



**HAL**  
open science

## Active Learning Algorithms for Multi-label Data

Everton Alvares Cherman, Grigorios Tsoumakas, Maria-Carolina Monard

► **To cite this version:**

Everton Alvares Cherman, Grigorios Tsoumakas, Maria-Carolina Monard. Active Learning Algorithms for Multi-label Data. 12th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Sep 2016, Thessaloniki, Greece. pp.267-279, 10.1007/978-3-319-44944-9\_23 . hal-01557592

**HAL Id: hal-01557592**

**<https://inria.hal.science/hal-01557592>**

Submitted on 6 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Active learning algorithms for multi-label data

Everton A. Cherman<sup>1</sup>, Grigorios Tsoumakas<sup>2</sup>, and Maria C. Monard<sup>1</sup>

<sup>1</sup> University of Sao Paulo,  
Institute of Mathematics and Computer Sciences, Sao Carlos - SP, Brazil  
{echerman,mcmonard}@icmc.usp.br

<sup>2</sup> Aristotle University of Thessaloniki,  
Department of Informatics, 54124 Thessaloniki, Greece  
greg@csd.auth.gr

**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, for labels. As the learner is allowed to interactively choose the data from which it learns, it is expected that the learner will 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, where objects can have more than one class labels and a multi-label learner is trained to assign multiple labels simultaneously to an object. We discuss the key issues that need to be considered in pool-based multi-label active learning and discuss how existing solutions in the literature deal with each of these issues. We further empirically study the performance of the existing solutions, after implementing them in a common framework, on two multi-label datasets with different characteristics and under two different applications settings (transductive, inductive). We find out interesting results that we attribute to the properties of, mainly, the data sets, and, secondarily, the application settings.

**Keywords:** Supervised learning, multi-label learning, active learning, pool-based strategies

## 1 Introduction

Different approaches to enhance supervised learning have been proposed over the years. As supervised learning algorithms build classifiers based on labeled 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 [6]. 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 developed substantially to support single-label learning, where 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, and the interesting challenges that it poses, multi-label learning has attracted great interest in the last decade [9].

We here focus on the pool-based active learning scenario [6], where a pool of unlabeled data is available to the learning algorithm. The first contribution of this paper is the presentation of the key issues that have to be considered when applying active learning on (multi-label) data, as well as the particular decisions of existing algorithms in the literature with respect to these issues (Section 2). We implemented existing algorithms in a common framework within the *Mulan* library [8] and empirically investigated their performance on two multi-label data sets with different properties and under two different application settings (transductive, inductive). The second contribution of this paper is the presentation of these experimental results, where novel and interesting conclusions are drawn with respect to the factors that affect the performance of the different algorithms (Section 3).

## 2 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.

### 2.1 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 [4]. 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<sup>3</sup>.

---

<sup>3</sup> Object-wise and label-wise annotation have been called global and local labeling respectively in [2]

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 label-wise annotation is:

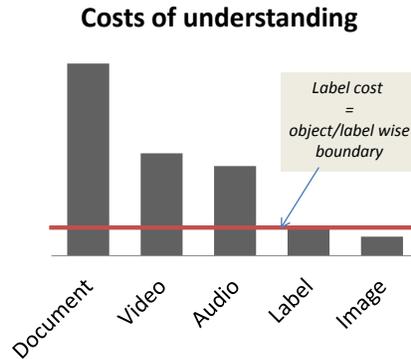
$$q[c_l + n(c_o + c_{lo})] = qc_l + nqc_o + nqc_{lo}$$

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

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

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 1 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. 1.** The cost of understanding a label in different types of data.

## 2.2 Full and partial annotation requests

In a 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 [5], 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 2.1, this is inefficient for most data types.

### 2.3 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 *scoring* function to evaluate object-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  $r = 1, 2, \dots, R$  do
     $H \leftarrow F(D_l)$ 
     $\{E_i\}_{i=1}^t \leftarrow \text{query}(H, L, D_u, t)$ 
     $\{Y_i\}_{i=1}^t \leftarrow \text{Oracle}(\{E_i\}_{i=1}^t)$ 
     $D_l \leftarrow D_l \cup \{(E_i, Y_i)\}_{i=1}^t$ 
     $D_u \leftarrow D_u - \{E_i\}_{i=1}^t$ 
end

```

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

Algorithm 2 shows the *query* function of a multi-label active learning procedure. The *scoring* function 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  $y_j \in L = \{y_1, y_2, \dots, y_q\}$ . The *aggregating* function considers the  $q$  evidence values  $e_{i,1}, e_{i,2}, \dots, e_{i,q}$  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
     $e_{i,j} \leftarrow \text{scoring}(D_u, H, E_i, y_j)$ 
  end
   $e_i \leftarrow \text{aggregating}(e_{i,1}, e_{i,2}, \dots, e_{i,q})$ 
end
 $query \leftarrow \text{best}(e_1, e_2, \dots, t, D_u)$ 

