Incremental Multi-Label Learning with Active Queries

Shen-Jun Huang[1,2], Guo-Xiang Li[1], Wen-Yu Huang[1] and Shao-Yuan Li[1,2,*]

[1] *Ministry of Industry and Information Technology Key Laboratory of Pattern Analysis and Machine Intelligence College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China*

[2] *Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University, Nanjing, 210023, China*

E-mail: huangsj@nuaa.edu.cn; guoxiangli@nuaa.edu.cn; huangwy@nuaa.edu.cn; lisy@nuaa.edu.cn

Received August 26, 2019 [Month Day, Year]; revised October 28, 2019 [Month Day, Year].

Abstract In multi-label learning, it is rather expensive to label instances since they are simultaneously associated with multiple labels. Therefore, active learning, which reduces the labeling cost by actively querying the labels of the most valuable data, becomes particularly important for multi-label learning. A good multi-label active learning algorithm usually consists of two crucial elements: a reasonable criterion to evaluate the gain of querying the label for an instance, and an effective classification model, based on whose prediction the criterion can be accurately computed. In this paper, we first introduce an effective multi-label classification model by combining label ranking with threshold learning, which is incrementally trained to avoid retraining from scratch after every query. Based on this model, we then propose to exploit both uncertainty and diversity in the instance space as well as the label space, and actively query the instance-label pairs which can improve the classification model most. Extensive experiments on 20 data sets demonstrate the superiority of the proposed approach to state-of-the-art methods.

Keywords active learning, multi-label learning, uncertainty, diversity

1 Introduction

In many applications, we have plenty of unlabeled data but few labeled data. As labeling is usually expensive since it requires the participation of human experts, training an accurate model with as few labeled data as possible becomes a challenge of great significance. Active learning, which reduces the labeling cost by actively selecting the most valuable data to query their labels, is a leading approach to this goal [1]. The key task in active learning is to design a selection criterion such that queried labels can improve the classification model most. During the past years, many active selection criteria have been proposed. For example, *uncertainty* measures the confidence of the current model on classifying an instance [2], *diversity* measures how different an instance is from the labeled data [3], *density* measures the representativeness of an instance to the whole data set [4], and so on. There are also some other approaches try to consider different criteria simultaneously [5, 6] and the aid of transferred knowledge from related tasks [7].

In traditional supervised classification problems, one instance is assumed to be associated with only one label. However, in many real world applications [8], an object can have multiple labels simultaneously. For example, a nature scene image may be tagged with

Regular Paper

This research was supported by the National Natural Science Foundation of China under Grant Nos. 61906089, the Aerospace Power Funds of China No. 6141B09050342 and the Fundamental Research Funds for the Central Universities of China, NO. NE2019104 and the Jiangsu Foundation No. BK20190408.

*Corresponding Author.

"trees", "mountain" and "sky", a web page introducing a city may be related to topics of "population", "economy" and "traffic". Multi-label learning is a framework dealing with such objects [9]. To label the multi-label examples, each of the multiple labels should be decided whether a proper one for an instance. Obviously, the labeling cost is even higher than that of single label learning, and thus active learning under the multi-label setting has attracted more and more attention.

Unlike in the single label setting, multi-label active learning methods have multiple choices on what to query at each iteration. Most existing methods query the whole label vector of one instance at a time. There are a few works trying to query the relevance of a instance-label pair, i.e., ask the oracle whether a specific label is relevant to the selected instance at each iteration [5, 10]. It has been shown that by utilizing external knowledge [11], such methods could be more effective because it can exploit the correlations among multiple labels. Especially for problems with many labels, experts may hardly identify all positive labels for an instance, but can easily decide whether a label is relevant to an instance or not. In this paper, we follow this setting and try to query if a label is positive on a specific instance at each iteration.

In multi-label active learning, the main efforts focus on exploiting uncertainty, leaving the other active selection criteria rarely considered. Different algorithms are designed to evaluate the uncertainty of unlabeled data. A commonness of them is that they usually evaluate the active selection criterion based on the predicted labels of instances. Thus an effective classifier which can accurately predict the labels for the unlabeled data is crucial for a successful active learning algorithm. Most existing methods decompose the multi-label task into a series of binary classification problems and learn each label independently. Such a strategy, however, ignores the correlations among different labels, which play an important role in multi-label learning; moreover, even with prediction values on each label, it is still a challenge to decide the threshold for separating positive and negative labels of an instance. As we know, under the single-label setting, given the outputs of the classifier, the positive label can be easily determined as the one with maximum prediction value. However, in multi-label learning, we do not know how many positive labels an instance should have. Also, the prediction values on different labels may not be comparable since the classifiers are independently trained. To address this challenge, some ad-hoc efforts have been attempted. For example, simply taking the sign of predictions as labels [12], normalizing the predictions on different labels [13], predicting the number of positive labels via an extra regression model [14], and so on.

In this paper, we propose to exploit both uncertainty and diversity in the instance space and label space with an incremental multi-label classification model. First, along with a label ranking algorithm which learns a subspace shared by all labels to exploit the label correlations, and optimizes the approximated ranking loss to rank positive labels before negative ones, we also introduce a dummy label for each instance, and train the model to rank the dummy label between positive and negative labels. Since the dummy label threshold is learned specifically for each instance along with the ranking model, it is expected to provide an accurate separation of positive and negative labels. Based on this model, we then integrate uncertainty with diversity as a new active learning criterion, and select the most valuable instance-label pairs to query. Specifically, in the instance space, we simultaneously evaluate the uncertainty with label cardinality inconsistency(LCI) [12] and the diversity with the number of labels not queried; while in the labels space, the distance between a la-

bel and the thresholding dummy label is employed to evaluate the uncertainty. Our multi-label model is incrementally updated based on only the newly added labeled data, avoiding the retraining from scratch. We performed extensive experiments on 20 data sets with regard to 2 performance measures. Results show that the proposed approach is superior to several state-of-the-art methods. We further studied the LCI criterion to show that it is important to incorporate other criteria at the beginning stage. Also, we disclosed that positive labels are more important for multi-label learning by studying the distribution of queried labels.

The rest of this paper is organized as follows. Section 2 reviews related work. In Section 3, the multi-label classification model as well as the active selection strategy are presented. Section 4 presents the experiments, followed by the conclusion in Section 5.

2 Related Work

The multi-label classification model used in this paper is extended from an online multi-class ranking model [15]. While similar techniques are used for optimizing the approximated ranking loss, the method in [15] is designed for single label setting, and thus can not decide how many labels should be selected as positive from the ranked label list. In contrast, by introducing the dummy label, our algorithm can automatically learn a threshold for separating positive and negative labels.

Fürnkranz et al. proposed a calibrated label ranking approach for multi-label classification in [16], where a mechanism similar to our dummy label is used for separating positive and negative labels. However, to the best of our knowledge, label ranking with threshold learning has not been exploited in multi-label active learning.

