Improving Multi-label Classification by Means of Cross-Ontology Association Rules

Fernando Benites and Elena Sapozhnikova

Department of Computer and Information Science,
{Fernando.Benites,Elena.Sapozhnikova}@uni-konstanz.de

Abstract. Recently several methods were proposed for the improvement of multi-label classification performance by using constraints on labels. Such constraints are based on dependencies between classes often present in multi-label data and can be mined as association rules from training data. The rules are then applied in a post-processing step to correct the classifier predictions. Due to properties of association rule mining these improvement methods often achieve low improvement expressed mostly in the better prediction of large classes. In the presence of class ontologies this is undesirable because larger classes correspond to higher levels in hierarchies presenting general concepts and can thus be trivial. In this paper we overcome the problem by focusing on improving multi-label classification performance on small classes. We present a new method of improvement based on mining cross-ontology association rules which is best suited for classification with multiple class ontologies, but can also be applied to multi-label classification with one class taxonomy.

1 Introduction

The increasing popularity of ontologies in different areas has led to the availability of data that can be annotated with multiple classes coming from different class taxonomies. This is a special case of multi-label classification. Generally, combining information from the ontologies providing different insights into a domain can be helpful in discovering new cross-ontology associations not evident from only one ontology. For example, if a film is classified by its genre in a genre ontology and by the producing company in an ontology of producers, one can find a possible interesting relation between a certain genre and a producing company, specialized in this genre. Recently, data mining techniques such as association analysis were applied to finding valuable cross-ontology Association Rules (ARs) between multiple ontologies corresponding to distinct categorizations of genes in bioinformatics [2, 9].

On the other hand, useful information from such cross-ontology rules can be successfully employed to improve performance in multi-label classification: ARs

In: R. Bergmann, S. Görg, G. Müller (Eds.): Proceedings of the LWA 2015 Workshops: KDML, FGWM, IR, and FGDB. Trier, Germany, 7.-9. October 2015, published at http://ceur-ws.org

found among classes of multiple ontologies can be used to correct predicted labels because the presence of a certain class or classes can be helpful for predicting another one. For example, a proper application of the association between a certain genre and a producing company specializing in this genre, as discussed above, can increase the probability of correctly predicting a genre, providing the corresponding company has been already correctly predicted. Thus it should lead to an improvement in classification performance. This is especially important with respect to very large ontologies with many thousands of classes, which are usually difficult to deal with.

Another problem with class ontologies that has not yet been dealt with sufficiently in recent research is that classifier performance in such a case is largely dominated by more general classes higher in the hierarchy because they are more present in the data and hence simpler to predict for a classifier. On the other hand, such general classes do not often provide interesting information and are sometimes trivial. For this reason, in mining cross-ontology rules rare association rules [14] are preferred, especially in large ontologies [2]. Similarly, in the improvement of multi-label classification performance, prediction improvement is more interesting for small and more specific classes in comparison to larger ones. As existing methods have not yet addressed this problem, our paper will focus on mining rare cross-ontology ARs and applying them to the multi-label classification improvement on small classes. For this purpose, a special interestingness measure well-suited for mining rare rules is utilized.

An important difference between our approach and several state-of-the-art methods discussed in the next section is that they use constraints for labels of one labelset and not two different class ontologies. Further, our approach focuses on rare labels, i.e. the ones with low support, since they are normally the greater part of the labelset.

The rest of the paper is organized as follows. A brief overview of the approaches to multi-label improvement with ARs is given in Section 2. Afterwards, our approach is explained in Section 3, followed by the experiments of Section 4. In Section 5 we conclude the paper.

2 Related Work

Recently several approaches to improving multi-label classification performance were proposed that dealt with dependencies between classes present in multi-label data. Some of them belong to the field of multi-label classification with constraints and apply constraints on labels to performance improvement usually in a post-processing step. The constraints are often mined in form of ARs from training data [8, 10].

