Novel Class Detection in Concept Drifting Data Streams Using Decision Tree Leaves

Deepita Saha
Md Mozzammel Haque
Akash Sarkar
Famina Alam

Department of Computer Science and Engineering

United International University

A thesis submitted for the degree of

BSc in Computer Science & Engineering

October 2017

Abstract

Concept drifting data streams often occurs in weather forecasting, intrusion detection and other applications. One of the difficulties with handling concept drifting data streams is the existence of novel classes in the data stream that arrives after the training of the model on the existing class instances. In this thesis, we present a novel class detection algorithm in concept based on the instance distribution in the decision tree leaves. Our proposed algorithm is easy to implement and use compared to complex ensemble based methods. We have tested the performance of our algorithm on several datasets and it shows significantly improved results compared to previous state-of-the-art algorithm using standard evaluation methods and metrics.

In memory of your family/ friends, like your mother or father.

Acknowledgements

We are grateful to the God for the good health and wellbeing that were necessary to complete this book.

We wish to express our sincere thanks to Chowdhury Mofizur Rahman, Vice Chancellor, for providing us with all the necessary facilities for the research. We place on record, our sincere thanks you to Dr. Dewan Md. Farid , Associate Professor, Department of Computer Science and Engineering, for the continuous encouragement.

We are also grateful to Dr. Swakkhar Shatabda, Associate Professor & Undergraduate Program Coordinator, in the Department of Computer Science and Engineering. We are extremely thankful and indebted to him for sharing expertise, and sincere and valuable guidance and encouragement extended to us.

We take this opportunity to express gratitude to all of the Department faculty members for their help and support. We also thank our parents for the unceasing encouragement, support and attention. We are also grateful to our partner who supported us through this venture.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

With the advent of modern computing technologies and communication through the set up of Internet of Things (IoT), there has been a enormous growth in continuous data streams resulting in many difficulties and challenges to deal with them [1, 2]. One of the most challenging tasks in handling data streams in the classification with concept drifts. Concept drifts in the data streams does not guarantee success of any pre-learned model or predictor. Concept drift in data streams occur for many reasons, including change in the distribution [3]. One of the major challenges in concept drifting data stream classification is to establish a prediction model or algorithm, when novel classes appear during the concept drift. Such cases often occur in many real life scenarios like intrusion detection, email spam detection, sensory data, etc.

Most of the traditional methods fail in handling concept drifting data with novel classes since they are pre-trained on existing classes and thus fails to predict novel class instances correctly that arrives later in the data stream. In the literature of classification of instances in concept drifting data streams, often adaptive models and ensemble based methods [4]. However, one of the major difficulties with ensemble based methods is that they are computationally expensive. Often simple base classifiers intelligently designed [5] are able to perform better in the case of data streams. This is due to the added difficulties in storing new instances and retrain of the model. Simple base classifiers are easy to train and maintain and often preferable to complex models like ensemble methods.

1.1 Contribution of the thesis

In this thesis, we propose a decision tree based algorithm to detect novel classes in concept drifting data streams. Our proposed algorithm firstly trains on the existing class instances using a simple decision tree. Later on, during the testing phase, it detects the novel classes based on the distribution of the instances in the leaves of the learned decision tree model. We have used several benchmark datasets to test the effectiveness of our algorithm. Experimental methods shows that our algorithm improves over the previous state of the art methods using standard evaluation methods and metrics on concept drifting data streams.

1.2 Thesis Organization

Rest of the thesis is organized as following: Chapter 2 presents necessary background knowledge required for novel class detection and decision tree learning algorithm; Section 2.3 briefly presents related work in the literature; Chapter 3 presents our novel class detection algorithm; Chapter 4 presents experimental results and discussion and Chapter 5 concludes the paper with a summary and outline of future work.

Chapter 2

Background

In this chapter, we provide a brief introduction to the novel class detection in concept drifting data streams and decision tree algorithm.

