Co-training based on Semi-supervised Ensemble Classification Approach for Multi-label Data Stream

Zhe Chu[1,2]
[1]*Key Laboratory of Knowledge Engineering with Big Data*
(Hefei University of Technology), Ministry of Education;
[2]*School of Computer Science and Information Engineering*
Hefei University of Technology, AnHui, 230601, China
zhechu@mail.hfut.edu.cn

Peipei Li[1,2]
[1]*Key Laboratory of Knowledge Engineering with Big Data*
(Hefei University of Technology), Ministry of Education;
[2]*School of Computer Science and Information Engineering*
Hefei University of Technology, AnHui, 230601, China
peipeili@hfut.edu.cn

Xuegang Hu[1,2]
[1]*Key Laboratory of Knowledge Engineering with Big Data*
(Hefei University of Technology), Ministry of Education;
[2]*School of Computer Science and Information Engineering*
Hefei University of Technology, AnHui, 230601, China
jsjxhuxg@hfut.edu.cn

***Abstract*—A large amount of data streams in the form of texts and images has been emerging in many real-world applications. These data streams often present the characteristics such as multi-labels, label missing and new class emerging, which makes the existing data stream classification algorithm face the challenges in precision space and time performance. This is because, on the one hand, it is known that data stream classification algorithms are mostly trained on all labeled single-class data, while there are a large amount of unlabeled data and few labeled data due to it is difficult to obtain labels in the real world. On the other hand, many of existing multi-label data stream classification algorithms mostly focused on the classification with all labeled data and without emerging new classes, and there are few semi-supervised methods. Therefore, this paper proposes a semi-supervised ensemble classification algorithm for multi-label data streams based on co-training. Firstly, the algorithm uses the sliding window mechanism to partition the data stream into data chunks. On the former *w* data chucks, the multi-label semi-supervised classification algorithm COINS based on co-training is used to training a base classifier on each chunk, and then an ensemble model with *w* COINS classifiers is generated ensemble model to adapt to the environment of data stream with a large number of unlabeled data. Meanwhile, a new class emerging detection mechanism is introduced, and the *w*+1 data chunk is predicted by the ensemble model to detect whether there is a new class emerging. When a new label is detected, the classifier is retrained on the current data chunk, and the ensemble model is updated. Finally, experimental results on five real data sets show that: as compared with the classical algorithms, the proposed approach can improve the classification accuracy of multi-label data streams with a large number of missing labels and new labels emerging.**

***Index Terms*—Data Stream Classification, Semi-supervised, Multi-label, New Label Emerging, Ensemble Model**

I. Introduction

With the rapid development and popularization of network technology, the fields of network traffic monitoring and credit card fraud detection have generated massive and rapid data. These data present the characteristics as being continuous, high-volume, high-speed and dynamically changing, called data streams. A large amount of potentially valuable information is hidden in these data streams. However, because they are massive, containing multiple labels and a lot of unlabeled data, especially the labels will be emerged as the data distribution dynamically changes (called new class emerging [24],[25]). It is hence challenging for existing data stream classification algorithms in the performance of accuracy and time-space consumption.

As we know, existing data stream classification algorithms mostly use then ensemble mechanism [12], which have the advantage that base classifiers constituting the ensemble model can be effectively updated and easily adapt to the rapid changes of data stream. However, most ensemble methods assume that the training data are fully labeled. In real-world applications, it is expensive and time-consuming to obtain fully labeled samples, and it is a practical method to use a large number of unlabeled and a small number of labeled samples (called incompletely labeled data) to train. The semi-supervised technology is a popular method in the handling of incompletely labeled data. Commonly used semi-supervised techniques include Co-training [1], Tri-training [5]. Existing data stream classification algorithms are mainly oriented to a single-class data stream, but data streams in real-world applications such as the text and image data streams often contain multiple labels. For example, a user may be interested

978-1-7281-4607-2/19/$31.00 ©2019 IEEE
DOI 10.1109/ICBK.2019.00016

in articles in specific cross-cutting areas. Thus, how to predict whether a user is interested in an article with multiple categories, which is a multi-label data stream classification problem.

At present, there are several representative works of the semi-supervised classification for multi-label data below. Zhan and Zhang et al. proposed inductive semi-supervised multi-label learning with Co-Training, called COINS[2]. In each co-training round, the original feature space is automatically dichotomized by maximizing the diversity between the two classifiers induced on either dichotomized feature subset. Thereafter, pairwise ranking predictions of unlabeled data are communicated between the two classifiers for model refinement. This method has a higher classification accuracy on multi-label data with incompletely labeled data. However, this method is a batch algorithm, which is difficult to adapt to the data stream environment.

In light of the above analysis, this paper proposes a semi-supervised ensemble classification algorithm based on co-training for multi-label data streams. Aiming at the problem of label missing in multi-label data stream, the algorithm firstly uses the sliding window mechanism to partition the data stream into many data chunks, a base classifier based on the COINS algorithm is constructed respectively on the first w data chunks, and then an ensemble model is generated by w basic classifiers. Secondly, because the label will be emerging with the change of data distributions, a new class emerging detection mechanism is introduced to solve this problem of class label missing in massive and fast data streams. That is, the w+1th data chunk is predicted by using the ensemble model to detect whether there is the new label emerging in it. If a new label is detected, the classifier is retrained on the current data chunk to update the ensemble model. Finally, experimental results conducted on syntactic multi-label data streams show that as compared with the benchmark algorithm, the proposed algorithm can improve the classification accuracy in the handling of multi-label data streams with incompletely labeled data.