The initial work was devoted to prediction corrections within the ranking by pairwise comparison framework [10]. The constraints were in the form of many-to-one ARs *labelset→label*, i.e. implications from a labelset to a single label. They could be positive or negative which involves either setting or removing a consequent label as a result of the presence of an antecedent label combination. The constraint rules were extracted using a standard support-confidence AR mining framework in order to change the predicted rankings. The results of the method obtained on real-world datasets were negative: no improvement in comparison to the baseline performance was observed. For this reason, this method will not be used for comparison with the proposed method below.

A more recent approach of [8] used only one-to-one ARs $label_i \rightarrow label_j$ in order to improve SVM performance in the Binary Relevance (BR) setting. The rules to apply were chosen by minimizing the Ranking Loss performance measure through a cross-validation process on the training data. The selected rules were then applied to predicted label rankings in the test phase, if an antecedent label was set, boosting the score of the corresponding consequent label. The improved results were obtained for two real-world datasets Yeast and Reuters. AR mining was based on the standard support-confidence framework. A subsequently extended approach [6] differs in that it uses subsets of labels gathered by clustering, and also extracts negative and many-to-one ARs. Still the evaluation of the extended approach was restricted to smaller datasets than in the earlier paper (e.g. the Reuters dataset was not included), perhaps pointing to a higher complexity of the algorithm, making it probably inapt to be applied to large datasets. Taking this into account, we selected only the initial method of [8] for comparison and will refer to it as Label Constraints for SVMs (LCS).

In contrast to the discussed post-processing methods evaluated in a certain multi-label classification setting (either pairwise or BR), a more general approach, Label Reduction with Association Rules (LRwAR), was proposed in [5]. It includes pre- and post-processing for the reduction of the label dimensionality. First, ARs are extracted and those labels that are only in the consequents of rules are removed from the data to be learned. Then a multi-label classifier is applied to the classification problem with a reduced labelset. After classification, the rules are applied to recover missing labels. An advantage of this approach is a shorter time needed to train a classifier on a smaller labelset. It also used the standard support-confidence framework to mine ARs, although its recent extension [4] proposed Conviction instead of Confidence. However it was not shown to provide significantly better results. The base method was evaluated on different multi-label classifiers including ML-kNN, BP-MLL and C4.5 (the latter in BR and label powerset settings) as well as six datasets. On several of them it showed either minimal (e.g. 0.6% relative improvement on the Yeast data) or no improvement at all. The performance of the extended method measured in terms of two performance measures was lower on the Yeast data in comparison to the baseline classifiers and it was generally inferior or equal to them in more than half of all experiments (79 from 140).

3 Improvement with Rare Association Rules

Besides relatively low improvement demonstrated by the existing methods, they have the problem of using the standard support-confidence framework for AR mining, which normally extracts high support rules that often exist between large classes. So they ignore small classes as a potential source for improvement because minimum support filtering can remove not only noise but also rare classes. The greatest problem of Confidence in such a setup is that associations of small classes to large ones are normally ranked very high. In the case of a class ontology these ARs simply show hierarchical parent-child relations, i.e. that one label a is more specific than another b. The rule extracted would be $a \rightarrow b$, which means that if label a appears, then label b should appear too. Such obvious relations can be derived from an extracted hierarchy, on the one hand, as in [3] and are misleading for classification improvement, on the other. The reason being that applying such rules in the case of LRwAR [5] for removing classes with a high support and setting them based on predictions of classes with a lower support, is prone to error. An example would be if class A appeared only 10 times, class B appeared 100 times and both appeared together 10 times, so a rule A→B would be extracted. LRwAR would then imply that B should be removed from the labelset and only in the post-processing step reinserted based on the prediction of A. Although in [4] Conviction was used instead of Confidence, it is still closely related to Confidence and behaves similarly.

Another problem with the standard AR framework is choosing the thresholds for Support and Confidence which is done manually and can therefore be suboptimal. So, the first issue to be dealt with improving multi-label classification performance by constraints is the acquisition of high-quality rules. In the proposed approach we solve this problem by omitting the minimum support threshold and using a special interestingness measure which is well-suited for rare rules. Additionally we tune its threshold automatically depending on the range of values for extracted rules. The idea is to use rare ARs between classes that is from a small class to another small one and that classes belong to two or more ontologies describing different aspects of a dataset. In this case hierarchical relations between the classes of one ontology will not be taken into account. In such a setting, training data are annotated with categories of both ontologies. Then mined rules are used to improve class predictions for one of both ontologies. So, rare cross-ontology rules can be helpful in order to solve the described problems.

