International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 5, May - 2014

Multiple Novel Class Detection in Concept Drifting Data Stream Using Decision Tree Classifier

Nithya B P
Computer science
kannur university
Kannur,India

Sree Rekha B
Computer science
kannur university
Kannur,India

Abstract—**Most existing data mining classifiers can not detect and classify the novel class instances in real-time data stream mining problems like weather conditions, economical changes, and intrusion detection etc, until the classification models are trained with the labeled instances of the novel class. In this paper, a new approach for detecting multiple novel class in data stream mining using decision tree classifier that can determine whether an unseen or new instance belongs to a novel class and detection of more than one novel class at a time are proposed. Arrival of a novel class in concept-drift occurs in data stream mining when new data introduce the new concept classes or remove the old ones.**

Keywords— ***Concept drift, data stream mining, decision tree, novel class, simultaneous multiple novel class***

I. INTRODUCTION

Data Stream Mining is the process of extracting knowledge from continuous data instances. The goal of many data stream mining applications is to predict the class value of new or unseen instances in the data stream. A data stream is an ordered sequence of instances. Some examples, includes attribute values and class values. The dynamic and evolving nature of data streams requires efficient and effective techniques that are significantly different from static data classification techniques. Most challenging and well-studied characteristics of data streams are its infinite length and concept-drift. Data stream is a fast and continuous phenomenon so it is assumed to have infinite length. Therefore, it is impractical to store and use all the historical data for training. Concept-drift occurs in the stream when the underlying concepts of the stream change over time. Other significant characteristics of data streams are concept-evolution and feature evolution. Concept-evolution occurs when new classes evolve in the data. For example, consider the case of a text data stream, such as that occurring in a social network such as Twitter. In this case, new classes may frequently emerge in the underlying stream of text messages.

Learning concept drift in data stream mining has focused on learning from real-time data, where the data distributions change over time such as weather conditions, economical changes, and intrusion detections etc. Most of the traditional data mining classifiers are trained on instances of the dataset with fixed number of classes, but in real-world data stream classification problems an unseen or new instance with new class may appear and the existing classification model misclassify the new instance. The concept drift means that the statistical properties of the target class in which the data mining classification models are trying to classify and change over time in unforeseen ways. The existing scheme proposes classification and novel class detection both in concept drifting and feature evolution data stream. This paper proposes Detection of multiple novel class in concept drifting data stream using decision tree classifier. Decision tree classifier is used for concept drifting data stream mining. Using decision tree continuously update the most recent data and it represent most recent data stream.

A. Learning with Concept-Drift:

In this section, we focus on the introduction of concept drifting in data stream classifications. Existing data mining algorithms for incremental learning assumed data streams come under stationary distribution, where data concepts remain unchanged. But the concept of any instance might change any time in real world applications. Concept-drift refers to a change in the class definitions over time, or underlying class (concept) of the data changing over time. The concept-drift may occur in three ways:

a) Class priors might change over time;

b) The distributions of one or several classes might change; and

c) The posterior distributions of the class memberships might change.

In data streams, Concept-drifting can be handled via:

a) window-based approaches,

b) weight-based approaches, and

c) ensemble classifiers.

A window-based approach builds a classification model by selecting the instances within a fixed or dynamic stream sliding window, and adjusts window sizes based on the classification accuracy rate . It combines all new and old instances together to generate a new training dataset, but only performs better for concept-drift in small datasets.

In a weight-based approach, each training instance is assigned a weight. Based on the weights, some outdated training instances will be opportunistically discarded from the training dataset.

The popular evolving technique for handling concept-drift in data streams is to use an ensemble classifier (combination of classifiers), which is shown inFigure1

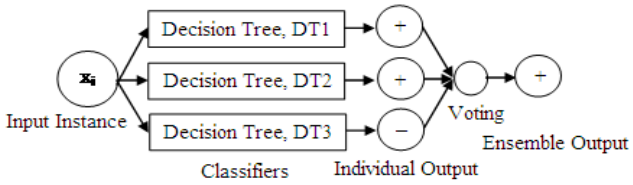

Fig 1: An example of an ensemble Classifier

The outputs of several classifiers are combined to determine a final classification, it is called fusion rules. Also the weights are assigned to the individual classifier's outputs at each point in time. The weight is a function of the historical performance in the past or estimated performance using 10-fold cross-validation. The best classifier among a number of classifiers (for either classification or prediction) can also be determined by performing 10-fold cross-validation. Data mining algorithms should be adaptive so that it can be continuously updated with the novel class instances as time passes. Most of the existing data mining algorithms are trained on datasets with a fixed number of class labels. Therefore, a new instance with a new class label will be misclassified by the traditional data mining classifiers. Figure 2 shows an example of novel class instances arriving in data streams. If we build a decision tree with a fixed number of class labels (see the left-hand side Fig 2), the decision rules are:

a) if(x > x1 and y < y2) or (x < x1 and y < y1) then class =plus, and

b) if(x > x1 and y > y2) or (x < x1 and y > y1) then class = minus.

