Active Learning from Stream Data

[1]Dhotre Virendradkumar A. and [2]Dr. Prakash Jayant Kulkarni

[1,2]Department of Computer Science, Walchand College of Engineering, Sangli
[1]virendradhotre@gmail.com [2]pjkulkarni@yahoo.com

ABSTRACT - In this paper, we propose a new research problem on active learning from data streams where data volumes grow continuously. The objective is to label a small portion of stream data from which a model is derived to predict future instances as accurately as possible. We propose a classifier-ensemble based active learning framework which selectively labels instances from data streams to build an ensemble classifier. Classifier ensemble's variance directly corresponds to its error rates and the efforts of reducing a classifier ensemble's variance is equivalent to improving its prediction accuracy. We introduce a Minimum-Variance principle to guide instance labeling process for data streams. The MV principle and the optimal weighting module are combined to build an active learning framework for data streams.

***Index Terms - Active* learning, classifier ensemble, stream data.**

1. INTRODUCTION

Developments in storage technology and networking architectures have made it possible for broad areas of applications to rely on data streams for quick response and accurate decision making [1]. In the domain of classification, in order to generate a predictive model it is essential to label a set of examples for training purposes. It is well accepted that labeling training examples is a costly procedure [4], which requires comprehensive and intensive investigations on the instances, and incorrectly labeled examples will significantly weaken the performance of the model built from the data [5], [6]. A common practice to address the problem is to use active learning techniques to selectively label a number of instances from which an accurate predictive model can be formed [8]. The goal of the active learning is to achieve a high prediction accuracy classifier by labeling only a very limited number of instances.

1.1 Concept Drifting in Data Streams

The complication of the data stream, in comparison with a static dataset, lies on the fact that one can only observe a portion of the stream data so both $P(x|ci)$ and $P(ci)$ may constantly change/drift across the stream.

Using probability product rule $P(ci,x)=P(ci|x)P(x)=P(x|ci)P(ci)$, for maximizing the posterior\probability where $P(ci)$ defines the priori probability (or density) of the class *ci* and $P(x|ci)$ denotes the class conditional probability of the sample *x* given the class *ci*.

Formally, the concept drifting in the data stream refers to the variance of the priori probability $P(ci)$ and the class conditional probability $P(x|ci)$ across the stream data. The drifting of the concept can be further decomposed into the following three categories: (1) priori probability drifting: the concept drifting is mainly triggered by the class priori probability $P(ci)$ only, (2) conditional probability drifting: the concept drifting is mainly trigged by the class conditional only, and (3) conjunct probability drifting: both $P(ci)$ and $P(x|ci)$ constantly change across the data stream.

1.2 Active learning from data streams

The objective of employing active learning for data streams is to label "important" samples, based on the data observed so far, such that the prediction accuracy on future unseen examples can be maximized. For static data sets whose whole candidate pools can be observed and their genuine decision boundaries are invariant, the active learning is supposed to answer "which samples should be labeled"? if there is no concept drifting involved in the incoming data, it makes sense to save the labeling cost for future samples which may have different distributions from the current data. Unfortunately, implementing such a *when-and-which* labeling paradigm is difficult for data streams; this is mainly because the concept drifting in data streams is mostly triggered by complicated factors so we may not be able to accurately estimate the best time for labeling.

1.3 Challenges of Active Learning from Stream Data

The challenge of active learning from stream data [7] is 1) in data stream environments, the candidate pool is dynamically changing, whereas existing active learning algorithms are mainly designed for static data sets only; 2) the concepts, such as the decision logics and class distributions, of the data streams are continuously evolving [2], [3] whereas existing active learning algorithms only deal with static concepts; and 3) because of the increasing data volumes, building one single model from all the labeled data is computationally expensive for data streams, even if memory is not an issue, whereas most existing active learning algorithms rely on a model built from the whole collection of data for instance labeling.

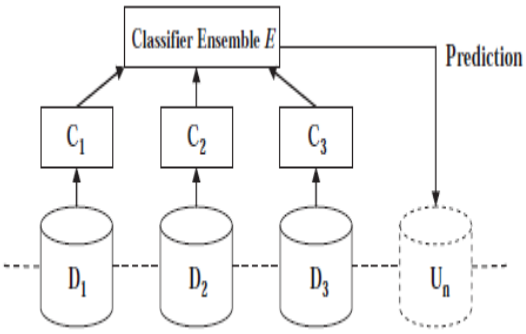
 ISSN : 0975-8283

2. ENSEMBLE FRAMEWORKS FOR MINING DATA STREAMS

The nature of continuous data volumes of the stream data raises the needs of designing effective classifiers with high accuracy in predicting future testing instances as well as good efficiency in handling massive volumes of training instances. An early solution is to build model which update a single model by incorporating newly arrived data [10], [13]. Ensemble learning intends to produce a number of models and relies on their voting for final predictions, such design brings two advantages for ensemble learning to handle data streams: (1) because models are trained from a small portion of stream data, it can efficiently handle streams with fast growing data volumes; and (2) because the final predictions are the voting of a number of base models, the concept drifting in the stream can be adaptively and rapidly addressed by changing the weight value of each voting member. A weighted ensemble, in which each classifier is assigned a weight reversely proportional to the classifiers accuracy on the most recent data chunk.