2.1 Novel Class Detection

One of the major concept drift in the data streams occur due to change in the class definitions or the posterior distribution membership of the class membership $P(C|x)$ [3]. Often that results in appearance of novel classes in the stream. In such cases, models trained on existing class instances fails miserably. A typical scenario of appearance of novel class instances in data streams is shown in Fig. 2.1. We could see that a previously trained machine learning algorithms fails when a new class of instances appear since it tries to classify them to existing classes.

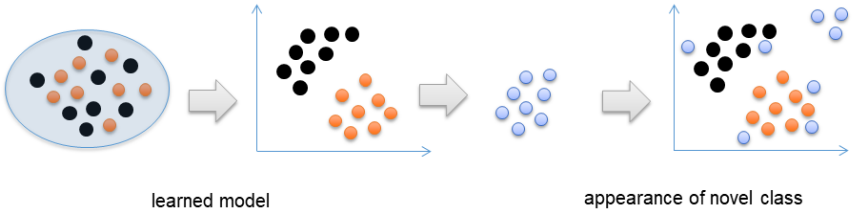

Figure 2.1: Appearance of novel classes in concept drifting data streams.

2.2 Decision Tree Learning

Decision tree is a widely used algorithm used in machine learning [6] that recursively builds a tree by selecting a suitable attribute for the nodes of the tree and splits the dataset using the values of the selected attribute. Pseudo-code of a basic decision tree learning algorithm is depicted in Algorithm 1. Information gain is often used as the criteria to select the attribute to split. Information gain, $Gain(A)$ for any attribute A on a dataset D is defined as in the following equation.

$$Gain(A) = Entropy(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} \times Entropy(D_i) \tag{2.1}$$

Here, D_i is the induced sub dataset from the original dataset D, using the ith value of the attribute A. Entropy is the statistical measure of the disorder in the resulting set defined as in the following equation.

$$Entropy(D) = -\sum_{i}^{m} p_i log(p_i) \tag{2.2}$$

Algorithm 1: BuildDecisionTree(Training Data D)

```
1  Select the attribute to split the dataset
2  Create a tree T with root the the selected attribute
3  for each values of the selected attribute do
4      D' is the sub dataset using the split
5      if termination condition is satisfied then
6          T' is a leaf node with appropriate label
7      else
8          T' ← BuildDecisionTree(D')
9      add T' to T as an edge in the tree
10 return T
```

2.3 Related Work

Most of the methods that are used in the literature of novel class detection in the literature are ensemble methods [4, 7–10]. Masud et al. [7] used a buffer to save the

new instances and later use them to re-train the ensemble of classifiers by updating decision boundaries. Liu et al. [9] used a fuzzy membership function for base classifiers in an ensemble using a majority voting for handling concept drifting data streams. Farid et al. [4] presented an adaptive algorithm using ensemble classifiers for novel class detection in concept drifting data streams. In their proposed method, they used clusters of instances and using the distance of the new instances from the existing instances in the clusters, novel classes were detected. They tested their algorithms on datasets taken from UCI machine learning repository.

Sidhu et al. [8] uses an ensemble method to detect concept drifts in data streams by comparing accuracy of the ensemble method on the recent methods with that of the accuracy starting from the beginning. In a recent work [10], the same authors proposed a dynamic weighted majority based ensemble method to detect concept drifts in data streams.

Farid et al. [5] proposed a decision tree based novel class detection algorithm where the distribution of the instances in the leaf nodes were used to detect the presence of novel classes in the data stream. In another work [11], the authors constructed a neighborhood graph and used local patterns to enhance the prediction capability of an ensemble of classifiers. Their method analyzed the connected components of the constructed graphs and tested the performance of the algorithm on synthetic and standard benchmark datasets.

Unsupervised methods are also employed to detect concept drifts. Reis et al. [12] used incremental Kolmogorov-Smirnov test on the input distribution of the data stream to detect concept drifts. Gamma et al. [3] provided an early work on review of the machine learning methods on concept drifting data streams. A few other work also discusses challenges and issue in concept drifting data streams [1, 2, 13, 14].

One of the issues in performing research on concept drifting data streams is with the absence of standard available datasets, evaluation methods and metrics. In [4] used a set of metrics particularly for multi-class settings of novel class detection in concept drifting data streams. Faria et al. [15] also emphasized on the methodology and presented several other metrics and methods in the multi-class setting. The testing scheme and preparation of datasets employed in [5] was also explored in later works [4, 15].