The rest of this article is organized as follows. Section II briefly reviews the related work of the semi-supervised classification of multi-label data, the semi-supervised classification of data streams, and the multi-label data stream classification. Section III introduces the algorithm in this paper. Section IV gives the experimental results and analysis. Finally, Section V is the conclusion.

II. Related Work

This section provides a brief overview of semi-supervised classification for multi-label data, semi-supervised classification for data streams, and related work on multi-label data stream classification.

A. Semi-Supervised Classification Method for Multi-Label Data

Multi-label learning deals with the problem where each example is represented by a single instance (feature vector) while associated with multi-labels simultaneously [3][4]. Correspondingly, the task is to learn a multi-label predictor which maps from the input space of instances to the output space of label sets. In practical applications, the process of obtaining training sample labels is often very demanding and time-consuming, especially for multi-label data that should annotate multiple class labels. Therefore, one measure is to consider semi-supervised multi-label learning.

The semi-supervised classification algorithm uses a large number of unlabeled samples and labeled samples to be trained to enhance the classification effect. The commonly used semi-supervised technique has the Co-training [1] paradigm. However, it is difficult to meet the requirements of the Co-training data set with two sufficient and redundant views. Tri-training [5] is an extension of the Co-training paradigm. It overcomes the shortcomings of the former. Three classifiers are built, and only one learning algorithm can be easily obtained by training on different training sets.

In addition, graph-based semi-supervised techniques are used to construct affinity matrices on labeled and unlabeled data. For instance, a semi-supervised multi-label classification method (SMILE) is proposed in [6]. First the method estimates label correlation from partially labeled instances and replenishes missing labels of these instances. Then, it takes advantage of labeled and unlabeled instances to construct a neighborhood graph. Next, the known labels along with unlabeled instances are exploited to train a graph based semi-supervised linear classifier.

The work in [7] proposed a multi-label classification algorithm based on semi-supervised learning (SSML-kNN). Firstly, a semi-supervised self-training model and correlation-based multi-label k-nearest neighbor classification are proposed for classifying unlabeled data sets. After that, the training intermediate result with a high confidence is selected and added to the training data set. At the same time, the training model continually loops to extend the label data set. Finally, the test set is classified by the training model.

The work in [8] proposed a new semi-supervised learning framework based on nuclear specifications for multi-label classification. An algorithm called ALSM is proposed in [9], which recovers the intermediate feature space of learning and unlabeled training samples through a low rank matrix, and uses an adaptive semi-supervised learning strategy to train multi-label classifiers. This method can be applied in generalizing semi-supervised multi-label classification problems.

B. Semi-Supervised Classification for Data Streams

There are two main types of data stream classification strategies: single and ensemble classifiers. Works in [10],[11] involves a single classifier, which is directly generated an initial model by learning given training examples. A single classifier is usually complicated in structure, and the model update operation is cumbersome, so the update speed is low and the accuracy is poor. In order to solve the above problems, scholars have proposed the integrated classifier.

The work in [20] proposed an integrated algorithm (SPASC) for classifying instances of non-stationary data streams in a semi-supervised environment. Furthermore, the method is intended to identify the repeated concept drift in the data stream.

The work in [12] proposed a semi-supervised classifier ensemble method for learning time-varying data streams. This algorithm maintains all the desirable properties of the semi-supervised Co-trained random FOREST algorithm (Co-Forest) and extends it into evolving data streams. In addition, the Adaptive Windowing (ADWIN2) is introduced to deal with concept drifts, which makes it adapt to the data stream environment. The work in [21] proposed a new data stream classification algorithm (DISSFCM), which overcomes the assumption of fully labeled data in a semi-supervised manner.

C. Multi-Label Data Stream Classification Method

A multi-label stream classification method was proposed in [13], and it deals with concept drift and class imbalance by creating two fixed-size windows, one for positive examples and one for negative cases. A binary correlation transform and a KNN algorithm are used as basic classifiers.

Another way to classify multi-label data streams is to use several random trees to maintain the labels that appear in the data stream. The correlation algorithm (SMART) [14], which deals with the problem of concept drift, and is capable of simulating the correlation between labels and their joint sparsity.

The MOA extension described in [15] can also be used to classify multi-label data streams. Authors compared different types of existing transformations and proposed several related improvements. These methods include training sets, pairing classifications and rankings, and thresholds. The literature [16] introduces an improvement to the multi-label Hoeffding tree, which uses a hierarchical increment technique. Authors proposed an additional filtering method to improve the performance of the model by choosing the instances that only contain the most frequent combination of labels to train a model. The work in [22] developed a Bayesian-based method for learning from multi-label data streams by taking into consideration the correlation between pairs of labels and the relationship between label and feature. To handle the concept drift, it proposes a decay mechanism focusing on the age of the arrived samples to incrementally adapt to the change of data.