Multi-label active learning (MLAL), which is expected to reduce the expensive annotation cost in multi-label learning, has attracted more and more interests [17–21]. Most existing MLAL approaches are based on the binary relevance model, which decomposes the multi-label classification into a series of binary classification problems. For example, [22] trains a SVM for each label, and selects the instance which maximizes the reduction of expected loss to query its labels. [14] also trains a SVM for each label and aims to maximize the expected loss reduction, but uses an extra regression model to predict the number of labels for each instance, and proposes to approximate the model loss with the size of version space. Besides, based on independently trained binary classifiers, the minimal, average and weighted summarization over the uncertainties measured on each label are taken as active selection criterion in [23], [24] and [25], respectively. In [12], label cardinality inconsistency is combined with the separation margin via a tradeoff parameter as a new criterion for active selection, where the labels of instances are directly determined by the sign of prediction values of individual binary SVMs.

Query type has a great impact on labeling cost for MLAL [26]. Most MLAL methods try to query the whole label vector of one instance at a time [12]. However, as different labels are usually correlated in multi-label learning, this simple query type could lead to information redundancy and wasting of annotators' effort. There are some works querying instance-label pairs in MLAL [5, 27, 28]. This method is more effective because it is easier to identify one label for an instance compared to query the whole label vector. Besides, Huang *et al.* propose a novel MLAL framework to query the relevance ordering of label pairs, which gets richer information from each query and requires less expertise of the annotator [26].

There are also some other multi-label active learn-

ing approaches proposed for different settings. Zhang *et al.* propose the MLAL method for multi-view data [29]. Wang *et al.* propose an ensemble-based active learning framework to handle the multi-label stream data [30].Chen *et al.* consider querying the most likely positive subexample-label pairs instead of the example-label pair [21]. It is worth noting that the active selection criteria and classification technique proposed in the preliminary version [31] of this work have been extended to multi-instance multi-label setting [32, 33].

3 The Algorithm

We first introduce an incremental multi-label model by extending the label ranking model proposed in [15] from single-label to the multi-label setting in subsection 3.1, and then propose a new active learning strategy in subsection 3.2 to iteratively query whether a label is positive on an instance.

3.1 Multi-Label Ranking with Separating Dummy Label

In [15], an online algorithm was proposed to optimize approximated ranking loss on single label data, aiming to rank the positive label before negative ones for each instance. In single-label learning, we can easily determine the positive label for an instance by selecting the one with maximum prediction value. However, in multi-label learning where one instance can have more than one label, we do not know how many labels should be selected as positive from the label list ranked based on the predictions values. To overcome this difficulty, inspired by [16], we introduce a dummy label to each instance, and train the model to rank the dummy label between positive and negative labels for thresholding.

We denote by $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ the training data with n examples, where each instance $\boldsymbol{x}_i$ is a d-dimensional feature vector. Assuming there are in all K possible labels, the label vector of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \cdots, y_{iK})^\top$, where $y_{ik} = 1$ if instance $\boldsymbol{x}_i$ has the k-th label, otherwise $y_{ik} = -1$.

Binary relevance (BR) [34], which learns a classification model for each label, is the simplest way to handle multi-label data. Since BR approaches learn each label independently, correlations among multiple labels are ignored. In our approach, the classification model is decomposed into two levels. On the first level, a subspace shared by all the labels is learnt from the original feature space, and then on the second level, different label classifiers are trained based on the subspace. Since different labels will contribute to the weight update in shared space during the optimization, it is expected that correlated labels may help each other, i.e., label correlations are utilized. Formally, given an instance $\boldsymbol{x}$, we define the classification model on label k as

$$f_k(\boldsymbol{x}) = \mathbf{w}_k^\top W_0 \boldsymbol{x},$$

where W_0 is an $m \times d$ matrix which maps the original feature vectors to the shared subspace, and $\mathbf{w}_k$ is a m-dimensional vector as the classifier for label k. Here d and m are the dimensionalities of the feature space and the subspace, respectively. Note that m is usually set to a small value, and thus for high dimensional data, the two-level model used here can significantly reduce the memory cost when the numbers of variables are reduced from $d \times K$ to $m \times (d + K)$.

As in [26], the ranking loss for the instance $\boldsymbol{x}$ and one of its positive labels y can be defined as:

$$\epsilon(\boldsymbol{x}, y) = \sum_{i=1}^{R(\boldsymbol{x},y)} \frac{1}{i}, \tag{1}$$

where $R(\boldsymbol{x}, y)$ counts how many negative labels are ranked before label y, and can be formally defined as Eq(2).

$$R(\boldsymbol{x}, y) = \sum_{\bar{y} \in \bar{Y}} I[f_{\bar{y}}(\boldsymbol{x}) > f_y(\boldsymbol{x})], \tag{2}$$

where $\bar{Y}$ denotes the set of all negative labels of $\boldsymbol{x}$, and $I[\cdot]$ is the indicator function which equals 1 if the argument is true and 0 otherwise. Obviously, the lower y is ranked, the larger $R(\boldsymbol{x}, y)$ is, and accordingly the ranking error $\epsilon(\boldsymbol{x}, y)$ becomes larger. By minimizing the ranking error on all positive labels of all training instances, the model is expected to rank positive labels before negative ones for an unseen instance. With the convention $0/0 = 0$ if $R(X, y) = 0$, by combining Eq.1 and Eq.2, the ranking error $\epsilon(\boldsymbol{x}, y)$ can be written as:

$$\epsilon(\boldsymbol{x}, y) = \sum_{\bar{y} \in \bar{Y}} \epsilon(\boldsymbol{x}, y) \frac{I[f_{\bar{y}}(\boldsymbol{x}) > f_y(\boldsymbol{x})]}{R(\boldsymbol{x}, y)}. \quad (3)$$

It is difficult to minimize Eq.3 due to its non-convexity and discontinuousness. So we instead try to optimize its convex surrogate loss as follows:

$$\Psi(\boldsymbol{x}, y) = \sum_{\bar{y} \in \bar{Y}} \epsilon(\boldsymbol{x}, y) \frac{\max\{0, 1 + f_{\bar{y}}(\boldsymbol{x}) - f_y(\boldsymbol{x})\}}{R(\boldsymbol{x}, y)}. \quad (4)$$

Obviously, the surrogate loss $\Psi(\boldsymbol{x}, y)$ is an upper bound of $\epsilon(\boldsymbol{x}, y)$. Actually, hinge loss has been shown as an optimal choice among all convex surrogate losses [35]. Accordingly, we also redefine $R(\boldsymbol{x}, y)$ with a penalized margin 1, as following:

$$R(\boldsymbol{x}, y) = \sum_{\bar{y} \in \bar{Y}} I[f_{\bar{y}}(\boldsymbol{x}) > f_y(\boldsymbol{x}) - 1]. \quad (5)$$

Further checking the elements of Eq.4, we can observe that for a negative label $\bar{y}$, it will contribute nonzero loss to $\Psi(\boldsymbol{x}, y)$ only if it is ranked before the positive label y. We call such $\bar{y}$ the violated label since it violates the order that positive label should be ranked before negative label. Then each triplet $(\boldsymbol{x}, y, \bar{y})$ will induce a loss

$$\mathcal{L}(X, y, \bar{y}) = \epsilon(X, y) \max\{0, 1 + f_{\bar{y}}(X) - f_y(X)\}. \quad (6)$$

In the cases $R(\boldsymbol{x}, y) > 0$, by excluding the inviolated negative labels from $\bar{Y}$, the probability of randomly sampling a violated negative label $\bar{y}$ is $1/R(\boldsymbol{x}, y)$, and thus $\Psi(\boldsymbol{x}, y)$ can be viewed as the expectation of $\mathcal{L}(\boldsymbol{x}, y, \bar{y})$.

