

Research perspective: the role of automated machine learning in fuzzy logic

Radwa El Shawi, Stefania Tomasiello and Henri Liiva

Institute of Computer Science, University of Tartu, Tartu, Estonia

Abstract

In this short paper, we briefly discuss the potential of Automated Machine Learning (AutoML) in the context of fuzzy-based systems. The scarce presence of related works in the current literature shows that this process is just in the beginning. We will give an overview of AutoML and its investigated and possible use to enhance fuzzy-based systems.

Keywords

fuzzy-based systems, automated machine learning, hyperparameter optimization

1. Introduction

In general, the process of building a high-quality machine learning pipeline is an iterative, complex, and time-consuming process that requires solid knowledge and understanding of the various techniques that can be employed in each step of the pipeline. With the continuous and vast increase of the amount of data in our digital world, it has been acknowledged that the number of knowledgeable data scientists can not scale to address these challenges. Thus, there is a crucial need for automating the process of building good machine learning pipelines where the presence of a human in the loop can be dramatically reduced. Research in Automated Machine Learning (AutoML) aims to alleviate both the computational cost and human expertise required for developing machine learning pipelines through automation with efficient algorithms. In particular, AutoML techniques are enabling the widespread use of machine learning techniques by domain experts and non-technical users.


The use of AutoML in fuzzy logic is just in the beginning. There are very few papers in the literature covering such a topic. For instance, in [1], the joint use of fuzzy logic and AutoML allowed to get an enhanced intelligent tiering system for storage optimization. The proposed architecture in [1] consists of a fuzzy inference system to select the eligible tiers, satisfying some business criteria, and an AutoML component, such as Auto-SKLearn, to recommend a classification algorithm on the basis of a list previously formed. The study in [2] introduces a fuzzy-based active learning method for predicting students' academic performance, exploiting AutoML practices. The authors considered first six auto-configured representative fuzzy classifiers, by performing many experiments to select the most promising ones. Secondly, four active

WILF 2021: The 13th International Workshop on Fuzzy Logic and Applications, December 20–22, 2021, Vietri sul Mare, Italy

✉ radwa.elshawi@ut.ee (R. E. Shawi); stefania.tomasiello@ut.ee (S. Tomasiello); henri.liiva@ut.ee (H. Liiva)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

learning models were constructed, while AutoML was employed to automate the selection of the fuzzy machine learning models and their respective hyperparameters, using the three prevailing classifiers of the previous comparative study. AutoML and fuzzy C-means are discussed in [3] in the context of biomedical data analysis. The latter is a field attracting growing interest, due to the complex data sets. To this end, the authors discussed a model based on fuzzy c-means with meta-heuristic optimization. In this short paper, we will briefly discuss the potential of AutoML in the design of fuzzy systems.

2. AutoML: an overview

In general, the process of building a high-quality machine learning model is an iterative, complex and time-consuming process (Figure 1). In particular, a data scientist is commonly *challenged* with a large number of choices and design decisions needed to be taken. For example, the data scientist needs to select among a large variety of algorithms including classification or regression techniques (e.g. Support Vector Machines, Neural Networks, Bayesian Models, Decision Trees, etc) in addition to tuning large number of hyper-parameters of the selected algorithm. The performance of the model can be optimised by the metric of choice (e.g., accuracy, sensitivity, specificity, F1-score). Naturally, the decisions of the data scientist in each of these steps affect the performance and the quality of the developed model [4, 5, 6]. For instance, in yeast dataset, different parameter configurations of a Random Forest classifier result in different range of accuracy values with around 5% difference¹. Also, using different classifier learning algorithms leads to widely different performance values for the fitted model that reached 20% on the same dataset. Thus, making such decisions require experts' knowledge. However, in practice, increasingly, users of machine learning tools are non-expert ones who require *off-the-shelf* solutions. Therefore, there has been a growing interest to automate the steps of building the machine learning pipelines.

