Active learning algorithms for multi-label data

Everton Alvares Cherman
University of Sao Paulo (USP)
P.O. Box 668, Zip code 13561-970
Sao Carlos - SP, Brazil
Tel.: +55-16-3373-9700
Fax: +55-16-3371-2238
Email: evertoncherman@gmail.com

Grigorios Tsoumakas
Department of Informatics
Aristotle University of Thessaloniki (AUTh)
Thessaloniki, Greece
Email: greg@csd.auth.gr

Maria Carolina Monard
University of Sao Paulo (USP)
Sao Carlos - SP, Brazil
E-mail: mcmonard@icmc.usp.br

***Abstract*—The iterative supervised learning setting, in which learning algorithms can actively query an oracle for labels, e.g. a human annotator that understands the nature of the problem, is called active learning. As the learner is allowed to interactively choose the data from which it learns, it is expected that the learner would perform better with less training. The active learning approach is appropriate to machine learning applications where training labels are costly to obtain but unlabeled data is abundant. Although active learning has been widely considered for single-label learning, this is not the case for multi-label learning, in which objects can have more than one class label and a multi-label learner is trained to assign multiple labels simultaneously to an object. There are different scenarios to query the annotator. This work focuses on the scenario in which the evaluation of unlabeled data is taken into account to select the object to be labeled. In this scenario, several multi-label active learning algorithms were identified in the literature. These algorithms were implemented in a common framework and experimentally evaluated in two multi-label datasets which have different properties. The influence of the properties of the datasets in the results obtained by the multi-label active learning algoritm is highlighted.**

I. Introduction

Different approaches to enhance supervised learning have been proposed over the years. As supervised learning algorithms build classifiers based on labelled training examples, several of these approaches aim to reduce the amount of time and effort needed to obtain labeled data for training. Active learning is one of these approaches. The key idea of active learning is to minimize labeling costs by allowing the learner to query for the labels of the most informative unlabeled data instances. These queries are posed to an oracle, e.g. a human annotator, which understands the nature of the problem. This way, an active learner can substantially reduce the number of labeled data required to construct the classifier.

Active learning has been widely considered to support single-label learning, in which each object (instance) in the dataset is associated with only one class label. However, this is not the case in multi-label learning, where each object is associated with a subset of labels. Due to the large number of real-world problems which fall into this category, the problem of multi-label classification has attracted great interest in the last decade.

There are different active learning scenarios to query the annotator. The focus of this work is on the scenario where the evaluation of unlabeled data is taken into account to select the objects to be labeled. In this scenario, several multi-label active learning algorithms proposed in the literature were identified. These algorithms were implemented in a common framework and experimentally evaluated in two multi-label datasets which have different properties. Several aspects considered by these algorithms, as well as the experimental protocol used to evaluate the results, are highlighted.

The remainder of this work is organized as follows: Section II briefly presents active learning and multi-label learning. Section III describes some important issues to be considered when applying active learning on multi-label data. Section IV presents the experiments carried out and the main results. Conclusions and future work are presented in Section V.

II. Background

A. Active Learning

Differently from the passive model of supervised learning where the values of the target variable(s) is/are obtained without taking into account the learning algorithm, in active learning the learner interactively requests supervision for the data points of its own choice.

There are basically the three main active learning scenarios [1], [2]:

1) membership query synthesis;
2) stream-based; and
3) pool-based.

In the first active learning scenario, the learner may query any unlabeled instance in the input space and also queries generates by the learner *de novo* (synthesis). The second scenario considers the data sequentially, deciding individually whether an unlabeled object should or not be labeled. In the pool-based scenario, all unlabeled data (or unlabeled pool) is evaluated before selecting one or more objects to be labeled. Figure 1 shows a standard pool-based active learning cycle.

This work focus on the pool-based scenario, as it 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 [1], [3], [4], [5].

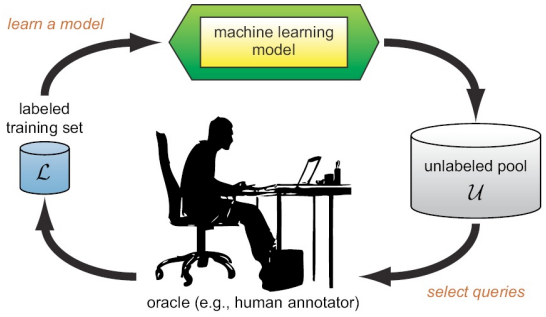


Fig. 1. Standard pool-based active learning cycle. Figure taken from [1].

B. Multi-Label Learning

In single-label learning, only one label from a disjoint set of labels L is associated to each example in the dataset. However, there are many applications in which the examples can be associated to several labels simultaneously, characterizing a multi-label learning problem.

Let D be a training set composed of N examples $E_i = (\mathbf{x}_i, Y_i)$, $i = 1..N$. Each example E_i is associated with a feature vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iM})$ described by M features X_j, $j = 1..M$, and a subset of labels $Y_i \subseteq L$, where $L = \{y_1, y_2, \ldots, y_q\}$ 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 a more generic scenario, the goal of multi-label learning is also to generate a model that is capable to predict a ranking of the labels, relevance scores (sometimes marginal probabilities) per label, or even the full joint probability distribution for the labels. k

Multi-label learning methods can be organized into two main categories: i) algorithm adaptation; and ii) problem transformation [6]. Methods in the first category extends 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 used methods in this category. BR decomposes the multi-label problem into q binary single-label problems, one for each label in L, and it solves each problem separately.

