2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

A New Ensemble Method for Multi-label Data Stream Classification in Non-stationary Environment

Ge Song
Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School Harbin Institute of Technology
Shenzhen, China
carroll0708@qq.com

Yunming Ye
Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen Graduate School Harbin Institute of Technology
Shenzhen, China
yeyunming@hit.edu.cn

Abstract—**Most existing approaches for the data stream classification focus on single-label data in non-stationary environment. In these methods, each instance can only be tagged with one label. However, in many realistic applications, each instance should be tagged with more than one label. To address the challenge of classifying multi-label stream in evolving environment, we propose a novel Multi-Label Dynamic Ensemble (MLDE) approach. The proposed MLDE integrates a number of Multi-Label Cluster-based Classifiers (MLCCs). MLDE includes an adaptive ensemble method and an ensemble voting method with two important weights, subset accuracy weight and similarity weight. Experimental results reveal that MLDE achieves better performance than state-of-the-art multi-label stream classification algorithms.**

Keywords—Ensemble learning; Concept drift; Multi-label classification; Data stream classification

I. Introduction

Nowadays, data is generated at an ever increasing rate from emails, publishing blogs, providing chatting rooms and forums. Real-time analysis of these data streams is becoming a realistic and challenging area of data mining research. Many classification researches from single-label data streams have been proposed, such as the incremental learning algorithms and ensemble learning algorithms [2]. However, in many emerging applications, data streams contain multi-label instances. For example, news on explanations of national defense policies belongs to both political news and military news. Therefore it is necessary to design multi-label classification approaches to accurately and dynamically classify instances into multiple classes.

Multi-label data stream classification has become a problem because of three important characters of the stream [14]: the infinite length of the stream, the concept drift environment and multiple classes of instances. Concept drift is the character of data stream, which is the change in the class distribution or the label of instances over the time [1][3]. The concept drift includes sudden drift, gradual drift and recurring drift. Sudden drift has occurred when the distribution of data abruptly changes. If the distribution changes during a period of time, it referred to as a gradual drift. The recurring concepts are the previous concepts which reappear some time later. In any case, the challenge in concept-drifting environment is to build a classifier that is consistent with the current concept. Multi-label is another particularly challenging of the stream classification. Compared with single-label classification, the challenge of multi-label stream classification is that each instance belongs to a set of labels. The possible label sets may be extremely large even with a small number of labels [17].

In our paper, Multi-label Dynamic Ensemble (MLDE) approach is proposed to deal with multi-label stream classification. Ensemble learning integrates several individual predictions of base classifiers to form a final prediction [1]. As an ensemble learning, MLDE could find a reasonable and adaptive method to gather base classifiers. A new multi-label cluster-based classifier (MLCC) algorithm is used as a base classifier to deal with multi-label problem. We then automatically combine a suitable number of MLCCs by an adaptive ensemble method to accomplish the optimal prediction result. To measure whether the base classifier is suitable for classifying a new concept, a dynamic threshold is defined according to the subset accuracy weight, rather than using the random prediction accuracy in most existing approaches [6][7]. In the ensemble voting method, a similarity weight is defined for each testing instance. This similarity weight relies on the similarity between the testing instance and the center of the cluster that this testing instance belongs to. The performance of MLDE is experimented on several multi-label stream datasets in comparison to other four state-of-the-art ensemble approaches, including IBR Dynamic Ensemble (IDE), Majority Voting Ensemble (MVE), MLOzaBagAdwin [19], and MajorityLabelset. The MLOzaBagAdwin algorithm and the MajorityLabelset algorithm are based on Massive Online Analysis (MOA) platform [8]. Experimental results show that MLDE delivers promising performance in term of four evaluation measures: subset accuracy, Hamming loss, example-based F-measure, and micro-average F-measure.

The rest of the paper is organized as follows. In Section II, we summary the previous works related to our study. In

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

Section III, we introduce the background information of the multi-label stream classification. In Section IV, we present the framework of our approach. We propose the MLCC in Section V. The adaptive ensemble method and ensemble voting method are proposed in Section VI. Experimental results are presented in Section VII. In Section VIII, we conclude the whole paper and give suggestions on future work propositions.

II. Related work

A. The Ensemble Approaches to Mining Data Stream with Concept Drift

Ensemble learning approach is a promising approach for data streaming mining, because: it seems more natural to use different parts of a stream to train base classifiers; it is proved that the ensemble approach can achieve the higher prediction accuracy, if the base classifiers are different from each other. In data stream classification, it is easy to construct an ensemble classifier. As the data stream is divided into unrelated chunks over time, these chunks should guarantee the diversity of base classifiers.

There are two kinds of ensemble approaches in existing ensemble algorithms. The first one is called racing approach [15], such as weighted majority algorithm [18], winnow, and mixture of experts. This approach only updates the weight of base classifiers by frequently verifying each base classifier. The weight is accumulated by all verified races. However, these races may represent both new concepts and old concepts that a stream contains. Moreover, old base learners may not be discarded even if they have become too unwieldy to cope with new concepts.