Therefore, several frameworks have been designed to support automating the Combined Algorithm Selection and Hyper-parameter tuning (CASH) problem [7, 8]. These techniques have commonly formulated the problem as an optimization problem that can be solved by wide range of techniques. Let A denote a machine learning algorithm with N hyperparameters. We denote the domain of the N -th hyperparameter by Λ_n and the overall hyperparameter configuration space as $\Lambda = \Lambda_{(1)} \times \Lambda_{(2)} \times \dots \times \Lambda_{(N)}$. A vector of hyperparameters is denoted by λ , and A with its hyperparameters instantiated to λ is denoted by A_λ . Given a dataset D , the CASH problem aims to find

$$A_{\lambda^*}^* \in \underset{A^* \in \mathbf{A}, \lambda \in \Lambda}{\operatorname{argmin}} L(A, D_{train}, D_{valid})$$

where $L(A, D_{train}, D_{valid})$ measures the loss of a model generated by algorithm A with hyperparameters λ on training data D_{train} and evaluated on validation data D_{valid} .

In the following, we review some of the-state-of-art frameworks for AutoML.

¹<https://www.openml.org/t/2073>

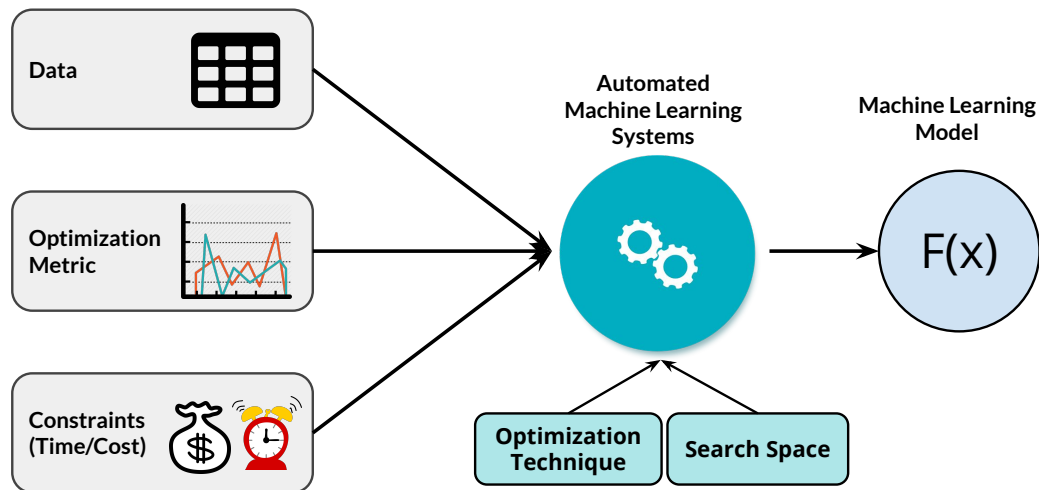


Figure 1: The general Workflow of the AutoML process.

2.1. AutoWeka

*Auto-Weka*² is considered the earliest open source automated machine learning framework with a basic GUI [9]. It was implemented in Java on top of *Weka*³, a popular machine learning library that offers an extensive suite of machine learning methods. *Auto-Weka* applies Bayesian optimization using Sequential Model-based Algorithm Configuration (SMAC) [10] as default optimizer but also support tree-structured parzen estimator (TPE) as optimizer for both algorithm selection and hyper-parameter optimization. The main advantage of using SMAC is its robustness by having the ability to discard low performance parameter configurations quickly after the evaluation on a low number of dataset folds. SMAC shows better performance on experimental results compared to TPE [10]. *Auto-Weka* tackles classification and regression type of machine learning problems.

2.2. Auto-Sklearn

*Auto-Sklearn*⁴ [11] has been implemented on top of *Scikit-Learn*⁵, a popular Python machine learning package. *Auto-Sklearn* introduced the idea of meta-learning in algorithm selection and hyper-parameter tuning instantiations and used SMAC as a Bayesian optimization technique. In addition, ensemble methods were used to improve the performance of output models. Both meta-learning and ensemble methods improved the performance of vanilla SMAC optimization.

