Combining Meta-learning and Active Selection of Datasetoids for Algorithm Selection

Ricardo B.C. Prudêncio[1], Carlos Soares[2], and Teresa B. Ludermir[1]

[1] Center of Informatics, Federal University of Pernambuco,
Cidade Universitária - CEP 50732-970 - Recife (PE) - Brazil
{rbcp,tbl}@cin.ufpe.br
[2] LIAAD-INESC Porto L.A., Faculdade de Economia, Universidade do Porto
Rua de Ceuta 118-6, 4050-190, Porto, Portugal
csoares@fep.up.pt

Abstract. Several meta-learning approaches have been developed for the problem of algorithm selection. In this context, it is of central importance to collect a sufficient number of datasets to be used as meta-examples in order to provide reliable results. Recently, some proposals to generate datasets have addressed this issue with successful results. These proposals include datasetoids, which is a simple manipulation method to obtain new datasets from existing ones. However, the increase in the number of datasets raises another issue: in order to generate meta-examples for training, it is necessary to estimate the performance of the algorithms on the datasets. This typically requires running all candidate algorithms on all datasets, which is computationally very expensive. One approach to address this problem is the use of active learning, termed active meta-learning. In this paper we investigate the combined use of active meta-learning and datasetoids. Our results show that it is possible to significantly reduce the computational cost of generating meta-examples not only without loss of meta-learning accuracy but with potential gains.

Keywords: Meta-learning, Active learning.

1 Introduction

A large number of learning algorithms are available for data analysis nowadays. For instance, decision trees, neural networks, support vector machines, among others, can be used in classification problems. After narrowing down the list of candidate algorithms taking into account problem-specific constraints (e.g., interpretability of the model), the goal of data analysts is to select the algorithm with higher chances to obtain the best performance on the problem at hand. The algorithm selection problem is addressed by *meta-learning* as a supervised learning task [3]. A learning algorithm is used to model the relation between the characteristics of learning problems (e.g., number of examples, proportion of symbolic attributes) and the relative performance of a set of algorithms.

An important issue in the development of meta-learning systems for algorithm recommendation is the computational cost of generating the meta-data [3]. This

E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part I, LNAI 6678, pp. 164–171, 2011.

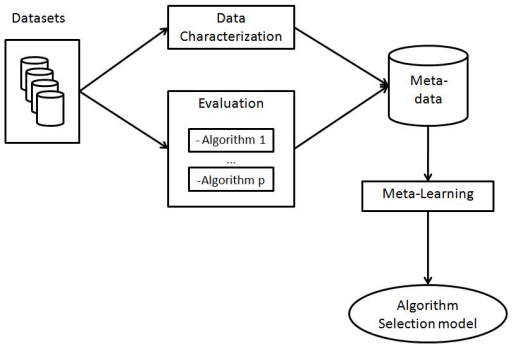

Fig. 1. The meta-learning process for algorithm selection (adapted from [3])

implies running the candidate algorithms on all the training datasets, which can be quite costly. In this paper, we address this problem using a hybrid approach, that combines two machine learning approaches [6]: an active learning approach with a dataset generation method, datasetoids [14], to guide the meta-data collection process. This combination is tested on an algorithm selection task. The contribution of this work is to show that these two methods simultaneously and successfully address two of the major issues in meta-learning: obtaining sufficient datasets for reliable meta-learning and reducing the computational cost of collecting meta-data.

We start by describing background information on meta-learning, including datasetoids, (Section 2) and active learning (Section 3). Next, we present the experimental setup used to evaluate the approach empirically (Section 4). We close the paper with conclusions and some ideas for future work (Section 5).

2 Meta-learning for Algorithm Selection

The meta-learning approach to algorithm selection is summarized in Figure 1. A database is created with meta-data descriptions of a set of datasets. These meta-data contain estimates of the performance of a set of candidate algorithms on those datasets as well as meta-features describing their characteristics (e.g., number of examples in the dataset, entropy of the class attribute, mean correlation between the numerical attributes). A machine learning algorithm (the so-called *meta-learner*) is applied to this database to induce a model that relates the values of the meta-features to the performance of the candidate algorithms (e.g., the best algorithm on the datasets). For more information on meta-learning for algorithm recommendation, the reader is referred to [3] and references therein.

