



The University of Sydney

A Landmarker Selection Algorithm Based on Correlation and Efficiency Criteria

Technical Report Number 556

September 2004

Daren Ler, Irena Koprinska and Sanjay Chawla

ISBN 1 86487 669 7

**School of Information Technologies
University of Sydney NSW 2006**

A Landmarker Selection Algorithm Based on Correlation and Efficiency Criteria

Daren Ler, Irena Koprinska, and Sanjay Chawla

School of Information Technologies, Madsen Building F09,
University of Sydney NSW 2006, Australia
{ler, irena, chawla}@it.usyd.edu.au

Abstract. Landmarking is a recent and promising meta-learning strategy, which defines meta-features that are themselves efficient learning algorithms. However, the choice of landmarks is often made in an ad hoc manner. In this paper, we propose a new perspective and set of criteria for landmarks. Based on the new criteria, we propose a landmarker generation algorithm, which generates a set of landmarks that are each subsets of the algorithms being landmarked. Our experiments show that the landmarks formed, when used with linear regression are able to estimate the accuracy of a set of candidate algorithms well, while only utilising a small fraction of the computational cost required to evaluate those candidate algorithms via ten-fold cross-validation.

1 Introduction

With the growing plethora of machine learning algorithms, and both theoretical [18] and empirical [12] results indicating that no single algorithm is generically superior, the issue of selecting an appropriate learning algorithm for a given dataset becomes increasingly important. Traditionally, the common practice is to evaluate all applicable (candidate) algorithms based on some form of *hold-out testing* (e.g. *cross-validation* and *bootstrapping*), and thereby determine which to use (e.g. [14]). However, such evaluation is typically computationally unviable due to the volume of available algorithms. To overcome this, various methods utilising past experience, or meta-knowledge [8], have been proposed. Typically referred to as *meta-learning* [8, 15], such solutions utilise experience on previous datasets (i.e. meta knowledge) to learn hypotheses that characterise the *domains of expertise* of the candidate algorithms. Given the set of all possible datasets, these domains of expertise correspond to subsets in which certain algorithms are deemed to be superior to others.

As in standard machine learning, the success of meta-learning is greatly dependent upon the quality of the features chosen. Various strategies for defining these *meta-features* have been proposed [4, 10, 5, 12, 1, 9]. However, to date there is no consensus on how good meta-features should be chosen. *Landmarking* [13, 6, 7] is an alternative and promising approach that characterises datasets by directly measuring the performance of simple and fast learning algorithms, called landmarks. The

main idea is that the performance of a learning algorithm on a dataset uncovers information about the nature of the dataset [2]. However, the selection of landmarkers is typically done in an ad hoc fashion, with the landmarkers generated focused on characterising the domains of expertise of an arbitrary set of algorithms.

In this paper, we reinterpret the role of landmarkers, defining each as a function over a *set* of learning algorithms that characterises the domain of expertise of *one* specific learning algorithm. Essentially, given a set of candidate algorithms, we wish to generate a set of landmarkers (i.e. for each candidate algorithm, we seek to find one corresponding landmarker), such that each landmarker: (1) is more *efficient* than its counterpart candidate algorithm¹, and (2) corresponds to a domain of expertise that is similar or *correlated* to that of its associated candidate algorithm (i.e. the domains of expertise of the landmarker and associated algorithm roughly overlap in the space of all possible datasets – henceforth labelled the expertise space). Via these new landmarker criteria, we propose a new approach for landmarker generation, where the main idea is to use some subset of the candidate algorithms to landmark each of those candidate algorithms.

The paper is organised as follows. In Section 2, we redefine the role of landmarkers and introduce the new criteria for landmarker selection. The subsequent section introduces a new method for landmarker generation based on the new criteria. Section 4 then describes and discusses the experiments, and corresponding results. The last section concludes the paper and suggests some paths for future work.

2 Establishing Good Meta-attributes: Landmarker Criteria

As is the case with any standard machine learning problem, the performance and success of meta-learning is greatly dependent upon the available inputs and the corresponding features used to describe of the problem. Thus, appropriate meta-features, or in our case, appropriate landmarkers, must be found.

