Novel Class Detection in Data Streams

Vahida Attar and Gargi Pingale

Abstract Data stream classification is challenging process as it involves consideration of many practical aspects associated with efficient processing and temporal behavior of the stream. Two such aspects which are well studied and addressed by many present data stream classification techniques are infinite length and concept drift. Another very important characteristic of data streams, namely, concept-evolution is rarely being addressed in literature. Concept-evolution occurs as a result of new classes evolving in the stream. Handling concept evolution involves detecting novel classes and training the model with the same. It is a significant technique to mine the data where an important class is under-represented in the training set. This paper is an attempt to study and discuss the technique to handle this issue. We implement one of such state-of-art techniques and also modify for better performance.

Keywords Novel class detection · Data stream classification · Concept evolution

1 Introduction

Advances in electronics and hardware technology have highly increased the capability to generate as well as to store enormous data in various forms. Mobile phones, laptops, easily available camera, the Internet are few among the various devices which generate huge amount of digital data. Though data mining provides with many good applications to mine this huge data, data stream classification is a major challenge for data mining community. With increasing volume of data it is no longer possible to process the data in multiple passes. Also the data stream may evolve over the time

V. Attar (✉) · G. Pingale
College of Engineering Pune Shivaji Nagar, Pune, India
e-mail: vahida.comp@coep.ac.in

G.Pingale
e-mail: gargipingale04@gmail.com

B. V. Babu et al. (eds.), *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28–30, 2012*, Advances in Intelligent Systems and Computing 236, DOI: 10.1007/978-81-322-1602-5_73, © Springer India 2014

period. Concept-evolution occurs when new classes evolve in the stream. Handling concept evolution involves detecting novel classes and training model with the same. When we are using continuous data stream for classification it is not possible to predict the class or type of data in advance that system is going to encounter. System should be able to detect an emergence of new class of data in stream.

Novel class detection is concerned with recognizing inputs that differ in some way from those that are usually seen or which already exist in the trained model. It is a significant technique to mine the data where an important class is under-represented in the training set. Traditional classifiers cannot detect presence of novel class. All the instances which belong to such class are misclassified by the algorithm unless there is some kind of manual intervention and model is trained with such a novel class. Novel class detection technique is immensely useful in medical sciences, intrusion detection, fraud detection, signal processing, and image analysis. Novel class detection refers to learning algorithms being able to identify and learn new concepts. In this case, the concepts to be detected and those to be learned correspond to an emerging pattern that is different from noise, or drift in previously known concepts. In simple words, systems detecting novel class must be able to identify new concept emerging in the stream and train the existing models with it so that in future instances belonging to that novel class can be correctly classified. Novelty class detection is now being under constant attention from researchers and academicians. Various approaches to detect the new concept as well as to learn the new concept have been devised.

In rest of the paper Sect. 2 describes about Novel class Detection. In Sect. 3, we explore different technique to handle this issue, Sect. 4 is about the implemented techniques and improvements over the same. Results of both the techniques are presented and compared in Sect. 5 and conclusion in Sect. 6.

2 Overview of Concept Evolution and Novel Class Detection

There are many algorithms in literature that address novelty detection. These algorithms can be divided into one-class approach and multi-class approach. In one class method [6, 7, 9], one class is able to detect a single class of data instances that is different from classes with which system is trained, however, multi-class method can detect more than one new class in training data set. These algorithms show different results for different data sets because their efficiency and accuracy depends on underlying method used and properties of data taken into consideration. Major challenge is to detect novel classes in presence of concept drift in data stream. There are broadly two approaches for novelty detection in data stream, Statistical and Neural Network approach.

Neural networks [6] are immensely used in Novelty Detection. In comparison to statistical methods some critical issues in neural networks are generalization of computational expense during training and expense involved in retraining. Some of these methods are multi-layer perceptions, support vector machines, radial basis function networks, self-organizing maps, oscillatory networks, etc.

Statistical Approach uses the statistical properties of data for creating models. It further uses these models to estimate whether the given instance belongs to the existing class or not. There exist various techniques for novelty detection using this approach. To mention a few, building a density function for data of known class and then calculate the probability of the coming test instance belonging to existing class. Another technique can be finding mean distance of test instance under consideration from the center of nearest cluster of existing class first to detect it as outlier and if there are several such outliers close to each other considering some threshold value for closeness as a novel class. The distance measure can be a Euclidian distance or some other probabilistic distance. Further down there are two types of statistical approach, Parametric and Non-parametric approaches.

In **parametric approach** data distributions are assumed to be known and then the parameters such as mean, variance of model are estimated, the test data falling outside the estimated parameters of distribution are declared as novel. But the parametric approach does not have much practical implications as the distribution of real data is already not known. Some of the parametric methods for novelty detection are Hypothesis testing, Probabilistic/Gaussian mixture modeling.