In sum, semi-supervised classification methods for multi-label data mentioned above present a good classification effect on incompletely labeled multi-label data. However, due to the batch processing mechanism and face the challenges in the time and space consumption, these methods are difficult to adapt to the massive data stream environment. And aforementioned semi-supervised classification method for data streams mainly consider single-class data, and do not involve multi-label and new-class emerging in data streams. While the aforementioned multi-label data stream classification methods consider the multi-label problem of data stream, but ignore the issues of class label missing and new class emerging. Therefore, contrary to the above methods, this paper proposes a semi-supervised ensemble classification algorithm for multi-label data streams based on co-training. The purpose is to solve the problems of class label missing and new class emerging for a higher accuracy in the multi-label data stream classification.

III. THE PROPOSED METHOD

In this section, a multi-label data stream ensemble classification method based on co-training will be proposed. It is based on COINS algorithm to build an ensemble model and introduces a new class emerging detection method. Combining the semi-supervised technology and the ensemble model, the proposed method aims to deal with the problem of class label missing and new label emerging in multi-label data stream classification.

We first give the problem formalization of our method below: An incompletely labeled data stream *S* is divided into n data chunks, denoted as $S = \{D_1, D_2, ..., D_n\}$ where n represents the number of basic classifiers. Each data chunk D_i contains the labeled data $\mathcal{L} = \{(x_1, y_1), (x_2, y_2), ..., (x_L, y_L)\}$, and the unlabeled data $\mathcal{U} = \{x_{L+1}, x_{L+2}, x_{L+U}\}$, namely $D_i = \{\mathcal{L},\mathcal{U}\}$. Where χ denotes a feature vector of d-dimension, and $\mathcal{Y} = \{\lambda_1, \lambda_2, ..., \lambda_q\}$ denotes a label space composed of q class labels, $x_i \in \chi$ is an instance of a d-dimensional feature vector, and $y_i \in \{+1, -1\}^q$ is a label vector corresponding to the q-dimensional of $\mathcal{Y}$. The purpose of learning for incompletely labeled multi-label data is to learn a multi-label classification model $h : \chi \rightarrow 2^{\mathrm{y}}$ from the training examples. For an unknown instance $x \in \chi$, there is $h(x) \subseteq \mathcal{Y}$. Since each data chunk contains labeled data and unlabeled data, the sampling rate α is set in this paper. For details, please refer to Section IV.C.

Figure 1 shows the framework of our proposed method. This method is built to handle incompletely labeled data based on co-training strategy. Let the data stream S contain labeled data and unlabeled data. The proposed method adopts a fixed window mechanism to divide the data stream *S* into a series of data chunks. The first *w* data chunks $D_1, D_2, ..., D_w$ are trained based on the COINS algorithm to obtain *w* base classifiers $T_1, T_2, ..., T_w$, then they are used to form an initial ensemble classification model. As each data chunk continually arrives, new class emerging may be implied, and the earliest established base classifiers are not well adapted to the latest incoming data. Therefore, the built-in ensemble model is used to perform new label emerging detection on the new data chunk, and the ensemble model is updated according to the detection result.

Main techniques involve the basic classifier model based on the COINS algorithm, the new class emerge detection mechanism and the ensemble model update. More technical details will be given in Sections III.A-III.C respectively. The *w* base classifiers are trained on the first *w* data chunks to predict the *w*+1 *th* data chunk. The set of known labels on the *i-th* data chunk is represented by $v_i = 1, 2, ..., l$ $(i \in 1, 2, ..., w)$,v_t is the set of labels on the prediction data chunk. When the new label detection value on the *t-th* data chunk is higher than the

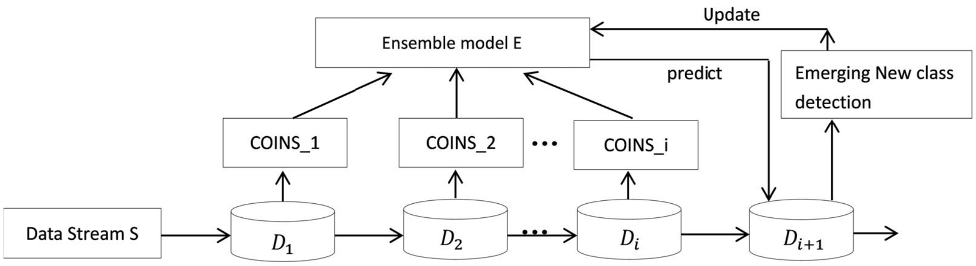


Fig. 1. Our multi-label data stream ensemble classification framework

threshold, it indicates that a new label is detected, and the label set will be enlarged with a new label $l^{'} = l+1 : v_t = \{v_i \cup l^{'}\}$. Otherwise the label set does not change:$v_t = v_{t-1}$. Whether to retrain the classifier on each data chunk instead of the original classifier depends on the new label is detected or not. Integrate the classification model based on the above structure.

