International Journal of Database Theory and Application
Vol.7, No.4 (2014), pp. 47-58
http://dx.doi.org/10.14257/ijdta.2014.7.4.05

Mining Textual Stream with Partial Labeled Instances Using Ensemble Framework

Ge Song[1], Yan Li[2], Chunshan Li[3], Jingjing Chen[4,5] and Yunming Ye[6]

[1,3,6]*Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School*
[2]*School of Computer Engineering, Shenzhen Polytechnic, China*
[4]*Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China*
[5]*School of Science and Technology, China Jiliang University, Hangzhou, China*
[1]*carroll0708@qq.com,* [2]*liyan@szpt.edu.cn,* [3]*lichunshan.hit@gmail.com,*
[4,5]*jjchen@comp.hkbu.edu.hk,* [6]*yeyunming@hit.edu.cn*

Abstract

Increasing access to large-scale, high-dimensional and non-stationary streams in many real applications has made it necessary to design new dynamic classification algorithms. Most existing approaches for the textual stream classification are able to train the model relying on labeled data. However, only a limited number of instances can be labeled in a real streaming environment since large-scale data appear at a high speed. Therefore, it is useful to make unlabeled instances available for training and updating the ensemble models. In this paper, we present a new ensemble framework with partial labeled instances for learning from the textual stream. A new semi-supervised cluster-based classifier is proposed as the sub-classifier in our approach. In order to integrate these sub-classifiers, we propose an adaptive selection method. Empirical evaluation of textual streams reveals that our approach outperforms state-of-the-art stream classification algorithms.

Keywords: *Ensemble learning, Cluster-based classifier, Textual stream classification, Unlabeled data, Concept drift*

1. Introduction

The growing availability of data in emails, publishing blogs and providing chatting rooms has made mining and learning from the textual streams increasingly dynamic. Textual stream classification has become a realistic and challenging problem because of three important characters of the stream [14]: the high-dimensional attribute of each instance, the infinite length (large amount) of the stream and the concept drift environment. High-dimensional attribute makes the classification task more difficult compared with low-dimensional data steam. The large-volume instances make manual labeling of data time consuming, thus only fraction of instances in streams are possible to be labeled. It is necessary to utilize unlabeled instances to aid classification. Concept drift is another character of textual stream, which is the change in the attribute or class distribution over the time [1, 3].

As textual stream classification has gained increasing attention, many algorithms have been proposed to classify evolving textual streams [3]. A set of promising approaches for textual stream classification are based on ensemble learning. These methods integrate a certain number of individual sub-classifiers to find the optimal prediction result. A new ensemble approach, Cluster Classifier Ensemble Model with partial labeled instances

ISSN: 2005-4270 IJDTA

(CCEM-PL) is proposed in this paper. We first propose a new cluster-based classifier algorithm (CC) as a sub-classifier in the model to deal with labeled and unlabeled training instances. Then these sub-classifiers are integrated in a proper method to accomplish the optimal prediction result. It is worth to point out that the unlabeled instances in our approach are divided into two classes: instances similar with labeled ones, and the others. If unlabeled instances are similar to labeled instances, they help to find more precise boundary between classes, whereas if they are dissimilar to labeled instances, they could correct the result or detect the new concept. During the ensemble process, we propose an adaptive selection method to select useful sub-classifiers. A threshold is proposed to determine whether the sub-classifier is too "old" to classify a new concept. The performance of CCEM-PL is experimented on several textual stream datasets in comparison to other four baseline ensemble models on Massive Online Analysis (MOA) platform [8]. Experimental results show that CCEM-PL delivers promising performance with respect to the average accuracy and the plotting accuracy.

2. Related Work

In various real life applications, textual stream is characterized by having the high-dimensional feature space, a large number of instances and the concept drift. Many researches seek to overcome the classification in the concept drift environment [17-21, 7]. Only few algorithms focus on utilizing both labeled and unlabeled instances to train and update the classifiers. In the existing study, two kinds of semi-supervised learning are used to deal with the evolving environment [10]. The first one is using clustering methods to label unlabeled instances [3, 9, 10]. Another one is estimate the label of instances using expectation maximization algorithm [11]. In reference [12], a method using clustering algorithm is proposed. The sub-classifier in this method is a set of micro-clusters. To handle recurring concepts, reference [9] utilizes the clustering algorithm to label unlabeled instances. A semi-supervised classification algorithm for data streams with concept drifts and unlabeled data (SUN) [13] is presented to extend reference [9]. SUN builds an incremental tree with concept clusters to labels unlabeled instances. To distinguish the concept drift from noise, a concept detection strategy is proposed in SUN based on deviations between current concept and historical ones. Zahra and Hamid [1] present semi-supervised ensemble learning (SSEL) which uses the majority vote of sub-classifiers to deal with both labeled and unlabeled instances. SSEL improve the self-training approach to improve the performance.

