Neurocomputing 194 (2016) 45–55

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Active learning and data manipulation techniques for generating training examples in meta-learning

Arthur F.M. Sousa [a], Ricardo B.C. Prudêncio [a,*], Teresa B. Ludermir [a], Carlos Soares [b]

[a] Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil
[b] INESC TEC, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal

ARTICLE INFO

Article history:
Received 20 April 2015
Received in revised form
12 August 2015
Accepted 11 February 2016
Communicated by Shiliang Sun
Available online 24 February 2016

Keywords:
Meta-learning
Algorithm selection
Active learning

ABSTRACT

Algorithm selection is an important task in different domains of knowledge. Meta-learning treats this task by adopting a supervised learning strategy. Training examples in meta-learning (called meta-examples) are generated from experiments performed with a pool of candidate algorithms in a number of problems, usually collected from data repositories or synthetically generated. A meta-learner is then applied to acquire knowledge relating features of the problems and the best algorithms in terms of performance. In this paper, we address an important aspect in meta-learning which is to produce a significant number of relevant meta-examples. Generating a high quality set of meta-examples can be difficult due to the low availability of real datasets in some domains and the high computational cost of labelling the meta-examples. In the current work, we focus on the generation of meta-examples for meta-learning by combining: (1) a promising approach to generate new datasets (called *datasetoids*) by manipulating existing ones; and (2) active learning methods to select the most relevant datasets previously generated. The datasetoids approach is adopted to augment the number of useful problem instances for meta-example construction. However not all generated problems are equally relevant. Active meta-learning then arises to select only the most informative instances to be labelled. Experiments were performed in different scenarios, algorithms for meta-learning and strategies to select datasets. Our experiments revealed that it is possible to reduce the computational cost of generating meta-examples, while maintaining a good meta-learning performance.

1. Introduction

Algorithm selection is a challenging task in different domains related to computational intelligence, machine learning, optimization, among others. Such domains have in common the availability of different algorithms to solve the problems of interest and a shared statement that no single algorithm can be considered as the *best one* for all problems [1]. For instance, in a machine learning context, different algorithms can be alternatively adopted to solve classification problems, but the performance of the candidate algorithms can vary a lot depending on the features of the problems (e.g., dimensionality, training data quality, class complexity) and on the measures adopted for performance assessment. Additionally, each algorithm may have specific hyperparameters to set, which can also affect algorithm performance depending on the problem.

In this work, the algorithm selection problem was addressed by the *meta-learning* approach [2,3,1]. In meta-learning, algorithm selection is treated as a supervised learning task. Each training example (or *meta-example*) is related to a learning problem (e.g., a classification problem), the predictor attributes are features of that problem (e.g., class entropy, number of training examples, number of attributes) and the target attribute usually indicates the best algorithm for that problem, assigned after an empirical evaluation procedure (e.g., cross-validation). A *meta-learner* is adopted to select the best algorithms for new problems by exploiting the relationship between problem features and algorithm performance. A substantial amount of research in this topic was done in the context of the METAL project,[1] resulting in new meta-learning

* Corresponding author.
E-mail addresses: amfs@cin.ufpe.br (A.F.M. Sousa), rbcp@cin.ufpe.br (R.B.C. Prudêncio), tbl@cin.ufpe.br (T.B. Ludermir), csoares@fe.up.pt (C. Soares).

http://dx.doi.org/10.1016/j.neucom.2016.02.007
0925-2312/

[1] METAL: A Meta-Learning Assistant for Providing User Support in Machine Learning and Data Mining – ESPRIT Project – http://www.metal-kdd.org/.

procedures and problem characterization methods. In the last decade, meta-learning has been extrapolated for algorithm selection in a variety of other domains of knowledge [1], with promising results and new perspectives.

As any other learning task, the success of meta-learning depends on a good set of training instances (in our case, a good set of meta-examples). A large amount of previous work is focused on constructing and selecting relevant meta-features, but few papers are concerned with the instances (problems) used to generate the set of meta-examples. Ideally, meta-examples have to be generated from a representative and large enough set of problems in order to result in good meta-learning performance. However, in different domains, there is a low availability of real problem instances or benchmark datasets to produce a rich and large set of meta-examples [4]. In fact, in [1], the author reported meta-learning studies in some domains with very scarce sets of meta-examples. This issue has received attention of the research community by generating new problem instances from either synthetic or manipulated datasets [5–7].

A second issue that can be pointed out is related to the cost of generating meta-examples. In fact, in order to generate a meta-example from a given problem, an empirical evaluation of each candidate algorithm is performed on the available dataset. This is specifically related to the labelling process of a meta-example, which requires assigning the best candidate algorithm for that problem. The labelling process can lead to a high computational cost, for instance, in situations when a large pool of problem instances is available (real, synthetic problems or both) or when there is a pool of time consuming candidate algorithms to evaluate. Selecting only informative and non-redundant datasets is an important issue in meta-learning, which was addressed in [8] by deploying *active learning* techniques.