An important issue in meta-learning is the availability of a sufficient number of datasets to enable reliable (meta-)induction. The UCI Repository [1] is the most common source of examples for meta-learning, however it contains slightly over 100 classification datasets. Given that each dataset represents one meta-example, most meta-learning research is based on approximately 100 meta-examples. This

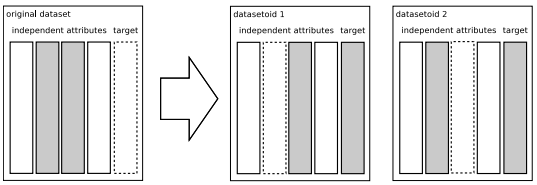

Fig. 2. Illustration of the generation of two classification datasetoids from a dataset with two symbolic attributes (reproduced from [14])

is a small number to obtain reliable models, particularly in such a complex application such as meta-learning. This problem is receiving an increasing amount of attention recently. Two common approaches are the generation of synthetic datasets and the manipulation of existing ones [7,2,9].

In this work we use *datasetoids*, a very simple data manipulation approach to generate new datasets which was recently proposed [14]. A datasetoid is generated from a given dataset by switching the target attribute with an independent attribute (Figure 2). Thus, the target attribute of the dataset becomes an independent attribute in the datasetoid and the independent attribute selected in the dataset becomes the target attribute in the datasetoid. To generate datasetoids for classification, the process is repeated for every symbolic attribute of the dataset, thus creating as many datasetoids as there are symbolic attributes in the dataset. Experiments on the problem of deciding whether to prune a decision tree using meta-data containing datasetoids obtained significant improvements when compared to meta-data that only contained datasets [14].

3 Active Learning and Meta-learning

Active learning is a paradigm of Machine Learning in which the learning algorithm has some control over the examples on which it trains [5]. It has been used in many tasks to reduce the number of training examples, while maintaining (or even improving) the learning performance [8,11,12,13]. Active learning is ideal for domains in which the acquisition of labeled examples is a costly process. The cost of acquiring labels for meta-learning is computationally expensive, as it is necessary to execute the candidate algorithms on the datasets used for training. This makes meta-learning a good candidate problem for active learning.

In [10], active learning is proposed to improve the generation of meta-examples. This proposal, termed as *Active Meta-learning*, is illustrated in Figure 3. The method starts with a small set of labeled examples and a large set of unlabeled ones. An active learning module receives these two sets as input and selects, from the latter, the next example to be labeled. The selection of unlabeled meta-examples is performed based on a pre-defined criterion which takes into account the meta-features of the problems and the current set of labeled examples. Labeling is done by evaluating the candidate algorithms on the selected problem and the best algorithm becomes the label of the corresponding meta-example. The

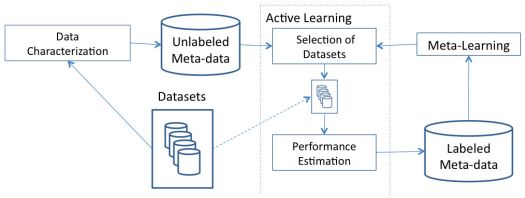

Fig. 3. Active meta-learning process

process iterates until some stopping condition. The algorithm selection model is then obtained by meta-learning on the labeled examples.

The Active Meta-learning was empirically evaluated in [10] by using an *uncertainty sampling method* originally proposed in [8] for the k-NN algorithm. This method selects unlabeled examples for which the current k-NN learner has high uncertainty in its prediction. The uncertainty of k-NN was defined in [8] as the ratio of: (1) the distance between the unlabeled example and its nearest labeled neighbor; and (2) the sum of the distances between the unlabeled example and its nearest labeled neighbor of every class. A high value of uncertainty indicates that the unlabeled example has nearest neighbors with similar distances but conflicting labeling.