²<https://www.cs.ubc.ca/labs/beta/Projects/autoweka/>

³<https://www.cs.waikato.ac.nz/ml/weka/>

⁴<https://github.com/automl/auto-sklearn>

⁵<https://scikit-learn.org/>

2.3. TPOT

*TPOT*⁶ framework represents another type of solution [12] that is implemented on top of *Scikit-Learn* as its backend to automatically find a machine learning pipeline. It is based on genetic programming by exploring many different possible pipelines of feature engineering and learning algorithms. Then, it finds the best one out of them. TPOT tackles both classification and regression problems and allows the user to restrict the optimization search space in terms of feature transformers, machine learning model and their hyper-parameters. TOPT also is scalable as it supports distributed computation through *dask*⁷.

2.4. Recipe

Recipe [13] follows the same optimization procedure as TPOT using genetic programming, which in turn exploits the advantages of a global search. However, Recipe considers the unconstrained search problem in TPOT, where resources can be spent into generating and evaluating invalid solutions by adding a grammar that avoids the generation of invalid pipelines, and can speed up optimization process. Second, it works with a bigger search space of different model configurations than Auto-SkLearn and TPOT.

3. AutoML and fuzzy logic: application, challenges and perspectives

The potential of AutoML to select a proper fuzzy classifier has been recently shown [2]. The performance of most computing schemes (both for classification and prediction) depends on the values assigned to their hyperparameters. Hyperparameters have usually a varying degree of complexity and dimension which are difficult to set. Hyperparameters optimization is one of the tasks that AutoML foresees. There are some examples in literature of hyperparameters optimization in the context of fuzzy-based systems. Major hyperparameters which affect the efficiency of fuzzy-based systems are related to the partitioning of the universe of discourse (e.g. length, type). For instance, in [14] an evolutionary hyperparameter optimization for Weighted Multivariate Fuzzy Time Series method was proposed. In [15], particle swarm optimization of partitions and fuzzy order for fuzzy time series forecasting was discussed. In [16], the evidence framework is applied to optimize the hyperparameters of Fuzzy HyperSphere Support Vector Machine. An attempt to find the best fuzzy partition to enhance the performance of Fuzzy Transforms is offered in [17]. All these examples are not in the context of AutoML, which instead could provide a better framework to select the most suitable fuzzy-based system.

It must be mentioned that existing AutoML systems and tools mainly target the domain of supervised learning. Unsupervised learning, in particular clustering, would also benefit from AutoML solutions (e.g. for the evaluation of clustering results). To address such issue, in [18], a framework for automated clustering, encompassing algorithm selection and hyperparameter tuning, was proposed. A similar tool would be valuable to support the best choice of a clustering method in neuro-fuzzy systems [19].

⁶<https://automl.info/tpot/>

⁷<https://dask.org/>

There is in literature an example of unsupervised fuzzy rule-based system, not using clustering. It is a self-developing (evolving) fuzzy-rule-based classifier system, called AutoClass [20]. AutoClass learns without specifying number of rules and number of classes, by using data clouds. Data clouds are subsets of given data samples with common properties. Although the concepts of data clouds and traditional clusters are different, since in the first case, there are no well-defined boundaries, the authors use a measure of zone of influence of a data cloud. This is the only user-defined parameter. While this scheme deserves attention, it would be interesting to compare its performance against the one of a fuzzy inference system built with the support of AutoML.

4. Conclusions

Choosing the proper method for a given problem formulation and configuring the optimal parameter setting is a demanding task. AutoML can help in this regard. The performance of most fuzzy-based systems is affected by the choice of the partition. In some unsupervised neuro-fuzzy systems, the choice of a proper clustering method is critical. The application of AutoML to such problems would be beneficial, as shown by very few examples in the current literature.