We minimize the ranking error with stochastic gradient descent (SGD). At the t-th iteration of SGD, assuming the current model parameters are W_0^t and w_k^t $(k = 1 \cdots K)$, we randomly sample an instance $\boldsymbol{x}$, one of its positive labels y, and one of its negative labels $\bar{y} \in \bar{Y}$ to form a triplet $(\boldsymbol{x}, y, \bar{y})$. If $\bar{y}$ is a violated label, then gradient descent is performed aiming to minimize $\mathcal{L}(X, y, \bar{y})$ according to:

$$W_0^{t+1} = W_0^t - \gamma_t \sum_{i=1}^{R(\boldsymbol{x}, y)} \frac{1}{i} (\mathbf{w}_{\bar{y}}^t \boldsymbol{x}^\top - \mathbf{w}_y^t \boldsymbol{x}^\top), \quad (7)$$

$$\mathbf{w}_y^{t+1} = \mathbf{w}_y^t + \gamma_t \sum_{i=1}^{R(\boldsymbol{x}, y)} \frac{1}{i} W_0^t \boldsymbol{x}, \quad (8)$$

$$\mathbf{w}_{\bar{y}}^{t+1} = \mathbf{w}_{\bar{y}}^t - \gamma_t \sum_{i=1}^{R(\boldsymbol{x}, y)} \frac{1}{i} W_0^t \boldsymbol{x}, \quad (9)$$

where γ_t is the step size. The updated parameters, i.e., $\mathbf{w}_y$, $\mathbf{w}_{\bar{y}}$ and each column of W_0 are then normalized to have a ℓ^2-norm smaller than a specific constant C.

From the above equations we can see, $R(\boldsymbol{x}, y)$ should be calculated in advance, which implies $f_y(\boldsymbol{x})$ should be compared with $f_{\bar{y}}(\boldsymbol{x})$ for each $\bar{y} \in \bar{Y}$. When the number of possible labels is large, this procedure can be quite time consuming. Therefore, we follow the idea in [15] to use an approximation to estimate $R(\boldsymbol{x}, y)$. Specifically, at each iteration of SGD, we randomly sample labels from $\bar{Y}$ one by one, until a violated label $\bar{y}$ occurs. If there are $R(\boldsymbol{x}, y)$ violated labels in $\bar{Y}$, we may need $k = |\bar{Y}|/R(\boldsymbol{x}, y)$ trails to get a violated label (every k labels contains a violated label on average). So assuming the first violated label is found at the v-th sampling step, $R(\boldsymbol{x}, y)$ can be approximated by $\lfloor |\bar{Y}|/v \rfloor$ [15].

We first summarize the training procedure in Algorithm 1, and then explain how the positive and negative labels are separated with the dummy label. We assume that every instances $\boldsymbol{x}_i$ has a dummy label, denoted by

y_{i0}. In line 7 of Algorithm 1, the sampled label y can be a positive label of $\boldsymbol{x}$ or the dummy label y_0. We construct the negative label set $\bar{Y}$ depending on the type of y. If y is the dummy label, then $\bar{Y}$ consists of all negative labels of instance $\boldsymbol{x}$, otherwise, $\bar{Y}$ contains all negative labels as well as the dummy label. Notice that there are only few positive labels for an instance, so y is very likely to be dummy label. After the training with such a mechanism, the dummy label will be ranked before all negative labels while positive labels will be ranked before both the dummy label and negative labels. So the dummy label provides a nature threshold to separate positive and negative labels. Given an unseen test instance, with the prediction values on each label, we can easily select the labels with larger predictions than that of the dummy label as positive labels.

Algorithm 1 The multi-label classification algorithm

1: **Input:**
2: training data, parameters m, C and γ_t
3: **Initialize:**
4: initialize W_0 and $\mathbf{w}_k$ $(k = 1 \cdots K)$ at random
5: **Repeat**:
6: randomly sample an instance $\boldsymbol{x}$
7: randomly select a positive or dummy label y of $\boldsymbol{x}$
8: $\bar{Y}$ = all negative labels of $\boldsymbol{x}$
9: **if** y is not the dummy label y_0
10: $\bar{Y} = \bar{Y} \cup \{y_0\}$
11: **end if**
12: **for** $i = 1 : |\bar{Y}|$
13: sample an negative label $\bar{y}$ from $\bar{Y}$
14: **if** $f_{\bar{y}}(X) > f_y(X) - 1$
15: $v = i$
16: update W_0, $\mathbf{w}_y$ and $\mathbf{w}_{\bar{y}}$ as Eqs. 7 to 9
17: normalize the updated parameters
18: break
19: **end if**
20: **end for**
21: **until** stop criterion reached

3.2 Active Selection

In this subsection, we present the strategy of active selection based on the previously introduce multi-label classification model. As stated before, we follow the setting in [10] to iteratively query if a label is positive on an instance. We denote by $\mathcal{D}$ the data set, and divide it into two parts: the labeled data $\mathcal{D}_l$ with N_l instances and unlabeled data $\mathcal{D}_u$ with N_u instances. Since we select instance-label pairs for querying, there will be some instances partially labeled. For convenience, such partially labeled instances are also taken as unlabeled data. In other words, D_l contains only the fully labeled instances, while $\mathcal{D}_u$ contains both the unlabeled and partially labeled instances. We also introduce $U(\boldsymbol{x})$ to denote the set of labels that have not been queried for instance $\boldsymbol{x}$. So the task in each iteration is to select an instance $\boldsymbol{x}^*$ from $\mathcal{D}_u$ and then select one label y^* from $U(\boldsymbol{x}^*)$ to query.