Unlike the single-label classification evaluation measures, multi-label classification must deal with *partially correct* classifications. To this end, several evaluation measures have been proposed. A complete discussion on multi-label classification evaluation measures is out of the scope of this paper, and can be found in [6]. In what follows, we briefly describe the label-based evaluation measures used in this work.

For each single label $y_i \in L$, the q binary classifiers are initially evaluated using any one of the binary evaluation measures proposed in the literature, which are afterwards averaged over all labels. Two averaging operations, *macro-averaging* and *micro-averaging*, can be used to average over all labels.

Let $B\left(T_{P_{y_i}}, F_{P_{y_i}}, T_{N_{y_i}}, F_{N_{y_i}}\right)$ be a binary evaluation measure calculated for a label y_i based on the number of true positive (T_P), false positive (F_P), true negative (T_N) and false negative (F_N). In this work we use the F-Measure $= \frac{2T_P}{2T_P+F_P+F_N}$. The macro-average version of B is defined by Equation 1 and the micro-average by Equation 2.

$$B_{macro} = \frac{1}{q}\sum_{i=1}^{q} B\left(T_{P_{y_i}}, F_{P_{y_i}}, T_{N_{y_i}}, F_{N_{y_i}}\right) \quad (1)$$

$$B_{micro} = B\left(\sum_{i=1}^{q} T_{P_{y_i}}, \sum_{i=1}^{q} F_{P_{y_i}}, \sum_{i=1}^{q} T_{N_{y_i}}, \sum_{i=1}^{q} F_{N_{y_i}}\right) \quad (2)$$

As macro-averaging would be more affected by labels that participate in fewer multi-labels, it is appropriate in the study of unbalanced datasets.

III. Active Learning from Multi-Label Data

There are a number of issues that need to be considered when attempting to apply active learning on multi-label data. In the following sections we focus on the most important ones.

A. Manual annotation approaches and effort

Similarly to a single-label active learning system, a multi-label active learning system can request the annotation of one or more objects. If the request is for just one object, then the annotator will observe (look at, read, hear, watch) the object in an attempt to understand it and characterize it as relevant or not to each of the labels. In practice, requests are made for a batch of objects. For example, 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 [7]. In such cases, there are two ways that an annotator can accomplish the task:

1) *object-wise*, where for each object the annotator determines the relevancy to each label; and
2) *label-wise*, where for each label the annotator determines relevancy to each object[1].

Consider a request for the annotation of n objects with q labels. Let

- c_o be the average cost of understanding an object;
- c_l be the average cost of understanding a label; and
- c_{lo} be the average cost of deciding whether an object should be annotated with a particular label or not.

If we set aside the cognitive and psychological aspects of the annotation process, such as our short-term memory capacity, then a rough estimation of the total cost of object-wise annotation is:

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

Similarly, a rough estimation of the total cost of object-wise annotation is:

[1]Object-wise and label-wise annotation have been called global and local labeling respectively in [8]

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

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

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

This means that choosing the annotation approach, largely depends on the object and label understanding costs. If object (label) understanding is larger, then the object (label) wise approach should be followed.

As Figure 2 illustrates, object understanding is less costly than label understanding only for images, which humans understand in milliseconds. Documents, audio and video require far more time to understand than typical label concepts.

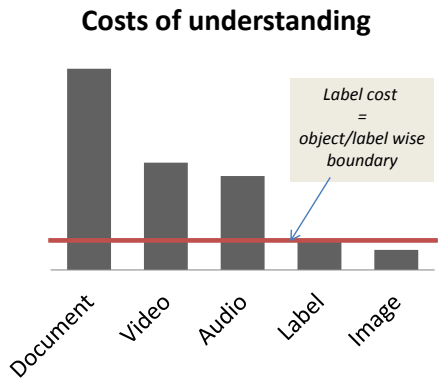


Fig. 2. The cost of understanding a label in different types of data.

B. Full and partial annotation requests

As mentioned in Section II-A, in classical supervised learning task the active learning system requests the value of the target variable for one or more objects. What can the learning system request in multi-label learning?

Normally it should request the values of all binary target variables (labels) for one or more objects. Then a (batch) incremental multi-label learning algorithm can update the current model based on the new examples. A different approach is taken in [9], where the system requests the values for only a subset of the labels and subsequently infers the values of the remaining labels based on label correlations.

Sticking to the values of just a subset of the labels would require an algorithm that is incremental in terms of partial training examples. Binary relevance (BR) is perhaps the sole algorithm fulfilling this requirement, but it is a standard and often strong baseline. Therefore, the development of active learning strategies that request partial labeling of objects could be a worthwhile endeavor. However, there is an implication on annotation effort that has to be considered. If the system requests the labeling of the same object at two different annotation requests, then the cost of understanding this object would be incurred twice. As discussed in Section III-A, this is inefficient for most data types.