Another approach aims at changing the structure of ensemble approaches by discarding unsuitable base classifiers. Accuracy Weight Ensemble (AWE) has been proposed by Wang et al. [6]. This algorithm integrates a fixed number of base classifiers to classifying testing instances. Another ensemble method, Accuracy Update Ensemble (AUE) [16], relies on both base classifiers' weights and the current feature distributions. But it is also a fixed-window ensemble method.

B. Multi-label Data Stream Classification

Despite the value and significance, there is very limited research on multi-label data stream classification problem. Some of the existing solutions focus on extending single-label stream classifiers to multi-label cases [17], without addressing some special challenges in multi-label data streams. Reference [17] based on ensemble learning adopts stacked binary relevance model to handle label correlations among multiple labels. This algorithm focuses on the class imbalance and concept drift problems. Binary relevance model is adopted with KNN as the base learner. And an ensemble of fading random trees is proposed in [17] to handle multi-label stream classification. This model can efficiently process high-speed multi-label stream data with concept drifts. Another research on multi-label data stream classifier [2] modifies single-label data stream classification approach. Multi-label Hoeffding Tree [2] is trained by building a batch multi-label classifier on each leaf node.

MLDE is proposed to deal with multi-label streams. Our approach is different with other existing ensemble approaches from the following aspects: (1) In order to process multi-label data classification, Multi-label Cluster-based Classifiers (MLCC) are used as base classifiers since MLCC performs well on multi-label data. Moreover, we can easily track the centroid of the clusters to help us computing the similarity weight in MLCCs. (2) Compared with most existing approaches with fixed-window, we propose an adaptive ensemble method in which the number of selected base classifiers is changed in according to whether the concept drifts or not. If the concept is not changed, the number of base classifiers increases for higher accuracy. When a concept drift is occurred, the number of base classifiers decreases automatically. To accomplish this, we have defined a new subset accuracy weight for the selection of base classifiers. (3) MLDE utilizes more information from the testing data by introducing a similarity weight. In theory, a classifier has different discriminative capabilities for certain parts of a data space. According to this theory, we have defined a similarity weight to reflect whether a base classifier is credible for classifying the current testing sample or not.

Notation	Description
x_{ij}	The jth instance of the ith chunk in the data stream
$\Omega = \{l_i\}$	All candidate labels
$Y_{ij} = \{y_{ij}^k\}$	The true label set of x_{ij}
$L_{ij} = \{l_{ij}^k\}$	The prediction label set of x_{ij}
$f^E = \{f_i\}$	The current ensemble classifiers
$M_{\max}$	The maximum number of f^E
$N = \{N_i\}$	The set of nodes in the MLCC

III. Multi-label Stream Classification

In this section, for clarity, we introduce the definitions of multi-label stream and multi-label data stream classification.

A multi-label data stream with multiple label concepts is an unbounded ordered sequence of instances [17]. It is impossible and unnecessary to process and store all the data in a stream. Only useful data should be processed and stored, while instances will be discarded when they become irrelevant or even harmful to current concepts. The relationship between different concepts usually appears in the stream, such as pairwise relationship.

In our paper, multi-label stream is represented by sequential chunks. Suppose that an incoming data stream is partitioned into a series of chunks with fixed size N in a chronological order, $(\mathrm{D}_1, \mathrm{D}_2 \cdots \mathrm{D}_t, \cdots)$, where D_t is the data chunk at the t-th time stamp. Let χ denote the feature space of instances, and $\Omega = \{l_1 \quad l_2 \quad \cdots \quad l_{|\Omega|}\}$ be the set including all the candidate labels. So the data chunk is represented as $D_t = (\langle \mathbf{x}_{t1}, \mathrm{Y}_{t1} \rangle, \langle \mathbf{x}_{t2}, \mathrm{Y}_{t2} \rangle, \cdots, \langle \mathbf{x}_{tn}, \mathrm{Y}_{tn} \rangle)$, where x_{ij} is the feature vector of the jth instance in the ith data chunk, $x_{ij} \subseteq \chi$. x_{ij} is assigned with a set of labels $Y_{ij} \subseteq \Omega$. This label vector is represented as $Y_{ij} = \{y_{ij}^1 \quad y_{ij}^2 \quad \cdots \quad y_{ij}^{|Y_{ij}|}\}$.

Multi-label stream classification aims to train a (set of) classifier(s) based on both historical and current instances (chunks) in the stream to predict the label sets of incoming instances.

IV. Framework of Multi-label Dynamic Ensemble Classifier (MLDE)

In this paper, we design MLDE for classifying a data stream with concept drift. Fig. 1 gives an overview of our framework. It includes three sections:

- A training section, which is to learn a base learner and build the original multi-label ensemble model;
- A verifying section, which is to adaptively select a certain number of base learners from original multi-label ensemble model;
- A testing section, which is to identify the label set of the incoming testing instance based on a voting method.

For clarity, we summarize the framework of MLDE as follows (the details of MLDE are shown in Algorithm 1):

- Classify testing instances by current MLDE (Step 2-7). We firstly compute the similarity weight for the testing instance (Step 5). We then obtain the global prediction result using a voting method.
- Build base classifiers for the original MLDE (Step 8-12). According to the assumption that only the newest base classifiers are useful for classifying current testing instances, "old enough" base classifiers should be discarded.
- Adaptively select the base classifiers to construct the optimal MLDE (Step 13). To accomplish the optimal MLDE, a subset accuracy weight of each base classifier is computed. We then combine the base classifiers based on an adaptive ensemble method to achieve the optimal MLDE.