This decision tree classification model correctly classifies the instances with a fixed number of class labels, i.e. those with 'plus' and 'minus' labels.It will misclassify any newly arrived class instances as shown on the right-hand side of Figure 2

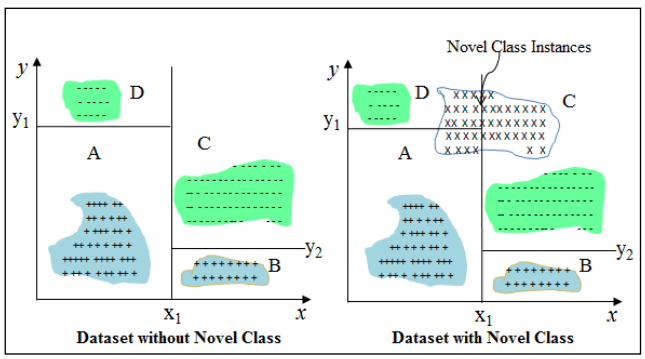

Fig 2:Instances with a fixed number of class labels (left) and instances of a novel class arriving in the data stream (right).

We organize this paper as follows. Section II discusses related works. Section III describes the background details.The proposed algorithm is introduced in SectionIV, followedby advantages and disadvantages in section V. SectionVI discusses future works .Finally,conclusions are drawn in SectionVII.

II. RELATED WORKS

In 2011, Masud et al. [1]proposed a datastream classification technique that integrates a novel class detection mechanism in to traditional classifiers and enabling automatic detection of novel classes before the true labels of the novel class instances arrive. Inorder to determine whether an instance belongs to a novel class then the classification model sometimes needs to wait for more test instances to discover similarities among those instances.

In 2007,J.Z.Kolter,and M.A.Maloof [13]presented an ensemble method for concept drift that dynamically creates and removes weighted experts in response to changes in performance using dynamic weighted majority(DWM).It trains online learners of the ensemble and add so removes experts based on the global performance of the ensemble.

Spinosa et al. [14] apply a cluster-based technique to detect novel classes in data streams. This approach builds a normal model of the data using clustering and is defined by the hypersphere encompassing all the clusters of normal data. This model updated continuously 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 multiclass data stream classification, since it corresponds to a "one-class" classifier.

Katakis et al. [5] propose a feature selection technique for data streams having dynamic feature space. This technique consists of an incremental feature ranking method and an incremental learning algorithm. In this approach, whenever a new document arrives, at first it is checked whether there is any new word in the document. If there is a new word, it is added to a vocabulary. After adding all the new words, the vocabulary is scanned and for all words in the vocabulary, statistics (frequency per class) are updated. Based on these updated statistics, new ranking of the words is computed and top N words are selected. The classifier (either kNN or Naive Bayes) is also updated with the top N words. When an unlabeled document is classified, only the selected top N words are considered for class prediction.

Masud et al. [1],[15] propose a classification and novel class detection technique that is a multiclass classifier and also a novel class detector. It uses an ensemble of models to classify the unlabeled data. A decision boundary is built in each of the models during training. If any test instance falls outside the decision boundary, it is considered as an outlier. If enough outliers are found that have sufficient cohesion among themselves, and separation from the existing data, then a novel class is declared. But this approach does not consider feature-evolution.

III. BACKGROUND

The approach described in this research paper is based on the previous work on classification and novel class detection in concept drifting data stream.This paper proposes detection of multiple novel class in concept drifting data stream using decision tree classifier.Using decision tree continuously update the most recent data. These data point obtained from these classifier are plotted in graph to detect multiple novel class. If there are found more than one novel class means multiple novel class is detected.

B. Decision TreeClassifier:

To make a decision using a DT, start at the root node and follow the tree down the branches until a leaf node representing the class is reached. In this approach, build a decision tree from training data points and calculate the percentage of number of data points for each leaf node in the tree with respect to data points in training dataset. And also cluster the data points based on the similarity of attribute values for each leaf node in the tree. When classifing the data streams in real-time, if number of data points classify by a leaf node increases than the percentage calculated before it means a novel class arrived. Then check in which cluster the new data point belongs based on the similarity of attribute values, if this new data point does not belongs to any cluster, it confirms a novel class arrived. Then add the new data point into training dataset and rebuild or updated the decision tree model. The decision tree classifier continuously updated so that it represents the most recent concept in the data stream.