Our new ensemble approach, CCEM-PL is proposed to handle textual streams with partial labeled instances. We proposed a new cluster-based classifier (CC) to deal with labeled and unlabeled instances and ensemble them based on a new voting method. The cluster node in CC is divided into labeled node with both labeled and unlabeled instances and unlabeled node with only unlabeled instances. The prediction results of CC are obtained by labeled node and unlabeled node, respectively. In our proposed CCs, unlabeled instances could help to obtain more precise boundary of labeled nodes according to cluster algorithm. Moreover, they could deal with the testing instances that are dissimilar to labeled instances and accomplish more accuracy result.

3 Cluster Classifier Ensemble Model with Partial Labeled Instances (CCEM-PL)

A new Cluster Classifier Ensemble Model with Partially Labeled instances (CCEM-PL) is presented in this section. This approach aims to handle textual stream classification in non-stationary environment. The main steps are shown as follows: (1) Training. We build the sub-

classifiers on the current labeled chunk and unlabeled chunk. So the original ensemble model can be constructed using the latest sub-classifiers. (2) Adaptive selection. Adaptively select a certain number of sub-classifiers to develop the optimal CCEM-PL. (3) Classifying. Predict the labels of the incoming testing instances (unlabeled instances in the latest chunk) by each sub-classifier. (4) Ensemble voting. Obtain the final label of the testing instances by our voting methods.

It is worth pointing out that our proposed CCEM-PL contains the following key strategies to improve the performance of ensemble model: (1) Building a cluster-based classifier (CC) using labeled training instances and unlabeled testing instances. Compared with most of the existing stream classification algorithms using only historical labeled training chunks to build the sub-classifiers, CCEM-PL makes full use of unlabeled instances at the current time stamp to train the sub-classifiers. Our sub-classifier consists of two kinds of nodes: the node with partially labeled instances (called the real-node) and the node with unlabeled instances (called the virtual-node). The real-node should be formed by labeled instances and some unlabeled instances which are similar to labeled instances, while the virtual-node should be formed by all the unlabeled instances which are dissimilar to labeled instances. (2) Ensemble voting method. In particular, a testing instance obtains two local predictions by a certain CC, that is the prediction by real-node and the prediction by virtual-node. We combine these two kinds of predictions in the "useful" CCs to obtain the ensemble result. In comparetion to the virtual-node, whose label relies on the performance of clustering algorithm, the real-node is more credible in the stability period since training these nodes depends on maximum precision and purity. However, as the virtual-node consists of the instances in the current chunk, it always represents the new concept. Therefore, we should integrate the predictions by both labeled node and unlabeled node of all the selected CCs in a certain weight to obtain the final prediction. The weight consists of two weights: accuracy weight associated with each CCs and the node weight to balance the result by real-node and by the virtual-node.

4. Cluster-based Classifiers (CCs) Using Partially Labeled Instances

In order to deal with a high-dimensional textual stream with partially labeled instances, we choose a new cluster-based classifier based on reference [5] as the sub-classifier of CCEM-PL. The CC algorithm is an approach that combines the decision tree and the clustering algorithm. We define the purity of the node is the maximum sample-frequent for each class in the node [4, 5]. The Maximum Precision of the real-node is the precision for real-node.

A typical training process of CC using labeled and unlabeled instances is as follows (See Algorithm 1): First, training instances are clustered into several clusters by k-means algorithm; Continue to split nodes when both the purity and the maximum precision of a cluster are less than a predefined threshold ($(pur_i < pur_\theta) \vee (\max pre_i < \max pre_\theta)$). If the purity $pur_i = 0$, split nodes until the size of the cluster is lower than the predefined $size > size_\theta$. Testing instances are classified by the Nearest Neighbor (NN) rule. Two predictions are obtained by labeled leaf and unlabeled Leaf.