Uncertainty is an effective and mostly used criterion for active learning [12]. In this paper, we will try to exploit the uncertainty in both the instance space and label space, and combine it with diversity for active selection. In single label learning, a classic implementation of uncertainty sampling is to query the instance closest to the decision boundary. This strategy can be easily extend to multi-label learning. For example, the average or minimal margin over all labels can be taken to measure the uncertainty of an instance [24,25]. Recently, a simple uncertainty criterion, named as label cardinality inconsistency was proposed in [12]. It measures the inconsistency between the number of predicted positive labels of an instance and the average label cardinality on the fully labeled data, and can be formally defined as:

$$LCI(\boldsymbol{x}_i) = (\sum_{k=1}^{K} I[\hat{y}_{ik} > 0] - \frac{1}{N_l} \sum_{j=1}^{N_l} \sum_{k=1}^{K} I[y_{jk} > 0])^2,$$

where $\hat{y}_{ik}$ $(k = 1 \cdots K)$ is the predicted labels for instance $\boldsymbol{x}_i$, and $I[\cdot]$ is the indicator function. In [12], LCI was combined with margin with a tradeoff parameter to measure the uncertainty, and has been shown to be effective. However, with the increase of labeled

data, the difference of LCI over different instances may get smaller and smaller, and thus it becomes more difficult to identify the most uncertain instance based on the LCI criterion. In this paper, we extend LCI to incorporate with diversity, and define a new criterion as follows:

$$C_1(\boldsymbol{x}_i) = \frac{\left|\sum_{k=1}^{K} I[\hat{y}_{ik} > 0] - \frac{1}{N_l}\sum_{j=1}^{N_l}\sum_{k=1}^{K} I[y_{jk} > 0]\right|}{\max\{\xi, K - card(U(\boldsymbol{x}_i))\}}, \tag{10}$$

where $card(\cdot)$ counts the number of elements in a set, and thus $K - card(U(\boldsymbol{x}_i))$ is the number of queried labels of $\boldsymbol{x}_i$. $\xi \in (0, 1)$ is a constant to avoid the zero divisor. The motivation here is that instances with less queried labels may contain more unknown information and should be preferentially queried. As we want to query the instance with maximum uncertainty and diversity, the instance $\boldsymbol{x}^*$ with maximum C_1 value is selected. Note in our experiments, we set $\xi = 0.5$ and randomly select one instance if multiple instances achieved the maximal C_1 value.

After selecting the instance $\boldsymbol{x}^*$, we need to decide which label to query. Since the dummy label stands for the separating threshold of the positive and negative labels, the uncertainty of a label y can be naturally measured by the distance from it to the dummy label, which is formally defined as:

$$C_2(\boldsymbol{x}^*, y) = |f_y(\boldsymbol{x}^*) - f_{y_0}(\boldsymbol{x}^*)| \tag{11}$$

The pseudo code of the proposed algorithm, termed AUDI (Active learning based on Uncertainty and Diversity for Incremental multi-label learning), is presented in Algorithm 2. First, a subset of the data set $\mathcal{D}$ is randomly sampled to initialize the labeled data $\mathcal{D}_l$. Note in the experiments, to be fair, for all algorithms, $\mathcal{D}_l$ is initialized with the same set of fully labeled instances rather than instance-label pairs. After the initialization, we apply the Algorithm 1 introduced in the previous subsection on $\mathcal{D}_l$ to train a multi-label classification model. Then at each iteration of active learning, we select an instance-label pair $(\boldsymbol{x}^*, y^*)$ according to:

$$\boldsymbol{x}^* = \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{D}_u} C_1(\boldsymbol{x})$$
$$y^* = \operatorname*{argmin}_{y \in U(\boldsymbol{x}^*)} C_2(\boldsymbol{x}^*, y)$$

and query if y^* is a positive label of $\boldsymbol{x}^*$. After that, y^* is removed from $U(\boldsymbol{x}^*)$ and $\boldsymbol{x}^*$ is moved from $\mathcal{D}_u$ to $\mathcal{D}_l$ if it is fully labeled. Since our multi-label classification model can be trained incrementally, we do not need to retrain the model from scratch, but only update f based on the newly labeled data. Note that adding a label to an instance may affect the rank of all labels on it, so f is updated on $\boldsymbol{x}^*$ and all of its labels, instead of only on the pair $(\boldsymbol{x}^*, y^*)$. This active querying and model updating process is repeated until enough data labeled.

Algorithm 2 The AUDI algorithm

Input:
  data set $\mathcal{D}$
Initialize:
  Divide $\mathcal{D}$ to $\mathcal{D}_l$ and $\mathcal{D}_u$
  train a model f on $\mathcal{D}_l$
Repeat:
  get predictions and labels for instances in $\mathcal{D}_u$ with f
  compute $C_1(\boldsymbol{x})$ for all $\boldsymbol{x} \in D_u$ as Eq.10
  select the instance $\boldsymbol{x}^*$ with maximum C_1 value
  compute $C_2(\boldsymbol{x}^*, y)$ for all $y \in U(\boldsymbol{x}^*)$ as Eq.11
  select the label y^* with minimal C_2 value
  query if label y^* is a positive one for instance $\boldsymbol{x}^*$
  remove y^* from $U(\boldsymbol{x}^*)$
  if $|U(\boldsymbol{x}^*)| = 0$
    move $\boldsymbol{x}^*$ from $\mathcal{D}_u$ to $\mathcal{D}_l$
    update the model f with $\boldsymbol{x}^*$ and its labels
  end if
until the number of queries reached

4 Experiments

4.1 Settings

In the experiments, the following six multi-label active learning approaches are compared:

- Random: the baseline which randomly selects instance-label pairs.

- 2DAL: the two-dimensional active learning method proposed in [10], which selects instance-label pairs with the expected classification error reduction criterion.

- MML: the mean max loss strategy proposed in [22].

- MMC: the method proposed in [14], which uses the maximum loss reduction with maximal confidence as selection criterion.

- Adaptive: the adaptive method proposed in [12], which combines the max-margin prediction uncertainty and the label cardinality inconsistency as the criterion for active selection.

- AUDI: the method proposed in this paper.

Table 1. Statistics on Datasets used in the Experiments

Data	# instance	# label	# feature	cardinality
Corel5K	5000	374	499	3.52
Emotions	593	6	72	1.87
Enron	1702	53	1001	3.38
Genebase	662	27	1185	1.25
Image	2000	5	294	1.24
Medical	978	45	1449	1.25
Reuters	2000	7	243	1.15
Scene	2407	6	294	1.07
Yeast	2417	14	103	4.24
Arts	5000	26	462	1.64
Business	5000	30	438	1.59
Computers	5000	33	681	1.51
Education	5000	33	550	1.46
Entertainment	5000	21	640	1.42
Health	5000	32	612	1.66
Recreation	5000	22	606	1.42
Reference	5000	33	793	1.17
Science	5000	40	743	1.45
Social	5000	39	1047	1.28
Society	5000	27	636	1.69

Experiments are performed on 20 data sets, most of which can be download at the web page of MULAN project*. Corel5K [36] contains 5000 images with 374 possible labels, where each image is represented with a 499-dimensional feature vector. Emotions [37] consists of 593 songs, each of which is represented with a 72-dimensional feature vector. The task is to predict the music emotions of songs. Enron is a subset of the Enron email corpus [38], including about 1700 emails, where each email is represented as a 1001-dimensional feature vector. Genebase [39] is a set of 662 proteins for gene function classification. Image is a data set for natural scene image classification. It contains 2000 images, each of which is represented by a 294-dimensional [40]. Medical is a data set of clinical text for medical classification. Each instance is represented as a 1449-dimensional feature vector. Scene contains 2407 images with 6 possible labels: beach, sunset, fall foliage, field, mountain and urban. Reuters is a data set for text categorization. It is a processed version of [41] with the method introduced in [42]. Each document is represented as a 243-dimensional feature vector by aggregating all the instances in a bag. Yeast is a data set for predicting the gene functional classes of the Yeast Saccha-romyces cerevisiae, we use the version preprocessed by [43], which contains 2417 genes. Each gene is represented as a 203-dimensional feature vector. Yahoo consists of 11 independent data sets, i.e., Arts, Business, Computers, Education, Entertainment, Health, Recreation, Reference, Science, Social, and Society. They are collected from "yahoo.com" domain [44] for web page categorization. Each of the 11 data sets contains 5000 documents. And 20% to 45% of the documents have more than one class labels. Detailed characteristics of these data sets are summarized in Table 1, including number of instances, number of