In the context of meta-learning, let E be the set of labeled meta-examples. Let $\mathcal{C} = \{c_1, \ldots, c_L\}$ be the domain of the class attribute C, with L possible class labels, representing the set of candidate algorithms. Each labeled meta-example e_i is represented as the pair $(\mathbf{x}_i, C(e_i))$ storing: (1) the description $\mathbf{x}_i$ of the problem e_i, where $\mathbf{x}_i = (x_i^1, \ldots, x_i^m)$ is a vector of m meta-features; and (2) the class attribute C associated to e_i, i.e., $C(e_i) = c_l$, where $c_l \in \mathcal{C}$.

Let $\widetilde{E}$ be the set of unlabeled meta-examples. Let E_l be the subset of labeled meta-examples associated to the class label c_l, i.e., $E_l = \{e_i \in E | C(e_i) = c_l\}$. Given E, the classification uncertainty of k-NN for each $\widetilde{e} \in \widetilde{E}$ is defined as:

$$\mathcal{S}(\widetilde{e}|E) = \frac{\min_{e_i \in E} dist(\widetilde{\mathbf{x}}, \mathbf{x}_i)}{\sum_{l=1}^{L} \min_{e_i \in E_l} dist(\widetilde{\mathbf{x}}, \mathbf{x}_i)}. \quad (1)$$

where $\widetilde{\mathbf{x}}$ is the description of the unlabeled meta-example $\widetilde{e}$ and *dist* is the distance function adopted by the k-NN algorithm. The unlabeled meta-examples in $\widetilde{E}$ are selected according to the following probabilities:

$$p(\widetilde{e}) = \frac{\mathcal{S}(\widetilde{e}|E)}{\sum_{\widetilde{e}_i \in \widetilde{E}} \mathcal{S}(\widetilde{e}_i|E)}. \quad (2)$$

The above probability is just a normalized value of the uncertainty. The roulette-wheel algorithm is often used to sample the unlabeled meta-examples according to their associated probabilities. In this method, the probability of a meta-example being sampled is proportional to its uncertainty. The selected meta-example is labeled (i.e., the class value $C(\widetilde{e})$ is defined) by estimating the

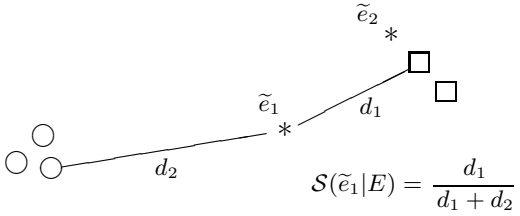

Fig. 4. Illustration of Uncertainty Sampling

performance of the candidate algorithms on the corresponding dataset. Finally, the new labeled meta-example $(\widetilde{\mathbf{x}}, C(\widetilde{e}))$ is then included in the meta-data.

Figure 4 illustrates the uncertainty sampling method. Circles and squares represent two classes of labeled examples. The stars named as $\widetilde{e}_1$ and $\widetilde{e}_2$ represent two unlabeled examples which are candidates to be labeled. The example $\widetilde{e}_2$ would be less relevant since it is very close to the labeled examples of one specific class. In our method, the $\widetilde{e}_1$ has a higher probability to be sampled since it is more equally distant from labeled examples of different classes.

4 Experiments and Results

Meta-data. We empirically evaluate the proposed approach exactly on the same meta-learning task used in [14]. It consists of predicting, a priori, if pruning a decision tree will improve the quality of the model or not. There are three classes, `p`, `u` or `t`, meaning, respectively, the winner is the pruned tree, the unpruned tree or that they are tied.

Concerning the meta-features to characterize datasets (and datasetoids), we used (1) the class entropy and (2) the average entropy of the attributes [4]. These meta-features are expected to contain some information about the behavior of decision trees because this algorithm uses the concept of entropy. But, most importantly, although there are certainly other meta-features that could contribute to improve the meta-learning results, the use of measures that previously obtained good results enables us to focus on the main goal of this paper, which is to test the combination of active meta-learning and datasetoids.