Non-parametric approach involves estimation of density of data for training for example, Parzen window and K-Nearest Neighbor. The instances which fall out of certain threshold density are regarded as novel. These methods do not make any assumption regarding data distribution functions. Thus in a way they are more flexible. But they do have shortcomings in handling of noise in data. In KNN method, the normal data distribution is defined by a few numbers of spherical clusters formed by k-nearest neighbor technique. Novel class is identified by calculating distance of data point from the center of the clusters which fall beyond its radius. Parzen windows method is used for estimation of data density. It uses Gaussian function. In this method a threshold for detecting novelty is set which is being applied on the probability of test pattern.

3 Techniques for Novel Class Detection

Various papers to deal with the novel class detection are noted in the literature. Smola [5], proposes approach to this problem by trying to estimate a function f which is positive on S and negative on the complement. The functional form of f is given by a kernel expansion in terms of a potentially small subset of the training data; it is regularized by controlling the length of the weight vector in an associated feature space. It provides a theoretical analysis of the statistical performance of algorithm. The algorithm is a natural extension of the support vector algorithm to the case of unlabeled data. Given a small class of sets, the simplest estimator accomplishing this task is the empirical measure, which simply looks at how many training points fall into the region of interest. This algorithm does the opposite. It proposes an algorithm which computes a binary function which is supposed to capture regions in input space where the probability density lives (its support), i.e., a function such that most of the data will live in the region where the function is nonzero. In doing so, it is in line with Vapnik's principle never to solve a problem which is more general than the one we actually need to solve. Moreover, it is applicable also in cases where the density of the data's distribution is not even well-defined, e.g., if there are singular components. It starts with the number of training points that are supposed to fall into the region, and then estimates a region with the desired property.

In [6], a new single class classification technique has been proposed that can detect novelty and handle concept drift. The proposed method uses clustering algorithm to produce the normal model. It relies on Discrete Cosine Transform (DCT) to build compact and effective generative models such that the closest model to a new instance will be an approximation of its K nearest neighbors. Also using the DCT coefficients it presents an effective method for discriminating normal concepts as well as detecting novelty and concept drift. The proposed method referred to as Discrete Cosine Transform Based Novelty and Drift Detection (DETECTNOD), consists of two phases. In the first phase, based on the normal data, it tries to generate an initial model with an effective and compact knowledge about the clusters. At the second phase, the testing data is divided into equal sized blocks whose size is limited only by the storage space. In this phase, using the previously obtained generative models, normal data is discriminated from novel classes and concept drift.

D. Martinez [8] introduced a neural competitive learning tree as a computationally attractive scheme for adaptive density estimation and novelty detection. This approach combines the unsupervised learning property of competitive neural networks with a binary tree-type structure. The initialization process can be performed with input data sampled either randomly from the training set (random initialization) or sequentially as data become available (sequential initialization) in case of an Independently Identically Distributed (IID) sequence and then nodes are splitting each at one time. To avoid this dependency of initialization process on particular data tree is built by taking into account entire dataset and splitting all nodes once. Thus, the learning rule provides an adaptive focusing mechanism capable of tracking time-varying distributions. The constructed tree from the training data serves as a reference tree. Another tree is built for the testing data and a novelty is detected when it differs too much from the reference tree.

Markos Markou [9] proposes a new model of "novelty detection" for image sequence analysis using neural networks. This model uses the concept of artificially generated negative data to form closed decision boundaries using a multilayer perceptron. It uses novelty filter to classify data as known and unknown. One neural network is trained per class where samples are labeled as belonging (positive) or not belonging (negative) to class. Negative data is used for novelty detection. Neural Network with Random Rejects (NN-RR) novelty filter works on thresholding the neural network output activation in response to an input test pattern. Rejected samples are then collected in data storage called bin. Clustering is done using k-means clustering method. This helps in deciding which clusters should be used for retraining.

4 Implemented Technique and Proposed Up-Gradation

In real streaming environment where new classes evolve it is not the case that total number of classes is fixed for classification purpose. Masud et al. [4] propose an algorithm to detect emergence of novel class in the presence of concept drift by quantifying cohesion among unlabeled instances and separation of them from training instances. Traditional novelty detection schemes assume or build a model of normal data and identify outliers that deviate from normal points. This scheme not only detects single point which deviates from normal data but also to find if there is any strong bond among the points. This technique uses ensemble approach to handle concept drift. Data stream is divided into equal sized chunks. Each chunk is accommodated in memory and processed online and classification model is trained from each chunk. Newly trained model replaces original model. And the ensemble evolves representing most up-to-date concepts in the stream. This paper forms the platform of our work. We implement this technique and also propose some improvements in the algorithm.