Chapter 3

Proposed Method

In this chapter, we detail our proposed method for novel class detection. Our algorithm is based on the basic decision tree. All the instances in the training phase of the decision tree are split by the attributes selected at the nodes of the decision tree and falls into the leaves. A pre-trained decision tree without any knowledge of novel classes performs this task based only on the existing classes. After the training phase is over the data stream with novel classes will provide the trained model in the first phase with instances of the new class. Unaware of the novel class the pre-trained model will fail to detect this and classify it as existing class. Thus traditional decision tree algorithms performs poorly on concept drifting datasets with novel classes. We have modified the prediction algorithm that uses a pre-trained decision tree. A typical prediction method using a trained decision algorithm works on any instance starting from the root attribute. In turns it follows the edges from each node attributes and falls into a leave were the decision is made.

Our algorithm is based on the hypothesis that the instances of the new class must be dissimilar to the existing class instances. The decision tree learning algorithm itself uses a termination criteria at each leaf node where all the instances are labeled to a single class. Now any instance falling into this particular leaf in the testing phase will also be classified as the same. However, any novel class instance falling into the same leaf must be dissimilar to the instances of the existing class. The dissimilarity in these instances can be utilized to detect the appearance of novel class instances. In order to capture this, instances that fall into the same leaf are thought to be in the same cluster. These clusters are created during the training phase and cluster centers are determined

for each of the leaf nodes. These cluster centers and the distance of the furthest instance from the cluster center is stored for later use in the novel class detection phase or testing phase. The pseudo-code of this modified decision tree learning algorithm is given in Algorithm 2. The algorithm first builds a decision tree using Algorithm 1. Then it stores the tree and calculates cluster centers, C_i of the instances falling into each leaf i and the average distance max_i that is the maximum distance from the center to all the instances in the cluster. This particular instance actually denotes an instance in the proximity of the cluster center and representative of the decision boundary. Now, any instance falling into this leaf using the stored tree T will be labeled as preset class label L_i.

Algorithm 2: ModifiedDecisionTreeLearningAlgorithm(Training Data D)

```
1 T is the decision tree learned using Algorithm 1
2 for each leaf i in T do
3     D_i is the set of instances falling into leave i
4     C_i is the cluster center for the instances in D_i
5     max_i is the maximum distance from C_i to all the instances in D_i
6     L_i is the label set for leaf i
7     store ⟨C_i, max_i, L_i⟩ for novel class detection
```

In the novel class detection phase, the previously stored tree T and the cluster related information are put into use. First of all, an instance is fed to the decision tree to find the leaf i that the instance falls into following the path starting from the root attribute as done in traditional decision making methods using decision trees for classification. The traditional algorithms will return the associated label L_i with this leaf node i. Our novel class detection differs here from the traditional algorithms. It then uses the stored information about the cluster center C_i and the maximum distance or decision boundary distance max_i to detect novel classes. If the distance between the test instance and the cluster center exceeds the distance max_i, a novel class is detected, otherwise it is treated as an existing class instance. The pseudo-code for this novel class detection algorithm is given in Algorithm 3.

The algorithm here is a simple extension of the typical decision tree classification algorithm. Note that we have used Manhattan distance as the distance measure in

Algorithm 3: NovelClassDetection(Instance x, Decision Tree T)

```
1 Apply decision tree T for instance x
2 i is the leaf of the tree returned by tree T
3 if distance(C_i, x) > max_i then
4     detect novel class
5 else
6     return existing class label L_i
```

this paper. Manhattan distance between two instances x_i and x_j are defined as in the following equation.

$$distance(x_i, x_j) = \sum_{k}^{N} |x_i(k) - x_j(k)| \tag{3.1}$$

Chapter 4

Experimental Analysis

We have used Python 3.6 for all of our experiments. Experiments were conducted on a machine with core i7 processor with 8GB of ram running Ubuntu 18.04. For experiments we used Scikit-learn machine learning library [16].