The ID3 (Iterative Dichotomiser) technique builds decision tree .The basic strategy used by ID3 is to choose splitting attributes from a dataset with the highest information gain and the amount of information associated with an attribute value is related to the probability of occurrence. The objective of decision tree classification is to iteratively partition the given data set in to subsets where all elements in each final subset belong to the same class.

Algorithm 1 Novel Class Detection and Classification:

1: Find the best splitting attribute in training data set with highest information gain value.
2: Create a node and label with splitting attribute.[First node is the root node, T of tree]
3: For each branch of the node, partition the given data points and grow sub training datasets Di by applying splitting predicate to training dataset D.
4: For each sub training datasets Di, if data points in Di are all of the same class, C then a leaf node labeled with C.Else continue steps1 to 4 until each final subset belong to the same class or leaf node created.
5: After the construction of decision tree is completed then partition the training data points in to clusters based on the similarity of data points of leaf nodes of the tree.
6: Count the number of data points in each final sub dataset of each leaf node ,and calculate the percentage of data points for each leaf node in the tree with respect to total data points in training dataset.
7: For classifying data points in real-time, if number of data points classify by a leaf node increases than the percentage calculated before, confirms a novel class is arrived.
8: Check in which cluster the new data point belongs based on the similarity of attribute values, if new data point does not belongs to any cluster, it confirms a novel class arrived. Then add new data point in to train and rebuild the decision tree model.

C. Simultaneous Multiple Novel Class Detection:

The algorithm takes as input the novel instances (N list) that are identified using the novel class detection technique. There are two phases of the algorithm: separation phase and merging phase. In the separation phase, we separate the novel class instances into multiple types of instances, and in the merging phase, we try to merge different types of instances.

Input:N_list: Novel class instance's List
Output: N_type: Predicted novel instance's class label
//separation phase
1: empty//initialize graph
2: K-means(N_list , kv)//clustering
3:for do
4: Nearest-neighbor( )
5: Compute-SC( ) //silhouette coefficient
6:V←VU{h}//add these nodes
7: V←VU{h.nn}
8:if h.sc < thsc then//relatively closer to the nearest neighbor
9: E←E U {h,h.nn} //add this directed edge
10: end if
11: end for
12: count← Con-Components(G)//find connected components
// Merge phase
13: for each pair of components(g1 ,g2) ϵ G do
14:μ1←mean_dist(g1), μ1←mean_dist(g2)
15: if ((μ1+ μ2)/(2*centroid_dist(g1,g2)))>1 then g1←Merge(g1,g2)
16: end for //Now assign the class labels
17:N_type←empty
18:for xϵN_list do
19:h←PseudopointOf(x) //find the corresponding pseudopoint
20:N_type← N_type U {(x,h.componentno)}
21: end for

IV. PROPOSED MODEL

The novel classes obtained from decision tree are plotted on graph. The main idea in detecting multiple novel classes is to construct a graph, and then identify the connected components in the graph. The number of connected components determines the number of novel classes. The basic assumption in determining the multiple novel classes follows. If more than one component is detected means multiple novel class is arrived. For example, if there are two novel classes, then the separation among the different novel class instances

should be higher than the cohesion among the same class instance.

The basic assumption in determining the multiple novel classes follows the property: A data point should be closer to the data points of its own class (cohesion) and farther apart from the data points of other classes (separation),if there is a novel class in the stream, instances.eg: if there are two novel classes, then the separation among the different novel class instances should be higher than the cohesion among the same-class instances.

