On the Use of Fast Subsampling Estimates for Algorithm Recommendation

Johannes Fürnkranz[2], Johann Petrak[2], Pavel Brazdil[1], and Carlos Soares[1]

[1] LIACC/FEP, University of Porto, R. Campo Alegre 823, 4150-180 Porto, Portugal
[csoares,pbrazdil]@liacc.up.pt
[2] Austrian Research Institute for Artificial Intelligence, Schottengasse 3, A-1010 Wien, Austria
[johann,juffi]@oefai.at

Abstract. The use of subsampling for scaling up the performance of learning algorithms has become fairly popular in the recent literature. In this paper, we investigate the use of performance estimates obtained on a subsample of the data for the task of recommending the best learning algorithm(s) for the problem. In particular, we examine the use of subsampling estimates as features for meta-learning, thereby generalizing previous work on landmarking and on direct algorithm recommendation via subsampling. The main goal of the paper is to investigate the influence of various parameter choices on the meta-learning performance, in particular the size of training and test sets and the number of subsamples.

1 Introduction

With the availability of a wide range of different classification algorithms, strategies for selecting the most adequate algorithm for a particular data mining problem become more important. Many characteristics of both the learning algorithm and the model class may influence the decision of which algorithms to select or which algorithms to try first. Most important are the accuracy and the understandability of the model and the amount of computing resources (e.g., run-time) required to find it.

The predominant approach for algorithm recommendation is to estimate the accuracy of the candidate algorithms on the problem and select the one that appears to be most accurate [23]. However, this approach is not always feasible. On the one hand, the number of available algorithms may be too large to try every one of them, while on the other hand the size of the database may prevent a full evaluation of an algorithm on the entire dataset.

The first problem—too many algorithms to try each one of them—may be tackled via meta-learning: the candidate algorithms are evaluated once on a fixed set of benchmark datasets (say, e.g., datasets within the UCI repository), each of which is characterized by a fixed number of *data characterization* attributes. This information is then used to learn a model that relates the measured characteristics of each dataset to the performance of the algorithms on the dataset. This model can then be used to predict the performance of the algorithms on new, unseen databases, without having to try any of the algorithms on the new dataset.

The second problem—the database is too large to evaluate all (or some) algorithms—is usually addressed via subsampling, i.e., the idea of using only part of the available data for training. However, while it has been frequently applied to scaling up data mining algorithms [21], its suitabilility for algorithm recommendation has not found much attention in the literature [19].

In this paper, we analyze the combination of subsampling and landmarking in a meta-learning scenario. In particular, we will generalize the idea of landmarking [2, 20] to the use of relative landmarks and landmarks based on subsamples [27, 11]. We will empirically evaluate this idea of *subsample-based relative landmarks* on three different meta-learning tasks (single algorithm recommendation, subset selection and ranking) on a meta-database derived from evaluating 10 machine learning algorithms on 45 datasets from the UCI repository. We will also look at the the effect of varying the three main parameters of subsample-based relative landmarks, namely the size of the training set, the size of the test set, and the number of sampling repetitions.

2 Meta-Learning and Algorithm Recommendation

Meta-learning has recently gained considerable popularility in the literature as a technique for learning how to efficiently use multiple algorithms for improving the learning performance on a problem. Two lines

of research can be distiguished: *meta-classification* and *algorithm recommendation*. Meta-classification schemes are algorithms that use the predictions of one or more base level algorithms as the input of additional layers of learning that aim at improving the predictions of the base classifiers. Members of this family are stacking [30], cascading [12], arbiters and combiners [7], and grading [25]. Such techniques will not be addressed in this paper.

Algorithm recommendation is the problem of learning to recommend an appropriate set of learning algorithms that should be tried on one's prediction problem. Typically, this type of meta-learning involves the steps of *data characterization*, in which datasets are characterized by measures that aim at capturing important properties of a dataset (cf. section 4), and the *meta-learning* phase proper, in which a model is built that relates these characteristics to the performance of learning algorithms on these tasks. One of the earliest applications of this line of research was in the STATLOG project [5, 17]. Currently, at least two European projects are concerned with this type of meta-learning: the MININGMART project investigates the idea of storing generalized data mining (in particular pre-processing) episodes in a case base for later re-use on new problems, and the METAL project investigates the possibility of meta-learning for ranking the available algorithms in a data mining tool according to their expected utility for a given task.

