Evolving Systems (2019) 10:63–78
DOI 10.1007/s12530-017-9202-z

ORIGINAL PAPER

Multi-label active learning: key issues and a novel query strategy

Everton Alvares Cherman[1] · **Yannis Papanikolaou**[2] · **Grigorios Tsoumakas**[2] · **Maria Carolina Monard**[1]

Received: 24 January 2017 / Accepted: 19 August 2017 / Published online: 30 August 2017

Abstract Active learning is an iterative supervised learning task where learning algorithms can actively query an oracle, i.e. a human annotator that understands the nature of the problem, to obtain the ground truth. The motivation behind this approach is to allow the learner to interactively choose the data it will learn from, which can lead to significantly less annotation cost, faster training and improved performance. Active learning is appropriate for machine learning applications where labeled data is costly to obtain but unlabeled data is abundant. Most importantly, it permits a learning model to evolve and adapt to new data unlike conventional supervised learning. Although active learning has been widely considered for single-label learning, applications to multi-label learning have been more limited. In this work, we present the general framework to apply active learning to multi-label data, discussing the key issues that need to be considered in pool-based multi-label active learning and how existing solutions in the literature deal with each of these issues. We further propose a novel aggregation method for evaluating which instances are to be annotated. Extensive experiments on 13 multi-label data sets with different characteristics and under two different applications settings (transductive, inductive) convey a consistent advantage of our proposed approach against the rest of the approaches and, most importantly, against passive supervised learning and reveal interesting aspects related mainly to the properties of the data sets, and secondarily to the application settings.

Keywords Supervised learning · Multi-label learning · Active learning · Pool-based strategies · Knowledge discovery

1 Introduction

Most of present-day applications involve operation in dynamically and drastically ever-changing environments. In such settings, systems that have the ability to adapt to the new conditions and evolve can have a decisive advantage over static and monolithic structures. More specifically, in the area of knowledge discovery and supervised learning, models that have the ability to continually take advantage of new data as they become available, can have a significant edge over conventional static approaches.

Active learning is a characteristic paradigm of such a dynamic approach, with the ability of constructing learning models that will be able to fully adapt to new data. As opposed to conventional supervised learning, it allows the model, in other words the classifier, to interactively ask for supervision from an oracle (most usually a human). The motivation is twofold: first, when dealing with learning tasks from domains with few labeled and abundant unlabeled data, this approach can effectively bypass the expensive task of labeling, since the classifier, based on some strategy, will only request manual annotation for a few characteristic

✉ Yannis Papanikolaou
ypapanik@csd.auth.gr

Everton Alvares Cherman
echerman@icmc.usp.br

Grigorios Tsoumakas
greg@csd.auth.gr

Maria Carolina Monard
mcmonard@icmc.usp.br

[1] Institute of Mathematics and Computer Sciences, University of Sao Paulo, Sao Carlos, SP, Brazil

[2] Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

instances. Second, this approach allows for classifiers that receive data in a stream-like fashion and dynamically choose which of this new data should be annotated to be used for their training (Zliobaite et al. 2011).

Furthermore, evolving predictive systems usually operate in domains where data is received in real time, continuously and changing over time. In such settings, it is often unfeasible to store data and it is critical to constantly update the model with the new training data. Obtaining such data though, can often be costly. Active learning is a technology for making the best of an annotation budget in such cases.

There has been a substantial body of work regarding active learning for single-label classification in the literature (McCallumzy and Nigamy 1998; Tong and Koller 2001; Settles 2010). However, this is not the case for multi-label learning, where each object can be associated with multiple labels simultaneously (Tsoumakas et al. 2012).

In this work, we present the general framework for applying active learning to multi-label tasks, studying the main aspects of an active learning model and discussing the key issues that need to be taken into account in such a configuration. We focus on the pool-based active learning scenario (Settles 2010), in which all unlabeled data are first evaluated and choices for which instances to be annotated are made subsequently by the model. Such an approach is suitable for a large number of real-world problems, such as text classification, image classification and retrieval, video classification, speech recognition and cancer diagnosis (Settles 2010; Zhang et al. 2014; Ye et al. 2015; Huang et al. 2015). Given that in a stream-based scenario new data arrive most often in batches that can be essentially treated with pool-based approaches, it is reasonable to assume that our work is, with minor adjustments, applicable to stream-based active learning as well.

An earlier and significantly shorter version of this work, has been previously presented in Cherman et al. (2016). We here extend this line of work, by substantially extending our experiments: we consider thirteen data sets instead of two in the previous paper, we employ two additional algorithms and consider also transductive inference apart from inductive inference for experiments that use the remaining examples in the query pool for testing,

Furthermore, in this work we propose a novel aggregation function that evaluates examples to be picked for active labeling. This approach considers the scores and the ranking of labels delivered by a given algorithm to assess if an example is to be picked for active labeling. The motivation behind this approach, is to try identifying the certainty of the algorithm in differentiating positive and negative labels for a given example. Our results show a consistent advantage of our proposed method with respect to passive supervised learning and to the rest of the methods as well.

To summarize, the contributions of this work are as follows:

- we present the key issues that have to be considered when applying active learning on multi-label data and we thoroughly describe the existing approaches regarding these issues in the literature (Sect. 2)
- we propose a novel aggregation method regarding the evaluation and subsequent choice of the unlabeled instances to be manually annotated (Sect. 2.4).
- we conduct extensive experiments on 13 multi-label data sets, with two multi-label algorithms and for both inductive and transductive inference, studying the performance and behavior of the different methods and approaches on a variety of conditions and comparing them with conventional passive supervised learning (Sect. 3).

2 Active learning for multi-label data

In this section, we first briefly present the concepts of active learning and multi-label learning and then focus on the key issues that need to be considered when attempting to apply active learning on multi-label data.

2.1 Active learning

In conventional supervised learning, the learner is passively given a set of labeled data points to be trained on. Active learning on the other hand, permits the learner to interactively request supervision, or labeling in other words, for the data points of its own choice.

There are mainly three active learning approaches (Settles 2010; Aggarwal et al. 2014)

1. Membership query synthesis;
2. stream-based;
3. pool-based.

In the first case, the learner may query any unlabeled instance in the input space. That also includes queries generated by the learner de novo (synthesis). In the second setting, data points are made available continuously in a stream-like fashion, and therefore decisions about whether an unlabeled instance should or not be labeled are made individually or in small batches. The pool-based scenario assumes that a pool of unlabeled data is made available from the onset of training. All instances from this unlabeled pool are evaluated before selecting which of them are to be labeled.

2.2 Multi-label learning

Unlike single-label or multi-class learning, multi-label learning concerns supervised learning tasks in which there exist multiple target variables and a subset of them can be assigned to an instance simultaneously. Formally, let D be a training set composed of N examples $E_i = (\mathbf{x}_i, Y_i), i = 1 \dots N$. Each example E_i is associated with a feature vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iM})$ described by M features X_j, $j = 1 \dots M$, and a subset of labels $Y_i \subseteq L$, where $L = \{y_1, y_2, \dots, y_q\}$ is the set of q labels. A multi-label learning task consists of generating a classifier H, which given an unlabeled instance $E = (\mathbf{x}, ?)$, is capable of accurately predicting its subset of labels Y, i.e., $H(E) \rightarrow Y$.

In the general setting, a multi-label learning model can produce a ranking of labels, relevance scores or marginal probabilities per label, or even the full joint probability distribution of labels per instance.

Multi-label learning methods are divided into two broad classes, algorithm adaptation and problem transformation methods (Tsoumakas et al. 2009). Methods in the first category extend specific single-label learning algorithms to deal with multi-label data directly. Methods in the second category transform a multi-label problem into one or more single-label problems in which any traditional single-label learning algorithms can be applied. *Binary Relevance* (BR), is one of the most widely employed problem transformation methods, that proceeds by decomposing the multi-label problem into q binary single-label problems, one for each label in L.

2.3 Manual annotation

A first key issue concerning an active learning system relates to the manual annotation of the instances selected by the learner. Most often, instances are annotated in batches, e.g. ground truth acquisition for the ImageCLEF 2011 photo annotation and concept-based retrieval tasks was achieved via crowd-sourcing in batches of 10 and 24 images (Nowak et al. 2011). An annotator can accomplish this task either *instance-wise* (for each instance the annotator determines the relevancy to each label) or *label-wise* (for each label the annotator determines relevancy to each instance).[1]

Let us consider a request for the annotation of n instances with q labels. Let c_o be the average cost of understanding an instance, c_l be the average cost of understanding a label and c_{lo} be the average cost of deciding whether an instance should be annotated with a particular label or not. Setting aside the cognitive and psychological aspects of the annotation process, such as our short-term memory capacity, a rough estimate of the total cost of instance-wise annotation will be given by:

$$n[c_o + q(c_l + c_{lo})] = nc_o + nqc_l + nqc_{lo}$$

Similarly, a rough estimate of the total cost of label-wise annotation will be:

$$q[c_l + n(c_o + c_{lo})] = qc_l + nqc_o + nqc_{lo}$$

Assuming that the cost of label-wise annotation is smaller than that of instance-wise annotation, we have:

$$\begin{aligned} qc_l + nqc_o + nqc_{lo} &< nc_o + nqc_l + nqc_{lo} \\ qc_l + nqc_o &< nc_o + nqc_l \\ n(q-1)c_o &< q(n-1)c_l \\ c_o < \frac{q(n-1)}{n(q-1)}c_l &\approx \frac{qn}{nq}c_l = c_l key \end{aligned}$$

In other words, the choice of the annotation approach, largely depends on the instance and label understanding costs.

