Algorithm to handle Concept Drifting in Data Stream Mining

[1]Ms. Snehlata Dongre, [2] Dr. Latesh Malik

[1] Department of Computer Science and Engineering
G. H. Raisoni College of Engineering
Nagpur, India

[2] Department of Computer Science and Engineering
G. H. Raisoni College of Engineering
Nagpur, India

Abstract

Data Stream Mining is the evolving field of research. Mining continuous data streams brings unique opportunities but also new challenges. This paper will describe and evaluate the proposed classifier which uses ensemble classifier along with the boosting concept. Adaptive windowing is also used for handling the data stream. Empirical study will show that the proposed classifier takes less memory, less time, gives the good accuracy also handles the concept drift.

Keywords: *Data Stream Mining, Concept Drift, Classification, Ensemble Classifier*

1. Introduction

Data stream means huge volume of data which is continuously flowing. Knowingly or unknowingly everyone is connected with the data streams. Whenever swapping the credit card it generates the data stream. There are number of applications which generate the data streams like sensors, retailing, telecommunications, ATM and credit card transactions, popular websites log. In traditional data mining approach, whole data is stored first in memory and then process it and the data is static in the nature. But this is not the case in the data stream mining. Data streams are huge in volume and continuously flowing as well as dynamic in nature. If want to store the data streams it will exhaust the whole memory in the system and it will require more time to process. Also the dynamic nature of the data stream makes the classification more difficult. So to overcome the above problems there is requirement of efficient algorithms which take less memory, less time and also handle the dynamic nature of the data stream. The dynamic nature of data stream is also known as the concept drift. The underlying concept that maps the features to the class labels is changing in the data stream. In data stream the concept may drift gradually or suddenly. In the gradual concept drift the time step is taken gradually for old concept to be completely replaced with the new concept. Similarly in sudden concept drift the time step is taken suddenly for old concept is completely replaced with the new. Handling the concept drift is the issue in the field of data stream mining.

The paper is organized as follows, Related Work including various data stream classification methods are discussed in Section 2. Section 3 described the preliminary definitions for the understanding of the paper. Proposed method is described and results are discussed in the Section 4 with concluding conclusion in Section 5.

2. Related Work

There are number of algorithms used in the literature. Domingos et al. [2] devised a novel decision tree algorithm, VFDT, to overcome the long training times issue. The VFDT algorithm is based on a decision tree learning method combined with sub sampling of the entire data stream. The size of the sub sample is calculated using distribution free bounds called *Hoeffding bounds* under the assumption that the data is generated by a stationary distribution. For this reason, the method can process each example in constant time and memory being able to incorporate tens of thousands of examples per second using off-the-shelf hardware. The main drawback of the VFDT algorithm is its inability to cope with concept drifts. Domingos et al. [3] extended VFDT algorithm to CVFDT in an attempt to handle concept drift. The CVFDT algorithm mines high speed data streams under the approach of one pass mining. The one pass mining approach does not recognize the changes which have occurred in the model during the data arrival process. Although the CVFDT algorithm seems to be an effective method for incremental updating of the classification model induced from a dynamic data stream, the claim is that the accuracy of such an incremental model cannot be greater than the best sliding window model. Freund and Schapire et. al. [1] introduced a new “boosting” algorithm called AdaBoost which, theoretically, can be used to significantly reduce the error of any learning algorithm that consistently generates classifiers whose performance is a little better than random guessing. They also introduced the related notion of a “pseudo-loss” which is a method for forcing a learning algorithm of multi-label concepts to concentrate on the labels that are hardest to discriminate. It is basic boosting algorithm. Wang et. al. [4] proposed a weighted classifier ensemble to mine streaming data with concept drifts. Instead of continuously revising a single model, train an ensemble of classifiers from sequential data chunks in the stream. This technique shows that, in order to avoid overfitting and the problems of conflicting concepts, the expiration of old data must rely on data’s distribution instead of only on their arrival time. The ensemble approach offers this capability by giving each classifier a weight based on its expected prediction accuracy on the current test examples. Masud et. al. [5] proposed a multi-partition, multi-chunk ensemble classifier based data mining technique to classify concept-drifting data streams. Existing ensemble techniques in classifying concept-drift data streams follow a single-partition, single-chunk approach, in which a single data chunk is used to train one classifier. In this approach, method to train a collection of *v* classifiers from *r* consecutive data chunks using *v*-fold partitioning of the data, and build an ensemble of such classifiers.