The second important issue is deciding when to apply a rule. Is the predicted label trusty enough to insert an additional label based on its presence? The worst case scenario would be that the rule is applied on the basis of a false positive inserting an additional false positive. Another undesirable outcome would be that the antecedent label is a true positive but applying a rule would create a false positive, i.e. the prediction of the classifier should not be overruled. The desirable decision is only to use a true positive to add another true positive, i.e. that a rule corrects the missclassification of a label. In [8] the rules are applied to all labels in the rules, assuming that all antecedents were reliably predicted. Although the rules were used before to optimize the Ranking Loss, it was not clear if the antecedent's ranking was high enough to be predicted. In order to solve this problem, the control of the quality of classifier predictions is proposed to create a basis for application of a rule. The detailed discussion of the proposed approach is presented in the next two subsections.

3.1 Selection of Rules

To extract pairwise ARs, we performed experiments with the interestingness measures well-suited for mining rare rules [14]. Such measures should possess the important property of null-transaction invariance [1]. Due to the lack of space we will focus here only on the Kulczynski measure (*Kulc*) which showed good results:

$$Kulc(A,B) = \frac{P_{AB}}{2} * (\frac{1}{P_A} + \frac{1}{P_B}).$$

In order to select only the best rules, adaptive thresholding without a predefined value was applied as follows: After calculating *Kulc* of all rules, the values are sorted in descending order as a curve C. We assume that there will be a slope between a few high scored rules and the rest. In order to select these rules the curve C is smoothed into S and only the part with a relatively low variance is analyzed. Thus we need first to determine whether the variance of the curve S is high:

$$CV = \frac{MEAN(S) - VAR(S)}{MEAN(S)} > \rho_m \qquad (1)$$

where $MEAN(S)$ is the average value of the curve S and $VAR(S)$ its variance. If the condition of Eq. (1) is true, we use only the values in the slope of the curve and calculate the median that defines a Threshold Value TV for the most interesting rules:

$$TV = C_{MEDIAN(\{i|DIFF(S_i)>MEAN(DIFF(S))\})}$$

where $DIFF(S)$ is the difference between two neighbor values in the curve S. Since the step size between two values is 1, $DIFF(S)$ can also be seen as the derivative of S. Otherwise, i.e. if the variance is high, the average of the values not much lower than the mean of the entire curve is taken:

$$TV = \underset{j\in\{i|S_i>MEAN(S)*\rho_t\}}{MEAN}(S_j) \quad .$$

Defining the threshold in this manner, we select only those rules that have *Kulc* values above TV as good enough to be applied to prediction improvement.

ρ_m and ρ_t should be set so that the changes are significant and the only high valued rules are selected, respectively.

3.2 Application of Rules

As discussed above, applying a rule for insertion of a label without taking the corresponding classifier's judgment into account can lead to no improvement or even poorer prediction performance. A better way would be to use rankings produced by the classifier. An attempt was proposed in [8] where the scores provided by the classifier for each label were used to optimize a parameter w varied from 0 to 1 for each pair of labels i and j. For a rule $i \rightarrow j$, new rankings of label j were calculated for each sample x as: $p_j(x) = w * p_j(x) + (1 - w) * p_i(x)$, where $p_i(x)$ is the score assigned to a label i, analogously for j. by its respective BR classifier for that sample. These new rankings were used to minimize Ranking Loss by varying w during cross-validation on a validation set. However the label i was chosen in the test phase, only if its score was above a threshold t used to turn predicted rankings into classes (also called decision boundary later on). Thus the rule $i \rightarrow j$ could not be applied otherwise.

A drawback of this method is that it relies on individual parameter optimization for each rule through a cost-intensive calculation of the Ranking Loss. This is not viable for large datasets as the later work [6] shows by using a fixed parameter value.

We propose a similar approach. The antecedent A of a rule $A \rightarrow B$ should be already positively predicted, i.e. should have a score greater than the threshold t, but an additional criterion should hold: $\frac{V_B}{V_A} > 0.5$, i.e. the score of the consequent should be at least 50% of the value of the antecedent in order to set the consequent.

