# Data Stream Mining using Decision Tree Learning Algorithms

Ayyappan. M
IInd Year – M.E. CSE
Srinivasan Engineering College
Perambalur
Tamil Nadu, India

Nandhini. G
IInd Year - M.E. CSE
Srinivasan Engineering College
Perambalur
Tamil Nadu, India

Jayanthi. S
HOD/CSE
Srinivasan Engineering College
Perambalur
Tamil Nadu, India

**Abstract— Data stream mining is an active research area** Givt **recent years. The huge database involves handling and transaction of large amount of data in varied applications such as credit card transactions, monitoring networks, telecommunications, tweets and many more. The data streams are generated and changed automatically as an inherited interior mechanism while mining the data without anyone's knowledge. One of the difficult problem is how to effectively classify and improve the stream detection performance. Hence, decision tree learning process is chosen to identify the problem in data streams. In this project, the proposed system implementation is based on the decision tree of McDiarmid's Bound to address the inequalities projected in stream mining. The proposed work also uses the network data streams to analyze the attacks using splitting attribute with gain values. Focusing on classification problem and mining the data streams, the CART tree (binary decision tree) is implemented by splitting a node into two child nodes repeatedly, where the process again begins with the root node that contains the whole learning of samples. The constructed tree can be used for classification of new observation. The experimental result shows the better performance than the Hoeffding trees and demonstrates the effectiveness of proposed method.**

***Keywords***

***Data streams, decision trees, Hoeffding's bound, McDiarmid's bound, information gain, Gini index.***

## I. INTRODUCTION

Streaming machine learning can be interpreted as performing machine learning in streaming setting. Streaming setting is characterized by:

a) High data volume and rate, such as transactions logs in credit card and ATM operations, call log in telecommunication company, and i.e. Twitter tweet stream and Face book status update stream.

b) Unbounded, which means these data always arrive to our system and it won't be able to fit them in memory or disk for further analysis with the techniques. So , this characteristic implies that is limited to analyze the data once and there is little chance to revisit the data.

These characteristics, conventional machine learning algorithms (which require all the data to be available in memory) are not suitable to handle . The requirements to handle stream setting vary between methods. For classification algorithms, it need to adhere these following four requirements:

- Process an event at a time and inspect it only once (at most)
- Use limited amount of memory
- Work in limited amount of time
- Be ready to predict at any point

Another aspect of streaming machine learning is change detections i.e. since the input data is unbounded; it need to have mechanism to handle and react to changes in incoming data characteristics.

A major challenge in data stream classification, which deserves attention but has long assumed that the numbers of classes are fixed. However, in data streams, new classes may often appear. For example, a new kind of intrusion may appear in network traffic, or a new category of text may appear in a social text stream such as Twitter. When a new class emerges, traditional data stream classifiers misclassify the instances of the new class as one of the old classes. In other words, a traditional classifier is bound to misclassify any instance belonging to a new class, because the classifier has not been trained with that class. It is important to be able to proactively detect novel classes in data streams.

For example, in an intrusion detection application, it is important to detect and raise alerts for novel intrusions as early as possible, in order to allow for early remedial action and minimization of damage. A recurring class is a special and more common case of concept-evolution in data streams. It occurs when a class reappears after long disappearance from the stream.

Recurring classes, when unaddressed, create several undesirable effects. First, they increase the false alarm rate because when they reappear, novel class will be falsely identified, whereas such classes may observe normal representative behavior. Second, they also increase human

effort, in cases where the output of the classification is used by human analyst. In such cases, the analyst may have to spend extra effort in analyzing the afore-mentioned false alarms. Finally, "novel class detection" has additional computational effort, which is costlier than regular "classification" process. Novel class detection is major concept of concept evolution. In data stream classification assume that total no of classes is fixed but not be valid in a real streaming environment. When new class may evolve at any time.

The goal of the project is designed to function as a multi-class classifier for concept-drifting data streams, detect novel classes, and distinguish recurring classes from novel classes. Keep an ensemble of size, and also keep an auxiliary ensemble where at most models per class are stored. This auxiliary ensemble stores the classes in the form of classification models even after they disappear from the stream. Therefore, when a recurring class appears, it is detected by the auxiliary ensemble as recurrent.