This is a generalized multi-partition, multi-chunk ensemble technique that significantly reduces the expected classification error over the existing single-partition, single-chunk ensemble methods. The Streaming Ensemble Algorithm (SEA) [6] copes with concept drift with a bagging ensemble of C4.5 classifiers. SEA reads a fixed amount of data and uses it to create a new classifier. If this new classifier improves the performance of the ensemble, then it is added. However, if the ensemble contains the maximum number of classifiers, then the algorithm replaces a poorly performing classifier with the new classifier. Performance is measured over the most recent predictions and is based on the performance of both the ensemble and the new classifier. Unfortunately, there are problems with this approach. One is that members of the ensemble stop learning after being formed. This implies that a fixed period of time will be sufficient for learning all target concepts. In addition, if concepts drift during this fixed period of time, the learner may not be able to acquire the new target concepts. Finally, replacing the worst performing classifier in an unweighted ensemble may not yield the fastest convergence to new target concepts. It is general method based on the Weighted Majority algorithm for using any online learner for concept drift. Dynamic Weighted Majority (DWM) [7, 8] maintains an ensemble of base learners, predicts using a weighted majority vote of the experts. The algorithm begins by creating a set of experts and assigning a weight to each. When a new instance arrives, the algorithm passes it to expert and receives a prediction from each expert. The algorithm predicts based on a weighted majority vote of the expert predictions. If an expert incorrectly classifies the example, then the algorithm decreases its weight by a multiplicative constant. Oza et. al. [9, 10] proposed online bagging and boosting methods. Online bagging is a good approximation to batch bagging to the extent that their base model learning algorithms produce similar models when trained with similar distributions of training examples. Given a training dataset T of size N, standard batch bagging creates M base models. Each model is trained by calling the batch learning algorithm L_b on a bootstrap sample of size N created by drawing random samples with replacement from the original training set.

The online boosting algorithm [9, 10] is designed to correspond to the batch boosting algorithm, AdaBoost. AdaBoost generates a sequence of base models □h_1, $h_2,\ldots,h_M$ using weighted training sets (weighted by D_1, D_2, $\ldots,D_M$) such that the training examples misclassified by model h_{m-1} are given half the total weight when generating model h_m and the correctly classified examples are given the remaining half of the weight. When the base model learning algorithm cannot learn with weighted training sets, one can generate samples with replacement according to D_m. In AdaBoost, an example's weight is adjusted based on the performance of a base model on the entire training set while in online boosting; the weight adjustment is based on the base model's performance only on the examples seen earlier. OzaBagAdwin [11] is the online bagging method of Oza and Rusell with the addition of the ADWIN algorithm. When a change is detected, the worst classifier of the ensemble of classifiers is removed and a new classifier is added to the ensemble. UCVFDT [12] i.e. Uncertainty-handling and Concept-adapting Very Fast Decision Tree method is based on CVFDT technique. C. Liang has proposed this method. This is an extended version of DTU and CVFDT. This method is mainly used for uncertain data. So this is especially suitable for real life applications.

3. Preliminary Definitions

3.1 Ensemble Classifier

Ensemble Classifier use a combination of models to obtain better predictive performance than the single model. Each combines a series of m learned models or classifiers $h_1,\ldots\ldots,h_m$ with the aim of creating an improved composite model. Boosting can be used for the classification.