As emphasized before, our approach was designed to work on classification problems with two different multi-label sets coming from two ontologies, but it can still be applied to the problems with only one class taxonomy in order to compare to other methods, as is shown in the next section.

4 Experiments

4.1 Data

We used two multi-label real-world datasets: Reuters and Yeast. The first one was used with two class ontologies "Topics" and "Industries" for mining cross-ontology ARs as well as in a simplified version with only "Topics" labels in order to compare our method to the results of other improvement methods published elsewhere.

The two-ontology Reuters dataset was formed by preprocessing with stopword removal and stemming the original data provided by `http://trec.nist.gov/data/reuters/reuters.html`. We used the 5000 most frequent terms in the training set and applied tf-idf weighting as well as column-wise normalization performed separately on training and test data. In the original 800k samples only 300k contained at least one "Industries" label. From these we selected random 30k samples and split them into a training and a test set with the ratio of 2:1, i.e. 20k training samples and 10k test samples. In total there were 103 "Topics" labels and 364 "Industries" labels. We will denote this dataset as Reuters 10k below.

The simplified version (Reuters 5k with only "Topics" classes) consisted of 5000 training and 5000 test samples chosen randomly from the original 23k training set. The data preprocessing was performed as described above.

For the sake of comparison, the Yeast dataset was also taken from the MEKA package [11]. It contains only 14 labels in one non-hierarchical label set and is therefore not very interesting for our experiments, but it is often used in the works on multi-label classification. From its 2417 samples, 1500 were selected randomly for training and the rest for test. We did not use cross-validation since certain aspects would be more difficult to analyze, for example, the graphs.

4.2 Experiment Setting

As a baseline classifier we used LIBLINEAR [7] in the BR setting, and on single class ontology datasets also ML-kNN as well as Classifier Chains (CC) based on LIBLINEAR. A crucial complication with LIBLINEAR is that the choice of the threshold t can be difficult. Normally, the value of 0.5 is recommended but in the work of [8] a different value and individual for each dataset (0.45 for Yeast and 0.47 for Reuters) was chosen. We also used different values for each dataset and additionally compared the results to the results of an adaptive method for selecting t, which automatically adjusts it to be close to the dataset label cardinality [13]. We will refer here to this method as Label Cardinality Approach (LCA). ML-kNN and CC were not used with two class ontologies because they do not scale well on large datasets, specially with LCS.

For LRwAR we also implemented a variation using rankings for all classes and only inserting new labels. We use the acronym OF (only fill) for this variation. The original method foresees deleting rankings of classes in the consequent of a rule as well.

Parameters of LRwAR and LCS were set as in their original works [5, 8], whereas the Confidence threshold was used to obtain the best results for both methods. In particular, we changed it for Reuters 5k so that the h-loss performance measure was comparable to the value of the baseline classifier.

Parameters for adaptive thresholding were set to $\rho_m = 0.2$ and $\rho_t = 0.75$. These values were obtained by the manual experimentation on the Reuters dataset and are, in our opinion, general enough to be used for all datasets.

4.3 Performance measures

We used the F-1 measure, which is the harmonic mean of Recall and Precision. It can be calculated in several ways depending on averaging [15]. First, we used instance-based averaging, i.e. we calculated F-1 for every single instance and then took the mean value (denoted as IF1). Additionally we used label-based F-1 both in micro-averaged version mF1 and in macro-averaged one LF1= $\frac{1}{Q}\sum_{i=1 \ldots Q}\frac{2*tp_i}{2*tp_i+fn_i+fp_i}$. Here Q is the number of labels and tp_i, fp_i and fn_i are, respectively, the number of true, false positives and false negatives for a label i. Micro-averaged mF1 is known to be dominated by the performance on large classes. Also Hamming Loss (h-loss) was used: $HL = \frac{fp+fn}{Q*N}$, where N is the number of test samples.

4.4 Results

Datasets with one class taxonomy: Yeast and Reuters 5k First we compared our approach to LRwAR,LCS, and LCA on the datasets used in other studies. In Table 1 the results for Yeast and Reuters 5k are depicted.

