Meta-mining Evaluation Framework: A Large Scale Proof of Concept on Meta-learning

William Raynaut(✉), Chantal Soule-Dupuy, and Nathalie Valles-Parlangeau

IRIT UMR 5505, UT1, UT3, Universite de Toulouse, Toulouse, France
{william.raynaut,chantal.soule-dupuy,nathalie.valles-parlangeau}@irit.fr

Abstract. This paper aims to provide a unified framework for the evaluation and comparison of the many emergent meta-mining techniques. This framework is illustrated on the case study of the meta-learning problem in a large scale experiment. The results of this experiment are then explored through hypothesis testing in order to provide insight regarding the performance of the different meta-learning schemes, advertising the potential of our approach regarding meta-level knowledge discovery.

1 Introduction

Meta-mining designates the very general task of finding an efficient (or most efficient) way to solve a given data mining problem (Fig. 1). As such, it covers a very wide range of tasks, a good many of which have already been extensively studied. For instance, if we consider the very specific problem of Boolean Satisfiability (SAT), we can find different approaches, such as [27], based on the selection of a most efficient algorithm to solve a particular problem instance. Such approaches are designated as *portfolio* for the SAT problem, but have equivalents on many other problems. Their most common denomination would be *algorithm selection* methods, many of which have been studied for machine learning problems, such as classification [11], regression [7], or instance selection [12]. These many different problems have been well studied on their own, but the next step for *meta-mining* research is to start unifying some of them. In particular the problem of *data mining workflow recommendation* has received an increased interest over the last few years [20,22,28]. It consists in the elicitation of workflows (sequences of operators) solving a range of different data mining problems, but remains mostly focused on predictive modelling.

As new approaches emerges, we face a new challenge: *How to evaluate and compare those different meta-mining approaches?* Indeed, the criteria used by authors to evaluate their specific approaches will differ greatly, as they address very dissimilar and sometimes unrelated problems. In order to compare the existing and upcoming approaches able to cover a range of different problems, we will need a *unified* meta-mining evaluation framework. The development of such framework implies a number of new issues, which would be better illustrated on an example.

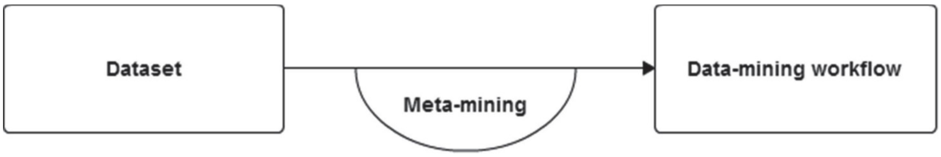

Fig. 1. A general meta-mining experiment

2 Example of Meta-mining Experiment

In this example, we will perform algorithm selection for the well studied problem of classification (Fig. 2). Indeed, as it is one of the most prominent cases of the meta-learning framework, it is notably easier to find well described experiments. The whole set of data presented hereafter is extracted from OpenML [25], an important database of machine learning experiments.

For a given dataset, the objective is then to find a particular classifier maximising a specified criterion. For the sake of the example, let us simply use the traditional (albeit recognized insufficient [10]) criterion of predictive accuracy. To supply our classifier selection, we extract from OpenML two sets of data. The first one should describe the predictive accuracy of different machine learning algorithms over a number of datasets (Table 1), while the second should characterize those datasets according to a number of descriptors (Table 2).

The next step is to decide *how* to solve the meta-mining problem. In this example, we will make the naïve choice of identifying the meta-mining problem with a classification problem over the dataset illustrated in Table 3 (which will be referred as the *meta-dataset*). The *Class* label of a dataset instance of the metadataset identifies which algorithm performed best (i.e. had the highest predictive accuracy) on this dataset, according to the data in Table 1.

