Detection of Novel Class for Data Streams

Megha Patel, Yesha Patel
Computer Engineering Department, Silver Oak College of Engineering & Technology, Gujarat Technological University, India

Abstract - Data stream mining is a process of extracting the information from continuously coming rapid data records. Data stream can be viewed as an ordered sequence of instances appears at time varying. Data stream classification has three major problems: infinite length, concept drift and concept evolution or arrival of novel class. In this paper, we propose a new approach for detection of novel class using decision tree classifier that determine whether a new or unseen data instance belongs to an existing class or novel class. It builds decision tree from training datasets, so the tree represents the most recent concept by constantly updating it. The experiment on different datasets from UCI machine learning repository evaluate the efficiency of proposed approach for detecting novel class under dynamic attribute set and classification accuracy by comparing with traditional data mining classifiers.

Index Terms: Data stream mining, Novel class, Incremental learning, Decision tree

I. INTRODUCTION

Data mining is the process of gathering, searching through and analyzing a large amount of data in a database, as to discover the patterns or relationships. Data stream is a sequence of instances that appears constantly at any time, doesn’t permit to hoard them permanently into the memory. This speedy creation of non-stop stream of information has challenged our storage, communication and computation potential in computing organism [1]. The traditional data mining techniques requires multiple scans of data to extract the information which is not realistic with data streams, so they cannot be directly apply to data stream. In stream mining, the knowledge structures are taken out from continuously arriving data records.

Mining includes two major functions: classification and clustering. Classification extracts the information and knowledge form continuously arriving data instances. Classifier is used to forecast the class value for unobserved new instances whose attribute value is known but the class value is not known [1]. In supervised learning data are mapped into predetermined groups by classification in which classes are defined before examining data, in which training data set analyzes and builds the model for each class using the characteristics at hand in the data. Classes are not predetermined in clustering, but described by data alone, which is known as unsupervised learning.

In stream classification, there are three major challenges [2].

1. Data stream is theoretically infinite in length, so it’s impossible to save and use all the past facts for training, since it needs unlimited space and executing time.
2. Concept drift occurs as the fundamental concept of the data may change over time.
3. Concept evolution occurs when a novel class may arrive in data stream.

The existing work related to first two problems in stream classification, the later one was not much concentrated. Here we concentrate on the third problem – concept evolution. Existing solution assumes that the total numbers of classes are fixed and so that only the instances of trained classes are correctly classify by traditional classifiers. In the data stream, when new class appears, until it has been identified by some expert manually and new model is trained with the labeled instances of that class, all the instances belongs to that class are misclassified, but in real world, novel classes may arrive from nonstop data stream at any time. Data mining classifier should be able to update continuously so that it reveals the most recent concept. Many techniques like neural network, decision trees, and rule-set based & nearest neighbor methods are available to develop classification model [3].

There are two approaches for stream classification: single model and ensemble model. Novel classes may arrive at time varying in real-world stream classification problems, like text classification, intrusion detection and fault detection [3]. A single model incrementally updates single classifier and
responds effectively to the appearance of novel class so that it represents the latest concept in the stream very efficiently and effectively, so it is more attractive. It is also known as incremental learning approach and beneficial to deal with the classification task when datasets are too huge or when new examples can arrive at any time \cite{4}. The combination of classifiers is used in ensemble model to create complex model, and handles concept drift very powerfully.

Incremental algorithms can be defined as follow:

**Definition 1**: A learning task is incremental if the training examples used to solve it become available over time, usually one at a time \cite{5}.

The incremental learning is that which is: Able to learn and update with every new appearance of fact (labeled or unlabeled), will not depend on the formerly learned knowledge, will exploit and use the knowledge in further learning, will produce a new class as needed and take judgments to divide or merge them as well as will allow the classifier itself to be dynamic in nature and evolve with the varying situations \cite{6}. Incremental learning approach is used by decision tree classifier for handling novel class detection problem. ID3 is very useful learning algorithm for novelty detection.

II. CLASSIFICATION AND NOVEL CLASS DETECTION

In stream classification, researches available for concept drift problem but not for novel class detection which is interesting research topic nowadays. This approach fall into two categories: Single model (Incremental approach), Ensemble Model. In many real-world applications, such as spam detection, climate change or intrusion detection, where data distributions inherently change over time, the data stream classification and novelty detection received increasing attention \cite{1}.

**Definition 1** (Existing class and Novel class): Let L be the current ensemble of classification models. A class c is an existing class if at least one of the models Li ∈ L has been with the instances of class c. Otherwise, c is a novel class \cite{2,7}.

To detect a novel class that has the following essential property:

**Property 1**: A data point should be closer to the data points of its own class (cohesion) and farther apart from the data points of any other classes (separation) \cite{3}.

Fig.1 shows decision tree to get basic idea of detection of novel class. A feature space occupied by any instance is denoted as notion of used space, and a feature space unoccupied by an instance is denoted as notion of unused space.

