



DATA STREAM CLASSIFICATION THROUGH ENSEMBLE CLASSIFIER USING Hoeffding Option Trees

Yamini Kadwe¹, Vaishali Suryawanshi²

¹Department of IT, M.I.T. College Of Engineering, Pune, India

²Department of IT, M.I.T. College Of Engineering, Pune, India

Abstract: Data stream classification is a challenging task. For real-time, data concepts of instances keep varying with time, such as weather prediction or intrusion detection etc. Classification is a supervised technique to mine useful patterns from these real data. There is a need to mine knowledge from these large data streams with techniques those provide accurate results and memory efficient. Using a set of classifiers i.e. ensemble classifier than a single one, proves to be more efficient. In this paper, Revised Accuracy Updated Ensemble(RAUE) classifier algorithm is proposed. In this algorithm, the data streams are processed in blocks. The ensemble used in RAUE uses Hoeffding Option Trees(HOT) as basic classifier set. The experimental evaluation and comparison of the proposed RAUE with existing ensemble algorithms is carried out. Out of all the compared algorithms, RAUE shows increase in accuracy and also uses less memory for identifying class labels for instances.

Keywords: concept drift, ensemble, experts, chunk, component classifiers

I. Introduction

The data streams are large volumes of high speed ordered, continuous data evolving from real-domain applications. Data Stream Mining is the process of extracting knowledge structures from continuous, rapid data records. The data stream can be read only once or a small number of times using limited computing and storage capabilities. The figure 1 below shows, [16]general model for data stream mining. The stream mining engine processes data streams and creates synopsis in memory; using time windows and various computational approaches. The stream mining engine gives approximate results for user request or after a certain lapse. Stream validation methods are applied to evaluate the performance of data stream algorithms. Classification is a supervised technique of [14] mining information from continuously generated data streams.

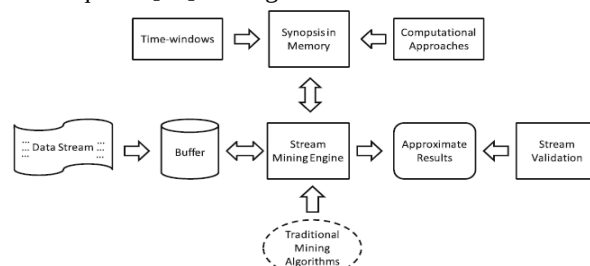


Fig 1: General model for data stream mining[16]

In this, we provide a set of training examples in the form (i,j) , where i is the vector of n attributes with j being the discrete class label and aim in producing a model of the form $j=f(a)$. The function $f(a)$ should accurately predict the class j for the future examples.

The real-domain applications[6] of data stream mining considering drifts are monitoring and control, personal assistance, decision support and artificial intelligence applications. For instance, in an intrusion detection systems incoming network traffic is filtered in search of suspicious behavior. The source of concept drift in this application is mainly connected with the attacker. Adversary actions taken by the intruder evolve with time, to surpass the also evolving security systems.

Section I gives overview on data stream mining. The work in data stream classification algorithms is summarized in section II. The proposed algorithm and the objectives are presented in section III. Section IV explains in detail about the experimental setup, datasets used and results recorded. Section V includes the empirical analysis based on comparative study between the proposed and existing approaches, for all experiments carried out.

II. Related work

The concepts in evolving data streams do not remain same. The change in concepts of the data is known as concept drift. [15] A drift can be *sudden* or *abrupt*, when concept switching is from one to another. The concept change can be *incremental*, consisting of many intermediate concepts in between. Drift may be *gradual*; change is not abrupt, but goes back to previous pattern for some time. Concept drift handling algorithms should not mix the true drift with an *outlier(blip)* or noise, which refers to an anomaly. A recurring drifts is when new concepts that were not seen before, or previously seen concepts may *reoccur* after some time.

Several classification algorithms that cope with concept drift have been put forward, however, most of them specialize in one type of change.

A. Single Classification approaches[6]

Traditional Learners are the popular classifiers proposed for stationary data mining fulfill both of the stream mining requirements - have the qualities of an online learner and a forgetting mechanism. Some of the methods are neural networks, Naive Bayes, nearest neighbor methods, and decision rules.

Windowing technique is an approach to dealing with time changing data involves the use of sliding windows, that limits the amount of examples introduced to the learner. They include weighted windows, FISH, ADWIN and so on.

The drift detectors detect concept drift and alarm the base learner(using statistical test). DDM and EDDM are the drift detectors.

Hoeffding Tree or VFDT is the standard decision tree algorithm that uses the Hoeffding bound to decide the minimum number of arriving instances to achieve certain level of confidence in splitting the node.

