ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 4, March 2017

Recurring and Novel Class Detection Using Class-Based Ensemble for Evolving Data Stream

J.Evangelin Deva Sheela
M.Phil Research scholar
VPMM Arts and Science College
Krishnankoil
sheela8mca@gmail.com

***Abstract*— Streaming data is one of the attention receiving sources for concept-evolution studies. When a new class occurs in the data stream it can be considered as a new concept and so the concept-evolution. One attractive problem occurring in the concept-evolution studies is the recurring classes from our previous study. In data streams, a class can disappear and reappear after a while. Existing studies on data stream classification techniques either misclassify the recurring class or falsely identify the recurring classes as novel classes. Because of the misclassification or false novel classification, the error rates increases on those studies. In this paper we address the problem by defining a novel ensemble technique "class-based" ensemble which replaces the traditional "chunk-based" approach in order to detect the recurring classes. We discuss the details of two different approaches in class-based ensemble and explain and compare them in detail. Different than the previous studies in the field, we also prove the superiority of both "class-based" ensemble method over state-of-art techniques via empirical approach on a number of benchmark data sets including web comments as text mining challenge.**

Keywords—component; formatting; style; styling; insert (key words)

I. INTRODUCTION

THE issues of one-pass learning and concept-drift are widely studied in the context of data stream classification [1], [2], [3]. One pass learning is required because of the massive volume and continuous delivery of the data stream. Concept-drift occurs in dynamic streams, and many different techniques have been designed with the same goal of maintaining an up-to-date classification model [1], [2], [4], [5], [6]. Another issue of considerable recent interest is that of concept-evolution, which refers to the emergence of a new class. Several approaches have been proposed to address this issue [7], [8]. These approaches pro-actively detect novel classes in the stream, before being trained with the novel class instances. Most other data stream classification methods fail to detect novel classes because they assume that the number of classes in the data stream is fixed.

In real data streams, the emergence of new classes is a common phenomenon. 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. A special case of concept-evolution is that of a recurring class, which occurs when a class reappears after a long disappearance from the stream. This special case is also common because an intrusion in network traffic may reappear after a long time, or social network actors may discuss an interesting topic in Twitter at a particular time every year (e.g., Halloween). Another challenge with the Twitter stream is to monitor topics and detect when trends emerge. This includes general changes in topics such as sports or fashion and it includes new quickly emerging trends such as deaths or catastrophes. It is a challenging problem to correctly associate tweet messages with trends and topics. These challenges are best addressed with a streaming model due to the continuous and large volumes of incoming messages and a large number of existing and recurring classes.

Data stream classifiers may either be single model incremental approaches, or ensemble techniques, in which the classification output is a function of the predictions of different classifiers. Ensemble techniques have been more popular than their single model counterparts because of their simpler implementation and higher efficiency [9]. Most of these ensemble techniques use a chunk-based approach for learning [7], [9], in which they divide the data stream into chunks, and train a model from one chunk. We refer to these approaches as "chunk-based" approaches. An ensemble of chunk-based models is used to classify unlabeled data. These approaches usually keep a fixed-sized ensemble, which is continuously updated by replacing an older model with a newly trained model. Some chunk-based techniques, such as [9], cannot detect novel classes, whereas others can do so [7]. Chunk-based techniques that cannot detect novel classes cannot detect recurrent classes as well. This is

All Rights Reserved © 2017 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 4, March 2017

because when a class disappears for a while, the ensemble eventually discards all models trained with that class. Therefore, when the class reappears as a recurrent class, none of the models in the ensemble can detect it. On the other hand, chunk-based techniques that can detect novel classes, also cannot detect recurrent classes. This is because a recurrent class is usually identified as a "novel class" by these techniques.

Recurring classes are important to detect, because they create several undesirable effects when falsely interpreted. First, they increase the false alarm rate because when they reappear, they may be either falsely detected as another class, or erroneously identified as novel. Second, when recurring classes are identified as novel, significant computational resources are wasted. This is because novel class detection is a memory and computationally intensive task. It also wastes human efforts, in cases where the output of the classification is used by a human analyst. In such cases, the analyst may have to spend extra effort in analyzing the afore-mentioned false alarms.

In this paper we propose two ensemble techniques overcome the drawbacks of chunk-based ensembles, referred to as a class-based ensemble. For each class c in the stream (seen so far), we keep an ensemble of L micro-classifiers, the details of which will be explained shortly. Therefore, the total number of ensembles is C, where C is the total number of classes seen so far in the stream. The collection of these C ensembles (CL micro-classifiers in total) constitutes the complete classification model.

