A Unified Model Representation of Machine Learning Knowledge

J. G. Enríquez[1,*], A. Martínez-Rojas[1], D. Lizcano[2]
and A. Jiménez-Ramírez[1]

[1]*Computer Languages and Systems Department. Escuela Técnica Superior de Ingeniería Informática, Avenida Reina Mercedes, s/n, 41012, Sevilla. Spain*
[2]*Universidad a distancia de Madrid. Carretera de La Coruña, KM.38,500, vía de Servicio, no 15, 28400, Collado Villalba, Madrid. Spain*
E-mail: jgenriquez@us.es
[*]*Corresponding Author*

Received 20 December 2019; Accepted 14 April 2020;
Publication 03 June 2020

Abstract

Nowadays, Machine Learning (ML) algorithms are being widely applied in virtually all possible scenarios. However, developing a ML project entails the effort of many ML experts who have to select and configure the appropriate algorithm to process the data to learn from, between other things. Since there exist thousands of algorithms, it becomes a time-consuming and challenging task. To this end, recently, AutoML emerged to provide mechanisms to automate parts of this process. However, most of the efforts focus on applying brute force procedures to try different algorithms or configuration and select the one which gives better results. To make a smarter and more efficient selection, a repository of knowledge is necessary. To this end, this paper proposes (1) an approach towards a common language to consolidate the current distributed knowledge sources related the algorithm selection in ML, and (2) a method to join the knowledge gathered through this language in a unified store that can be exploited later on, and (3) a traceability links maintenance. The preliminary evaluations of this approach allow to create a

Journal of Web Engineering, Vol. 19_2, 319–340.
doi: 10.13052/jwe1540-9589.1929

unified store collecting the knowledge of 13 different sources and to identify a bunch of research lines to conduct.

Keywords: Machine Learning, Automated Machine Learning, Knowledge Representation, Model-Driven Engineering.

1 Introduction

Machine Learning (ML) entails the study of algorithms that automatically improve through experience [18]. This kind of algorithms has been successfully and broadly applied in the past [19] and nowadays is receiving increasing attention due to the affordable access to bigger computation power of machines.

A ML project requires selecting an appropriate algorithm to process the data to learn from, which is typically named *creating the data model*. However, there are thousands of algorithms under the paradigm of ML, each of them tailored to some specific tasks or contexts. In addition, many of these algorithms offer a different set of parameters to be configured (e.g., selecting the number of layers in a neural network).

Many existing approaches focus on the latter task, i.e., supporting the user after the algorithm selection is done, and few of them recommend an algorithm always after the user has provided the dataset. As an example, the recent research area of AutoML [28] aims to automate the different steps of ML projects. Nonetheless, such approaches neglect the early stages of the project. Many of them just provide a brute force mechanism that runs several algorithms in later stages of the project, i.e., when the dataset is ready. Thus, little effort has been done to support the user in the algorithm selection in an efficient manner (i.e., without applying brute force) and based on the problem characteristics (i.e., the early information).

The algorithm selection is specifically challenging since the existing knowledge regarding this task is distributed across different sources and each of them is specified in a non-standard manner, thus, making it difficult to consolidate information from different sources, i.e., the name of the algorithms —or family of algorithms—, the selection criteria, and the characteristics of the problem that affect the selection are heterogeneous (cf. Figure 1).

To reduce the risk of taking inaccurate decisions due to a lack of information, a central repository of the ML Knowledge which stores the information in a structured way is required. In order to address this problem, this paper proposes (cf. Figure 2), on the one hand, a unified language for representing