Only in the unused spaces, the novel class must appear, as an accordance of property 1 (cohesion). There must be strong consistency among the instances of the novel class. For detection of novel class, the two basic steps are there: First, the classifier is trained such that an inventory of the used spaces is created and saved, done by clustering and saving the cluster summary as “pseudo point”. Secondly, these Pseudo points are used to recognize outliers in the test data, and if there is strong cohesion among the outliers, declare a novel class.

III. RELATED WORK

Data stream classification and novelty detection lately received increasing attention in applications like, spam detection, intrusion detection and climate change, where the distribution of data changes over time. Traditional stream classification methods are not able to detect the novel class instances until the emergence of novel class is identified manually, and labeled instances of that class are given for training to the leaning algorithm. The problem becomes more challenging when there is concept drift that is underlying concept of data changes over time. Novelty detection is closely related to anomaly or outlier detection methods. The main difference is that here primary objective is novel class detection, not outlier detection. Outliers are the by-product of intermediary computational steps in novel class detection algorithm.
MineClass\textsuperscript{[2]} is non-parametric, ensemble based learning approach in which decision tree and K-NN classifiers are used for novelty detection. It is an efficient and novel method that detects the novel class in the presence of concept drift automatically by computing the closeness among unlabeled test instances and separation of test instances from training instances \textsuperscript{[2]}. Clustering is used that can detect both concept-drift and arrival of novel class; it assumes that there is only one ‘normal’ class and all other are novel classes. If more than one class is to be considered as ‘normal’ or ‘non-novel’, it may not work well. It detects the novel class on a multi-class classification framework, but doesn’t address the limited labeled data problem.

ActMiner\textsuperscript{[7]} is non-parametric ensemble learning based approach extending MineClass, and problem of limited labeled is tackled by addressing other three stream classification problems. The labeling cost is reduced by selecting those data points for which the usual classification error is high, but not applicable to dynamic feature set and multi-label classification problem.

ECSMiner\textsuperscript{[8]} pronounced as “ExMiner”, non-parametric, ensemble based approach applied to decision tree and K-NN classifier. It is “multiclass” novelty detection technique. Decision tree is built by using each training data chunk, when decision tree classifier is used; whereas classification model become inefficient with K-NN classifier, both in terms of running time and memory.

SCANR\textsuperscript{[9]} is ensemble based novelty detection technique which detects both the novel and recurring class. A recurring class is a special case of concept evolution, which take place when a class appears in the stream, then disappears for a long time and again appears. SCANR is more reasonable novelty detection method that remembers a class and declares it as “not novel” when it recurs after long disappearance and greatly reduce the human effort, false alarm rate and resource consumption.

IV. PROPOSED ALGORITHM

Decision tree is built by choosing the best splitting attribute from a dataset with the highest information gain. Algorithm 1 outlines our approach. Take the latest instance from the stream and classify it using decision tree classifier. If it is not classified immediately, store into the buffer and when the buffer is full, classify the instances of it. For classifying the instances, check whether it is actually a novel class instance or not. If it is Foutlier (Filtered Outlier) means it is not classify by any existing class; then it is novel class instance; otherwise it is existing class instance and classify it immediately. For tree construction, until the feature set is empty do following: randomly choose the untested feature \(F_i\) of the instance from the feature set \(X\) without training data points and construct children for all the valid value of \(F_i\). When the tree construction is completed and that feature is also there in the training set, then update the tree structure using the training data points.

**Algorithm 1: Decision Tree Miner**

**Inputs:** Training Set \(S = \{(x_1,t_1), (x_2,t_2),\ldots,(x_n,t_n)\}\)

Feature Set \(X = \{F_1, F_2,\ldots, F_n\}\)

\(F\) is a feature descriptor

**Output:** Decision Tree

**Procedure Classify (DT, xj, buf)**

1. \(buf \leftarrow \) empty // temp buffer
2. \(L \leftarrow \) empty //labeled data buffer
3. \(U \leftarrow \) empty // unlabeled data buffer
4. \(\) while true do
5. \(xj \leftarrow \) the latest instance in the stream
6. \(\) Classify (DT, xj, buf)
7. \(\) if class label cannot predict immediately store into \(U\)
8. \(\) Else label it and store into \(L\)
9. \(\) End if
10. \(\) End while
11. \(\) while (Instances in \(U\))
12. \(\) Classify(DT, xj, U)
13. \(\) End while
14. \(\) End

**Procedure RandomTreeGeneration**

1. \(\) For \(i = \{1, 2,\ldots, n\}\)
   \(\) ConstructTreeStructure(Ti,X);
2. \(\) For \((x,t) \in S\)
   \(\) For \(i = \{1, 2,\ldots, n\}\)
   \(\) UpdateTreeSet(T,(x,t));

**Procedure ConstructTreeStructure(T,X)**
1. If \( X = \phi \)
   Make a leaf node;

   Else
   Choose \( F \) randomly;
   Construct \( m \) children node \( n_j \), for each \( m \)
   valid value \( d_j \) of \( F \);
   For \( j = \{1, 2, \ldots, m\} \)
   ConstructTreeStructure(\( n_j, X - F \))

Procedure UpdateTreeSet(\( T,(x,t) \))

1. \( n[t] \) is number of example with class label \( t \);
2. \( n[t] \leftarrow n[t] + 1 \);
3. If node is not leaf node
   \( d \) is value of node’s testing feature \( F \),
   \( d = F(X) \);
   \( n \) is corresponding child node for \( d \);
   UpdateTreeSet(\( n, (x,t) \));

V. EXPERIMENTS

A. Datasets

Dataset is a set of data items, a basic concept of
machine learning and data mining research. Table 1
shows the dataset description which we have used.