This approach greatly reduces false alarm rate as well as the overall error. If, however, a completely new class appears in the stream, it is detected as *novel* by the auxiliary ensemble as well. This is the first work that addresses the recurring concept-evolution in data streams and class issue.

Proposed solution, which uses an auxiliary ensemble for recurring class detection, reduces overall classification error and false alarm rates. Second, this technique can be applied to detect periodic classes, such as classes that appear weekly, monthly, or yearly.It will be useful for better predicting and profiling the characteristics of a data stream. Finally, apply our technique on a number of real and synthetic datasets, and obtain superior performance over state-of-the-art techniques.

## II. RELATED WORK

In existing system use act miner applies an ensemble classification technique but used for limited labeled data problem and addressing the other three problem so reducing the cost. Act miner is extends from mine class. Act miner integrates with four major problem concept drift, concept evolution, novel class detection, limited labeled data instances. But in this technique dynamic feature set problem and multi label classification in data stream classification. Based on clustering methods for collecting potential novel instances so memory is required to store. Another disadvantage is that using clustering method first find centroid. And also incremental so time overhead occurs. And also not possible classify streamed data continuously. Because streamed data continuously come and classification become continuous task.

Mining streaming data is one of the recent challenges in data mining. Data streams are characterized by a large amount of data arriving at rapid rate and require efficient processing. Moreover, the data may come from non-stationary sources, where underlying data distribution changes over time. It causes modifications in the target concept definition, which is known as concept drift.

The main types of changes are usually divided into sudden or gradual concept drifts depending on the rate of changes. Classical static classifiers are incapable of adapting to concept drifts, because they were learned on the out-of-date examples. This is the reason why their predictions become less accurate with time. Some methods have already been proposed to deal with the concept drift problem.

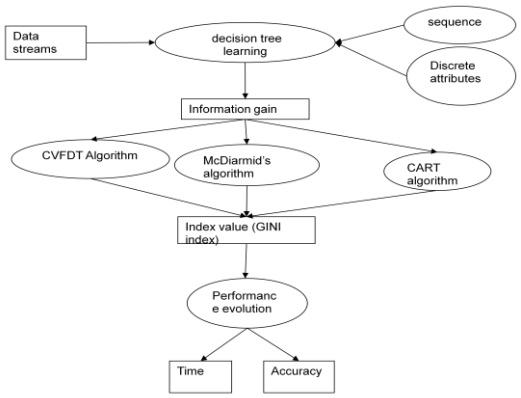
They can be divided into two main groups: trigger based and evolving. Trigger-based methods use a change detector to identify the occurrence of a change. If the change is detected, then the online classifier, connected with the detector, is re-trained. One of the most popular detectors is DDM described. On the other hand, evolving methods attempt to update their knowledge without explicit information whether the change occurred. An example of such methods is an adaptive ensemble.

This paper focuses mainly on block-based ensembles, which component classifiers are constructed on blocks (chunks) of training data. In general, a block-based approach operates in a way that when a new block is available, it is used for evaluation of already existing component and for creation of a new classifier. The new component usually replaces the worst one in the ensemble.

## III. OUR SYSTEM AND ASSUMPTIONS

Data streams are obtained continuously by applications such as sensor networks, credit card transactions and financial applications, generating a large volume of data every day. The classification task aims to build a model to describe classes of data. Traditional decision tree algorithms load the entire dataset into memory and build a static model to describe this data. Every time new samples of data arrive, the model must be rebuilt, considering the existing dataset and including the new data into it.

Traditional techniques for data mining require multiple scans of data to extract the information, which is not feasible for stream data. In the data stream context, incremental techniques are used to eliminate the need of rebuilding the model every time a new example arrives. In this project we describe a non parametric incremental decision tree algorithm called McDiarmid bound tree which is based on splitting attribute and proposes a decision tree model constructed from numerical data using statistics as a heuristic to decide when to perform for split the attribute and which attribute to use. In this work we intend to describe the behavior of Mcdiarmid inequality and CART tree for analyze the data streams containing noise.




