A Framework To Decompose And Develop Metafeatures

Fábio Pinto[1] and **Carlos Soares**[2] and **João Mendes-Moreira**[3]

Abstract.

This paper proposes a framework to decompose and develop metafeatures for Metalearning (MtL) problems. Several metafeatures (also known as data characteristics) are proposed in the literature for a wide range of problems. Since MtL applicability is very general but problem dependent, researchers focus on generating specific and yet informative metafeatures for each problem. This process is carried without any sort of conceptual framework. We believe that such framework would open new horizons on the development of metafeatures and also aid the process of understanding the metafeatures already proposed in the state-of-the-art. We propose a framework with the aim of fill that gap and we show its applicability in a scenario of algorithm recommendation for regression problems.

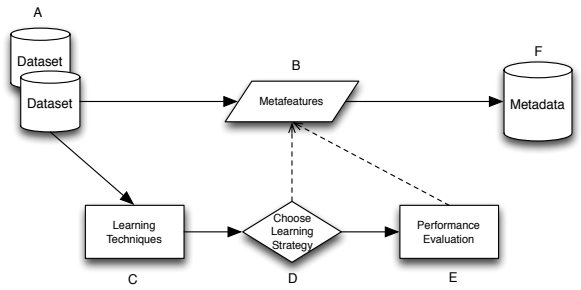

Figure 1. Metalearning: knowledge acquisition. Adapted from [2]

1 Introduction

Researchers have been using MtL to overcome innumerous challenges faced by several data mining practitioners, such as algorithm selection [3][23], time series forecasting [9], data streams [19][20][5], parameter tuning [22] or understanding of learning behavior [6].

As the study of principled methods that exploit metaknowledge to obtain efficient models and solutions by adapting machine learning and data mining processes [2], MtL is used to extrapolate knowledge gained in previous experiments to better manage new problems. That knowledge is stored as metadata, particularly, metafeatures and metatarget, as outlined in Figure 1. The metafeatures (extracted from *A* to *B* and stored in *F*) consist in data characteristics that describe the correlation between the learning algorithms and the data under analysis, *i.e.*, correlation between numeric attributes of a dataset. The metatarget (extracted through *C-D-E* and stored in *F*) represents the meta-variable that one wishes to understand or predict, *i.e.*, the algorithm with best performance for a given dataset.

Independently of the problem at hands, the main issue in MtL concerns defining the metafeatures. If the user is able to generate informative metafeatures, it is very likely that his application of MtL is going to be successful. The state-of-the-art shows that there is three types of metafeatures: 1) simple, statistical and information-theoretic. In this group we can found the *number of examples* of the dataset, *correlation between numeric attributes* or *class entropy*, to name a few. Application of these kind of metafeatures provides not only informative metafeatures but also interpretable knowledge about the problems [3] 2) model-based ones [13]. These capture some characteristic of a model generated by applying a learning algorithm to a dataset, *i.e.*, the *number of leaf nodes of decision tree*. Finally, a metafeature can also be a 3) *landmarker* [14]. These are generated by making a quick performance estimate of a learning algorithm in a particular dataset.

Although the state-of-the-art proposes several metafeatures of all types for a wide range of problems, we state that the literature lacks an unifying framework to categorize and develop new metafeatures. Such framework could help MtL users by systematizing the process of generating new metafeatures. Furthermore, the framework could be very useful to compare different metafeatures and assess if there is no overlap of the information that they capture. In this paper, we propose a framework with that purpose and we use it in the analysis of the metafeatures used in several MtL applications. We also show its applicability to generate metafeatures in a scenario of algorithm recommendation for regression problems.

The paper is organized as follows. In Section 2 we present a brief overview of MtL applications and respective metafeatures. Section 3 details the proposed framework to decompose and develop metafeatures. In Section 4 we use the framework to decompose and understand how our framework would characterize metafeatures already proposed in the literature. Section 5 exemplifies how the framework could be used to develop new metafeatures in a scenario of algorithm recommendation for regression problems. Finally, we conclude the paper with some final remarks and future work.

2 Metalearning

MtL emerges as the most promising solution from machine learning researchers to the need for an intelligent assistant for data analysis [21]. Since the majority of data mining processes include several non-trivial decisions, it would be useful to have a system that could guide the users to analyze their data.