C. Evaluation of unlabelled instances

The key aspect in a single-label active learning algorithm is the way it evaluates the informativeness of unlabelled instances. In multi-label data, the evaluation function (*query*) of active learning algorithms comprises two important parts:

1) a function (*scoring*) to evaluate object-label pairs; and
2) a function (*aggregating*) 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;
L: set of labels;
F: multi-label learner;
$Oracle$: the annotator;
t: bach size;
R: number of rounds
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 object-wise annotation approach.

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

The following measures have been proposed in the related work for evaluating object-label pairs (*scoring*):

Confidence-based score: [10], [8], [11]. The value of the instances prediction's confidence returned by the base classifier is used. The nature of this value depends on the bias of learner. It could be a margin-based value, a probability-based value, or others.

Ranking-based score: [11]. This strategy works like a normalization approach for the values obtained from the Confidence-based strategy. The confidence given by the base classifiers are used to rank the unlabeled examples for each label. The value returned by this approach represents how far

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

an example is from the boundary decision threshold between positive and negatives examples.

Disagreement-based score: [12], [13]. Unlike the other approaches, this strategy uses two base classifiers and measures the difference between their predictions. The intuitive idea is to query the examples that most disagree in their classifications and could be most informative. Three ways to combine the confidence values output by the classifiers have been proposed:

1) MMR;
2) HLR; and
3) SHLR.

MMR uses a major classifier which outputs confidence values and an auxiliary classifier that outputs decisions (positive or negative only). The auxiliary classifier is used to determine how conflicting the predictions are. HLR 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. SHLR tries to make a balance between MMR and HLR through a function that defines the influence of each approach in the final score.

After having the object-label scores, there are two main aggregation strategies to combine the object-label scores to an overall object score:

1) AVG; and
2) MIN/MAX.

AVG averages the object-label scores across all labels. Thus, given the q object-label scores $e_{i,j}$ of object E_i, the overall object-label score of object E_i is given by:

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

On the other hand, MIN/MAX considers the optimal (minimum or maximum) of the object-label scores, given by:

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

D. Experiment protocol

Besides the multi-label active learning strategies themselves, the way how the evaluation of these methods was performed is also an important characteristic for the related work. Some important 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 for each related work.

Regarding the initial labeled pool, each work built it in different ways. In [11] the examples are chosen to have at least one example positive and one negative for each label. In [13], from 100 to 500 examples were selected randomly to compose the initial labeled pool. In [8], the first 100 chronologically examples were selected. In [10], the author choose randomly 10 examples to compose the initial labeled pool.

The batch size defines how many examples are queried in each round of active learning. In [11], [10], only one example was queried per round. [8] and [13] choose 50 examples in each round, but the last one also performed experiments with batch size of 20.

There are basically two different ways to define the testing set.

The first way is to consider a totally separated testing set — Figure 3. This way was used in [8] and [10][2].

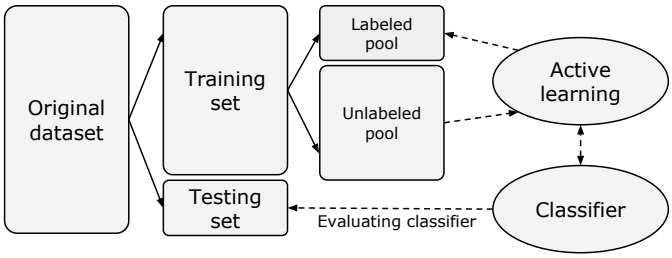


Fig. 3. Experimental protocols using a separated testing set

The other way is to use the remaining examples in the unlabeled pool as testing — Figure 4. This approach was used in [11], [13].

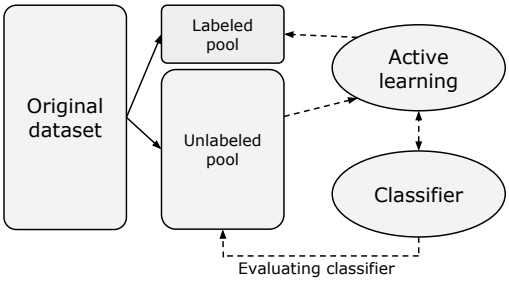


Fig. 4. Experimental protocols using the remaining unlabeled pool

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

[2]Actually, there is no explicit description about the testing set, however, it seems that the authors in [10] used a separated one.

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 separated testing set (Figure 3) is the correct evaluation approach for the performance assessment for the inductive inference setting.

The remaining evaluation approach (Figure 4) is biased by the active learner and hence the evaluation on these remaining examples will not be representative of the 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. The work [11] is an example. The authors argue that 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. For this application, the learning assessment should be done by using the remaining examples in the query pool (Figure 4).

The learning curve is the most common evaluation approach used to assess active learning techniques, and was used in the related work. A learning curves 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 [11], [13], [8] also used the value of the evaluation measure in the end of some specific number of rounds to assess the active learning techniques.

IV. Experiments

The active learning algorithms described in Section III-C, as well as the active learning evaluation framework, were implemented using Mulan[3] [14], a Java package for multi-label learning based on Weka[4]. Our implementation is publicly available to the community at http://www.labic.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/active-learning.zip.

A. Setup

The experiments were performed using the datasets *Scene* and *Yeast*, two classic multi-label datasets, which can be found in the Mulan website[5]. *Scene* dataset addresses the problem of semantic image categorization. Each instance in this dataset is an image associated with some of the six available semantic classes (beach, sunset, fall foliage, eld, mountain, and urban). *Yeast* is a biological dataset for gen function classification. Each instance is a yeast gene described by the concatenation of micro-array expression data and phylogenetic prole associated with one or more different functional classes.