Input data stream is divided into equal sized chunks. Each unlabeled chunk is given as input to the algorithm. It first detects presence of novel class. Instances belonging to the novel class are separated from the chunk and the remaining instances are classified normally. A new model is trained using the instances of the latest chunk. Finally the ensemble is updated by choosing the best M classifiers from $M+1$ classifiers. The base learners used are K Nearest Neighbors and Decision tree. Clusters of the training instances are built and hence store only cluster summaries. These clusters are called as Pseudo points. Any strong cohesion among the instances falling in the unused space indicates presence of a novel class. Novel class detection is a two step process. Initially the training data is clustered and stored as cluster summaries called Pseudopoints which are used to keep the track of used spaces. Later these Pseudopoints are used to detect outliers and if a strong cohesion exists among the outliers a novel class is declared. Every time the data chunk is clustered and the cluster centroid and relevant information is stored as Pseudopoints. Clustering is specific to each base learner. In case of decision tree at its each leaf node where as for KNN classifier the already existing Pseudopoints are used. K Means clustering approach is adapted for the same. The desired value of K parameter in K Means algorithm should be determined experimentally. We build $K_i = (t_i/S)*K$ clusters in l_i, where t_i denoted number of training instances in the leaf node l_i, S is the chunk size.

Cluster summary is stored in Pseudopoints which consist of Weight (W)—total number of instances in the cluster. Centroid (C). Radius(R)—maximum distance between the centroid and the data instances belonging to the cluster. Mean distance—it is the mean distance from each data instance to the cluster centroid. Once the Pseudopoints are formed the raw data is discarded. The union of regions covered by all Pseudopoints represents union of all the used spaces which forms a decision boundary.

R-Outlier is a data instance such that the distance between the centroid of the nearest Pseudopoint and the instance is greater than the radius of this Pseudopoint.

For KNN approach R-Outliers are determined by testing each data instance against all the Pseudopoints. For decision tree each data point is tested against only the Pseudopoints stored at the leaf node where the instance belongs. So any data instance outside the decision boundary is an R-Outlier for that classifier.

Up-Gradations: Early Novel Class Detection (ENCD)

Sometimes test data instance may be considered as an R-outlier because of one or more reasons like: The test instance belongs to an existing class but it is a noise, shift in the decision boundary, Insufficient training data. To avoid an ordinary instance being declared as a novel class instance filtering is done. If a test instance is an R-Outlier to all the classifiers in the ensemble only then it is considered as filtered outlier that is F-Outlier. Rest all are filtered out. Hence being an F-Outlier is a necessary condition for being in a new class. Detection of novel class basically means to verify whether F-Outliers satisfy the two properties of a novel class that is separation and cohesion. λ_C neighborhood is the set of η nearest neighbours of x belonging to class C where η is user defined parameter. In the existing algorithm a new classifier is built only when the test chunk is completed. As soon as a new class is found a new classifier must be built, instead of waiting for the test chunk to finish, and then to train a classifier. Now we don't need to wait for the test chunk to complete, instead we train the new classifier with whatever part of test chunk at hand and this puts us in a position to detect and identify the forthcoming novel instances in that present chunk itself. To achieve this, we put an additional condition for building a classifier. During testing, whenever a new class is detected, a flag is set. Building or training of new classifier is done when this flag is set or when the test chunk is completed. We call this as Early Novel Class Detection (ENCD).

5 Experiments and Results

5.1 *Datasets*

10 % of KDDCup 99 network intrusion detection [5] contains around 4,90,000 instances. Here different classes appear and disappear frequently, making the new class detection very challenging. There are 22 types of attacks, each record consists of 42 attributes. We have also used synthetically generated dataset from [4] which simulates both concept-drift and novel-class. The data size varies from 100 to 1000 k instances, class label varies from 5 to 40 and data attributes from 20 to 80 (Fig. 1).

5.2 *Implementation Environment*

The proposed algorithm is implemented in Java programming language on Linux platform. We have used MOA-Massive Online Analysis tool for all the experimen-

