Online Data Stream Classification with Incremental Semi-supervised Learning

H. R. Loo
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 Johor Bahru, Malaysia
loohuiru@gmail.com

M. N. Marsono
Faculty of Electrical Engineering
Universiti Teknologi Malaysia
81310 Johor Bahru, Malaysia
nadzir@fke.utm.my

ABSTRACT

This paper proposes an online data stream classification that learns with limited labels using selective self-training. Data partitioning steps are proposed to improve stream mining efficiency. Simulation on Cambridge and KDD'99 datasets shows up to 99.3% average accuracy for 10% labeled data and 98.4% for 1% labeled data. Data partitioning also speeds up classification process by 80% with only 0.2% reduction in accuracy.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

General Terms

Algorithms, Performance

Keywords

Online classification, semi-supervised, data stream mining, incremental learning

1. INTRODUCTION

Online incremental data stream classification classifies and learns simultaneously as data arrive. However, most data stream classifiers assume completely all labeled data. This is not viable as data labeling is time-consuming and requires human inputs [5]. Semi-supervised data mining techniques allow learning based on both labeled and unlabeled data. Hence, they are able to solve limited labeling in data stream mining, although not all are able to perform online classification and incremental learning simultaneously.

Reference [4] proposed a k-mean clustering based classifier with retraining mechanism to handle concept drift. Although it can perform online classification, retraining is dependent on accurate feedback that make it slow to react to concept drift. References [1,3] proposed semi-supervised ensembles that learn with label propagation methods. Both algorithms learn in batch to train new data and to update ensemble models. These methods allow new concepts to be learned without forgetting previously-known concepts. However, the time for retraining is highly dependent on batch size (also known as chunk size). Big batch size results in slow learning whereas small chunk size will reduce the reliability of the training model.

In this paper, an algorithm which is able to incrementally learn from both labeled and unlabeled data while performing online classification, is proposed. We extend the technique proposed in previous work [2] to include distance-based data partitioning to improve classification and retraining speed.

2. PROPOSED METHOD

We propose a selective self-training method to incrementally learn from both labeled and unlabaled data. Hence, the learning delay that is caused by batch retraining is reduced and this allows online classification and learning to be executed simultaneously.

Algorithm 1 shows the overall process of classification and learning. The steps are similar to our previous work [2], except on the determination of confidence level and retraining. The cluster model classifies each incoming data instance and determines the prediction confidence. Let C_1 be the nearest cluster and C_2 be the second nearest cluster, y_{C_i} be the class of C_i, R_{C_i} be the radius of C_i, and μ_{C_i} be the centroid of C_i. Confidence level is set to L_0 by default. In the case of $y_{C_1} = y_{C_2}$, confidence level will increase by one (L_1). If $x_i \in R_{C_1}$ for C_1 with more than one instance or $x_i \approx \mu_{C_i}$ for C_1 with only one instance, confidence level will increase to two (L_2) if the condition $y_{C_1} = y_{C_2}$ is satisfied.

In order to reduce false learning, our algorithm selects only those instances with prediction confidence L_2 for learning. For labeled data, a simple retraining based on Table 1 is initiated. The online classification and learning stage continue processing simultaneously until there are no more incoming data streams. Periodically, cluster reductions are initiated to erase outdated and unused clusters.

In order to reduce classification time, incoming data and clusters can be partitioned, such that only selected clusters are considered in the classification process. Partitioning of data can be based on their position in Euclidean space. The distance of an instance x from the origin o, $d(x,o) = \sqrt{\sum_{m=0}^{d}(x_m)^2}$ is used to determine the partitions.

To apply data and cluster partitioning, the distance of each labeled data to the origin is calculated before pretraining. The distances of each instance in dataset D are sorted and D is partitioned into b blocks based on thesorted

CODS'15, Mar 18-21, 2015, Bangalore, India
ACM 978-1-4503-3436-5/15/03.
http://dx.doi.org/10.1145/2732587.2732614

Algorithm 1 Proposed algorithm

x_i: Incoming data streams
C_1, C_2: First and second nearest clusters from x_i
y_i, $y_i^{'}$: True and predicted labels for x_i
—**Pre-training phase**—
Generate k-cluster using pre-collected data
Summarize k-cluster into Clustering Feature, CF
Store clusters in time-series and set timestamp to 0
—**Classification & Learning**—
while new x_i **do**
  calculate C_1 and C_2
  $y_i^{'} \leftarrow y_{C_i}$
  increase timestammp of C_1
  compute confidence level
  if confidence level $>= 2$ **then**
    merge x_i to C_i
  end if
  if x_i is labeled **then**
    retrain x_i
  end if
end while
—**Cluster Reduction (Periodically)**—
set $ts = 0$
while total cluster $>=$ user-defined threshold, r_k **do**
  if timestamp $= ts$ **then**
    erase cluster
  end if
  if end of series **then**
    increase ts by 1
  end if
end while