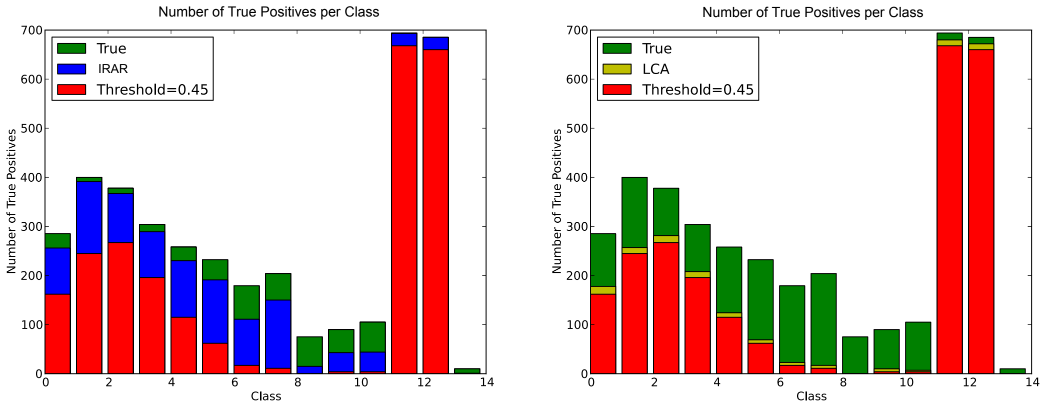
On the Yeast data, IRAR, LCS, and LRwAR OF could improve the results of BR and ML-kNN classifiers in terms of mF1, IF1, and LF1. The improvement achieved by IRAR was the highest. H-loss for this dataset could not be increased by any of the compared methods. Among them LRwAR was the worst because its results were even worse as those of the baseline classifiers in terms of all performance measures. In contrast to the other methods, IRAR was also better than LCA for BR and comparable to it for ML-kNN. This shows that a powerful thresholding strategy can outperform many improvement methods based on label constraints. IRAR was the only improvement methods that could increase the CC results. This was the highest LF1 value by far on this dataset.

This is consistent with the important fact that IRAR could increase the LF1 value significantly more than the other methods in almost all configurations. The only exception was for Reuters 5k and BR where it improved second best. This can be explained by the better improvement of the classification performance on small classes. Indeeed, as Figure 1a shows, the number of true positives on small and middle-size classes obtained by IRAR was higher than that of LCA (Figure 1b). This difference is even more pronounced if we compare F-1 values for each class obtained by all improvement methods and presented in Figure 2a. One can see, for example, that IRAR achieves a significant improvement in F-1 for the last class where the other methods show no improvement at all or that it has much more improvement on the classes 5-10.

Analyzing the curves of mF1 and LF1 in dependence on the threshold t one can see a trade-off between them (Figure 2b). IRAR is able to achieve both high LF1 and mF1 values near their crossing point.

On the Reuters 5k dataset, IRAR had again the highest improvement against the baseline classifier as compared to the other improvement methods in terms of all performance measures, except for h-loss. The largest performance difference was again in LF1. IRAR performance was comparable to that of LCA. LCS and LRwAR achieved a very small improvement against the baseline classifier and were worse than LCA in terms of all performance measures, except for h-loss. Here we can see that CC had was the second best classifier, but no improvement could beat LCA method. Again, the exception remains IRAR with LF1, having a 18% value increase over the baseline performance and 3% over LCA.