4.1 Benchmark Datasets

In this paper, we have used two datasets taken from the UCI machine learning repository that are widely used in the literature and previous used in the context of novel class detection in concept drifting datasets [4, 5, 15]. These two datasets are Iris and Soybean datasets. A summary of the datasets is given in Table 4.1.

Table 4.1: Summary of Datasets.

Dataset	No of Attributes	Attribute Type	Total Instances	No of Classes	Missing Values
Iris	4	Real	150	3	None
Soybean	35	Real	683	19	None

4.2 Preparation of Datasets

To make these datasets appropriate for concept drift task, we have followed the same methodology applied in [4, 5]. Each of the datasets were split into test and train based

on the number of instances in each classes. In the case of Iris dataset, since there are 3 classes, we made one class novel and unknown in the training data. Thus 2 classes were considered as existing class and the other one was considered as novel class. The resulting dataset contained 94 training instances in the train set and the rest in the test set. In case of the Soybean dataset, 15 classes were set as existing classes and 4 were set as novel classes. We have made all the preprocessed datasets used in this paper available from `https://gitlab.com/deepitauiu/concept-drift-datasets`.

4.3 Evaluation Metrics

We used the same set of metrics used in [4, 5] as we have compared the performance of our algorithm with those methods. First metrics used is called percentage of novel class instances misclassified as existing class, M_{new} defined as following:

$$M_{new} = \frac{F_n * 100}{N_c} \tag{4.1}$$

Here, F_n is the total novel class instances misclassified as existing class and N_c is the total number of instances of the novel class in the data stream. The next metrics called percentage of existing class instances falsely identified as novel classes, F_{new} is defined as following:

$$F_{new} = \frac{F_p * 100}{N - N_c} \tag{4.2}$$

Here, N is the total number of instances and F_p is the total existing class instances misclassified as novel classes. The last metrics is called total misclassification error ERR defined below:

$$ERR = \frac{(F_p + F_n + F_e) * 100}{N} \tag{4.3}$$

Here, F_e is the total existing class instances misclassified.

4.4 Results

We have shown comparison among different algorithms on the performance measures following the same methodology on both of the datasets. We have compared the performance of our proposed algorithm with four other algorithms: ensemble method proposed in [4], decision tree based method proposed in [5], traditional decision tree

Table 4.2: Comparison of results among different algorithms.

Classifier	**Dataset**	ERR	M_{new}	F_{new}
Proposed Method	Iris	**1.84**	**2.04**	**0.0**
	Soybean	11.86	**0.0**	10.56
Ensemble Method [4] (EM)	Iris	–	–	–
	Soybean	2.0	0.0	2.0
Ensemble Method [4] (C4.5)	Iris	–	–	–
	Soybean	4.0	5.0	1.0
Ensemble Method [4] (k-NN)	Iris	–	–	–
	Soybean	5.1	5.6	1.9
Decision Tree Based Method [5]	Iris	3.3	6.0	0.0
	Soybean	15.3	16.3	4.3
Traditional Decision Tree	Iris	6.6	6.0	2.0
	Soybean	21.4	27.2	6.7
k-NN	Iris	20.0	18.0	7.0
	Soybean	25.0	30.9	7.5

method and k-Nearest neighbor algorithm. Note that for our case, we run our algorithm for 5 times and only the average of the results found are reported. In case of the other algorithms, we took the results as they were reported in the published papers in [4, 5]. The results are shown in Table 4.2. The bold values in the table shows where our algorithm performed best. Note that, the ensemble method proposed in [4] did not use the iris dataset and this that particular row is missing. Also note that there were different versions of the ensemble method using three different base classifiers: EM algorithm, C4.5 and k-NN.

Note that, out algorithm performed best for all the cases in the case of Iris dataset. However, in the case soybean dataset, our algorithm performed better in terms of novel class detection error M_{new} which is very important in this context. To further test the performance we compared the total classification accuracy and novel class detection

Table 4.3: Comparison in terms of classification accuracy among different algorithms on Soybean Dataset.

