com*

ABSTRACT: The large volume of data produced by real-time applications is difficult to organize and handle. Data stream algorithms extract information from volatile real-time applications' data and classify the network traffic. Stream data algorithms classify network data more efficiently than batch data mining algorithms. Stream classifier model works recursively every time new data arrives in the network. Decision tree classification using Hoeffding bound makes tree classification less time consuming. Streaming Random forest algorithm, an ensemble classifier consisting of many decision trees, works efficiently on large databases and estimates missing data effectively. CVFDT (Concept Adapting very fast decision tree) makes use of sliding window for data sets to provide consistency and offers ability to detect and respond if any changes occur in example generating process. In this paper, we compare Hoeffding tree, Streaming Random forest and CVFDT (Concept Adapting Very Fast Decision Tree) which are used for stream data classification.

Keywords: Decision tree learning; classification tree; regression tree; Hoeffding Bounds; Streaming Random forest and CVFDT.

INTRODUCTION: Data streams have received a lot of attention over the last decade, which is an important aspect in real-world applications like Credit card operations, sensor networking and banking services. Database transactions, telecommunication services generate logs and other forms of stream data [1]. The generated data by these applications is dynamic which is difficult to handle and organize. The volume of data, produced by real-time applications, which the stream comprises of, is large when compared to the limited storage of primary memory. Data stream mining algorithms extract information from volatile streaming data. Stream data algorithm sometimes cannot process the data more than once. So, the algorithms have to be designed such that they work effectively in that single pass only and check the concept drift. In this paper, we analysis the Random Forest, CVFDT which are based on Hoeffding tree and give an overview of decision tree learning. Decision tree learning creates a model (classification tree or regression tree) predicting the target variable value based on various input variables. Hoeffding tree uses Hoeffding bound for construction and analysis of decision tree. Hoeffding tree is capable of learning from massive data streams with assumption that the distribution generating examples do not change over time. Random forest uses a divide-and-conquer approach where a group of "weak learners" group together to form a "strong learner" [11].CVFDT (Concept Adapting very fast decision tree) algorithm uses windows systems, which makes use of sliding window of a number of data sets to provide consistency. CVFDT handles 'concept drift' very efficiently by creating alternative sub-tree to find best attribute at root node [2-3].

A. Difference between batch and stream classification: Data mining cannot store the complete data and is not available at the time of classification [4]. Also, it does not have sufficient amount of resources to create numerous data sets or patterns. Stream data classification has limited power and memory, which cannot handle and store gigantic volume of traffic as well. For the last few years, most of the applications have been working on stream data, widely used in Peer to Peer a (P2P) application which includes Bit Torrent, Emule, Kaaza etc., resulting in increased internet traffic. These applications increase the internet traffic by around 85% and create huge amounts of internet data. Several messenger-based applications like Yahoo and Google Talk, used by most people in peak hours, are again a major reason to rise in internet traffic. Some other most-used applications like web, e-mails and file transfer also increase the internet traffic data significantly. Traditional data mining algorithms work on the assumption that they will have sufficient resources to process particular data. This assumption does not have any chance in data stream mining [5] due to continuous evolvement of new data. Every Stream data mining algorithms should take less time to learn provided data with few amount of memory.

Table I: Problems in Data Stream Mining.

Batch data mining	Stream data mining
1. Require complete data set to create numerous pattern 2. In Batch data, data mining uses multiple passes technique 3. Require more time to access the specific data 4. No issue of 'concept drift'	1. Require only those data which is available when store the data 2. In Stream data, multiple passes not allow because of continuous arrival of new data. 3. Require less time to access the data. 4. Issue of 'concept drift'

B. CLASSIFICATION AND REGRESSION TREE: Decision tree learning uses decision tree as a predictive model mapping observations about an item to conclusions about the item's target value. Decision tree learning is a common method used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. These tree models are also called classification trees or regression trees.

However there is a significant difference in classification and regression.

- Regression and classification are both related to prediction, where regression predicts a value from a continuous set, whereas classification predicts the 'belonging' to the class
- In regression, the output variable takes continuous values, while the output variable takes class labels in classification.
- Classification trees have dependent variables that are categorical and unordered. Regression trees have dependent variables that are continuous values or ordered whole values. Regression means to predict the output value using training data.
- Classification means to group the output into a class. e.g. we use regression to predict the house price from training data and use classification to predict the type of tumor i.e. harmful or not harmful using training data.

Types of decision tree learning: In data mining, trees have additional categories:

- Classification tree analysis is when the predicted outcome is the class to which the data belongs [13].
- Regression tree analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).
- Classification and Regression Tree (CART) analysis is used to refer to both of the above procedures, first introduced by reference [6].
- A Random Forest classifier uses a number of decision trees, in order to improve the classification rate.