Depending on whether the user is willing to experiment with different options, we can discriminate the following algorithm recommendation scenarios:

- **Selection:** select a single algorithm to be tried on the problem
- **Subset Selection:** select a set of algorithms that are expected to perform well on the problem
- **Ranking:** determine the order in which the candidate algorithms should be tried (most promising first)

Obviously, algorithm selection is a special case of ranking, namely the case where we are only interested in the top-ranked algorithm(s). However, it is useful to consider this as a special case as it might be solved better with more specific techniques.

In both cases, we need to evaluate different algorithms according to some quality criterion. In this paper, we will be simply concerned with accuracy, i.e., we estimate the predictive accuracy of each candidate via 10-fold cross-validation. In other works, we have also considered to trade off accuracy and time into a single measure [26].

3 Subsampling

Subsampling—the idea of training the algorithm on only part of the sample space—is commonly used if the size of the database prevents the application of the algorithm to the entire sample. Different subsampling strategies have been described in the literature to achieve good performance by using only a small amount of data. The crucial parameter is the size of the subsample that has to be drawn in order to guarantee a satisfactory performance at a reasonable time. Approaches to tackle this problem range from statistical estmiates for appropriate subsample sizes [15, 29, 24, 9], over attempts to model the shape of the learning curve [13, 22], to active learning and windowing techniques that give the learning algorithm itself control over the subsampling process [8, 10].

However, the main focus of previous works on this topic has been on evaluating the performance of single algorithms on a subsample. The use of subsampling for algorithm recommendation has only been addressed recently. Petrak [19] performed a systematic study of this approach in the supervised classification setting. Even with simple subsampling strategies, positive results were obtained on the task of single algorithm selection. The learning curves presented in that work indicate that although large samples are often required for reasonable performance estimates, one can sometimes obtain a clear picture of the relative performance of the algorithms on a much smaller sample size. This is illustrated in Figure 1, taken from [19], which shows how the absolute values of the error estimates for the different learning algorithms change considerably with training set size, while the order or the algorithms remains fairly constant.

In the following, we will generalize this approach by combining the use of subsampling with landmarking, an idea from meta-learning, which is described in more detail in the following section.

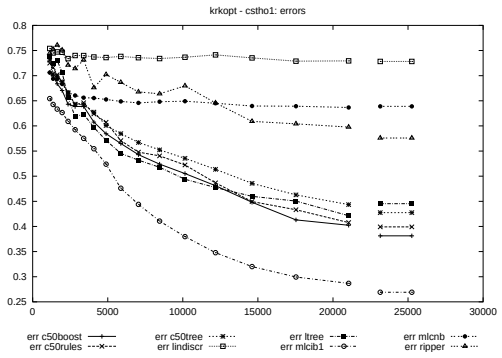

Fig. 1. Sample learning curves showing holdout error estimates for 10 different algorithms for increasing training set size and a constant 10% test set on the UCI dataset krkopt. The separated pieces of the curve give the estimate obtained by 10-times crossvalidation on the full dataset.

4 Data Characterization and Landmarking

As meta-learning is concerned with the learning of mappings from machine learning tasks to machine learning tools, the problem is often formulated as a learning task where the objects are datasets and the class values are candidate algorithms to be used on these datasets. The first problem that has to be solved in such a framework is the definition of a set of descriptors that can be used to describe a dataset in a way that can be used by the meta-learner. This step is called *data characterization*.

Typically, datasets are characterized by measures that can be computed directly from a dataset, ranging from simple things such as the number of (symbolic and numeric) attributes, classes, and instances, to statistical correlation measures and information-theoretic entropy measures. The current version of the METAL Data Characterization Tool DCT [16] computes about 50 base measurements for each datasets, and a variable number of additional measurements for each attribute or each pair of attributes. While the former are quite straight-forward to use, the variable number of the per-attribute measures forces one to use summary measures for the latter. Alternative approaches try to capture the distribution of these values by computing histograms [14] or the use of more expressive learning algorithms [28].