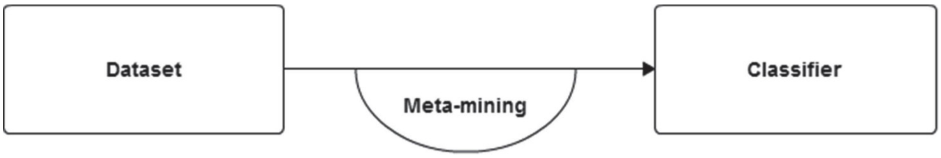

Fig. 2. An algorithm selection experiment for the classification problem

Table 1. Predictive accuracy of a set of classifiers over a range of datasets

	classifier_1	classifier_2	...	classifier_{93}
dataset_1	0.8	0.9	...	...
dataset_2	0.9	0.7	...	...
...	...	...	...	...
dataset_{434}	...	...	...	...

Table 2. Characterization of the datasets

	NumberOfInstances	NumberOfFeatures	...	$\text{MetaAttribute}_{105}$
dataset_1	100	62	...	...
dataset_2	5000	13	...	...
...	...	...	...	...
dataset_{434}	360	20	...	...

Table 3. Meta-dataset for a classification meta-problem

	NumberOfInstances	...	$\text{MetaAttribute}_{105}$	Class
dataset_1	100	...	4	$classifier_{18}$
dataset_2	5000	...	92	$classifier_7$
...	...	...	...	...
dataset_{434}	360	...	13	$classifier_{63}$

Next, we have to solve this classification problem. In this example, we will do so according to the following pseudocode:

```
foreach $dataset_i$ (Dataset instance) do
    Exclude $dataset_i$ from the metadataset
    Apply ReliefF [19] attribute selection algorithm on the metadataset
    Learn a decision tree from the reduced metadataset using a C4.5 tree based
    classifier
    Use this decision tree to predict a class label $classifier_j$ for $dataset_i$
```

For each of the datasets, we then have a predicted class label identifying which algorithm *should* perform best on it, according to a decision tree grown on every other dataset instances. We now want to evaluate the efficiency of this example experiment. For that purpose, we would require a criterion *as independent as possible* from all the particular choices made in the experiment. This can be achieved to some extent by the following:

Definition 1. *Let* ***x*** *be the actual value of the objective criterion (accuracy) achieved on* $\mathbf{dataset}_i$ *by the classifier* $\mathbf{classifier}_j$ *predicted by our experiment. Let* ***best*** *be the best value of the objective criterion achieved on* $\mathbf{dataset}_i$ *among the classifiers* $\mathbf{classifier}_{1..m}$*. Let* ***def*** *be the actual value of the objective criterion achieved on* $\mathbf{dataset}_i$ *by the default classifier (majority class classifier). We define the performance of our example meta-mining experiment on* $\mathbf{dataset}_i$*:*

$$perf(experiment,\ dataset_i) = 1 - \frac{|best - x|}{|best - def|}$$

This performance criterion is maximal at 1 when the predicted classifier achieves the best accuracy among the studied classifiers, and hits 0 when the predicted classifier achieves the same accuracy as the default classifier. Though simple, this criterion allows to compare the performance of meta-mining experiments solving different meta-problems, but needs to be supplied with a *default value* for the considered base criterion.

3 Dimensions of Study

The previous example details one single experiment, giving insight on the performance of one particular method of addressing a restricted area of the meta-mining problem. In order to gain a meaningful insight, one must explore a more significant domain of both problem and solution. This implies iterating the previous experiment over a number of dimensions illustrated in Fig. 3 by their particular values in the example.

Meta-problem. In the example, we chose to identify meta-mining with a classification problem. This was one of the first stances of meta-learning, and leads to a very simple experiment, but much more efficient formulations exist. We could for instance identify meta-mining with a set of regression problems, modelling the performances of the base classifiers, or a set of classification problems, modelling the applicability of the classifiers [1]. Meta-learning studies introduced many different definitions of the problem [4]. The approach followed in [7] consists in learning a model for each pair of base classifiers, predicting if one will significantly outperform the other on a given dataset. This pairwise vision is also adopted in [21], where particular sets of rules are used to compare the performance of the different base learners. [11] introduces active testing, a strategy minimizing the number of tests necessary to select a good classifier. Growing