2.1 Redefining Landmarkers

In the literature [13, 6, 7], a landmarker is typically associated with a single algorithm with low computational complexity. Landmarkers have thus far been employed much in the same manner as the prototypical meta-features; they simply serve as meta-features whose purpose is to help define an algorithm-generic expertise space, in which the domain of expertise of *any* candidate algorithm may be defined. This former definition of landmarkers is exemplified in Figure 1 (annotated from [2]).

¹ It may seem that by estimating one candidate algorithm via a set of learning algorithms, we are actually increasing computational complexity. However, as we will see from our proposed landmarker generation algorithm in Section 3, we may limit the set of algorithms from which we construct our landmarkers, such that this set has less computational complexity than the set of candidate algorithms.

This figure denotes the space of all datasets S , where each rectangle denoted by a l^{old}_i represents the domain of expertise of a landmarker (under the old definition – i.e. each l^{old}_i corresponds to the domain of expertise of single algorithm), and each dotted ellipse a_i represents the domain of expertise of a candidate algorithm. The general idea is that by noting which landmarker domains of expertise the dataset falls under, it would be possible to induce which candidate algorithm domains of expertise to which it belongs. For example, when a dataset falls within the domains of expertise of l^{old}_3 and l^{old}_4 , we may infer that the dataset in question probably belongs to the domain of expertise of a_2 . And correspondingly, when another dataset falls within the domains of expertise of l^{old}_2 or similarly, if it does not fall within the domains of expertise of the other l^{old}_i , then it is likely to belong to the domain of expertise of a_1 .

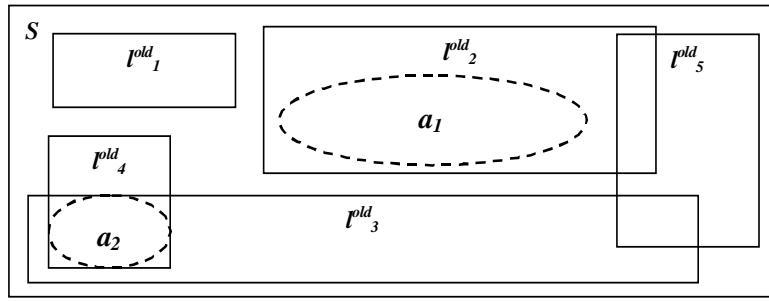


Fig. 1. An example depicting the use of landmarks under the old definition

We introduce a new perception of landmarks that defines each landmarker to be:

1. A function over set of learning algorithms.
2. Specific to one candidate algorithm; i.e. the role of the specified landmarker is to characterise only one single algorithm's domain of expertise.

For example, a landmarker for a boosted C4.5 decision tree algorithm could correspond to the accuracies of several learning algorithms such as a decision stump, a naïve Bayes learner, etc. A counterpart of Figure 1, using our new perception of a landmarker is depicted in Figure 4 of Section 2.2 (after further explanation of the criteria used to generate such landmarks).

This new perception additionally suggests that predicting the domain of expertise of each candidate algorithm should be treated as an isolated learning problem.

The characteristics associated with this new perception of landmarks (i.e. points 1 and 2 above), are important for two reasons. Firstly, recent empirical results have shown that different candidate algorithms require different characteristics to better predict their domains of expertise (i.e. each meta-feature can have varying significance depending on the candidate algorithm involved) [13, 10, 9]. And secondly, recent work has also shown empirically that estimating the individual predictive

accuracy of each candidate algorithm is better than attempting to learn an aggregated concept (e.g. best performing algorithm in the set) over the set of algorithms [11].

Intuitively, while one can consider an overall concept regarding the generic expertise space, this requires that the dataset characteristics be relevant in defining the domain of expertise of *any* algorithm. This meta-feature space would most likely be very complex, and thus difficult to define. Additionally, with this universal meta-feature paradigm, there is also a much higher probability of learning chance concepts.

2.2 New Landmarking Criteria

In previous landmarking work, two landmarker criteria have been defined: efficiency and bias diversity [13]. The rationale behind the efficiency criteria is obvious – we wish to incur less computational cost than directly evaluating the set of candidate algorithms. Conversely, the rationale behind bias diversity is more vague; the general idea behind bias diversity is to ensure that different landmarkers measure different dataset properties, at least implicitly [13]. This may be interpreted as a criterion that requires the domains of expertise of the landmarkers to be non-correlated.