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

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

[5]http://mulan.sourceforge.net/datasets.html

Table I describes the datasets, where CL (cardinality) and DL (density) are defined as $CL(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}|Y_i|$ and $DL(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i|}{q}$, respectively.

TABLE I
DATASETS DESCRIPTION

Dataset	domain	#ex	#feat	q	CL	DL	#dist
Scene	image	2407	294	6	1.074	0.179	15
Yeast	biology	2417	103	14	4.237	0.303	198

These two datasets have different properties. Although both datasets have similar number of examples, *Scene* dataset has low number of labels (6), few different multi-labels (15) and low cardinality (1.074). On the other hand, *Yeast* dataset has 14 labels, 198 different multi-labels, and a reasonably high cardinality (4.237). This means that instances in the *Yeast* dataset have more complex label space than the instances in the *Scene* dataset. Thus, learning from the *Yeast* dataset would be more difficult than learning from the *Scene* dataset.

Information related to label frequency is also important to characterize multi-label datasets. To this end, Table II shows summary statistics related to labels frequency, where (Min) Minimum, (1Q) 1^{st} Quartile, (Med) Median, (3Q) 3^{rd} Quartile and (Max) Maximum. Recall that 1Q, Med and 3Q divide the sorted labels frequency into four equal parts, each one with 25% of the data. Note that *Yeast* dataset in unbalanced.

TABLE II
LABELS FREQUENCY STATISTICS

Dataset	domain	Min	1Q	Med	3Q	Max
scene	image	364	404	429	432	533
yeast	biology	34	324	659	953	1816

Figure 5 shows a graphic distribution of the datasets label frequency using the Violin plot representation, which adds the information available from local density estimates to the basic summary statistics inherent in box plots. Note that the Violin plot may be viewed as boxplots whose boxes have been curved to reflect the estimated distribution of values over the observed data range. Moreover, observe that the boxplot is the black box in the middle, the white dot is the median and the black vertical lines are the whiskers, which indicate variability outside the upper and lower quartiles.

As mentioned in Section III-C, the active learning algorithms implemented in this work are combinations of functions to evaluate object-label pairs and to aggregate these scores. The functions to evaluate the object-label pairs, *i.e.*, the *scoring* function, are:

- Confidence-based (CONF)
- Ranking-based (RANK)
- HLR Disagreement-based (HLR)
- MMR Disagreement-based (MMR)
- SHLR Disagreement-based (SHLR)

and the functions to aggregate the outputted scores, *i.e.*, the *aggregating* function, are:

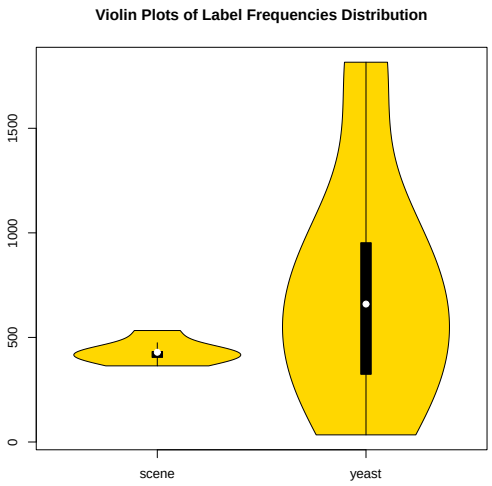


Fig. 5. Violin plots of label frequencies distribution.

- average (AVG)
- maximum (MAX)
- minimum (MIN)

In this work, 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 datasets. $N_{ini} = 5, 10, 20$ was used in order to evaluate the influence of different sizes of the initial labeled pool. The general procedure — Algorithm 1 — was executed with a batch size $t = 1$, *i.e.*, one example is annotated in each run. The Binary Relevance approach and LR-based (logistic regression) as major classifier were used. For the disagreement-based approaches, we used Support Vector Machines with LR normalization, which gives probability values as output. Both learners, named respectively SGD and SMO, are implemented in the Weka framework.

B. Results and Discussion

Besides the learning curve, another alternative to summarize the active learning performance is the area under the learning curve (AULC). In this work, we use the values of AULC to evaluate the multi-label active learning algorithms.

All results were obtained using 10-folds cross-validation. All experimental results can be found in the supplementary material at http://www.labic.icmc.usp.br/pub/mcmonard/ExperimentalResults/CLEI2015-ALLRESULTS.xls. In what follows, the main results are presented.

Tables III to VI show the rankings of the AULC obtained by the different *scoring* and *aggregating* functions using the three different initial labeled pool of examples and the two experimental protocols to evaluate the classifiers: *separated* and *remaining*.