3.2 Boosting

Boosting[10] is a somewhat more complex process that generates a series of base models $h_1,\ldots\ldots,h_m$. Each base model h_m is learned from a weighted trained set whose weights are determined by the classification errors of the preceding model h_{m-1}. Specially, the examples misclassified by $h_{m-1}$are given more weight in the training set for $h_{m.}$, such that the weights of all misclassified examples constitute half the total weight of the training set

3.3 Adaptive Windowing

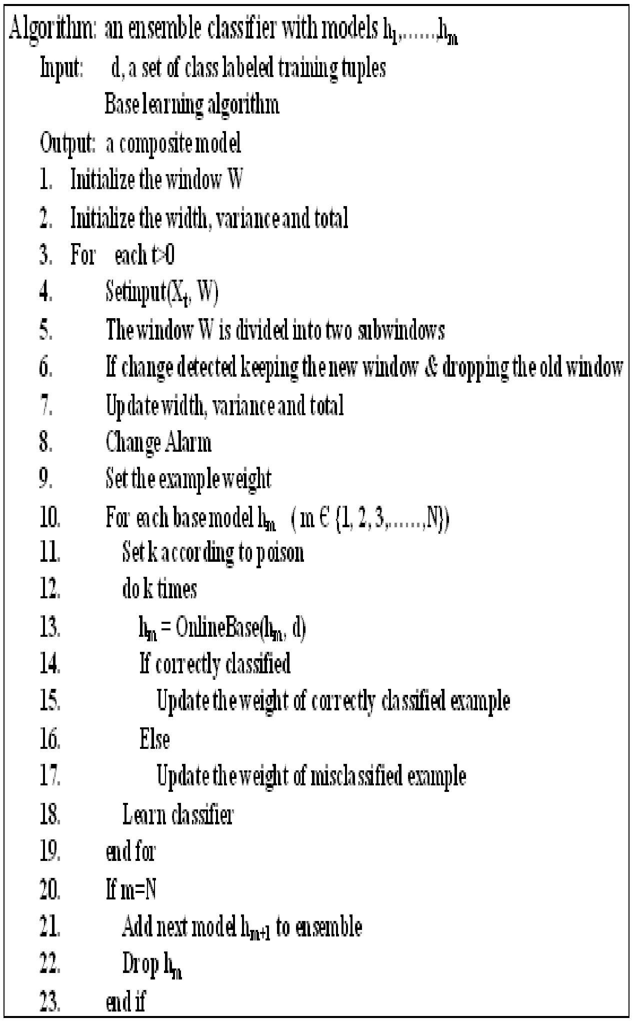
Adaptive Windowing ADWIN[11] is a change detector and estimator that solves in a well-specified way the problem of tracking the average of a stream of bits or real-valued numbers. ADWIN keeps a variable length window of recently seen items, with the property that the window has the maximal length statistically consistent with the hypothesis “there has been no change in the average value inside the window”. ADWIN automatically detects and adopts to the current rate of change.

4. Empirical Results

4.1 Proposed Method

This proposed ensemble classifier is using the concept of boosting and Adaptive windowing. The figure 1 shows the Algorithm for the proposed Method. There are the h_1,……,h_m models in the ensemble where m Є {1, 2, 3,….,N}. The d is the data used for training the model. Firstly the input data is given to the proposed method, ADWIN will divide the window in two sub windows and finding the change. If change is detected then it keeps the new window and dropping the old window and also generating the change alarm else proceed simply. Weight is assigned to ach example from the data used for training.