Classifier	Instances	Existing Classes	Novel Classes	Novel Class Detection Accuracy	Total Accuracy
Proposed Method	Train:572 Test:105	15	4	**100.00%**	**96.67%**
Ensemble Method [4]	Train:630 Test:53	15	4	100.0%	93.23%

rate on the soybean dataset with the ensemble method [4]. Here, we can observe that our algorithm is performing better in terms of overall classification accuracy. Note that, compared to the ensemble method, our decision tree is a very light weighted and simple and it is possible to extend this basic decision tree algorithm to be used as ensembles and performance comparison will make more sense in that setting. However, we put that as a future work to this work.

Chapter 5

Conclusion

In this thesis, we have proposed an effective algorithm for detection of novel classes in concept drifting data streams using instance distribution in decision tree leaves. On several benchmark datasets, our proposed algorithm shows significantly improved performances using standard evaluation methods and metrics. We have tested the performance of our algorithm on datasets widely used for traditional classification problems. In future, we wish to test the performance of the algorithm on real concept drifting data streams and on other several benchmark datasets used in the literature. We also wish to analyze the performance of the algorithm using different distance measures. Moreover, we wish to develop an adaptive decision tree algorithm based on the proposed algorithm for classification of concept drifting data streams.

Bibliography

[1] M. M. Masud, C. Woolam, J. Gao, L. Khan, J. Han, K. W. Hamlen, and N. C. Oza, "Facing the reality of data stream classification: coping with scarcity of labeled data," *Knowledge and information systems*, vol. 33, no. 1, pp. 213–244, 2012. 1, 5

[2] J. Read, A. Bifet, G. Holmes, and B. Pfahringer, "Scalable and efficient multi-label classification for evolving data streams," *Machine Learning*, vol. 88, no. 1-2, pp. 243–272, 2012. 1, 5

[3] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian symposium on artificial intelligence*. Springer, 2004, pp. 286–295. 1, 3, 5

[4] D. M. Farid, L. Zhang, A. Hossain, C. M. Rahman, R. Strachan, G. Sexton, and K. Dahal, "An adaptive ensemble classifier for mining concept drifting data streams," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5895–5906, 2013. 1, 4, 5, 9, 10, 11, 12

[5] D. M. Farid and C. M. Rahman, "Novel class detection in concept-drifting data stream mining employing decision tree," in *Electrical & Computer Engineering (ICECE), 2012 7th International Conference on*. IEEE, 2012, pp. 630–633. 1, 5, 9, 10, 11

[6] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. 4

[7] 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 Transactions on Knowledge and Data Engineering, vol. 23, no. 6, pp. 859–874, 2011. 4

[8] P. Sidhu and M. Bhatia, "An online ensembles approach for handling concept drift in data streams: diversified online ensembles detection," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 6, pp. 883–909, 2015. 5

[9] A. Lui, G. Zhang, and J. Lu, "A novel weighting method for online ensemble learning with the presence of concept drift," in *Decision Making and Soft Computing: Proceedings of the 11th International FLINS Conference*. World Scientific, 2014, pp. 550–555. 5

[10] P. Sidhu and M. Bhatia, "A novel online ensemble approach to handle concept drifting data streams: diversified dynamic weighted majority," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 1, pp. 37–61, 2018. 4, 5

[11] P. ZareMoodi, H. Beigy, and S. K. Siahroudi, "Novel class detection in data streams using local patterns and neighborhood graph," *Neurocomputing*, vol. 158, pp. 234–245, 2015. 5

[12] D. M. dos Reis, P. Flach, S. Matwin, and G. Batista, "Fast unsupervised online drift detection using incremental kolmogorov-smirnov test," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1545–1554. 5

[13] T. S. Sethi and M. Kantardzic, "Handling adversarial concept drift in streaming data," *Expert Systems with Applications*, vol. 97, pp. 18–40, 2018. 5

[14] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014. 5

[15] E. R. Faria, I. J. Gonçalves, J. Gama, and A. C. Carvalho, "Evaluation methodology for multiclass novelty detection algorithms," in *Intelligent Systems (BRACIS), 2013 Brazilian Conference on*. IEEE, 2013, pp. 19–25. 5, 9

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011. 9