[1] LIAAD-INESC TEC, Universidade do Porto, Portugal, e-mail: fhpinto@inesctec.pt
[2] CESE-INESC TEC, Universidade do Porto, Portugal, e-mail: csoares@fe.up.pt
[3] LIAAD-INESC TEC, Universidade do Porto, Portugal, e-mail: jmoreira@fe.up.pt

The main focus of MtL research has been the problem of algorithm recommendation. Several works proposed systems in which data characteristics were related with the performance of learning algorithms in different datasets. Brazdil et al. [3] system provides recommendations in the form of rankings of learning algorithms. Besides the MtL system, they also proposed an evaluation methodology for ranking problems that is useful for the problem of algorithm ranking. Sun and Pfahringer [23] extended the work of Brazdil et al. with two main contributions: the pairwise meta-rules, generated by comparing the performance of individual base learners in a one-against-one manner; and a new meta-learner for ranking algorithms.

Another problem addressed by MtL has been the selection of the best method for time series forecasting. The first attempt was carried by Prudêncio and Ludermir [16] with two different systems: one that was able to select among two models to forecast stationary time series and another to rank three models used to forecast time series. Results of both systems were satisfactory. Wang et al. [26] addressed the same problem but with a descriptive MtL approach. Their goal was to extract useful rules with metaknowledge that could aid the users in selecting the best forecasting method for a given time series and develop a strategy to combine the forecasts. Lemke and Bogdan [9] published a similar but with more emphasis on improving forecasts through model selection and combination.

MtL has also been used to tune parameters of learning algorithms. Soares et al. [22] proposed a method that by using mainly simple, statistical and information-theoretic metafeatures was able to predict successfully the width of the Gaussian kernel in Support Vector Regression. Results show that the methodology can select settings with low error while providing significant savings in time. Ali and Miles [1] published a MtL method to automatically select the kernel of a Support Vector Machine in a classification context, reporting results with high accuracy ratings. Reif et al. [17] used a MtL system to provide good starting points to a genetic algorithm that optimizes the parameters of a Support Vector Machine and a Random Forests classifier. Results state the effectiveness of the approach.

Data stream mining can also benefit from MtL, especially in a context where the distribution underlying the observations may change over time. Gama and Kosina [5] proposed a metalearning framework that is able to detect recurrence of contexts and use previously learned models. Their approach differs from the typical MtL approach in the sense that uses the base-level features to train the metamodel. On the other hand, Rossi et al. [19] reported a system for periodic algorithm selection that uses data characteristics to induce the metamodel (all the metafeatures are of the simple, statistical and information-theoretic type).

Another interesting application of MtL is to use it as a methodology to investigate the reasons behind the success or failure of a learning algorithm [25]. In this approach, instead of the typical predictive methodology, MtL is used to study the relation between the generated metafeatures and a metatarget that represents the base-level phenomenon that one wishes to understand. Kalousis et al. [6] published a paper on this matter. They adress the problem of discovering similarities among classification algorithms and among datasets using simple, statistical and information-theoretic metafeatures.

All the MtL applications that we mentioned previously use differents sets of metafeatures. It is mandatory to adapt the set of metafeatures to the problem domain. However, as stated previously, we believe that would be useful to decompose all these metafeatures into a common framework. Furthermore, such framework must also help the MtL user in the development of new metafeatures.

3 Metafeatures Development Framework

In this section, we propose a framework in order to allow a more systematized and standardized development of metafeatures for MtL problems. This framework splits the conception of a metafeature into four components: object, role, data domain and aggregation function. Within each component, the metafeature can be generated by using different subcomponents. Figure 2 illustrates the framework.

The object component concerns which information is going to be used to compute the metafeature. It can be an instance(s), dataset(s), model(s) or a prediction(s). The metafeature can extract information from one subcomponent (*i.e.*, class entropy of a dataset), several units of a subcomponent (*i.e.*, mean class entropy of a subset of datasets) and for some problems it might be useful to select multiple subcomponents (*i.e.*, for dynamic selection of models, one could relate instances with models [11]).

The role component details the function of the object component that is going to be used to generate the metafeature. The focus can be in the target variable, predicted or observed, in a feature or in the structure of the object component (*i.e.*, decision tree model or the representation of a dataset into a graph). Several elements can be selected, *i.e.*, the metafeature can relate the target variable with one or more features.

The third component defines the data domain of the metafeature and it is decomposed into four subcomponents: quantitative, qualitative, mixed or complex. This component is highly dependent of the previous ones and influences the metric used for computation (*i.e.*, if the data domain is qualitative, the user can not use correlation to capture the information). A metric can be quantitative (if the object component is numerical), qualitative (if the object component is categorical), mixed (if the object component has both numerical and categorical data) or complex (in special situations in which the object is a graph or a model).