Every model of the ensemble try to classify the data. If it classify correctly then the weight of correctly classified example will decreased and if it misclassify the example then the weight of misclassify example will increased. In this way the model is trained. If the size of ensemble is equal to the maximum size of ensemble in that case we need to update the ensemble by adding a new model by replacing the weaker model in the ensemble.



```
Algorithm: an ensemble classifier with models $h_1$,......$h_m$
   Input:    d, a set of class labeled training tuples
             Base learning algorithm
   Output:  a composite model
   Initialize the window W
   Initialize the width, variance and total
   For    each t>0
       Setinput($X_t$, W)
       The window W is divided into two subwindows
       If change detected keeping the new window & dropping the old window
       Update width, variance and total
       Change Alarm
       Set the example weight
       For each base model $h_m$   ( m Є {1, 2, 3,......,N})
          Set k according to poison
          do k times
             $h_m$ = OnlineBase($h_m$, d)
             If correctly classified
                Update the weight of correctly classified example
             Else
                Update the weight of misclassified example
          Learn classifier
       end for
       If m=N
          Add next model $h_{m+1}$ to ensemble
          Drop $h_m$
       end if
```

Fig. 1 Proposed Method

4.2 Implementation and Analysis

The proposed method is implemented and tested for the synthetic data Hyperplane which is generated by the MOA framework. In which total no. of attribute is 10, the 1000000 instances has been taken for two class problem. The drift has been added to the data. We are varying the ensemble size and analyzing the performance of the proposed method. The performance can be evaluated in terms of accuracy, time and memory.

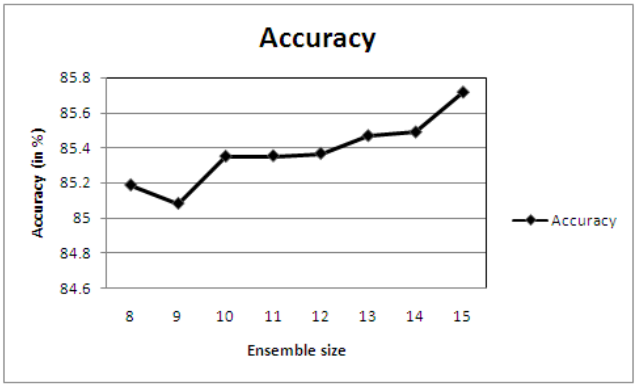


Fig. 2 Performance evaluation in terms of Accuracy

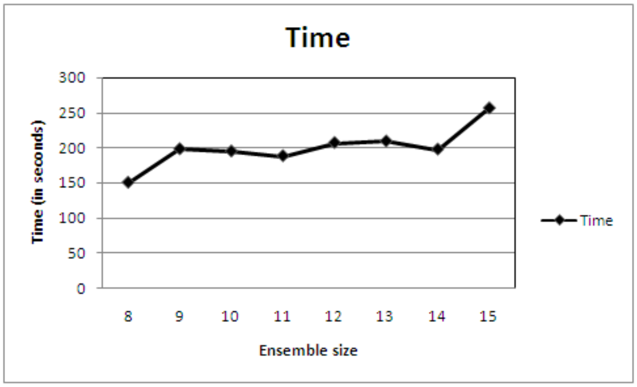


Fig. 3 Performance evaluation in terms of Time

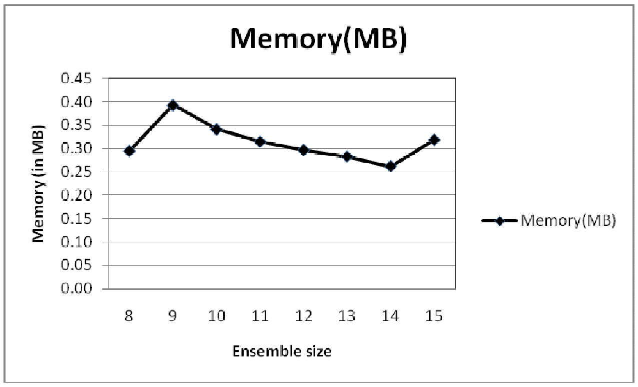


Fig. 3 Performance evaluation in terms of Memory

Figure 2 shows that when the ensemble size is increasing from 8 to 15 the accuracy of the proposed method is also increasing. In figure 3 performance is evaluated in terms of time. When the ensemble size is increased the time required to process data is also increased. Figure 4 shows that the requirement of memory while varying the ensemble size. In this way the performance of proposed method is analyzed when the ensemble size is 8, 9, 10, 11, 12, 13, 14 and 15.