Algorithm 1 Cluster-based Classifiers (CCs) using partially labeled instances

Output: Cluster-based Classifier
Input: D_t: The data chunk at each time stamp,
$D_t^{train} = D_{t-1}^{L} \cup D_t^{U}$
pur_θ : The threshold of purity
$\max pre_\theta$: The threshold of precision
$size_\theta$: the threshold of the number of a Cluster
1: at the t-th time stamp
2: initialize CC with *X*
3 SELECT($Leaf_i$):
For each $Leaf_i$
4: CASE 1
$(0 < pur_i < pur_\theta) \vee (0 < \max pre_i < \max pre_\theta)$
5: Cluster-based($Leaf_i$)
6: Grow(CC) based on Step 5.
7: CASE 2
$(pur_i > pur_\theta) \wedge (\max pre_i > \max pre_\theta)$
8: Goto Step 3
9: CASE 3 $pur_i = 0$
10: If $size > \max size_\theta$,
11: Cluster-based($Leaf_i$)
12: Grow(SCC) based on Step 11.
13: else
14: Goto Step 3
End if
End case
End for
15: For each ($Leaf_i^L$)
16: If $size < \min size_\theta$
17: REMOVE ($Leaf_i^L$);
Endfor
15: For each ($Leaf_i^L$)
16: PREDICT($Leaf_i^L$) based on $\max pre_i$
Endfor
17: For each $Leaf_i^U$
18: PREDICT($Leaf_i^U$) based on SC
19:Endfor.

5. Selection method and Voting Method

5.1. Selection Method and Accuracy Weight

Selection method in our framework is used to select "good enough" CCs. We select these useful sub-classifiers according to the accuracy weight. When a new textual chunk arrives, we build a new sub-classifier by this chunk and add this sub-classifier to the original forest. To obtain the accuracy weight, we then estimate the accuracy of each sub-classifier using the latest chunk. If the value of this accuracy weight is higher than a certain threshold, these CCs are regarded as the useful sub-classifiers to predict the testing instances. Obviously, the number of sub-classifiers increases during the stable period. If the value of this accuracy weight is less than a certain threshold, we discard the useless historical sub-classifiers. After

selecting all useful CCs to obtain the individual prediction result, the ensemble prediction is obtained by voting method.

We use the following equation to compute the accuracy weight:

$$\phi_i = \frac{\eta_i - \eta_\theta}{\sum_{i=1}^{M} (\eta_i - \eta_\theta)},$$

where M is the number of sub-classifiers, η_i is the accuracy of each tree using real-node, that is:

$$\eta_i = \frac{\left| l_{ij}^L = y_{ij} \right|}{\left| x_{ij} \right|},$$

the threshold η_θ is used to decide whether each tree is discarded or not.

5.2. Voting Strategy

We use two kinds of accuracy to ensemble the individual prediction by each CC. The first one is accuracy weight, which reflects the importance of each CC. The other one ρ is used to balance the individual result by real-node and virtual-node. A voting weight for the j-th testing instance is calculated as

$$v_i(\mathbf{x}_j) = \rho \phi_i f_i^L(\mathbf{x}_j) + (1-\rho)\phi_i f_i^U(\mathbf{x}_j),$$

The ensemble prediction label can be set to the maximum value of ensemble function $f^E(\mathbf{x}_j)$, *i.e.*, (see Algorithm 3),

$$\begin{aligned} l_j &= \arg\max_l f^E(\mathbf{x}_j, l) = \arg\max_l \sum_{i=1}^{M} v_i(\mathbf{x}_j) f_i(\mathbf{x}_j) \\ &= \arg\max_l \left(\sum_{i=1}^{M} (1-\rho)\phi_i f_i^L(\mathbf{x}_j) + \sum_{i=1}^{M} \rho \phi_i f_i^U(\mathbf{x}_j) \right) \end{aligned},$$

6. Experiments

6.1. Datasets

We use the Spam Assassin Collection (Spam for short) [20], Spam-Enron stream, and Spam1 stream as the real-world textual stream. 9,324 instances with 500 attributes in the Spam corpus are arranged chronologically. According to the sigmoid function, we construct Spam-Enron stream by the Spam Assassin Collection and Enron Email Dataset. The Spam-Enron stream consists of 125,000 instances and each instance contains 2044 attributes. We add another stream (Spam2 for short) by sampling 8100 instances from the Spam Assassin stream.