Table 1: Retraining Handling Method

Case	Confidence level	Prediction	Procedure
1	0	True	Merge x with C_1 if $x \in R_{C_1}$; else add new cluster for x.
2	0	False	Merge x with C_2 if $x \in R_{C_2}$ and $y_x = y_{C_2}$ then delete $C1$, else add new cluster for x.
3	1	True	If x is not in $R(C2)$, add new cluster for x
4	1	False	Add new cluster for x
5	2	True	Do nothing
6	2	False	Erase $C1$ and add new cluster for x

distances. Parameter b is defined based on data range. When $d(x,o) \in D(\bar{x},o)$, set $b = \frac{(max[D(\bar{x},o)]-min[D(\bar{x},o)]}{mean[D(\bar{x},o)]}$. Each D_b is used to create its own k-clusters, and results in b-cluster model. During simultaneous online classification and traning, $d(x_i,o)$ is calculated to determine its partition block b_i to be used throughout the classification and learning process. The number of clusters in each partition will be limited to $\frac{r_k}{b}$, such that the total number of clusters is maintained as r_k.

3. RESULTS & DISCUSSION

We verify our proposed method using the Cambridge and KDD'99 datasets as in [2]. Table 2 shows the performance comparison between our proposed method and related works [1,3]. Figure 1 shows the accuracy over time for Cambridge dataset. Our proposed classification model outperforms existing work [3] with 14× shorter model update time. The distance-based data partitioning provides 80% speedup in classification time with the tradeoff of 0.2% accuracy.

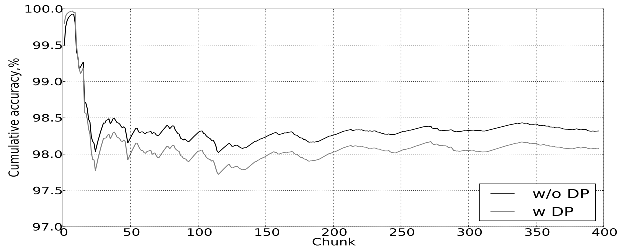

Figure 1: Accuracy of Cambridge dataset

Table 2: Performance comparison

Dataset	Cambridge		KDD'99			
Method	A	B	A	B	C	D
Average accuracy (%)	98.32	98.08	99.53	99.31	96.2	90.89
Running time (s/1,000 instances)	0.134	0.037	0.342	0.059	0.83	-
Classification time (s/1,000 instances)	0.132	0.034	0.335	0.052	0.36	-
Memory required (MB)	0.2	0.2	0.3	0.4	10	-

Note:
A : Our proposed method without data partitioning
B : Our proposed method with data partitioning
C : ReaSC [3]
D : ECMBDF [1]

4. CONCLUSION

This paper proposed an efficient online data stream classification algorithm with incremental learning based on incoming stream with limited label. The proposed model outperforms previous works in terms of both classification accuracy and execution speed.

5. ACKNOWLEDGMENT

The first author is funded by UTM Zamalah schorlaship. This work is funded by Ministry of Science, Technology, and Innovation Science Fund grant (UTM vote no. 4S095)

6. REFERENCES

[1] Liu, J., Xu, G.-s., Zheng, S.-h., Xiao, D., and Gu, L.-z. Data streams classification with ensemble model based on decision-feedback. *The Journal of China Universities of Posts and Telecommunications 21*, 1 (2014), 79–85.

[2] Loo, H., Andromeda, T., and Marsono, M. Online data stream learning and classification with limited labels. *Proceeding of the Electrical Engineering Computer Science and Informatics 1*, 1 (2014), 161–164.

[3] Masud, M. M., Woolam, C., Gao, J., Khan, L., Han, J., Hamlen, K. W., and Oza, N. C. Facing the reality of data stream classification: coping with scarcity of labeled data. *Knowledge and information systems 33*, 1 (2012), 213–244.

[4] Thakar, U., Tewari, V., and Rajan, S. A higher accuracy classifier based on semi-supervised learning. In *Computational Intelligence and Communication Networks (CICN), 2010 International Conference on* (2010), IEEE, pp. 665–668.

[5] Zhu, X. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison 2* (2006), 3.