Next, we briefly discuss the construction and operation of the micro-classifiers. We train r micro-classifiers from a data chunk, where r is the number of classes in the chunk. Each micro-classifier is trained using only the positive instances of a class. During training (see Section 3.1), a decision boundary is built surrounding the training data. The newly trained micro-classifiers update the existing ensemble of models by replacing the old micro-classifiers. The function of an ensemble of micro-classifiers is two-fold. First, it checks whether a test instance falls within the decision boundary of the ensemble. A test instance x falls inside the decision boundary of an ensemble if it falls inside the decision boundary of the majority micro-classifiers in the ensemble. In this case, the ensemble is called bounding ensemble. Second, it outputs a micro classification for a test instance. The micro outputs of all the bounding ensembles are then combined to get the final classification (i.e., class prediction) of a test instance (c.f. Section 3.3).

The contributions of this work are as follows. First, to the best of our knowledge, this is the first work that proposes a class-based ensemble technique to address both the recurring class issue and concept-evolution in data streams. Our proposed solution reduces false alarm rates and overall classification error. Second, this technique can be applied to detect periodic classes, such as classes that appear weekly, monthly, or yearly. This will be useful for a better prediction and profiling of the characteristics of a data stream. Third, we analytically show the impact of the ensemble size and concept-drift on error rates and experimentally confirm this finding. Finally, we empirically show the power of our algorithm over traditional approaches on a number of benchmark data sets, including author recognition textual stream. To enhance the textual features we not only rely on bag of words but also we consider natural language processing techniques such as POS tagging to get additional tags as features.

The remainder of this paper is organized as follows. Section 2 discusses the related works in data stream classification and novel class detection. Section 3 briefly discusses the proposed approach, and Sections 3.1 and 3.3 describes the proposed technique in details. Section 4 reviews the second class-based ensemble technique, SCANR in order to compare in Section 5 where the datasets are reported and experimental results of different approaches are compared, and Section 6 concludes with directions to future work

A. *Related Work*

Most data stream classification techniques handle concept drift using a number of different techniques [2], [4], [5], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Two popular alternatives to handle the massive volume of data streams and concept-drift issues are the single-model incremental approach, and hybrid batch-incremental approach. In the single model approach, a single model is dynamically maintained with the new data. For example, [4] incrementally updates a decision tree with incoming data, and [2] incrementally updates micro-clusters in the model with the new data. The batch incremental approach builds each classification model using a batch learning technique. However, older models are replaced by newer models when the older models become obsolete ([5], [6], [9], [10], [11]). Some of these hybrid approaches use a single model to classify the un-labelled data (e.g., [10]), whereas others use an ensemble of models (e.g., [5], [9]). The advantage of the hybrid approaches over the single model incremental approach is that the hybrid approaches require much simpler operations to update a model (such as removing a model from the ensemble). However, none of these techniques are capable of detecting novel classes and most of these approaches also fail to detect recurring classes, in which a class disappears from the stream and then reappears after a while.

Another branch of recently developed enhanced data stream classification techniques deals with concept-evolution, in addition to one-pass learning and concept-drift. Spinosa et al. [18] apply a cluster-based technique to detect novel classes in data streams. Their approach builds a "normal

All Rights Reserved © 2017 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 4, March 2017

model" of the data using clustering, defined by the hypersphere encompassing all the clusters of normal data. This model is continuously updated with stream progression. If any cluster is formed outside this hypersphere, which satisfies a certain density constraint, then a novel class is declared. However, this approach assumes only one "normal" class, and considers all other classes as "novel". Therefore, it is not directly applicable to multi-class data stream classification, since it corresponds to a "one-class" classifier. Furthermore, this technique assumes that the topological shape of the normal classes in the feature space is convex. This may not be true in real data.

Gomes et al. [19] presented an ensemble-based classification framework. The authors use drift detection methods which monitors error-rate to detect concepts drift. To minimize the space required to store models, only the most relevant features are selected. Katakis et al. [20] presented an ensemble technique for classifying email stream data with recurring concept drifts (REDLLA). The ensemble consists of incremental classifiers. Abad et al. [21] presented a framework for managing recurrent concept drifts. The approach stores context information in meta-models by learning the previous concept drifts. Those meta-models are used to predict when a drift is going to happen in near future and if that drift is a recurrent one. Li et al. [22] presented a semisupervised classification method for streaming data with recurring concept drifts. Unlike our approach, REDLLA does not use ensembles however it creates decision tree. When new a data point is received, the decision tree stores it in a corresponding leaf based on the values of its features. When the number of points in a leaf exceeds a predefined threshold, the leaf points are clustered using k-means clustering method. These clusters are used to produce concepts and to label unlabeled data using majority class classification method. Concept drift is detected if the distance between the centroids of the new data cluster and reference cluster is greater than the sum of their radius. These approaches adopt additional methods to manage both concept drifts and model updating. However, in our approach we use only accuracy statistics to detect drift and update the ensembles. This reduces the overhead imposed on the classification framework.