2.4 Evaluation of unlabeled instances

The most fundamental part of an active learning algorithm concerns the way it evaluates the informativeness of unlabeled instances. In a multi-label setting, the evaluation function (*query*) comprises two important parts:

1. a *scoring* function to evaluate instance-label pairs; and
2. an *aggregating* function to aggregate these scores.

Algorithm 1 shows the general procedure for a batch-size = t, *i.e.*, t examples are annotated in each round. The evaluation function *query* calculates the evidence value of each example $E_i \subset D_u$ and returns the t most informative instances, according to the evidence value used. In each round, these t examples will be labeled by the oracle and included in the set D_l of labeled examples.

input : D_l: labeled pool; D_u: unlabeled pool; E_i: multi-label example;
L: set of labels; Y_i: subset of labels associated to E_i; t: batch size;
R: number of rounds; F: multi-label learner; *Oracle*: the annotator;
for $r = 1, 2, .., R$ **do**
  $H \leftarrow F(D_l)$
  $\{E_i\}_{i=1}^{t} \leftarrow query(H, L, D_u, t)$
  $\{Y_i\}_{i=1}^{t} \leftarrow Oracle(\{E_i\}_{i=1}^{t})$
  $D_l \leftarrow D_l \cup \{(E_i, Y_i)\}_{i=1}^{t}$
  $D_u \leftarrow D_u - \{E_i\}_{i=1}^{t}$
end

Algorithm 1: Multi-label active learning procedure for the instance-wise annotation approach.

[1] Instance-wise and label-wise annotation have been called global and local labeling respectively in Esuli and Sebastiani (2009).

Algorithm 2 shows the *query* function of a multi-label active learning procedure. The *scoring* function considers instance-label pairs (E_i, y_j) and evaluates the participation $(e_{i,j})$ of label y_j in instance E_i. It returns an evidence value $e_{i,j}$ for all instances $E_i \subset D_u$ and for each label $y_j \in L = \{y_1, y_2, \ldots, y_q\}$. The *aggregating* function considers the q evidence values $e_{i,1}, e_{i,2}, \ldots, e_{i,q}$ of each instance E_i given by the *scoring* function, and combines these values into a unique evidence value e_i.

input : D_u: unlabeled pool; L: set of labels; H: multi-label classifier
output: The t instances with higher evidences

for $E_i \in D_u$ **do**
  for $y_j \in L$ **do**
    $e_{i,j} \leftarrow scoring(D_u, H, E_i, y_j)$
  end
  $e_i \leftarrow aggregating(e_{i,1}, e_{i,2}, ..., e_{i,q})$
end
$query \leftarrow best(e_1, e_2, ..., t, D_u)$

Algorithm 2: The *query* function

The following three families of measures have been proposed in the literature for evaluating instance-label pairs (*scoring*):

1. Confidence-based score (Brinker 2006; Esuli and Sebastiani 2009; Singh et al. 2010). The distance of the confidence of the prediction from the *average* value is used. The nature of this value depends on the bias of learner. It could be a margin-based value (distance from the hyper-plane), a probability-based value (distance from 0.5) or other. The value returned by this approach represents how far an example is from the boundary decision threshold between positive and negatives examples. We are interested in examples that minimize this score. In the following, we will denote this method as *conf*.
2. Ranking-based score (Singh et al. 2010). This strategy works like a normalization approach for the values obtained from the confidence-based strategy. The confidences given by the classifier are used to rank the unlabeled examples for each label. We are interested in examples that maximize this score. This score will be represented by *rank* in the rest of the paper.
3. Disagreement-based score (Hung and Lin 2011; Yang et al. 2009). Unlike the other approaches, this strategy uses two base classifiers and measures the difference between their predictions. We are interested in maximizing this score. The intuitive idea is to query the examples that most disagree in their classifications and could be most informative. In the literature, there have been proposed three ways to combine confidence values from two base classifiers:

I. The Maximum Margin Reduction (MMR) criterion uses a major classifier which outputs confidence values and an auxiliary classifier that outputs decisions (positive/negative). The auxiliary classifier is used to determine how conflicting the predictions are.
II. The Hamming Loss Reduction (HLR) approach considers a more strict disagreement using the decisions output by both classifiers to decide if there is disagreement or agreement between each label prediction of an example.
III. The soft Hamming Loss Reduction (SHLR) method tries to make a balance between MMR and HLR through a function that defines the influence of each approach in the final score.

In the experiments, we do not consider the disagreement-based strategies, due to the inferior results that were obtained in previous work (Cherman et al. 2016). After having obtained the instance-label scores, there are two main aggregation strategies for combining the instance-label scores to an overall instance score:

1. averaging of the instance-label scores across all labels (*avg*). Thus, given the q instance-label scores $e_{i,j}$ of instance E_i, the overall instance-label score of instance E_i is given by:

$$e_i = aggregating_{avg}\left(\{e_{i,j}\}_{j=1}^{q}\right) = \frac{\sum_{j=1}^{q} e_{i,j}}{q}$$

2. considering the optimal (minimum or maximum) of the instance-label scores (*min/max*), given by:

$$e_i = aggregating_{min/max}\left(\{e_{i,j}\}_{j=1}^{q}\right) = min/max\left(\{e_{i,j}\}_{j=1}^{q}\right)$$

Note that for HLR, only the average aggregation strategy makes sense, as taking the maximum would lead to a value of 1 for almost all unlabeled instances and would not help in discriminating among them. We here propose a new aggregation strategy which we will denote as *dev*.

3. *dev* is based on the differences (deviations) between the values of evidence $e_{i,j}$ of each instance. The motivation behind this strategy is that an instance that contains small differences in the values between the evidences of the labels predicted as positive and the evidence of the labels predicted as negative indicate uncertainty in the prediction of the instance, which makes it a potential candidate for oracle active labeling. Equation 1 defines the *dev* strategy.

Table 1 Illustrative example of the evaluation method *conf*

	Raw score				$Scoring_{conf}$		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
E_1	0.70	0.30	0.31		0.20	0.20	0.19
E_2	0.35	0.42	0.60		0.15	0.08	0.10
E_3	0.45	0.51	0.80	⇒	0.05	0.01	0.30
E_4	0.48	0.52	0.80		0.02	0.02	0.30
E_5	0.20	0.30	0.49		0.30	0.20	0.01

In this example, the threshold 0:5 is used to process the raw values of score toobtain the confidence values

Table 2 Illustrative example of the evaluation method *score*

	Raw score				$Scoring_{score}$		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
E_1	0.70	0.30	0.31		0.70	0.30	0.31
E_2	0.35	0.42	0.60		0.35	0.42	0.60
E_3	0.45	0.51	0.80	⇒	0.45	0.51	0.80
E_4	0.48	0.52	0.80		0.48	0.52	0.80
E_5	0.20	0.30	0.49		0.20	0.30	0.49

In this case, the actual values of score are used as the evaluation function

Table 3 Illustrative example of the evaluation method *rank*

	Raw score				$Scoring_{rank}$		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$
E_1	0.70	0.30	0.31		0.5	1.5	1.5
E_2	0.35	0.42	0.60		2.5	0.5	0.5
E_3	0.45	0.51	0.80	⇒	1.5	0.5	2.5
E_4	0.48	0.52	0.80		0.5	1.5	2.5
E_5	0.20	0.30	0.49		3.5	1.5	0.5

The threshold 0.5 and uniform distribution of labels (without imbalance) are considered

$$aggregating_{dev}\left(\left\{e_{i,j}\right\}_{j=1}^{q}\right) = e_i = avgpos\left(\left\{e_{i,j}\right\}_{j=1}^{q}\right) - firstneg\left(\left\{e_{i,j}\right\}_{j=1}^{q}\right) \quad (1)$$

where the *avgpos* function returns the average value of the labels evidences classified as positive and the *firstneg* function returns the evidence value of the label closest to being classified as positive but is actually classified as negative.

Thus, the lower the value of the $aggregating_{dev}$ function, the higher the instance's priority to be selected for oracle active labeling. We should note that this strategy is appropriate when applied directly to the raw score produced from the given classifier (in the rest we denote this approach as *score*) and therefore is not applicable to confidence or ranking-based scoring strategies that manipulate the raw scores. To illustrate how the methods proceed, Tables 1, 2, 3, 4, 5 and 6 depict some characteristic examples of each of the scoring and aggregation methods.

2.5 Experimental protocol

Besides the multi-label active learning strategies themselves, the way that they are evaluated is another important issue to consider. Some aspects to be considered are the size of the initial labeled pool, the batch's size, the set of examples used as testing, the sampling strategy and also the evaluation approach. Next, these aspects are described with references to previous work in the literature.