Independently of the *aggregating* function used (AVG, MAX or MIN), most of the methods ranked first use $N_{ini} = 20$ to set up the initial labeled pool. The number of methods ranked first using $N_{ini} = 20$, from a total of 10 in each table is: 8 in Table III; 7 in Table IV; 9 in Table V; and 7 in Table VI. Moreover, methods using $N_{ini} = 20$ were never ranked last. All the remaining methods ranked first use

TABLE III
RANKING AULC - SEPARATED - SCENE

		Macro					Micro				
		conf	**hlr**	**mmr**	**rank**	**shlr**	**conf**	**hlr**	**mmr**	**rank**	**shlr**
	5	9	2	4	2	8	6	**1**	6	5	7
avg	**10**	8	3	8	6	9	8	3	9	3	9
	20	7	**1**	5	3	5	5	2	7	**1**	8
	5	2		7	4	6	7		4	6	3
max	**10**	5		9	8	7	9		5	4	4
	20	**1**		6	**1**	3	2		**1**	2	2
	5	4		2	7	**1**	3		3	8	5
min	**10**	6		3	9	4	4		8	9	6
	20	3		**1**	5	2	**1**		2	7	**1**

TABLE IV
RANKING AULC - REMAINING - SCENE

		Macro					Micro				
		conf	**hlr**	**mmr**	**rank**	**shlr**	**conf**	**hlr**	**mmr**	**rank**	**shlr**
	5	9	2	5	3	8	9	2	4	3	8
avg	**10**	8	3	7	6	9	8	3	7	6	9
	20	7	**1**	4	2	5	7	**1**	5	2	4
	5	2		9	4	7	2		8	4	6
max	**10**	3		8	5	6	4		9	5	7
	20	**1**		6	**1**	4	**1**		6	**1**	3
	5	5		2	8	**1**	5		**1**	8	**1**
min	**10**	6		3	9	3	6		3	9	5
	20	4		**1**	7	2	3		2	7	2

TABLE V
RANKING AULC - SEPARATED - YEAST

		Macro					Micro				
		conf	**hlr**	**mmr**	**rank**	**shlr**	**conf**	**hlr**	**mmr**	**rank**	**shlr**
	5	6	2	5	5	6	3	2	2	5	2
avg	**10**	8	3	9	3	9	2	3	8	4	9
	20	5	**1**	7	2	7	**1**	**1**	7	**1**	7
	5	7		4	6	3	9		**1**	3	3
max	**10**	9		6	4	4	8		4	6	6
	20	2		**1**	**1**	2	7		3	2	4
	5	4		3	8	5	4		5	8	5
min	**10**	3		8	9	8	6		9	9	8
	20	**1**		2	7	**1**	5		6	7	**1**

TABLE VI
RANKING AULC - REMAINING - YEAST

		Macro					Micro				
		conf	**hlr**	**mmr**	**rank**	**shlr**	**conf**	**hlr**	**mmr**	**rank**	**shlr**
	5	6	**1**	6	5	7	3	**1**	2	6	2
avg	**10**	8	3	9	3	9	2	3	8	4	9
	20	5	2	7	**1**	8	**1**	2	6	**1**	7
	5	7		4	6	3	9		**1**	3	3
max	**10**	9		5	4	4	8		3	5	6
	20	2		**1**	2	2	7		4	2	5
	5	3		3	8	5	4		5	7	4
min	**10**	4		8	9	6	6		9	9	8
	20	**1**		2	7	**1**	5		7	8	**1**

$N_{ini} = 5$. However, diferently than the previous case, methods using $N_{ini} = 5$ were also ranked last: 1 in Table III; 3 in Table IV; 1 in Table V; and 2 in Table VI. Although it is expected that a greater initial labeled pool of examples could help active learning, note that in some cases good results can also be obtained with smaller labeled pool of examples,

Table VII shows the best aggregation and initial labeled pool configuration (N_{ini}) for each active learning approach based on the AULC obtained using both experimental protocols, *remaining* and *separated*. The last two columns refer to the *Random* strategy which selects at random the examples to label and it is considered as a baseline. The best results are highlighted in bold. Results lower or equal than the corresponding baseline are underlined.

Observe that using the experimental protocol *remaining* all results in terms of best aggregation, N_{ini} and active learning approach are different for both datasets. However, using the experimental protocol *separated*, the same best configuration was found (AVG(20)) for both datasets when using the active learning approach HLR. When using RANK, the same best configuration was found (MAX(20) but only when Macro-F1 measure is used to evaluate the model.

Note that the aggregation approach (MAX/MIN) has been chosen in 85% of all the cases as the best aggregation option. As previously observed, the most frequent size of the initial pool is $N_{ini} = 20$, followed by $N_{ini} = 5$. It is worth noting that not only SHLR and HLR do not appear among the best options, but their best results are most of the time worse than the corresponding baseline. Although MMR, CONF and RANK obtain the best results in 4, 2 and 2 cases, respectively, all of them better than the corresponding baseline, they also present results which are worse than the corresponding baseline.

Considering the best cases in each of the experimental protocols, *remaining* and *separated*, the same aggegation and initial labeled pool configurations were found in 15 out of the 20 cases. Note that in 10 of these 15 cases the value of AULC using the remaining protocol is greater than the one using the separated protocol. Recall that the quality of the model assessed by the remaining protocol holds for examples in the unlabeled pool, and does not necessarily holds for new unlabeled data. To this end, the separated protocol should be used.

To illustrate, the following figures show the learning curves for MMR, CONF and RANK using MAX(20) as configuration options, as well as the random baseline, for the first 1000 instances (or rounds as only one instance is labeled in each round) labeled. Figures 6 and 7 use *separated* and *remaining* as testing protocol, respectively.