Algorithm 1 Multi-label Dynamic Ensemble Classifier (MLDE)

```
Output: $L_{ij}$, $\mathrm{con}_{l_i}$
Input: $D_i^{test}$, $D_i^{train}$, $M_{\max}$
1:  for each time stamp t-th do
2:    for $\forall x_{ij} \in D_i^{test}$ do
3:      for $\forall f_k \in f^E$ do
4:        $(L_{ij}^k, \mathrm{con}_{l_i}^k) \leftarrow$ PREDICT( $x_{ij}$ )i
5:        OBTAIN( $W_{sim}(\mathrm{x}_{ij}, \mathrm{f}_k)$ )
6:        VOTE( $f_k$ )
7:      $(L_{ij}, \mathrm{con}_{l_i}) \leftarrow$ ENSEMBLEPREDICT ( $x_{ij}$ )
8:    BUILD( $f_t$ ) using $D_t$
9:    if $t \le M_{\max}$ then
10:     $f^E \leftarrow f^E \cup f_t$
11:   elseif $t > M_{\max}$ then
12:     $f^E \leftarrow f_{t-M_{\max}+1} \cup \cdots \cup f_t$
13:   UPDATE( $f^E$ )
```

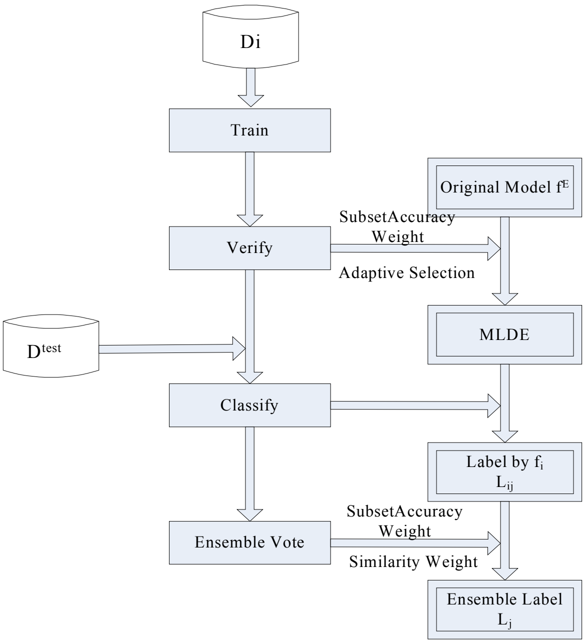

Fig. 1. The process of MLDE

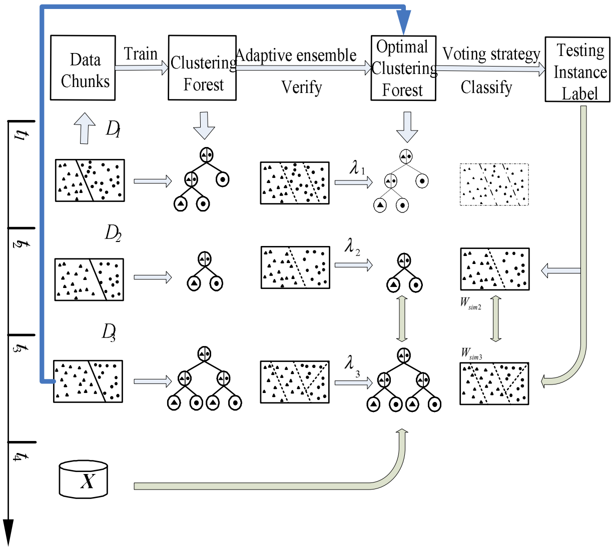

Fig. 2. An example of MLDE

For clarity, we give an example to illustrate the above-mentioned framework. Distributions of D_1, D_2 and D_3 are shown in Fig. 2. The solid line is the decision boundary of the true data distribution between Label-set 1 and Label-set 2 (marked by triangles and cycles, respectively). At the t_3 time stamp, we first built the base classifier f_3 based on data chunk D_3. Suppose that MLCC f_1, f_2 and f_3 are the three latest classifiers ($M_{\max} = 3$). Then three classifiers are verified using D_3, and acquire the subset accuracy weight λ_1, λ_2 and λ_3, respectively. As the subset accuracy weight $\lambda_1 < \lambda_\theta$ (that means many instances in D_3 are misclassified by f_1), the MLCC f_1 is discarded. At the t_4 time stamp, the arrival of the testing instance x should be classified. We first obtain the two predictions of the testing instance by two base classifiers, f_2 and f_3. Then the similarity weights W_{sim2} and W_{sim3} of f_2 and f_3 should be compute. The global prediction is acquired based on ensemble voting method.