Finally, the aggregation function component. Typically, this is accomplished by some descriptive statistic, i.e., mean, standard deviation, mode, etc. However, for some MtL problems it might be useful to not aggregate the information computed with the metric component. This is particularly frequent in MtL applications such as time series or data streams [20] were the data has the same morphology. For example, instead of computing the mean of the correlation between pairs of numerical attributes, one could use the correlation between all pairs of numerical attributes.

4 Decomposing Metafeatures

We used the framework to decompose metafeatures proposed in several applications to assess its applicability and consistence. We show examples from the three types of state-of-the-art metafeatures: simple, statistical and information-theoretic; model-based and landmarkers.

Figure 3 illustrates the decomposition of six simple, statistical and information-theoretic metafeatures. The first three (*number of examples*, *class entropy* and *absolute mean correlation between numeric attributes*) are common metafeatures used in several published papers [3][6][22]. The framework allows to detail the computation of the metafeature. Furthermore, it allows to compare two or more metafeatures. For example, the *absolute mean correlation between numeric attributes* is very similar to *correlation between numeric attributes* (used in data streams applications [20]) except for the aggregation function. In this case, the application domain makes it feasible and potentially more informative to not aggregate the correlation values.

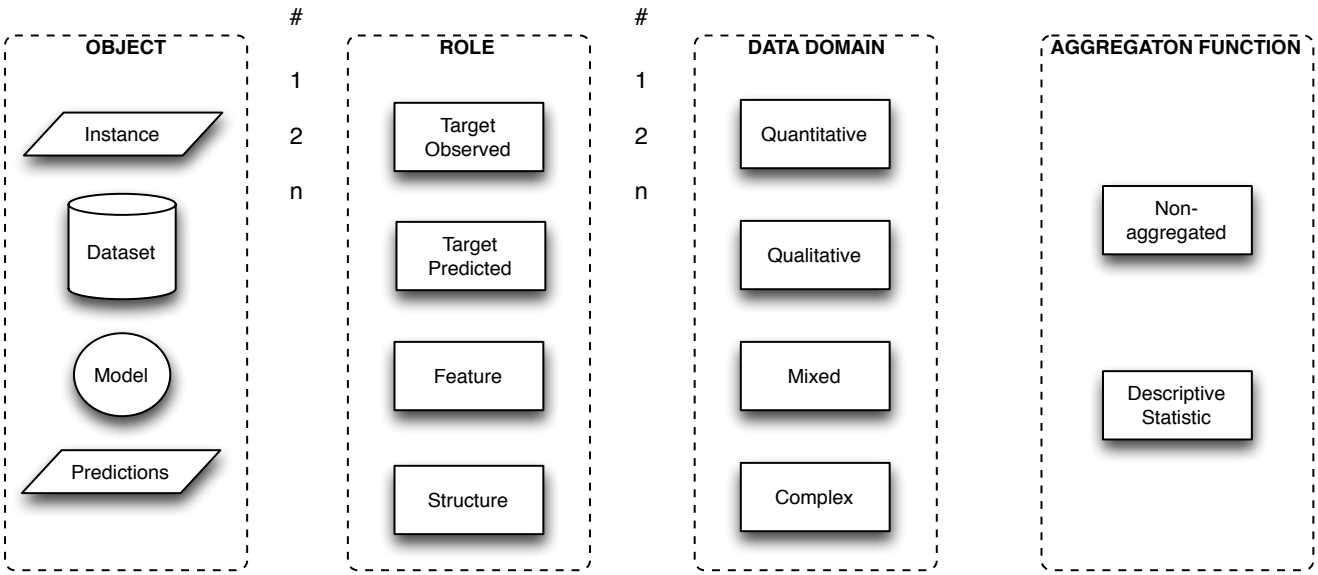

Figure 2. Metafeatures Development Framework.