Recently, it was proposed that the use of learning algorithms may provide valuable additional information for this part of the meta-learning problem. *Landmarking* is one such proposal [2, 20], which tries to characterize a dataset by the performance measures of simple learning algorithms. Other proposals focus on using properties of the concepts that are learned with certain algorithms [1] or even the concepts themselves [3].

Landmarking tries to model the practitioner who familiarizes herself with a new problem by first trying a few fast and familiar learning algorithms. Consequently, the basic idea of landmarking is to model the performance of complex and computationally expensive algorithms using performance measures from simple and computationally fast algorithms, the *landmarkers*. Such an approach is based on the assumption that different learning algorithms have different *regions of expertise* in the data space, and that the regions of expertise for complex algorithms can be described by identifying regions of expertise of simpler algorithms. For example, an intuitive rule, such as *"If a decision stump performs well on your data, try growing a decision tree"*, could be learned from a meta-dataset in which several datasets have been annotated with landmarkers, one of them being a decision stump algorithm.

In [20], two important criteria for choosing appropriate landmarkers were proposed:

Efficiency: A good landmarker should be fairly cheap to compute. If expensive computations are required to obtain the landmark, these might be better invested for directly testing the candidate algorithms on the dataset.[3] It makes sense to use one-level decision stumps as landmarkers for predicting whether one should use C50boost, but not the other way around.

[3] This, of course, holds for all data characterisation techniques. For example, one statistical test originally proposed for DCT had a complexity of $O(n^3)$. This turned out to be too expensive and was removed from consideration. [20] suggest to limit the complexity of landmarkers to $O(n \log(n))$.

Bias Diversity: A good set of landmarkers should consist of landmarkers with diverse biases. If two landmarkers have very similar performance measures on all data sets, it would probably suffice to include only one of them.

Based on these criteria, typical choices of landmarkers were simple, computationally efficient algorithms with a high bias. For example, [2] chose one-level decision stumps (with the best, the worst, and a random node), the Naïve Bayes classifier, two versions of nearest neighbour algorithms (one of them operating only on a subset of the available features), a linear discriminant classifier, and the default predictor.

5 Subsample-based Relative Landmarks

Note that the basic requirements for landmarkers, namely to be efficient and diverse, can also be met in a different way, namely with *subsampling landmarkers*: instead of resorting to computationally simple algorithms, one could also select computationally complex algorithms, but evaluate their performance only on a subset of the available data. In fact, we could include evaluations of the same algorithms whose performance we want to predict, on smaller subsets of the data. Clearly, evaluating the algorithms on subsets of the available data is cheaper than a full evaluation of the algorithm on the entire dataset, so that the meta-learning approach may still save costs compared to the latter approach. In addition to the above criteria for landmarkers, we believe that this approach has another interesting property:

Similarity of Regions of Expertise: It is not unreasonable to expect that the region of expertise of an algorithm on the collection of "full" data sets corresponds fairly well to the region of expertise on the collection of subsamples of these databases.

This expectation is based on the results in [19], where the extent to which the order of fast subsampling estimates of classifier performance correspond to the true order of their performance was investigated.

Although we will only focus on accuracy-based landmarkers in this paper, the use of sample-based landmarks is not restricted to this task. For instance, the time to obtain the landmark can also be used to predict the time to run the algorithm on the full set of data and the same applies to the complexity of the model.

Note, however, that conventional landmarkers only capture whether the performance of the landmarker is relatively high or relatively low (compared to the performance of the same landmarker on the other data sets), but not whether the algorithm performs well in comparison to the other algorithms or in comparison to the same algorithm on a different training set size (such a comparison could, e.g., be useful in getting a local estimate for the steepness of the learning curve).

Hence, we propose the use of *relative landmarking*, which provides the meta-learner not with the absolute performance measures of the landmarkers, but with relations that capture the landmarkers' performance relative to each other. In the simplest case, such relations could be inequalities between each pair of landmarkers, but more complicated relations, like inequalities involving significant tests or rankings of the landmarkers are possible (cf. Section 6.1).