When training CCEM-PL, we assume that 60% randomly chosen instances are labeled. In a comparison, for training baseline methods, all of the instances in the training chunk are labeled instances. For a fair comparison, the CCEM-PL and baseline methods share the same chunk size (See Table 1).

Table 1. The Properties of Textual Streams

	The number of instances	Size of attribute	Training chunk	Testing chunk	The number of time stamps
Spam Assassin	9,324	500	300	300	30
Spam-Enron	125,000	2,044	500	500	250
Spam1	8,100	500	300	300	26

6.2. Compared Models

For comparison, our algorithm has been compared with four state-of-the-art ensemble algorithms, which are much related to our work. These algorithms are: Accuracy Weighted Ensemble (AWE) [6], Accuracy Updated Ensemble (AUE) [16], Leveraging Bag (LB), OzaBagAdwin (OZA).

In our experiment, we choose Random Tree (RT), Random Forest (RF), Hoeffding Tree (HT for short), as basic learners. Since all these three sub-classifiers are used in AUE, AWE, LB, and OZA, respectively, 12 combined methods take part in comparison to CCEM-PL. All these algorithms are implemented in the Massive Online Analysis (MOA) framework [8]. We assign the cost function of SVM c = 1. The maximum number of sub-classifiers in all tested ensemble models to M_{max} = 15 in the Spam-Enron stream, and to M_{max} = 5 in the other streams.

6.3. Results

We experiment on 3 textual streams with concept drift. Evaluation measures in the comparative experiment involve averaged accuracy and plotting accuracy.

Figure 1 shows the average accuracy of the 13 algorithms in spam stream. Our CCEM-PL model obtains the highest accuracy (89.68%) of all the tested algorithms. It achieves 3.09% accuracy improvement in comparison to the second best algorithm, AUE-RF. The other two algorithms with highest average accuracy are LB-HT (86.17%) and AUE-HT (83.08%). The plotting accuracy of these four algorithms is described in Figure 2. The curve of plotting accuracy of CCEM-PL is above the other three curves in most of the time. Though the plotting accuracy of CCEM-PL is lower than LB-HT at the first time stamp, it achieves better performances during the rest of the 29 time stamps. As seen in Figure 2, we can observe that the concept drift occurs at the second time stamp, where the plotting accuracy of these four algorithms drops abruptly. However, CCEM-PL is the least affected by concept drift, for the decrease of the plotting accuracy is only 12%. After the 2nd time stamp, the curves of all the algorithms tend to relatively stable.