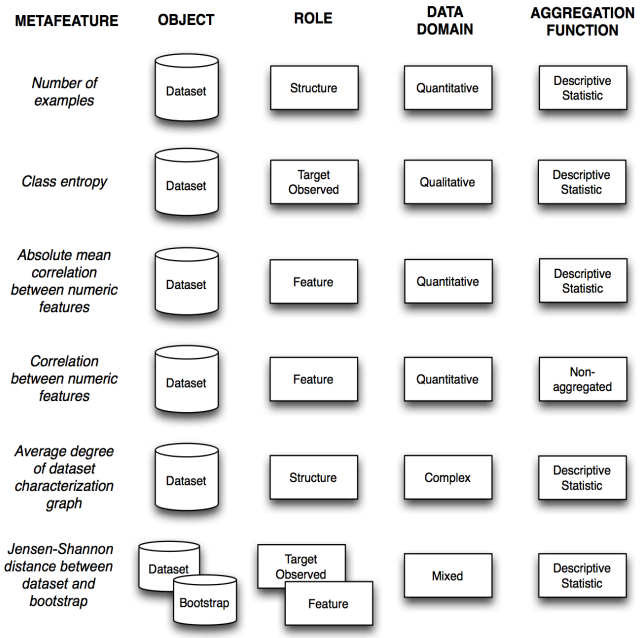

Figure 3. Simple, statistical and information-theoretic metafeatures decomposed using our framework.

Still regarding Figure 3, the decomposition of the two last metafeatures shows that is possible to use the framework for more complex data characteristics. Morais and Prati [12] published a paper in which they use measures from complex network theory to characterize a dataset. Their approach consists in transforming the dataset into a graph by means of similarity between instances. Then, they compute typical measures such as *number of nodes* or *average degree*. Another example would be the Jensen-Shannon distance between dataset and bootstrap [15]. In this example, the authors used the Jensen-Shannon distance to measure the differences caused by the bootstrapping process in the distribution of the variables (features and target).

In Figure 4, we show an example of a model-based metafeature decomposed using our framework. For computing the *number of nodes of a decision tree*, the object component is the model, with particular focus on its structure (as role component). Peng et al.[13] published a paper in which several model-based metafeatures are proposed (for decision trees models).

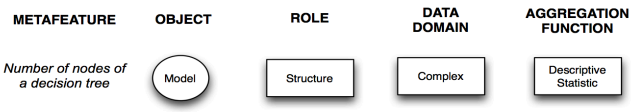

Figure 4. Model-based metafeatures decomposed using our framework.

Finally, in Figure 5, we show the framework applied to landmarkers. The first example, the *decision stump landmarker* [4], uses as object a set of predictions, both the predicted and the observed. Assuming a 0-1 loss function for classification problems, the data domain of a *decision stump landmarker* is always quantitative. Last but not least, the aggregation function in this case is a descriptive statistic, usually a mean. The second example concerns the metafeatures used in the meta decision trees proposed by Todorovski and Džeroski [24]. The authors used the class probabilities of the base-level classifiers as metafeature, particularly, the *highest class probability* of a classifier for a single instance.

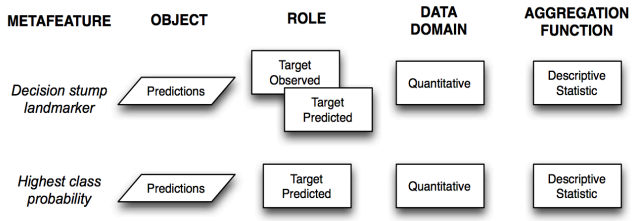

Figure 5. Landmarkers metafeatures decomposed using our framework.

5 Developing Metafeatures

In this Section we present a case study of the proposed framework with a metric widely used in MtL problems [2]: correlation between numeric variables. We show that it is possible to generate new metafeatures by combining elements of different components of the framework. Furthermore, using such framework allows a systematic reasoning in the process of developing metafeatures for a given problem. It becomes easier to detect gaps of non measured information in a set of metafeatures, if it is available a theoretical framework that can guide the user by pointing new research directions.

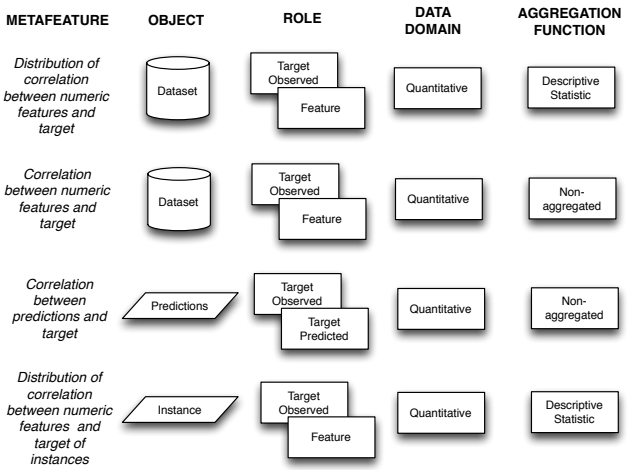

Figure 6. Examples of correlation metafeatures developed using the proposed framework.