6 Experimental Results

There is a variety of options for characterizing the data (the input variables), for determining the optimal algorithm (the class variable), for choosing the application scenario (algorithm recommendation vs. ranking) and for choosing the meta-learning algorithm. In this work, our main focus is on the additional degrees of freedom that we have by being able to choose the training and test set sizes and the number of subsampling repetitions, which form the basis of our sample-based relative landmarking technique.

6.1 Experimental Setup

First, we had to pick our pool of base learners and benchmark problems. Within the METAL-project, we collected detailed results of 10 classification algorithms on a large set of datasets, most of them from the UCI repository [4]. The learning algorithms are C50 (trees, rules and boosted), Ltree, Ripper, a linear

discriminant classifier, the instance-based learner and the Naive Bayes learner from MLC++, and the multi-layer perceptron and the radial basis function from Clementine. For this study, we selected 45 classification problems with 1000 or more examples.

For simplicity, we decided to use predictive accuracy as the quality criterion, i.e., we consider the most accurate algorithm as the best. We evaluated single algorithm recommendation (predict the best algorithm), subset recommendation (predict a suitable subset), and ranking (sort the algorithms according to their suitability). Single algorithm recommendation was implemented and evaluated as a conventional 10-class classification problem. In case of multiple best algorithms, we broke ties in favor of the fastest algorithm. Subset recommendation was implemented by learning a separate model for each of the algorithms that predicts whether the algorithm is among the best algorithms or not. Here we defined "best" as no different than the most accurate algorithm at a 5% significance level (McNemar test). This scenario was evaluated by reporting the average accuracy on the 10 learning problems.[4]

In both recommendation scenarios we tried several classification algorithms at the meta-level, but will only report on the result of C50 because it is well-known and in general performs well. Finally, algorithm ranking was addressed with an instance-based ranking technique [26] combined with the average ranks ranking method [6]. For a new example, this algorithm computes a neighborhood of the k most similar problems, determines the rank of each candidate algorithm on each dataset in this neighborhood, and computes a final ranking of the algorithms based on the average rank in the neighborhood around the example. This was evaluated using Spearman's rank correlation of the predicted ranking with the true ranking, yielding 1 if the rankings perfectly match, -1 if one is the inverse of the other, and 0 if they are unrelated. For details of the evaluation methodology we refer to [26]. Again, we have results for several settings of the size of the neighborhood, but will focus on size $k = 5$ here. This value represents approximately 10% of the total number of datasets, which obtained good results in earlier work [26].

The main goal for the present study, however, was to look at various parameter settings for obtaining subsample estimates. Following [19], we use subsampling strategies that randomly select a training sample of a fixed size (100, 200, or 500 examples), and test the learned theories on a different subset of the available data (100 examples, 5%, or 10% of the data). The motivation for these choices is that the (usually critical) training effort is reduced to constant time complexity, while testing time is reduced by a constant factor (sampling itself is comparably fast with linear time complexity for sequential access files). In addition, we also varied the number of samples used for the estimation (one, five, ten iterations).

Based on the obtained accuracy estimates for subsamples of various sizes, we used the following five derived landmarkers:

Absolute (LM): The 10 original accuracy estimates.

Ranks (RK): Like with conventional landmarking, there is one attribute corresponding to each landmarker. However, this attribute does not encode the accuracy estimate obtained for the landmarker, but its performance rank among its competitors (i.e., a number from ≥ 1, where 1 means it was best, 2 means it was 2nd-best, etc.). In the case of ties, all tied values receive the same value (the best rank, i.e., the lowest number). The ranks are used as continuous features and not as symbolic values.

Simple Pairwise (SP): This strategy enables the learner at the meta-level to make use of pairwise comparisons between the accuracies of the landmarkers. For each pairs of landmarkers, these relations returned $+1$ if the first value was larger, -1 if the second value was larger, and 0 for equal values.

Confidence Interval-based Pairwise (CP): This expands the pairwise comparisons by taking the 95% confidence intervals of the error estimates into account [18].

