A Comparative study of Data stream classification using Decision tree and Novel class Detection Techniques

[1]Mistry Vinay R, [2]Ms. Astha Baxi

[1]M.E. Computer Science
[1]Parul Institute of Technology of Gujarat Technical University, Ahmedabad

***Abstract* - The rapid development in the e-commerce and distributed computing generates millions of the transaction, continuously. This continues arrival of data is considered as a DataStream. Data mining process for classification needs considerable modification to cope with continuous data. As Mining continues stream of data, conceptually has infinite length, and the class of data may change in sudden or gradual or hike, for which classification model is completely unknown or not prepared. Here investigation is made on different techniques proposed for the data stream classification using decision trees. Different approaches of decision tree classification for the stream data are analyzed & compared. The primary comparison parameters are time and accuracy. Also shown efforts made for handling the change in the concept and they are compared in terms of memory, technique and accuracy.**

***Index Terms* - Data stream, Novel class, Incremental learning, Ensemble Technique, Decision tree, Concept drift**

I. INTRODUCTION

A. Data Mining And Stream Data Mining

Data mining is the process of extracting hidden useful information from large volume of database. A data stream is an ordered sequence of instances that arrive at any time does not permit to permanently store them in memory. Data mining process has two major functions: classification and clustering. Data stream classification is the process of extracting knowledge and information from continuous data instances. The goal of data mining classifiers is to predict the class value of a new or unseen instance, whose attribute values are known but the class value is unknown [1]. Classification maps data into predefined that is referred to a supervised learning because the classes are determined before examining the data and that analyses a given training set and develops a model for each class according to the features present in the data. In clustering class or groups are not predefined, but rather defined by the data alone. It is referred to as unsupervised learning. There are three major problems related to stream data classification [2]. It is impractical to store and use all the historical data for training there may be concept-drift in the data, meaning, the underlying concept of the data may change over time.

In data stream classification most of the existing work related to infinite length and concept drift here we focus on the novel class detection and concept drift. Stream classification problems, such as intrusion detection, text classification, fault detection, novel classes may arrive at any time in the continuous stream. Several approaches are exists in order to develop the classification models including decision trees, neural networks, nearest neighbor methods and rough set-based methods [4]. Data stream classifiers can be categorize as: single model and ensemble model [1]. Single model incrementally update a single classifier and effectively respond to concept drifting so that reflects most recent concept in data stream. Ensemble model use a combination of classifiers with the aim of creating an improved composite model, and also handle concept drifting efficiently. In the traditional tree induction algorithm, the time in which the data arrived is not considered. The incremental classifier that reflects the changing data trends effective and efficient so it is more attractive. Incremental learning is an approach which deals with the classification task. When datasets are too large or when new examples can arrive at any time [5]. Incremental learning most important in applications where data arrives over long periods of time and storage capacities are very limited. In [7] author Defines incremental tasks and incremental algorithms as follows:

Definition 1: A learning task is incremental if the training examples used to solve it become available over time, usually one at a time.

As per [8] the learning to be one that is: Capable to learn and update with every new data (labeled or unlabeled), Will use and exploit the knowledge in further learning, Will not rely on the previously learned knowledge, Will generate a new class as required and take decisions to merge or divide them as well Will enable the classifier itself to evolve and be dynamic in nature with the changing environment. Decision tree that provide the solution for handling novel class detection problem. ID3 is very useful learning algorithm for decision tree.C5.0 algorithm improves the performance of tree using boosting. Hoeffding tree containing additional option nodes. Option tree represent middle ground between Incremental and Ensemble approach. HOT that control tree growth and determine number of option to explore [13].