*http://mulan.sourceforge.net/datasets.html

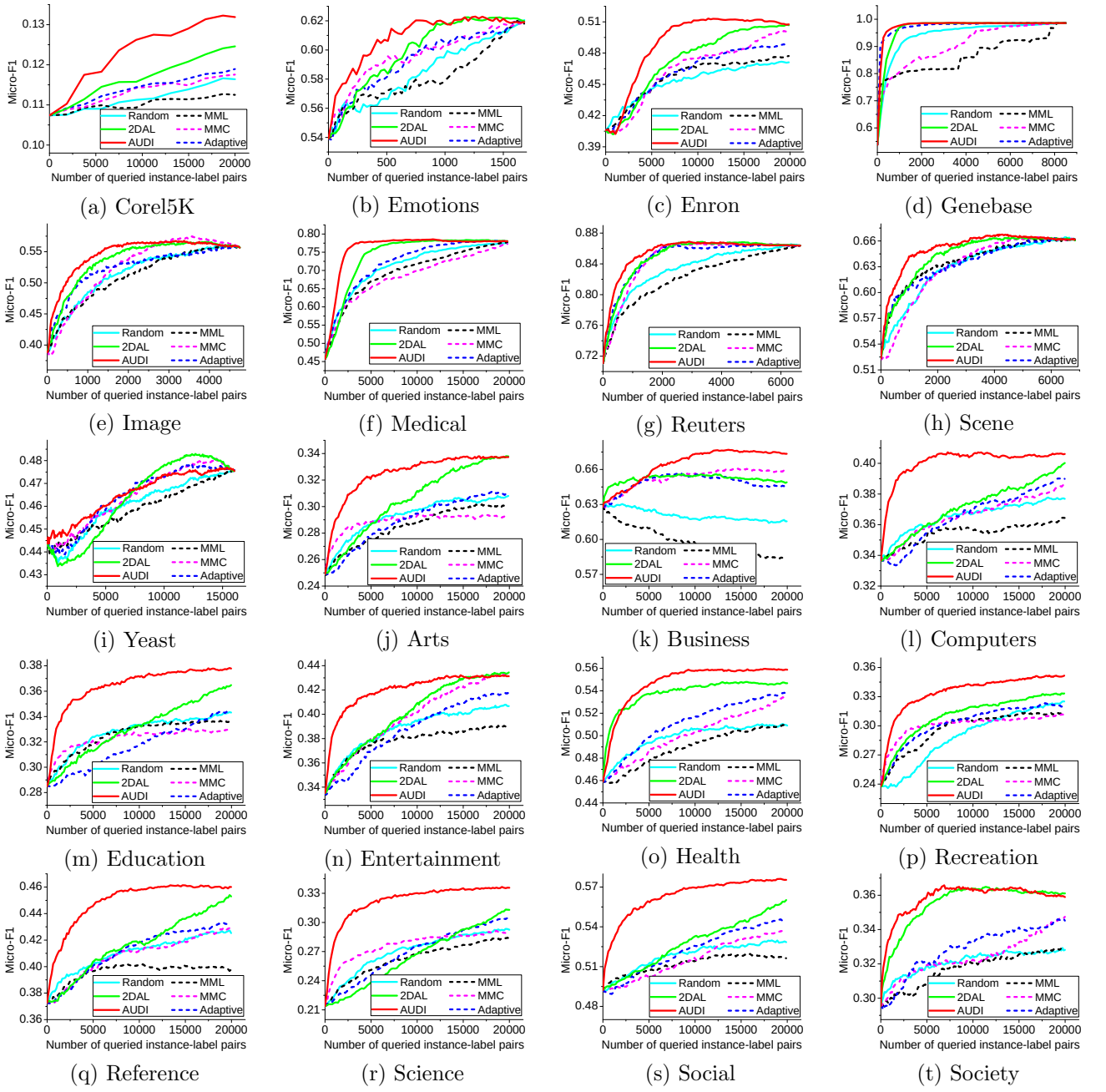


Fig.1. Comparison results on 20 datasets with regard to Micro-$F1$. The curves show the performances of different methods with the number of queries increasing.

labels, feature space dimensionality and label cardinality (LC), where LC counts the average number of labels per instance.

For each data set, we randomly divide it into two parts with equal size, take one part as test set and the other part as the unlabeled pool for active selection. The random data partition is repeated for 10 times, and average results over the 10 repeats are reported. At the very beginning of active learning, we randomly sample 5% instances from the unlabeled pool as initial labeled data. At each iteration of active learning, one instance or one instance-label pair is selected by the ac-

tive learning methods based on their own strategy, and then added into the labeled data. After $2 \times m$ instance-label pairs queried, we train a classification model on the labeled data and evaluate its performance on the holdout test data. The querying process is stopped if all data are fully labeled or the number of queried instance-label pairs reaches 20000.

We evaluate the performances of compared approaches on both micro-$F1$ and macro-$F1$, which are commonly used in multi-label learning [12, 14]. Micro-$F1$ computes the $F1$ measure by considering predictions of all instances on all labels together, while macro-$F1$ averages the F1 measure on each label. They are formally defined as Eqs(12) and (13), respectively.

$$\textit{micro-}F1 = \frac{2\sum_{i=1}^{n}\sum_{k=1}^{K} I[y_{ik}=1]\cdot I[\hat{y}_{ik}=1]}{\sum_{i=1}^{n}\sum_{k=1}^{K}(I[y_{ik}=1]+I[\hat{y}_{ik}=1])}, \tag{12}$$

$$\textit{macro-}F1 = \frac{1}{K}\sum_{k=1}^{K}\frac{2\sum_{i=1}^{n} I[y_{ik}=1]\cdot I[\hat{y}_{ik}=1]}{\sum_{i=1}^{n}(I[y_{ik}=1]+I[\hat{y}_{ik}=1])}, \tag{13}$$

where $\hat{y}_{ik}$ denotes the predicted label of the i-th instance on the k-th label, and $I[\cdot]$ is the indicator function.

To be fair, we use one-versus-all linear SVM (implemented with LIBLINEAR [45]) as the classification model for evaluating all the compared approaches. For MMC, the regression model is also implemented with LIBLINEAR. For AUDI, we use constant step size for SGD and set $m = 100$ as default. The other parameters are selected via 5-folds cross validation on the initial labeled data from the following candidate values $C \in \{1, 5, 10\}$, $\gamma \in \{0.01, 0.1\}$. For the other approaches, parameters are determined in the same way if no values suggested in their literatures.