B. Ensemble Classification Approaches[10]

In ensemble approaches, prediction of multiple classifiers are combined. The figure 2 shows the basic ensemble methodology. The idea of *ensemble* methodology is to build a predictive model by integrating multiple models. They are learning algorithms that construct a set of classifiers and then classify new data points by taking a weighted or unweighted vote of their predictions.

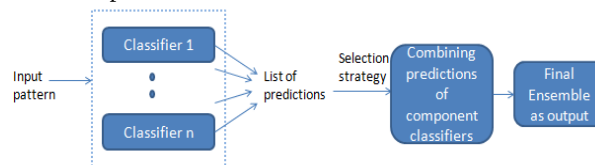


Fig 2 : Ensemble Methodology

Streaming Ensemble Algorithm (SEA) [4] is a heuristic replacement strategy of the weakest base classifier based on accuracy and diversity with simple majority voting and base classifiers unpruned. *Accuracy Weighted Ensemble (AWE)* trains a new classifier *C'* on each incoming data chunk and use that chunk to evaluate all the existing ensemble members to select the best component classifiers.

Adaptive Classifier Ensemble (ACE) is a hybrid approach, in which a data chunk ensemble is aided by a drift detector. *Hoeffding Option Trees (HOT)* provide a compact structure that works like a set of weighted classifiers, and are built in an incremental fashion. It allows each training example to update a set of option nodes rather than just a single leaf. *Adaptive-Size Hoeffding Tree Bagging (ASHT Bagging)* diversifies ensemble members by using trees of different sizes and uses a forgetting mechanism.

Compared to AWE, the *Accuracy Updated Ensemble (AUE1)* conditionally updates component classifiers. It uses Hoeffding trees as component classifiers. Compared to AUE1, AUE2[11] introduces a new weighting function, does not require cross-validation of the candidate classifier, does not keep a classifier buffer, prunes its base learners, and always updates its components. It does not limit base classifier size and use any windows. *The Online Accuracy Updated Ensemble (OAUE)*[12], tries to combine block-based ensembles and online processing.

Generalizing, [1] there are 3 categories of ensemble classifiers.

Expert ensemble classifier method changes integration rules of experts with occurrence of concept drift. Ensemble classifier method updating the basic classifier set assigns "age" property to every basic classifier, then take the youngest basic classifier generated from the latest training sample to replace the oldest basic classifier. Ensemble classifier method with multi-algorithm use a variety of different classification algorithms. Such ensemble classifier methods are fit for dealing with data streams with continuous mutation.

III. Proposed work

The proposed ensemble approach uses Hoeffding option trees(HOT) as the basic classifier set. So, it is an expert ensemble classifier that uses a single algorithm and predicts class based on most recent chunk of data.

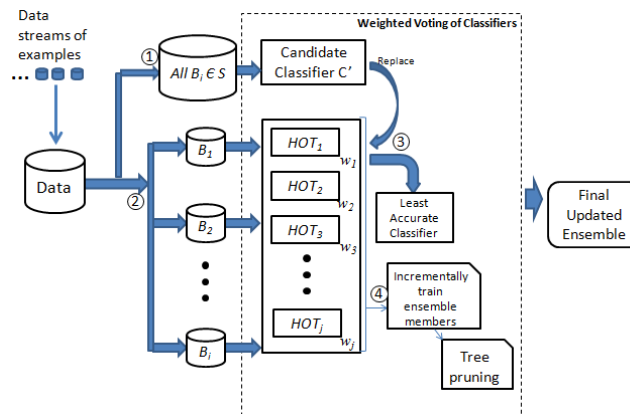


Fig 3 : Working of Revised Accuracy Updated Ensemble(RAUE) algorithm

The architecture below in fig 3, shows working of the proposed ensemble classifier, Revised Accuracy Updated Ensemble(RAUE). The Revised Accuracy Updated Ensemble maintains a weighted pool of HOT component classifiers and predicts the class of incoming examples by combining the predictions of components using a weighted voting rule. It generates HOT component classifiers sequentially from fixed size blocks of training examples called data chunks. In such ensembles, when a new block arrives, existing component classifiers are evaluated and their combination weights are updated. The chunk size and number of classifiers will be 200 and 7 as proved in later sections.