We have compared the proposed method with the well known algorithms OzaBag, OzaBoost and OzaBagADWIN in terms of again accuracy time and memory. For that we have used 100000, 1000000 and 10000000 number of instances. The drift 0.0010 is added.

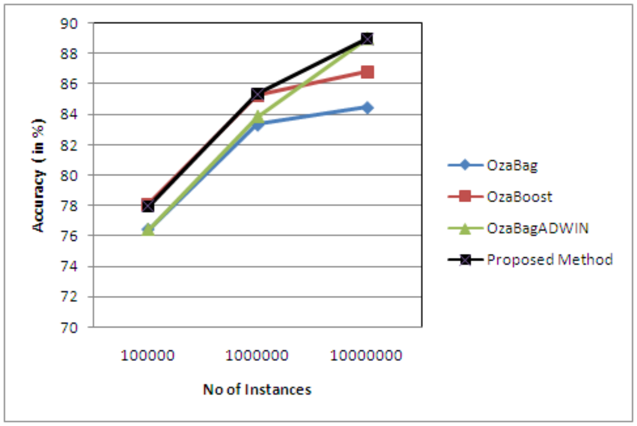


Fig. 5 Comparisons in terms of Accuracy

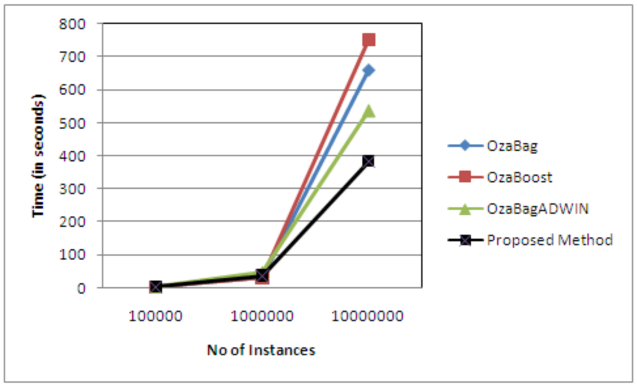


Fig. 6 Comparisons in terms of Time

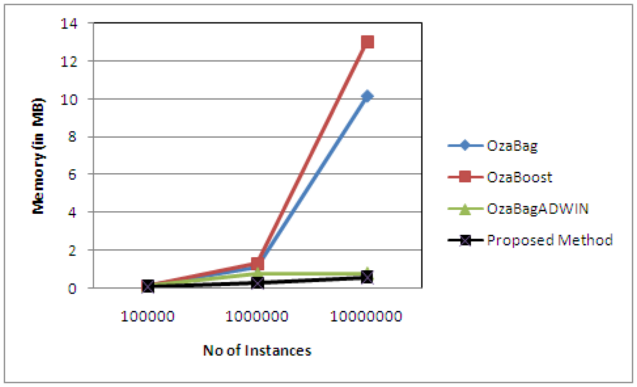


Fig. 7 Comparisons in terms of Memory

The figure 5 shows the comparison of the OzaBag, OzaBoost, OzaBagADWIN with the proposed method where the proposed method have the higher accuracy as compared to other algorithm. In figure 6 the proposed method takes the less time as compared to the other algorithms. Figure 7 shows the comparison of algorithms in terms of memory requirement again the proposed method takes the less memory.

5. Conclusion

We conducted extensive experiments on synthetic data stream. Our goal is to compare the proposed method, to evaluate the impact of the concept drifts on prediction accuracy, time and memory and to analyze the advantage of our approach over alternative methods such as Ozabag, , OzaBoost and OzaBagADWIN. The paper has described the proposed method in detail. The Proposed Method has been implemented, analyzed and compared with the other algorithms. The results shows that the proposed method is taking less time, less memory and giving the higher accuracy as compared to the other algorithms. It shows that the proposed method can also handle the concept drift properly.