4.2 Comparison Results

We plot the curves of micro-$F1$ and macro-$F1$ with the number of queried instance-label pairs increasing in Figs.1 and 2, respectively. Note that three approaches: Random, 2DAL and AUDI, which query instance-label pairs, are plotted in solid line, while the other three methods which query instances are plotted in dashed line.

First, we observe that results on micro-$F1$ and macro-$F1$ are consistent on most data sets. Generally speaking, methods querying instance-label pairs are more effective than those query instances, which is consistent with the results in [10]. This phenomenon is probably because multiple labels may be correlated, and thus redundancy of information may exist among the multiple labels of the same instance. This also explains why random selection can be better than some active approaches on some data. Among the methods querying instances only, Adaptive and MMC tend to be more effective than MML.

When comparing the proposed AUDI with other methods, no matter querying instance-label pairs or instances only, our method achieves the best performance in most cases. Especially on data sets with more labels, such as *Corel5K*, *Enron* and *Yahoo* data sets, the superiority of AUDI gets more significant. The only special case is on *yeast* data set, where AUDI achieves comparable performance with the best baseline on micro-$F1$, while is outperformed by MMC and MML on macro-$F1$.

4.3 Further Study on LCI

As stated before, the label cardinality inconsistency criterion may be less discriminative after a number of queries since the difference of label cardinality between instances may become very small. Suppose the difference is measured with the standard deviation of the label cardinality on all instances. First, we can obtain the standard deviation based on the ground-truth of training data, and denote it as LCstd_gt. Then, after

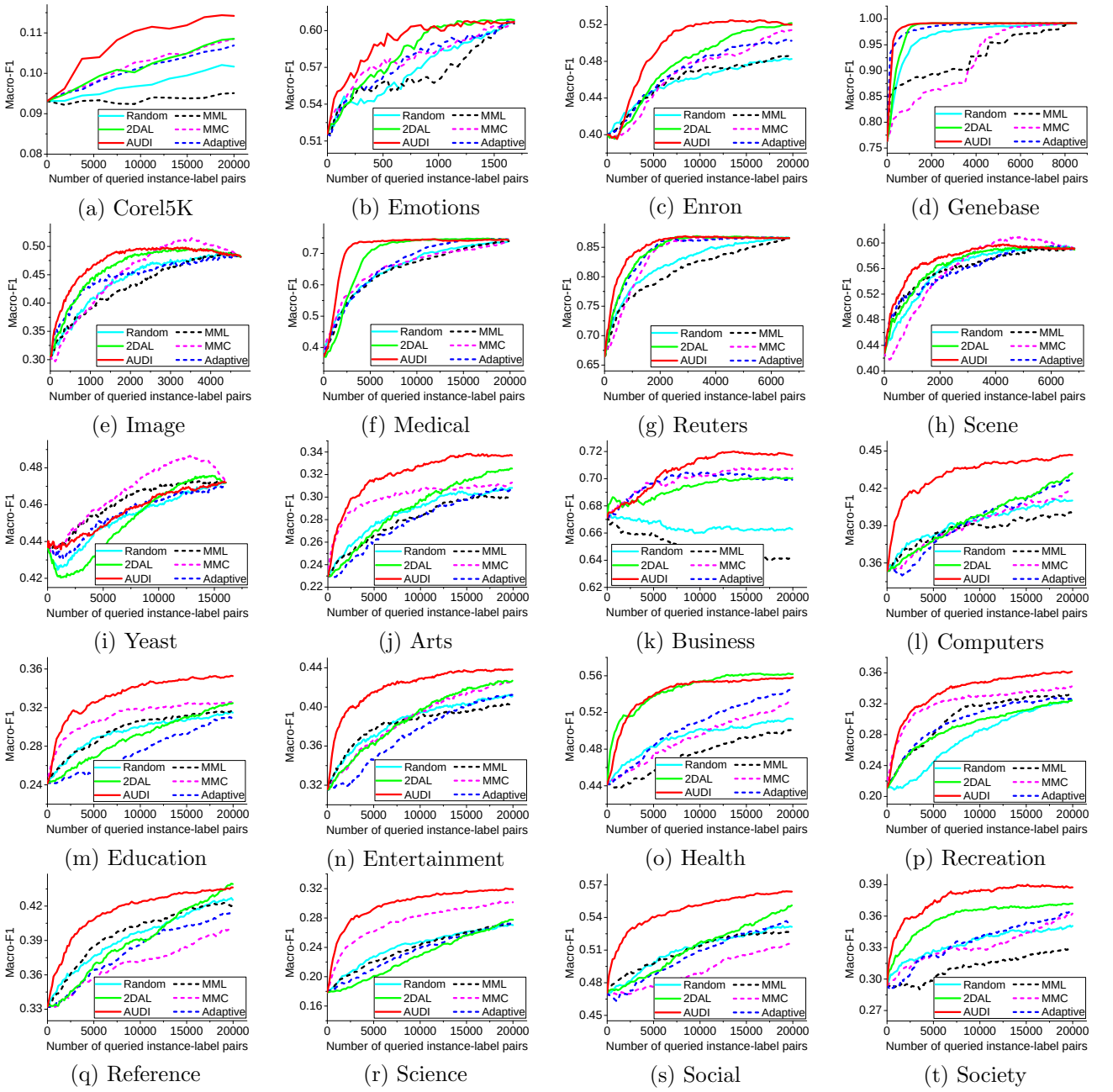
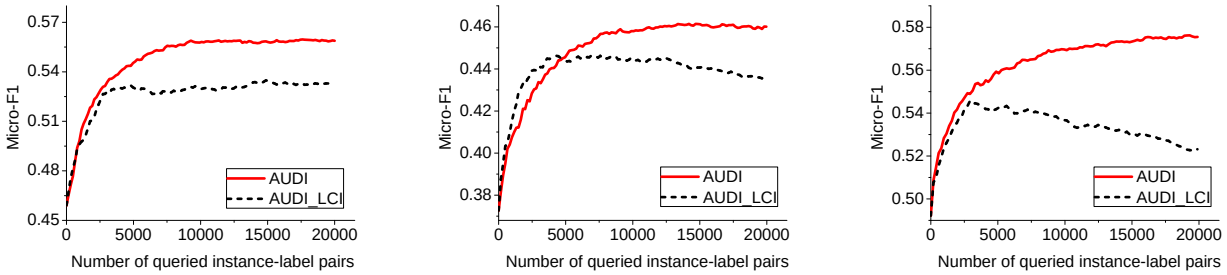