V. Multi-label Cluster-based classifier (MLCC)

In order to deal with a multi-label data stream, we choose a multi-label cluster-based classifier (MLCC) based on [5] as the base classifier of MLDE. The MLCC algorithm combines the decision tree and the clustering algorithm. A typical training process of MLCC is described as following: a training chunk is clustered into several clusters by a clustering algorithm; continue to split nodes if the class-purity of a cluster is not higher than a predefined threshold. Testing samples are classified by the Nearest Neighbor (NN) rule. More details can be found in [5]. Purity and the label set of the node N_i in MLCCs are described below.

Definition 1: (Purity) Let x belonging to a certain node (cluster i) be an instance having class label set $Y=\left\{y^1 \quad y^2 \quad \cdots \quad y^{|Y|}\right\}$. The purity of the node is the maximum sample-frequent for label set in the node [4][5]. Formally, we define

$$pur_i = \frac{\max_Y |Y|_{(x,\mathrm{Y})\in N_i}}{|x|_{(x,\mathrm{Y})\in N_i}}, \tag{1}$$

where $|x|_{(x,\mathrm{Y})\in N_i}$ is the total number of instances in the node.

Definition 2: (Label Set of Node) Label set of a node N_i relies on the maximum sample-frequent. It is formally defined by:

$$L_{N_i} = \arg\max_Y |Y|_{(x,\mathrm{Y})\in N_i} \,. \tag{2}$$

VI. Handling Concept Drifts

We focus on the useful information of historical (training) instances and incoming (testing) instances to handle concept drifts. To deal with historical instances, we propose the adaptive ensemble method to integrate a certain number of "good enough" MLCCs. We present the similarity weight to make full use of incoming instances which represent the current concept.

A. Subset Accuracy Weight and Similarity Weight

Subset accuracy is used to select the useful MLCCs and give the ensemble voting prediction result of MLDE. We use the following equation to compute the subset accuracy weight λ_i of each MLCC:

$$\lambda_i = \frac{SA_i \text{-} \lambda_\theta}{\sum_{i=1}^{M}(SA_i \text{-} \lambda_\theta)}, \tag{3}$$

where SA_i is the subset accuracy of each MLCC and the threshold λ_θ is used to decide whether each MLCC is discarded or not. The threshold λ_θ is given by

$$\lambda_\theta = \begin{cases} MinSA_i & \overline{SA} - MinSA_i \le \varepsilon \\ \overline{SA} & \overline{SA} - MinSA_i > \varepsilon \end{cases}. \tag{4}$$

In the subset accuracy weight, the number of MLCCs changes adaptively according to different conditions (with concept drift and without concept drift). Moreover, we use the threshold λ_θ to discard the MLCCs with lower accuracies and to select the MLCCs that are more efficient for classification.

In MLDE, testing data in a stream can be fully utilized to construct the training model. Similarity weight is defined relying on the similarity between a testing instance and the centroid of leaf node that this instance belongs to. Unlike subset accuracy weight, which may become less useful when concept drift occurs, similarity weight using current instances seems to measures the credible level of a MLCC, especially in non-stationary environment.

For the i-th MLCC, a similarity weight $W_{sim}(\mathrm{x}_{ij},\mathrm{f}_k)$ of the testing instance x_{ij} is calculated as

$$W_{sim}(\mathrm{x}_{ij},\mathrm{f}_k) = \frac{sim(\mathrm{x}_{ij},\mathrm{N}_k)_{L_{ij}=L_{\mathrm{N}_k}}}{\sum_{k=1}^{M} sim(\mathrm{x}_{ij},\mathrm{N}_k)_{L_{ij}=L_{\mathrm{N}_k}}}, \tag{5}$$

where M is the number of the MLCCs, $sim(\mathrm{x}_{ij},\mathrm{N}_k)$ is the similarity between the instance x_{ij} and the node N_k of MLCC f_k ($N_k \in f_k$).

B. Adaptive Ensemble Method

Adaptive ensemble method in our framework is used to select "good enough" MLCCs. At each time stamp, we compute the subset accuracy weight of each MLCC. If the value of this weight is above a certain threshold, this MLCC is regarded as the useful base classifier to predict the testing instances. The number of MLCCs increases if the concept drift does not occur. Algorithm 2 illustrates the selection method in MLDE. When a new chunk arrives, we build a new MLCC by this chunk and add this MLCC to the original MLDE. It is worth noting that the original MLDE contains the latest M MLCCs. We then estimate the subset accuracy of all the MLCCs by this new data chunk. After achieving the subset accuracy weight, we use all the selected MLCCs to classify the testing instance.