Note that the general behavior of the learning curves is quite different in each dataset, independently of the testing protocol used. Recall that although both datasets have similar number of examples, the *Yeast* dataset is unbalanced, it has more than twice the number of labels than *Scene*, as well as greater label density — Table II.

For dataset *Scene* using *separated* as testing protocol — Figure 6 — the behavior of the learning curves is similar for Macro-F1 and Micro-F1. In both cases the MMR and RANK learning curves dominate the baseline. However, after labeling 400 examples, there is little improvement. The learning curve of MMR is dominated by the baseline until 400 examples are labeled. Afterwards MMR, as well as MMR and RANK show little improvement. Using *remaining* as testing protocol — Figure 7 — as before, the MMR and RANK learning curves dominate the baseline and the learning curve of MMR is dominated by the baseline until 300 examples are labeled. Afterwards MMR, as well as MMR and RANK show improvement, being MMR the one that shows the grater improvement.

For dataset *Yeast* using *separated* as testing protocol — Figure 6 — only the RANK learning curve dominates the baseline. The best Macro-F1 and Micro-F1 measure values are obtained after 200 examples are labeled. The difference using *remaining* as testing protocol is that some better Macro-F1 and Micro-F1 measure values are obtained by RANK after labeling more examples.

Moreover, observe that differently than for *Scene* dataset, in which the Micro-F1 and Macro-F1 measure values are in the same range, there is a considerable difference among these values for *Yeast* dataset, in which Macro-F1 is worse than Micro-F1. This is due to the fact that *Scene* is an unbalanced dataset and Macro-F1 is more affected by labels that participate in fewer multi-labels.

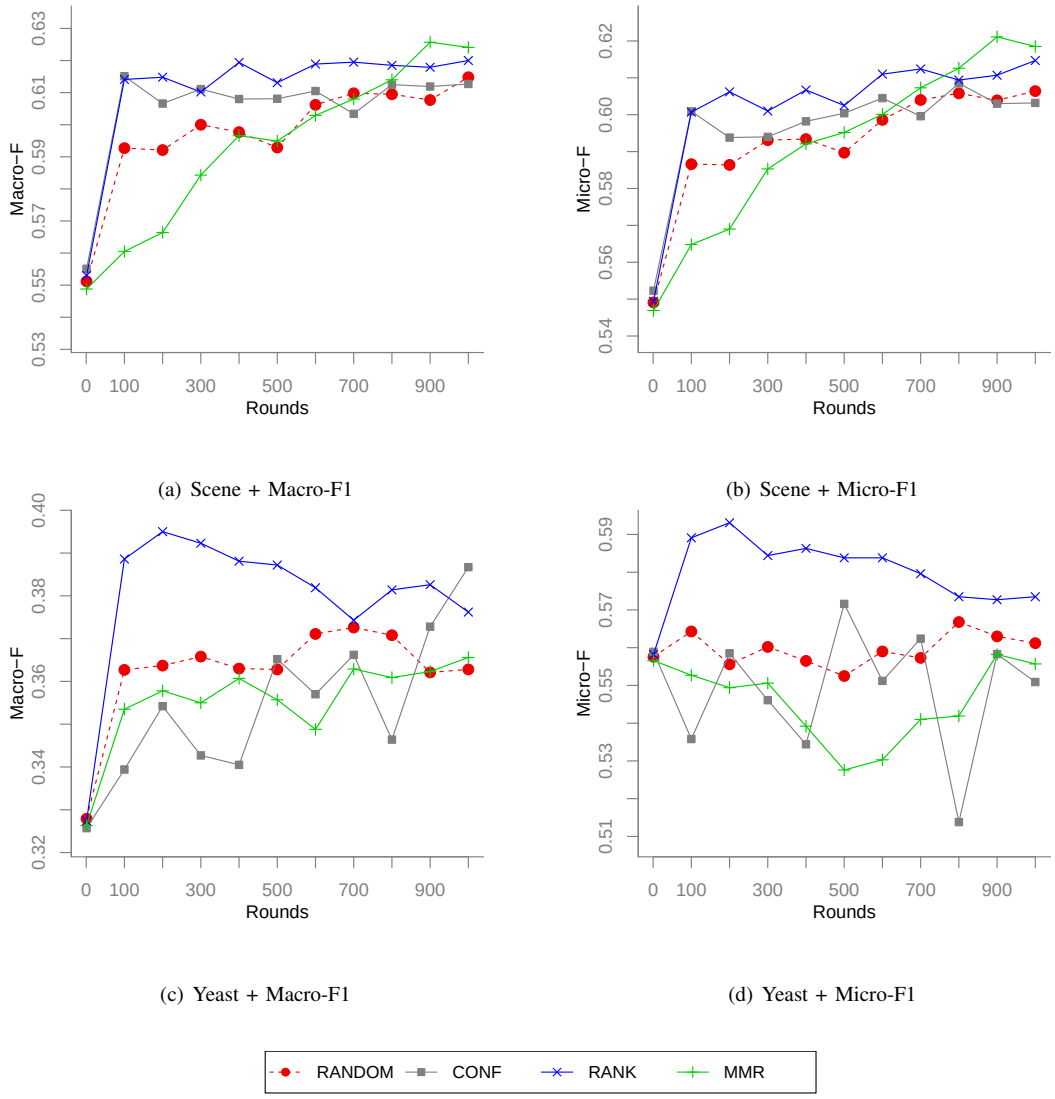
Comparing *Random* (passive learning) to the evaluated active learning methods, *Rank* was the only strategy that always outperfomed *Random* in both datasets.