Regarding the initial labeled pool, different papers built it in different ways. In Singh et al. (2010), the examples are chosen to have at least one example positive and one negative for each label. In Yang et al. (2009), 100–500 examples were selected randomly to compose the initial labeled pool. In Esuli and Sebastiani (2009), the first 100 chronologically examples were selected. In Brinker (2006), the author choose randomly ten examples to compose the initial labeled

Table 4 Illustrative example of the evaluation method *conf* with application of *avg* and MIN aggregation functions

	Raw score				$Scoring_{conf}$				*Aggregating*	
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$		AVG	MIN
E_1	0.70	0.30	0.31		0.20	0.20	0.19		0.20	0.19
E_2	0.35	0.42	0.60		0.15	0.08	0.10		**0.11**	0.08
E_3	0.45	0.51	0.80	⇒	0.05	0.01	0.30	⇒	0.12	**0.01**
E_4	0.48	0.52	0.80		0.02	0.02	0.30		0.11	0.02
E_5	0.20	0.30	0.49		0.30	0.20	0.01		0.17	0.01

Bold objects would be those selected for oracle labeling

Table 5 Illustrative example of the *score* evaluation method with application of *avg*, MIN and *dev* aggregation functions

	Raw score				$Scoring_{rank}$				*Aggregating*		
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$		AVG	MIN	div
E_1	0.70	0.30	0.31		0.70	0.30	0.31		0.44	0.30	0.39
E_2	0.35	0.42	0.60		0.35	0.42	0.60		0.46	0.35	**0.18**
E_3	0.45	0.51	0.80	⇒	0.45	0.51	0.80	⇒	0.59	0.45	0.29
E_4	0.48	0.52	0.80		0.48	0.52	0.80		0.60	0.48	0.28
E_5	0.20	0.30	0.49		0.20	0.30	0.49		**0.33**	**0.20**	0.19

For *dev*, only the evidence with the highest score value was considered positive. Bold objects would be those selected for oracle labeling

Table 6 Illustrative example of the evaluation method *rank* with application of aggregation functions *avg* and MIN

	Raw score				$Scoring_{rank}$				*Aggregating*	
	$f(y_1)$	$f(y_2)$	$f(y_3)$		$e_{i,1}$	$e_{i,2}$	$e_{i,3}$		AVG	MIN
E_1	0.70	0.30	0.31		0.5	1.5	1.5		**1.17**	0.50
E_2	0.35	0.42	0.60		2.5	0.5	0.5		1.17	**0.50**
E_3	0.45	0.51	0.80	⇒	1.5	0.5	2.5	⇒	1.50	0.50
E_4	0.48	0.52	0.80		0.5	1.5	2.5		1.50	0.50
E_5	0.20	0.30	0.49		3.5	2.5	3.5		1.83	0.50

Bold objects would be those selected for oracle labeling

pool. Gao et al. (2016) randomly sample 5% of the instances from the unlabeled pool as initial training labeled data.

The batch size defines how many examples are queried in each round of active learning. In Singh et al. (2010) and Brinker (2006), only one example was queried per round. Esuli and Sebastiani (2009) chose 50 examples in each round, while Yang et al. (2009) performed experiments with both 50 and 20 examples. Finally, Gao et al. (2016) perform five fold cross-validation to choose for each data set the batch size taking values between five and ten instances per round.

There are basically two different ways to define the test set. The first one is to consider a totally separated test set. This was followed by Esuli and Sebastiani (2009) and though not explicitly mentioned, it seems to have also been followed by Brinker (2006). The second way is to use the remaining examples in the unlabeled pool for testing. This approach was used by Singh et al. (2010), Yang et al. (2009) and Gao et al. (2016).

It is worth noting that the quality of the model assessed using this second approach holds for examples in the unlabeled pool, and does not necessarily hold for new unlabeled data. Although there is a lack of discussion about this topic in the active learning literature, the decision of which evaluation approach to use depends on the application's nature. Most learning applications are interested in building a general model from a training set of examples to predict future new examples, e.g., this kind of application uses inductive inference algorithms to make its predictions. An experimental protocol using a separate test set is the correct evaluation approach for the performance assessment in the inductive inference setting. The remaining evaluation approach is biased by the active learner and hence the evaluation on these remaining examples will not be representative of the

Table 7 Statistics of the data sets used throughout the experiments

					Labels							
Name	Domain	Instances	Features	#Dist	|L|	*Cardinality*	*Density*	*Min*	1*Q*	*Med*	3*Q*	*Max*
bibtex	Text	7395	1836	2856	159	2.402	0.015	51	61	82	129	1042
cal500	Music	502	68	502	174	26.044	0.150	5	15	39	109	444
corel16k	Image	13811	500	4937	161	2.867	0.018	25	67	115	264	3170
corel5k	Image	5000	499	3175	374	3.522	0.009	1	6	15	39	1120
emotions	Music	593	72	27	6	1.869	0.311	148	166	170	185	264
enron	Text	1702	1001	753	53	5.31	0.064	1	13	26	107	913
llog	Text	1460	1004	304	75	1.18	0.02	1	4	11	22	171
medical	Text	978	1449	94	45	1.245	0.028	1	2	8	34	266
ohsumed	Text	13929	1002	1147	23	1.663	0.007	135	386	712	1220	3952
scene	Image	2407	294	15	6	1.074	0.179	364	404	429	432	533
slashdot	Text	3782	1079	156	22	1.18	0.05	0	26	179	250	584
tmc2007	Text	28596	500	1341	22	2.158	0.098	403	548	1483	2914	16918
yeast	Biology	2417	103	198	14	4.237	0.303	34	324	659	953	1816

actual distribution of new unseen examples, which is the case for inductive inference.

However, there are active learning applications that want to predict labels of an *a priori* known specific set of examples. For example, in a real world personal image annotation scenario, the user would like to annotate some images of his/her collection and after few rounds of active learning, the system would annotate the remaining image in the collection (Singh et al. 2010). For such an application, the learning assessment should use the remaining examples in the query pool.

The learning curve is the most common evaluation approach used to assess active learning techniques. A learning curve plots the evaluation measure considered as a function of the number of new instance queries that are labeled and added to D_l. Thus, given the learning curves of two active learning algorithms, the algorithm which dominates the other for more or all the points along the learning curve is better than the other. Besides the learning curve, Singh et al. (2010), Yang et al. (2009) and Esuli and Sebastiani (2009) also used the value of the evaluation measure in the end of some specific number of rounds to assess the active learning techniques.

3 Experiments

We here describe the experiments performed, presenting the data sets, evaluation measures, experimental setup and the relevant results. The active learning algorithms described in Sect. 2.4, as well as the active learning evaluation framework, were implemented under Mulan[2] (Tsoumakas et al. 2011), a Java package for multi-label learning based on Weka.[3] Our implementation is publicly available at http://www.labic.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/active-learning.zip.

3.1 Data sets

We employed 13 data sets from different domains. Specifically, *bibtex*, *cal500*, *corel16k*, *corel5k*, *emotions*, *enron*, *medical*, *scene*, *tmc2007* and *yeast* were obtained from Mulan's website,[4] while *llog* and *slashdot* were obtained from Meka's website.[5] Finally, *ohsumed* is a widely used data set that is a subset of the MEDLINE database from years 1987–1991, with a labelset of the 23 Medical Subject Headings (MeSH) tags of cardiovascular diseases group.

In Table 7, we show the data sets statistics, with *Instances* denoting the number of total instances, *Features* the number of features and *#Dist* the number of distinct label sets. Similarly, |*L*| stands for the number of labels, *Cardinality* for the average number of labels of the examples in *D*, *Density* for the the average number of labels of the examples in *D* divided by |*L*| while *Min*, *Med* and *Max* refer to the minimum, average and maximum label frequencies respectively. Finally, the first and third quartiles of the label distributions are represented by *1Q* and *3Q*.

3.2 Evaluation measures

For the evaluation of the multi-label classification models, we employed three measures in total, *Micro-F*, *Macro-F* and

[2] http://mulan.sourceforge.net.

[3] http://www.cs.waikato.ac.nz/ml/weka.

[4] http://mulan.sourceforge.net/data sets.html.

[5] http://meka.sourceforge.net/.

Ranking Loss. The first two measures essentially consist two different averaging schemes of the *F-measure*, which is used for single-label classification. Specifically, the F-measure is defined as

$$F\text{-}measure = \frac{2T_P}{2T_P + F_P + F_N} \tag{2}$$

with T_P denoting the true positives, F_P the false positives and F_N the false negatives. The F-measure combines both the *Precision* and *Recall* measures, being the harmonic mean of them and obtains values between zero and one, with F-measure = 1 signifying a perfect classification.

The Micro-F and Macro-F measures are defined as the micro- and macro- averages of the F-measure respectively:

$$Micro\text{-}F = \frac{2 \times \sum_{l=1}^{|L|} tp_l}{2 \times \sum_{l=1}^{|L|} tp_l + \sum_{l=1}^{|L|} fp_l + \sum_{l=1}^{|L|} fn_l} \tag{3}$$

$$Macro\text{-}F = \frac{1}{|L|} \sum_{l=1}^{|L|} \frac{2 \times tp_l}{2 \times tp_l + fp_l + fn_l} \tag{4}$$

We note that, by definition, Micro-F typically favors frequent labels while Macro-F is more influenced by rare labels.

Finally, Ranking Loss expresses the number of times that irrelevant labels are ranked higher than relevant labels and is defined as:

$$Ranking\text{ - }Loss = \frac{1}{N} \sum_{d=1}^{N} \frac{1}{|Y_d||\overline{Y_d}|} \left| \left\{ (y_a, y_b) : r_d(y_a) > r_d(y_b), \times \ (y_a, y_b) \in Y_d \times \overline{Y_d} \right\} \right| \tag{5}$$

In the experiments, we consider the respective learning curves for each of the above measures and use the Final Value (*FV*) (Yang et al. 2009) and the Area Under the Learning Curve (*AULC*) (Settles and Craven 2008). Specifically, FV represents a measure's value for the last iteration of the learning curve for each active learning method, while AULC is calculated by summing over all points of the learning curve, since we are using a discrete curve and all points are equally spaced in the x-axis (the number of iterations) for all learning curves.