Ratios (RT): Here, we encode the pairwise accuracies ratios for all pairs of landmarkers. This representation may be seen as a generalization of the previous one, where the symbolic, binary attribute with the values $-1, 0, +1$ is represented as a continuous range. We use ratios instead of differences because we wanted to have differences relative to the order of magnitude of the error, instead of absolute values.[5]

[4] Note that in this setting it may happen that the empty subset is predicted, if all of the 10 classifiers predict that their algorithm is not applicable. We also tried a different setting where we predict a fixed default algorithm, C50boost(the most accurate algorithm on average) in such cases, which is probably more realistic for practical applications, but the results were qualitatively the same.

[5] The fact that the resulting value range is $(0, \infty)$ is assymetric around 1, the value of equal performance, should not make a difference to algorithms like C4.5 or Ripper, which treat continuous attributes by dynamically discretizing

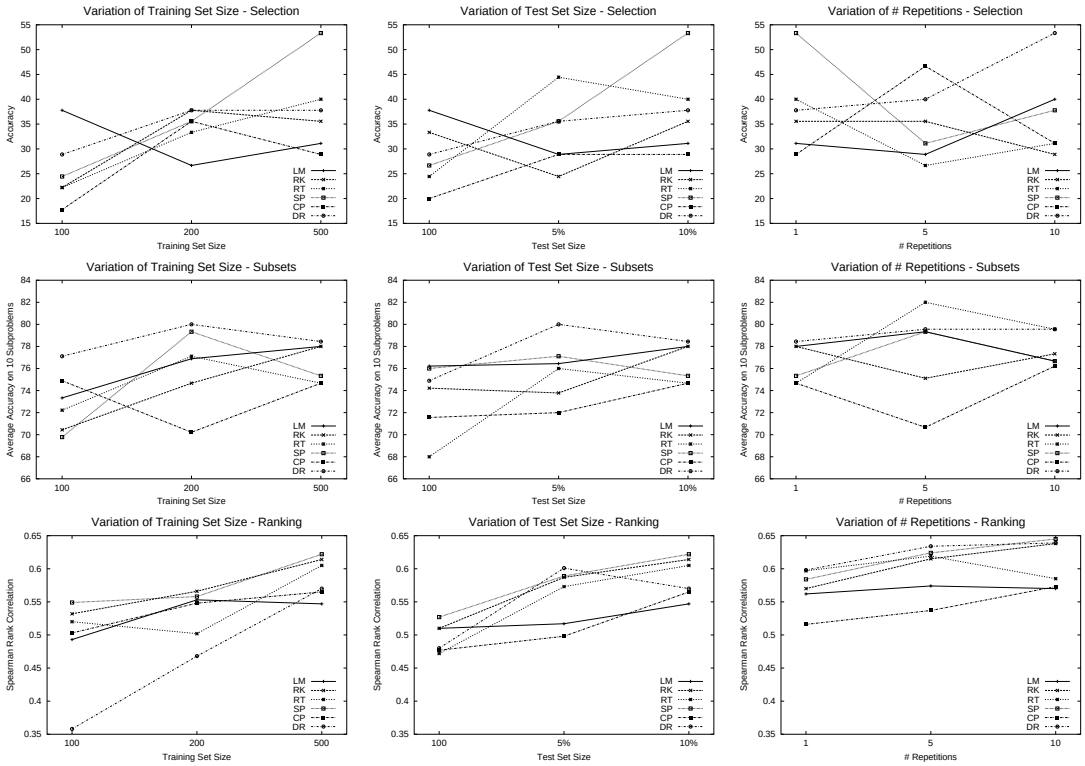

Fig. 2. Varying sampling parameters. The left column show the results for varying the training set size (100, 200, and 500 examples), the middle column the results of varying the test set size (100, 5%, 10%), and the right column the results of varying the number of subsampling repetitions (1, 5, and 10). The three different rows show the different algorithm recommendation settings: recommending a single algorithm, recommending a subset, and ranking the algorithms.