A. Base Classifier Generation based on COINS

Zhan and Zhang et al overcame the shortcomings of traditional co-training, and proposed the COINS algorithm [2], which is the inductive semi-supervised multi-label learning with co-training. Specific steps are as follows:

For the incoming *i-th* data chunk $D_i = \{\mathcal{L}, \mathcal{U}\}$ in the data stream, the goal is to learn a classification model on each data chunk: $\mathcal{W} = \{w_k | 1 \leq k \leq q\}$,where $w_k = [w_{k1}, w_{k2}, ..., w_{kd}]^T$ represents the weight vector of the k-th label λ_k. The two index sets $I^1 = I^2 = \{(i,k,l) | 1 \leq i \leq L, y_{ik} = +1, y_{il} = -1\}$ are initialized by the labeled data set L in D_i. The ranking loss is written in Eq.1:

$$\mathcal{RL}(W,I) = \frac{1}{|I|} \sum_{(i,k,l) \in I} \max(0, 1- < w_k - w_l, x_i >) \quad (1)$$

It is used for classifier induction as shown in Eq.2:

$$\mathcal{V}(W,I) = \frac{1}{2} \sum_{k=1}^{q} \|w_k\|^2 + C \cdot \mathcal{RL}(W,I) \quad (2)$$

where C represents the balancing parameters. And the objective function is obtained below:

$$\min_{w^1, w^2} \log(e^{V(W^1, I^1)} + e^{V(W^2, I^2)}) \quad (3)$$

Wherein, W^1 and W^2 represent classification models constructed on the two index sets I^1 and I^2, respectively. Considering that the difference between the two models W^1 and W^2 must be large, so the following constraints on both models were added to enable effective supervision information communication between models for optimization as shown in Eq.4:

$$\sum_{a=1}^{d} (w_{ka}^1)^2 \cdot (w_{ka}^2)^2 = 0 (\forall 1 \leq k \leq q) \quad (4)$$

The above constraint dichotomy the feature space and a multi-label prediction model was generated by the view splitting strategy. COINS adapts the popular ϵ-expansion property of co-training to characterize the diversity between models W^1 and W^2. And choose the prediction difference of the model on a pair of class labels to represent the confidence of the multi-label classification. The confidence is used as shown in Eq. 5:

$$\mathcal{C}_W^{k,l}(X) = \frac{1}{1 + \exp(-\gamma(p_W^{k,l}(x) - \tau))} \quad (5)$$

We can get Eq. 6 by replacing the probability function in the original -expansion attribute:

$$\Pr(S^1 \bigoplus S^2) \geq \epsilon \cdot \min[Pr(S^1 \bigwedge S^2), Pr(\overline{S^1} \bigwedge \overline{S^2}) \quad (6)$$

Applying the α-expansion attribute on the unlabeled data set U in the data chunk D_i is written in Eq.7:

$$\begin{aligned} &\sum_{x \in U} [C_{W^1}^{k,l} \cdot C_{W^2}^{-k,l} + C_{W^1}^{-k,l} \cdot C_{W^2}^{k,l}] \geq \\ &\epsilon \cdot \min[\sum_{x \in U} C_{W^1}^{k,l} \cdot C_{W^2}^{k,l}, \sum_{x \in U} C_{W^1}^{-k,l} \cdot C_{W^2}^{-k,l}] \end{aligned} \quad (7)$$

the classification models W^1 and W^2 are generated by optimizing Eqs. (3), (4), and (7).

Model optimization is performed between the classification models by using pairwise ranking predictions as the supervised information communicated. Specifically, for each unlabeled data $x_j (x_j \in U, j \in \mathcal{J}), \mathcal{J} \subseteq \{L+1, ..., L+U\}$ in the data chunk D_i. The empirical ranking loss of W^1 on x_j is calculated according to the index set I_j^1, that is, $RL(W^1, I_j^1)$.

The ranking index set Δ^1 is formed by identifying n unlabeled data which have least $RL(W^1, I_j^1)$. For each of the identified unlabeled data, one of the elements from I_j^1 is randomly picked up and added to Δ^1. Thereafter, the supervision information conveyed by Δ^1 is communicated to W^2 for model update ($I^2 = I^2 + \Delta^1$). Equivalently, the supervision information of the update W^1 is obtained ($I^1 = I^1 + \Delta^2$). The supervision information between the two classification models W^1 and W^2 is delivered by updating the ranking index set and the unlabeled data U. The co-training classifier is obtained as a base classifier by the above method on each data chunk.

B. New Label Emerging Detection

The new label data may be generated as an incoming data chunk arrives. New labels may appear in the form of a set of feature values or a label having not been seen before or both of them. Therefore, both the feature and label spaces should be taken into account.

In this subsection, to detect new label emerging in the newly arrived data chunk, we introduce a new label emerging detection technique called MuENLForest [23] which considers the feature difference and label relationship simultaneously. MuENLForest consists of 100 MuENLTrees, each MuENLTree is built using a random subset of the latest incoming data chunk of size φ=64. The definition of MuENLTree is given as follows:

MuENLTree is a binary tree consisted of internal nodes and leaf nodes. For the latest incoming data chunk *w*+1, the first *w* basic classifiers make predictions on it respectively. For a new instance *x*, the prediction yields $h_t(x)$ (t represents the *t-th* base classifier). Let a=[*x*,$h_t(x)$] denote a training sample with the predictive value as the input of root node in each MuENLTree. In each internal node of a MuENLTree, the training instance node is divided into two children nodes by the following division rule: $\|a^q - p_1\| \leq \|a^q - p_2\|$, where $\mathbf{p}_1$ and $\mathbf{p}_2$ are two cluster centers having **q** attributes, $\mathbf{a}^q$ is the q projection of **a**. Each leaf node defines a ball covering *S*(i.e., the set of all training instances falling into this leaf node) having radius $r = \max_{x \in S} \|a - m\|$, where $m = mean(S)$. To grow a MuENLTree during the training process, the training set is recursively partitioned into internal nodes until any one of the following conditions (C) is satisfied: (a) the tree reaches a height limit $\mathbf{e}_m$; (b) $|\mathbf{S}| = 1$; (c) all instances in S have the same $\mathbf{x}^q$ value.