Formulae: Decision tree construction algorithms generally use top-down approach by choosing an attribute at each phase to split the given data set. This splitting is based on the best attribute chosen at each phase and the process keeps on repeating on each resultant subset recursively until the next splitting no longer adds value to the predictions. Different algorithms use different formulae for predicting "best attribute".

Here are some formulae which are applied to each candidate subset, and the resulting values are combined (e.g., averaged) to provide a measure of the quality of the split.

Gini impurity: Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it were randomly labelled according to the distribution of labels in the subset.

To compute Gini impurity for a set of items, suppose y takes on values in {1, 2, ..., n}, and let f_i = the fraction of items labelled with value i in the set.

$$I_G(f) = \sum_{i=1}^{n} f_i(1 - f_i) = \sum_{i=1}^{n}(f_i - f_i^2) = \sum_{i=1}^{n} f_i - \sum_{i=1}^{n} f_i^2 = 1 - \sum_{i=1}^{n} f_i^2 \quad (1)$$

Information gain: Information gain is based on the concept of entropy used in information theory by equation 2.

$$I_E(f) = -\sum_{i=1}^{n} f_i \log_2 f_i \quad (2)$$

C. MACHINE LEARNING STREAM ALGORITHMS: There are several algorithms available for data stream classification based on Hoeffding bound. Algorithms for classification of data streams based on data mining tasks are:

- Hoeffding tree algorithm works on decision tree.
- Random Forests is a Supervised and Unsupervised and works on Classification and Regression random forests.
- CVFDT (Concept-Adapting Very Fast Decision Tree) algorithm works on Hoeffding Bound decision tree.

D. HOEFFDINGTREE: Hoeffding tree uses the Hoeffding bound for construction and analysis of the decision tree. Hoeffding bounds used to decide the number of instances to be run in order to achieve a certain level of confidence.

A Hoeffding tree is capable of learning from massive data streams with assumption that the distribution generating examples do not change over time.

Classification problem is a set of training examples of the form (m, n), where 'm' is a vector of n attributes and n is a discrete class label. The objective is to produce a model n=f (m) so as to provide and predict the classes n for future examples m with high accuracy. Decision tree learning is a powerful technique in classification. Decision tree learning node has a check on attributes and each branch providing output of the check.

Step 1: Data is stored in the main memory and tree data structure with a single root node is initialized.
Step 2: Our main objective is to create decision tree learner which takes less time and reads data more efficiently. Filter down each and every training data inclemently to a suitable leaf.

Step 3: Each leaf node has enough data required to make decision about next step. This data at leaf node estimates the information gain when any attribute is split.

Step 4: We have to find the best attribute at a node and perform a test based on provided data to decide whether a particular attribute has produced better result than other attributes using Hoeffding bound.

Step 5: After applying a number of tests, the attribute, which provide better result than any other node, results in splitting the node for growth of tree.

Hoeffding tree algorithm compares attributes better than other algorithms. Also, memory consumption is less and delivers enhanced utilization with sampling of data. However, it spends lot of time in inspecting if ties occur.

E. STREAMING RANDOM FOREST: Random forest (or random forests) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees [6].The term came from random decision forests that was first proposed by [7-8]. The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Ho [7] and Amit and Geman [9] in order to construct a collection of decision trees with controlled variation. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.

Formation of tree involves various steps:

- Assuming S number of cases in training set, S cases sampled randomly with deviation from original data. Produced sample would then be treated as training set for growth of tree.
- At each node, p variables are to be selected randomly such that p <<P out of all the P input variables. Out of all the possible splits on p variables, the best one is used to split the node. During the growth of forest, the value p is taken to be constant.
- Each tree is grown to the largest extent possible. There is no pruning.

Streaming Random Forest learning Algorithm
Random forest algorithm [1] involves following steps:

Step 1: Assume S be the number of training cases, while P be the number of variables in the classifier.

Step 2: Let *p* be number of input variables used to determine decision at tree node where *p* has to be much less than P.

Step 3: Select training set for given tree by selecting *S* times with replacement from all *S* available training cases. By prediction of classes, the rest of the cases are used to estimate the tree error.

Step 4: For making a decision at a node, select p variables randomly for each tree node. Compute the best split in the training set based on *p* variables.

Step 5: Each tree is to be grown at its largest possible extent so that there is no further pruning.

The above algorithm works efficiently on large data bases which have the ability to manage large volumes of input variables without deletion. It provides estimation about the important variables in the classification. The algorithm is unbiased towards the estimation of generalized error during the forest formation. Random forest algorithm is also considered effectively estimating missing data and preserves accuracy with methods available for balancing errors in unbalanced class population data sets. Resultant forests can also be treated as input to the future data sets. It gives information about the relation between the variables and the classification. It works very efficiently for outlier detection, labeling the unsupervised clustering and data views.