3.3 Setup

As mentioned earlier, the multi-label active learning algorithms are instantiated with two functions:

1. a *scoring* function to evaluate object-label pairs; and
2. an *aggregating* function to aggregate these scores.

Table 8 Different combinations of multi-label active learning strategies considered in this work

Scoring function	*Aggregation* function
Conf	*Avg*
Rank	*Avg*
Score	*Avg*
Score	*Dev*

Table 9 Active learning settings used in experiments

	Experimental protocol	
	Separated BR SVMs	*Remaining* LPBHN + RCut
Random	✓	✓
Conf avg	✓	–
Rank avg	✓	–
Score avg	✓	✓
Score dev	✓	✓

Three strategies were considered for the *scoring* function:

- Confidence-based score (*conf*)
- raw score (*score*)
- Ranking-based score (*rank*)

For the *aggregation* function, two strategies were considered:

- Average (*avg*)
- Deviation (*dev*)

The other strategies were not considered since they exhibited inferior results in Cherman et al. (2016). In Table 8, we present the combinations of the *scoring* and *aggregating* functions employed throughout our experiments.

With respect to the multi-label learning algorithms used throughout the experiments, we employed both an inductive and a transductive algorithm. Specifically, the inductive algorithm, Binary Relevance with Linear SVMs as binary classifiers (BR SVMs), is used in experiments with the *separated* protocol. Regarding the *remaining* protocol, we chose to employ a multi-label classification algorithm with transductive inference, LPBHN with RCut. Even if both algorithms could be used for the *remaining* protocol, a transductive algorithm is better suited with the nature of that protocol, which calls for transductive inference.

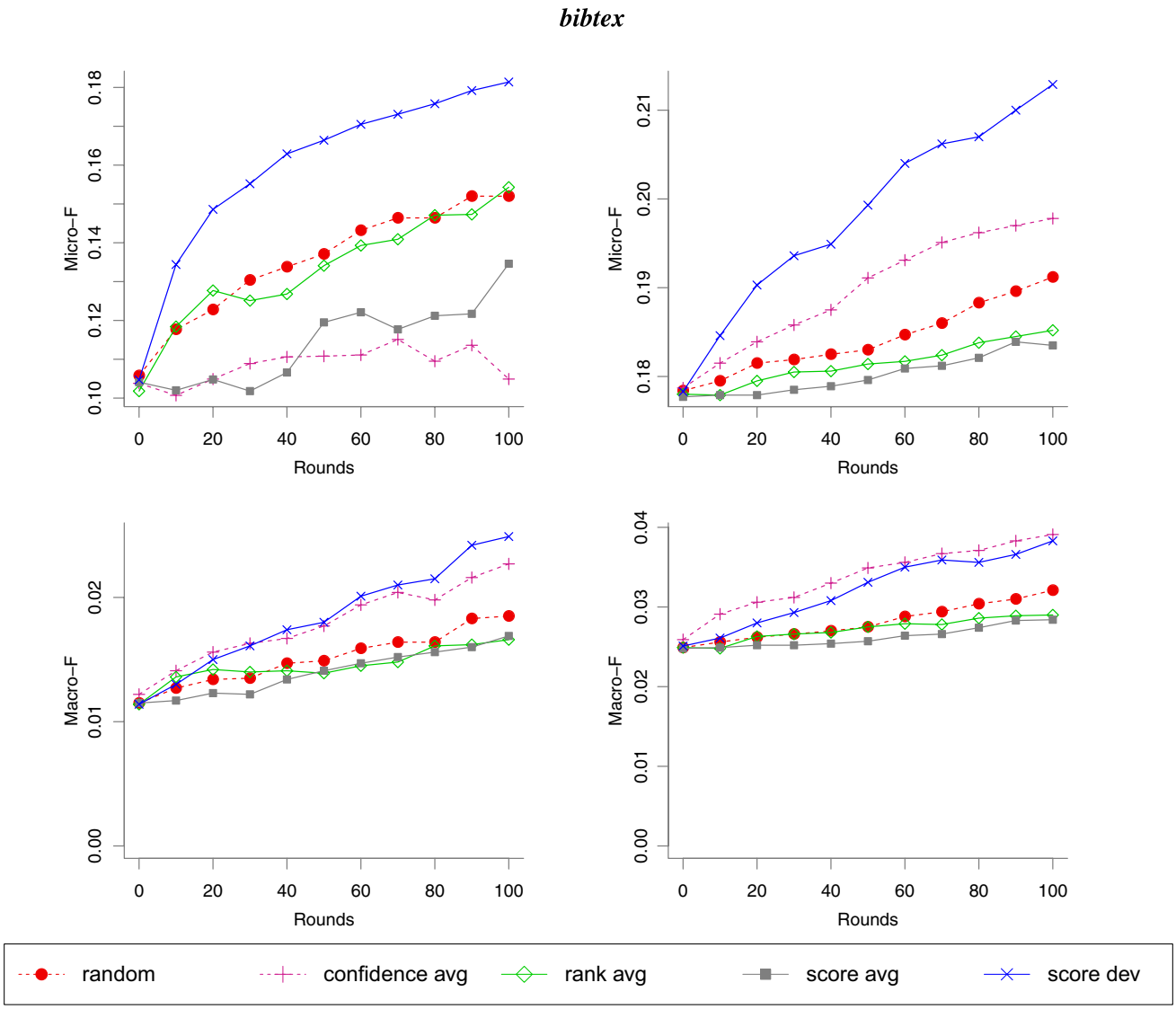

Fig. 1 Learning curves for BR-SVMS and the *separated* protocol on *bibtex*. The *left plots* are for $N_i ni = 1$ while the ones in the *right* are for $N_i ni = 5$

BR SVMs were implemented based on the LIBLINEAR library.[6] This implementation is optimized to handle efficiently and effectively sparse data, a crucial feature for active learning experiments due to the time cost involved in training a large number of models: we need to train one model for each label and for each iteration of the active learning procedure. In addition, the library can output normalized values of probability for the predictive confidence. In our experiments, we kept all parameters for the SVMs at default values, setting $C = 1$, $e = 0.01$ and employing the L2-loss SVC dual solver.

LPBHN was proposed by Rossi et al. (2013). The algorithm is based on graphs and more specifically on the Gaussian Fields and Harmonic Functions (GFHF) algorithm and is optimized for sparse data. Since the algorithm originally outputs a ranking of labels for each new instance to be predicted, we employed the *RCut* ranking strategy (Yang 2001), in order to apply a threshold. This method proceeds by choosing the t first labels of the ranking, with t being the closest integer to the training data set's cardinality (average number of labels of the examples). Table 9 presents the active learning settings used in our experiments and presented in this work.

All experiments were performed using tenfold cross validation. In the transductive context, where the *remaining*

[6] http://www.csie.ntu.edu.tw/~cjlin/liblinear/.

Table 10 *FV* represents the performance value for the last iteration of the learning curve for each active learning method