1: Find the best splitting attribute in training data set with highest information gain value.
2: Create a node and label with splitting attribute.[First node is the root node, T of tree]
3: For each branch of the node, partition the given data points and grow sub training datasets Di by applying splitting predicate to training dataset D.
4: For each sub training datasets Di, if data points in Di are all of the same class, C then a leaf node labeled with C.Else continue steps1 to 4 until each final subset belong to the same class or leaf node created.
5: After the construction of decision tree is completed then partition the training data points in to clusters based on the similarity of data points of leaf nodes of the tree.
6: Count the number of data points in each final sub dataset of each leaf node ,and calculate the percentage of data points for each leaf node in the tree with respect to total data points in training dataset.
7: For classifying data points in real-time, if number of data points classify by a leaf node increases than the percentage calculated before, confirms a novel class is arrived.
8.For multiple novel class repeate step1 to 7.
9. G = (V, E) ←empty //initialize graph
10:Apply K-means clustering to create Kv clusters, where Kv= K⎪N _List⎪/S .where S is the chunk size and k is the number of pseudopoints.
11. Each cluster is saved as a pseudopoint *h*, and those pseudopoints *h* are stored in the *NP _list*(novel pseudopoints list)
12: graph G is created
13: For each pseudopoint
14: $h.nn \leftarrow$Nearest-neighbor($NP\_list - \{h\}$) //find the nearest pseudopoint of h
15:$h.sc \leftarrow$ Compute-SC($h, h.nn$) //silhouette coefficient
16:$V \leftarrow VU\{h\}$//add these nodes
17: $V \leftarrow VU\{h.nn\}$
18:if $h.sc < th_{sc}$ then//relatively closer to the nearest neighbor
19: $E \leftarrow E\ U\{h,h.nn\}$ //add this directed edge
20:Find the connected component(g_1,g_2) in the graph.
21: For each pair of components (g_1,g_2)ϵG do
22:$\mu 1 \leftarrow mean\_dist(g_1)$, $\mu 1 \leftarrow mean\_dist(g_2)$
23:if $((\mu_1 + \mu_2)/(2*centroid\_dist(g_1,g_2))) > 1$ then merge(g_1,g_2)
24:Find the corresponding pseudopoint of a novel instance.
25**:** Assign the corresponding component number of that pseudopoint as the class label of the instance.
26: Then add new data point in to training datapoints and rebuild the decision tree model.

Apply K-means clustering to create Kv clusters, where Kv= K⎪N _List⎪/S .Recall that S is the chunk size. Next, the summary of each cluster is saved as a pseudopoint, and those pseudopoints are stored in the *NP _list* (novel pseudopoints list). Then graph G is created. For each pseudopoint $h \epsilon NP\_list$, we find the nearest pseudopoint of *h* and compute the silhouette coefficient of *h* using the following formula ,$h.sc=(dist(h,h.nn)-h\mu)/max(dist(h,h.nn),h.\mu)$ where *dist(h,h.nn)*is the distance between the centroids of *h*, and *h.nn*, i.e, the nearest neighbor of *h*. Also, $h.\mu$ is the mean distance from the centroid of *h* to all instances belonging to *h*. Therefore, *h.sc* is a measure of how tight the cluster is, and *h.sc* ranges from _1 to +1. If *h.sc* is high (close to 1), it indicates *h* is a tight cluster and it is far from its nearest cluster; and if *h.sc* is low, then *h* is considered as not so tight, and it is close to its nearest cluster.

Add both *h* and *h.nn* to the vertex list *V* . Then we check whether *h.sc* is less than a certain threshold (th_{sc}), and add the directed edge *(h, h.nn*) to the edge list *E* if indeed *h.sc* is less than the threshold. Therefore, we are adding an edge only if *h.sc* is lower than the threshold, meaning, *h* is closer to its neighbor and less tight . Once we have the graph G, we can find the connected components, and mark each pseudopoint with the corresponding component number. For example, if there are two connected components, all pseudopoints belonging to the first component will be tagged as "1" and all pseudopoints belonging to the second component will be tagged as "2."

In the merging phase, we examine different components of the graph to see whether they can be merged. For each pair of components (g_1, g_2), we first find the mean distance of each pseudopoint from the global centroid of the corresponding component , and then merge them if the sum of the mean distances is greater than twice the global centroid distance between g_1 and g_2. In other words, two components are merged if the mean intracomponent distance is higher than the intercomponent distance. Finally, assign class labels to each novel class instance. To do this, first we find the corresponding pseudopoint of a novel instance, and assign the corresponding component number of that pseudopoint as the class label of the instance .After obtaining the multiple novel class the new data points can be added to training data points and the decision tree can be rebuild .

V ADVANTAGES/DISADVANTAGES

A. *ADVANTAGES:*

- Classification and novel class detection is easy to implement.
- The cost of novel class detection is reduced.
- The performance of decision tree classifier is Improved**.**
- It has improooved classification accuracy.

B. DISADVANTAGES:

- Perfomance can be improoved by boosting.But boosting does not help when the training data contains a lot of noise.
- Addressing the datastream classification problem under dynamic attribute sets is is not possible.

VI FUTURE WORKS

This paper proposed multiple novel class detection in concept drifting data stream.But this paper does not propose multiple novel class detection under dynamic attribute sets.So the future work mainly focus on addressing the datastream classification problem under dynamic attribute sets and multilabel classification problem in data streams.