```

**Algorithm 2:** The *query* function

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

1. Confidence-based score [1, 2, 7]. 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.
2. Ranking-based score [7]. 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.
3. Disagreement-based score [3, 10]. 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. Three ways to combine the confidence values output by the two base classifiers have been proposed:
  - i. MMR 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. 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.
  - iii. 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 obtained the object-label scores, there are two main aggregation strategies for combining the object-label scores to an overall object score:

1. 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:

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

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

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

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.

## 2.4 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 reference to related work.

Regarding the initial labeled pool, different papers built it in different ways. In [7], the examples are chosen to have at least one example positive and one negative for each label. In [10], 100 to 500 examples were selected randomly to compose the initial labeled pool. In [2], the first 100 chronologically examples were selected. In [1], 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 [7, 1], only one example was queried per round. In [2] 50 examples were chosen in each round, while in [10] experiments with both 50 and 20 examples were performed.

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 in [2] and though not explicitly mentioned, it seems to have also been followed in [1]. The second way is to use the remaining examples in the unlabeled pool for testing. This approach was used in [7, 10].

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 for 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 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 [7]. 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  $D_t$ . 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, [7, 10, 2] 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

The active learning algorithms described in Section 2.3, as well as the active learning evaluation framework, were implemented using Mulan<sup>4</sup> [8], a Java package for multi-label learning based on Weka<sup>5</sup>. Our implementation is publicly available to the community at <http://www.labicc.icmc.usp.br/pub/mcmonard/Implementations/Multilabel/active-learning.zip>.

#### 3.1 Setup

The experiments were performed using the datasets *Scene* and *Yeast*, two classic multi-label datasets, which can be found in the Mulan website<sup>6</sup>. *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 gene 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.

Table 1 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.

<sup>4</sup> <http://mulan.sourceforge.net>

<sup>5</sup> <http://www.cs.waikato.ac.nz/ml/weka>

<sup>6</sup> <http://mulan.sourceforge.net/datasets.html>

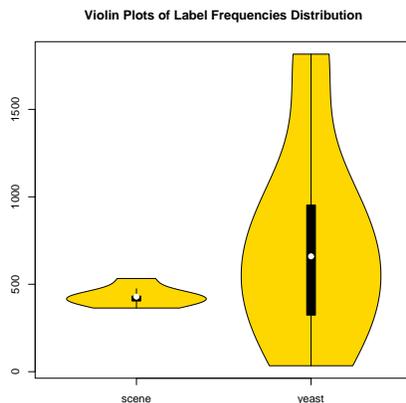
**Table 1.** Datasets description and label frequency statistics

Dataset	domain	#ex	#feat	$q$	CL	DL	#dist	Min	1Q	Med	3Q	Max
<i>Scene</i>	image	2407	294	6	1.074	0.179	15	364	404	429	432	533
<i>Yeast</i>	biology	2417	103	14	4.237	0.303	198	34	324	659	953	1816

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 1 also shows summary statistics related to labels frequency, where (Min) Minimum, (1Q) 1<sup>st</sup> Quartile, (Med) Median, (3Q) 3<sup>rd</sup> 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 is unbalanced.

Figure 2 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.

**Fig. 2.** Violin plots of label frequencies distribution.

As mentioned in Section 2.3, 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). The functions to aggregate the outputted scores, *i.e.*, the *aggregating* function, are: average (AVG) and maximum or minimum (MAX/MIN), depending on the score function.

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  $N_{ini} \times q \geq \sum_{i=1}^{|D_i|} Y_i$ , 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 was used as the multi-label classifier, using stochastic gradient descent with hinge loss as the base classifier. For the disagreement-based approaches, we used the sequential minimal optimization algorithm with a linear kernel. Both learners, are implemented in the Weka framework, and are named SGD and SMO respectively.