Once MuENLForest, i.e.,$D_t(\cdot)$, is constructed, it is ready for prediction. When evaluating each instance *x* in the latest data chunk, $D_t(x) = 1$, i.e., if *x* falls outside the ball, there is a new label. Otherwise,$D_t(x) = -1$, *x* has no new label. The final output of MuENLForest is determined by the majority voting of all MuENLTrees. Based on the voting result, it is judged whether there is a new label emerging on the new data. In terms of the above idea, the new label emerging detection is performed on the *w*+1 *th* data chunk using the first *w* base classifiers. According to the detection result, the ensemble model is optimized and updated.

C. Ensemble Model Update

As the data chunk continuously arrives, the arrived data may contain multi-labels and unlabeled data, and the labels are probably emerging as the data distribution dynamically changes. The classifier trained on the previous data chunk is not well adapted to the subsequent incoming data chunk, so the previous classifier needs to be updated on the new data chunk for better adapting to the new data chunk.

Details of our model update strategy are below: Predict the latest incoming data chunk D_{i+1} by the former *w* classifiers $T_1, T_2, ..., T_w$. The MuENLForest is used to determine whether a new label is emerging or not. If the new label detection value on the $r(r \in 1, 2, ..., w)$ classifier T_r, is higher than the threshold *th*, it indicates that a new label appears, and T_r is retrained on the latest data chunk and replaces the original classifier with the newly obtained classifier. And then we detect whether the remaining base classifier has a new label emerging compared with the current data chunk. If it is above the threshold again, the original classifier is retrained again, and looping until the last classifier is updated.

D. Our Algorithm

Algorithm 1 shows the pseudo code of our multi-label data stream ensemble classification algorithm based on COINS : *w* represents the number of base classifiers; readChunk(*S*,*m*) implements the window mechanism, reads the data chunk of size *m* from *S*; *out* represents the new label emerging detection output value, MuENLForest(*Trn*,*Tstlb*, T_j) calculates the new label detection of the data chunk by the classifier T_j; *Trn* stores the unlabeled data and *Tstlb* stores the prediction label of the data set D_{i+1}; *th* represents a threshold in the new label emerging detection.

The main processing of our algorithm is as follows: On the premise of size of the number of base classifiers *w*. Constructing an ensemble *E* (Step 2-6) using the COINS algorithm on the first incoming data chunks respectively; if the number of base classifiers exceeds and then read the next data chunk D_{i+1}(Step 8). Using the section III.B new label emerging detection mechanism to verify the ensemble *E* on D_{i+1} and calculate a new label emerge detection value of a classifier on the data chunk, and compare the obtained value with the set threshold(Step 10).

Update the classifier according to the value, if it is greater than the threshold, retrain the classifier on the new data chunk, replace the original classifier with the new generated classifier(Step 11-12). Test the remaining classifiers to detect the new labels on the newly data chunk, and so on, until all the classifiers have been updated to obtain the final ensemble classifier.

IV. EXPERIMENT

A. Data Set

To evaluate the performance of the proposed method, five benchmark multi-label data sets were used for experimental studies. Given a multi-label data set $D = \{(x_i, y_i) | 1 \leq i \leq |D|\}$, use $|D|$, dim(D), CL(D) to represent the number of instances, features, and class labels, respectively. Table I summarizes the detailed features of the data set[1].

TABLE I
BENCHMARK MULTI-LABEL DATASET

Date Set	***|D|***	***dim(D)***	**CL(D)**	**Domain**
enron	1702	1001	16	text
image	2000	294	5	images
scene	2407	294	6	images
yeast	2417	103	14	biology
slashdot	3782	1079	14	text

[1] 1.These data sets are publicly available at http: // mulan . sourceforge . net / datasets and http: // meka . sourceforge . net / datasets.

Algorithm 1 Our COINS-based Multi-Label Data Stream Ensemble classification algorithm(CMLDSE)

```
Input: Data stream $S$,data chunk size $m$
Output: An ensemble model $E$
$E = NULL$;
for $1 \leq i \leq w$ do
    $D_i = readChunk(S, m)$;
    Using the COINS algorithm trained on $D_i$ to obtain the base classifier $T_i$;
    $E = E + T_i$;
end for
while $S \neq NULL$ do
    $D_{i+1} = readChunk(S, m)$;
    for each classifier $T_j$ in $E$ do
        $out = MuENLForest(Trn, Tstlb, T_j)$;
        if $out > th$ then
            Retrains on $D_{i+1}$ to get a base classifier $T_i^{'}$ instead of $T_i$;
            BREAK;
        end if
    end for
end while
```

B. Parameter Setting

In our experiments, all parameters are specified below. For integrated learning, the number of base classifier is not as good as possible, and too many classifiers lead to redundancy. The number of base classifiers of the algorithm is taken as *w*=6, and *th*=0.3. In order to ensure a sufficient number of data chunks, $m \leq |D|/12$.