2.1 Weighted Ensemble Framework

Consider a data stream containing an infinite number of data chunks. Due to limitation of the storage space, the system buffer can only accommodate at most n consecutive chunks each of which contains a certain number instances. The buffered data chunks are denoted by D1, D2, ……, Dn. In order to predict data in a newly arrived chunk Dn+1, one can choose a learning algorithm L to build a base classifier fi from each of the buffered data chunks Di, then predict each instance x in Dn+1 by combining the predictions of the classifiers to form a classifier ensemble.



2.2 A weighted variable ensemble

Weighted ensemble minimizes the variance error e of each base classifier on the up-to-date data chunk then assigns each classifier a weight that is reversely proportional to the error rate. An alternative version of horizontal ensemble is to add weight values to the base classifiers [17], [20]. The advantage of weighted ensemble is 1) They can reuse the information of the buffered data chunks, which may beneficial for testing data chunk; and 2) They are robust to noisy streams because the final decisions are based on the classifiers trained from different chunks. Even if noisy data chunks may deteriorate some base classifiers, the ensemble can still maintain relatively stable prediction accuracy.

The main idea of an ensemble methodology is to combine a set of classifiers a better composite, global classifier. We weigh individual opinions and combine them to reach a final decision. A probabilistic network is used to construct several classifiers for detection. Finally the individual decisions of classifiers are combined to create combining rules. The selected class is chosen according to the highest value in the vector. It can be written as,

$$\text{Class}(x) = \arg\max_{c_i \in \text{dom}(y)} \sum_k \hat{P}_{M_k}(y = c_i|x)$$

In Performance Weighing, the weight of each classifier is set proportional to its accuracy performance on a validation set,

$$\alpha_i = \frac{(1 - E_i)}{\sum_{j=1}^{t} (1 - E_i)}$$

where Ei is the normalization factor which is based on the performance evaluation of classifier on the validation set.

The weight associated with each classifier is the posterior probability is of the classifier given the training set,

$$\text{Class}(x) = \arg\max_{c_i \in \text{dom}(y)} \sum_k P(M_k|S) \cdot \hat{P}_{M_k}(y = c_i|x)$$

Where P(Mk | S) denotes the probability that the classifier Mk is correct given the training set S.

The idea is to create a data set containing a tuple for each tuple in the original data set. However, instead of using the original input attributes, it uses the predicted classifications by the classifiers as the input attributes. The target attributes remains as in the original training set. A test instance is first classified by each of the base classifiers. These classifiers are fed into a training set from which a meta classifier is produced. This classifier combines the different predictions into a final one.

2.3 Active Learning (AL)

The proposed Active Learning as selective method acquiring the most important files in a stream and improving a classifiers performance. In this method the active learner identifies new examples which are

expected to be unknown and present the ranked list of the most informative examples, which probably very different from what is already coded in the classifiers. Major parameters used to generate synthetic data streams

variable	Description
r	Number of attributes
xt	Example generated at time stamp t
yt	Class label of example xt
at	bt Coefficient vectors for generating label yt
ε	Noise
μt	Distribution center of xt
Σt	Covariant matrix of xt
s	Controls concept drifting direction
d	Controls concept drifting step length

3. EXPERIMENTS

To evaluate the performance of AL we carry out experimental strategy studies on both synthetic and real world data streams by implementing all algorithms in Java and the WEKA [18] data mining package. We use Decision Tree [14], Logistic Regression (LR) classifier, and LibSVM [9] to build AL. All tests are carried out on a PC machine with 1.7 G CPU and 2 GB Memory.

3.1 Assessment Criteria

For ease of comparisons, we first summarize the assessment criteria of the ensemble based data stream mining models. Due to importance of prediction accuracy in assessing a classification model, many existing ensemble-based models [15], [16], [19], [20] compare the average prediction accuracy to its peers.

On the other hand, considering that a good ensemble classifier should have high prediction accuracy and low computational overhead. Wang et. al [17] evaluated their method with respect to both the prediction accuracy and system training time. Similar work can be found in many other data stream classification methods [11],[12],[13]. In our experiment, we first compare the ensemble-based models with respect to the prediction accuracy on a synthetic data stream.

