International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 4 Issue: 5
ISSN: 2321-8169
67 – 70

Handling Gradual Concept Drift in Stream Data

Ms. Priyanka B. Dongre (Author)
Department of Information Technology
Priyadarshini College of Engineering, Nagpur.
Nagpur, India
e-mail: priyankadongre90@gmail.com

Abstract: Data streams are sequence of data examples that continuously arrive at time-varying and possibly unbound streams. These data streams are potentially huge in size and thus it is impossible to process many data mining techniques (e.g., sensor readings, call records, web page visits). Tachiniques for classification fail to successfully process data streams because of two factors: their overwhelming volume and their distinctive feature known as concept drift. Concept drift is defined as changes in the learned structure that occur over time. The occurrence of concept drift leads to a drastic drop in classification accuracy. The recognition of concept drift in data streams has led to sliding-window approaches, instance selection methods, drift detection, ensemble classifiers. This paper describes the various types of concept drifts that affect the data examples and discusses various approaches inorder to handle concept drift scenarios. The aim of this paper is to review and compare single classifier and ensemble approaches to data stream mining respectively and propose a methodology towards its contribution successfully.

Keywords: *Data Stream; concept drift; single classifier; ensemble classifiers; dynamic environment.*

*****

I. Introduction

Most of the previous and current research in data stream mining is carried out in static environments, wherein a complete dataset is presented to the learning algorithm. These data were usually electronically stored and, if needed, was easily accessed by the learning algorithms several times. However, the target concepts which should be learned were always fixed. Over these years, many solutions to the static classification have been developed and several quite accurate classifiers are now available in large scale. However, in some of the newest and latest applications, learning algorithms work in dynamic environments, where a huge amount of data are continuously generated. Traffic management Sensor networks, monitoring, web log analysis or telecommunication are examples of such applications. In these dynamic environments, incoming data form a data stream are characterized by huge number of instances and they arrive rapidly, which often requires real-time and quick response.

Concept drift occurs when the concept about which data are being collected shifts from time to time after a minimum stability period. Such changes are reflected in incoming instances and decreases the accuracy of classifiers learned from past training tuples(input data examples). Examples of real life concept drifts include monitoring systems, financial fraud detection spam categorization weather predictions and customer preferences.

Changes of target concepts are categorized into sudden, incremental, gradual, recurring, blip or noise drifts. Figure 1. Shows the six basic types of drifts. The first plot (Sudden) shows abrupt changes that instantly and irreversibly change the data examples of respective class. The next two plots (Incremental and Gradual) illustrate changes that take place slowly over time. Incremental drift occurs when data example slowly change their values over time, and gradual drift occurs when the change in data example involves the class distribution of various data.

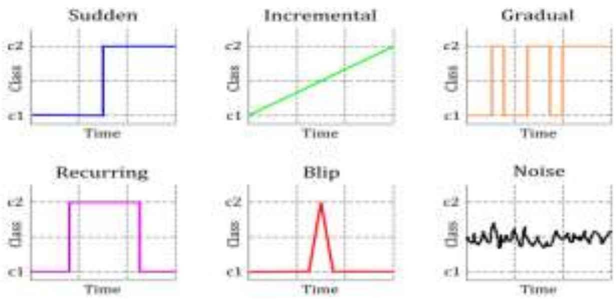

Fig. 1.Types of Concept Drifts

Some researchers do not differentiate these two types of drift and use the terms incremental gradual and as synonyms. The left-bottom plot (Recurring) represents changes in data example that are only temporary and are reverted after some time. The fifth plot (Blip) represents a “rare event”, which can be considered as an outlier in a static distribution. The last plot in Figure 1. represents random changes in data examples, which should be filtered out efficiently. A good classifier must learn incrementally and adapt to such changes.

In recent years, several approaches that try to handle various concept drift scenarios have been proposed effictively, including mainly sliding window approaches [1],adaptive sliding window techniques [2], new online algorithms, special detection techniques,Drift detection methods [4], Hoeffding trees [6], Ensemble drift detection methods and adaptive ensembles [8]. Ensembles are popular approaches for improving classification accuracy in static learning environment, however, they must be generalized for changing environments. Such a generalization can either modify the structure of the ensemble (weaker components are replaced by base classifiers trained on the most recent data examples), updating the aggregation technique (e.g., updating weights in the voting formula).