Table 1 Dataset Description

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attributes</th>
<th>Instances</th>
<th>Class</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>4</td>
<td>150</td>
<td>3</td>
<td>Real</td>
</tr>
<tr>
<td>Breast Cancer</td>
<td>32</td>
<td>569</td>
<td>2</td>
<td>Real</td>
</tr>
<tr>
<td>Large Soybean</td>
<td>35</td>
<td>683</td>
<td>19</td>
<td>Nominal</td>
</tr>
<tr>
<td>NSL-KDD</td>
<td>41</td>
<td>25192</td>
<td>23</td>
<td>Real</td>
</tr>
</tbody>
</table>

B. Results

We implement our algorithm in Java. The code for
decision tree has been adapted from Weka open
source repository. The experiments were run on Intel
Core 2 Duo Processor with 2 GB RAM.

Table 2 Used symbol and term

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N )</td>
<td>Total instances in stream</td>
</tr>
<tr>
<td>( N_c )</td>
<td>Total novel class instances in stream</td>
</tr>
<tr>
<td>( F_n )</td>
<td>Total novel class instances as existing class</td>
</tr>
<tr>
<td>( F_p )</td>
<td>Total existing class instances misclassified as novel class</td>
</tr>
<tr>
<td>( F_e )</td>
<td>Total existing class instances misclassified</td>
</tr>
<tr>
<td>( F_{new} )</td>
<td>% of existing class instances misclassified as novel class</td>
</tr>
</tbody>
</table>

Table 2 summarizes the symbols and terms used
throughout the 1 to 3, which are used to evaluate our
technique.

\[
M_{new} = \frac{F_n}{N_c} \times 100
\]

\[
F_{new} = \frac{F_p}{N - N_c} \times 100
\]

\[
ERR = (F_p + F_n + F_e) \times 100 / N
\]

Table 3 Performance comparison

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Dataset</th>
<th>( M_{new} )</th>
<th>( F_{new} )</th>
<th>ERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Approach</td>
<td>Iris</td>
<td>4.0</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>Breast Cancer</td>
<td>1.2</td>
<td>2.0</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td>Large Soybean</td>
<td>13.5</td>
<td>4.0</td>
<td>13.0</td>
</tr>
<tr>
<td></td>
<td>NSL-KDD</td>
<td>7.5</td>
<td>1.0</td>
<td>3.8</td>
</tr>
<tr>
<td>Existing Approach</td>
<td>Iris</td>
<td>6.0</td>
<td>0.0</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>Breast Cancer</td>
<td>1.4</td>
<td>2.5</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>Large Soybean</td>
<td>16.3</td>
<td>4.3</td>
<td>15.3</td>
</tr>
<tr>
<td></td>
<td>NSL-KDD</td>
<td>8.4</td>
<td>1.2</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table 3 shows the performance comparison between
our approach and existing approach.

VI. CONCLUSION

In this paper, we introduce a new approach for novel
class detection in concept drifting data stream mining
that works under dynamic attribute set, in which tree
represents the most recent concept by continuously
updating it. The main purpose is to improve the
performance of decision tree classifier and make it to
work under dynamic attribute set. Decision tree is
supervised learning algorithm which is easy to
implement and requires little prior knowledge, so it is
very popular. In future, it should be able to detect
novel class under dynamic attribute set using other
base learner.

REFERENCE

Class Detection in Concept Drifting Data Stream
Mining Employing Decision Tree,” 7th international
cconference on Electrical and Computer Engineering.
978-1-4673-1436-1/12, 2012 IEEE.

[2] Mohammad M. Masud, Jing Gao, Latifur Khan,
Jiawei Han and Bhavani Thuraisingham, “Integrating
Novel Class Detection with Classification for
Concept-Drifting DataStreams,” W. Buntine et al.


[7] Mohammad M. Masud, Jing Gao, Latifur Khan, Jiawei Han and Bhavani Thuraisingham, “Classification and Novel Class Detection in Data Streams with Active Mining,” M.J.Zaki et al. (Eds.):PAKDD 2010, Part II, LNAI 6119, pp. 311-324 Springer-Verlag Berlin Heidelberg 2010.


[9] Mohammad M. Masud, Tahseen M. Al-Khateeb, Latifur Khan, Charu Aggarwal, Jing Gao, Jiawei Han and Bhavani Thuraisingham, “Detecting Recurring and Novel Classes in Concept-Drifting Data Streams,” icdm, pp.1176-1181, 2011 IEEE 11th International Conference on Data Mining, 2011.