Towards Automatic Classification of Learning Objects: Reducing the Number of Used Features

Pedro González[1], Eva Gibaja[1], Alfredo Zapata[2], Víctor H. Menéndez[2], Cristóbal Romero[1]
[1]University of Cordoba, Dept. of Computer Science, 14071, Córdoba, Spain
[2]Autonomous University of Yucatan, Faculty of Education, 97305, Mérida, Mexico
{pgonzalez,egibaja,cromero}@uco.es, {zgonzal, mdoming}@correo.uady.mx

ABSTRACT

The automatic classification of LOs into different categories enables us to search for, access, and reuse them in an effective and efficient way. Following this idea, in this paper, we focus specifically on how to automatically recommend the classification attribute of the IEEE LOM when a user adds a new LO to a repository. To do it, we propose the use of the multi-label classification approach, since each LO might be simultaneously associated with multiple labels. An initial problem we have found is that the number of terms or pure text features that characterize LOs tends to be very high. So, we propose to apply a dimensionality reduction process. We have carried out an experiment using 515 LOs from the AGORA repository in order to try to reduce the number of features or attributes used, improving execution time without losing prediction accuracy.

Keywords

Multi-label classification, feature selection, learning object

1. INTRODUCTION

The IEEE Learning Object Metadata standard (IEEE LOM) defines several attributes that may be assigned to each Learning Object (LO). However, manual entering all these metadata is a time-consuming process and automated techniques are required for a wider adoption of the standard [2]. In this paper, we focus on how to automatically recommend the classification attribute of the IEEE LOM when a user adds a new LO to a repository. Our idea is to recommend the user what are the possible categories that a LO belongs to from just user-provided information about the LO (such as the title, keywords and description). In order to do it, we propose to use multi-label classification for automatic categorization of LOs from the terms or pure text features that characterize these LOs. Multi-label classification (MLC) is a variant of the classification problem where multiple target labels can be assigned simultaneously to each instance [1]. In traditional classification classes are mutually exclusive, that is, a specific instance can belong to just a single class. However, there are occasions where classes present overlapping, that is, a specific instance can belong to several classes. In our case, we use MLC because a specific LO could belong to several categories.

2. PROPOSED METHODOLOGY

Our proposed approach for automatically classifying of LOs is represented in figure 1. First, we create the data file starting from the terms or pure text features that characterize LOs extracted from the LOs metadata, and categories to which the LO belongs to. Therefore, our next step consists in performing an attribute selection. The final step is the application of a MLC algorithm that will give us a model for classifying new LOs.

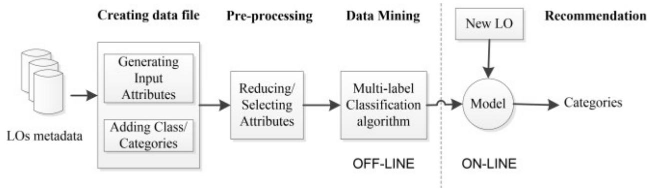

Figure 1. LO multi-label classification approach.

3. EXPERIMENTAL WORK

The data file used in this work has been extracted using 515 LOs from the AGORA repository [3] as follows. When a user adds a new LO to AGORA, he must provide information such as title, keywords, description and other related IEEE LOM metadata. Starting from these information about all the LOs we extracted 1336 terms (features) after removing stop words and stemming (to reduce the terms to their roots). Next, we compute the frequency of these roots for the LO at issue obtaining its term frequency (TF) representation. So, we obtained an example-term matrix, in which each element represents how many times a term appears in an example. We also normalized the count to term frequency to measure the importance of a term. Besides, in AGORA, a user has to specify one or several categories to which the LO belongs to from a predefined set of five academic disciplines: Engineering and Technology; Natural and Exact Science; Social and Administrative Science; Education, Humanities and Art; Health Science. So, we added the 5 labels (in binary format) to each LO as classes to predict. Then, we applied a dimensionality reduction process for reducing the number of attributes in the dataset. The motivation is to reduce training and classification times and removing noisy and irrelevant attributes, which can have a negative impact on accuracy results. Usually, there exists a wide range of possible terms that can refer to LOs of very different topics, and hence, the number of attributes describing LOs tends to be very high. Feature selection has been performed according to a specific method for MLC suggested in [5]. First, the χ^2 feature ranking method was separately applied to each label. Thus, for each label, the worth of each attribute is estimated by computing the χ^2 statistic with respect to the label to determine its independence. The core idea is that, if an attribute is independent on a class, this attribute could be removed. The result of this step is a ranking of all features for each label according to the statistic. Finally, the top-*n* features were selected based on their maximum rank over all labels. Finally, 13 different state-of-the-art MLC algorithms [1] have been applied to the different versions of the data set. They include 3 adaptation algorithms: AdaBoost.MH, Multi-Label k-Nearest Neighbor (MLkNN) and Instance-based Logistic Regression (IBLR), and 10 transformation algorithms in which the J48 implementation of C4.5 decision tree algorithm has been used as base classifier: Binary Relevance (BR), Classifier