C. Ensemble Voting Method

The process of ensemble voting algorithms is seen in algorithm 2. A voting weight for the testing instance x_{ij} is calculated as

$$W(\mathrm{x}_{ij},\mathrm{f}_k) = \lambda_k W_{sim}(\mathrm{x}_{ij},\mathrm{f}_k)\,. \tag{6}$$

The ensemble prediction label set L_{ij} can be set to the maximum value of ensemble function $f^E(\mathbf{x}_{ij})$, i.e.,

$$\begin{aligned} L_{ij} &= \arg\max_{L_{ij}} f^E(\mathbf{x}_{ij}) \\ &= \arg\max_{L_{ij}} \sum_{k=1}^{M} v_k(\mathbf{x}_{ij}) f_k(\mathbf{x}_{ij}) \\ &= \arg\max_{L_{ij}} \sum_{k=1}^{M} \lambda_k W_{sim}(\mathrm{x}_{ij},\mathrm{f}_k) f_k(\mathbf{x}_{ij}) \end{aligned} \,. \tag{7}$$

Algorithm 2 Voting Method

```
Output: $L_{ij}$ , $\text{con}_{l_i}$
Input: $D_i^{test}$ , $D_i^{train}$ , $M_{\max}$
//Adaptive select steps:
for each time stamp t-th do
  OBTAIN( $f^E$ )
  for $\forall f_k \in f^E$ do
    ESTIMATE( $ACC_k$ )
    COMPUTE( $\lambda_k$ )
    if $\lambda_k < 0$ then
      REMOVE( $f_k$ )
      UPDATE( $f^E$ )
//Voting steps
for each time stamp (t+1)-th do
OBTAIN( $f^E$ )
for $\forall x_{ij} \in D_i^{test}$ do
  for $\forall f_k \in f^E$ do
    $(L_{ij}^k, \text{con}_{l_i}^k) \leftarrow$ PREDICT( $x_{ij}$ )i
    OBTAIN( $W_{sim}(\text{x}_{ij}, \text{f}_k)$ )
  for $\forall L_{ij} \in \Omega$ by each $f_k$ do
    COMPUTE( $W(\text{x}_{ij}, \text{f}_k, L_{ij})$ )
  OBTAIN( $\max W(\text{x}_{ij}, \text{f}_k, L_{ij})$ )
  $(L_{ij}, \text{con}_{l_i}) \leftarrow \arg\max_{(L_{ij}, \text{con}_{l_i})} W(\text{x}_{ij}, \text{f}_k, L_{ij})$
```

VII. Experiments

A. Datasets

Lacking of benchmark datasets is the problem of performance evaluation on multi-label stream. We choose Reuters21578-top10, Ohsumed and tmc2007 collection as basic datasets to simulate the multi-label streams. Reuters21578-top10 corpus consists of the 10 most frequent classes. After preprocessing, this corpus contains 9034 instances with 500 attributes. Ohsumed corpus is a subset consisting of medical articles, labeled with disease categories. We select 6286 instances with 14527 attributes in this corpus. Tmc2007 corpus consists of 28596 instances with 522 attributes (after preprocessing). These instances belong to 22 different categories.

Four multi-label streams are used in our experiments. Two of them are synthetic multi-label streams on the gradual and sudden drifting environment, respectively. In the simulated gradual stream generated from Reuters (called Reuters Gradual stream), three concepts are formed in this stream. Part of each concept is gradually changed into another one over ten time stamps. Likewise, the sudden concept drifting is simulated based on tmc2007 corpus (called Tmc Sudden stream). Two concepts are generated during 10 time stamps. At the 5-th time stamp where the sudden drift happens, one concept is changed into the other one. More detailed information about the synthetic streams is described in Table 1.

We construct two multi-label streams by the Reuters, Ohsumed and Tmc collections. The sigmoid function is used to formulate a weighted combination of two datasets in order to characterize the target concepts [8]. The first stream (called Reuters-Tmc stream) consists of 20000 instances with 1031 attributes. 40 chunks are divided from Reuters-Tmc stream, and each chunk contains 500 instances. The second stream (called Reuters-Oh stream) is generated by Reuters and Ohsumed collections, which still contains 20000 instances with 1031 attributes. But this stream is divided into 20 chunks, each chunk contains 1000 instances. More detailed information on the above three streams are seen in Table 1.

We arrange the training chunks and the testing chunks by the Interleaved Chunk method. According to this method, we first collect the instances to construct a testing chunk. After evaluating the base learners by this chunk, we train it again to update the training model.

B. Evaluation Measures and Benchmark Methods

Two kinds of evaluation measures are used in our paper. The first one is example-based measure, such as subset accuracy and example-based F measure. This kind of measures is evaluated based on the average differences of the true and the predicted sets of labels. Another one is called label-based measure, such as Hamming loss and micro-averaged F measure. To achieve the label-based evaluation, we first separately evaluate instances for each label, then average over all labels [20]. More details are shown in [19][20].

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:

- IBR Dynamic Ensemble (IDE) [2]: This approach is a fixed-window approach based on the assumption that “old enough” base classifiers should be discarded. Therefore, it always removes the oldest base classifier from the ensemble when updating.
- Majority Voting Ensemble (MVE): This ensemble method is widely used in single-label streams with base classifiers sharing the same weight. The global prediction tends to the prediction of most base learners.
- MLOzaBagAdwin [19]: This method uses the Adwin algorithm to detect and estimate the drift for providing the ensemble weights of the boosting method.
- MajorityLabelset: Majority-class algorithm is widely used in single-class classification. MajorityLabelset is the multi-label version of majority-class.

In IDE and MVE method, MLKNN and Binary Relevance are used as the basic learners. The number of base classifies in these ensemble models are 5 in the Reuters-Gradual and Tmc-Sudden streams, and 10 in Reuters-Tmc and Reuters-Oh streams.