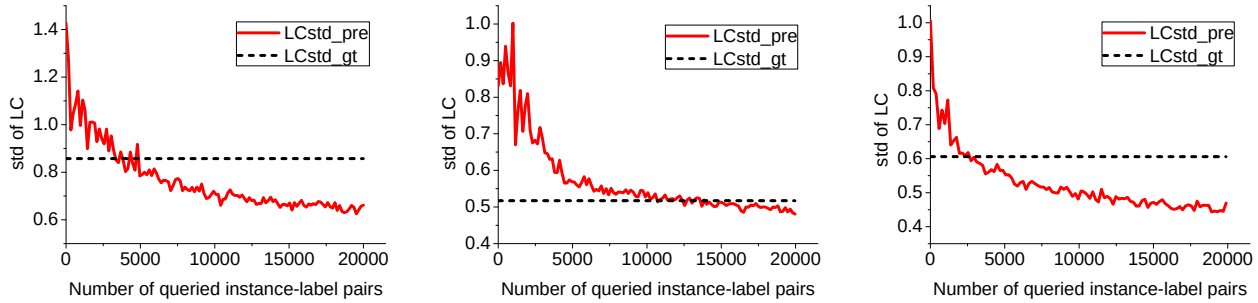
Fig.2. Comparison results on 20 datasets with regard to Macro-$F1$. The curves show the performances of different methods with the number of queries increasing.

some instance-label pairs queried, we can also calculate the standard deviation of label cardinality based on the predictions of the classification model, and denote it as LCstd_pre. At the beginning stage of active learning, the numbers of predicted labels on different instances can be very different, i.e., LCstd_pre can be large, thus we can easily identify the most uncertain instance as the one with largest LCI. However, with more and more queries, LCstd_pre may get smaller and smaller, and thus the LCI criterion gets less discriminative. Especially when LCstd_pre gets even smaller than LCstd_gt, LCI provides very limited information



(a) Health (b) Reference (c) Social

Fig.3. Micro-F1 curve of AUDI and AUDI_LCI. AUDI_LCI is comparable to AUDI at the beginning, and gets less effective after a few thousands of queries.



(a) Health (b) Reference (c) Social

Fig.4. Standard deviation of label cardinality.

on evaluating the uncertainty of instances.

To validate this hypothesis, we perform a further empirical study in this subsection. First, we construct a baseline approach termed AUDI_LCI, which is the same with AUDI except that: AUDI selects instances based on both LCI and diversity (as Eq.10) while AUDI_LCI considers LCI only. We plot the micro-$F1$ curves of AUDI and AUDI_LCI in Fig.3, and the LCstd_pre curve in Fig.4 with aligned x-axis. Due to space limitation, we only present the results on three data sets with typical results: *Health*, *Reference* and *Social*. In Fig.4, as expected, the LCstd_pre goes down as the number of queries increases, and finally gets smaller than LCIstd_gt. On the other hand, in Fig.3, AUDI_LCI is comparable to AUDI at the beginning, and gets less effective after a few thousands of queries. Surprisingly, we can see that AUDI_LCI losses its edge exactly when LCstd_pre is very close to or smaller than LCstd_gt. This observation validates the shortcoming of LCI as discussed above.

4.4 Distribution of Queried Labels

In multi-label learning, an instance is usually associated with only a small subset of all labels, and thus class-imbalance problem is usually suffered from. Based on such a phenomenon, it is guessed that positive labels may be more important than negative ones to train an accurate multi-label classification model, and it would be interesting to examine the distribution of labels queried by AUDI. We calculate the percentage of positive labels among all queried labels by AUDI at different stages of active learning, and present them in Table 2. The first column is the percentage of positive label calculated on the whole training set, while

the second to fifth columns are calculated after 25%, 50%, 75% and 100% of all queries, excluding the initial labeled data. It is worthy noting here *all queries* does not mean the whole data set, but the queries performed in our experiments as stated in Section 4.1.

Table 2. Percentage of Positive Labels Among Queried Labels at Different Stages of Active Learning

Dataset	all data	25%	50%	75%	100%
Corel5k	0.009	0.193	0.171	0.151	0.133
Emotions	0.312	0.414	0.356	0.328	0.311
Enron	0.064	0.116	0.128	0.112	0.100
Genebase	0.047	0.072	0.056	0.049	0.046
Image	0.247	0.439	0.382	0.309	0.247
Medical	0.028	0.074	0.049	0.037	0.029
Reuters	0.165	0.263	0.224	0.189	0.165
Scene	0.179	0.408	0.326	0.236	0.179
Yeast	0.303	0.287	0.280	0.286	0.303
Arts	0.063	0.098	0.122	0.116	0.105
Business	0.053	0.081	0.099	0.093	0.084
Computers	0.046	0.071	0.093	0.093	0.085
Education	0.044	0.070	0.099	0.099	0.090
Entertainment	0.067	0.114	0.132	0.119	0.107
Health	0.052	0.088	0.122	0.121	0.109
Recreation	0.065	0.105	0.122	0.113	0.103
Reference	0.035	0.053	0.073	0.076	0.070
Science	0.036	0.053	0.072	0.083	0.079
Social	0.033	0.052	0.070	0.075	0.068
Society	0.062	0.090	0.114	0.109	0.100

As we can see in Table 2, the percentage of positive labels in the queried labels is usually higher than that on the whole data set, especially at the early stages of active learning. The only outlier is *yeast* data set, on which, coincidentally, AUDI is less competitive. Obviously, AUDI favors positive label during the active selection. Given the superior performance of AUDI, this interesting phenomenon implies that positive labels may play a more important role in multi-label classification. Further, while no special effort is taken to predict whether the selected label is positive or negative in our algorithm, it suggests that explicitly exploiting the possible positive labels may be an interesting future direction of research on multi-label active learning.

5 Conclusion

An effective classification model along with a good selection criterion are the key factors of a successful active learning approach. This paper extends our preliminary research [31], and proposes a new multi-label active learning approach to iteratively query whether a label is positive on an instance. First, we present an incremental model for effective multi-label classification by incorporating label ranking with a threshold mechanism. Based on the model, we then propose to combine uncertainty and diversity as the criterion to select the most valuable instance-label pairs to query. The results of our extensive experiments demonstrate the superiority of our algorithm. In the future, we will try to study other active query strategies based on the label ranking model. Also, our experimental observation suggests that it is worthy to pay more attention on positive labels in multi-label active learning.

References

[1] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[2] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Learning Theory, 20th Annual Conference on Learning Theory, COLT2007*, pages 35–50, San Diego, CA, 2007.

[3] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th International Conference on Machine Learning*, pages 59–66, Washington, DC, 2003.

[4] Jingbo Zhu, Huizhen Wang, Benjamin K Tsou, and Matthew Ma. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on audio, speech, and language processing*, 18(6):1323–1331, 2010.

[5] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 36(10):1936–1994, 2014.

[6] Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):1–23, 2015.

[7] Hao Shao. Query by diverse committee in transfer active learning. *Frontiers of Computer Science*, 13:1–12, 03 2018.

[8] Yuling Ma, Chaoran Cui, Xiushan Nie, Gongping Yang, Kashif Shaheed, and Yilong Yin. Precourse student performance prediction with multi-instance multi-label learning. *Science China Information Sciences*, 62(2):29101, 2019.