F. CONCEPT ADAPTING VERY FAST DECISION TREE (CVFDT) ALGORITHM: CVFDT (Concept Adapting very fast decision tree) uses windows systems over VFDT, which delivers better speed and accuracy. It also offers ability to detect and respond if any changes occur in example generating process. Several systems with this ability [10], [12], CVFDT makes use of sliding window of a number of datasets to provide consistency. CVFDT continuously monitors the quality of new data and adjusts those that are no longer correct as compared to other existing systems, which needs to examine new model after arrival of new data. CVFDT increases counts for new data and decrements counts for oldest data in the window every time new data arrives. CVFDT handles 'concept drift' very efficiently by creating alternative sub-tree to find best attribute at root node. New best tree replaces old sub-tree every time which is consider more accurate on new data.

CVFDT (Concept Adapting VFDT) Algorithm

Step 1: Initialize HT (Hoeffding Tree) with a single node i.e. the root node. Let ALT to be an empty set of alternate trees for root node. W represents sliding windows which is empty at the start.

Step 2: Process the Examples from the stream uncertainly.

Step 3: For Each Example (m, n) in S, sort (m, n) to form an HT and every alternate tree of the nodes (m, n) passes through.

Step 4: Whenever a new example (m, n) arrives, it is added to the sliding window. Previous example is overlooked and (m, n) is fused into the present model. CVFDT regularly monitors HT and every single alternate tree searching for internal nodes whose adequate data demonstrate that some new attribute makes a superior test over the selected split attribute.

Step 5: CVFDT Grow

Step 6: Whenever a new best attribute is found at a node, Check Split Validity starts an alternate sub-tree. Philosophical Return HT.

There is continuous monitoring on the validity of previous decisions, which is handled by maintaining more than sufficient statistics at every node in Decision tree.

CONCLUSION: In this paper, we have discussed decision tree learning and data streaming. We have reviewed different classification algorithms such as Streaming Random forest and CVFDT. Both the algorithms use Hoeffding bound while splitting the decision tree. Hoeffding tree are better than batch trees in terms of learning time required. Streaming Random forest algorithm, an ensemble classifier consisting of many decision trees, uses a divide-and-conquer approach where a group of "weak learners" group together to form a "strong learner". CVFDT makes use of sliding window to provide consistency and offers ability to detect and respond if any changes occur. CVFDT handles 'concept drift' very efficiently by creating alternative sub-tree to find best attribute at root node. The decision trees made by these algorithms can also be extended in form of decision graphs, where we can use disjunction to join two more paths together using Minimum Message Length. The graphs allow unstated attributes to be learnt dynamically, which provides better accuracy without incurring much overhead.

REFERENCES:

1. Bifet, A., Holmes, G., Kirkby, R. and Pfahringer, B. 2011. Data Stream Mining a Practical Approach.
2. Symbal, A. T. 2004. The problem of concept drift: definitions and related work, Department of Computer Science, Trinity College Dublin, Ireland.
3. Brzezinski, 2010. Mining Data Streams With Concept Drift, Poznan University of Technology.
4. Aggarwal, C., Han, J., Wang, J., and Yu, P. S., 2004. On Demand Classification of Data Streams. In Proceedings of 2004 International Conference on Knowledge Discovery and Data Mining (KDD '04). Seattle, WA.
5. Agrawal, C. C. 2007. Data Streams: Models and Algorithms. Springer.
6. Breiman, L. 2001. Random Forests. *Machine Learning* 45: 5–32.
7. Ho, T. 1995. Random Decision Forest (http:/ / cm. bell-labs. com/ cm/ cs/ who/ tkh/ papers/ odt. pdf). 3rd Int'l Conf. on Document Analysis and Recognition. 278–282.
8. Ho, T. 1998. The Random Subspace Method for Constructing Decision Forests (http:/ / cm. bell-labs. com/ cm/ cs/ who/ tkh/ papers/ df.pdf). *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20: 832–844.
9. Amit, Y. and Geman, D. 1997. Shape quantization and recognition with randomized trees (http:/ / www. cis. jhu. edu/ publications/papers_in_database/ GEMAN/ shape. pdf). *Neural Computation* 9: 1545–1588.
10. Domingo"s, P. and Hulten, G. 2000. Mining High-Speed Data Streams. In Proceedings of the Association for Computing Machinery Sixth In-

ternational Conference on Knowledge Discovery and Data Mining.

11. Abdulsalam, H., Skillicorn, D. B., and Martin, P. 2007. Streaming random forests. In *Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International* 225-232.

12. Hulten, G., Spencer, L., and Domingos, P. 2001. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* 97-106.

13. Rokach, L. and Maimon, O. 2005. Top-down induction of decision trees classifiers-a survey". *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 35: 476–487.