### A. Upload Datasets

In this module upload the datasets. Datasets may be Network streams or Twitter data streams or Tele communication data streams. Because analyze the stream classification and extract the accuracy. The data streams are divided into datasets. Perform the preprocessing steps to remove the noise data from various datasets in data streams.

### B. Decision Tree Learning

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and the possible consequences, which includes chance event outcomes, utilities, and resource costs .The decision trees are commonly used in operations research, mainly in decision analysis, and to help in identifying a strategy most likely to reach a goal.

Decision tree learning is the construction of a decision tree from class-labeled training tuples. The decision trees are a flow-chart-like structures, where each internal (non-leaf) node denotes a test on an attributes, each branch represents the outcome of a test, and each terminal node holds a class label. Topmost node in a tree is the root node.

### C. Mcdiarmid Algorithm

In this module get the information gain from decision tree learning. As a measure of the effectiveness of an attribute in classifying the training data, the information gain was used. The interpretation is the expected reduction in entropy caused by splitting the examples according to this attribute. Then calculate the McDiarmid's bound for gini index. The Gini coefficient (also known as the Gini index or Gini ratio) is a measure of statistical dispersion intended to represent the distribution. The Gini coefficient can then be thought of as the ratio of the area that lies between the line of equality and the Lorenz curve over the total area under the line of equality.

### D. Cart Algorithm

Classification and regression trees (CART) is a non-parametric decision tree learning technique that produces either classification or regression tree, depending on whether the dependent variable is categorical or numerical, respectively.

Decision tree is formed by a collection of rules based on variables in the modeling data set:

Rules based on variables, the values are selected to get the best split to differentiate observations based on the dependent variable.

Once a rule is selected and splits a node into two, and the same process is applied to each "child" node (i.e. it is a recursive procedure)

Splitting stops when CART detects no further gain can be done, or some pre-set stopping rules are met. (Alternatively, data are split as much as possible and then the tree is later pruned.)

Each branch of the tree ends in a leaf node. And each observation falls into one and exactly one leaf node, and each terminal node is uniquely defined by a set of rules.

## IV. SYSTEM PRELIMINARIES

### A. Mcdiarmid Inequality

Classifications (and also machine learning in general) always assume that the training and testing data are available in the memory**,** since the associated algorithms can easily perform data analysis on it (such as perform multiple passes on training data to enhance machine learning models). The abundance of data and the need to process larger amounts of data have triggered machine learning development.

The classic classification algorithms are modified into scaled-up version, i.e. executing the algorithm in multiple machines with message-passing or expanding the limited memory with temporary external storage.

The McDiarmid tree algorithm is a state-of-the-art method for inducing decision trees from data streams. The aim of this project is to improve this algorithm. To measure the improvement, a comprehensive framework for evaluating the performance of data stream algorithms is developed.

In the framework memory size is fixed in order to simulate realistic application scenario. In order to simulate continuous operation, the classes of synthetic data are generated providing an evaluation on a large scale.

### B. Cart Algorithm

The basic idea of tree growing is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". In this algorithm, only univariate splits are considered. This means, each split depends on the value of only one predictor variable. The all possible splits consist of possible splits of each predictors. If X is a nominal categorical variable of I categories, there are $2I-1-1$ possible splits for the predictor. If X is a ordinal categorical or continuous variable with K different values, there are $K-1$ different splits on X. The tree is grown starting from the root node by repeatedly using the following steps on each node.

a) Non-parametric technique, by using the methodology of tree building.
b) Classifies the objects or predicts outcomes by selecting from a large number of variables the most important ones in determining the outcome variable.
c) CART analysis is the form of binary recursive partitioning.

### C. Steps For Cart

- Find each predictor's best split

For each continuous and ordinal predictor, sort the values from the smallest to the largest value. For the sorted predictor, go through the each value from top to examine each candidate split point (call it v, if $x \leq v$, the case goes to the left child node, else, it goes to the right) to find the best attribute. Best split point is the one that maximize the splitting criterion the most when the node is split according to the split point. The definition of splitting criterion is in a later section.