ISSN: 2321-8169
67 – 70

II. RELATED WORK

Drift detection techniques include the windowing technique that provide a way that limit the amount of data examples introduced to the learner, thus eliminating those data examples that come from an old concept.The basic windowing algorithm is straightforward in nature. Each data example updates the window and later the classifier is updated by that window itself. In the way it models the forgetting process. In the simplest approach sliding windows are of fixed size and include only the most recent examples from the data stream under consideration. With each new data point the oldest data example that does not fit in the window is thrown away.Another approach proposed by ZliobaiteFISH [2] is a family of algorithms which is known as FISH(FISH1,FISH2,FISH3), that uses time and space similarities between data examples as a way of dynamically creating a window. To explain the approach, let us consider an illustrative example, which we present in Figure 2. A binary classification problem is represented by black and white dots as shown below.

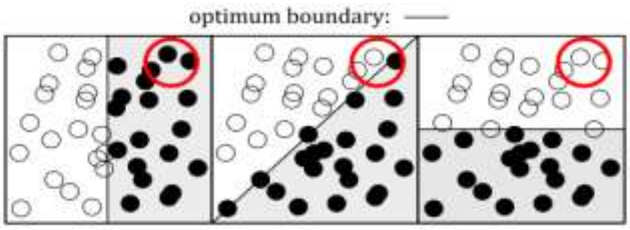

Fig. 2. Rotating hyperplane example (left) initial source S1, (center) source S2 after 45° rotation, (right) source S3 after 90° rotation. Black and white dots represent the two classes.

The data generating sources change along with time, gradually rotating the optimal classification hyperplane. For a given fixed space area, shown with a big round circle, the correct class changes as the optimal boundary rotates. The examples shows that similarity in an evolving environment depends on both time and space respectively.

Bifet proposed an adapting sliding window algorithm called ADWIN[3] suitable for data streams with sudden drift scenarios. The algorithm keeps a sliding window W with the most recently read data examples. The main idea of ADWIN is as follows: whenever two “large enough” subwindows of W exhibit “distinct enough” averages, one can conclude that the corresponding expected values are not similar, and thus the older portion of the window is dropped. After windowing techniques, another group of algorithms allowing to adapt almost any learner to evolving data streams are drift detectors. Their task is to detect concept drift and alarm the base learner that its model should be rebuilt or updated. Drift detector methods include the Drift Detection Method[4] that implies, in each iteration an online classifier predicts the decision class of an example. That prediction can be either true or false, thus for a set of example the error is a random variable. That is why the number of classification errors with a Binomial distribution calculated as given by Equation

$$s_i = \sqrt{p_i(1-p_i)/i} \qquad (1)$$

These values are used to calculate and a warning level condition presented in Equation 2. and an alarm level condition presented in Equation 3. Each time a warning level is reached, the data examples are remembered in a separate window.

$$pi + si \geq p_{min} + \alpha \cdot s_{min} \qquad (2)$$

$$pi + si \geq p_{min} + \beta \cdot s_{min} \qquad (3)$$

The values α and β in the above conditions decide about the confidence levels at which the warning and alarm signals are triggered only then. The authors proposed α = 2 and β = 3, giving approximately 95% confidence of warning and 99% confidence of concept drift. Baena-Garcia et al. proposed a modification of Drift Detection Method called EDDM[5]. The authors use the same warning-alarm mechanism that was proposed by Gama, but instead of using the classifier’s error rate, the distance error rate was proposed.

III. DISCUSSION

Above discussed classification technique fail to classify data examples inorder to detect drift ,in windowing technique when fixed size window is used, the user is caught in a tradeoff. In case of small window size the classifier will react quickly to changes, but may loose on accuracy in periods of stability and on the other hand, choosing a large size window will result in increasing accuracy in periods of stability, but will fail to adapt to rapidly changing concept drifts. That is why dynamic ways of modelling the forgetting process have been proposed which is known as Weighted Windows. A simple way of making the forgetting process more dynamic is providing the window with a decay function that assigns a weight to each data example. Older examples receive smaller weights and are treated as less important by the base classifier algorithm. Cohen and Strauss analyzed the use of different decay functions for calculating data stream aggregates.