Note that active learning seems to be more useful for *Yeast* than for *Scene*, as the difference between random and the active method (RANK) was clearly higher for *Yeast*. This behavior could be explained by the datasets properties. *Scene* is an easier dataset to learn from than *Yeast*. Consequently, *Scene* has less room for active learning improvements.

V. Conclusion

In the classic supervise learning approach, all labels are obtained in advance, independently of the learning algorithm. On the other hand, in the active learning approach the learning algorithm interactively chooses which objects are to be labeled, aiming to reduce the number of labeled examples needed to learn. The active learning approach is particularly important whenever there is abundant unlabeled data available, but labeling this data is an expensive task. Although active learning in single-label learning has been investigated over several decades, this is not the case for multi-label learning. This work provides a general introduction to multi-label active learning, focusing on the scenario where the evaluation of unlabeled data is taken into account to select the objects to be labeled. In this scenario, several multi-label active learning algorithms proposed in the literature were identified, implemented in a common framework and experimentally evaluated in two multi-label datasets which have different properties.

Multi-label active learning seemed to be more useful for *Yeast*, a more difficult to learn dataset, than for *Scene*.



(a) Scene + Macro-F1

(b) Scene + Micro-F1

(c) Yeast + Macro-F1

(d) Yeast + Micro-F1

Fig. 6. Learning curves using *separated* as testing protocol.

Moreover, the *Rank* strategy was the only one that always outperformed the passive learning method (*Random*).

In future work, we plan to experimentally evaluate these multi-label learning algorithms in more datasets. Moreover, we plan to explore the active learning capability in multi-label semi-supervised learning, which aims to also learn from unlabeled data. In this case, the disagreement among two (or more) classifiers can be used by the semi-supervised algorithm to decide on querying the labels of an object.

ACKNOWLEDGMENT

This research was supported by the São Paulo Research Foundation (FAPESP), grants 2010/15992-0 and 2011/21723-5.

REFERENCES

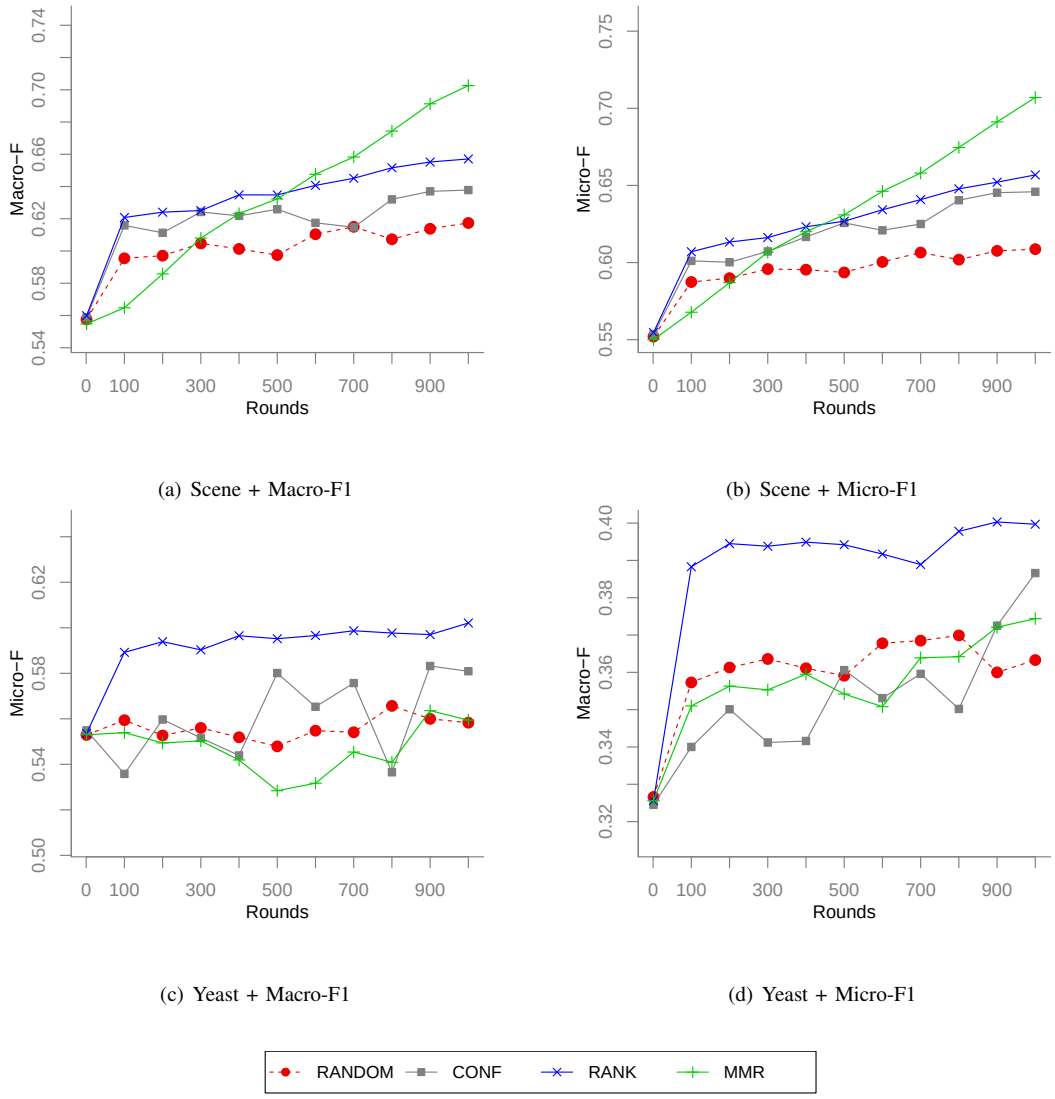
[1] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Tech. Rep. 1648, 2010.