For comparison, we also used a set of 25 **General, Statistical and Information-theoretic (GSI)** measures for data characterization. We chose the same subset of features used in [26]. We also compare the results to the **direct recommendation (DR)** strategy, which simply predicts that the algorithm which performs best on the subsample will also perform best on the entire data set. Note that relative sample-based landmarks allow to encode this classifier in the form of a simple rule with one condition (namely the pairwise relation of the subsample landmarks). This means that a meta learner that uses a relative encoding of subsample landmarks should be able to fall back on this default classfier, and, ideally, be able to improve upon it by learning a more complex theory.

6.2 Effects of Sampling Parameters

We first look at the effects that different sampling parameters have on the metalearning task. Starting from a single holdout estimate based on a training set with 500 cases and a test set of 10% of the data, we individually varied the training set size (100, 200, and 500 cases), the test set size (100 cases, 5% and 10% of the data), and the number of holdout repetitions (1, 5, and 10). The results are shown in Figure 2.

The results show that overall, metalearning accuracy tends to increase with the size of the training set and test set used for the landmarker. Larger training sets for the subsample learning task give better recommendation results because they presumably give theories that are closer to the final theory. Similarly, larger test sets also give better recommendation results, presumably because they estimate the quality of the theories learned from the subsample more accurately. The effect is less pronounced for the number of

them, and which we will use as meta-algorithms. It might, however, make a difference if numerical algorithms are used (e.g., linear regression).

Table 1. Results of direct recommendation, statistical data characterization, and sample-based relative landmarks for predicting a single algorithm (accuracy), a subset of algorithms (average accuracy on 10 problems), and for ranking (Spearman's rank correlation), for training set sizes 500 and test set sizes 10%.

	single	subset	ranking
DR	37.78%	78.44%	0.570
GSI	37.78%	74.22%	0.495
LM	31.11%	78.00%	0.562
RK	35.56%	78.00%	0.571
RT	40.00%	74.67%	0.597
SP	53.55%	75.33%	0.584
CP	28.89%	74.67%	0.516
default	42.22%	77.78%	0.529

repetitions. This again, is not surprising, as an increase in the number of repetitions should only reduce the variance in the landmark values, but not change their expected values. Thus, while the order of the landmark values on the subsample becomes more stable with an increasing number of repetitions (which may make learning more reliable), it does not necessarily contain more information about the target, i.e., the ranking of the algorithms on the full data set.

Figure 2 suggests that ranking is more stable across different landmarkers than the other two approaches. This difference could be explained by the higher bias of the k-NN algorithm used in the former when compared to the decision trees used in the former. Therefore, ranking results are less affected by using the various types of landmarkers, given that they are really different representations of the same basic information. The validity of this hypothesis reamains to be investigated.

6.3 Comparison of Strategies

Here we compare the results of the various strategies for algorithm recommendation. We compare direct recommendation based on subsamples, meta-learning using GSI/DCT measures, and meta-learning using five different kinds of sample-based relative landmarks for subsample size 500 and using 10% of the data as test sets. This setting is the most reliable setting (in the sense that it uses the largest training and test sets and its estimates consequently have the lowest bias and variance) among the subsampling strategies we investigated (for single, not iterated samples).

Table 1 shows the results. We can draw several conclusions from them: For one, sample-based relative landmarks can outperform direct algorithm recommendation, and are almost never worse. The exception is the case where we try to learn whether an algorithm is applicable or not. In spite of the positive results, the gains obtained with meta-learning were smaller than we expected beforehand. Figure 3 provides a simple explanation for this. The direct sampling obtains better results for smaller datasets. These are datasets where the training sample is larger in the sense that the 500 cases used represent a larger proportion of the data. Therefore, the models obtained with the sample tend to be more similar to the true models and, thus, provide good estimates of their accuracy. On the other hand, the meta-learning approach as a clear advantage on the larger datasets.

Nevertheless, it seems to be the case that meta-learning can make use of the additional information that is provided by sample-based landmarks, in particular for the case of direct recommendation. Secondly, sample-based relative landmarks seem to be a viable alternative to statistical and information-theoretic data characterization techniques. In some cases, they produce much better results. Finally, it seems to be the case that different feature sets are good for different meta-learning tasks. While there seems to be some correspondence between the single selection task and the ranking task, the results on the subset selection task are completely reversed. Here, direct selection is the best performer, while ratios (RT) and pairwise comparisons (CP), which produce the best results on the other tasks, are at the bottom end.