B. *Our old Work*

Our previous work [7] proposed a classification and novel class detection technique called ECSMiner that is a multi-class classifier and also a novel class detector. However, this approach does not consider the issue of recurring classes. When a class disappears from the stream for a long time and again reappears, ECSMiner identifies it as a novel class. In this paper we show analytically and imperially that our proposed method outperforms ECSMiner. In a more recent work [23] we addressed the recurring class problem by distinguishing between novel class and recurring classes. But it incurs high error (see Section 5) due to the complex structure and application of the ensembles.

In this paper, we compare our techniques with the previous approaches and show the effectiveness of our technique in classification and novel class detection on benchmark data streams. Different than our previous works [1], we have considered a new textual data set for the author recognition (IMDB62) first time. Here, for feature extraction, we not only considered bag of words but also natural language processing guided POS tagging features.

In this paper, we first introduce preliminary information about class-based ensemble and details of its notation on the next section. After the preliminary information we introduce the two recurring class aware techniques. The first class-based ensemble technique, we call CLAM is in Section 3. Just after the CLAM technique we introduce our second class-based ensemble technique, we call SCANR in Section 4. Finally the comparison of chunkbased and class-based ensemble techniques on competitive data sets is placed.

II. OVERVIEW OF CLAM

First we give an informal definition of the data stream classification problem. We assume that a data stream is a continuous flow of data, and that data arrive in chunks, as follows:

D1 ¼ x1;...;xS;D2 ¼ xSþ1;...;x2S;

; ...;Dn ¼ xðn1ÞSþ1;...;xnS;

where xi is the ith instance (i.e., data point) in the stream, S is the chunk size, Di is the ith data chunk, and Dn is the latest data chunk. Assuming that the class labels of all the instances in Dn are unknown, the problem is to predict their class labels. Let yi and y^i be the actual and predicted class labels of xi, respectively. If y^i ¼ yi, then the prediction is correct; otherwise it is incorrect. The goal is to minimize the prediction error. The predicted class labels are then used for various purposes depending on the application. For example, in a credit card fraud detection problem, each transaction is an instance (data point) and is classified (i.e., class label is predicted) as either "authentic", or "fraud" by a data stream classification technique. If the predicted class is "fraud", action is taken immediately to block the transaction and notify the card holder. Often, the classification model makes mistakes in prediction, which is discovered when the card holder examines his card statement and reports incorrect predictions (such as "fraud" transactions predicted as "authentic" or vice versa). This feedback from the card holder can be considered as "labeling" of the past data. These labeled data are then used to refine the classification model.

All Rights Reserved © 2017 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 4, March 2017

A.Training and Building the Decision Boundary

Recall that each training chunk is first split into r disjoint partitions fs1;...;srg of instances based on class labels, where r is the total number of classes in the chunk (algorithm 1, line 8). Therefore, partition si contains only the i-th class instances, and so on (see also Fig. 1). Each such partition sb is then used to train a Mb (Algorithm 1, line 9) as follows. We use K-means clustering to build K clusters using the instances of a partition sb. For each cluster H, we compute a summary h, containing i) m the centroid, ii) r: the radius of H defined as the distance between the centroid and the farthest data point in H, and iii) n: the number of instances in H. The summary h is called a micro-cluster. Micro-clusters have been applied in several stream mining algorithms (e.g., Clustream [24]). After computing the micro-clusters, the raw data points are removed. The set of micro-clusters for each partition sb constitute Mb.

B. *Training and Ensemble Construction in SCANR*

A decision boundary (to be explained shortly) is built during training, which decides whether an instance is outlier. Also, each ensemble (primary and auxiliary) undergoes modification when a new model is trained from the training data. Similar to the CLAM approach, the classification model is based on K-NN and the summary of pseudopoints of the classification model contains the centroid, radius, and frequencies of data points belonging to each class which corresponds to a "hypersphere" in the feature space.

C.Evaluation

Evaluation approach: We used the following performance metrics for evaluation: Mnew ¼ % of novel class instances misclassified as existing class, Fnew ¼ % of existing class instances Falsely identified as novel class, OTH ¼ % of existing class instances misclassified as another existing class, ERR ¼ Average misclassification error (percent) (i.e., average of OTH, Mnew and Fnew). We build the initial models in each method with the first three chunks. Starting from chunk four, we first evaluate the performances of each method on that chunk, then use that chunk to update the existing models. The performance metrics for each chunk are saved and averaged for each method to produce the summary result.