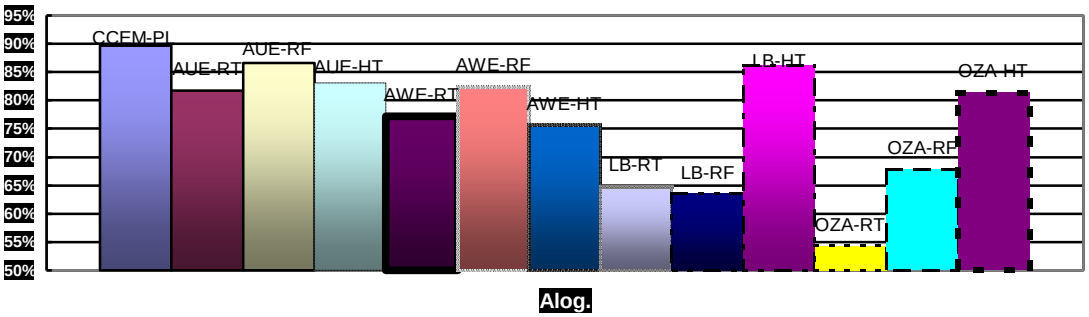


Figure 1. Average Accuracy of Different Algorithms in Spam Stream

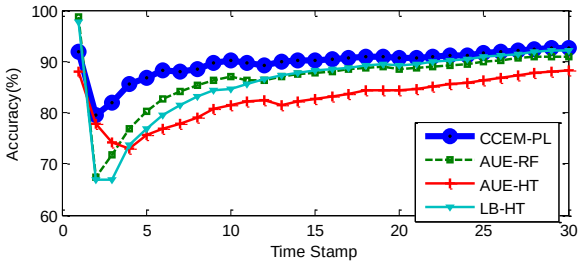


Figure 2. Average Accuracy of Different Algorithms in Spam Stream

Figure 3 summarizes the average accuracy of different approaches in Spam-Enron stream. The average accuracy of CCEM-PL is higher than the other tested algorithms. The best four approaches' (CCEM-PL, AUE-RF, LB-HT, OZA-RF) plotting accuracies are shown in Figure 4. As the fluctuation of the curve in CCEM-PL is relatively smaller than other methods, CCEM-PL accomplishes the better performance regarding to both average accuracy and plotting accuracy. The range of plotting accuracy in CCEM-PL is [35%,100%], while in AUE-RF and OZA-RF, the range of plotting accuracy is almost (0,100%). when a concept changes drastically, other tested approaches fall down even below the level of CCEM-PL at most of the time. In the periods of rebuilding concept, the plotting accuracy of other algorithms grows slowly. This indicates that these algorithms are difficult to react to new concepts, especially after an adrupt concept drift, while CCEM-PL is able to adapt to concept drifts well.

The average accuracy in the Spam1 stream is shown in Figure 5. CCEM-PL gives the best result (91.13%) with respect to average accuracy, while AUE-RF (89.74%), LB-RF (84.28%) and AUR-RT (82.20%) are the second best level of methods. As observed from Figure 6, the plotting accuracy curve of CCEM-PL is above other curves during the first ten time stamps. However, the curve of LB-RF appears more stable than the other algorithms. The reason seems to be that the LB algorithm is based on incremental learning, which relies more on historical data.

We summarize the experimental results of different approaches in terms of the average accuracy and plotting accuracy in three streams. As observed from experimental results, CCEM-PL achieves the highest average accuracy in comparison to all baseline algorithms in all the streams. Moreover, CCEM-PL keeps the higher performance with respect to plotting accuracy in most of the streams.