7 Conclusions

In this paper we have presented fast subsampling landmarkers in various contexts of metalearning. We have shown that the choice of the training set and test set size of the landmarker influences the quality of

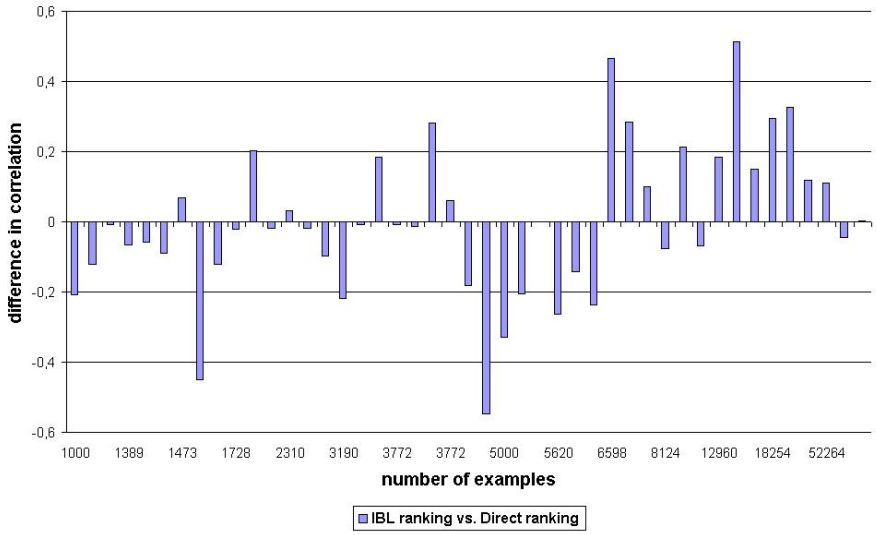

Fig. 3. Effect of the dataset size on the difference between the correlation obtained by the meta-learning and the direct ranking approaches.

the metalearning task. Metalearning using subsampling-based landmarkers are competitive with traditional metalearning aproaches that use statistical and information–theoretic database measurements and can outperform the traditional method of using subsamples for direct selection of the best algorithm. Similarily, subsample-based landmarkers can be used successfully for ranking recommendations, especially for large datasets. These results were obtained with a very simple strategy of picking constant-size subsets of the data for the landmarkers. Future work will investigate how to adapt the size of the training sets used for the landmarkers better to the complexity and size of the database, while still guaranteeing competitive time requirements.

References

[1] Hilan Bensusan. God doesn't always shave with occam's razor - learning when and how to prune. In C. Nédellec and C. Rouveirol, editors, *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, pages 119–124, 1998.

[2] Hilan Bensusan and Christophe Giraud-Carrier. Casa Batló is in Passeig de Gràcia or landmarking the expertise space. In J. Keller and C. Giraud-Carrier, editors, *Proceedings of the ECML-00 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 29–46, Barcelona, Spain, 2000.

[3] Hilan Bensusan, Christophe Giraud-Carrier, and Claire Kennedy. A higher-order approach to meta-learning. In *Proceedings of the ECML'2000 workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 109–117. ECML'2000, June 2000.

[4] C. L. Blake and C. J. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`, 1998. Department of Information and Computer Science, Unifirsty of California at Irvine, Irvine CA.

[5] Pavel B. Brazdil, João Gama, and Bob Henery. Characterizing the applicability of classification algorithms using meta-level learning. In F. Bergadano and L. De Raedt, editors, *Proceedings of the 7th European Conference on Machine Learning (ECML-94)*, number 784 in Lecture Notes in Artificial Intelligence, pages 83–102, Catania, Italy, 1994. Springer-Verlag.

[6] Pavel Brazdil and Carlos Soares. A Comparison of Ranking Methods for Classification Algorithm Selection. *Machine Learning: Proceedings of the 11th European Conference on Machine Learning (ECML2000)*, pages 63–74, 2000. Springer-Verlag.