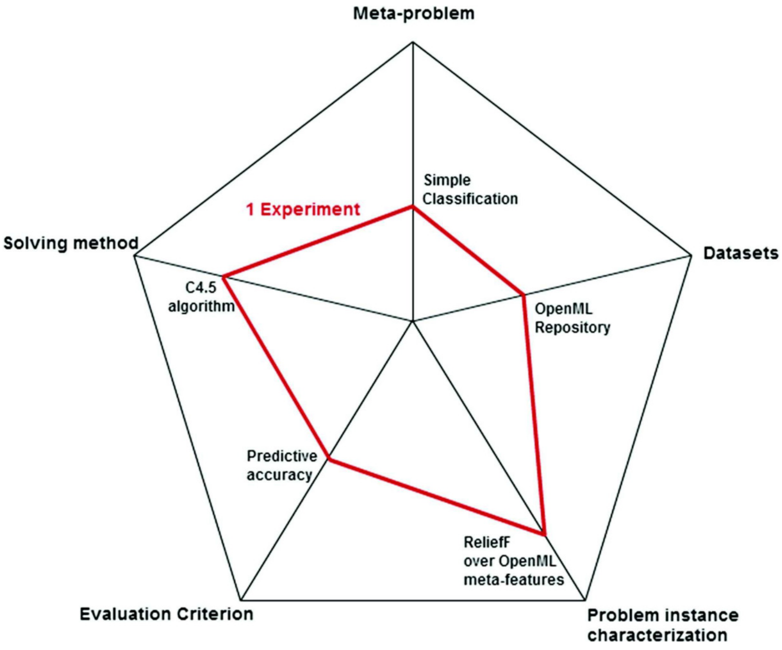

Fig. 3. Dimensions of the example experiment for the classification meta-problem

apart from the meta-learning framework, [14] identifies algorithm selection with a collaborative filtering problem, addressing in particular the problems of stochastic optimisation and boolean satisfiability. [22] tries to address the selection of different data mining operators, considering also the optimisation of their parameters. Finally, [6,15] use the DMOP Ontology to characterize data mining workflows, and learn models exploited in the construction of such workflow for new problem instances.

Repeating the experiment with those more complex definitions of the problem would allow to explore a greater area of the meta-mining problem. Iterating over other dimensions would then provide a sound comparison of those approaches.

Datasets. In order to allow some generality to the results, the datasets used in the experiment should reflect well what *"real world"* datasets are. This is a well know issue in machine learning and meta-learning, where the validation of new techniques requires a *good enough* population of test datasets. Yet, the inherent properties of *real world datasets* remains very unclear. In applications validated over relatively few datasets, it is common to find areas of absolute inefficiency when testing over new datasets. The common assumption is that a *large* sample of datasets provides enough guarantee of generalisability. But once again, *"large"* doesn't mean much, and seems to be often perceived as "larger than last year". In this context the meta-database of OpenML [25] provides a good number of datasets, coming from both the classic literature and from particular applications. To our knowledge, it could be considered one of the most accurate depiction of the set of *real world datasets* available to date, but such matters are difficult to assess and would deserve further studies.

Dataset Characterization. This problem has been addressed along two directions:

- In the first one, the dataset is described through a set of statistical or information theoretic measures. This approach, notably appearing in the STATLOG project [9,13], and in most studies afterwards [7,26], allows the use of many expressive measures. But its performance depends heavily on the adequateness of bias between the meta-level learner and the chosen measures. Experiments have been done with meta-level feature selection [8,23] in order to understand the importance of different measures. But the elicited optimal sets of meta-features to perform algorithm selection over two different pools of algorithms can be very different, revealing no significant tendencies among the measures themselves.
- The second approach to dataset characterization focuses, not on computed properties of the dataset, but on the performance of simple learners over the dataset. It was introduced as landmarking in [17], where the accuracies of a set of very simple learners are used as meta-features to feed a more complex meta-level learner. There again, the performance of the method relies heavily on the adequate choice of both the base and meta-level learner, with no absolute best combination. Further development introduced more complex measures than predictive accuracy over the models generated by the simple learners. For

instance, [16] claims that using as meta-features different structural properties of a decision tree induced over the dataset by simple decision-tree learners can also result in well performing algorithm selection.