While these criteria aid to restrict the search space of algorithms (i.e. potential landmarkers), their utility in terms of pinpointing or directing the search for viable landmarkers is questionable. Essentially, the efficiency and bias diversity criteria do not emphasise the selection of landmarkers that would map the domains of expertise of a specific set of candidate algorithms; they do not place any requirement on the relationship between the selected landmarkers and the set of candidate algorithms.

Consequently, it could be interpreted that these criteria seek to find a set of landmarkers that is able to characterise the space of all datasets well enough so that the domain of expertise of *any* algorithm (i.e. a generic domain of expertise) may be defined. However, these criteria do not ensure the generation of a set of landmarkers that characterises the right dataset features such that there is sufficient generality required to locate the domains of expertise of the given set of candidate algorithms. For example, although the landmarkers depicted in Figure 2 have uncorrelated domains of expertise, they cannot distinguish between the domains of expertise of the five algorithms. This is because the landmarkers do not relate to the algorithms they are meant to characterise. Thus, what landmarker criteria should we then use?

In meta-learning, we are primarily interested in the performance measurements² of the algorithms available to us. Thus, to meta-learn, we must map the landmarker measurements to the performance measurement of the candidate algorithm whose domain of expertise we are attempting to learn. This implies that the landmarker for a candidate algorithm should output measurements that are indicative of the performance measurements on that candidate algorithm. More specifically, in order to pick a landmarker for some algorithm, we should ensure that the measurements output by the landmarker are associated or *correlated* to the performance the candi-

² Essentially, all the meta-target types are functions of the performance measurements over the candidate algorithms.

date algorithm; this would be one way to ensure that the landmarker is related to the target. This may also be explained in terms of the expertise space.

Two algorithms whose domains of expertise are overlapping will be closer to each other in a space of all domains of expertise. Conceptually, the distance between two algorithms \mathbf{a} and \mathbf{l} can be regarded as $\|\mathbf{a} - \mathbf{l}\|$. Also, we may express $\|\mathbf{a} - \mathbf{l}\|^2$ as $\|\mathbf{a}\|^2 + \|\mathbf{l}\|^2 - 2\mathbf{a} \cdot \mathbf{l}$. Thus, if \mathbf{a} is close to \mathbf{l} , this implies that $\|\mathbf{a} - \mathbf{l}\|^2$ is small, and thus that $\mathbf{a} \cdot \mathbf{l}$ is relatively large. This pertains to the correlativity criterion we use to check if a landmarker (e.g. \mathbf{l}) is representative of some candidate algorithm (e.g. \mathbf{a}). However, in order to operationalise $\mathbf{a} \cdot \mathbf{l}$, we must move into the space of datasets, as depicted in Figure 3. Thus, we measure correlativity based on r [16] as:

$$r = \mathbf{a} \cdot \mathbf{l} = \cos \angle(\mathbf{a}, \mathbf{l}) = \frac{\mathbf{a} \cdot \mathbf{l}}{\|\mathbf{a}\| \|\mathbf{l}\|} = (\sum a_i l_i) / \sqrt{(\sum a_i^2)(\sum l_i^2)} \quad (1)$$

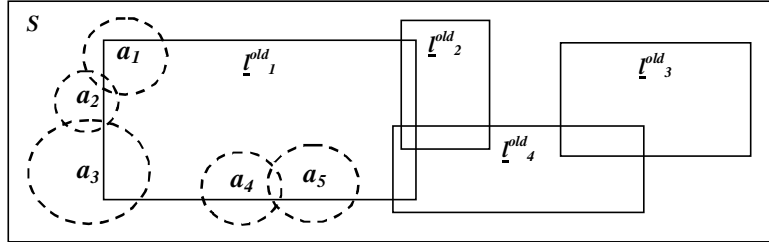


Fig. 2. A scenario in which the bias diversity criteria would fail to map the domains of expertise of the given set of candidate algorithms

It should also not be forgotten that while attempting to derive landmarkers whose measurements are correlated to the cross-validation measurements, the computational cost of running the landmarkers should not exceed the computational cost of performing cross-validation – otherwise there would be no benefit over using cross-validation! Thus, we define the following criteria for landmarkers:

- **Correlativity** – each landmarker should as closely as possible resemble their complex algorithm counterpart; fluctuations in the landmarker performance measurements (e.g. accuracy) should correlate to fluctuations in the same performance measurements of its counterpart algorithm.
- **Efficiency** – the computational cost of running the set of landmarkers should be less (and preferably significantly less) than the computational cost of running all algorithms.