Chais (CC), Calibrated Label Ranking (CLR), Label Powerset, Prued Sets (PS), Ensemble of Pruned Sets (EPS), Ensemble of Classifier Chains (ECC), Random-k-LabelSets (RAkEL), Hierarchy Of Mul-tilabel classifiERs (HOMER) and Stacking. The MULAN software for MLC [4] has been used for running both the feature selection method and the MLC algorithms. We have used a 10-fold cross validation with 10 seeds. Our experimentation takes into consideration two main factors: number of attributes and MLC performance. Overall, the time employed by a MLC algorithm to generate a model will be proportional to the number of training instances and the number of attributes describing each instance. So, if we reduce the number of attributes then the computational cost will be reduced as well. However, as a reduction of the number of attributes could discard relevant information, the induced model could perform poorly. This is why we have performed an attribute selection with different reduction levels in order to determine the more suitable reduction level without damaging the classification performance. Our original data set contains 515 LO instances, each one characterized by 1336 attributes. From these, we have selected 1000, 750, 500, 250, 150, 100 and 50 attributes with highest ranking to create different datasets. Next, we have applied 13 MLC algorithms to each different version of the data set, in order to know if there are differences in computational costs and performance by checking some evaluation measures. Therefore, in addition to train time the next five multi-label evaluation measures have been computed: a) Example-based metrics: Hamming loss (H-loss) and Accuracy (E-Acc) b) Label-based measures: Accuracy (L-Acc) and c) Ranking-based measures: Ranking loss (R-loss) and Average precision (A-Pre). On the one hand, we have found a significant reduction of computational costs as the number of features decrease (Figure 2), especially up to 250 features. The algorithms reducing training time at higher degrees are ECC, RAkEL and EPS.

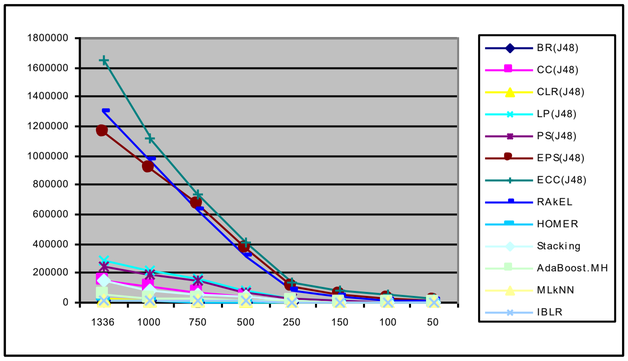

Figure 2. Training time (milliseconds).

On the other hand, in order to compare the classification performance of the algorithms, a Friedman test has been carried out for each evaluation metric by considering results for each feature reduction level. Ranking values and p-values are detailed in Table 1. These p-values (≤ 0,05) show significant differences between reduction levels with high confidence level (95%). We can also observe that for Ranking loss (R-loss) and Average Precision (A-Pre), the best ranking value is obtained for 1000 features instead of the original 1336 features. Besides, a meta-ranking (the rank of rank) of reduction levels was built performing another Friedman test. This way we can evaluate which number of features has the best overall performance in most of the metrics. The last column of Table 1 shows the resulting meta-rank. It is interesting to see that the best ranking does not correspond to the complete feature set. As the test detected significant differences between reduction levels (p-value ≤ 0,01), a Bonferroni-Dunn test was performed. This test found that algorithms performed significantly worst with less than 250 attributes at 95% confidence level. So, we established 250 as the optimum reduction level.

Table 1. Avg. rankings for all metrics and reduction levels.

Number Features	↓H-loss	↑E-Acc	↑L-Acc	↓R-loss	↑A-Pre	Meta Rank
1336	**2,92**	**3,07**	**2,92**	4,50	4,19	2,60
1000	3,76	3,23	3,76	**3,11**	**3,11**	**2,40**
750	3,11	3,57	3,11	3,88	3,42	2,80
500	2,96	3,34	2,96	4,42	3,76	2,80
250	4,19	3,96	4,19	3,96	3,88	4,40
150	5,73	5,57	5,73	4,88	5,46	6,00
100	6,50	6,50	6,50	5,96	6,23	7,40
50	6,80	6,73	6,80	5,26	5,92	7,60
p-values	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001

Finally, a comparison of 13 MLC algorithms when using the optimum reduction level (250 features) has been performed. The goal was to identify which algorithm yields the best results in this specific dataset considering the previous 5 evaluation metrics. The algorithm with the overall best results in the five evaluation measures (higher in E-Acc, L-Acc and A-Pre; and lower in H-Loss and R-Loss) was RAkEL. So, this algorithm will be used in our proposed approach for recommending the categories to which the new LOs belong. In the future we want to use more evaluation measures and also information about LO usage in order to try to improve classification performance.

4. ACKNOWLEDGMENTS

Authors gratefully acknowledge the financial subsidy provided by Spanish Ministry of Science and Technology TIN2014-55252-P.

5. REFERENCES

[1] Gibaja, E., Ventura, S. 2014. Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4, 6, 411-444.

[2] Kannampallil, T. G., Farrell, R. G. 2005. Automatic Learning Object Categorization For Instruction Using An Enhanced Linear Text Classifier. *Knowledge Management: Nurturing Culture, Innovation, and Technology*, 299-304.

[3] Menéndez, V., Prieto, M., Zapata, A. 2010. Sistemas de gestión integral de objetos de aprendizaje, *Revista Iberoamericana de Tecnologias del Aprendizaje*, 5, 2, 56-62.

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

[5] G. Tsoumakas, I. Katakis, I. Vlahavas. 2011. Random k-labelsets for multilabel classification", *IEEE Transactions on Knowledge and Data Engineering*, 23, 7, 1079-108.