C. Comparison Algorithm

The semi-supervised classification algorithm applied in multi-labeled data stream is compared with five algorithms, including the algorithm of updating the ensemble model by using the error rate based detection mechanism, the COINS algorithm itself, and a fully supervised multi-label classification algorithm, the classic ML-KNN algorithm, and an inductive semi-supervised multi-label learning algorithm:

- Our CMLDSE algorithm without New Label Emerging Detection (called CMLDSE-no-NLE): The sliding window mechanism is used to partition the data stream. On the former w block, the multi-label semi-supervised classification algorithm COINS training base classifier based on the cooperative training mechanism is used to construct the ensemble model. The traditional prediction error rate detection mechanism is introduced to update the base classifier and update the ensemble model.
- COINS [2]: COINS performs multi-label classification in the semi-supervised form of induction, overcoming the shortcomings of traditional co-training and maximizing the difference between the two classifiers. A pairwise ranking prediction of unlabeled data is communicated between the two classifiers to re-optimize the model, and a good experimental result is obtained.
- Ecc [17]: Ecc works in a fully supervised manner by transforming multi-label learning problems into a binary classification problem chain, where the predictions of the pre-classifier are used as additional features to learn the subsequent classifiers in the chain. For Ecc, integrated learning is used to take advantage of the randomness of the chain order and to exhibit highly competitive performance when learning multi-label data.
- ML-KNN [18]: The multi-label classification method based on kNN is one of the most advanced methods in off-line setting. To accommodate the stream classification problem, we used a method similar to the sliding window method, in which we train the ML-KNN classifier on the latest data chunk and use it to classify the next data chunk. Due to real-time constraints and concept drift in the stream environment, it is not feasible to use a large training set to train the ML-KNN model. In many data flow classification problems, traditional classifiers trained on the latest data perform better than using more historical data in the evolved data stream.
- iMLCU [19]: iMLCU works in a semi-supervised manner by applying large margin criteria to both labeled and unlabeled data, where the optimization problem is non-convex and allows for a iterative CCCP (convex-concave procedure) solution. For iMLCUs, empirical ranking loss and pseudo-hinge loss are used to instantiate objective terms on labeled and unlabeled data, respectively.

D. Evaluation Protocol

Since each data set in Table I is static data, in order to simulate the environment of the data stream, a sliding window mechanism is adopted, and when m piece of data is read, a data chunk is created. For each data chunk, we randomly sample the $\alpha \times 100\%$ example to form the marked data set $\mathcal{L}$. For the remaining examples, 40% of them were randomly sampled to form an unlabeled data set $\mathcal{U}$, 10% of them are random samples to form a test data set $\mathcal{T}$. In the following experiments, the comparison algorithm was trained and tested as follows:

- For the fully supervised learning algorithm Ecc, it is trained on the labeled data set $\mathcal{L}$ and evaluated on the test data set $\mathcal{T}$;
- For the inductive semi-supervised learning algorithms Coins and iMLCU, and CMLDSECMLDSE (the error rate threshold is taken as 0.4), they are trained on $\mathcal{L} \bigcup \mathcal{U}$ and evaluated on $\mathcal{T}$;
- For the ML-KNN algorithm (k value 2), some processing is done on the data set, it is trained on the labeled data set $\mathcal{L}$, and all remaining data is used as a test set ,on which the test evaluation is performed.

In this paper, we take the sampling rate α of the label data from 1% to 5% in steps of 1%, and perform 10 experiments for each data set at each sampling rate. Five widely used multi-label metrics were used for performance evaluation, including hamming loss, one-error, coverage, ranking loss, and average precision. The first four indicators, the smaller the metric

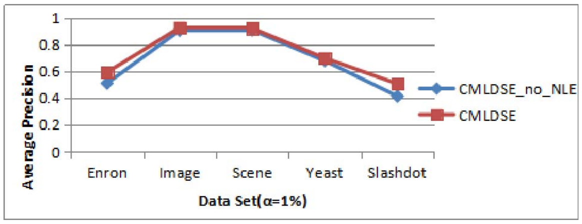


Fig. 2. Performance comparison between the introduction of a new class emerge detection algorithms and the use of error rate mechanism algorithms

values the better the performance. The last indicator, the larger the metric values the better the performance.

The experiments respectively report the detailed experimental results of each comparison algorithm for all data sets in Hamming loss, Ranking loss, One-error, Coverage, and Average Precision. (Due to space constraints, only experimental comparisons of Average Precision($\alpha = 1\%$) is shown on Table II in the present paper).