B. Data Stream Classification

A classification algorithm must meet several requirements in order to work with the assumptions and be suitable for learning from data streams [9]

Requirement 1: Practice an example at a time. Inspect it only once (at most). Traditional Data Mining Stream Data Mining. No. of passes Multiple Single Processing time Unlimited Restricted Memory usage Unlimited Restricted Type of result Accurate Approximate Concept Static Evolving Distributed

Requirement 2: Consumption of a limited quantity of memory

Requirement 3: Work in a limited quantity of time

Requirement 4: Ready to predict (foretell) at any point

Here in the Figure 1 the typical use of a data stream classification algorithm is illustrated. Moreover is shows how the requirements fit in. The general model of data stream classification follows these three steps in a repeating cycle:

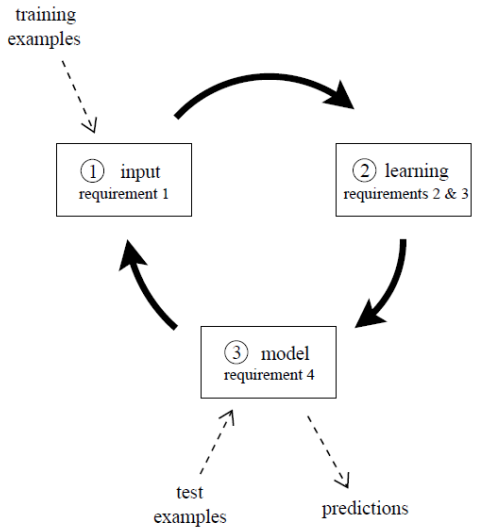

Fig 1: The data stream classification cycle [15]

Step - 1. Pass the next available example from the stream (requirement 1).

Step - 2. Process the example. Update its data structures (by fulfilling requirement 2 & 3).

Step - 3. Ready to accept the next example. On request it is able to supply a model which can be used to predict the class of unseen examples.

II. NOVEL CASS DETECTION AND HANDLING

Novel class detection in stream data classification is interesting research topic and researches available for concept drift problem but not attention on the Novel class detection. Data stream classification and novelty detection recently received increasing attention in many practical real-world applications, such as spam, climate change or intrusion detection, where data distributions inherently change over time[6]. Ensemble techniques maintain a combination of models, and use ensemble voting to classify unlabeled instances. As per [6] In 2011, Masud et al. proposed a novelty detection and data stream classification technique, which integrates a novel class detection mechanism into traditional mining classifiers that enabling automatic detection of novel classes before the true labels of the novel class instances arrive. In [9], [10] author gives the definition of the existing class and Novel class.

Definition 1 (Existing class and Novel class): Let L is the current ensemble of classification models. A class c is an existing class if at least one of the models Li ∈ L has been trained with the instances of class c. Otherwise, c is a novel class in 12 points boldface italic and capitalize the first letter of the first word only. Do not underline any of the headings, or add dashes, colons, etc.

In [10] show the basic idea of novel class detection using decision tree in Figure 1. That introduces the notion of used space to denote a feature space occupied by any instance, and unused space to denote a feature space unused by an instance.

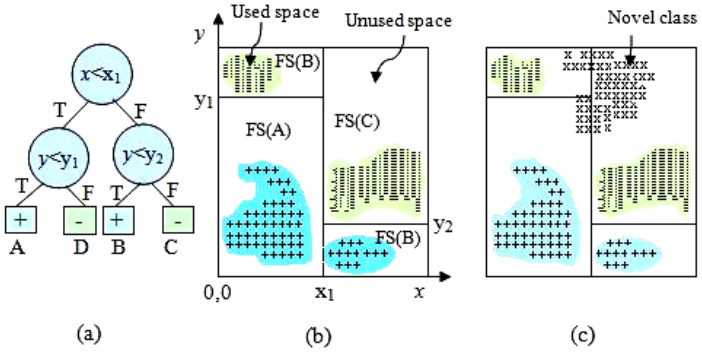

Fig. 2 (a) A decision tree, (b) corresponding feature space partitioning where FS(X) denotes
The Feature space defined by a leaf node X The shaded areas show the used spaces of each partition.(c) A Novel class (denoted by x) arrives in the unused space