OpenML features more than a hundred of such dataset characteristics, getting the most of both approaches. But as the few comparative experiments showed [8,23], the efficiency of a particular characterization mostly depends on its adequation with the problem at hand and the solving method. As some successful approaches focus on developing specific characterization adapted to restricted problems [12], we wish to experiment further by adapting the characterization to the particular set of datasets used in an experiment. In the example, the ReliefF algorithm [19] was used on the metadataset deprived from the i^{th} dataset instance to select best suited dataset characteristics for these particular datasets. But other attribute selection methods exist, that would return potentially different subsets of dataset characteristics. Repeating the experiment with different attribute selection methods or different sets of potential attributes would allow to investigate the relation between the characterization and the other aspects of the problem, while also providing good comparison grounds between the diverse sets of dataset characteristics proposed in the literature.

Solving method. In the example, to predict which method would perform best on each dataset, we used a decision tree built by a particular C4.5 implementation, with a given (default) hyperparameter setting. As a model produced with a different method would possibly be very different, the method employed to solve the chosen meta-problem has to be considered as a dimension of the experiment. For the meta-problem of the example, any method capable of nominal classification could be used. If we also consider different possible hyperparameter settings of those methods, this dimension rapidly grows very large. However, exploring it as well as possible appears critical to the characterization of the different meta-problems. An ideal setup would be to use hyperparameter optimizations techniques on the different solving methods, making hyperparameter optimization a new separate dimension. However, for reasons of dimensional complexity (to be discussed later on), we will for now restrict ourselves to defined hyperparameter settings (hand picked or defaults).

Criterion. The chosen criterion is the measure we wish to enhance through the meta-mining process. In the example, we used predictive accuracy, which is a traditional comparison criterion for classification algorithms. But in another scenario, a different criterion could have made more sense. For instance, in a situation where false negatives are to be avoided in priority (such as in medical diagnosis), a more sensible criterion would have been recall. In practice, one should use a combination of measures to best describe the particular operating conditions of the data-mining experiment to be produced. However, for the sake of generality and simplicity, we will only consider a set of 11 simple measures, such as Cohen's *kappa*, or Kononenko's *Information score* [10].

Since different criteria will likely behave differently, optimal meta-mining processes will likely differ over them. This leads us to iterate experiments over this new dimension of criteria, in order to determine how the previous dimensions of study impact performance for each individual criterion.

Dimensional complexity. In order to get any insights, the space of meta-mining experiments, defined along those dimensions, has to be explored. This means actually realising as many as possible potential experiments we can build along those dimensions. Even with low estimations of the size of the different dimensions, it implies a number of individual experiments in the rough order of magnitude of the billion. As each individual experiment consists in one run of both meta-attributes-selection and a data-mining algorithm, for *each* considered dataset, the exploration of the full dimensional space could span over many years of machine time. However, as each experiment is completely independent from the others, the problem can be addressed through massively parallel computing, which makes exploration possible, even if still time consuming.

4 Experiment Setup

The metadataset is constructed from the OpenML database, which features more than 2700 datasets and 2500 base algorithms. As the construction of the metadataset requires a number of algorithms that were evaluated on the same datasets, we solved a maximal bi-clique problem with an efficient pattern enumerator [24], to find the largest sets of datasets and algorithms such as each element of both sets has been evaluated on every element of the other. This restricted us to 93 algorithms and 434 datasets from the OpenML database. We then extracted the evaluated values of 11 chosen criteria over those 40k runs, and the values of 105 dataset descriptors over each dataset (see *Ressources* section for listings).

In order to run the meta-level experiments, we needed to define a source for the candidate solution algorithms for the different meta-problems, and for the different feature selection algorithm. As it is one of the most widely used and features implementation of many state of the art algorithms, we decided to use the Weka [5] API framework. Spread over 4 classic meta-problems from the literature, we thus evaluate more than 2600 solving methods and 60 feature selection methods built from the Weka API (Fig. 4) (see *Ressources* section for listings).

The individual Weka experiments are then generated by a java program (see *Ressources* section for source code), that delegates their execution to a SLURM job scheduler system managing the *OSIRIM* 640 nodes cluster. The 800k resulting experiments sum up to more than three thousand billion individual executions of machine learning algorithms. Even with good computing power, these experiments are quite costly: 800k experiments of the magnitude order of the minute take almost 100 years of computer time, which reduces to 50 days of parallel execution over the 640 nodes.