Figure 4 illustrates our new perception of landmarkers in relation to the new criteria. Here, each \mathbf{l}_i corresponds to the domain of expertise of the new form of landmarker proposed by us (i.e. a close approximation of the domain of expertise of the candidate algorithm being landmarked). Essentially, by ensuring that a landmarker (i.e. a function over the domains of expertise of several algorithms) correlates to the

domain of the candidate algorithm in question, we are able to ensure (or at least quantify) the relevance of the meta-features provided by the landmarks.

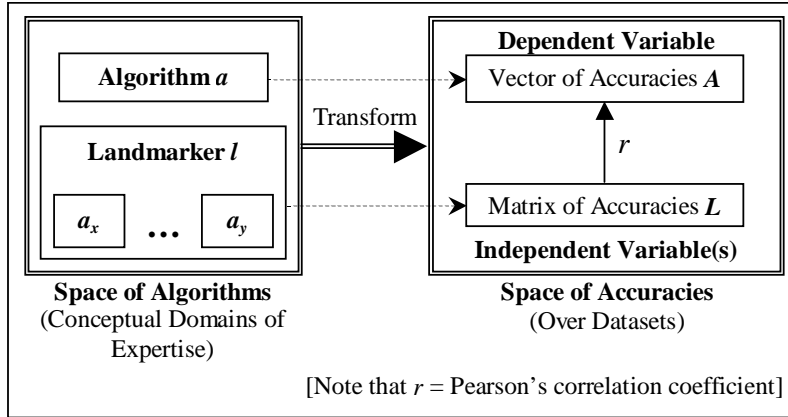


Fig. 3. An example depicting of how $a.l$ may be operationalised

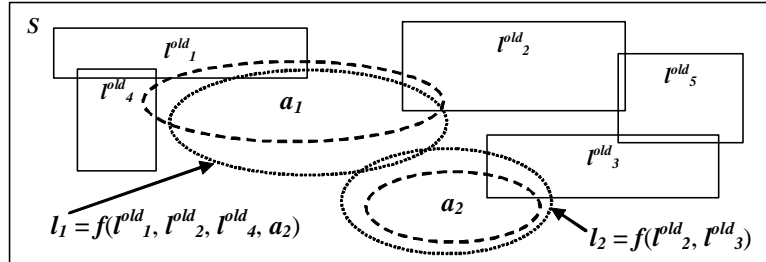


Fig. 4. A scenario illustrating the domains of expertise corresponding to the new version of landmarks under the new criteria

3 The Proposed Landmarker Generation Approach

A description of the proposed landmarker generation algorithm is given in Figure 5. The general idea of the proposed landmarker generation algorithm is to use a subset of the available algorithms as a landmarker for each of these algorithms. More specifically, given a set of candidate algorithms $A = \{a_1, \dots, a_n\}$, we seek to select a set of landmarks $L' = \{l'_1, \dots, l'_n\}$, where each l'_i is the landmarker selected for the

algorithm a_i , $l'_i \subset A$, and the union of all $l'_i \in L'$ (written as $union(L')$) is a (proper) subset of A . For example, given $A = \{a_1, a_2, a_3, a_4\}$, a possible set of selected landmarks is $L' = \{l'_1, l'_2, l'_3, l'_4\}$, where $l'_1 = \{a_1\}$, $l'_2 = \{a_2\}$, $l'_3 = \{a_1, a_2\}$, $l'_4 = \{a_2\}$.