A.*Concept Drift*

In the reference of predictive analytics and machine learning, the concept drift means that the statistical properties of the target variable which the model is trying to predict can change over time in unforeseen ways. This causes problems because of the predictions become less accurate as the time passes. The term concept refers to the quantity to be predicted. Usually it can also refer to other phenomena of interest besides the target concept such as an input but, in the context of concept drift the term commonly refers to the target variable or target concept. In order to stop deterioration in prediction accuracy because of concept drift, both active and passive solutions can be adopted. Active solutions rely on triggering mechanisms, e.g., change-detection tests to explicitly detect concept drift as a change in the statistics of the data-generating process. In stationary conditions, any of the fresh information made available can be integrated to improve the model. Differently, when concept drift is detected, the current model is no more up-to-date and must be substituted with a new one to maintain the prediction accuracy .On the contrary, in passive solutions the model is continuously updated, e.g., by retraining the model on the most recently observed samples Contextual information, when available, can be used to better explain the causes of the concept drift: for instance, in the sales prediction application, concept drift might be compensated by adding information about the season to the model. By providing information about the time of the year, the rate of deterioration of your model is likely to decrease; concept drift is unlikely to be eliminated altogether. This is because genuine shopping behavior does not follow any static finite model. New factors may arise at any time that effect shopping behavior. The influence of the known factors or their interactions may change.

III. DECISION TREE CLASSIFIER TECHNIQUES

In [11] paper, authors have proposed incremental option trees for regression on fast non-stationary data streams (FIOT). The option nodes are introduced in order to improve the bias management in incremental learners, introduce ambiguous splits for learning under gradual concept drift and enable faster growth without instability in the splitting decisions. They have shown that the option tree is able to achieve better accuracy faster than a regular regression tree. This is especially pronounced for the data with gradual concept drift. Option nodes act as an improved look-ahead strategy and present an interpretable version of ensemble methods. In the [12] work presents an incremental learning algorithm appropriate for processing high-speed numerical data streams. The main contributions of this work are the ability to use multivariate splitting tests, and the ability to adapt the decision model to concept drift. While the former has impact in the performance of the system, the latter extends the range of applications to dynamic environments. To detect concept drift, we maintain, at each inner node, a naive-Bayes classifier trained with the examples that cross the node. In [11] authors have proposed a Sensitive Concept Drift Probing Decision Tree algorithm (SCRIPT). The main contributions of SCRIPT are: a) it can avoid unnecessary system cost for stable data streams; b) it can efficiently rebuild classifier while data streams are instable; c) it is more suitable for the applications in which a sensitive detection of concept drift is required. our Sensitive Concept Drift Probing Decision Tree (SCRIPT) algorithm in this section. Based on the variation of CDAV (Class Distribution on the Attribute Value), SCRIPT aims to apply to large scale and high speed applications which also require the sensitiveness to handle the drifting concepts. SCRIPT cuts the data stream into sequential data blocks.

When the test threshold is set as α, each training data in the block should be read no more than once and processed in minimal constant time while concepts are stable. While concepts are instable, SCRIPT would detect concept drift, and then build the alternate tree to correct previously built tree. In [7] paper, Authors have proposed a novel method for on-line learning regression trees with option nodes (ORTO) from data streams. It is basically works on Hoeffding bound. Our method for learning Hoeffding-based option trees for regression addresses the problem of instability of tree learning, commonly seen in the case of highly correlated or equally discriminative attributes, i.e., in tie situations. Hoeffding trees can suffer from a delay in the learning process in such tie situations, because they assume that the data is in abundance and will never stop to stream in: Decisions on split selection are postponed resulting in lower learning rates. They show that option nodes are a natural and effective solution to the problem of dealing with multiple equally discriminative attributes (the tie problem). The additional structure of the option trees provides interesting and useful information on the ambiguity of the splits and thus on the existence of several equally relevant attributes. In [12] authors have demonstrated the efficacy of incorporating multiple paths via option nodes in Hoeffding trees (HOT). They described a method for controlling tree growth, and determined a reasonable number of options to explore. In all but one of our datasets the additional

structure improved the performance of the classifier. Option trees represent a useful middle ground between single trees and ensembles. At a fraction of the memory cost an option tree can provide comparable accuracy performance and superior prediction speed which are important factors in data stream processing. In particular, Fine-Grained Access Control (FGAC) system and Role-Based Access Control (RBAC) are widely used secure systems [10]. The FGAC system determines access control rights based on individual data. The access rights of every user allowed to access the data are stored inside the data itself [11]. In RBAC, access rights are defined based on the position or role of the user [12]. There may be multiple users with the same role. These multiple users may have access to the group or collection of data.