FISH3 is an algorithm that always need to iterate through many window sizes and time/space proportions each time performing leave-one-out cross validation. This is a costly process and may be unfeasible for rapid data streams scenarios.

The ideal part of ADWIN lies in the definition of cut and the test it is used for. The authors stated, the different statistical tests can be used for this purpose, but propose one specific implementation only.The verification of all subwindows is very costly process due to the number of possible split points. ADWIN works only for 1-dimensional data only, e.g., the running error. For n-dimensional raw data, a separate window must be maintained for each dimension which results in handling more than one window. This modified model, although costly, reflects the fact that the importance of each feature may change at different pace.

DDM works best on data streams with sudden concept drift as gradually changing concept drifts can pass without triggering the alarm level. When no changes are detected, DDM works like a lossless learner constantly enlarging the window size that can lead to the memory limit being exceeded.

EDDM works better than DDM for slow gradual drift, but is more sensitive to noise issues. Another drawback of this EDDM is that it considers the threshold values and searches for concept drift when a minimum of 30 errors have occurred. This is necessary to approximate the Binomial distribution by a Normal distribution, but can take a large amount of examples to happen over.

IV. PROPOSED PLAN OF WORK

From above discussion it has been observed that there is a need of quick and efficient classifier to detect concept drift and classify data examples accurately. This section gives a design flow and description of preliminary model for classification and drift detection.

The proposed model aims to remove the problems related to inefficiency in accurately classifying the data examples in the presence on concept drift as the base classifier was not able to learn the old concept well and thus a need of completely new classifer is needed to train on new concept in order to make negative instances positive and detect drifted data along with classification based on completely new concept. The model aims not only to accurately classify the data but also detect drifted data correctly. The Figure3. show the work flow of the proposed system.

The steps that would be followed to obtain accurately classified data along with drifted data are:
1)The system evaluates the dataset properties.
2)The redundant and irrelevant attributes are eliminated.
3)Minority class is discovered and iteratively the boundaries are refined to obtain accurately classified instances.

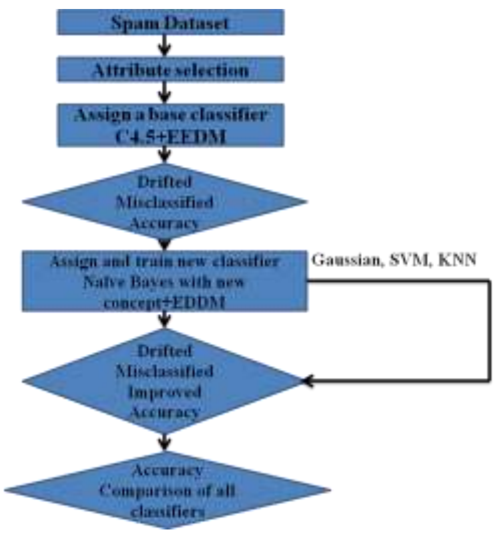

Figure 3. Work Flow of proposed plan

Firstly the dataset properties are evaluated. The second step comprises of eliminating the attributes by checking their relevance value. Thus we can say that irrelevant and duplicate columns are eliminated. This is done to improve the accuracy for classifying the instances. After feature selection a base classifier will be assigned to classify the data examples and along with this base classifier EDDM will be implemented so inorder to detect drifted data. The last step describes the final output wherein the misclassified instances and the drifted data obtained by the previous step vl be taken as next input and they will be trained on new concept by completely new classifier. After performing the last step the final output will have higher accuracy and will learn the old concept as well as new concept for better classification and detection of drift respectively.