Input: $A = \{a_1, \dots, a_n\}$, a set of candidate algorithms.
Output: $L' = \{l'_1, \dots, l'_n\}$, a corresponding set of landmarks where each l'_i is the chosen landmark for a_i .

Let:

- the powerset of A excluding ϕ and A itself be $L = \{union(L_1), \dots, union(L_m)\}$, where $m = 2^n - 2$,
- the powerset of $union(L_i)$ excluding ϕ be $L_i = \{l_1, \dots, l_p\}$, with $p = 2^{union(L_i)} - 1$,
- the mean computational cost of each l_i be denoted $eff(l_i)$, where this mean is over the training datasets, and
- the computational cost of ten-fold cross-validation on the training datasets with A be denoted $eff(A)$.

Landmarker generation algorithm:

- [1] For each $union(L_i)$:
- [2] For each $a_j \in A$:
- [3] If $a_j \notin L_i$ then:
- [4] For each $l_k \in L_i$:
- [5] Find $\psi(a_j, l_k) = r^2 + ((eff(A) - eff(l_k)) / eff(A))$, where r^2 is value of the linear regression function whose dependent = a_j and independent(s) = $\{a_x \mid a_x \in l_k\}$
- [6] Let $num_landmarkers(L_i) = num_landmarkers(L_i) + 1$
- [7] Find $best_landmarker(a_j, L_i) = l_y \mid \arg \max_{\forall l_z \in L_i} \psi(a_j, l_z) = \psi(a_j, l_y)$
- [8] Find $mean_r2(L_i) = 1 / num_landmarkers(L_i) \sum_{\forall a_j \in A} \psi(a_j, l_x) \mid l_x = best_landmarker(a_j, L_i)$
- [9] Let $L' = \{l'_i \mid l'_i = best_landmarker(a_i, L_i), L_j = \arg \max_{\forall union(L_k) \in L} mean_r2(L_k)\}$

Fig. 5. Pseudo code for the proposed landmarker generation algorithm

To choose between the various potential landmarks, we select the landmarker with the highest combined correlation and efficiency gain. More specifically, for each potential landmarker l_i , we find $r^2(l_i) + ((eff(A) - eff(l_k)) / eff(A))$. $r^2(l_i)$ is the mean coefficient of determination obtained over each $a_j - l_i$ pairing (i.e. given the independent(s) l_i , several linear regression functions can be formed, each with l_i paired with a dependent $a_j \in A$). $eff(l_i)$ and $eff(A)$ are the mean computational cost of running l_i and A (i.e. the time taken for training and testing) observed over the train-

ing datasets respectively. Note that the values used in the linear regression functions are the accuracy values observed when applying the relevant a_j to a training set of datasets.

This landmarker generation algorithm assumes:

- Several of the algorithms will have similar domains of expertise, and thus, not all have to be used.
- If a candidate algorithm has a very dissimilar domain of expertise (as compared to the other algorithms), that domain of expertise can be correlated to the conjunction of several others.

4 Experiments, Results and Analysis

For our experiments we utilise 10 classification learning algorithms from WEKA [17] (i.e. naïve Bayes, k-nearest neighbour (with $k = 1$ and 7), support vector machine, decision stump, J4.8 (a WEKA implementation of C4.5), random forest, decision table, Ripper, and ZeroR) and 34 classification datasets randomly chosen from the UCI repository [3]. To evaluate the accuracy of each candidate algorithm on each dataset, stratified ten-fold cross-validation was employed. The effectiveness of the proposed landmarker generation algorithm is evaluated using the leave-one-out cross-validation approach. This corresponds to n -fold cross-validation, where n is the number of instances, which in our case is 34, each pertaining to one UCI dataset.

For each fold we use 33 of the datasets to generate landmarkers as described in Section 3. The resultant set of landmarkers indicates which algorithms must be evaluated and which will be estimated (i.e. the algorithms in $union(L')$, and the remaining $A \setminus union(L')$ respectively). On the dataset left out, we first run the algorithms that must be evaluated and then use their accuracy results to estimate the performance of the other algorithms using the regression functions computed during landmarker generation.