The set of problems used to generate meta-examples are 64 UCI classification datasets. By swapping the target attribute with every nominal attribute, 983 datasetoids were obtained. Table 1 presents the class distribution, both in the meta-data obtained from datasets as in the meta-data obtained from datasetoids.

Meta-level performance estimation. Given that the datasetoids are generated from datasets, they cannot be treated as independent problems. In other words,

Table 1. Class distribution (%) of the meta-data (for datasets and datasetoids)

metadata	pruned tree wins (`p`)	unpruned tree wins (`u`)	tie (`t`)
datasets	36	23	41
datasetoids	37	10	53

datasetoids generate meta-examples which are not independent from the meta-examples representing the corresponding datasets. Therefore, to estimate meta-learning performance, we adopted the same methodology as in [14], which is based on the following principles:

- predicting which algorithm is the best on the datasetoids is not relevant. As stated above, datasetoids are not interesting as applications *per se*. Therefore, only the original UCI datasets are used as test set.
- to ensure independence between the training and test sets, we must guarantee that the datasetoids generated from the test datasets are not included in the training set. Thus, the meta-examples corresponding to the datasetoids obtained from test datasets are removed from the training meta-data.

By removing the datasets which are obtained from test datasets, the amount of meta-data available for learning can reduce significantly. To minimize this, we use a leave-one-out (LOO) approach as in [14], which means that, at each iteration, a single dataset is used as test and the corresponding datasetoids are removed from the training set. The measure of meta-learning performance is the classification accuracy.

Active Meta-learning Setting and Baseline. Every experiment starts with a single labeled meta-example which is selected randomly. Then, we allow the active meta-learning method to sample and label up to 500 training meta-examples (about 50% of the available candidate problems). Given that the methods has two random components (selection of the first labelled example and roulette wheel), we repeat the experiments 100 times to reduce the variance of the results. The k-NN is run with a single neighbor, $k = 1$. Good results have been obtained in previous meta-learning experiments with the k-NN algorithms [4].

As a basis for comparison, we tested the Random Sampling method for selecting unlabeled problems (also repeated 100 times). Despite its simplicity, the random method has the advantage of performing a uniform exploration of the example space [8]. Another reference is the execution of the meta-learning method without active learning. In this case, we compare the active meta-learning with two results obtained by applying the meta-learning method on two different training sets: only datasets and the set of datasets and datasetoids together. The accuracies of those methods were 52% and 58%, respectively.

Results. Figure 5 presents the average curves of accuracy obtained by the two methods, uncertainty sampling and the random sampling, as well as the line representing the accuracy of meta-learning with all the meta-data. For both methods the accuracy of the k-NN meta-learner increases as the number of labeled meta-examples increases. However, the curve of the active meta-learning method is clearly above the random method. This means that the active learning method is able to identify the examples that are really helpful for the meta-learner.

When compared to the traditional meta-learning approach, the active meta-learning method is able to achieve the same level of accuracy that was obtained by meta-learning with all datasets and datasetoids (58%) using only 245 labeled meta-examples. This is less than 30% of the available problems, which represents

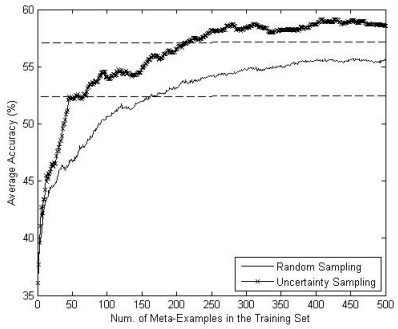

Fig. 5. Average accuracy obtained by uncertainty and random sampling. The dashed lines represent the accuracy obtained using all datasets and datasetoids (top) or just the datasets (bottom).