A. Hoeffding Option Tree

As per describe in [13] Hoeffding trees are state-of-the-art for processing high-speed data treams. Hoeffding Option Trees are regular Hoeffding tree which contains additional option nodes, that permit several tests to be applied & leading to multiple Hoeffding trees as separate paths. When training a model on a data stream it is important to make a single scan of the data as quickly as possible. Hoeffding trees achieve this by accumulating sufficient statistics from examples in a node to the point where they can be used to make a sensible split decision. The sufficient statistics are beneficial for both tree growth and prediction as they can be used to build Naive Bayes models at the leaves of the tree that are more accurate than majority class estimates. Option trees represent a middle ground between single trees and ensembles. They are capable of producing useful and interpretable, additional model structure without consuming too many resources. Option trees are comprise of a single structure that efficiently embodies multiple trees. A precise illustration can travel down multiple paths of the tree to different options.

B. Option Node

Fig 3 is an example of what the top few levels of an option tree can look like. The tree is a regular decision tree in form except for the presence of option nodes, depicted in the figure as rectangles. At these nodes multiple tests will be applied, implying that an example can travel down multiple paths of the decision tree, and arrive at multiple leaves. Option Tree is control tree growth, and determined a reasonable number of options to explore it. As per [14] option nodes are a natural and effective solution to the problem of dealing with multiple equally discriminative attributes (the tie problem). The additional structure of the option trees provides interesting and useful information on the ambiguity of the splits and thus on the existence of several equally relevant attributes.

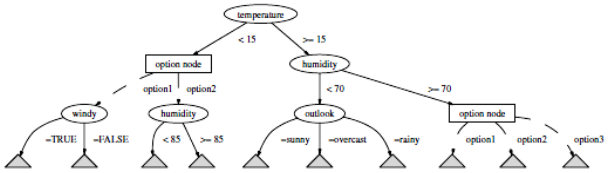

Fig 3 An option tree[3]

IV. NOVEL CLASS DETECTION TECHNIQUES

1. MineClass[12]

"MineClass", this stands for mining novel Classes in data streams with base learner K-NN (K-nearest neighbor) and decision tree. K-NN based approaches for novelty detection is also non-parametric. Novelty detection is also closely related to outlier/anomaly detection techniques. There are many outlier detection techniques available some of them are also applicable to data streams. Herein this paper work in data stream mining domain describes a clustering approach that can detect both concept-drift and novel class and assumes that there is only one "normal" class and all other classes is novel. Thus, it may not work well if more than one class is to be considered as "normal" or" nonnovel". Mine class can detect novel classes in the presence of concept-drift, and proposed model is capable of detecting novel classes even when the model consists of multiple "existing" classes. MineClass Technique adapting Ensemble classification. That maintains M number of classifier for classifying Unlabel data. Here categorize novelty detection techniques into two categories: statistical and neural network based. This MineClass technique related to the statistical approach. Statistical approaches are of two types: parametric, and non-parametric. Parametric approaches assume that data distributions are known e.g. Gaussian. And try to approximation the parameters e.g. mean and variance of the distribution. If any test data falls outside the normal parameters of the model, it is declared as novel. MineClass is a non-parametric approach. Non-parametric approaches estimate the density of training data and reject data whose density is beyond a certain threshold.

2. ActMiner[8]

As per [8] ActMiner, which stands for Active Classifier for Data Streams with novel class Miner, performs classification and novel class detection in data streams while requiring small amount of labeled data for training. ActMiner use an ensemble classification technique by addressing the limited labeled data problem. ActMiner extends MineClass, and addresses the Limited labeled data problem thereby reducing the labeling cost. It also applies active learning, but its data selection process is different from the others. An unsupervised novel concept detection technique for data streams is proposed, but it is not applicable to multi-class classification. As per work mention in MineClass addresses the concept evolution problem on a multi-class classification framework. MineClass does not address the limited labeled data problem, and requires that all instances in the stream be labeled and available for training. ActMiner also applies active learning, but its data selection process is different from the others. Unlike other active mining techniques such that requires extra computational overhead to select the data.