Acknowledgments

Stefania Tomasiello and Henri Liiva acknowledge support from the European Social Fund through the IT Academy Programme. The work of Radwa El Shawi is funded by the European Regional Development Funds via the Mobilitas Plus programme (grant MOBTT75)

References

- [1] Batrouni, M., A Hybrid Architecture for Tiered Storage with Fuzzy Logic and AutoML, Y. Luo (Ed.): CDVE 2020, LNCS 12341, pp. 67–74, 2020.
- [2] Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., Ragos, O., Fuzzy-based active learning for predicting student academic performance using autoML: a step-wise approach, *Journal of Computing in Higher Education*, 2021, <https://doi.org/10.1007/s12528-021-09279-x>
- [3] Nayak, J., Naik, B., Dash, P.B., Pelusi, D., Optimal fuzzy cluster partitioning by crow search meta-heuristic for biomedical data analysis, *International Journal of Applied Metaheuristic Computing* 12(2), 49-66, 2021
- [4] Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., Chatzisavvas, K., A comparison of machine learning techniques for customer churn predictio, *Simulation Modelling Practice and Theory*, 55, 1–9, 2015
- [5] Probst, P., Boulesteix, A., To Tune or Not to Tune the Number of Trees in Random Forest, *Journal of Machine Learning Research*, 18, 181–191, 2017
- [6] Pedregosa, F. et al., Scikit-learn: Machine learning in Python, *Journal of machine learning research*, 12, 2825–2830, 2011

- [7] El Shawi, R., Maher, M., Sherif, S., Automated Machine Learning: State-of-The-Art and Open Challenges, 2019, <http://arxiv.org/abs/1906.02287>
- [8] He, X., Zhao, K., Chu, X., AutoML: A Survey of the State-of-the-Art, 2019, arXiv preprint arXiv:1908.00709
- [9] Kotthoff, L. et al., Auto-WEKA 2.0: Automatic Model Selection and Hyperparameter Optimization in WEKA, *JMLR*, 18(1), 2017
- [10] Hutter, F., Hoos, H., Leyton-Brown, K., Sequential model-based optimization for general algorithm configuration, *International Conference on Learning and Intelligent Optimization*, 507–523, 2011
- [11] Feurer, M., Klein, A., Eggenberger, K., Springenberg, J.T., Blum, M. and Hutter, F., 2019. Auto-sklearn: efficient and robust automated machine learning. In *Automated Machine Learning* (pp. 113-134). Springer, Cham.
- [12] Randal S. O., Jason H. M., TPOT: A Tree-based Pipeline Optimization Tool for Automating Machine Learning, *Proceedings of the Workshop on Automatic Machine Learning*, 2016
- [13] de Sá, A.G., Pinto, W.J.G., Oliveira, L.O.V. and Pappa, G.L., 2017, April. RECIPE: a grammar-based framework for automatically evolving classification pipelines. In *European Conference on Genetic Programming* (pp. 246-261). Springer, Cham.
- [14] Silva, P.C.L., De Oliveira E Lucas, P., Sadaei, H.J., Guimaraes, F.G., Distributed Evolutionary Hyperparameter Optimization for Fuzzy Time Series, *IEEE Transactions on Network and Service Management* 17(3),9034097, 1309-1321, 2020
- [15] N. Kumar, S. Susan, Particle swarm optimization of partitions and fuzzy order for fuzzy time series forecasting of COVID-19, *Applied Soft Computing* 110, 107611, 2021
- [16] Tian, J., Zhimin, Z., Qian, S., Wenge, C., The evidence framework applied to fuzzy hypersphere SVM for UWB SAR landmine detection, *International Conference on Signal Processing Proceedings, ICSP 3*,4129197, 2007
- [17] Loia, V., Tomasiello, S., Troiano, L., Improving approximation properties of fuzzy transform through non-uniform partitions, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10147 LNAI, pp. 63-72, 2017
- [18] Poulakis, Y., Doulkeridis, C., Kyriazis, D., *Proceedings, AutoClust: A framework for automated clustering based on cluster validity indices*, *IEEE International Conference on Data Mining, ICDM 2020*,9338346, pp. 1220-1225, 2020
- [19] Pillai, G., Recent advances in neuro-fuzzy system: A survey, *Know. Bas. Syst.*, vol. 152, pp. 136–162, 2018
- [20] Costa, B. S. J., Angelov, P. P., Guedes, L. A. Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. *Neurocomputing*, 150, 289-303, 2015