Algorithm 1

Revised Accuracy Updated Ensemble(RAUE)Algorithm

Input: S : data stream of examples partitioned into chunks, k : number of ensemble members, m : memory limit,

E: Ensemble of k members, $k(\text{maximum}) = 7$, $d = 200$

$$l: E \leftarrow \emptyset;$$
2: **for all** data chunks $B_i \in S$ **do**

3: *new component HOT classifier C' built on B_i ;*

4: weight $w_{C'}$ assigned to C' based on (4)

5: **for all** HOT classifiers $C_j \in E$ **do**

6: *apply C_j on B_i to derive MSE_{ij} ;*

7: compute weight w_{ij} based on (3);

8: *end for*

9: **if** $|E| < k$ **then**

10: *add C' to the ensemble E ;*

```

11:  else

```

12: substitute least accurate HOT classifier in E with C' ;

13: *end if*14: **for all** classifiers $C_j \in E$ except C' **do**

15: incrementally train HOT classifier C_i with B_i ;

16: *end for*

17: **if** $\text{memory_usage}(E) > m$ **then**

18: *prune (decrease size of) component HOT classifiers;*19: *end if*20: *end for*

From the algo

Hoeffding Op

given to each

component cl

data chunk B :

data chunk D_i

IJETCAS 15-718; ©

From the algorithm I above for RAUE, the continuous data streams are given as input to the Hoeffding Option Tree(HOT) classifiers. This training data is divided into blocks $B_1 \dots B_n$ and given to each component classifiers $C_1 \dots C_n$. For every incoming chunk B_i , the weights w_{ij} of component classifiers $C_j \in E$ ($j = 1, 2, \dots, k$) are calculated by estimating the error rate on data chunk B_i as shown in (1)-(3)[11]

$$\text{MSE}_{ij} = \frac{1}{|B_i|} \sum_{\{x,y\} \in B_i} (1 - f_{jy}(x))^2 \quad (1)[11]$$

$$MSE_r = \sum_y p(y)[(1 - p(y))^2] \quad (2)[11]$$

$$w_{ij} = \frac{1}{MSE_r + MSE_{ij} + \epsilon} \quad (3)[11]$$

Function $f_{ij}^j(\mathbf{x})$ [11] denotes the probability given by classifier C_j that $\mathbf{x}$ is an instance of class y . The value of MSE_{ij} estimates the prediction error of classifier C_j on chunk B_i , while MSE_r is the mean square error of a randomly predicting classifier and is used as a reference point to the current class distribution. Additionally a very small positive value ϵ is added to the equation to avoid division by zero problems. The candidate classifier C' is trained on the most recent data, it is treated as a “perfect” classifier and assigned a weight according to (4)[11]

$$wC' = \frac{1}{MSE_r + \epsilon} \quad wC' = \frac{1}{MSE_r + \epsilon} \quad (4)$$

The candidate classifier is then used to classify newly arriving data. The component classifiers are incrementally trained for the incoming data.

IV. Experimental Evaluation

The proposed algorithm is evaluated in several experiments to simulate scenarios, involving different real data sets. In the following sections, we describe all of the used datasets, discuss experimental setup, and analyze experiment results.

a. Datasets

Forest Covertype dataset: It contains the forest cover type for 30 x 30 meter cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data.

Poker-Hand dataset: Each record of the Poker-Hand dataset is an example of a hand consisting of five playing cards drawn from a standard deck of 52. Each card is described using two attributes (suit and rank), for a total of 10 predictive attributes. There is one class attribute that describes the “Poker Hand”.

Electricity dataset: It was collected from the Australian New South Wales Electricity Market. In this market, prices are not fixed and are affected by demand and supply of the market. They are set every five minutes. The class label identifies the change of the price relative to a moving average of the last 24 hours.

b. Experimental setup

The proposed algorithm Revised Accuracy Updated Ensemble is implemented in Java using Eclipse IDE. The algorithm, after successful implementation, was added in the classifier list of the MOA framework. The experiments are carried out on Massive Online Analysis(MOA) Tool, developed in Java.

MOA is an open-source framework[8] for dealing with massive evolving data streams. It includes a collection of offline and online methods as well as tools for evaluation. The experiments were performed on a machine equipped with an Intel Core i7-2630QM @ 2.00 GHz processor and 8 GB of RAM.

c. Experiments and results

Below are the results for the comparative analysis of different existing ensemble classification approaches AWE[4], AUE1[3] and AUE2[12] with the proposed RAUE algorithm. Evaluation is carried out based on different performance measures viz. classifier accuracy, memory usage and Kappa statistic.