	$N_{ini} = 1$									
	Micro-F					Macro-F				
	Random	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*	*Random*	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*
#1 bibtex	.15 (2.5)	.10 (5.0)	.15 (2.5)	.13 (4.0)	**.18 (1.0)**	**.02 (3.0)**	**.02 (3.0)**	**.02 (3.0)**	**.02 (3.0)**	**.02 (3.0)**
#2 cal500	.34 (4.5)	.34 (4.5)	**.36 (2.0)**	**.36 (2.0)**	**.36 (2.0)**	.20 (3.0)	.19 (5.0)	**.21 (1.0)**	.20 (3.0)	.20 (3.0)
#3 corel16k	.03 (4.0)	.02 (5.0)	**.04 (2.0)**	**.04 (2.0)**	**.04 (2.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**
#4 corel5k	.04 (3.0)	.03 (5.0)	.04 (3.0)	**.05 (1.0)**	.04 (3.0)	**.31 (3.0)**	**.31 (3.0)**	**.31 (3.0)**	**.31 (3.0)**	**.31 (3.0)**
#5 emotions	.58 (4.0)	.60 (3.0)	.61 (2.0)	.57 (5.0)	**.64 (1.0)**	.54 (4.0)	.56 (3.0)	.58 (2.0)	.53 (5.0)	**.61 (1.0)**
#6 enron	.47 (4.0)	.40 (5.0)	.48 (3.0)	**.52 (1.0)**	.49 (2.0)	.26 (3.0)	.25 (4.5)	**.27 (1.5)**	**.27 (1.5)**	.25 (4.5)
#7 llog	**.03 (2.0)**	**.03 (2.0)**	.02 (4.0)	.01 (5.0)	**.03 (2.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**
#8 medical	.55 (5.0)	.57 (4.0)	.60 (2.5)	**.61 (1.0)**	.60 (2.5)	.56 (5.0)	.58 (2.5)	.58 (2.5)	**.60 (1.0)**	.57 (4.0)
#9 ohsumed	.12 (4.0)	.11 (5.0)	.15 (2.5)	**.24 (1.0)**	.15 (2.5)	.04 (3.5)	.03 (5.0)	.05 (2.0)	**.08 (1.0)**	.04 (3.5)
#10 scene	.55 (4.0)	.43 (5.0)	**.58 (1.0)**	.57 (2.5)	.57 (2.5)	.54 (4.0)	.39 (5.0)	**.58 (1.5)**	.57 (3.0)	**.58 (1.5)**
#11 slashdot	.09 (4.0)	**.23 (1.0)**	.12 (3.0)	.06 (5.0)	.16 (2.0)	.18 (4.5)	**.21 (1.0)**	.19 (3.0)	.18 (4.5)	.20 (2.0)
#12 tmc2007	.51 (3.0)	.47 (5.0)	.50 (4.0)	**.56 (1.5)**	**.56 (1.5)**	.19 (2.5)	.13 (5.0)	.18 (4.0)	**.31 (1.0)**	.19 (2.5)
#13 yeast	.58 (3.0)	.56 (5.0)	**.60 (1.0)**	.58 (3.0)	.58 (3.0)	.32 (3.0)	.28 (5.0)	.33 (2.0)	**.34 (1.0)**	.30 (4.0)
avg ranking	3.6	4.2	2.5	2.6	2.1	3.4	3.7	2.4	2.5	2.9
better/equal random	–	38%	85%	69%	100%	–	54%	92%	92%	85%
	$N_{ini} = 5$									
	Micro-F					Macro-F				
	Random	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*	*Random*	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*
#1 bibtex	.19 (3.5)	.20 (2.0)	.19 (3.5)	.18 (5.0)	**.21 (1.0)**	.03 (4.0)	**.04 (1.5)**	.03 (4.0)	.03 (4.0)	**.04 (1.5)**
#2 cal500	.34 (4.0)	.33 (5.0)	.35 (3.0)	**.36 (1.5)**	**.36 (1.5)**	.19 (4.5)	.19 (4.5)	**.20 (2.0)**	**.20 (2.0)**	**.20 (2.0)**
#3 corel16k	.06 (3.0)	.05 (5.0)	.06 (3.0)	**.07 (1.0)**	.06 (3.0)	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**
#4 corel5k	.08 (3.5)	.08 (3.5)	.08 (3.5)	**.09 (1.0)**	.08 (3.5)	**.32 (3.0)**	**.32 (3.0)**	**.32 (3.0)**	**.32 (3.0)**	**.32 (3.0)**
#5 emotions	.59 (4.0)	.62 (2.0)	.58 (5.0)	.60 (3.0)	**.64 (1.0)**	.54 (5.0)	.60 (2.0)	.55 (4.0)	.57 (3.0)	**.62 (1.0)**
#6 enron	.49 (4.0)	.48 (5.0)	.51 (2.5)	**.53 (1.0)**	.51 (2.5)	.28 (3.0)	.28 (3.0)	.28 (3.0)	**.29 (1.0)**	.27 (5.0)
#7 llog	.06 (3.5)	.07 (2.0)	.06 (3.5)	.04 (5.0)	**.09 (1.0)**	.39 (3.5)	.39 (3.5)	.39 (3.5)	.39 (3.5)	**.40 (1.0)**
#8 medical	.68 (5.0)	.70 (3.0)	.69 (4.0)	**.72 (1.5)**	**.72 (1.5)**	.63 (4.0)	.64 (2.0)	.63 (4.0)	**.65 (1.0)**	.63 (4.0)
#9 ohsumed	.19 (2.0)	.15 (4.5)	.15 (4.5)	**.23 (1.0)**	.18 (3.0)	.06 (2.5)	.05 (4.5)	.05 (4.5)	**.08 (1.0)**	.06 (2.5)
#10 scene	.57 (4.0)	.49 (5.0)	.60 (2.0)	.61 (1.0)	.58 (3.0)	.56 (4.0)	.47 (5.0)	**.61 (1.5)**	**.61 (1.5)**	.59 (3.0)
#11 slashdot	.19 (4.0)	.26 (2.0)	.23 (3.0)	.12 (5.0)	**.28 (1.0)**	.22 (4.0)	.23 (2.0)	**.23 (2.0)**	.20 (5.0)	**.23 (2.0)**
#12 tmc2007	.54 (3.5)	.53 (5.0)	**.56 (1.5)**	**.56 (1.5)**	.54 (3.5)	.22 (4.0)	.19 (5.0)	.24 (2.0)	**.28 (1.0)**	.23 (3.0)
#13 yeast	.59 (4.0)	.59 (4.0)	**.60 (1.5)**	**.60 (1.5)**	.59 (4.0)	.31 (4.5)	.32 (3.0)	**.34 (1.5)**	**.34 (1.5)**	.31 (4.5)
avg ranking	3.7	3.7	3.1	2.2	2.3	3.8	3.2	2.9	2.3	2.7
better/equal random	–	54%	85%	77%	92%	–	77%	92%	92%	92%

The values in parentheses refer to the ranking of the method. Experimental protocol: *separated*

protocol is used, the independent test partitions of each of the ten folds are discarded, since the remaining examples of the query set are used to evaluate the predictive performance.

Finally, the initial labeled pool of examples was built by randomly choosing examples until having $N_{ini} \times q$ positive single labels, *i.e.*. until $N_{ini} \times q \geq \sum_{i=1}^{|D_l|} Y_i$, where N_{ini} is user-defined. This strategy allows for fairer comparison across the data sets. We used $N_{ini} = 1, 5$ in order to evaluate the influence of different sizes of the initial labeled pool.

3.4 Results

In this section we present the results of our experiments. We show the learning curves and the results in terms of Micro-F and Macro-F (for the *separated* protocol) and Ranking Loss (for the *remaining* protocol) for all data sets, using the *FV* and the *AULC* measures. The *random* method represents the *baseline* (passive learning) strategy and the *score dev* method refers to our proposed approach, while the rest of

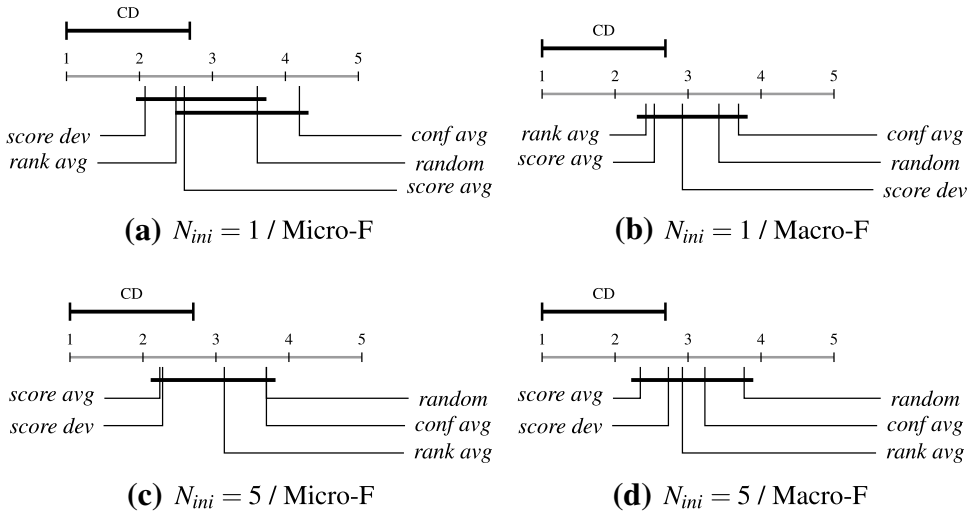

Fig. 2 Friedman ranking with Nemenyi post-test for the BR SVMs and *FV* measure. Experimental protocol: *separated*

the methods refer to the ones previously proposed in the literature.

3.4.1 Experiments with the separated test protocol: inductive inference

Figure 1 present the learning curves for the algorithms considered in this work using the *bibtex* data set. The learning curves for the rest of the data sets are presented in an on-line appendix.[7]

From the plots, we can easily observe that our proposed method *score dev* is consistently outperforming conventional supervised learning (*random*) for all four considered scenarios. The *score avg* method shows also a steady advantage compared to *random* in all scenarios, except for $N_{ini} = 1$ and for the *Micro-F* measure, in which case it performs worse than all other methods. The other active learning methods show mixed results, not being able to exhibit a steady advantage compared to the *random* method in any scenario.

Table 10 shows the results for all data sets and the *separated* protocol, in terms of Micro-f and Macro-F. As we can see, several of the active learning methods have *FV* values equal to or greater than those presented by the *random* method. The *conf avg* method, however, is inferior to *random* for $N_{ini} = 1$ for both evaluation measures (*Micro-F / Macro-F*) and for $N_{ini} = 5$ for *Micro-F*. The other active learning methods have *average ranking* values higher than the one for *random* in all cases.

An important aspect to be evaluated in an active learning method is its consistency in outperforming the *random* method, since, unlike the evaluation of standard learning algorithms, one does not have the data labeled beforehand, and the purpose of active learning is to obtain good labeled examples. Thus, it is not possible to predict the effectiveness of various active learning methods beforehand in a way similar to the one followed for standard supervised learning algorithms. Thus, in an effort to measure the stability of the active learning method, the *better or equal to random* value is displayed in the last line of Table 10. This value refers to the percentage of data set where the active method was greater than or equal to the *random* method.

score dev presents the best stability values considering Micro-F as the evaluation measure. This method obtained results better than or equal to *random* for 100 and 92% of the data sets for $N_{ini} = 1$ and $N_{ini} = 5$, respectively. *score dev*, along with *rank avg* and *conf avg*, also presented the best stability value for *Macro-F* and $N_{ini} = 5$, with 92% of cases better than or equal to *random*. In the scenario with *Macro-F* and $N_{ini} = 1$, *rank avg* and *score avg* presented the best stability values with 92%, followed by the *score dev* method with 85%.