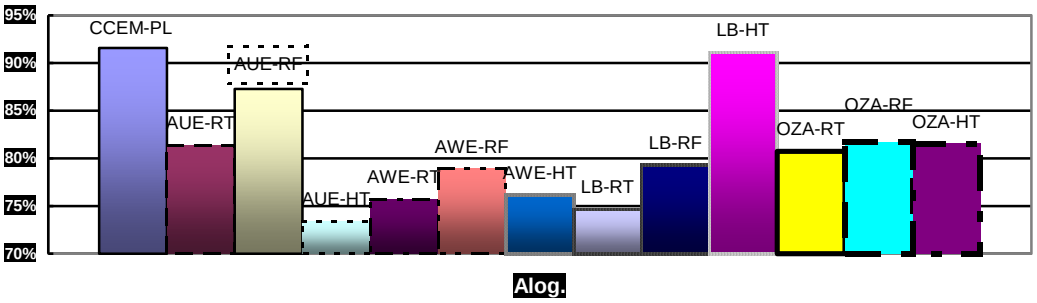


Figure 3. Average Accuracy of Different Algorithms in Spam-Enron Stream

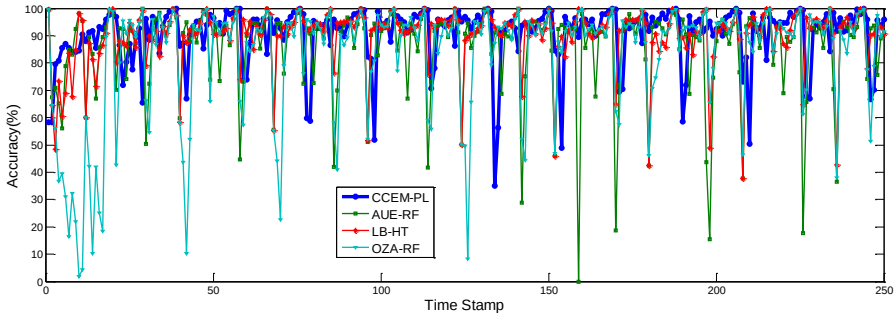


Figure 4. Plotting Accuracy of Different Algorithms in Spam-Enron Stream

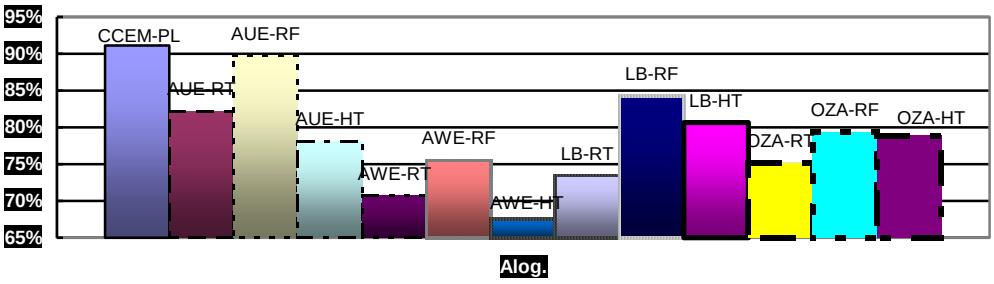


Figure 5. Average Accuracy of Different Algorithms in spam1 Stream

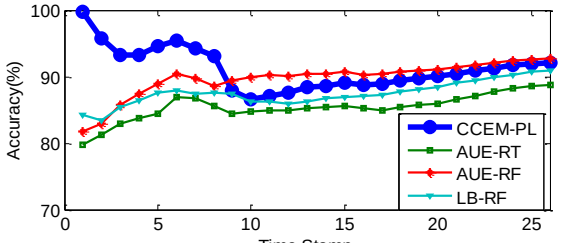


Figure 6. Plotting Accuracy of Different Algorithms in spam1 Stream

7. Conclusion

In this paper, a new ensemble approach, CCEM-PL, is presented to deal with the textual stream classification with partial labeled instances and concept drift. This work involves the following aspects:

We design a new Cluster-based Classifier (CC) as a sub-classifier to deal with labeled and unlabeled instances.

We propose an adaptive selection method to select the better sub-classifiers and a voting method to obtain the ensemble prediction result.

Experiments on textual streams are carried out to evaluate the performances of CCEM-PL, AUE, AWE, LB and OZA based on two evaluation metrics: average accuracy and plotting accuracy. The experimental results demonstrate that our model is more effective than other algorithms on most of the streams.

In the future work, we plan to further extend our work in several aspects. We will improve our proposed algorithm by re-sampling the instances to remedy the model for noisy textual streams.

Acknowledgements

This research was supported in part by NSFC under Grant No. 61300209, and No. 61303103, National Key Technology R&D Program of MOST China under Grant No. 2012BAK17B08, Shenzhen Science and Technology Program under Grant no. JCY20130331150354073, Shenzhen Strategic Emerging Industries Program under Grants No. JCYJ20130329142551746 and No. JCYJ20120613135329670.

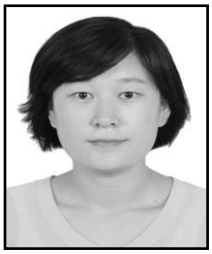
References

[1] Ahmadi Z, Beigy H. “Semi-supervised ensemble learning of data streams in the presence of concept drift”, Hybrid Artificial Intelligent Systems. Springer Berlin Heidelberg(2012), pp. 526-537.
[2] A. Bifet, G. Holmes, B. Pfahringer, R.B. *et al*., “New ensemble methods for evolving data streams.” (2009), pp.139–148.
[3] Woolam C, Masud M M, Khan L. “Lacking labels in the stream: classifying evolving stream data with few labels”, Foundations of Intelligent Systems. Springer Berlin Heidelberg, 2009, pp. 552-562.
[4] Z. Sun, Y. Ye, W. Deng, and Z. Huang. “A Cluster-based tree method for text categorization.” Procedia Engineering, (2011), vol.15, pp.3785–3790.
[5] Y. Li, E. Hung, and K. Chung. “A subspace decision Cluster-based classifier for text classification.” Expert Systems with Applications, vol. 38, no. 10, (2011), pp.12475–12482.
[6] H. Wang, W. Fan, P.S. Yu, and J. Han. “Mining concept-drifting data streams using ensemble classifiers.” In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, (2003), pp. 226–235.
[7] I. Katakis, G. Tsoumakas, and I. Vlahavas. “Dynamic feature space and incremental feature selection for the classification of textual data streams.” Knowledge Discovery from Data Streams, (2006), pp. 107–116.
[8] A. Bifet, G. Holmes, B. Pfahringer, P. Kranen *et al*., “Moa: Massive online analysis, a framework for stream classification and Cluster-baseding.” Journal of Machine Learning Research (JMLR) Workshop and Conference Proceedings, 22, 2010.
[9] Li, P., Wu, X *et al*., “Mining Recurring Concept Drifts with Limited Labeled Streaming Data”, In: 2nd Asian Conference on Machine Learning (ACML 2010). JMLR, Tokyo (2010).
[10] Zhang, P., Zhu, X *et al*., ”Mining Data Streams with Labeled and Unlabeled Training Examples.” In: Proceedings of the 2009 Ninth IEEE International Conference on Data Mining. IEEE Computer Society (2009)
[11] Borchani, H., Larrañaga, P., “Mining Concept-Drifting Data Streams Containing Labeled and Unlabeled Instances”. In: García-Pedrajas, N., Herrera, F., Fyfe, C., Benítez, J.M., Ali, M. (eds.) IEA/AIE 2010, Part I. LNCS, vol. 6096, (2010) pp. 531–540.
[12] Masud M M, Woolam C, Gao J, et al. “Facing the reality of data stream classification: coping with scarcity of labeled data”. Knowledge and information systems, vol.33 no. 1 (2012), pp. 213-244.