III.Algorithms

Algorithm 1. Class Based Ensemble (CLAM)

1: Build initial ensembles E=fE1;...;ECg with first init num chunks

2: while stream not empty do

3: Dn Latest data chunk in stream for all instance x 2 Dn do

4. Classify(E,x)

5://After the instances in Dn have been labeled by human experts

6: if there is a novel class in Dn then C ¼ C þ 1 for b=1

7: to C do

8: sb all class b instances in Dn //Partitioning if sb ¼6

9:null then Mb Train-classifier (sb)

10: Eb Eb b,sb)

11:Update-ensemble(,M end for

12: end while

IV.CONCLUSION

We have proposed a novel ensemble technique, which is superior to other data stream classification techniques because of its ability to detect novel class, and distinguish a recurring class from novel class. A recurring class is a class that disappears from the stream for a long time and reappears. Existing data stream classification techniques misclassifies a recurring class as another class, or identifies it as a novel class, they forget the class during its absence.
Our proposed approach can be considered a class-based ensemble, as opposed to the chunk-based ensemble that is more popular in data stream classification. The class-based ensemble creates an ensemble of models for each class and each such model is called a micro-classifier. The proposed approach CLAM has been very successful in detecting novel classes, and preventing recurring classes from being detected as a novel class, thereby increasing the classifier accuracy. We have shown both analytically and empirically that the CLAM approach outperforms the chunk-based ensemble approaches both in classification and novel class detection. We have also experimented our approaches with the real benchmark textual data sets including internet movie reviews for author attribution. In the future, we will enhance our approach to other base learners. We will extend our work for multi label text classification areas where a data point may have more than one class labels instead of single label.

V.References

1. T. Al-Khateeb, M. M. Masud, L. Khan, C. Aggarwal, J. Han, and B. Thuraisingham, "Stream classification with recurring and novel class detection using class-based ensemble," in Proc.

All Rights Reserved © 2017 IJARTET

IEEE 12th Int. Conf. Data Mining, 2012, pp. 31–40.

2. C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for on-demand classification of evolving data streams," IEEE Trans. Knowl. Data Eng., vol. 18, no. 5, pp. 577–589, May 2006.
3. J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Comput. Surveys, vol. 46, no. 4, pp. 44, 2013.
4. G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2001, pp. 97–106.
5. J. Kolter and M. Maloof, "Using additive expert ensembles to cope with concept drift," in Proc. 22nd Int. Conf. Mach. Learn., Aug. 2005, pp. 449–456.
6. A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavalda, "New ensemble methods for evolving data streams," in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2009, pp. 139–148.
7. M. M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, "Classification and novel class detection in concept-drifting data streams under time constraints," IEEE Trans. Knowl. Data Eng., vol. 23, no. 1, pp. 859–874, Jun. 2011.
8. M. M. Masud, Q. Chen, L. Khan, C. C. Aggarwal, J. Gao, J. Han, and B. M. Thuraisingham, "Addressing concept-evolution in concept-drifting data streams," in Proc. IEEE Int. Conf. Data Mining, 2010, pp. 929–934.
9. H. Wang, W. Fan, P. S. Yu, and J. Han. (2003). Mining conceptdrifting data streams using ensemble classifiers, in Proc. Int. Conf. Knowl. Discovery Data Mining [Online]. pp. 226–235. Available: http://portal.acm.org/citation.cfm?id=956750.956778
10. Y. Yang, X. Wu, and X. Zhu, "Combining proactive and reactive predictions for data streams," in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2005, pp. 710–715.
11. J. Gao, W. Fan, and J. Han., "On appropriate assumptions to mine data streams." in Proc. Int. Conf. Data Mining, 2007, pp. 143–152.
12. S. Hashemi, Y. Yang, Z. Mirzamomen, and M. Kangavari, "Adapted one-versus-all decision trees for data stream classification," IEEE Trans. Knowl. Data Eng., vol. 21, no. 5, pp. 624–637, May 2009.
13. P. Zhang, X. Zhu, and L. Guo, "Mining data streams with labeled and unlabeled training examples," in Proc. 9th Int. Conf. Data Mining, 2009, pp. 627–636.
14. G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," Mach. Learn., vol. 23, no. 1, pp. 69–101, Apr. 1996.
15. P. Wang, H. Wang, X. Wu, W. Wang, and B. Shi, "A low-granularity classifier for data streams with concept drifts and biased class distribution," IEEE Trans. Knowl. Data Eng., vol. 19, no. 9, pp. 1202–1213, Sep. 2007.

All Rights Reserved © 2017 IJARTET