Online Active Learning with Drifted Data Streams Using Paired Ensemble Framework

Ji-Cheng Shan[1a], Wei-Ke Liu[2], Chen-Xi Chu[3], Chao-Fan Dai[1], and Qing-Bao Liu[1]

1Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, 410073 Changsha, China
2College of Atmospheric Sciences, Lanzhou University, 730107 Lanzhou, China
3Sino-Dutch Biomedical and Information Engineering School of Northeastern University, 110819 Shenyang, China

Abstract. In learning to classify data streams, it is impractical and expensive to label all of the instances. Online active learning over streaming data poses additional challenges for its increasing volumes and concept drifts. We propose a new online paired ensemble active learning framework consisting of a stable classifier and a timely substituted dynamic classifier to react to different types of concept drifts. Classifiers are built in block based way and will learn new instances incrementally online. According to a combination strategy of uncertainty strategy and random strategy, the decision whether to label the incoming instance for the updating of the stable classifier and the dynamic classifier will be made. Experimental evaluation results on real datasets show the advantage of the proposed work in comparison with other approaches.

1. Introduction

Data streams are widely produced in areas such as financial activities, traffic flow, sensor networks and web applications with the development of storage technology and networking architecture [1]. In these dynamic environments, data streams are massive, temporally ordered, fast changing and potentially infinite [2].

In classification domain for data streams, usually unlabelled data is massive while labelled data is limited. Mostly it will be costly and time consuming to obtain labels of the instances in data streams due to oracle cost. Real time data streams can hardly provide sufficient labelled instances, restricting the generalization capability and predicting accuracy. Therefore, active learning focuses on how to label instances selectively to maximize the prediction accuracy as possible with limited label budget. Based on some criteria, active learning method selects the most representative and informative instances interactively [3]. Usually the representativeness of an incoming instance can be measured by its uncertainty according to the classifier model trained from the labelled instances [4]. Therefore, how to design appropriate metrics to assess the representativeness of an incoming instance becomes important concern [5].

In the scenario of data streams, classifiers need to make the decision whether to request the class label for every incoming instance immediately with no re-access as the data continuously arrive in real time [6]. Due to the non-stationary nature of data streams, concept drift, which refers to changing relations between the input attributes and the target labels, tend to often emerge over time. Moreover the decision threshold or a region of uncertainty cannot be kept fixed as concept drift negatively impacts the accuracy of the model that learned from the past training instances.

Usually concept drifts can be divided into sudden, gradual, or recurring drifts. Typical approaches that deal with concept drift mainly include: sliding window methods, new online means, special detection techniques, and adaptive ensembles [7]. Adaptive ensembles generate base classifiers sequentially from fixed size blocks of training examples called data chunks [8]. However, once the chunk size is too big the ensembles may react too slowly for sudden drifts as old classifiers still have remain weights. It can offer partly help in detecting sudden changes to use small size chunks, while this may also damage the stability and computational costs performance of the ensemble. Simple incremental learning [9-14] keeps sensitivity to sudden changes to make self-adaption timely but is not enough for coping with gradual drifts as forgetting history data and sharp adaptation to newest status. In order to adapt to both sudden and gradual changes, it could be suitable to combine significant features from block-based ensembles and incremental learning approaches.

In this paper, a new online paired ensemble active learning framework consisting of a stable classifier and a timely substituted dynamic classifier is proposed to react to different types of concept drift better. Classifiers are built using block based method. Once built, the two classifiers predict and learn from incoming instances in an online incremental way. The stable classifier will make suitable reactions to gradual changes; meanwhile the dynamic classifier keeps sensitivity to sudden changes.

The rest of the paper is organized as follows. Section II briefly reviews the related work. Section III presents the online paired ensemble active learning framework in detail. Experimental results and analysis are discussed in Section IV, and conclusion is given in Section V.

2. Related Work

In stream data classification, a set of infinite instances $S=\{(x_t,y_t)|t=1,\ldots,T\}$ appear with time flowing, where x_t is a vector for attribute values, y_t is the class label satisfying $y_t \in \{c_1,..,c_K\}$, and t indicates the time sequence. The active learning is to selectively label instances and build a classifier L from them to predict class labels for the future instances.