The version of the algorithm described in Section 3 will attempt to find a set of landmarkers L' such that $|union(L')| < |A|$. We have modified the algorithm so that the maximum number of candidate algorithms used by a chosen set of landmarkers (i.e. $|union(L')|$, the landmarker set size of L') may be defined by the user. In our experiments we generate and test all 9 possible landmarker sets sizes ($9 \geq |union(L')| \geq 1$, given that the number of available algorithms is 10).

This leaves us with 9 sets of accuracy *estimates* for each of the candidate algorithms over each of the UCI datasets. Three evaluations are performed over the accuracy estimates:

- Efficiency gained (*EG*): for each held-out dataset and landmarker set size, we compute the percentage of computation saved by employing the landmarker. This saving is the portion of the computational time incurred by conducting ten-fold cross-validation over all the candidate algorithms that is saved by instead running only the algorithms associated with the landmarker in question. For each landmarker set size, we report the mean *EG* recorded over all datasets.

- Rank order correlation (r_s): for each held-out dataset and landmarker set size, we utilise the Spearman’s rank order correlation coefficient r_s , to determine the correlation between: (i) the rank order of the accuracies estimated via the landmarkers and regression, and (ii) the rank order of the accuracies evaluated via ten-fold cross-validation. For each landmarker set size, we report the mean r_s recorded over all datasets.
- Algorithm-pair ordering (AP): for each dataset and landmarker set size, we compare the order of each pair of algorithms (e.g. if $acc(\mathbf{a}_1) > acc(\mathbf{a}_2)$) based on the estimated (via the landmarkers and regression) and evaluated accuracies (via ten-fold cross-validation). For each landmarker set size, we report the mean (across all datasets) of the percentage of pairings in which the order is predicted correctly. Note that there are $^{10}C_2 = 45$ algorithms pairings with 10 algorithms. However, one notices that when all 10 algorithms are employed by the set of landmarkers, no landmarkers are required, and we are simply performing ten-fold cross-validation. Accordingly, for a landmarker set size of x , those x algorithms are evaluated, not estimated. Thus, xC_2 algorithm pairs will correspond to the ordering that is found via ten-fold cross-validation. We denote this as assured AP , which is the accuracy associated with pairings that are guaranteed to be correct.

Table 1. The mean efficiency gained (EG), r_s , algorithm-pair ordering (AP), and r^2 values, and the assured AP value observed from our experiments

No. Algorithms, $ union(L') $	Mean EG	Mean r_s	Mean AP	Assured AP	Mean r^2
1	92.9	0.54	71.3	0.0	0.64
2	85.4	0.59	74.1	2.2	0.81
3	84.6	0.68	77.7	6.7	0.89
4	83.1	0.73	80.6	13.3	0.92
5	82.6	0.77	82.4	22.2	0.94
6	67.0	0.83	86.1	33.3	0.94
7	69.4	0.88	89.6	46.7	0.95
8	23.8	0.92	92.7	62.2	0.96
9	8.5	0.97	96.3	80.0	0.97
10*	0.0	1.00	100.0	100.0	NA

* This corresponds to 10-fold cross-validation, which we are comparing against.

Table 1 presents the results from our experiments. It shows the mean EG , the ranking order correlation (mean r_s), the mean accuracy over algorithm-pair orderings (mean AP), the percentage of algorithm-pair orderings guaranteed to be correct (assured AP), and the mean r^2 . Each i -th row of the table presents the results of the landmarker set(s) of set size i from each fold. The results show that the landmarkers generated, when used with the linear regression models, are very encouraging. Even when only utilising a single candidate algorithm (i.e. $|union(L')| = 1$), the chosen set of landmarkers is still able to produce a reasonable result (i.e. mean $r_s = 0.54$, mean

$AP = 71.3$). The efficiency gained is also substantial (i.e. mean $EG = 92.9$, this means the landmarker only incurred 7.1% of the computational cost that is observed with ten-fold cross-validation on all ten candidate algorithms!). As expected, when we allow the generation algorithm to utilise larger sets of candidate algorithms as landmarkers (i.e. as we allow larger $|\text{union}(\mathbf{L}')|$), the r^2 , r_s , and AP all increase, and the efficiency gained decreases, with all the values approaching the ten-fold cross-validation result.