### 3.2 Results and Discussion

We report results in terms of the micro  $F_1$  measure, and in particular its average over 1500 iterations of active selection of one example in each iteration. This is proportional to the area under the corresponding learning curve of the different algorithms. Figure 3 presents the results. Bold typeface is used to highlight the relative best performance of the different scoring functions and aggregation strategies for each particular experimental setting (dataset and protocol pair). All results were obtained using 10-folds cross-validation. The full experimental results are available online as supplementary material<sup>7</sup>.

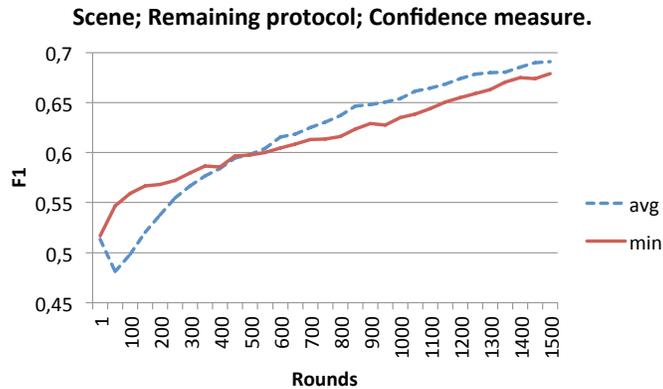
	Scene				Yeast			
	Remaining		Separated		Remaining		Separated	
	AVG	MAX	AVG	MAX	AVG	MAX	AVG	MAX
Confidence	0,6478	0,6274	0,5867	0,5918	0,6100	0,5556	0,5557	0,5415
Rank	0,6377	0,6376	0,6024	0,6029	0,5790	0,5907	0,5670	0,5689
MMR	<b>0,6979</b>	0,6562	<b>0,6038</b>	0,5961	0,5878	0,5586	0,5622	0,5534
HLR	0,6885		0,5999		<b>0,6248</b>		<b>0,5705</b>	
<u>SHLR</u>	<u>0,6815</u>	<u>0,6617</u>	<u>0,5960</u>	<u>0,5970</u>	<u>0,5900</u>	<u>0,5571</u>	<u>0,5609</u>	<u>0,5583</u>
Random		0,5928		0,5916		0,5589		0,5602

Fig. 3. Experimental results.

<sup>7</sup> <http://www.labic.icmc.usp.br/pub/mconard/ExperimentalResults/AIAI2016-ALLRESULTS.xls>

The first question we want to answer is **how does the size of the initial pool of training examples affect the performance of the methods?** Here we noticed the same strange general pattern across both data sets and application settings and across all algorithms: Having 5 and 20 examples per label leads to similar performance, which is slightly better compared to having 10 examples per label. In the rest of the experiments we removed this factor by considering the average results of the three different sizes of the initial pool.

The next question we want to answer is **which aggregation strategy works best for each scoring function and under what conditions?** For the confidence-based score function, in the separated protocol *min* is the best aggregation strategy in both yeast and scene, while for the remaining protocol, *avg* works best in both yeast and scene. Taking the *min* of the confidence-based score stresses more the labels for which the classifier is most uncertain (e.g. rare labels that it has not seen yet), while *avg* treats all labels equally. We hypothesize that in the remaining protocol instances with rare (difficult to be predict) labels are removed from the test set and hence stressing the performance in such labels is meaningless. In contrast, in the separate protocol, rare labels in the test set remain rare and important. Figure 4 shows the learning curves of *min* and *avg* in scene for the remaining protocol. It confirms our hypothesis, as in the initial steps, *avg* does not perform as well as *min*, but as more and more rare labels are being removed from the test set, it eventually does better. This is an important conclusion for researchers developing methods for a particular protocol, or practitioners applying methods in a particular protocol setting.



**Fig. 4.** Average vs minimum in scene for the confidence-based score function.

For the rank-based score function, in the scene data set, *max* and *avg* work equally well for both the remaining/separate protocols, while in the yeast database *max* gives slightly better results. The rank-based score function normalizes the absolute values of uncertainty across the labels and hence makes itself all labels equal in this sense. This alleviates the issue we discussed above. We hypothesize that the aforementioned difference between yeast and scene is due to the corresponding differences in label frequencies, as shown in Figure 2. The *max*

aggregation pays more attention to labels where the *relative* uncertainty with respect to other labels is higher, and this pays-off better, in accordance with the theory of active learning. In scene, as there are fewer labels with similar distributions, it makes no difference in focusing on all labels or only on the most uncertain one.