3. ECSMiner[4]

In [4] author describes ECSMiner means Enhanced Classifier for Data streams with Novel Class Miner. Novelty detection is the identification of different or anonymous data or signal. Generally machine learning system is not aware of such new & unknown data during training. Novel class detection using ECSMiner is different from traditional one class detection technique. This approach offers a "multiclass" framework for the novelty detection problem. This Algorithm can also distinguish between different classes of data and discover the arrival of a novel class. This technique is a nonparametric approach, and therefore, it is not restricted to any specific data distribution. ECSMiner is different from other technique in some aspects are first It not only considers difference of test instance from training data but also similarities among them. Technique discovers novelty collectively among several coherent test points to detect the presence of a novel class. Second It is "multiclass" novelty detection technique, and also discover emergence of a novel class. Third Approach can detect novel classes even if concept-drift occurs in the existing classes. "ECSMiner" (pronounced like ExMiner).This technique on two different classifiers: decision tree and k-nearest neighbor. When decision tree is used as a classifier, each training data chunk is used to build a decision tree. K-NN strategy would lead to an unproductive classification model, both in expressions of memory and running time. ECSMiner detect novel classes automatically even when the classification model is not trained with the novel class instances. ECSMiner first find the Foutlier than check if instance is not Foutlier, this type of instances classified immediately. Foutlier does not require it is existing class. In the second phase that again do the classification of Foutlier instances for finding the actual Novel instances.

4. SCANR[6]

As referred from [6] a recurring class is a special case of concept-evolution. It occurs when a class reappears after long disappearance from the stream. ECSMiner identifies recurring classes as novel class. Each incoming instance of data stream is first check by primary ensemble if it is outlier called it primary outlier (P-outlier) than again check through auxiliary ensemble if it is outlier than called secondary outlier(S-outlier), and it is temporarily stored in a buffer for further analysis. The novel class detection module is invoked when there are enough instances in the buffer. In this technique compute a unified measure of cohesion and separation for an S-outlier x, called q-NSC (neighborhood silhouette coefficient), range of q-NSC is [-1, +1]. The q-NSC(x) value of an S-outliers x is computed separately for each classifier. A novel class is declared if there are S-outliers having positive q-NSC for all classifiers. Recurring class instance, they should be Poutliers but not S-outliers because the primary ensemble does not contain that class, but secondary ensembles shall contain that class. The instances that are classified by the auxiliary ensembles are not outliers. The technique for Classification with novel and recurring class is called SCANR (Stream Classifier and Novel and Recurring class detector). As describe from figure 4 ensemble models consists of two ensembles, a primary ensemble M, and an auxiliary ensemble MA. In short, M is an ensemble of L classification models and MA is an ensemble of LAC auxiliary models where C is the number of classes seen in the stream so far, and LA is the number of auxiliary models per class. The data stream is divided into equal sized chunks, and the data points in the most recent data chunk are first classified using the ensemble. When the data points in a chunk become labeled that chunk is used for training a classification model. Since the number of models in each ensemble is fixed, the newly trained model replaces the existing model(s) in each ensemble approach incoming instance in the data stream is first examined by the outlier detection module of the primary ensemble in order to check whether it is an outlier.

5. Decision Tree

In [1] authors have proposed New decision tree learning approach for detection of Novel class. This decision tree approach is work on Incremental learning. In this approach build a decision tree with data stream that update with continuously coming example and update with new data point so that represent the most recent concept in data stream. Here that calculates the threshold value based on the ratio of percentage of data points between each leaf node in the tree and the training dataset and also clusters the data points of training dataset based on the similarity of attribute values. That check If number of the data points classify by a leaf node of the tree increases than the threshold value that calculated before, which means a novel class arrived. Than after that compare the new data point with existing data point based on similarity of attributes, if the attribute value of new data point is different than existing data point that means this is the actual novel class.