[9] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2014.

[10] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Two-dimensional active learning for image classification. In *Proceedings of 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, AK, 2008.

[11] Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 62(7):76101, 2019.

[12] Xin Li and Yuhong Guo. Active learning with multi-label svm classification. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, Beijing, China, 2013.

[13] Mohan Singh, Anthony Brew, Derek Greene, and Pádraig Cunningham. Score normalization and aggregation for active learning in multi-label classification. Technical report, 2010.

[14] Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and ZhengChen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 917–926, Paris, France, 2009. ACM.

[15] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2764–2770, Barcelona, Spain, 2011.

[16] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[17] Chen-Wei Hung and Hsuan-Tien Lin. Multi-label active learning with auxiliary learner. In *Proceedings of the 3rd Asian Conference on Machine Learning*, pages 315–330, Taoyuan, Taiwan, 2011.

[18] Wei Bi and James Tin-Yau Kwok. Efficient multi-label classification with many labels. In *Proceedings of the 30th International Conference on Machine Learning*, pages 405–413, Atlanta, GA, 2013.

[19] Deepak Vasisht, Andreas C. Damianou, Manik Varma, and Ashish Kapoor. Active learning for sparse bayesian multilabel classification. In *The*

20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 472–481, New York, NY, 2014.

[20] Marc-André Carbonneau, Eric Granger, and Ghyslain Gagnon. Bag-level aggregation for multiple-instance active learning in instance classification problems. *IEEE transactions on neural networks and learning systems*, 30(5):1441–1451, 2018.

[21] Xia Chen, Guoxian Yu, Carlotta Domeniconi, Jun Wang, Zhao Li, and Zili Zhang. Cost effective multi-label active learning via querying subexamples. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 905–910. IEEE, 2018.

[22] Xuchun Li, Lei Wang, and Eric Sung. Multilabel svm active learning for image classification. In *Proceedings of 2004 International Conference on Image Processing*, volume 4, pages 2207–2210, Singapore, 2004.

[23] Klaus Brinker. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, pages 206–213. Springer, 2006.

[24] Mohan Singh, Eoin Curran, and Pádraig Cunningham. Active learning for multi-label image annotation. In *Proceedings of the 19th Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin, Ireland, 2008.

[25] Andrea Esuli and Fabrizio Sebastiani. Active learning strategies for multi-label text classification. In *Advances in Information Retrieval*, pages 102–113. Springer, 2009.

[26] Sheng-Jun Huang, Songcan Chen, and Zhi-Hua Zhou. Multi-label active learning: Query type matters. In *Proceedings of the 24rd International Joint Conference on Artificial Intelligence*, pages 946–952, Buenos Aires, Argentina, 2015.

[27] Jian Wu, Anqian Guo, Victor S. Sheng, Pengpeng Zhao, Zhiming Cui, and Hua Li. Adaptive low-rank multi-label active learning for image classification. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1336–1344, Mountain View, CA, 2017.

[28] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1837–1845, Honolulu, HI, 2017.

[29] Xiaoyu Zhang, Jian Cheng, Changsheng Xu, Hanqing Lu, and Songde Ma. Multi-view multi-label active learning for image classification. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 258–261, New York City, 2009.

[30] Peng Wang, Peng Zhang, and Li Guo. Mining multi-label data streams using ensemble-based active learning. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 1131–1140, 2012.

[31] Sheng-Jun Huang and Zhi-Hua Zhou. Active query driven by uncertainty and diversity for incremental multi-label learning. In *Proceedings of the 13th IEEE International Conference on Data Mining*, pages 1079–1084, Dallas, TX, 2013.

[32] Sheng-Jun Huang, Wei Gao, and Zhi-Hua Zhou. Fast multi-instance multi-label learning. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 1868–1874, Quebec, Canada, 2014.

[33] Sheng-Jun Huang, Nengneng Gao, and Songcan Chen. Multi-instance multi-label active learning. In *Proceedings of the 26rd International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017.

[34] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[35] Shai Ben-David, David Loker, Nathan Srebro, and Karthik Sridharan. Minimizing the misclassification error rate using a surrogate convex loss. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, 2012.

[36] Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, pages 97–112, Denmark, 2002.

[37] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis Vlahavas. Multi-label classification of music into emotions. In *Proceedings of the 9th International Conference of Music Information Retrieval*, page 325, Philadelphia, PA, 2008.

[38] Bryan Klimt and Yiming Yang. Introducing the enron corpus. In *Proceedinds of the 1st Conference on Email and Anti-Spam*, Mountain View, California, 2004.

[39] Sotiris Diplaris, Grigorios Tsoumakas, Pericles A Mitkas, and Ioannis Vlahavas. Protein classification with multiple algorithms. In *Advances in Informatics*, pages 448–456. Springer, 2005.

[40] Min-Ling Zhang and Zhi-Hua Zhou. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[41] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

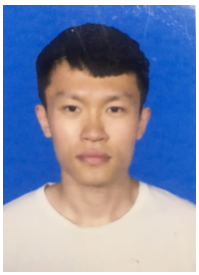
[42] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

[43] André Elisseeff and Jason Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, pages 681–687, Vancouver, Canada, 2002.

[44] Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*, pages 721–728, Vancouver, Canada, 2003.

[45] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

Sheng-Jun Huang received the BSc and PhD degrees in computer science from Nanjing University, NanJing, in 2008 and 2014, respectively. He is now a professor in the College of Computer Science and Technology at Nanjing University of Aeronautics and Astronautics. His main research interests include machine learning and data mining. He has been selected to the Young Elite Scientists Sponsorship Program by China Association for Science and Technology in 2016, and won the China Computer Federation Outstanding Doctoral Dissertation Award in 2015, the KDD Best Poster Award at the in 2012, and the Microsoft Fellowship Award in 2011. He is a Junior Associate Editor of *Frontiers of Computer Science*.



Guo-Xiang Li is a second year M.Sc. student of College of Computer Science and Technology in Nanjing University of Aeronautics and Astronautics and a member of PARNEC Group. He received his B.Sc. degree in School of Computer Science in June 2018 from Nanjing University of Posts and Telecommunications. In the same year, he was admitted to study for a M.Sc. degree in Nanjing University of Aeronautics and Astronautics.

Wen-Yu Huang is a second year M.Sc. student of College of Computer Science and Technology in Nanjing University of Aeronautics and Astronautics and a member of PARNEC Group. He received his B.Sc. degree in software engineering in June 2018 from Wuhan University of Technology. In the same year, he was admitted to study for a M.Sc. degree in Nanjing University of Aeronautics and Astronautics.

Shao-Yuan Li is an assistant professor in Nanjing University of Aeronautics and Astronautics. Her research interest is mainly on machine learning and data mining. She is currently working on learning with incomplete/nonperfect information, specifically on multi-label and multi-view problems. She has published serveral papers on and reviewer of top conferences and journals, was winner of the Grand and SME Segment winner of the PAKDD 2012 Data Mining Competition, and the Best Paper Award of PRICAI 2018.