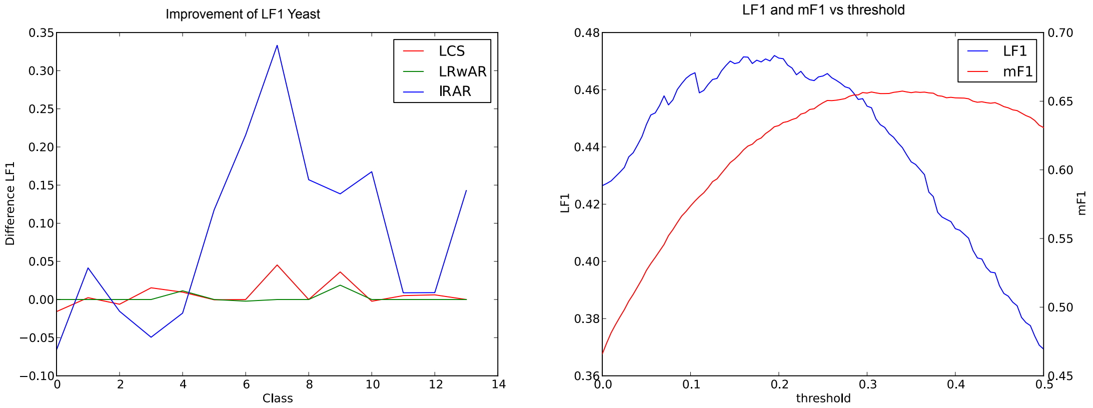
CC did not outperform BR in the experiments, although CC does consider the connections between the labels in a certain way. A solution would be to use Ensembles of CC (ECC) [12], since the order of the labels can be taken into account. However for ECC, the issue of larger label sets will be even much severe, since the label order must be permutated when creating a new CC to exhaust all alternatives at best.



(a) Yeast, IRAR, number of true positives

(b) Yeast, LCA, number of true positives

Fig. 1. Distributions of true positives on Yeast data.



(a) Yeast, improvement of LF1 per class for LCS, LRwAR, and IRAR

(b) Yeast, LF1 and mF1 calculated using the respective threshold on the rankings

Fig. 2. Improvement comparison on Yeast data.

Table 1. LCS, LRwAR, and IRAR applied to Yeast and Reuters 5k, OF=Only Filling, t = threshold, bold values mark the best values per dataset and column.

	BR				ML-kNN				CC			
Metrics	h-loss	mF1	IF1	LF1	h-loss	mF1	IF1	LF1	h-loss	mF1	IF1	LF1
Yeast												
LCA	0.2121	0.6498	0.6362	0.4088	0.2066	**0.6583**	**0.6467**	0.4042	0.2148	**0.6455**	**0.6326**	0.4053
	t=0.45				t=0.5				t=0.45			
baseline	**0.2043**	0.6477	0.6288	0.3915	**0.1990**	0.6221	0.5978	0.3496	**0.2097**	0.6370	0.6184	0.3854
LCS Cnf=0.6	0.2071	0.6516	0.6326	0.3983	0.1996	0.6263	0.6016	0.3596	0.2141	0.6313	0.6144	0.3540
LRwAR Cnf=0.6	0.2071	0.6328	0.6160	0.3479	0.2047	0.6034	0.5838	0.2982	0.2121	0.6223	0.6057	0.3401
LRwAR OF Cnf=0.6	0.2047	0.6480	0.6293	0.3935	**0.1990**	0.6221	0.5978	0.3496	0.2105	0.6367	0.6183	0.3857
IRAR	0.2269	**0.6544**	**0.6400**	**0.4453**	0.2169	0.6560	0.6446	**0.4101**	0.3553	0.6031	0.5992	**0.4760**
Reuters 5k												
LCA	0.0131	0.7893	**0.7955**	**0.4177**	**0.0164**	**0.7361**	**0.7446**	0.4306	0.0148	**0.7608**	**0.7716**	0.3910
	t=0.3											
baseline	**0.0122**	0.7849	0.7817	0.3690	**0.0164**	0.7339	0.7376	0.4303	0.0133	0.7567	0.7492	0.3302
LCS n=6,Cnf=0.8	**0.0122**	0.7856	0.7823	0.3696	0.0165	0.7335	0.7377	0.4311	0.0140	0.7382	0.7305	0.3246
LCS n=6, Cnf=.85	**0.0122**	0.7856	0.7823	0.3696	0.0165	0.7335	0.7377	0.4311	0.0140	0.7383	0.7307	0.3249
LRwAR Cnf=0.8	0.0138	0.7465	0.7442	0.3576	0.0177	0.7002	0.7015	0.4191	0.0148	0.7170	0.7119	0.3194
LRwAR Cnf=0.85	0.0130	0.7667	0.7633	0.3614	0.0169	0.7189	0.7204	0.4231	0.0140	0.7377	0.7302	0.3231
LRwAR OF Cnf=0.8	**0.0122**	0.7851	0.7819	0.3690	**0.0164**	0.7340	0.7378	0.4303	**0.0132**	0.7584	0.7508	0.3313
LRwAR OF Cnf=0.85	**0.0122**	0.7849	0.7817	0.3690	**0.0164**	0.7339	0.7376	0.4303	**0.0132**	0.7584	0.7508	0.3313
IRAR	0.0125	**0.7900**	0.7895	0.3958	0.0187	0.7174	0.7347	**0.4433**	0.0164	0.7452	0.7490	**0.4001**