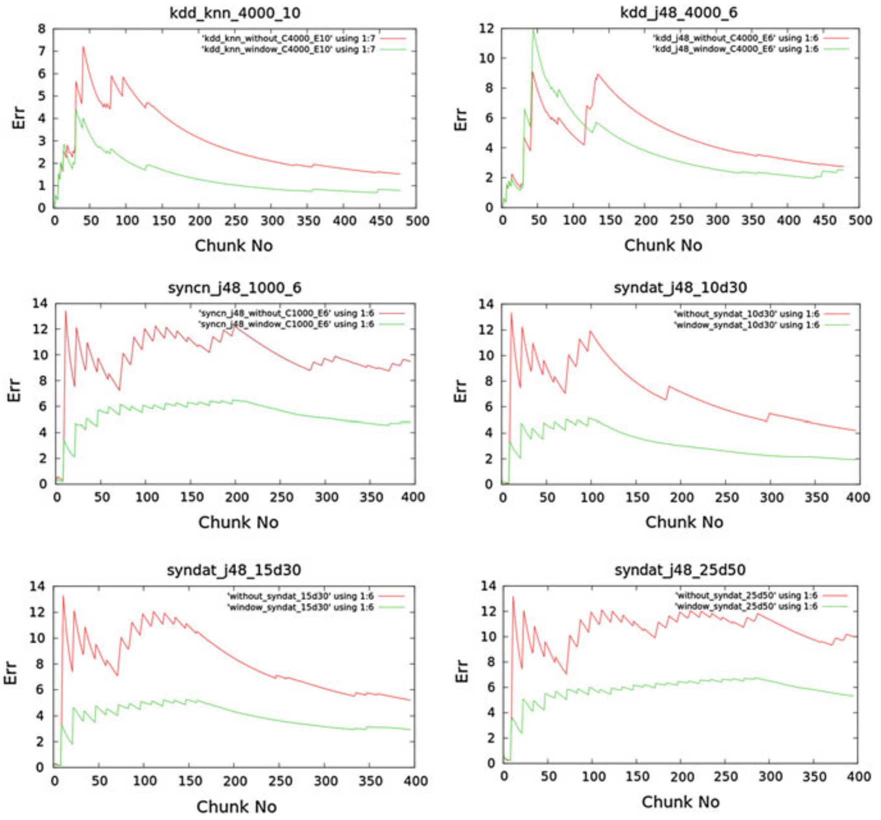

Fig. 1 Error rate versus Chunk No. for Datasets: KDD (KNN, DT), SyncCN ()

tation of the proposed approach. Performance existing algorithm is compared with ETNCD proposed by us. For plotting the graphs of chunk no. and error rate of the two algorithms, we used GNU plot version 4.2.

Parameters-Error: Whenever predicted class value of the instance under consideration is different from its actual class value error is noted. If an existing class instance is misclassified as a novel class instance error is incremented.

Global Error = (100 * Err) / total no. of instances in dataset.

Chunk Size: We have experimented with chunk size from 500 to 5,000 and selected 4,000 for real and 1,000 for synthetic dataset.

Ensemble Size: 10 for KNN and 6 for Decision tree classifier.

Minimum Number of Points required to declare a novel class: 50.

5.3 *Results Based on Up-Gradations*

Initially models are built with the first N chunks. From the N + 1 chunk performance of each method is evaluated for that chunk and then the same chunk is used to update the existing models. We compared the proposed approach that is ENCD and the existing algorithm in [4]. We perform comparisons for all the mentioned datasets.

6 Conclusion and Future Work

Various techniques in novel class detection have been studied and analyzed. We propose early detection and identification of every new class. This improvement gives us an edge to classify the novel instances correctly. Experimental results show that the proposed up gradations improve the existing algorithm. The current technique does not take into consideration multiple-label instances. So we would also like to apply our technique to multiple-label instances.

References

1. Mohammad, M.M., Jing, G., Latifur, K., Jiawei, H., Bhavani, M.T.: Classification and novel class detection in concept drifting data streams under time constraints.In: Preprints, IEEE Transactions on knowledge and Data Engineering (TKDE), 2010.
2. Mohammad, M., Masud, J.G., Latifur, K., Jiawei, H.: Bhavani. Classification and novel class detection in data streams in a dynamic feature space, M.T. (2010)
3. Mohammad, M.M., Jing, G., Latifur, K., Jiawei, H.: Bhavani. Classification and novel class detection in data streams with active mining, M.T. (2010)
4. Mohammad, M.M., Jing, G., Latifur, K., Jiawei, H.: Bhavani. M.T , Integrating novel class detection with classification for concept drifting data streams (2009)
5. Smola, A.J., Shawe,-T., Scholkopf, B.P., Williamson R.C.. Advances in neural information processing systems (1999).
6. Zi, M.H., Hashemi M.R.: A DCT based approach for detecting novelty and concept drift in data streams (2010).
7. Martinez, D.: Neural tree density estimation for novelty detection (1998).
8. Markos, M.: Sameer. A neural network-based novelty detector for image sequence, Analysis, S. (2006)
9. Stephen, D.B.: The UCI KDD archive. http://kdd.ics.edu 1999
10. Markou, M., Singh S.: Novelty detection: a review-part 1: statistical approaches part 2: neural network based approaches (2003).