TABLE I. THE PROPERTIES OF STREAMS

	Reuters- Gradual	Tmc- Sudden	Reuters -Tmc	Reuters -Oh
Instances /chunk	900	5,600	500	1,000
Attributes	500	522	1,031	1031
Concept	3	2	--	--
Time stamp	30	10	40	20
Total instances	27,000	56,000	20,000	20,000
Label	10	22	10	10

C. *Results*

We experiment on four streams. In view of the overall performance, the best performance is delivered by MLDE with respects to subset accuracy, Hamming loss, example-based F measure, and micro-average F measure. We summarize the results of different approaches in terms of the Hamming Loss for all the streams in Table 2. We also describe the subset accuracy of tested algorithms in Table 3. The bold numbers in Table 1 and Table 2 show the three algorithms which achieve the highest value of evaluation indexes. In order to better distinguish the performance of our MLDE and other algorithms and to clearly describe the trend of tested algorithms at each time stamp, we show the plotting indexes of our MLDE and the other best algorithm in Fig. 3-18. As observed from the experimental results, MLDE achieves the highest performance in comparison to all the algorithms in most of the streams.

TABLE II. HAMMING LOSS OF DIFFERENT ALGORITHMS ON ALL SCENARIOS

	Reuters- Gradual	Tmc- Sudden	Reuters -Tmc	Reuters-Oh
MLDE	**0.0471**	**0.0199**	**0.0530**	**0.0449**
IDE-MLKNN	0.0572	0.0711	0.0772	0.0586
IDE-BR	0.0686	0.0689	0.0703	0.0972
MVE-MLKNN	0.0562	0.0678	0.0701	0.0465
MVE-BR	0.0723	0.0705	0.0727	0.0966
MLOzaBagAdwin	**0.0268**	**0.0490**	**0.0337**	**0.0122**
MajorityLabelset	**0.0334**	**0.0527**	**0.0178**	**0.0127**

Fig. 3-6 with the time stamps on the x-axis describe the results of MLDE and tested algorithms in Reuters-Gradual Stream regarding to Hamming loss, subset accuracy , example-based F measure, and micro-average F measure, respectively. Though the MajorityLabelset perform better with a respect to Hamming Lose, its subset accuracy is too low compared to other methods. But our MLDE achieve the better performance regarding to subset accuracy, example-based F measure, and micro-average F measure for the gradual drift stream generated from Reuters. Fig. 4 shows the plotting subset accuracies of MLDE and IDE-BR. When the gradual drift occurs at the 11-th and 21-th time stamps, all of the methods react to the changes with a great drop of plotting subset accuracy. After adjusting to the new concept, these algorithms produce the improvements of plotting subset accuracy gradually. But MLDE still outperforms the other method. In comparison to curve of IDE-BR, the curve of MLDE is still above it. During the periods with stable concepts, the plotting accuracies of MLDE are steady. Similar phenomenon is described in Fig. 5-6.

TABLE III. SUBSET ACCURACY OF DIFFERENT ALGORITHMS ON ALL SCENARIOS

	Reuters- Gradual	Tmc- Sudden	Reuters- Tmc	Reuters -Oh
MLDE	**0.7277**	**0.8184**	**0.7503**	**0.7801**
IDE-MLKNN	0.5010	**0.3202**	0.5229	0.5300
IDE-BR	**0.5754**	0.2525	**0.6065**	**0.6690**
MVE-MLKNN	0.5067	**0.3211**	**0.5289**	0.5355
MVE-BR	0.5238	0.2305	0.5881	**0.6605**
MLOzaBagAdwin	0.1240	0.0686	0	0
MajorityLabelset	**0.6470**	0.0322	0.5585	0.1503