[7] Philip K. Chan and Salvatore J. Stolfo. A comparative evaluation of voting and meta-learning on partitioned data. In *Proceedings of the 12th International Conference on Machine Learning (ICML-95)*, pages 90–98. Morgan Kaufmann, 1995.

[8] David A. Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[9] Pedro Domingos and Geoff Hulten. A general method for scaling up machine learning algorithms and its application to clustering. In C. Brodley and A. Danyluk, editors, *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, pages 106–113, Williams College, MA, 2001. Morgan Kaufmann.

[10] Johannes Fürnkranz. Integrative windowing. *Journal of Artificial Intelligence Research*, 8:129–164, 1998.

[11] Johannes Fürnkranz and Johann Petrak. An evaluation of landmarking variants. In C. Giraud-Carrier, N. Lavrač, Steve Moyle, and B. Kavšek, editors, *Proceedings of the ECML/PKDD Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2001)*, pages 57–68, Freiburg, Germany, 2001.

[12] Joao Gama and Pavel Brazdil. Cascade generalization. *Machine Learning*, 41(3):315–343, 2000.

[13] George H. John and Pat Langley. Static versus dynamic sampling for data mining. In E. Simoudis and J. Han, editors, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 367–370. AAAI Press, 1996.

[14] Alexandros Kalousis and T. Theoharis. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*, 3(5):319–337, November 1999.

[15] Jyrki Kivinen and Heikki Mannila. Approximate dependency inference from relations. *Theoretical Computer Science*, 149(1):129–149, 1995.

[16] Guido Lindner and Rudi Studer. AST: Support for algorithm selection with a CBR approach. In C. Giraud-Carrier and B. Pfahringer, editors, *Proceedings of the ICML-99 Workshop on Recent Advances in Meta-Learning and Future Work*, Bled, Slovenia, 1999.

[17] Donald Michie, D.J. Spiegelhalter, and Charles C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[18] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[19] Johann Petrak. Fast subsampling performance estimates for classification algorithm selection. In J. Keller and C. Giraud-Carrier, editors, *Proceedings of the ECML-00 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 3–14, Barcelona, Spain, 2000.

[20] Bernhard Pfahringer, Hilan Bensusan, and Christophe Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)*, Stanford, CA, 2000.

[21] Foster Provost and Venkateswarlu Kolluri. A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, 3(2):131–169, 1999.

[22] Foster J. Provost, David Jensen, and Tim Oates. Efficient progressive sampling. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pages 23–32, 1999.

[23] Cullen Schaffer. Selecting a classification method by cross-validation. *Machine Learning*, 13(1):135–143, 1993.

[24] Tobias Scheffer and Stefan Wrobel. Incremental maximization of non-instance-averaging utility functions with applications to knowledge discovery problems. In C. Brodley and A. Danyluk, editors, *Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, Williams College, MA, 2001. Morgan Kaufmann.

[25] Alexander K. Seewald and Johannes Fürnkranz. An evaluation of grading classifiers. In *Advances in Intelligent Data Analysis: Proceedings of the 4th International Symposium (IDA-01)*, pages 115–124, Lisbon, Portugal, 2001. Springer-Verlag.

[26] Carlos Soares and Pavel Brazdil. Zoomed ranking: Selection of classification algorithms based on relevant performance information. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2000)*, pages 126–135, 2000. Springer-Verlag.

[27] Carlos Soares and Johann Petrak and Pavel Brazdil. Sampling-based relative landmarks: Systematically test-driving algorithms before choosing. *Proceedings of the Portoguese AI Conference (EPIA-01)*, 2001. Springer-Verlag.

[28] Ljupčo Todorovski and Sašo Džeroski. Experiments in meta-level learning with ILP. In *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-99)*, pages 98–106. Springer-Verlag, 1999.

[29] Hannu Toivonen. Sampling large databases for association rules. In *Proceedings of the 22nd Conference on Very Large Data Bases (VLDB-96)*, pages 134–145, Mumbai, India, 1996.

[30] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–260, 1992.