Žliobaitė et al. [15] presents a generic framework for active learning incrementally from drifting data streams. Several active learning strategies are incorporated into the framework. The strategies are equipped with mechanisms to control and distribute the labelling budget over time for learning more accurate classifiers adapted to changes. The three new proposed effective active learning strategies include *Variable Uncertainty Strategy* (*VarUn*), *Uncertainty Strategy with Randomization* (*RanVarUn*), and *Split Strategy* (*Split*). As discussed in the paper, different strategies perform best in different situations. Usually *RanVarUn* and *Split* strategies work well as they combine the uncertainty strategy and random strategy so that they will label the instances that are close to the decision boundary more often, but occasionally they will also label some distant instances. However with single classifier to learn incrementally, the combination strategy may still miss concept drifts occur in short period or even suffer accuracy decay as the drifted distribution differs from the historical trend. Xu et al. [16] adopts paired learning framework to cope with concept drift, and incorporates hybrid active learning strategies to identify both the most valuable instances and the potential changes in data streams. The stable classifier predicts based on all available labelled instances, while the reactive one predicts based on a window of recent instances from random strategy. So the reactive classifier is trained with instances uniformly distributed in the whole space, while the stable classifier is trained with instances mainly around the decision boundary. Whenever the reactive classifier has a better accuracy, the stable classifier is replaced by the reactive one, and the reactive classifier is reset. This replacing way makes both classifiers forget distributions of instances far away from the current window as the reactive classifier pay more attention to instances labelled in the window. Therefore it weakens the ability of the stable classifier to detect gradual drift. It may also cause late reaction to real drift as the replacement always happens after drift having been detected. So the framework may be over fit to sudden or local drifts and cannot deal gradual and recurring drifts well. To make the classifier be sensitive to both sudden and gradual drifts, this paper consider combining significant features from paired ensembles and incremental learning approaches.

3. Online Paired Ensemble Active Learning Framework for Drifted Data Streams

In this section, the new online paired ensemble active learning framework is described in detail. The main procedure is shown in Fig. 1. A circular array structure buffer window with the block size is used to cache incoming stances. When the window gets filled for the first time, the stable classifier and the first dynamic classifier will be built, as show in Fig. 2. Instances in the window will be accessed in order and be selected to learn according to the uncertainty strategy and random strategy as instances continue to be read from the stream. The new coming instance will replace the instance that has just been processed, as shown in Fig. 3. A new dynamic classifier will be built once the window is fulfilled by new instances again. Along with the building and updating of the new dynamic classifier, the stable classifier will incrementally learn from new instances online.

```
Algorithm 1: Online Paired Ensemble Active Learning Framework

Input: $S$: incoming data streams with unknown label,
$x$: new instance
$W$: chunk size to build new classifier
$C$: circular array of W size to cache instances
$p$: counter for processed instances, initialized with 0
Objective: build $C_d$ and update $C_s$ from $S$ to classify instances as accurate as possible.

While (S hasNext) do
  $x$ = S.nextInstance
  p=p+1
  If (p < W)                    //filling the first chunk
    C[p] = $x$
  Else If (p == W)
    C[p]= x                 //first chunk fulfilled
    CreateNewDynamicClassifier()
  Else          //p > W
    DealInstance(x)   // deal incoming instances after
                       //first chunk fulfilled
  End If
End While
For each instance I in C       //the chunk C still have
                                //W instances to deal
   DealInstance(I)
End For
```

Figure 1. Online paired ensemble active learning framework.

```
Algorithm 2: CreateNewDynamicClassifier()
$L_n$←Randomly label a tiny portion of instances from chunk $C$
Build new classifier $C_d$ from $L_n$
Update the stable classifier $C_s$ with $L_n$
```

Figure 2. Procedure to build new dynamic classifier.

```
Algorithm 3: DealInstance($x$)
Initialization: $i$=0, i is the index for the instances to be processed and cached
instance $I_x$ =C[$i$]   //new instance comes, get cached
                         //instance to from chunk C
*labelling=uncertaintyStrategy($I_x$)*
If *labelling=true*, then
  label $I_x$, then update $C_s$ and $C_d$ with labelled $I_x$
Else
  *labelling =randomStrategy(σ)*
  If *labelling=true*, then
    label $I_x$, then update $C_s$ and $C_d$ with labelled $I_x$
  End If
End If
C[$i$]= x   //the new instance will be cached in position
            // $i$ to overwrite $I_x$
*i=(i+1)%W*
If *i==0* then   // new instances fulfil the chunk again
  CreateNewDynamicClassifier()
End If
```

Figure 3. Algorithm to deal new instance.