Dataset with two class ontologies: Reuters 10k Table 2 depicts the results of classification improvement for the Reuters 10k dataset, first classified separately in “Topics” and “Industries” and then with improved “Industries” predictions, by using cross-ontological ARs. In general, the classification performance for “Topics” was higher than for “Industries” classes. The results of the improvement methods LCS and IRAR for this class ontology were better than those of the baseline classifier, except that h-loss of IRAR was lower. At the same time, LRwAR showed negative improvement and LRwAR OF only improvement at the fourth place after the decimal point. In contrast, IRAR was able to achieve the overall best LF1. Its results were also somewhat similar to the results of LCA. It is interesting to note that LCS outperformed LCA in terms of mF1 and IRAR in terms of LF1. So, we can conclude that LCS is more effective for classes with large support and IRAR for those with small support. This will be due to the use of confidence to extract the rules.

The results for Reuters 10k “Industries” are similar to those obtained for “Topics”. Here LRwAR had even more negative improvement in terms of all performance measures and LRwAR OF showed again only marginal improvement. LCS was equal or better than the baseline and achieved again the highest mF1 value. This time both LCA and IRAR were worse than the baseline in terms of h-loss and mF1, but improved IF1 and LF1. However IRAR was better than LCA in three out of four performance measures and had again the best LF1.

Using cross-ontology ARs for the improvement of "Industries" predictions revealed an interesting fact: the results of LCS and both LRwAR variants worsened in comparison with those shown in the previous experiment while IRAR could improve its h-loss and mF1 values. Here, LCS uses the thresholds of different classifiers trained with different labelsets that may obstruct its performance. Also the low occurrence of labels in the labelsets may lead to poor results of the Confidence-based methods.

Table 2. LCS, LRwAR, and IRAR applied to Reuters BR's predictions for "Topics" and "Industries" 10k, OF=Only Filling, t = threshold, bold values mark the best values per dataset and column.

Metrics	h-loss	mF1	IF1	LF1	h-loss	mF1	IF1	LF1
	Reuters 10k "Topics" t=0.45				Reuters 10k "Industries" t=0.3			
LCA	0.0123	0.8257	**0.8335**	0.4237	0.0070	0.6462	**0.6466**	0.2884
baseline	**0.0116**	0.8258	0.8282	0.3938	**0.0061**	0.6589	0.6060	0.2772
LCS k=5,n=6,Cnf=0.7	**0.0116**	**0.8264**	0.8287	0.3942	**0.0061**	**0.6605**	0.6077	0.2778
LCS k=5,n=6,Cnf=0.85	**0.0116**	**0.8264**	0.8287	0.3942	**0.0061**	0.6603	0.6076	0.2776
LRwAR Cnf=0.7	0.0134	0.7912	0.7890	0.3819	0.0071	0.5491	0.4649	0.2672
LRwAR Cnf=0.85	0.0122	0.8143	0.8145	0.3871	0.0067	0.5936	0.5138	0.2698
LRwAR OF Cnf=0.7	**0.0116**	0.8259	0.8284	0.3940	**0.0061**	0.6592	0.6068	0.2773
LRwAR OF Cnf=0.85	**0.0116**	0.8260	0.8285	0.3940	**0.0061**	0.6592	0.6067	0.2773
IRAR	0.0132	0.8187	0.8312	**0.4298**	0.0067	0.6539	0.6120	**0.2918**
Reuters 10k "Topics"→"Industries", t=0.3								
LCS Cnf=0.7,	**0.0061**	0.6589	0.6060	0.2772				
LCS k=5,n=6,Cnf=0.85,	**0.0061**	0.6589	0.6060	0.2772				
LRwAR Cnf=0.7	0.0087	0.3482	0.2880	0.2293				
LRwAR OF Cnf=0.7	**0.0061**	0.6585	0.6056	0.2771				
IRAR	0.0062	**0.6590**	**0.6092**	**0.2825**				