Motivated by the previous two issues, in our work we investigate the combination of manipulation approaches for generating datasets and active learning to support the selection of meta-examples. More specifically, in our proposal a previous approach for manipulating datasets, called *datasetoids* [7], is initially adopted to produce a large pool of problem instances. Following, active learning techniques based on uncertainty sampling are used to select from this pool only the most relevant problem instances, avoiding the generation of meta-examples from redundant or irrelevant problem instances. The goal of this combination is to address at the same time two challenges of meta-learning: obtaining a significant number of datasets for generating accurate meta-models and reducing the computational cost of collecting meta-data by actively selecting relevant problem instances.

Different aspects of our proposal were investigated: (1) alternative scenarios were considered to evaluate the usefulness of the datasetoids approach and also concerning how to integrate these data among the pool of real problem instances; (2) the selection of problem instances was accomplished by adopting an uncertainty sampling method based on *entropy*; (3) we also investigated the effect of peripherical instances in the performance of the uncertainty sampling method, which is a drawback already known in the literature of active learning [9]; (4) we adopted two different algorithms as meta-learner: the *k*-Nearest Neighbor (*k*-NN) (which has been a standard method in meta-learning [4]) and the Random Forest algorithm (specially motivated by its good comparative performance in the literature [10]).

The remaining of this paper is organized as follows. First, Section 2 provides some background on meta-learning, including a presentation of the datasetoids approach. Next, active learning is discussed in the context of meta-learning (Section 3). Section 4 presents the proposed solution. Section 5 presents the experiments and obtained results. Finally, Section 6 brings some conclusions and future work.

2. Meta-learning for algorithm selection

Based on the Rice's framework [12], reproduced in [1], the algorithm selection problem can be defined by considering four components: (1) a problem space *P*, which represents the possible instances related to a particular problem of interest (e.g., classification problems); (2) the feature space *F*, which defines the features adopted to describe the problem instances (e.g., number of training examples and number of classes); (3) the algorithm space *A*, which defines a set of candidate algorithms that the user will consider to solve the problem of interest (e.g., decision trees, *k*-NN and naive bayes classifier); and (4) the performance space *Y*, which represents the performance measures deployed to evaluate the candidate algorithms (e.g., accuracy rate estimated by cross-validation). Algorithm selection in the Rice's framework aims to find a suitable model that maps problem features into the algorithm space, in such a way that the chosen performance measures are optimized for each problem instance at hand.

Meta-learning treats algorithm selection by adopting a supervised learning strategy, summarized in Fig. 1. The meta-data is generated from experiments conducted to evaluate the candidate algorithms *A* in a subset of problem instances of *P*. For each problem instance considered, a single meta-example is derived by storing: (1) the meta-features *F* describing the characteristics of the problem; and (2) a target attribute usually indicating the best candidate algorithm for that problem according to a desired performance measure in *Y*. A machine learning algorithm (the so-called *meta-learner*) is applied to the meta-data to induce a model that relates meta-features to the best algorithms. The derived model is then adopted to predict the best algorithm for new problem instances not previously seen.

Meta-learning has been more deeply investigated to select algorithms for classification and regression tasks, although it has also been applied to algorithm recommendation in several domains of applications, including time series model selection and data streams [13–15], clustering [16], gene expression analysis [17,18] and optimization problems [19,20]. For more information about meta-learning for algorithm recommendation, the reader is referred to [3] and references cited therein.

Extensive research has been done specially to define suitable features to describe problems and to produce meta-learners with new functionalities. Our research focuses on a different issue, which is not always carefully considered in previous work: the generation of the meta-examples. This issue is important since the success of meta-learning is strongly dependent on a sufficient number of problem instances that are at the same time relevant and non-redundant.

The most natural strategy to meta-data acquisition is to produce it from real problem instances related to real-world

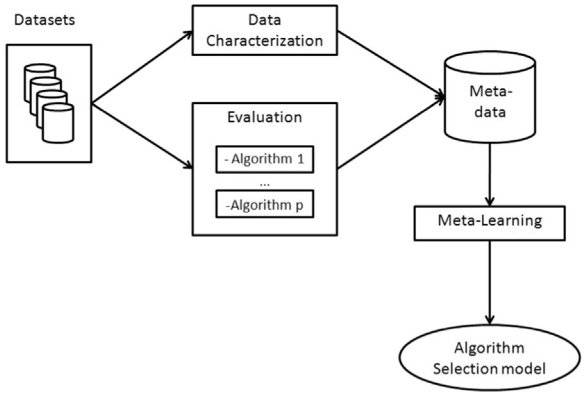

Fig. 1. The Meta-Learning process for algorithm selection (adapted from [3]).