[2] C. C. Aggarwal, X. Kong, QuanquanGu, J. Han, and P. S. Yu, "Active learning: A survey," in *Data Classification: Algorithms and Applications*, C. C. Aggarwal, Ed. CRC Press, 2014, pp. 571–606.

[3] B. Zhang, Y. Wang, and F. Chen, "Multilabel image classification via high-order label correlation driven active learning," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1430–1441, 2014.

[4] C. Ye, J. Wu, V. S. Sheng, S. Zhao, P. Zhao, and Z. Cui, "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), 2015, pp. 583–586.

[5] S. Huang, S. Chen, and Z. Zhou, "Multi-label active learning: Query type matters," in *Proceedings of the Twenty-Fourth International Joint Conference onArtificial Intelligence, IJCAI 2015*, 2015, pp. 946–952.



(a) Scene + Macro-F1

(b) Scene + Micro-F1

(c) Yeast + Macro-F1

(d) Yeast + Micro-F1

Fig. 7. Learning curves using *remaining* as testing protocol

[6] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," *Data Mining and Knowledge Discovery Handbook*, pp. 1–19, 2009.

[7] S. Nowak, K. Nagel, and J. Liebetrau, "The clef 2011 photo annotation and concept-based retrieval tasks," in *CLEF (Notebook Papers/Labs/Workshop)*, 2011, pp. 1–25.

[8] A. Esuli and F. Sebastiani, "Active learning strategies for multi-label text classification," in *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ser. ECIR '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 102–113.

[9] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang, "Two-Dimensional Multilabel Active Learning with an Efficient Online Adaptation Model for Image Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1880–1897, 2009. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2008.218

[10] K. Brinker, "On active learning in multi-label classification," in *From Data and Information Analysis to Knowledge Engineering*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nurnberger, and W. Gaul, Eds. Springer Berlin Heidelberg, 2006, pp. 206–213. [Online]. Available: http://dx.doi.org/10.1007/3-540-31314-1_24

[11] M. Singh, A. Brew, D. Greene, and P. Cunningham, "Score Normalization and Aggregation for Active Learning in Multi-label Classification," University College Dublin, Tech. Rep., 2010.

[12] C.-W. Hung and H.-T. Lin, "Multi-label active learning with auxiliary learner," in *3rd Asian Conference on Machine Learning*, Taoyuan, Taiwan, 2011, p. to appear.

[13] B. Yang, J.-T. Sun, T. Wang, and Z. Chen, "Effective multi-label active learning for text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 917–926. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557119

[14] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, pp. 2411–2414, 2011.

TABLE VII

BEST CONFIGURATION $<aggregation>(<N_{ini}>)$ FOR EACH ACTIVE LEARNING APPROACH AND BOTH EXPERIMENTAL PROTOCOLS.

		conf	**hlr**	**mmr**	**rank**	**shlr**	**rand(5)**	**rand(10)**	**rand(20)**
Remaining									
Scene	**Macro**	MAX(20)	AVG(20)	**MIN(20)**	MAX(20)	MIN(5)			
		0.608	0.584	**0.614**	0.610	0.591	0.592	0.583	0.604
	Micro	MAX(20)	AVG(20)	**MIN(5)**	MAX(20)	MIN(5)			
		0.592	0.582	**0.603**	0.598	0.592	0.587	0,574	0.597
Yeast	**Macro**	MIN(5)	AVG(5)	MAX(20)	**AVG(20)**	MIN(20)			
		0.348	0.345	0.344	**0.375**	0.352	0.346	0.346	0.362
	Micro	**AVG(20)**	AVG(5)	MAX(5)	AVG(20)	MIN(20)			
		0.588	0.569	0.556	0.584	0.562	0.562	0.549	0.556
Separated									
Scene	**Macro**	MAX(20)	AVG(20)	**MIN(20)**	MAX(20)	MIN(5)			
		0.603	0.573	**0.606**	0.604	0.587	0.591	0.577	0.600
	Micro	MAX(20)	AVG(20)	**AVG(5)**	MAX(20)	MIN(5)			
		0.587	0.572	**0.594**	0.593	0.578	0.587	0.572	0.594
Yeast	**Macro**	MIN(20)	AVG(20)	MAX(20)	**MAX(20)**	MIN(20)			
		0.344	0.344	0.345	**0.377**	0.354	0.346	0.346	0.365
	Micro	**AVG(20)**	AVG(20)	MAX(5)	AVG(20)	MIN(20)			
		0.585	0.565	0.557	0.584	0.564	0.562	0.549	0.560