5 Conclusion

In this paper we proposed a novel method of classification improvement in multi-label classification IRAR. It uses cross-ontology association rules and focuses on the improvement of predictions for small classes. Additionally, we compared it with state-of-the-art methods developed to correct predicted rankings by using constraints on labels in a post-processing step. One of the methods, LRwAR, showed negative improvement in most of the experiments and its variation only marginal improvement. LCS scored better in terms of improvement, but a better thresholding strategy such as LCA often achieved even more improvement. IRAR could outperform LCA in three out of four performance measures on the Yeast and Reuters 10k datasets. More importantly, it boosted the LF1 value significantly and showed the best LF1 result in three of five experiments. This means that IRAR is well suited for improving performance on small classes. This method is also able to achieve the trade-off between LF1 and mF1, i.e. it was able to achieve a high LF1 at a relatively low number of false positives. This points to the fact that the method can be used effectively with datasets exhibiting highly skewed label distributions as, for example, in the case of class ontologies.

References

1. Benites, F., Sapozhnikova, E.: Evaluation of hierarchical interestingness measures for mining pairwise generalized association rules. IEEE Trans. Knowl. Data Eng. 26(12), 3012–3025 (2014)
2. Benites, F., Simon, S., Sapozhnikova, E.: Mining rare associations between biological ontologies. PLoS ONE 9, e84475 (2014)
3. Brucker, F., Benites, F., Sapozhnikova, E.P.: Multi-label classification and extracting predicted class hierarchies. Pattern Recognition 44(3), 724–738 (2011)
4. Charte, F., Rivera, A., del Jesus, M., Herrera, F.: LI-MLC: A label inference methodology for addressing high dimensionality in the label space for multilabel classification. IEEE Trans. Neural Netw. Learn. Syst. 25(10), 1842–1854 (2014)
5. Charte, F., Rivera, A., del Jesus, M., Herrera, F.: Improving multi-label classifiers via label reduction with association rules. In: Hybrid Artificial Intelligent Systems, LNCS, vol. 7209, pp. 188–199 (2012)
6. Chen, B., Hong, X., Duan, L., Hu, J.: Improving multi-label classification performance by label constraints. In: IJCNN 2013. pp. 1–5 (Aug 2013)
7. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. JMLR 9, 1871–1874 (2008)
8. Gu, W., Chen, B., Hu, J.: Combining binary-svm and pairwise label constraints for multi-label classification. In: Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. pp. 4176–4181 (2010)
9. Manda, P., McCarthy, F.M., Bridges, S.M.: Interestingness measures and strategies for mining multi-ontology multi-level association rules from gene ontology annotations for the discovery of new go relationships. J. of Biomedical Informatics 46(5), 849–856 (2013)
10. Park, S.H., Fürnkranz, J.: Multi-label classification with label constraints. In: ECML PKDD 2008 Workshop on Preference Learning. pp. 157–171 (2008)
11. Read, J., Reutemann., P.: Meka multi-label dataset repository, `http://meka.sourceforge.net/`, May 20 2015
12. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: European Conference on Machine Learning and Knowledge Discovery in Databases: Part II. pp. 254–269. ECML PKDD '09, Springer-Verlag, Berlin, Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-04174-7_17`
13. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine learning 85(3), 333–359 (2011)
14. Surana, A., Kiran, U., Reddy, P.K.: Selecting a right interestingness measure for rare association rules. In: 16th Int. Conf. on Management of Data (COMAD). pp. 115–124 (2010)
15. Tsoumakas, G., Katakis, I., Vlahavas, I.: Mining multi-label data. In: In Data Mining and Knowledge Discovery Handbook. pp. 667–685 (2010)