In summary, *conf avg* did not perform in a satisfying manner regarding stability for any scenario; The *rank avg* method showed the best stability value in 2 / 4 out of scenarios, the *score avg* method also had the best stability value in 2 / 4 out of scenarios and the *score dev* method, our proposed method, exhibited the best result for 3 / 4 out of the cases, that is, presented the best stability ratio.

Figure 2 presents the average ranking plotted against the *FV* measure of each Friedman test with a Nemenyi post hoc test with a significance level of 95% to identify statistically significant differences between the methods (Demšar, 2006).

Although there are consistent differences between the active learning methods and the *baseline random*, no active method showed improvement with a statistically significant difference compared to the *random* method. The only difference observed is between *score dev* and *conf avg* for $N_{ini} = 1$ and *Micro-F1*, which again indicates the difficulty of *conf avg* in obtaining satisfying results.

[7] https://www.dropbox.com/s/cxyf27wzp9xzlxr/appendix.pdf?dl=0.

Table 11 *AULC* values for each active learning method using the *separated* experimental protocol

	$N_{ini} = 1$									
	Micro-F					Macro-F				
	Random	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*	*Random*	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*
#1 bibtex	.14 (2.5)	.12 (4.5)	.14 (2.5)	.12 (4.5)	**.17 (1.0)**	**.02 (3.0)**	**.02 (3.0)**	**.02 (3.0)**	**.02 (3.0)**	**.02 (3.0)**
#2 cal500	.35 (5.0)	.36 (3.0)	.36 (3.0)	**.37 (1.0)**	.36 (3.0)	.20 (4.5)	.20 (4.5)	**.21 (2.0)**	**.21 (2.0)**	**.21 (2.0)**
#3 corel16k	**.03 (2.5)**	.02 (5.0)	**.03 (2.5)**	**.03 (2.5)**	**.03 (2.5)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**
#4 corel5k	.03 (4.0)	.03 (4.0)	**.04 (1.5)**	**.04 (1.5)**	.03 (4.0)	**.31 (3.0)**	**.31 (3.0)**	**.31 (3.0)**	**.31 (3.0)**	**.31 (3.0)**
#5 emotions	.54 (3.5)	.54 (3.5)	**.57 (1.5)**	.52 (5.0)	**.57 (1.5)**	.49 (3.0)	.47 (4.5)	**.52 (1.5)**	.47 (4.5)	**.52 (1.5)**
#6 enron	.43 (4.0)	.32 (5.0)	.45 (2.0)	**.49 (1.0)**	.44 (3.0)	.25 (3.0)	.23 (5.0)	.25 (3.0)	**.26 (1.0)**	.25 (3.0)
#7 llog	.03 (3.0)	**.04 (1.5)**	.02 (4.5)	.02 (4.5)	**.04 (1.5)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**
#8 medical	.47 (4.0)	.45 (5.0)	.51 (2.5)	**.53 (1.0)**	.51 (2.5)	.54 (4.5)	.54 (4.5)	.56 (2.0)	**.57 (1.0)**	.55 (3.0)
#9 ohsumed	.09 (4.0)	.07 (5.0)	.10 (3.0)	**.17 (1.0)**	.14 (2.0)	.03 (4.0)	.02 (5.0)	.04 (2.5)	**.05 (1.0)**	.04 (2.5)
#10 scene	.46 (3.5)	.38 (5.0)	.48 (2.0)	**.49 (1.0)**	.46 (3.5)	.44 (3.5)	.34 (5.0)	**.47 (1.0)**	.46 (2.0)	.44 (3.5)
#11 slashdot	.06 (4.0)	**.18 (1.0)**	.07 (3.0)	.05 (5.0)	.13 (2.0)	.18 (4.0)	**.20 (1.0)**	.18 (4.0)	.18 (4.0)	.19 (2.0)
#12 tmc2007	.46 (3.0)	.42 (5.0)	.43 (4.0)	**.50 (1.5)**	**.50 (1.5)**	.15 (2.5)	.11 (5.0)	.13 (4.0)	**.23 (1.0)**	.15 (2.5)
#13 yeast	.56 (3.5)	.53 (5.0)	**.58 (1.0)**	.56 (3.5)	.57 (2.0)	.29 (3.5)	.26 (5.0)	.30 (2.0)	**.31 (1.0)**	.29 (3.5)
Avg ranking	3.4	4.0	2.6	2.3	2.7	3.3	3.2	3.0	2.5	3.0
Better/equal random	–	38%	85%	69%	100%	–	54%	92%	92%	100%
	$N_{ini} = 5$									
	Micro-F					Macro-F				
	Random	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*	*Random*	*Conf avg*	*Rank avg*	*Score avg*	*Score dev*
#1 bibtex	.19 (3.0)	.19 (3.0)	.19 (3.0)	.18 (5.0)	**.20 (1.0)**	.03 (3.5)	**.04 (1.0)**	.03 (3.5)	.03 (3.5)	.03 (3.5)
#2 cal500	.34 (4.5)	.34 (4.5)	.35 (2.5)	**.36 (1.0)**	.35 (2.5)	.19 (4.5)	.19 (4.5)	**.20 (2.0)**	**.20 (2.0)**	**.20 (2.0)**
#3 corel16k	**.06 (2.5)**	.05 (5.0)	**.06 (2.5)**	**.06 (2.5)**	**.06 (2.5)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**	**.01 (3.0)**
#4 corel5k	**.08 (3.0)**	**.08 (3.0)**	**.08 (3.0)**	**.08 (3.0)**	**.08 (3.0)**	**.32 (3.0)**	**.32 (3.0)**	**.32 (3.0)**	**.32 (3.0)**	**.32 (3.0)**
#5 emotions	.56 (3.5)	**.57 (1.5)**	.55 (5.0)	.56 (3.5)	**.57 (1.5)**	.52 (3.5)	.54 (2.0)	.51 (5.0)	.52 (3.5)	**.55 (1.0)**
#6 enron	.48 (4.5)	.48 (4.5)	.49 (2.5)	**.51 (1.0)**	.49 (2.5)	.27 (3.5)	.27 (3.5)	.27 (3.5)	**.28 (1.0)**	.27 (3.5)
#7 llog	.06 (3.0)	.06 (3.0)	.06 (3.0)	.05 (5.0)	**.07 (1.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**	**.39 (3.0)**
#8 medical	.67 (4.5)	.68 (2.5)	.67 (4.5)	**.69 (1.0)**	.68 (2.5)	.62 (3.5)	.62 (3.5)	.62 (3.5)	**.63 (1.0)**	.62 (3.5)
#9 ohsumed	.15 (3.0)	.12 (5.0)	.13 (4.0)	**.18 (1.0)**	.16 (2.0)	.05 (3.5)	.05 (3.5)	.05 (3.5)	**.06 (1.0)**	.05 (3.5)
#10 scene	.53 (3.0)	.46 (5.0)	**.55 (1.5)**	**.55 (1.5)**	.52 (4.0)	.52 (3.0)	.45 (5.0)	**.55 (1.0)**	.53 (2.0)	.51 (4.0)
#11 slashdot	.15 (4.0)	**.22 (1.5)**	.18 (3.0)	.11 (5.0)	**.22 (1.5)**	.21 (3.5)	**.22 (1.5)**	.21 (3.5)	.20 (5.0)	**.22 (1.5)**
#12 tmc2007	.52 (2.0)	.51 (4.0)	.51 (4.0)	**.53 (1.0)**	.51 (4.0)	.20 (2.0)	.17 (5.0)	.19 (3.5)	**.22 (1.0)**	.19 (3.5)
#13 yeast	.58 (4.5)	**.59 (2.0)**	**.59 (2.0)**	**.59 (2.0)**	.58 (4.5)	.31 (3.5)	.31 (3.5)	**.32 (1.0)**	.31 (3.5)	.31 (3.5)
Avg ranking	3.6	4.0	2.5	2.5	2.3	3.5	3.4	3.1	2.5	2.5
Better/equal random	–	69%	77%	77%	85%	–	85%	85%	92%	85%

The values in parentheses refer to the rankingposition of the method

In Table 11, we present the results for *AULC* as the evaluation method.

rank avg, *score avg* and *score dev* show better average rankings than *random* for all setups and evaluation measures, whereas *conf avg* has worse average ranking than *random* for Micro-F for both N_{ini} parameter choices.

Regarding stability (*equal or better* than *random*), we observe a similar behavior to the one for the *FV* measure. For $N_{ini} = 1$, the *score dev* method was the only one to be better or equal to *random* in all data sets, for both *Micro-F* and *Macro-F*. For the scenario with $N_{ini} = 5$ and *Micro-F*, the *score dev* method showed the best stability (85%). The *score avg* method was higher for the scenario with $N_{ini} = 5$ and *Macro-F*, where it was better than or equal to the *random* method in 92% of the data sets. The other three methods showed a stability of 85%.