As mentioned previously, we use correlation between numeric variables as example in the context of a MtL application for regression algorithm selection [7]. This a problem addressed in a relatively small number of papers in comparison with the classification scenario.

Figure 6 shows an illustration of four metafeatures that use correlation between numeric variables. The first metafeature, *distribution of correlation between numeric features and target*, although present in the literature [2], differs from *absolute mean correlation between numeric features* presented in Figure 3 by adding the element *target* to the role component. This simple change transforms completely the nature of the metafeature in the sense that instead of being a metric of redundancy is a metric of information. The greater the correlation between a numeric feature and target, the more informative that feature can be. Furthermore, it can be more useful to use a specific descriptive statistic (maximum, minimum, etc) instead of the typical mean.

Similarly, the *correlation between numeric features and target* has the same purpose of *distribution of correlation between numeric features and target* but it is indicated for MtL in which the base-level data has the same morphology (as in the data streams scenario [20]). The output of the metafeature is the correlation between the target and each numeric feature.

The two last metafeatures presented in Figure 6, (*correlation between predictions and target* and *absolute mean correlation between numeric features and target of two instances*) were developed using our framework by changing elements of specific components. *Correlation between numeric predictions and target* is another form of landmarker in which instead of using a typical error measure as RMSE, one uses correlation to assess the similarity between the real values and the predicted ones. In terms of the framework decomposition, this metafeature differs from the typical landmarkers in the aggregation function component. Although we did not yet executed experiments on the usefulness of metafeature, it is here proposed to exemplify the applicability of the framework to uncover new metafeatures for a given problem.

Finally, the *distribution of correlation between numeric features and target of instances* can be particularly useful for dynamic selection of algorithms/models in a regression scenario [18][10]. If the MtL problem concerns the selection of an algorithm for each instance of the test set (instead of an algorithm for a dataset) it could be useful to collect information that relates instances. This metafeature would allow to measure the correlation between the numeric variables of the instances. Once again, to the best of our knowledge, there are no reported experiments on the dynamic selection of algorithms using MtL. This metafeature is here proposed as another example of metafeatures that can be developed using correlation as metric.

6 Final Remarks and Future Work

This paper proposes a framework to decompose and develop new metafeatures for MtL problems. We believe that such framework can assist MtL researchers and users by standardizing the concept of metafeature.

We presented the framework and we used it to analyze several metafeatures proposed in the literature for a wide range of MtL scenarios. This process allowed to validate the usefulness of the framework by distinguishing several state-of-the-art metafeatures. We also provide insights on how the framework can be used to develop new metafeatures for a algorithm recommendation in a regression scenario. We use correlation between numeric variables to exemplify the applicability of the framework.

As for future work, we plan to use this framework to generate new metafeatures for algorithm recommendation in a classification scenario and empirically validate the framework. Furthermore, we also plan to use the framework in MtL problems that we have been working on, particularly, MtL for pruning of bagging ensembles and dynamic integration of models.

Acknowledgements

This work is partially funded by FCT/MEC through PIDDAC and ERDF/ON2 within project NORTE-07-0124-FEDER-000059, a

project financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

REFERENCES

[1] Shawkat Ali and Kate A Smith-Miles, 'A meta-learning approach to automatic kernel selection for support vector machines', *Neurocomputing*, **70**(1), 173–186, (2006).

[2] Pavel Brazdil, Christophe Giraud Carrier, Carlos Soares, and Ricardo Vilalta, *Metalearning: applications to data mining*, Springer, 2008.

[3] Pavel B Brazdil, Carlos Soares, and Joaquim Pinto Da Costa, 'Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results', *Machine Learning*, **50**(3), 251–277, (2003).

[4] Johannes Fürnkranz and Johann Petrak, 'An evaluation of landmarking variants', in *Working Notes of the ECML/PKDD 2000 Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, pp. 57–68, (2001).

[5] João Gama and Petr Kosina, 'Recurrent concepts in data streams classification', *Knowledge and Information Systems*, 1–19, (2013).

[6] Alexandros Kalousis, João Gama, and Melanie Hilario, 'On data and algorithms: Understanding inductive performance', *Machine Learning*, **54**(3), 275–312, (2004).

[7] Christian Köpf, Charles Taylor, and Jörg Keller, 'Meta-analysis: from data characterisation for meta-learning to meta-regression', in *Proceedings of the PKDD-00 workshop on data mining, decision support, meta-learning and ILP*. Citeseer, (2000).