TABLE II
INDUCTIVE PERFORMANCE (AVERAGE) FOR EACH COMPARISON ALGORITHM IN TERMS OF AVERAGE PRECISION (BOLD IS OPTIMAL)

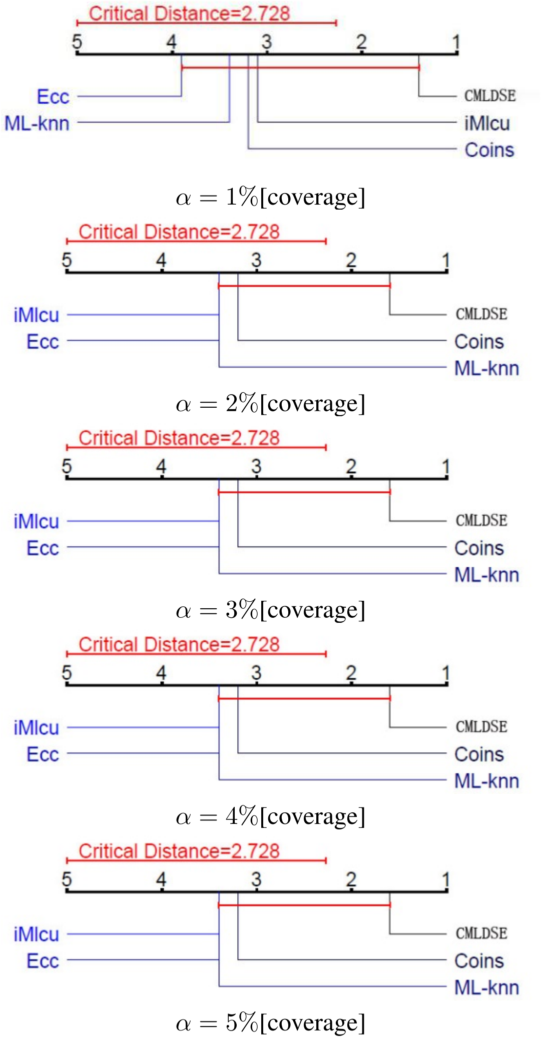
Comparing Algorithm	$\alpha = 1\%$				
	Enron	Image	Scene	Yeast	Slashdot
CMLDSE	0.591	**0.925**	**0.919**	**0.697**	**0.505**
CMLDSE-no-NLE	0.509	0.904	0.906	0.677	0.413
COINS	0.577	0.626	0.687	0.692	0.429
ML-KNN	0.600	0.345	0.334	0.689	0.439
Ecc	0.610	0.634	0.716	0.690	0.400
iMLCU	**0.627**	0.633	0.697	0.666	0.479

Fig.2 shows the difference between the algorithm for introducing a new class emerge detection method and the error rate detection mechanism algorithm for each evaluation criterion (due to the space, only the comparison chart when the marker rate $\alpha = 1\%$ of Average Precision is selected).

In addition, the Bonferroni-Dunn test was used to compare the experimental performance of all algorithms. The work of this paper is regarded as the control algorithm in statistical test. The average rank of all data sets for each algorithm is recorded, where the difference between this work and a comparison algorithm is calibrated with a critical difference (CD). Here, if their average rank differs by at least one CD (CD = 2.728 in this case; comparison algorithm k = 5, data set N=5), the performance difference is considered significant.

In order to visually display the performance difference between the comparison algorithms, Fig.3 shows a CD diagrams for each sampling rate of the label data, which is in terms of Coverage.

In each CD diagrams, the average rank of the comparison algorithm is labeled along the axis with the lower right rank. In addition, any comparison algorithm with an average rank within one CD to that of the proposed method is interconnected to each other with a thick lines. Otherwise, its performance is considered to be significantly different to the algorithm in this paper.



$\alpha = 1\%$[coverage]

$\alpha = 2\%$[coverage]

$\alpha = 3\%$[coverage]

$\alpha = 4\%$[coverage]

$\alpha = 5\%$[coverage]

Fig. 3. Comparison of the proposed algorithm (control algorithm) against other comparing algorithms with the Bonferroni-Dunn test. Algorithm not connected with the algorithm proposed in this paper in the CD diagram are considered to have significantly different performance from the control algorithm (CD = 2.728 at 0.05 significance level).

- The proposed algorithm outperforms other algorithms in terms of Hamming loss at all sampling rates of the data sets .
- After the sampling rate reaches 2%, in terms of Ranking loss, the proposed method is superior to other comparison algorithms in the former four data sets expect slashdot.
- In terms of Coverage and Average-precision, the proposed algorithm outperforms other algorithms on both image and scene data sets. The proposed method performs better than the classical algorithm ML-KNN and the difference between the other data sets and the optimal algorithm is also very small.
- The multi-label data stream semi-supervised integrated classification algorithm that introduces a new class emerge detection methods is superior to the traditional er-

ror rate detection mechanism algorithm in each detection standard.

In sum, the algorithm performs well on the latest competing algorithms across various data sets, the sampling rate and evaluation metrics of labeled data. The effectiveness of the algorithm on the semi-supervised multi-label data stream classification problem is well demonstrated.

V. Conclusion

In this paper, the sliding window mechanism is used and the ensemble model is built based on COINS algorithm. The semi-supervised technique is applied to the multi-label data stream classification problem. At the same time, a new label emerge detection algorithm is introduced and applied to data stream classification for model refinement. The accuracy of the classification algorithm is improved. Experimental results show that the algorithm can well solve the classification problem of multi-label data streams by semi-supervised method. However, in the data stream classification problem, a hot research problem involves the new class not being generated, but the concept drift caused by the data distribution changing. How to detect the concept drift in the case with missing labels to further improve the accuracy of the semi-supervised classification algorithm for multi-labeled data streams will be our next work.