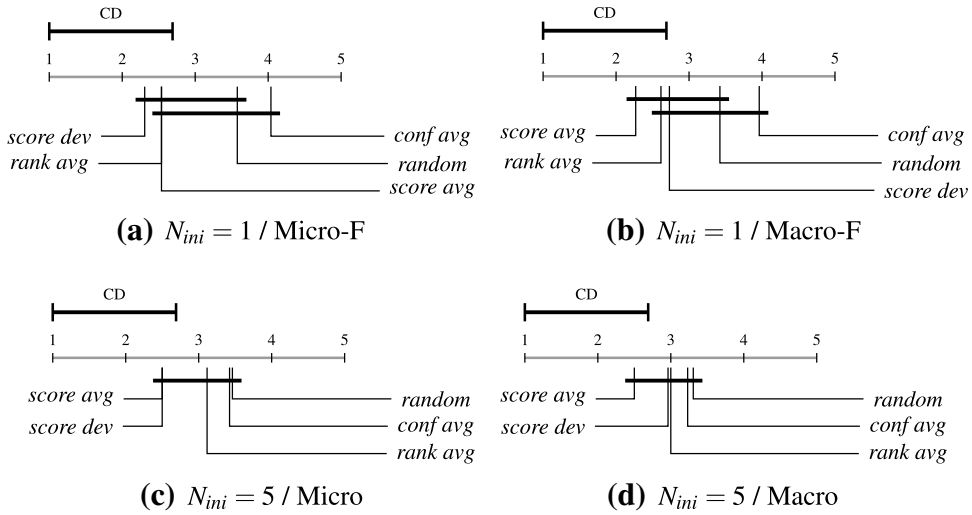

Fig. 3 Friedman ranking with Nemenyi as post-test for the BR-SVMs and the *AULC* measure. The experimental protocol *separated* is followed

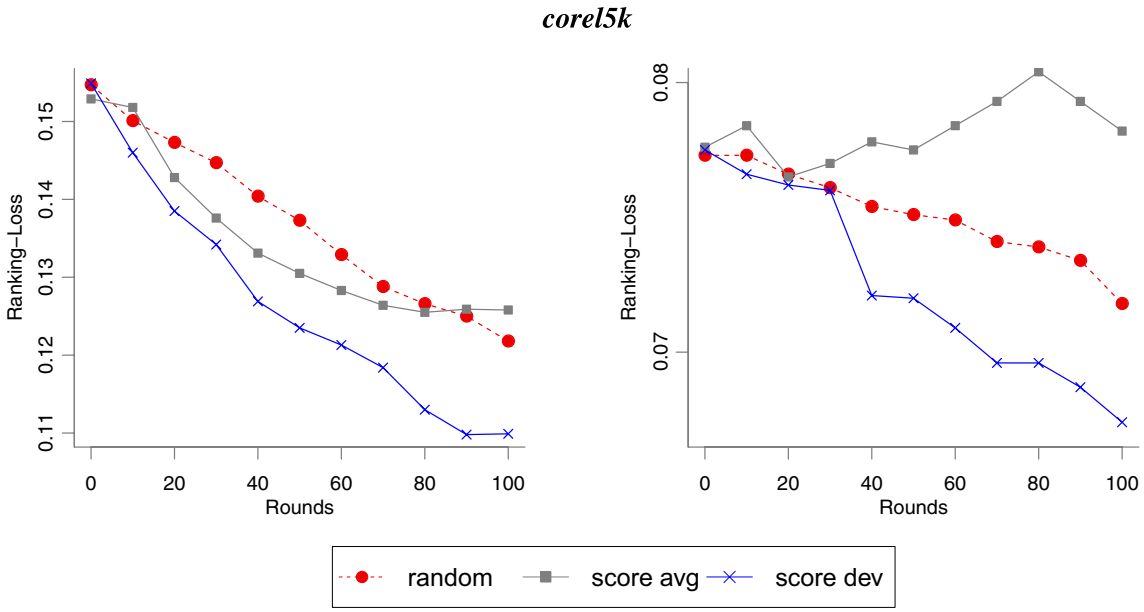

Fig. 4 Learning algorithm: *LPBHN-Rcut*. Experimental protocol: *separated*. Data set: *corel5k*

Figure 3 shows the average ranking plotted for the *AULC* measure of each method. Again, no active method showed improvement with statistically significant difference in relation to *random*. Significant differences considering *AULC* as evaluation measure were found only between *score dev* and *conf avg* in terms of *Micro-F* for $N_{ini} = 1$/ and between *score avg* and *conf avg* in terms of *Macro-F* again for $N_{ini} = 1$.

3.4.2 *Experiments with the remaining test protocol: transductive inference*

The experiments performed using the *remaining* protocol simulate applications that are intended to annotate examples in the context of transductive inference, i.e. applications in which the test data is observed *a priori*. Inductive inference methods, such as BR-SVMs method, could also be used in this context. However, inductive inference is intended to solve a more general problem than what is necessary in that case. Also, we should note that there are cases in which transductive inference may be more effective, such as a scenario with extremely few labeled examples, with all unlabeled data available beforehand.[8]

[8] https://en.wikipedia.org/wiki/Transduction_(machine_learning).

Table 12 Results for the *FV* measure in the last iteration of the learning curve for each active learning method with the *remaining* protocol

	Ranking-loss					
	$N_{ini} = 1$			$N_{ini} = 5$		
	Random	*Score avg*	*Score dev*	*Random*	*Score avg*	*Score dev*
bibtex	0.10 (2.0)	0.16 (3.0)	**0.09 (1.0)**	**0.02 (1.5)**	**0.02 (1.5)**	0.03 (3.0)
cal500	0.17 (3.0)	**0.15 (1.5)**	**0.15 (1.5)**	0.17 (3.0)	**0.16 (1.5)**	**0.16 (1.5)**
corel16k	0.14 (2.5)	0.14 (2.5)	**0.13 (1.0)**	0.10 (2.5)	0.10 (2.5)	**0.09 (1.0)**
corel5k	0.12 (2.0)	0.13 (3.0)	**0.11 (1.0)**	**0.07 (1.5)**	0.08 (3.0)	**0.07 (1.5)**
emotions	0.06 (2.5)	0.06 (2.5)	**0.03 (1.0)**	0.06 (2.5)	0.06 (2.5)	**0.01 (1.0)**
enron	0.15 (3.0)	**0.13 (1.5)**	**0.13 (1.5)**	0.14 (3.0)	**0.12 (1.0)**	0.13 (2.0)
llog	**0.09 (2.0)**	**0.09 (2.0)**	**0.09 (2.0)**	0.06 (3.0)	**0.05 (1.5)**	**0.05 (1.5)**
medical	0.05 (2.0)	0.06 (3.0)	**0.04 (1.0)**	**0.04 (2.0)**	**0.04 (2.0)**	**0.04 (2.0)**
ohsumed	0.15 (3.0)	**0.12 (1.5)**	**0.12 (1.5)**	0.13 (3.0)	**0.10 (1.5)**	**0.10 (1.5)**
scene	**0.03 (1.0)**	0.16 (2.5)	0.16 (2.5)	0.02 (2.0)	0.03 (3.0)	**0.01 (1.0)**
slashdot	0.05 (2.0)	0.09 (3.0)	**0.04 (1.0)**	**0.04 (1.5)**	0.05 (3.0)	**0.04 (1.5)**
tmc2007	0.14 (3.0)	**0.11 (1.5)**	**0.11 (1.5)**	0.14 (3.0)	**0.12 (1.5)**	**0.12 (1.5)**
yeast	**0.06 (1.0)**	0.07 (2.5)	0.07 (2.5)	0.07 (2.0)	**0.06 (1.0)**	0.08 (3.0)
avg ranking	2.2	2.3	1.5	2.3	2.0	1.7
better/equal random	–	54%	85%	–	77%	85%

Values in parentheses refer to the method's ranking position

Table 13 Results for the *AULC* measure in the last iteration of the learning curve for each active learning method with the *remaining* protocol

	Ranking-loss					
	$N_{ini} = 1$			$N_{ini} = 5$		
	Random	*Score avg*	*Score dev*	*Random*	*Score avg*	*Score dev*
bibtex	1.73 (2.0)	1.95 (3.0)	**1.70 (1.0)**	**0.28 (1.5)**	**0.28 (1.5)**	0.29 (3.0)
cal500	1.90 (3.0)	1.81 (2.0)	**1.80 (1.0)**	1.75 (3.0)	**1.69 (1.5)**	**1.69 (1.5)**
corel16k	1.86 (2.0)	1.89 (3.0)	**1.78 (1.0)**	1.04 (3.0)	1.03 (2.0)	**0.98 (1.0)**
corel5k	1.40 (3.0)	1.37 (2.0)	**1.31 (1.0)**	0.76 (2.0)	0.79 (3.0)	**0.73 (1.0)**
emotions	**1.05 (1.0)**	1.32 (3.0)	1.28 (2.0)	0.72 (2.0)	0.83 (3.0)	**0.39 (1.0)**
enron	1.59 (2.5)	1.59 (2.5)	**1.51 (1.0)**	1.44 (3.0)	**1.30 (1.0)**	1.32 (2.0)
llog	1.18 (3.0)	1.17 (2.0)	**1.09 (1.0)**	0.64 (3.0)	0.61 (2.0)	**0.59 (1.0)**
medical	0.98 (3.0)	0.96 (2.0)	**0.87 (1.0)**	0.48 (3.0)	0.44 (2.0)	**0.42 (1.0)**
ohsumed	2.11 (3.0)	1.77 (2.0)	**1.75 (1.0)**	1.49 (3.0)	**1.31 (1.0)**	1.32 (2.0)
scene	**0.87 (1.0)**	1.94 (3.0)	1.92 (2.0)	0.39 (2.0)	0.40 (3.0)	**0.28 (1.0)**
slashdot	0.86 (2.0)	1.20 (3.0)	**0.59 (1.0)**	0.48 (2.0)	0.50 (3.0)	**0.46 (1.0)**
tmc2007	1.67 (3.0)	1.41 (2.0)	**1.39 (1.0)**	1.52 (3.0)	1.31 (2.0)	**1.29 (1.0)**
yeast	0.84 (3.0)	0.82 (2.0)	**0.72 (1.0)**	**0.80 (1.0)**	0.83 (2.0)	0.89 (3.0)
avg ranking	2.4	2.4	1.2	2.4	2.1	1.5
better/equal random	–	62%	85%	–	62%	85%

Values in parentheses refer to the method's ranking position

In this round of experiments we employed the LPBHN algorithm, a transductive inference multi-label algorithm that outputs ranking of labels, additionally using the RCut method to apply a threshold on the ranking and obtain a hard assignment of labels for each test instance.