<u>Algorithm:</u>
Input: spam dataset instances, Base learner = C4.5, Drift Detector =EDDM
1: Calculate weight of instances (Feature selection algorithm)
2: Assign a base classifier integrated with EEDM
3: predicted instances by C4.5 <- Original instances
4: if classification==false then
5: add a new window consisting of misclassified instances
6: level <- *computeEDDMLevel*(inst, classification)
7: if level==WARNING then start picking instances till the DRIFT level
8: Drifted data(rows/instances) are obtained
9: Assign this drifted rows as training set to naïve bayes
10: Naïve bayes trains on these drifted data and performs the testing on remaining instances.
11: if classification==false then
assign a new window for misclassified instances
12: level <- *computeEDDMLevel*(inst, classification)
13: if level==WARNING then start picking instances till the DRIFT level
14: Drifted data(rows/instances) are obtained
15: Compute accuracy(improved accuracy is obtained) as no of drifted rows decreased and many of the misclassified instances are correctly classified
16: Again a new classifier is assigned and the drifted rows of naïve bayes are given as a trainings et to gaussian classifier as the previous drifted rows by c4.5 is refreshed by drifted rows of naïve bayes.
17: Similar approach is followed for as described above
18: Thus five different classifier performance is evaluated i.e., C4.5,Naïve Bayes, Gaussian, SVM,KNN.
19: Hence performance is enhanced as the no of drifted rows go on decreasing and many of the misclassified instances are correctly classified.

V. CONCLUSION

Each method discussed above detects drift in a different manner. Bus classification accuracy is key factor that failed to be accomplished by above discussed methods. The objective is to detect the change in concept in the underlying data distribution and the its respective class label along with base classifier that will classify instances depending upon the parameters for classification along with drift detection. The proposed method involve the concept of using single classifiers to evaluate the performance of each respective classifier in order to detect drift along with classification and improve the accuracy of new classifiers by training the negatively classified instances and drifted data on completely new concept as these instances were considered to be the kind of instances that have not learnt the old concept well leading to misclassification. Thus the key factor of accuracy improvement can be achieved.

ACKNOWLEDGMENT

The author would like to thank "Dariusz Brzezinski" for making available all the latest technique applied to mining data stream with concept drift and experimental evaluation and comparison of different approaches towards drift detection and classification.

REFERENCES

[1] SACHDEV, M.S. ; SASKATCHEWAN UNIV., SASKATOON, SASK., CANADA ; SIGHU, T.S. ; GHOTRA, D.S., "IMPLEMENTATION OF AN ADAPTIVE DATA WINDOW TECHNIQUE IN A DISTANCE RELAY" 8TH IEEE INT. CONF. DATA MINING, DEC. 2008, PP. 929–934.

[2] L. L. MINKU, A. P. WHITE, AND X. YAO,"THE IMPACT OF DIVERSITY ON ONLINE ENSEMBLE LEARNING IN THE PRESENCE OF CONCEPT DRIFT," *IEEE TRANS. KNOWL. DATA ENG., VOL. 22, NO. 5, PP. 730–742, MAY 2010.*"

[3] Chu F, Wang Y Z, and Zaniolo C. An adaptive learning approach for noisy data streams[C]//Proc of 4th IEEE International Conference on Data Mining, Brighton: IEEE Computer Science, 2004: 351-354

[4] L. L. Minku and X. Yao , "DDD: A new ensemble approach for dealing with concept drift," *IEEE Trans. Knowl. Data Eng., vol. 24, no. 4,* pp. 619–633, Apr. 2012.

[5] W. N. Street and Y. Kim , "A streaming ensemble algorithm (SEA) for large-scale classification" , *IEEE Trans. Knowl. Data Eng., vol. 29, no. 3,* pp. 619–633, Apr. 2010.

[6] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[7] Dariusz Brzezinski and Jerzy Stefanowski,"Reacting to Different Types Of Concept Drifts- An Accuracy Updates Algorithm," *IEEE Trans. Neural networks and learning system.,2013.,vol. 23*.

[8] I. Zliobaite, "Combining time and space similarity for small size learning under concept drift," in *Proc. 18th ISMIS Int. Symp.*, 2009, pp. 412–421.

[9] Kolter, J Z and Maloof M A. Dynamic weighted majority: an ensemble method for drifting concepts[J]. Machine Learning Research, 2007, 8:2755-2790

[10] Zliobaite, I. and Kuncheva, L. (2009). Determining the training window for small sample size classi_cation with concept drift. In Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on, pages 447-452. IEEE.

[11] Nishida, K. and Yamauchi, K. (2009). Learning, detecting, understanding, and predicting concept changes.