When processing instances in the cache window, uncertainty strategy and random strategy are used in order. If an instance $\boldsymbol{x}$ is chosen to be labelled for its high uncertainty, the true class label of the instance $\boldsymbol{x}$ will be requested. If uncertainty strategy is failed to be satisfied, the algorithm then using random strategy to decide whether this x should be labelled. The uncertainty strategy is presented in Algorithm 4. It takes the instance $\boldsymbol{x}$, the ensemble classifier $\boldsymbol{E}$, the uncertainty threshold $\boldsymbol{\theta_m}$ and the adjustment step for threshold $\boldsymbol{s}$ as input. The algorithm calculates the margin for instance $\boldsymbol{x}$ using ensemble classifier $\boldsymbol{E}$ built by the stable classifier $\boldsymbol{C_s}$ and the dynamic classifier $\boldsymbol{C_d}$. Then, it compares the margin of $\boldsymbol{x}$ with threshold $\boldsymbol{\theta_m}$ If *margin(x)<*$\boldsymbol{\theta_m}$, then instance $\boldsymbol{x}$ is requested for labelling (*labelling=true*) and the threshold $\boldsymbol{\theta_m}$ will be adjusted. Otherwise, instance $\boldsymbol{x}$ should not be labelled according to the uncertainty strategy (*labelling=false*). A margin-based metric to measure the uncertainty of an instance considers both the maximum a posteriori probability and the second most probable class label. The margin for instance x is defined as in (1)

$$margin(x)=P_L(\hat{y}_{c_1}|x)\text{-}P_L((\hat{y}_{c_2}|x), \quad (1)$$

where $\hat{y}_{c_1}$ and $\hat{y}_{c_2}$ are respectively the class with the maximum posteriori probability and the second most posteriori probability [17] [18], $\boldsymbol{L}$ is the ensemble classifier used. The margin-based metric is prone to select instances with minimum margin between posteriori probabilities of the two most likely class labels. The random strategy is the same with that used in [16] because of its concision.

```
Algorithm 4: uncertaintyStrategy($x$)
input:  $x$: incoming instance,
        $E$: ensemble classifier built with $C_s$ and $C_d$
        $\theta_m$ =0.6/numberOfClasses ,uncertainty threshold
        $s$=0.1, step to adjust threshold $\theta_m$
output: boolean variable *labelling* indicates whether to
        request the true label of $I_x$.
*margin(x)=*$P_E(\hat{y}_{c1}|x)$*-PE(*$\hat{y}_{c2}$*|x)*;
If (*margin(x)<* $\theta_m$) then
  $\theta_m$ =$\theta_m$ * *(1 - s/numberOfClasses)*;
  return *labelling=true*;
Else
  return *labelling=false* ;
End If
```

Figure 4. Uncertainty active learning strategy algorithm.

4. Experimental Evaluation

In this section, the proposed online paired ensemble active learning framework (*OnPEAL*) is empirically evaluated on real streaming classification problems. *OnPEAL* is compared with the paired ensemble framework for active learning (*PEFAL*) presented in [16] and three representative active learning strategies described in [15], including *Variable Uncertainty Strategy* (*VarUn*), *Uncertainty Strategy with Randomization* (*RanVarUn*), and *Split Strategy* (*Split*). All the experiments are performed using the MOA data stream software suite [19]. MOA is an open source software environment for stream data mining, including evaluation measures and a collection of implemented algorithms. For comparison, the *PEFAL* algorithm is implemented using MOA according to the procedure described in [16] and the active learning classifier using those strategies has already been integrated into MOA by authors of [15]. All algorithms use Hoeffding tree as base classifier to perform the classification and to produce the maximum a posteriori probability.

4.1 Datasets

In the experiments, four real world public datasets referred in [12] are used: Airlines [20], Cover Type [21], Electricity [22], and NSL-KDD [23]. The characteristics of each dataset are listed in Table 1. Detailed information for the datasets can be referred to [12].