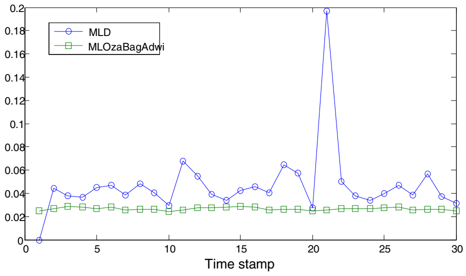

Fig. 3. Hamming Loss of tested approaches in Reuters-Gradual stream

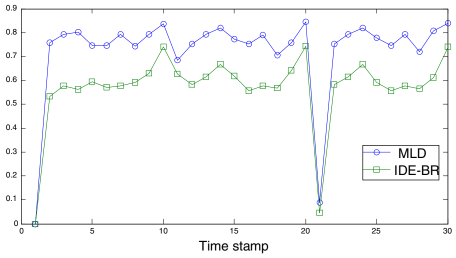

Fig. 4. Subset accuracy of tested approaches in Reuters-Gradual stream

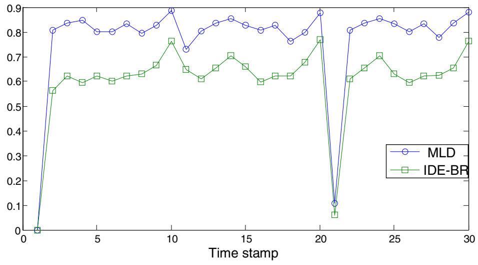

Fig. 5. Example-based F measure of tested approaches in Reuters-Gradual

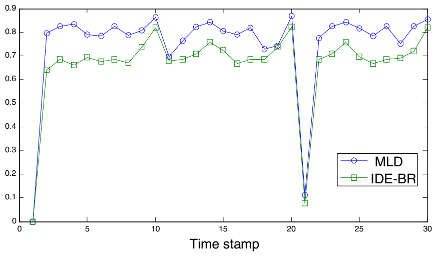

Fig. 6. Micro-average F measure of tested approaches in Reuters-Gradual

We analyze the ability of MLDE to cope with the sudden drift scenario in Tmc-Sudden stream. Fig. 7-10 illustrate the results of different algorithms with Hamming loss, subset accuracy, example-based F measure, and micro-average F measure, respectively. From results, MLDE is still the best algorithm. In Fig. 8, all the methods respond to the concept change with a large decrement of the curve at the 5-th time stamp. In the period of rebuilding models, the plotting subset accuracy obtained by MLDE increases gradually, because the method is reconstructed after adjusting to the new concept. We observe that MLDE obtains relatively stable plotting subset accuracy after concept drift happens. Thus, we can clearly distinguish the stability period from the drift period.

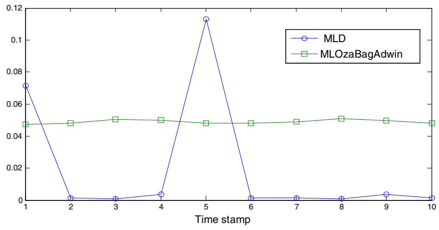

Fig. 7. Hamming Loss of tested approaches in Tmc-Sudden stream

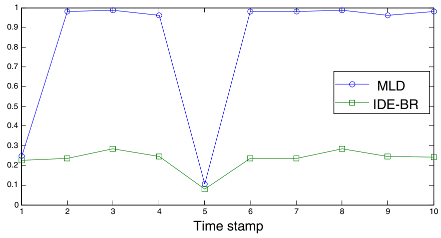

Fig. 8. Subset Accuracy of tested approaches in Tmc-Sudden stream

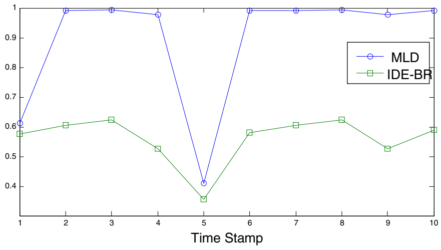

Fig. 9. Example-based F measure of tested approach in Tmc-Sudden

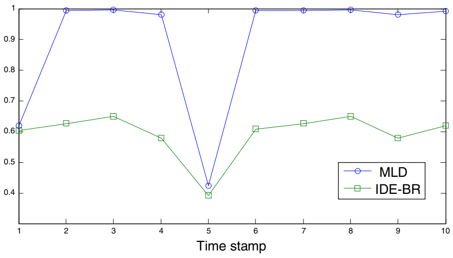

Fig. 10. Micro-average F measure of tested approach in Tmc-Sudden

In Reuters-Tmc stream (see Fig. 11-14), though the Hamming Loss of MLDE is a little higher than MajorityLabelset and MLOzaBagAdwin, MLDE gains the great improvement for subset accuracy measure, example-based F measure and micro-average F measure in comparison to the other 5 algorithms. It is noticed that MLDE accomplishes the better performance in term of the plotting subset accuracy and example-based F measure, as the fluctuation of curves in MLDE is smaller than that in the other methods as seen in Fig. 12-13. For example, we can observe that a concept changes during the period between the 6-th stamp and the 12nd stamp. MLDE keeps steady and performs slightly better if a relatively small change of a concept occurs. Likewise, MLDE delivers the best performance in the Reuters-Oh stream illustrated in Fig. 15-18 owing to the quick reaction of MLDE to drifts and keeps steadier in the rebuilding period.