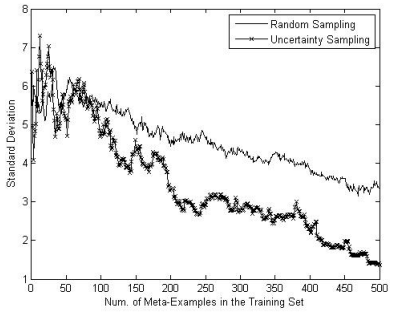

Fig. 6. Standard deviation of the accuracy obtained by uncertainty and random sampling

a significant gain in computational cost. Another interesting observation can be made. The active meta-learning can achieve an accuracy that is higher than the traditional meta-learning using all meta-examples. This can be explained by the fact that datasetoids are expected to have some noise [14]. These results indicate that the active meta-learning method may be also avoiding those examples.

To analyze the robustness of the methods, we computed the standard-deviation of their accuracies. Figure 6 shows that the standard-deviation of the two methods is similar until labeling approximately 100 meta-examples but then the uncertainty method has lower values. This indicates that, not only active meta-learning is more accurate than random sampling, as it is also more robust.

5 Conclusion

We proposed the combination of Active Meta-Learning and datasetoids to simultaneously address two important issues of meta-learning for algorithm selection: augmenting the number of datasets to produce meta-examples and reducing the computational cost of collecting meta-data. Our results show that it is possible to take advantage of the large number of meta-examples provided by datasetoids without incurring into significant extra computational costs. Additionally, if sufficient resources are available, it is even possible to achieve improvements, possibly due to the elimination of irrelevant and misleading meta-examples.

These results were obtained with a simple active learning method with known limitations (e.g., sensitivity to outliers). This opens up a number of possibilities for improvement, by using more complex active learning methods, possibly adapting them for meta-learning. Finally, we point out that we need to test the approach on other meta-learning problems, with different learning problems (e.g., regression), sets of base-level algorithms and meta-features.

Acknowledgements. The authors would like to thank CNPq, CAPES, FACEPE (Brazilian Agencies), FCT (Programa de Financiamento Plurianual de Unidades de I&D and project Rank! - PTDC/EIA/81178/2006) their financial support.

References

1. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
2. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 6–17. Springer, Heidelberg (2007)
3. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. Cognitive Technologies. Springer, Heidelberg (2009)
4. Brazdil, P., Soares, C., da Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Mach. Learn. 50(3), 251–277 (2003)
5. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine Learning 15, 201–221 (1994)
6. Corchado, E., Abraham, A., de Carvalho, A.: Editorial: Hybrid intelligent algorithms and applications. Inf. Sci. 180, 2633–2634 (2010)
7. Hilario, M., Kalousis, A.: Quantifying the resilience of inductive classification algorithms. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 106–115. Springer, Heidelberg (2000)
8. Lindenbaum, M., Markovitch, S., Rusakov, D.: Selective sampling for nearest neighbor classifiers. Machine Learning 54, 125–152 (2004)
9. Macià, N., Orriols-Puig, A., Bernadó-Mansilla, E.: Genetic-based synthetic data sets for the analysis of classifiers behavior. In: Proceedings of 15th International Conference on Hybrid Intelligent Systems, pp. 507–512 (2008)
10. Prudêncio, R.B.C., Ludermir, T.B.: Selective generation of training examples in active meta-learning. Int. J. of Hybrid Intelligent Systems 5, 59–70 (2008)
11. Raghavan, H., Madani, O., Jones, R.: Active learning with feedback on both features and instances. Pattern Recognition Letters 7, 1655–1686 (2006)
12. Riccardi, G., Hakkani-Tur, D.: Active learning - theory and applications to automatic speech recognition. IEEE Transactions on Speech and Audio Processing 13(4), 504–511 (2005)
13. Sampaio, I., Ramalho, G., Corruble, V., Prudêncio, R.: Acquiring the preferences of new users in recommender systems - the role of item controversy. In: Proceedings of the ECAI 2006 Workshop on Recommender Systems, pp. 107–110 (2006)
14. Soares, C.: UCI++: Improved support for algorithm selection using datasetoids. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) PAKDD 2009. LNCS, vol. 5476, pp. 499–506. Springer, Heidelberg (2009)