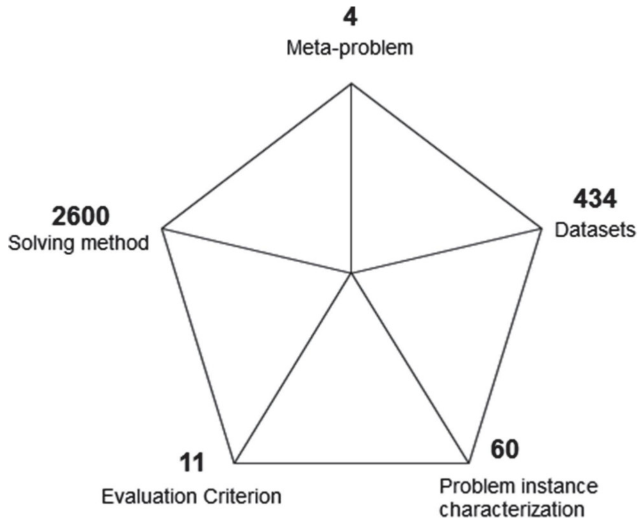

Fig. 4. Rough size of the dimensions

5 Results Interpretation

The setup described in the previous section yields a performance measure for each of those thousands of experiments. In practice, it takes the form of an important database, which can be seen as a population of individual experiment runs, characterized along the earlier described dimensions, and each associated with its performance. In this framework, every question we may aim to answer takes the form of a comparison of sub-populations of those experiment runs. For instance, comparing a new algorithm selection approach to other existing methods can be done by comparing the population of runs featuring the new technique to populations featuring the current state of the art for the problem.

A sheer comparison of mean performance already allows the discovery of tendencies. But to manage the risk of such a tendency not reflecting an actual difference in the sub-populations, one should conduct appropriate hypothesis testing over the results. Hypothesis testing has a reputation to be among the most misused tools in different research areas, and particular care has often been recommended in its application [2]. In this section, we will thus consider the appropriate tests for different situations, and demonstrate them over some cases. Since no assumption can be made regarding the underlying distribution of the performances, we will have to restrict ourselves to non-parametric tests. This implies a lower power than parametric tests could offer, but we will see that the scale of the experiment makes up for this loss.

The first situation we will review is the comparison of two matched sub-population of same size. For instance, let us say we want to compare the performance (in addressing our classifier selection task) of two variants of the Sequential Minimal Optimization algorithm [18] using different kernels. The sub-populations of the runs featuring those two variants have close means (difference

Table 4. Results of some Wilcoxon signed-rank tests

Population 1	Population 2	Difference of means over standard deviation	p-value	Effect size
SMO PolyKernel	SMO Puk	0,02	1,14E-09	0,4
Full set of meta-attributes	50 best from ReliefF	1,2	2,5E-89	0,48
Bagging of RandomForest	RotationForest of RandomForest	1,8	0,91	0,49
RotationForest of RandomForest	RotationForest of NaiveBayesTree	0,04	8,6E-137	0,32

of less than two percent of the population standard deviation), and we can legitimately wonder if there is an actual difference of performance. This situation is ideal for the Wilcoxon signed-rank test (for HO: all performances are identically distributed), which requires independent pairs of values that we can form with the runs of the two variants having all other dimensions equal. Wilcoxon's last assumption of ordinal measure is also met by our numeric performance criterion. The test results in a p-value of 10^{-9} with a 0.4 effect size. This implies that the observed difference between the two variants have a negligible chance not to represent an actual performance difference. Table 4 shows some results that can be obtained with such tests. To interpret these other examples, we could say that the selection of the 50 best meta-attributes with ReliefF will often result in a loss of performance on the whole studied meta-problems, relatively to keeping the full set. The difference of performance between Bagging and RotationForest as enhancer of RandomForest has little chance of betraying a general tendency on the whole studied meta-problems, while the way smaller gap between RandomForest and NaiveBayesTree in a RotationForest has great chances of denoting an actual difference.