References

[1] Y. Freund, R. Schapire, “Experiments with a new boosting algorithm”, In ICML, pp 148-156, 1996.

[2] P. Domingos, G. Hulten, “Mining high-speed data streams”, In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining(KDD’00), pp 71-80, 2000

[3] G. Hulten, L. Spencer, P. Domingos, “Mining time-changing data streams”, In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'01), San Francisco, CA, pp 97-106, 2001.

[4] H. Wang, W. Fan, V. Yu and J. Han, “Mining concept-drifting data streams using ensemble classifiers”, In ACM SIGKDD, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 226 – 235, 2003.

[5] M. Masud, J. Gao, L. Khan, J. Han and B. Thuraisingham, “A multipartition multi chunk ensemble technique to classify concept drifting data streams”, In Proceedings of the 13th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'09), Springer-Verlag Berlin, Heidelberg, pp 363-375, 2009.

[6] W. Street, Y. Kim, “A streaming ensemble algorithm (sea) for large-scale classification”, In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, pp 377-382, 2001.

[7] J. Kolter, M. Maloof, “Dynamic weighted majority: a new ensemble method for tracking concept drift”, Journal of Machine Learning Research, pp 2755-2790, 2007.

[8] J. Z. Kolter, M. A. Maloof, “Using additive expert ensembles to cope with concept drift”, In Proceedings of the 22nd international conference on Machine learning (ICML), Bonn, Germany, pp 449-456, 2005.

[9] N. Oza, S. Russell, “Experimental comparisons of online and batch versions of bagging and boosting”, In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’01), pp 359-364, 2001.

[10] N. Oza, S. Russell, “Online bagging and boosting”, In Artificial Intelligence and Statistics, Morgan Kaufmann, pp 105-112, 2001.

[11] A. Bieft, G. Holmes, B. Pfahringr, R. Kirkby, R. Gavalda “New ensemble methods for evolving data streams”, In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD’09), Paris, France, pp 139-148, 2009.

[12] C. Liang, Y. Zhang, Q. Song, “Decision Tree for Dynamic and Uncertain Data Streams”, In Proceedings of 2nd Asian Conference on Machine Learning (ACML2010), Tokyo, Japan, pp 209-224, 2010.

S. S. Dongre She received B. E. degree in Computer Science and Engineering from Pt. Ravishankar Shukla University, Raipur, India in 2007 and M. Tech. degree in Computer Engineering from University of Pune, Pune, India in 2010. She is currently working as Assistant Professor the Department of Computer Science and Engineering at G. H. Raisoni College of Engineering, Nagpur, India. Number of publications is in reputed International conferences like IEEE and Journals. Her research is on Data Stream Mining, Machine Learning, Decision Support System, ANN and Embedded System. Her book has published on titled Data Streams Mining: Classification and Application, LAP Publication House, Germany, 2010. Ms. Snehlata S. Dongre is a member of IACSIT, IEEE and ISTE Professional Societies.

Dr. Latesh Malik She received B. E. degree in Computer Science and Engineering from Rajasthan University, India in 1996 and M. Tech. degree in Computer Science & Engineering from Banasthali Vidyapith, India in 2001. She is currently working as Professor and Head of the Department in Department of Computer Science and Engineering at G. H. Raisoni College of Engineering, Nagpur, India. Number of publications is in reputed International conferences like IEEE and Journals. Her research is on Document Processing, Soft Computing and Image Processing. She received Best teacher award and she is gold medalist In M.Tech. and B.E. She also received the grant from AICTE of Rs. 7.5 lacs under RPS. Dr. Latesh Malik is a member of CSI, IEEE and ISTE Professional Societies.