In terms of the disagreement-based score functions, in MMR *avg* works better than *max* for both data sets and protocols, while for SHLR, *avg* works better than *max* in the remaining setting, while they perform similarly in the separate setting. Here, we would expect similar results with confidence-based scoring and indeed we see that *avg* does better than *max* in the remaining setting. However, in contrast with confidence-based scoring, here *avg* dominates also in the separate protocol. It seems that while uncertainty is maximized in the case of rare labels, the same does not happen for the disagreement between the two classifiers. We hypothesize that this occurs because with limited training data for rare labels both classifiers' output is similarly uncertain. We also argue that the disagreement of classifiers per label, again in itself, brings all labels to the same measurement level (normalization). This also explains the good results of the average strategy.

The next question we want to answer is **which scoring function works best and under what conditions?** Comparing the different scoring functions with each other, we notice that the disagreement-based functions do best overall, with MMR giving the best results in scene and HLR the best results in yeast for both protocols. HLR is the most robust method, delivering near-top results also in scene. Recall that HLR takes into account crisp decisions instead of confidences. This shows that looking at actual confidence values can be misleading, particularly in the presence of rare labels and imbalanced distributions across the labels. In scene, where labels are similar in frequency, MMR did best, hence in these - rare in practice - cases, we expect actual confidences to offer benefits.

Further interesting results are obtained by **comparing random selection of unlabeled instances (passive learning) with the active learning approaches**. In particular, we notice that large gains are achieved in the transductive setting, while active learning methods are struggling to beat passive learning in the separated setting. This shows that in the remaining setting, the benefits of active learning are coming mostly from the removal of difficult instances from the test set rather than from the incorporation of useful instances to the training set, an interesting conclusion for active learning in general (non multi-label) that to the best of our knowledge has not been previously discussed in the literature.

## 4 Summary and Future Work

Although active learning in single-label learning has been investigated over several decades, this is not the case for multi-label learning. This work discussed key issues in pool-based (multi-label) active learning based on existing algorithms in the literature, which were implemented in a common framework and experimen-

tally evaluated in two multi-label data sets with different properties and under two different application settings (transductive, inductive).

Results show that taking the average across all labels of disagreement-based scoring functions perform best, and that in particular the MMR function works better in the absence of imbalance among the labels, while HLR works better in the presence of such imbalance. Moreover, the transductive setting was found to be easier for active learning due to the removal of difficult examples.

In the future, we plan to expand our empirical study with more data sets, in order to assess the generality of our conclusions.

## Acknowledgment

This research was supported by the São Paulo Research Foundation (FAPESP), grants 2010/15992-0 and 2011/21723-5, and Brazilian National Council for Scientific and Technological Development (CNPq), grant 644963.

## References

1. Brinker, K.: 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*, pp. 206–213. *Studies in Classification, Data Analysis, and Knowledge Organization*, Springer Berlin Heidelberg (2006)
2. Esuli, A., Sebastiani, F.: Active learning strategies for multi-label text classification. In: *Proceedings of the 31st European Conference on IR Research*. pp. 102–113. ECIR '09, Springer-Verlag, Berlin, Heidelberg (2009)
3. Hung, C.W., Lin, H.T.: Multi-label active learning with auxiliary learner. In: *3rd Asian Conference on Machine Learning*. p. to appear. Taoyuan, Taiwan (2011)
4. Nowak, S., Nagel, K., Liebetrau, J.: The clef 2011 photo annotation and concept-based retrieval tasks. In: *CLEF (Notebook Papers/Labs/Workshop)*. pp. 1–25 (2011)
5. Qi, G.J., Hua, X.S., Rui, Y., Tang, J., Zhang, H.J.: Two-Dimensional Multilabel Active Learning with an Efficient Online Adaptation Model for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 1880–1897 (2009), <http://dx.doi.org/10.1109/TPAMI.2008.218>
6. Settles, B.: Active learning literature survey. Tech. Rep. 1648, University of Wisconsin–Madison (2010)
7. Singh, M., Brew, A., Greene, D., Cunningham, P.: Score Normalization and Aggregation for Active Learning in Multi-label Classification. Tech. rep., University College Dublin (2010)
8. Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., Vlahavas, I.: Mulan: A java library for multi-label learning. *Journal of Machine Learning Research* 12, 2411–2414 (2011)
9. Tsoumakas, G., Zhang, M.L., Zhou, Z.H.: Introduction to the special issue on learning from multi-label data. *Machine Learning* 88(1-2), 1–4 (2012)
10. Yang, B., Sun, J.T., Wang, T., Chen, Z.: Effective multi-label active learning for text classification. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 917–926. KDD '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1557019.1557119>