Table 1. Dataset characteristics

Dataset	Number of Instances	Number of Attributes	Number of Classes
Airlines	539383	7	2
Cover Type	581012	54	7
Electricity	45312	8	2
NSL-KDD	148517	41	2

4.2 Accuracy Evaluation

In this subsection, the accuracy on different datasets with labelling budget varying from 0.1 to 0.5 is evaluated. Fig. 5 plots the accuracy of different methods as a function of the labelling budget. The results show that *OnPEAL* obtains better accuracies on all the four datasets with limited labelled percentage. For Airlines, the accuracy improvement of the *OnPEAL* is clear when labelled percentage arises. For Cover Type and Electricity, the accuracy is relatively stable with small improvement trend with the labelled percentage increasing. As for the NSL-KDD dataset, the accuracy of *OnPEAL* is stable and the difference between *OnPEAL* and Split is slight when the labelled percentage is higher than 0.2. *OnPEAL* gets high accuracy on all the datasets when the labelled percentage is low and the accuracy keeps stable with improvement trend when the labelled percentage arises. In *OnPEAL* framework, the stable classifier can follow the long-time trend of the instance space and the dynamic classifier keeps sensibility to sudden changes. Therefore the online paired ensemble framework can make adaptive adjustment to drifts in data streams.

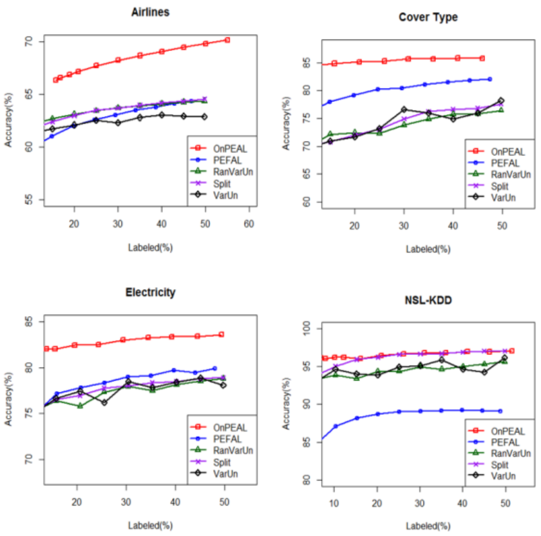

Figure 5. Accuracy on evaluation datasets with different labelled percentages.

4.3 Sensitivity to Parameters

The impact of the window size W over the accuracy and labelled percentage of *OnPEAL* is experimentally evaluated in this subsection. The initial random selection percentage when building new classifier using block based method is set at 0.3 to control the active labelled percentage in certain region. As shown in Table 2, both the accuracy and labelled percentage are relatively stable for Cover Type and NSL-KDD datasets. Airlines gets stable accuracy and decreasing labelled percentage with window size increasing. However, both the accuracy and labelled percentage reduce when window size improves for Electricity. This may be caused by the small amount of Electricity data and its continual drift changes.

Table 2. Accuracy and labelled percentage for different window sizes

	Airlines		Cover Type	
Window Size	Accuracy (%)	Labelled (%)	Accuracy (%)	Labelled (%)
100	69.232	54.738	85.239	35.797
150	69.332	50.924	86.055	35.956
200	69.415	48.371	86.001	36.009
250	69.463	46.511	85.958	36.014
300	69.429	45.102	85.713	35.979
350	69.451	44.024	85.608	35.94
400	69.474	43.099	85.224	35.922
450	69.491	42.33	84.975	35.884
500	69.476	41.705	84.8	35.897
	Electricity		NSL-KDD	
Window Size	Accuracy (%)	Labelled (%)	Accuracy (%)	Labelled (%)
100	85.748	41.121	96.650	36.679
150	84.7	40.759	96.569	36.539
200	84.012	40.167	96.545	36.466
250	83.712	39.867	96.576	36.308
300	83.294	39.467	96.895	36.185
350	83.482	39.017	96.720	36.063
400	82.483	38.736	96.789	35.992
450	82.33	38.301	96.615	35.945
500	82.285	38.258	96.762	35.870

5. Conclusion

Cost limitation of labelled instances and the potential concept drifts have posed significant challenges on stream data classification in practice. Therefore a new online paired ensemble framework for active learning with drifted data streams using combination labelling strategies is proposed. The ensemble classifier consists of a long stable classifier built since beginning and a timely substituted dynamic classifier. Two different active learning strategies, uncertainty strategy and random strategy, are incorporated into the framework in order to find out the most informative instances without missing the potential changes happened anywhere in the instance space. Experimental results on real world datasets demonstrated that the novel approach gets good prediction accuracy. For future work we would like to investigate the ensemble framework and active learning strategies in more detail.