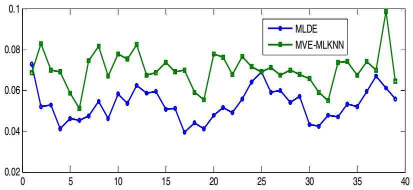

Fig. 11. Hamming Loss of tested approach in Reuters-Tmc stream

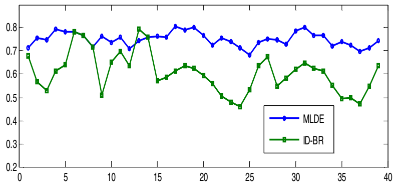

Fig. 12. Subset accuracy of tested approach in Reuters-Tmc stream

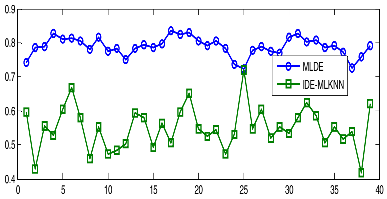

Fig. 13. Example-based F measure of tested approach in Reuters-Tmc

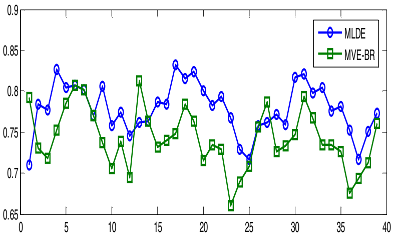

Fig. 14. Micro-average F measure of tested approach in Reuters-Tmc

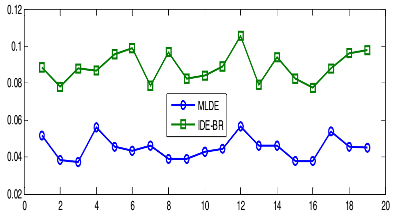

Fig. 15. Hamming Loss of tested approach in Reuters-Oh stream

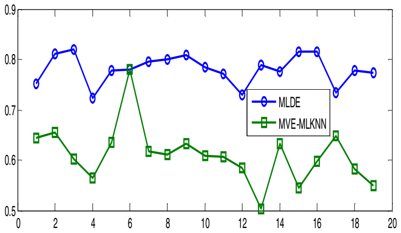

Fig. 16. Subset accuracy of tested approach in Reuters-Oh stream

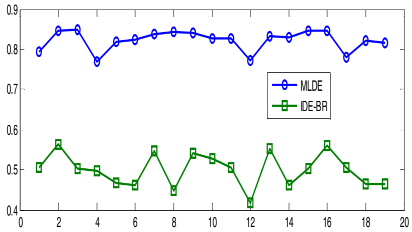

Fig. 17. Example-based F measure of tested approach in Reuters-Oh

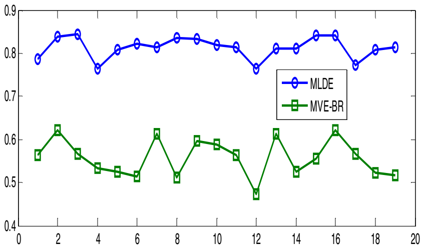

Fig. 18. Micro-average F measure of tested approach in Reuters-Gradual

VIII. Conclusion

In this paper, a new ensemble approach, MLDE, is proposed to deal with the multi-label stream classification with concept drift. As an ensemble approach, we select the useful MLCCs by the adaptive ensemble method and combine them by voting method to achieve the better global prediction result. Experiments on both synthetic and real-world streams are carried out to evaluate the performances of MLDE, IDE, MVE, MLOzaBagAdwin and MajorityLabelset based on four evaluation measures. The experimental results demonstrate that our MLDE perform better than other algorithms.

In the future work, we plan to further extend our work in several aspects. First, we will investigate how to design an ensemble model in a noisy stream environment. Second, we plan to further extend our approach to semi-supervised multi-label stream classification. All of these will be of great importance in applying the proposed MLDE to more real-world data streams.

Acknowledgment

This research was supported in part by NSFC under Grant No. 61300209, 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] Wang, Peng, Peng Zhang, and Li Guo. "Mining Multi-Label Data Streams Using Ensemble-Based Active Learning." SDM. 2012.

[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 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 classifier for text classification.” Expert Systems with Applications, 2011, vol. 38(10), 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] Albert Bifet, Geoffrey Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen, and Thomas Seidl. “Moa: Massive online analysis, a framework for stream classification and clustering.” Journal of Machine Learning Research (JMLR) Workshop and Conference Proceedings, 2010.

[9] Nikunj C. Oza., “Online ensemble learning”. AAAI/IAAI, 2000, pp.1109.

[10] Leo Breiman., “Pasting small votes for classification in large databases and on-line”. Machine Learning, vol.36(1-2) , 1999, pp.85-103.

[11] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan., “Mining data streams under block evolution”. ACM SIGKDD Explorations Newsletter, 2002, vol.3(2), pp.1-10.

[12] Baena-Garcia M, Campo-Avila J D, Fidalgo R, Bifet A, Gavalda R, and Morales-Bueno R., “Early drift detection method”. Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams, Berlin, 2006, Germany, pp.77-86.

[13] Nikunj C. Oza., “Online ensemble learning”. PhD thesis, The University of California, Berkeley, CA, 2001.

[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] Dariusz Brzezinski., “Mining data streams with concept drift”. Masters thesis, Poznan University of Technology, Poznan, 2010.

[16] Dariusz Brzezinski, and Jerzy Stefanowski., “Accuracy updated ensemble for data streams with concept drift”. Proceedings of the 6th international conference on Hybrid artificial intelligent systems (HAIS’11), Springer 2011, vol.6679, pp.155-163.

[17] Kong X, Yu P S. “An ensemble-based approach to fast classification of multi-label data streams” Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on. IEEE, 2011, pp. 95-104.

[18] J. Zico Kolter, and Marcus A. Maloof., “Dynamic weighted majority: an ensemble method for drifting concepts”. Journal of Machine Learning Research, 2007, vol.8, pp. 2755-2790.

[19] Read J, Bifet A, Holmes G, et al. “Scalable and efficient multi-label classification for evolving data streams”. Machine learning, 2012, vol.88(1-2), pp. 243-272.

[20] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Mining multi-label data." Data mining and knowledge discovery handbook, Springer US, 2010, pp. 667-685.