VII CONCLUSION

Data Stream Mining is the process of extracting knowledge from continuous data instances. Concept-drift occurs in the stream when the underlying concepts of the stream change over time. This paper proposed multiple novel class detection in concept drifting data stream.Classification and novel class detection using decision tree classifier is easy to implement. The performance of decision tree classifier and classification accuracy is also is Improved. Perfomance can be improoved by boosting.But boosting does not help when the training data contains a lot of noise The cost of novel class detection is reduced.Using decision tree classifier addressing the datastream classification problem under dynamic attribute sets is is not possible. We shall consider the drift detection issue in the future to make our approach more dynamic and robust. The future work focus on addressing the datastream classification problem under dynamic attribute sets and multilabel classification problem in data streams.

REFERENCES

[1]Mohammad M. Masud, JingGao,Latifur Khan"Classification and Novel Class Detection in Concept- Drifting Data Streams under Time Constraints ," IEEE Transactions on knowledge and Data Engineering, Vol.23, No.6, June2011,pp.859-874.

[2] C.C. Aggarwal, J. Han, J. Wang, and P.S. Yu, "A Framework for On-Demand Classification of Evolving Data Streams," IEEE Trans.Knowledge and Data Eng., vol. 18, no. 5, pp. 577-589, May 2006.

[3] Mohammad M. Masud, Qing Chen, Latifur Khan, " Classification and Adaptive Novel Class Detection of Feature-Evolving Data Streams " IEEE Transactions on knowledge and Data Engineering, VOL. 25, NO. 7, JULY 2013

[4] Dewan Md. Farid,_ and Chowdhury Mofizur Rahman "Novel Class Detection in Concept-Drifting Data Stream Mining Employing Decision Tree" 2012 7th International Conference on Electrical and Computer Engineering 20-22 December, 2012, Dhaka, Bangladesh

[5] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic Feature Space and Incremental Feature Selection for the Classification of Textual Data Streams," Proc. Int'l Workshop Knowledge Discovery from Data Streams (ECML/PKDD), pp. 102-116, 2006.

[6] WeiFan,"Mining Concept Drifting DataStreams Using Ensemble Classifiers" Proc.10thACMSIGKDDInt 'lCon. Knowledge discovery and datamining, 2004,pp.128-137.

[7] B. Wenerstrom and C. Giraud-Carrier, "Temporal Data Mining in Dynamic Feature Spaces," Proc. Sixth Int'l Conf. Data Mining (ICDM), pp. 1141-1145, 2006.

[8] H. Wang, W. Fan, P.S. Yu, and J. Han, "Mining Concept-Drifting Data Streams Using Ensemble Classifiers," Proc. ACM SIGKDD Ninth Int'l Conf. Knowledge Discovery and Data Mining, pp. 226-235, 2003.

[9] W.Y.Loh, and X.Shih, "Split selection methods for classification tree," Statistica Sininca, Vol.7,1997,pp.815-840.

[10] John Shafer, Rakeeh Agrawal, and Manish Mehta, "SPRINT: A scalable parallel classifier for data mining," Morgan Kaufmann, 1996,pp.544-555

[11] G.Widmer, and M.Kubat ,"Learning in the Presence of Concept Drift and Hidden Contexts," Machine Learning, Vol.23,No.1,April1996, pp. 69-101.

[12] M.M. Masud, Q. Chen, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Classification and Novel Class Detection of Data Streams in a Dynamic Feature Space," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 337-352, 2010.

[13] J.Zico Kolter, and MarcusA.Maloof, "Dynamic Weighted Majority: An ensemble method for drifting concepts," Journal of Machine Learning Research Vol.8,2007,pp.2755-2790.

[14]E.J. Spinosa, A.P. de Leon F. de Carvalho, and J. Gama, "Cluster-Based Novel Concept Detection in Data Streams Applied to Intrusion Detection in Computer Networks," Proc. ACM Symp.Applied Computing (SAC), pp. 976-980, 2008.

[15] M.M. Masud, J. Gao, L. Khan, J. Han, and B.M. Thuraisingham, "Integrating Novel Class Detection with Classification for Concept- Drifting Data Streams," Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD), pp. 79-94,2009.

[16] Dewan Md. Farida, Li Zhanga, Alamgir Hossaina, Chowdhury Mofizur Rahmanb, Rebecca Strachana, Graham Sextona, and Keshav Dahalc "An Adaptive Ensemble Classifier for Mining Concept-Drifting Data Streams".