Another possibility would be to compare n matched populations of identical size. This is perhaps the most interesting setting, but also the most complex to study, and will feature the use of the Friedman test, with the same assumptions as the Wilcoxon signed-rank test. As an example, let us compare different kNN classifiers with varying k parameter and distance used, (as shown in Table 5), for use in ensembles of nested dichotomies (END) [3] addressing the classifier selection problem. Those approaches have very close mean performance, differing from one another by less than a percent of the global standard deviation. Yet the Friedman test concludes with a p-value of 10^{-90} that some are significantly different from the others.

Finding which one differ for the better will necessitate post-hoc tests, such as the Nemenyi test, in order to control the family-wise error rate (risk of making

Table 5. Mean performance of END built on different kNN classifiers

k	Distance	Mean performance
5	Manhattan	0,8828
5	Euclidean	0,8853
10	Euclidean	0,8899
20	Euclidean	0,8872
20	Manhattan	0,8832

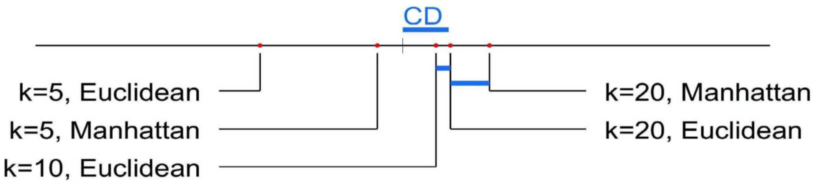

Fig. 5. Results of Nemenyi test, connected groups are not significantly different

at least one incorrect rejection among multiple hypotheses tests). This is often a real problem, as control of the family-wise error rate imposes much more conservative significance levels of the individual tests. But the scale of our experiment allows the extreme values it requires. Results are presented in Fig. 5, where the critical difference is adjusted following Nemenyi procedure to account for the 10 comparison being made (0.05 family-wise error rate). We can see that only the $k = 10$, Euclidian and $k = 20$, Manhattan variants cannot be considered different from the $k = 20$, Euclidian at this significance level, while all others differ by more than the critical difference. In particular, the $k = 10$, Euclidian and $k = 20$, Manhattan variants *are* significantly different from one another.

Running the Friedman test on a much larger number of populations also allows to draw interesting results. For instance, let us compare the different attribute selection methods used. We extracted from the results the performance of the runs featuring the attribute selection methods in identical setups, for over a million setups. This represents more than 50 million performance values to be ranked by Friedman's procedure. The test returns a p-value of zero (beyond machine precision), ascertaining the existence of differences among the methods performance. A Nemenyi test comparing every one of those attribute selection methods with all the others requires an important number of comparison setups in order to be able to find any significant differences. Figure 6 shows that even on a sample 100k setups, the test allows to build groups of methods of *equivalent* performance.