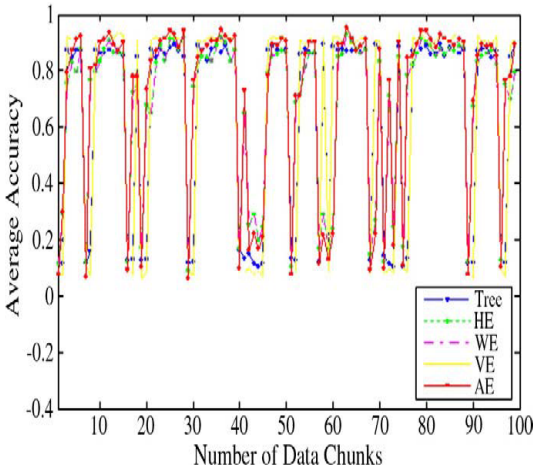
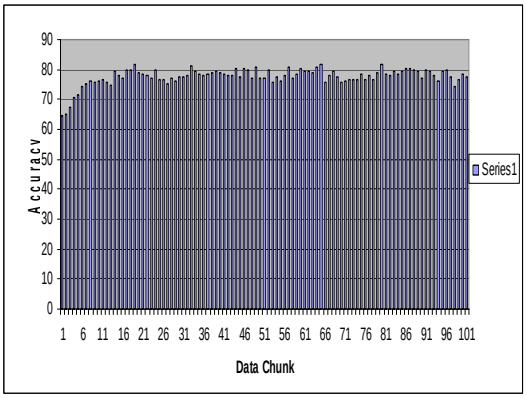


Fig. The two-group synthetic data stream, each chunk has 1000 instances; there are total 100 data chunks

To investigate the situations where concept drifting and noise interruption occur simultaneously, we report accuracies across 100 data chunks in figure we can observe that there is always a significant drop in the accuracy once a noisy data chunk emerges.



As shown in Table 1, when using the normal chunk to predict the noisy chunk. We can observe that AE also performs the best, with the highest average accuracy and ranking, the most winning and least losing chances.

Table 2

Algo	Tree	HL	WL	VL	AL
Acc	0.158	0.767	0.767	0.071	0.808

4. CONCLUSIONS

In order to build accurate prediction models from noisy data streams, existing solutions largely rely on some data preprocessing algorithms to cleanse noise from data streams, such that the cleansed stream data can be used to build accurate prediction models. in this paper, we proposed a robust aggregate ensemble (AE)

ISSN : 0975-8283

learning model to assist the knowledge discovery for noisy data streams. AE first trains base classifiers using different learning algorithms on different data chunks, and then combines all the base classifiers to form an ensemble classifier through model averaging. By doing so, AE is capable of handling the concept drifting challenge, as well as tolerating the data errors. Theoretical and empirical studies demonstrated that AE is superior to existing ensemble-based models, such as the horizontal ensemble, the weighted ensemble and vertical ensemble models, for noisy data streams.

5. REFERENCES

[1] C. Aggarwal, Data Streams: Models and Algorithms, New York: Springer-Verlag, 2007

[2] P. Domingos and G. Hulten, "Mining concept-drifting data streams using ensemble classifiers," in Proc. KDD, 2003, pp. 226-235.

[3] X. Zhu, P. Zhang, X. Wu, D. He, C. Zhang, and Y. Shi, "Cleansing noisy data streams," in Proc. 8th IEEE Int. Conf. Data Mining, 2008, pp. 1139-1144.

[4] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," Mach. Learn., vol. 15, no. 2, pp. 201-221, May 1994.

[5] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in Proc. ICML, 2003, pp. 920-927.

[6] X. Zhu, X. Wu, "Class noise vs attribute noise: A quantitative study of their impacts," Artif. Intell. Rev., vol. 22, no. ¾, pp., 177-210, Nov. 2004.

[7] X. Zhu, P. Zhang, X. Lin, and Y. Shi, "Active learning from data streams," in Proc. ICDM, 2007, pp. 757-762.

[8] W. Hu, W. Hu, N. Xie, and S. Maybank, "Unsupervised active learning based on hierarchical graph-theoretic clustering," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 39, no. 5, pp. 1147-1161, Oct. 2009.

[9] C. Chang, and C. Lin, LIBSVM Toolbox, Available online: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[10] P. Domingos, G. Hulten, Mining high-speed data streams, Proc. Of KDD (2000)

[11] W. Fan, Systematic data selection to mine concept-drifting data streams, Proc. Of KDD (2004) 128-137.

[12] M. Gaber, P. Yu, Detection and Classification ofnChanges in Evolving Data Streams, International Journal of Information Technology and Decision Making (IJITDM) 5 (4) (2006) 659-670.

[13] G. Hulten, L. Spencer, P. Domingos, Mining time-changing data streams, Proc. Of KDD (2001) 97-106.

[14] J. Quinlan, The Effect of Noise on Concept Learning, Machine Learning (1986).

[15] M. Scholz, R. klinkenberg, An ensemble classifier for drifting concepts, Proc. Of ECML/P KDD Workshop on Knowledge Discovery in Data Streams (2005) 53-64.

[16] W. Street, Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, Proc. Of KDD (2001) 377-382.

[17] H. Wang, W. Fan, P. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, Proc. Of KDD (2003) 226-235.

[18] I.Witten, E.Frank, Data Mining:practical machine learning tools and techniques, Morgan Kaufmann(2005).

[19] P. Zhang, X. Zhu, Y. Shi, Categorizing and Mining concept drifting data streams, Proc. Of KDD (2008) 820-821.

[20] X. Zhu, P. Zhang, X. Lin, Y. Shi, Active Learning from Stream data Using Optimal Weight Classifier Ensemble, IEEE Transactions on System, Man, Cybernetics, Part B (2010) 1-15.