The Ranking Loss measure was used to evaluate the quality of predictions since LPBHN outputs rankings. As mentioned earlier in the paper, we also used the RCut method in this experimental setup, in order to obtain a hard assignment of labels from the rankings. We remind here that RCut selects the t first labels from the ranking, with t being the nearest integer to the training set cardinality. Since in the active learning setting the number of labeled examples

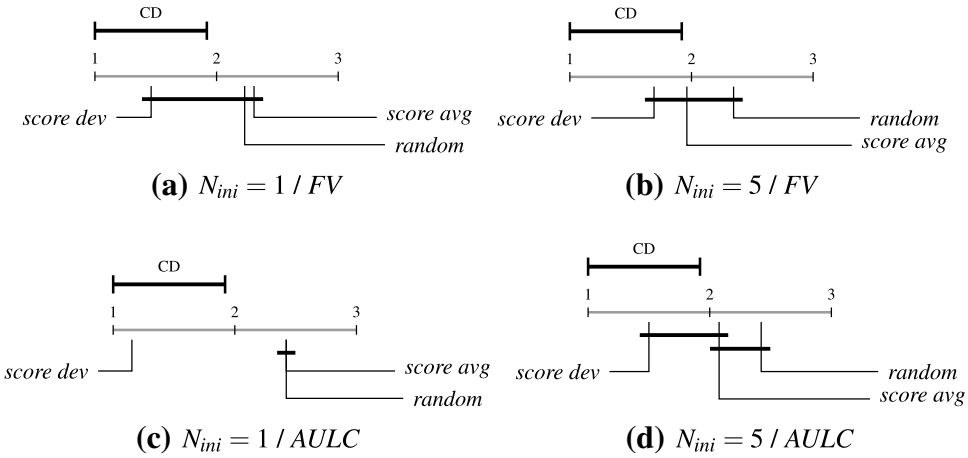

Fig. 5 Friedman test with a Nemenyi post-test for the LPBHN classifier and for the *AULC* and *FV* measures. The *remaining* protocol was used.

is reduced, we expect that the estimation of the cardinality using the training set could be less precise in our case.

Figure 4 presents the learning curve for the corel5k data set. The figures referring to the learning curves of other data sets are, similar to the plots for the *separated* protocol, presented in an on-line appendix.[9] from the plots we can observe that *score dev* is consistently superior to both *random* and *score avg* for the two considered scenarios. *score avg* on the other hand, fails to consistently outperform the passive method, especially for $N_{ini} = 5$.

Tables 12 and 13 presents results for the *remaining* protocol for all data sets concerning learning evaluation measures *FV* and *AULC*,[10] respectively. *score dev* shows the best average ranking and the best stability (better or equal than *random*) for all scenarios (85%). furthermore, *score avg* has lower stability for the *remaining* protocol than the one exhibited for the *separated* protocol.

Finally, Fig. 5 depicts the average ranking of each method together with the performed statistical significance test. Again, *score dev* shows the best values of average ranking for all scenarios, with a statistically significant difference to *random* when considering *AULC* as the evaluation measure. Additionally, we can observe a statistically significant difference between *score dev* and *score avg* when considering *AULC* for $N_{ini} = 1$.

4 Conclusions

Dealing with learning tasks in constantly changing environments, requires approaches that, far from relying on static and passive learning models, enable effective evolving of the learning system when prompted with new data. Active learning is such an approach, enabling a given classifier to actively choose which of the new data will be manually annotated for training. In this manner, apart from reducing annotation costs and requiring fewer training examples, the classifier is capable of evolving to better represent new data.

Although active learning for single-label learning has been a well investigated topic of research, this is not the case for multi-label learning. In this work, we discussed key issues in pool-based multi-label active learning based on previous work in the literature. We presented the main approaches regarding the scoring and aggregation strategies of multi-label active learning and proposed a novel aggregation approach, called *score dev*, We implemented all previously existing approaches, as well as our method in a common framework and performed extensive experimental comparisons for two different multi-label learning algorithms, on thirteen multi-label data sets and under two different application settings (transductive, inductive).

The results on two different evaluation protocols, an inductive and a transductive learning scenario with BR-SVMs and LPBHN as base classifiers respectively, show a consistent advantage of our aggregation method *dev* with *score* as the evaluation approach, compared to the rest of the methods and to conventional passive learning, followed by the average aggregating strategy, again with *score* for evaluation. It should be noted that the raw score used by *dev* is advantageous since one does not need to normalize or define thresholds to calculate it. *rank* and *conf* strategies, on the other hand, obtain their results using manipulated or normalized (by the base classifiers) scores, which make them more base classifier dependent.

Acknowledgements We would like to thank the anonymous reviewers for their constructive comments that helped in improving our paper. E.A. Cherman and M.C. Monard were supported by the São Paulo Research Foundation (FAPESP), Grants 2010/15992-0 and

[9] https://www.dropbox.com/s/cxyf27wzp9xzlxr/appendix.pdf?dl=0.

[10] *AULC* values for the *Ranking-Loss* measure were multiplied by 10 to consider the third decimal place in the comparison.

2011/21723-5, and Brazilian National Council for Scientific and Technological Development (CNPq), Grant 644963.

References

Aggarwal CC, Kong X, Gu Q, Han J, Yu PS (2014) Active learning: a survey. In: Aggarwal CC (ed) Data classification: algorithms and applications. CRC Press, Boca Raton, pp 571–606

Brinker K (2006) On active learning in multi-label classification. In: Spiliopoulou M, Kruse R, Borgelt C, Nurnberger A, Gaul W (eds) From data and information analysis to knowledge engineering, studies in classification, data analysis, and knowledge organization. Springer, Berlin, pp 206–213

Cherman EA, Tsoumakas G, Monard MC (2016) Active learning algorithms for multi-label data. In: Proceedings of the 12th IFIP international conference on artificial intelligence applications and innovations (AIAI 2016), Thessaloniki, pp 1–12

Demšar J (2006) Statistical comparison of classifiers over multiple data sets. J Mach Learn Res 7(1):1–30

Esuli A, Sebastiani F (2009) Active learning strategies for multi-label text classification. In: Proceedings of the 31st European conference on IR research, ECIR '09. Springer, Berlin, pp 102–113

Gao N, Huang SJ, Chen S (2016) Multi-label active learning by model guided distribution matching. Front Comput Sci 10(5):845–855

Huang S, Chen S, Zhou Z (2015) Multi-label active learning: query type matters. In: Proceedings of the twenty-fourth international joint conference on artificial intelligence, IJCAI 2015, pp 946–952

Hung CW, Lin HT (2011) Multi-label active learning with auxiliary learner. In: Asian conference on machine learning, pp 315–332

McCallumzy AK, Nigamy K (1998) Employing EM and pool-based active learning for text classification. In: Proceedings of the international conference on machine learning (ICML), Citeseer, pp 359–367

Nowak S, Nagel K, Liebetrau J (2011) The CLEF 2011 photo annotation and concept-based retrieval tasks. In: CLEF (notebook papers/labs/workshop), Amsterdam, Netherlands, pp 1–25

Rossi RG, de Andrade Lopes A, Rezende SO (2013) A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. In: Proceedings of symposium on applied computing (ACM SAC'2014), New York, NY

Settles B (2010) Active learning literature survey. Tech. Rep. 1648. University of Wisconsin–Madison, Madison

Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 1070–1079

Singh M, Brew A, Greene D, Cunningham P (2010) Score normalization and aggregation for active learning in multi-label classification. Tech. rep. University College Dublin, Dublin

Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. J Mach Learn 2:45–66

Tsoumakas G, Katakis I, Vlahavas I (2009) Mining multi-label data. Data mining and knowledge discovery handbook, Springer, pp 1–19

Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I (2011) Mulan: a java library for multi-label learning. J Mach Learn Res 12:2411–2414

Tsoumakas G, Zhang ML, Zhou ZH (2012) Introduction to the special issue on learning from multi-label data. Mach Learn 88(1–2):1–4

Yang B, Sun JT, Wang T, Chen Z (2009) Effective multi-label active learning for text classification. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '09, ACM, New York, pp 917–926. doi:10.1145/1557019.1557119

Yang Y (2001) A study of thresholding strategies for text categorization. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval, ACM, New York, NY, pp 137–145

Ye C, Wu J, Sheng VS, Zhao S, Zhao P, Cui Z (2015) Multi-label active learning with chi-square statistics for image classification. In: Proceedings of the 5th ACM on international conference on multimedia retrieval—ICMR'15, Association for Computing Machinery (ACM), New York, NY, pp 583–586

Zhang B, Wang Y, Chen F (2014) Multilabel image classification via high-order label correlation driven active learning. IEEE Trans Image Process 23(3):1430–1441

Zliobaite I, Bifet A, Pfahringer B, Holmes G (2011) Active learning with evolving streaming data. In: Joint European conference on machine learning and knowledge discovery in databases, Springer, Berlin, pp 597–612