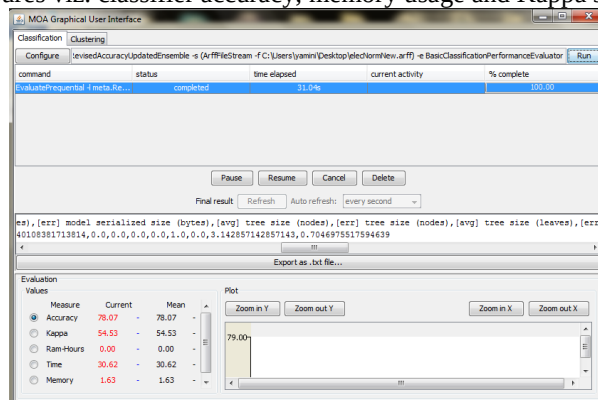


Fig 4 : Simulating different scenarios using MOA tool

These results are carried out on three data sets Electricity, Poker and Forest Covertypes datasets.

i. Comparison w.r.t. Accuracy(in %)

Below shown table I provides accuracy comparison for different state-of-the-art ensemble techniques.

Table I : Recorded results for accuracy analysis of RAUE with state-of-the-art techniques

Dataset	AWE	AUE1	AUE2	RAUE
Electricity	70.92	72.86	76.82	78.35
Poker	60.4	61.29	65.66	67.26
Forest Covertypes	80.48	85.09	86.18	90.36

For the results recorded for accuracy, the graphical representation of the analysis is as shown in figure 5:

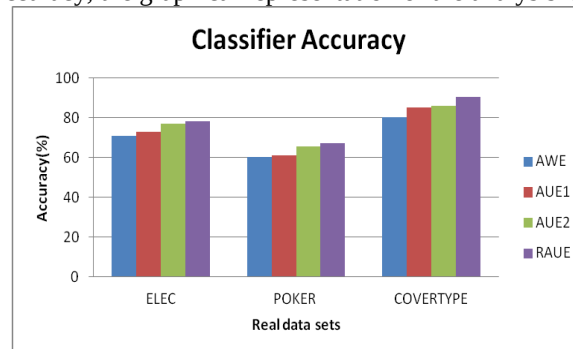


Fig 5 : Graphical representation of classifier accuracy analysis

i. Comparison w.r.t. Memory(in megabytes)

Below shown table II provides memory comparison for different state-of-the-art ensemble techniques.

Table II : Recorded results for memory usage analysis of RAUE with state-of-the-art techniques

Dataset	AWE	AUE1	AUE2	RAUE
Electricity	0.42	0.9	0.95	0.63
Poker	0.36	0.4	0.56	0.24
Forest Covertypes	0.91	3.64	0.48	0.2

For the results recorded for memory, the graphical representation of the analysis is as shown in figure 6:

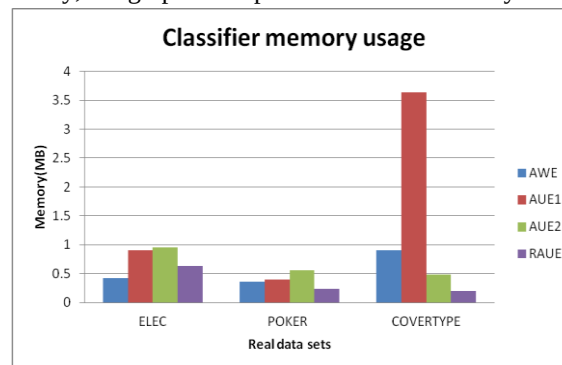


Fig 6 : Graphical representation of classifier memory usage analysis

ii. Comparison w.r.t. Kappa Statistic

Below shown table III provides memory comparison for different state-of-the-art ensemble techniques.

Table III : Recorded results for Kappa statistic analysis of RAUE with state-of-the-art techniques

Dataset	AWE	AUE1	AUE2	RAUE
Electricity	40.46	43.29	53.25	55.23
Poker	31.33	31.56	40.45	40.54
Forest Coverttype	65.04	73.35	79.34	84.49

For the results recorded for memory, the graphical representation of the analysis is as shown in figure 7:

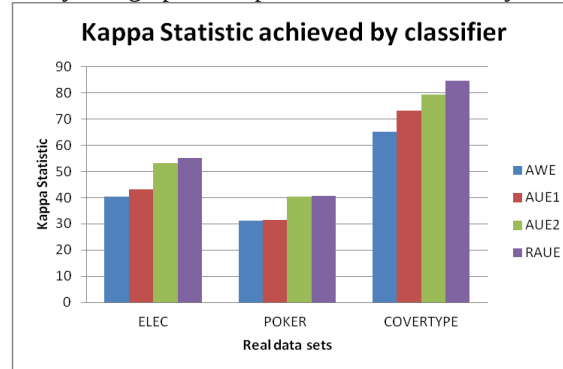


Fig 7 : Graphical representation of classifier Kappa statistic analysis

V. Analysis

From the graphical representation, in figure 5, the classifier accuracy is analyzed. It is clearly seen that RAUE provides more accuracy as compared to the existing ensemble approaches AUE2, AWE and AUE1. The increase in accuracy was observed by almost 5% when compared with other ensemble approaches. In figure 6, the memory usage by each classifier model in comparison is analyzed. It shows that RAUE uses less memory for classification as compared to the existing ensemble approaches AUE2, AWE and AUE1. The memory consumed by the proposed RAUE has decreased by half of the memory consumed by the existing approaches. In figure 7, the Kappa statistic is analyzed. It is seen that RAUE provides more level of agreement between expected and observed accuracy as compared to the existing ensemble approaches AUE2, AWE and AUE1.

VI. Conclusion

In this paper, we have presented and evaluated Revised Accuracy Updated Ensemble(RAUE) classifier for classifying data streams. This ensemble classifier has used Hoeffding Option Trees(HOT) as basic classifier set. The data streams are processed by the component classifiers in blocks. This component classifiers of the ensemble will be updated based on class prediction error. Considering the computational memory and time complexity, the member count as 7 and chunk size as 200 gives better results for accuracy and memory as compared to the existing AUE2 algorithm[11] which uses 500 as chunk size and 10 as member count. Comparison is done based on accuracy achieved by classifier, memory consumed by the model and Kappa Statistic. It is observed that the RAUE identifies class labels more accurately and uses half the memory consumed by the existing approaches.

In future work, we can plan to incorporate clustering techniques with proposed classifier, for detection of newly evolving classes.

References:

- [1] OUYANG Zhenzheng *et al.*, "Study on the Classification of Data Streams with Concept Drift", Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2011.
- [2] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia (2014), "A Survey on Concept Drift Adaptation", ACM Computing Surveys, Vol. 46, No. 4, Article 44.
- [3] D. Brzezinski and J. Stefanowski (2011), "Accuracy updated ensemble for data streams with concept drift," Proc. 6th HAIS Int. Conf. Hybrid Artificial Intelligent. Syst., II, pp. 155–163.
- [4] W. N. Street and Y. Kim (2001), "A streaming ensemble algorithm (SEA) for large-scale classification," in Proc. 7th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 377–382.
- [5] Sobhani P. and Beigy H. (2011), "New drift detection method for data streams", Adaptive and intelligent systems, Lecture notes in computer science, Vol. 6943, pp. 88–97.
- [6] D Brzezinski, J Stefanowski (2011), "Mining data streams with concept drift " Poznan University of Technology Faculty of Computing Science and Management Institute of Computing Science.
- [7] I Žliobaite (2010), "Adaptive Training Set Formation", Doctoral dissertation Physical sciences, informatics (09P) Vilnius University.
- [8] A Bifet (2009), "Adaptive Learning and Mining for Data Streams and Frequent Patterns", Doctoral Thesis.
- [9] D Brzezinski, J Stefanowski (2012), "From Block-based Ensembles to Online Learners In Changing Data Streams: If- and How-To", ECML PKDD Workshop on Instant Interactive Data Mining, pp. 60–965.

- [10] P. B. Dongre, L. G. Malik(2014), " A Review on Real Time Data Stream Classification and Adapting To Various Concept Drift Scenarios", IEEE International Advance Computing Conference (IACC), pp. 533-537.
- [11] D Brzezinski, J Stefanowski (2014),"Reacting to Different Types of Concept Drift:The Accuracy Updated Ensemble Algorithm" IEEE Transactions On Neural Networks And Learning Systems, Vol. 25, pp. 81-94.
- [12] D Brzezinski, J Stefanowski(2014), "Combining block-based and online methods in learning ensembles from concept drifting data streams", An International Journal: Information Sciences 265, 50–67.
- [13] R. Elwell and R. Polikar(2011), "Incremental learning of concept drift in nonstationary environments," IEEE Trans. Neural Networks, vol. 22, no. 10, pp. 1517–1531.
- [14] S S Khan et al., "Comparative Study of Streaming Data Mining Techniques", International Conference on Computing for Sustainable Global Development (INDIACom), 2014, pp. 209-214.
- [15] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia(2014), "A Survey on Concept Drift Adaptation ", *ACM Computing Surveys*, Vol. 46, No. 4, Article 44.
- [16] Hai-Long Nguyen, Yew-Kwong Woon, Wee-Keong Ng, " A survey on data stream clustering and classification", *Journal of Knowledge of Information Systems*,\Springer, December 2014.