Table 2: Comparative Analysis of Novelty Detection Techniques

Algorithm	Learning Approach	Classifer	Advantage	Disadvantage
MineClass	Ensemble	Decision tree and K-NN (Train and create inventory baseline techniques.)	Nonparametric. Does not require data in convex shape.	That requires 100% label instance.
ACT Miner	Ensemble	Active classifier work with K-NN and decision tree.	Work on the less label instance. It saves 90% or more labeling time and cost.	Not directly applicable to multiclass. Not work for the multi label classification.
ECS Miner	Ensemble	Classical classifier Work with K-NN and decision tree.	Non parametric Does not require data in convex shape	Not efficient in terms of memory and run time. It Identifies recurring class as Novel class.
SCANR	Ensemble	Multiclass classifier	Remembers a class and identifies	Auxiliary ensemble is

			it as "not novel" when it reappears after a long disappearance.(Detect Recurring class)	used so running time is more than other detection method

V. CONCLUSION

The comparative analysis of decision tree classification technique shows that the Hoeffding Option tree classification is the most promising method for handling continuous stream of data. But HOT is lacking in handling concept drift or arrival of novel class. The efforts for handling concept drift or arrival of novel class are mainly based on the ensemble learning like MINECLASS, ACTMINER, ECSMINER, SCANR and are not efficient as compared to HOT. Most of the efforts for handling novel class are based on ensemble learning are memory intensive, and the one incremental learning approach is based on c4.5 which is basically not designed for the stream classification.

ACKNOWLEDGMENT

With the cooperation of my guide, I am highly indebted to Ms. Astha Baxi, for her valuable guidance and supervision regarding my topic as well as for providing necessary information regarding review paper. I am very much thankful to Asst. Prof. G.B Jethva for helping me in text preparation.

REFERENCES

[1] Mohammad M Masud, Tahseen M, Al-khateeb, Latifur Khan, Charu Aggrawal, Jing Gao, Jiawei Han and Bhawani Thuraisinghum "Detecting Recurring and Novel classes in Concept Drift Data Streams "icdm, pp. 1176- 1181, 2011 IEEE 11th International Conference On Data Mining.

[2] S.Thanngamani DYNAMIC FEATURE SET BASED CLASSIFICATION SCHEME UNDER DATA STREAMS International Journal Of Communication And Engineering Volume 04 – No .04, Issue:01 March-201. Bertino E., and Damiani M. L., "A Controlled Access to Spatial Data on Web", Conference on Geographic Information Science, *AGILE* Conference, Heraklion, Greece, Vol., pp., April 29-May 1, 2004.

[3] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han, Bhavani Thuraisingham Classification And Novel Class Detection In Data Stream With Active Mining M.J.Zaki etal.(Eds.): PAKDD 2010, Part II,LNAI 6119, pp.311-324 Springer- Verlag Berlin Heidelberg 2010.

[4] Amit Biswas, Dewan Md. Farid and Chowdhary Mofizur Rahman A New Decision Tree Learning Approch For Novel Class Detection In Cocept Drifting Data Stream Classification JOURNAL OF COMPUTER SCIENCE AND ENGINEERING, VOLUME 14, ISSUE 1, JULY 2012.

[5] S.PRASANNALAKSHMI,S.SASIREKHA INTERGATING NOVEL CLASS DETECTION WITH CONCEPT DRIFTING DATA STREAMS International Journal Of Communication And Engineering Volume 03, No.03, Issue:04 March 2012.

[6] JIGNASA N. PATEL, SHEETAL MEHTA Detection Of Novel Class With Incremental Learning For Data Streams International Journal Of Research in Modern Engineering and Emerging Technology Vol.1, Issue:3 April-2013.

[7] Geoffrey Holmes, Richard Kirkby, and Bernhard P Fahringer Mining Data Stream Using Option Trees(revised edition 2004).

[8] Pedro Domingos, Geoff Hulten Mining High-Speed Data Streams in proceeding of the 6th ACMSIGKDD International Conference On Knowledge Discovery and Data Mining, pp.71-80, ACM, August-2000

[9] Mi losz R. Kmieciak and Jerzy Stefanowski Stream Handling Sudden Concept Drift in Enron Messages Data Mat. III KNTPD Conf., Poznan 21-23 April 2010, WNT Press, 2010, 284-296

[10] Bernhard Pfahringer, Geoffrey Holmes, and Richard Kirkby New Options for Hoeffding Trees Springer-Verlag Berlin Heidelberg M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830,2007, pp. 90–99