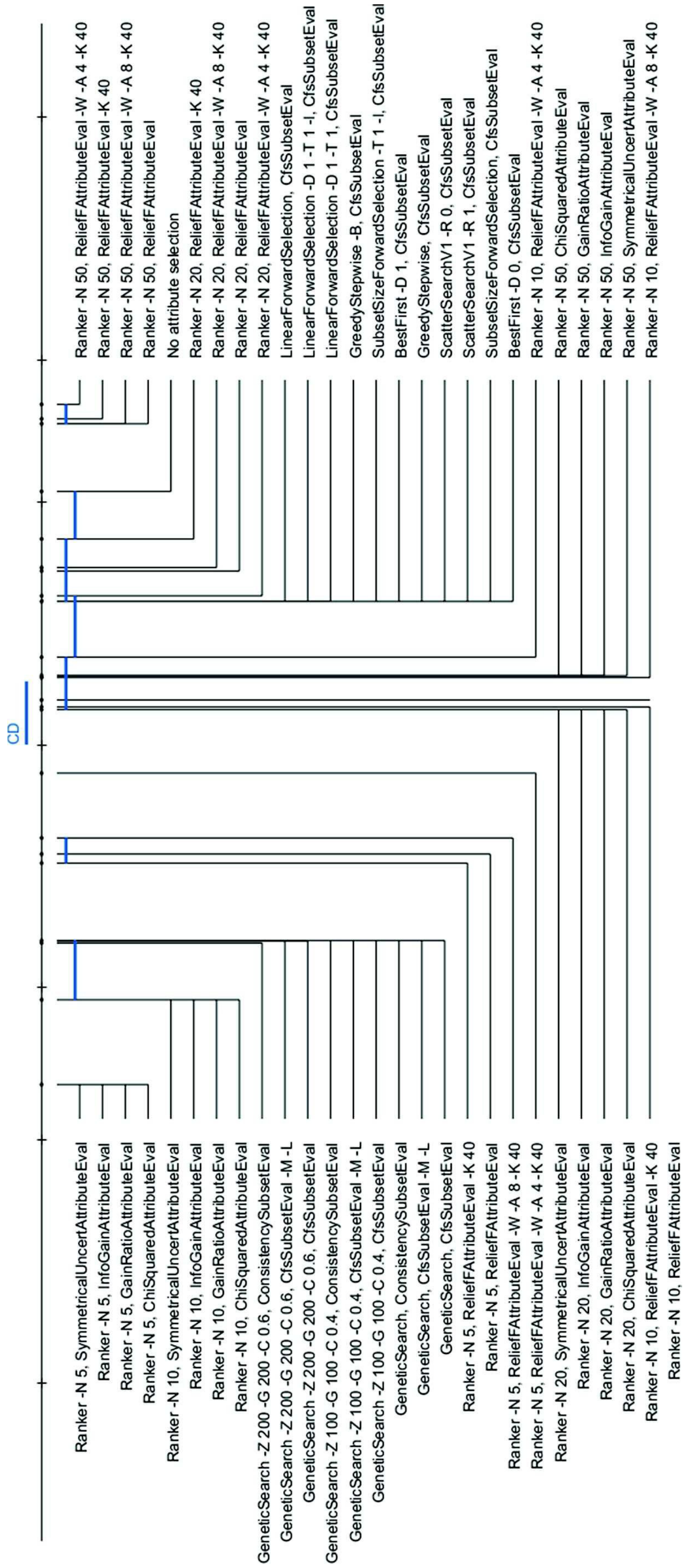

Fig. 6. Overview of a Nemenyi test over all the attribute selection methods

6 Conclusion

In this paper we introduced a meta-mining evaluation framework relying on a unified performance criterion, and demonstrated it on the problem of classifier selection. We characterized the different dimensions of the solutions, instantiated a large number of those classifier selection experiments, and applied statistical hypothesis testing methodologies to the results. These test procedures allowed to draw precise statistical results regarding the comparative performance of different approaches. They are able to produce general insight regarding the optimization potential of particular dimensions. This last result may reveal very interesting, as it can suggest that a second layer of (meta) algorithm selection could maximize the performance of the first. Such result meets the insights of [26] in the suggestion of a recursion of adaptive learners as a possible new paradigm of meta-learning.

Coming back to the evaluation framework, different aspects will require further work, such as the addition of a dimension of hyperparameter optimization, and the use of more dataset meta-attributes from the literature. To our best knowledge, no thorough comparison and review of existing dataset meta-attributes is available, and as they figure among the dimensions our framework allows to study, we intend to apply ourselves to such comparisons. The knowledge gained from the experiments described in this paper will be invaluable to that end, as it will allow to reduce drastically the size of the dimensions, by considering only the elements that were found significantly different. Similar approaches could be considered regarding any or all of the possible dimensions.

Ressources. All materials available at: https://github.com/WilliamR03/Meta-Mining-Evaluation.

