Concept of rule-based configurator for Auto-WEKA using OpenML

Patryk Kiepas, Szymon Bobek, and Grzegorz J. Nalepa

AGH University of Science and Technology,
al. A. Mickiewicza 30, 30-059 Krakow, Poland
kiepas@student.agh.edu.pl, {sbobek,gjn}@agh.edu.pl

Abstract. Despite a large amount of research devoted to improving meta-learning techniques, providing and using background knowledge for this task remains a challenge. In this paper we propose a mechanism for automatic recommendation of suitable machine learning algorithms and their parameters. We used OpenML database and use rule-based configurator to improve Auto-WEKA tool. This paper discusses the concept of our approach and the prototype tool based on the HEARTDROID rule engine being developed.

Introduction The objective of our work is to build a meta-learning recommendation system that guides a user through the process of solving a machine learning task. We use the data from OpenML's experiments to build a meta-knowledge which is later encoded with rules. This knowledge is then used for matching new dataset's meta-attributes with current meta-knowledge to obtain a set of possibly best algorithms. Finally, we use Auto-WEKA for optimizing the parameters of this narrowed set of algorithms.

In our approach we follow the general meta-learning architecture previously proposed by Pavel Brazdil et.al. [1]. We use data about machine learning from on-line collaborative platform known as OpenML[1]. In the creation of meta-knowledge we use the Amelia-II algorithm for imputation of missing data which could not be obtained with OpenML [2]. In rule-based configurator we take advantage of HEARTDROID inference engine[2]. Auto-WEKA does hyper-parameter optimization which we use for additional tuning of created recommendation [3].

We distinguish three phases in the recommendation mechanism: 1) knowledge acquisition, 2) recommendation, and 3) tuning. During the 1st phase meta-knowledge is built from OpenML's data only. In 2nd one the system uses that meta-knowledge and a new dataset to build a set of suitable algorithms. Finally an automatic configuration of these algorithms is performed with an usage of Auto-WEKA.

Building meta-knowledge In the acquisition phase main goal is to build meta-knowledge that describes dependecies between datasets and performance of machine learning algorithms executed on them. For every dataset in the OpenML database, a set of meta-attributes is available that includes: statistical information (e.g. number of classes and features, kurtosis of numeric attributes), information-theoretic characteristics (e.g. class

[1] http://www.openml.org/

[2] http://bitbucket.org/sbobek/heartdroid

or mean attribute entropy), and model-based information (e.g. J48 or kNN AUC). Each of such characteristics has a different non-missing value coverage that varies from 6.5% to 100%. We choose threshold for required values coverage to 20% to leave meaningful meta-attributes. Missing values are filled with Amelia-II algorithm [2].

Meta-knowledge combines meta-attributes from dataset characteristics with corresponding algorithm label or ranking. We choose only fixed number of algorithms that are taken into consideration (usually N top used in OpenML). After that we filter the results with respect to performance and leave only the set of best algorithms. Afterwards we consider meta-knowledge as labeled dataset. Using the WEKA J48 algorithm we create decision tree which is converts to the XTT2 rule representation (ang. *eXtended Tabular Trees*).

Making recommendation We start with computing meta-attributes of new dataset by uploading it to OpenML. Then we choose only characteristics used in created meta-knowledge. In the next step we match meta-attributes of new dataset with meta-knowledge. This is done with use of meta-rules and rule-based configurator. The result consist of algorithm name or ranking and set of parameters that according to the configurator fits best the given dataset.

In the third stage we reduce Auto-WEKA's search space only to the recommended algorithms. This is done by preparing experiment with so called XML-based *BATCH* file. In that file we fill path to our new dataset in *ARFF* format and set up list of allowed classifiers. Then we create an experiment and run optimization process. Result is in form of classifier name with single set of parameters.

Conclusion The main contribution of our work is a mechanism that allows to speed-up the meta-learning task by reducing search space for Auto-WEKA software with an usage of knowledge from OpenML database. We tested our approach and the tool on 570 datasets. We built meta-knowledge using 15 most used algorithms from OpenML focused on optimizing area under ROC. We benchmarked our best recommendations against Random Forest method as standard criteria. In general for most datasets area under ROC of our recommendations were higher (for 401 datasets with avg. 0.044). For 169 datasets AUC of our suggestions were lower (avg. 0.057). It is worth to notice that after a single setup, our system makes an instant recommendation.

Our future work includes learning and gaining additional meta-knowledge during recommendation mode, adding parameter suggestion in form of value ranges, adding guidance for data preprocessing methods and including more data sources.

References

1. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Meta-learning: Concepts and techniques. In: Metalearning: Applications to Data Mining. Springer Publishing Company, Incorporated, 1 edn. (2008)
2. Honaker, J., King, G., Blackwell, M.: Amelia II: A program for missing data. Journal of Statistical Software 45(7), 1–47 (12 2011), http://www.jstatsoft.org/v45/i07
3. Thornton, C., Hutter, F., Hoos, H., Leyton-Brown, K.: Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In: Proc. of KDD-2013. pp. 847–855 (2013)