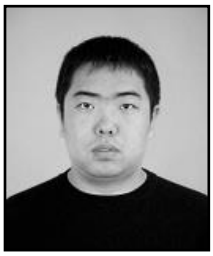
[13] Wu X, Li P, Hu X. "Learning from concept drifting data streams with unlabeled data". Neurocomputing, vol.92, (2012), pp. 145-155.
[14] M. Scholz and R. Klinkenberg. "An ensemble classifier for drifting concepts." In Proceedings of the Second International Workshop on Knowledge Discovery in Data Streams, (2005), pp.53–64.
[15] Ditzler, G., Polikar, R., Chawla, N.V.. "An incremental learning algorithm for nonstationary environments and class imbalance." In: ICPR. IEEE, New York (2010)
[16] D. Brzezi′nski and J. Stefanowski. "Accuracy updated ensemble for data streams with concept drift." Hybrid Artificial Intelligent Systems, (2011), pp.155-163.
[17] X. Wang, C.X. Zhai, X. Hu, and R. Sproat. "Mining correlated bursty topic patterns from coordinated text streams." In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, (2007), pp. 784–793.
[18] X. Li, PS Yu, B. Liu, and S.K. Ng. "Positive unlabeled learning for data stream classification." SDM, SIAM, (2009), pages 256–270.
[19] Y. Zhang, X. Li, and M. Orlowska. "One-class classification of text streams with concept drift." In Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on, (2008), pp.116–125.
[20] I. Katakis, G. Tsoumakas, and I. Vlahavas. "Tracking recurring contexts using ensemble classifiers: an application to email filtering." Knowledge and Information Systems, vol.22, no.3, (2010), pp.371–391.
[21] B. Liu, Y. Xiao, L. Cao, and P.S. Yu. "Vote-based lelc for positive and unlabeled textual data streams." In Data Mining Workshops (ICDMW), 2010 IEEE International Conference on, (2010), pp.951–958.

Authors

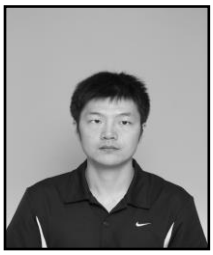
Ge Song, received the M.S. degree in Computer Science from Guizhou University in China in 2010. She is currently working towards the Ph.D. degree in the Department of Computer Science, Harbin Institute of Technology. Her research interests include textual classification, especially textual stream classification based on ensemble model.



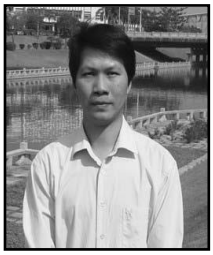
Yan Li, received he Ph.D. in computer science from the Hong Kong Polytechnic University. She was a lecturer at the Software Institute, Nanjing University before her Ph.D. program. Her research interests include data mining, pattern recognition and machine learning. Currently, she is a lecturer at School of Computer Engineering, Shenzhen Polytechnic, China.



Chunshan Li, received the Master degree in Harbin Institute of Technology. He is now a PHD candidate in the Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, text mining, social computing and topic model.



Jingjing Chen, currently is the Ph. D student from the Department of Computer Science, Hong Kong Baptist University. He is also a lecturer of China Jiliang University. His research interests include educational data mining, e-learning technology and computer supported collaborative work.



Yunming Ye, received the Ph.D. degree in Computer Science from Shanghai Jiao Tong University. He is now a professor in the Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, text mining, and ensemble learning algorithms.