For each nominal predictors, examine each possible subset of categories (call it A, if $x \in A$, that case goes to the left child node, else, it goes to the right) to find best split.

- Find the node's best split

Among the best split found in the step 1, and choose the one that maximizes the splitting criterion.

- Split the node using the best split found in step 2, if the stopping rules are not satisfied.

Splitting Criteria and Impurity Measures (CART algorithms)At node t, the best split s is chosen to maximize a splitting criterion $\Delta i(s,t)$. When the impurity measure for a node can be defined, splitting criterion corresponds to a decrease in impurity. $\Delta I(s,t)=p(t)\Delta i(s,t)$is referred to as the improvement.

## V. EXPERIMENTAL EVALUATION

Analyze the performance with accuracy and time stamps and calculate the training data size with McDiarmid's bound and CVFDT algorithms. The proposed method is i.e., fast processing, quite good accuracy and low memory consumption, make McDiarmid Trees a proper tool for data stream mining.

## VI. CONCLUSION

Stream mining can potentially enable any user to discover what is happening to data streams in real time. There are number of methods for handling the data streams, such as tree-based algorithms (C4.5 decision tree, bagging and boosting decision tree, decision stump, boosted stump, random forest etc), neural-network, Support Vector Machine (SVMs), rule-based algorithms(conjunctive rule, RIPPER, PART, PRISM etc), naive bayes, logistic regression and many more. For stream mining, various methods are analyzed for utility to select the appropriate algorithm. Finally, the problem of data streams is solved by McDiarmid's bound tree and accuracy is improved by CART algorithm. The experimental results provide meaningful comparisons of accuracy and processing speeds between different decision tree algorithms under various memory limits. In the proposed system, the amount of memory does not depend on the size of the training data set. Using these statistics, perform a study on data stream mining using aforesaid algorithms for efficient training of data streams.

## REFERENCES

[1]. C.C. Aggarwal, "On Classification and Segmentation of Massive Audio Data Streams," Knowledge and Information System, vol. 20, pp. 137-156, July 2009.

[2]. C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu, "A Framework for On-Demand Classification of Evolving Data Streams," IEEE Trans. Knowledge and Data Eng., vol. 18, no. 5, pp. 577-589, May 2006.

[3]. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda`, "New Ensemble Methods for Evolving Data Streams," Proc. ACM SIGKDD 15th Int'l Conf. Knowledge Discovery and Data Mining, pp. 139-148, 2009.

[4]. S. Chen, H. Wang, S. Zhou, and P. Yu, "Stop Chasing Trends: Discovering High Order Models in Evolving Data," Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE), pp. 923-932, 2008.

[5]. W. Fan, "Systematic Data Selection to Mine Concept-Drifting Data Streams," Proc. ACM SIGKDD 10th Int'l Conf. Knowledge Discovery and Data Mining, pp. 128-137, 2004.

[6]. J. Gao, W. Fan, and J. Han, "On Appropriate Assumptions to Mine Data Streams," Proc. IEEE Seventh Int'l Conf. Data Mining (ICDM), pp. 143-152, 2007.

[7]. S. Hashemi, Y. Yang, Z. Mirzamomen, and M. Kangavari, "Adapted One-versus-All Decision Trees for Data Stream Classification," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 5, pp. 624-637, May 2009.

[8]. G. Hulten, L. Spencer, and P. Domingos, "Mining Time-Changing Data Streams," Proc. ACM SIGKDD Seventh Int'l Conf. Knowledge Discovery and Data Mining, pp. 97-106, 2001.

[9]. I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams," Proc. Int'l Workshop Knowledge Discovery from Data Streams (ECML/PKDD), pp. 102-116, 2006.

[10]. I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking Recurring Contexts Using Ensemble Classifiers: An Application to Email Filtering," Knowledge and Information Systems, vol. 22, pp. 371-391, 2010.