VI. Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901, the Natural Science Foundation of China under grants (61976077,61673152,91746209).

References

[1] Mitchell B T. Combining labeled and unlabeled data with co-training[C]//11 th Annual Conference on Computational Learning Theory by ACM. Madison, WI, USA: ACM, 1998: 92-100.

[2] Zhan Wang and M. L.Zhang.Inductive Semi-supervised Multi-Label Learning with Co-Training.In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.(2017):1305-1314.

[3] E. Gibaja and S. Ventura. 2015. A tutorial on multi label learning. Comput. Surveys47, 3 (2015), Article 52.

[4] M.-L. Zhang and Z.-H. Zhou. 2014. A review on multi-label learning algorithms.IEEE Transactions on Knowledge and Data Engineering 26, 8 (2014), 1819-1837.

[5] Zhou Zhihua.Li Ming. Tri-training-exploiting unlabeled data using three classifiers[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1529-1541.

[6] Tan Q , Yu Y, Yu G , et al. Semi-supervised multi-label classification using incomplete label information[J]. Neuro computing, 2017: 192-202.

[7] Ding J., Song Q., Jia L., You J., Jiang Y.Multi-label K-Nearest Neighbor Classification Method Based on Semi-supervised. In: Deng K., Yu Z., Patnaik S., Wang J. (eds) Recent Developments in Mechatronics and Intelligent Robotics. ICMIR 2018. Advances in Intelligent Systems and Computing, vol 856. Springer, Cham.

[8] Liu Y,Nie F,Gao Q X.Nuclear-norm Based Semi-supervised Multiple Labels Learning[J]. Neuro computing, 2017: 940-947.

[9] Li S,Fu Y.Robust multi-label semi-supervised classification[C]//IEEE International Conference on Big Data. IEEE,2018.

[10] Domingos P, Huhen G. Mining Hish-Speed Data Streams[C]// 6th ACM SIGKDD International Conference on Knowledge Diseovery and Data Mining. Boston, USA, 2000: 71-80.

[11] Huhen G, Spencer L. Dominges P. Mining Time-Changing DataStream[C]// 7th ACM SIGKDD Intemafional Conference on Knowledge Discovery and Data Minillg.San Francisco, USA, 2001: 97-106.

[12] Wang, Yi, and Tao Li. "Improving semi-supervised co-forest algorithm in evolving data streams." Applied Intelligence(2018): 1-15.

[13] Xioufis, E.S., Spiliopoulou, M., Tsoumakas, G., Vlahavas, I.P.: Dealing with concept drift and class imbalance in multi-label stream classification. In: Walsh, T.(ed.) IJCAI, pp. 1583C1588. IJCAI/AAAI (2011).

[14] Kong, X., Yu, P.S.: An ensemble-based approach to fast classification of multi-label data streams. In:Georgakopoulos, D., Joshi, J.B.D. (eds.) 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom 2011,Orlando, FL, USA, pp. 95-104. ICST/IEEE, 15-18 October 2011.

[15] Read, J., Bifet, A.,Holmes, G.,Pfahringer, B.: Scalable and efficient multi-label classification for evolving data streams.Mach. Learn. 88(1-2), 243-272 (2012).

[16] Shi, Z., Xue, Y., Wen, Y., Cai, G.: Efficient class incremental learning for multi label classification of evolving data streams. In: 2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, pp. 2093-2099. IEEE, 6-11 July 2014.

[17] J. Read, B. Pfahringer, G. Holmes, and E. Frank. 2011. Classifier chains for multi label classification. Machine Learning 85,3 (2011), 333C359.

[18] M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. Pattern Recognition, 40(7):2038C2048, 2007.

[19] L. Wu and M.-L. Zhang. 2013. Multi-label classification with unlabeled data: An inductive approach. In Proceedings of 5th Asian Conference on Machine Learning. Canberra, Australia, 197C212.

[20] Hosseini M J , Gholipour A , Beigy H . An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams[J]. Knowledge and Information Systems, 2016, 46(3):567-597.

[21] Casalino, Gabriella, Giovanna Castellano, and Corrado Mencar. "Incremental adaptive semi-supervised fuzzy clustering for data stream classification." 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS). IEEE, 2018.

[22] NGUYEN, T.T., NGUYEN, T.T.T., LUONG, A.V., NGUYEN, Q.V.H., LIEW, A.W.-C. and STANTIC, B.2019. Multi-label classification via label correlation and first order feature dependance in a data stream. Pattern recognition [online], 90, pages 35-51.

[23] Zhu Y , Ting K M , Zhou Z H. Multi-label Learning with Emerging New Labels[C]// IEEE International Conference on Data Mining. IEEE, 2017.

[24] Mu X , Zhu F , Liu Y , et al. Social Stream Classification with Emerging New Labels[C]// Pacific-asia Conference on Knowledge Discovery & Data Mining. Springer, Cham, 2018.

[25] MU, Xin; ZHU, Feida; DU, Juan; Ee-peng LIM; and ZHOU, Zhi-Hua. Streaming classification with emerging new class by class matrix sketching. (2017). Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17): San Francisco, CA, February 4-9. 2373-2379.