References

1. J. Gama: Knowledge Discovery from Data Streams, 1st ed. Chapman & Hall/CRC (2010).
2. P. Domingos, G. Hulten.: Mining high-speed data streams. Proc. 6th ACM Int. Conf. on Knowledge Discovery and Data Mining, pp. 71–80 (2000).

3. X. Zhu, P. Zhang, X. Lin, Y. Shi.: Active learning from stream data using optimal weight classifier ensemble. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 40, pp. 1607–1621 (2010).
4. J. Attenberg, F. Provost.: Online active inference and learning. Proc. 17th ACM Int. Conf. on Knowledge Discovery and Data Mining, pp. 186–194 (2011).
5. D. Cohn, L. Atlas, R. Ladner.: Improving generalization with active learning. Mach. Learn 15(2), 201–221 (1994).
6. L. I. Kuncheva.: Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. in Proc. 2nd Workshop SUEMA 2008, pp. 5–10 (2008).
7. D. Brzezinski, J. Stefanowski. Reacting to different types of concept drift: the accuracy updated ensemble algorithm. Neural Networks & Learning Systems IEEE Transactions on, 25(1), 81-94 (2014).
8. H. Wang, W. Fan, P. Yu, J. Han.: Mining concept-drifting data streams using ensemble classifiers. in Proc. KDD, pp. 226–235 (2003).
9. R. Elwell, R. Polikar.: Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks, vol. 22, pp. 1517–1531 (2011).
10. A. Bifet et al..: Leveraging bagging for evolving data streams. in Proc. ECML/PKDD, Part I, pp. 135–150 (2010).
11. L. L. Minku, X. Yao.: DDD: A new ensemble approach for dealing with concept drift. IEEE Transactions on Knowledge and Data Engineering, 24(4), pp.619–633 (2012).
12. N. C. Oza, S. J. Russell.: Experimental comparisons of online and batch versions of bagging and boosting. in Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, pp. 359–364 (2001).
13. J. Z. Kolter, M. A. Maloof.: Dynamic weighted majority: An ensemble method for drifting concepts. J. Machine Learning Research, vol. 8, pp. 2755–2790 (2007).
14. R. Kirkby.: Improving Hoeffding trees. Ph.D. dissertation, Department of Computer Science, University of Waikato, (2007).
15. I. Zliobaite, A. Bifet, B. Pfahringer B.: Active learning with drifting streaming data. IEEE Transactions on Neural Networks and Learning Systems, vol. 25, pp. 27–39 (2014).
16. W. Xu, F. Zhao, Z. Lu. "Active learning over evolving data streams using paired ensemble framework." Eighth International Conference on Advanced Computational Intelligence, pp.180-185, (2016).
17. Y. Fu, X. Zhu, B. Li, "A survey on instance selection for active learning," Knowledge and Information Systems, vol. 35, pp. 249–283 (2013).
18. D. Ienco, B. Pfahringer, I. Zliobaite, "High density-focused uncertainty sampling for active learning over evolving stream data, " Proc. 3rd Int. Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, pp. 133–148 (2014).
19. A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer.: MOA: Massive online analysis. Journal of Machine Learning Research, vol. 11, 1601–1604 (2010).
20. E. Ikonomovska, J. Gama, S. Dzeroski.: Learning model trees from evolving data streams. Data Mining Knowl. Discovery, 23(1), 128–168 (2011).
21. A. Frank, A. Asuncion. (2010). UCI machine learning repository [Online]. Available: http://archive.ics.uci.edu/ml
22. M. Harries, C. Sammut, K. Horn.: Extracting hidden context. Machine Learning, 32(2), 101–126 (1998).
23. M. Tavallaee, E. Bagheri, W. Lu.: A detailed analysis of the KDD CUP 99 data set. Proc. 2nd IEEE Symposium on Computational Intelligence for Security and Defence Applications, pp. 1–9 (2009).