It should be noted that the computational costs of the algorithms used vary quite drastically. In fact, from Table 1, the large dip in the mean EG of the landmarkers utilised when going from 8 to 9 algorithms is primarily caused by the use of the SVM algorithm, which is significantly more computationally expensive as compared to the other candidate algorithms. However, as our results indicate, the landmarker generation algorithm selects algorithms with higher correlation and lower computational costs, before attempting to utilise the ones with similar levels of correlation, but higher computational costs.

5 Conclusions

In this paper, we have provided a new definition of landmarkers, specifying each to be: (i) a function over a set of learning algorithms, (ii) that is focused on characterising the domain of expertise of one candidate algorithm. Correspondingly, we have identified new criteria for the generation of landmarkers, in that each should be: (i) efficient, and (ii) correlated as compared with its associated algorithm. Based on these criteria, we have proposed a simple landmarker generation algorithm that considers subset combinations of the set of candidate algorithms as landmarkers. The experimental results show that even when the number of algorithms employed by the set of landmarkers is small (and thereby, the gain in efficiency is large – up to a 92.9% saving over the cost of running the entire set of candidate algorithms), the lowest rank order correlation is 0.54, which is a very promising result. Furthermore, as the number of algorithms used by the set of landmarkers increases, the accuracy of the performance estimations approaches that of ten-fold cross-validation, even though the gain in efficiency mostly sustained. These results also suggest that the heuristic employed for the landmarker selection (see Section 3) is a viable one. As future work, we would like to: (i) explore a wider variety of potential landmarkers, (ii) explore the use of different heuristics, (iii) conduct more extensive experimentation, and (iv) ground this approach in a theoretical framework.

References

1. Bensusan, H.: God doesn't always shave with Occam's Razor: learning when and how to prune. Proc. ECML (1998) 119-124

2. Bensusan, H., Giraud-Carrier, C.: Casa Batló is in Passeig de Gràcia or landmarking the expertise space. Proc. ECML, Wkshop. on Meta-learning: Building automatic advice strategies for model selection and method combination (2000) 29-46
3. Blake, C., Merz, C.: UCI repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998)
4. Brazdil, P., Gama, J., Henery, R.: Characterizing the applicability of classification algorithms using meta level learning. Proc. ECML (1994) 84-102
5. Brazdil, P., Soares, C., Costa, J.: Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning vol. 50(3) (2003) 251-277
6. Fürnkranz, J., Petrak, J.: An evaluation of landmarking variants. Proc. ECML, Wkshop. on integrating aspects of data mining, decision support and meta-learning (2001) 57-68
7. Fürnkranz, J., Petrak, J., Brazdil, P., Soares, C.: On the use of fast subsampling estimates for algorithm recommendation. Technical Report, Austrian Research Institute for Artificial Intelligence (2002)
8. Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Machine Learning vol. 54(3) (2004) 187-193
9. Kalousis, A., Hilario, M.: Feature selection for meta-learning. Proc. PAKDD (2001) 222-233
10. Kalousis, A., Hilario, M.: Model selection via meta-learning: a comparative study. Int. J. Artificial Intelligence Tools vol. 10(4) (2001) 525-554
11. Köpf, C., Taylor, C., Keller, J.: Meta-analysis: from data characterisation for meta-learning to meta-regression. Proc. PKDD, Wkshop. on Data Mining, Decision Support, Meta-learning and ILP (2000)
12. Michie, D., Spiegelhalter, D., Taylor, C.: Machine learning, neural and statistical classification. Ellis Horwood (1994)
13. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Meta-learning by landmarking various learning algorithms. Proc. ICML (2000) 743-750
14. Schaffer, C.: Technical note: selecting a classification method by cross-validation. Machine Learning vol. 13(1) (1993) 135-143
15. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. J. Artificial Intelligence Review vol. 18(2) (2002) 77-95
16. Wickens, T.: The geometry of multivariate statistics. LEA Publishers (1995)
17. Witten, I., Frank, E.: Data mining: practical machine learning tools with Java implementations. Morgan Kaufmann (2000)
18. Wolpert, D.: The supervised learning no-free-lunch theorems. Proc. Soft Computing in Industry - Recent Applications (2001) 25-42