References

1. Brazdil, P., Gama, J., Henery, B.: Characterizing the applicability of classification algorithms using meta-level learning. In: Bergadano, F., Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 83–102. Springer, Heidelberg (1994). doi:10.1007/3-540-57868-4_52
2. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
3. Dong, L., Frank, E., Kramer, S.: Ensembles of balanced nested dichotomies for multi-class problems. In: Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 84–95. Springer, Heidelberg (2005). doi:10.1007/11564126_13
4. Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Mach. Learn. **54**(3), 187–193 (2004)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD Explor. Newsl. **11**(1), 10–18 (2009)

6. Hilario, M., Nguyen, P., Do, H., Woznica, A., Kalousis, A.: Ontology-based meta-mining of knowledge discovery workflows. In: Jankowski, N., Duch, W., Grąbczewski, K. (eds.) Meta-Learning in Computational Intelligence. Studies in Computational Intelligence, vol. 358, pp. 273–315. Springer, Heidelberg (2011)
7. Kalousis, A., Hilario, M.: Model selection via meta-learning: a comparative study. Int. J. Artif. Intell. Tools **10**(04), 525–554 (2001)
8. Kalousis, A., Hilario, M.: Feature selection for meta-learning. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 222–233. Springer, Heidelberg (2001). doi:10.1007/3-540-45357-1_26
9. King, R.D., Feng, C., Sutherland, A.: StatLog: comparison of classification algorithms on large real-world problems. Int. J. Appl. Artif. Intell. **9**(3), 289–333 (1995)
10. Kononenko, I., Bratko, I.: Information-based evaluation criterion for classifier's performance. Mach. Learn. **6**(1), 67–80 (1991)
11. Leite, R., Brazdil, P., Vanschoren, J.: Selecting classification algorithms with active testing. In: Perner, P. (ed.) MLDM 2012. LNCS (LNAI), vol. 7376, pp. 117–131. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31537-4_10
12. Leyva, E., Gonzalez, A., Perez, R.: A set of complexity measures designed for applying meta-learning to instance selection. IEEE Trans. Knowl. Data Eng. **27**(2), 354–367 (2015)
13. Michie, D., Spiegelhalter, D.J., Taylor, C.C.: Machine Learning, Neural and Statistical Classification. Ellis Horwood, Upper Saddle River (1994)
14. Misir, M., Sebag, M.: Algorithm selection as a collaborative filtering problem, p. 43 (2013). hal-00922840
15. Nguyen, P., Hilario, M., Kalousis, A.: Using meta-mining to support data mining workflow planning and optimization. J. Artif. Intell. Res. **51**, 605–644 (2014)
16. Peng, Y., Flach, P.A., Brazdil, P., Soares, C.: Decision tree-based data characterization for meta-learning. In: IDDM-2002 p. 111 (2002)
17. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: 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)
18. Platt, J., et al.: Sequential minimal optimization: A fast algorithm for training support vector machines (1998)
19. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Mach. Learn. **53**(1–2), 23–69 (2003)
20. Serban, F., Vanschoren, J., Kietz, J.U., Bernstein, A.: A survey of intelligent assistants for data analysis. ACM Comput. Surv. (CSUR) **45**(3), 31 (2013)
21. Sun, Q., Pfahringer, B.: Pairwise meta-rules for better meta-learning-based algorithm ranking. Mach. Learn. **93**(1), 141–161 (2013)
22. Sun, Q., Pfahringer, B., Mayo, M.: Full model selection in the space of data mining operators. In: Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, pp. 1503–1504. ACM (2012)
23. Todorovski, L., Brazdil, P., Soares, C.: Report on the experiments with feature selection in meta-level learning. In: Proceedings of the PKDD 2000 Workshop on Data Mining, Decision Support, Meta-learning and ILP: Forum For Practical Problem Presentation and Prospective Solutions, pp. 27–39. Citeseer (2000)
24. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Suzuki, E., Arikawa, S. (eds.) DS 2004. LNCS (LNAI), vol. 3245, pp. 16–31. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30214-8_2

25. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. SIGKDD Explor. **15**(2), 49–60 (2013). http://doi.acm.org/10.1145/2641190.2641198
26. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artif. Intell. Rev. **18**(2), 77–95 (2002). http://dx.doi.org/10.1023/A:1019956318069
27. Xu, L., Hutter, F., Shen, J., Hoos, H.H., Leyton-Brown, K.: SATzilla2012: improved algorithm selection based on cost-sensitive classification models, pp. 57–58 (2012)
28. Zakova, M., Kremen, P., Zelezny, F., Lavrac, N.: Automating knowledge discovery workflow composition through ontology-based planning. IEEE Trans. Autom. Sci. Eng. **8**(2), 253–264 (2011)