[8] Petr Kuba, Pavel Brazdil, Carlos Soares, and Adam Woznica, 'Exploiting sampling and meta-learning for parameter setting for support vector machines', in *Proc. of Workshop Learning and Data Mining associated with Iberamia 2002, VIII Iberoamerican Conference on Artificial Intellignce*, pp. 209–216, Sevilla (Spain), (2002). University of Sevilla.

[9] Christiane Lemke and Bogdan Gabrys, 'Meta-learning for time series forecasting and forecast combination', *Neurocomputing*, **73**(10), 2006–2016, (2010).

[10] João Mendes-Moreira, Alipio Mario Jorge, Carlos Soares, and Jorge Freire de Sousa, 'Ensemble learning: A study on different variants of the dynamic selection approach', in *Machine Learning and Data Mining in Pattern Recognition*, 191–205, Springer, (2009).

[11] João Mendes-Moreira, Carlos Soares, Alípio Mário Jorge, and Jorge Freire De Sousa, 'Ensemble approaches for regression: A survey', *ACM Computing Surveys (CSUR)*, **45**(1), 10, (2012).

[12] Gleison Morais and Ronaldo C Prati, 'Complex network measures for data set characterization', in *Intelligent Systems (BRACIS), 2013 Brazilian Conference on*, pp. 12–18. IEEE, (2013).

[13] Yonghong Peng, Peter A Flach, Carlos Soares, and Pavel Brazdil, 'Improved dataset characterisation for meta-learning', in *Discovery Science*, pp. 141–152. Springer, (2002).

[14] Bernhard Pfahringer, Hilan Bensusan, and Christophe Giraud-Carrier, 'Tell me who can learn you and i can tell you who you are: Landmarking various learning algorithms', in *Proceedings of the 17th international conference on machine learning*, pp. 743–750, (2000).

[15] Fábio Pinto, Carlos Soares, and João Mendes-Moreira, 'An empirical methodology to analyze the behavior of bagging', in *Submitted for publication*, (2014).

[16] Ricardo BC Prudêncio and Teresa B Ludermir, 'Meta-learning approaches to selecting time series models', *Neurocomputing*, **61**, 121–137, (2004).

[17] Matthias Reif, Faisal Shafait, and Andreas Dengel, 'Meta-learning for evolutionary parameter optimization of classifiers', *Machine learning*, **87**(3), 357–380, (2012).

[18] Niall Rooney, David Patterson, Sarab Anand, and Alexey Tsymbal, 'Dynamic integration of regression models', in *Multiple Classifier Systems*, 164–173, Springer, (2004).

[19] André Luis Debiaso Rossi, ACPLF Carvalho, and Carlos Soares, 'Meta-learning for periodic algorithm selection in time-changing data', in *Neural Networks (SBRN), 2012 Brazilian Symposium on*, pp. 7–12. IEEE, (2012).

[20] André Luis Debiaso Rossi, André Carlos Ponce De Leon Ferreira De Carvalho, Carlos Soares, and Bruno Feres De Souza, 'Metastream: A meta-learning based method for periodic algorithm selection in time-changing data', *Neurocomputing*, **127**, 52–64, (2014).

[21] Floarea Serban, Joaquin Vanschoren, Jörg-Uwe Kietz, and Abraham Bernstein, 'A survey of intelligent assistants for data analysis', *ACM Computing Surveys (CSUR)*, **45**(3), 31, (2013).

[22] Carlos Soares, Pavel B Brazdil, and Petr Kuba, 'A meta-learning method to select the kernel width in support vector regression', *Machine Learning*, **54**(3), 195–209, (2004).

[23] Quan Sun and Bernhard Pfahringer, 'Pairwise meta-rules for better meta-learning-based algorithm ranking', *Machine learning*, **93**(1), 141–161, (2013).

[24] Ljupčo Todorovski and Sašo Džeroski, 'Combining classifiers with meta decision trees', *Machine learning*, **50**(3), 223–249, (2003).

[25] Joaquin Vanschoren and Hendrik Blockeel, 'Towards understanding learning behavior', in *Proceedings of the Annual Machine Learning Conference of Belgium and the Netherlands*, pp. 89–96, (2006).

[26] Xiaozhe Wang, Kate Smith-Miles, and Rob Hyndman, 'Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series', *Neurocomputing*, **72**(10), 2581–2594, (2009).