



**The University of Sydney**

**Comparisons between Heuristics Based on Correlativity  
and Efficiency for Landmarker Generation**

Technical Report Number 557

October 2004

Daren Ler, Irena Koprinska and Sanjay Chawla

ISBN 1 86487 681 6

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

# Comparisons between Heuristics Based on Correlativity and Efficiency for Landmarker Generation

Daren Ler, Irena Koprinska, Sanjay Chawla

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

*{ler, irena, chawla}@it.usyd.edu.au*

## Abstract

*In this paper we propose a new meta-learning approach using landmarking. This approach, which is based on a new set of criteria for selecting landmarks, generates a set of landmarks that are each subsets of the candidate algorithms being landmarked.*

*In our experiments, we compare three heuristics based on correlativity and the efficiency gained. With each heuristic, the landmarks generated using linear regression were able to estimate accuracy well, even when only utilizing a small fraction of the given algorithms. The results also show that the heuristic in which efficiencies are estimated via 1-nearest neighbor outperformed the other heuristics.*

## 1 Introduction

Theoretical [16] and empirical [9] results indicate that no single algorithm is generically superior. This, coupled with a growing plethora of machine learning algorithms, makes the selection of an appropriate learning algorithm for a given dataset an increasingly important issue. Traditionally, the selection of an appropriate algorithm is done based on some form of *hold-out testing* (e.g. *cross-validation* and *bootstrapping*) [12]. 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 [6], have been proposed. Typically referred to as *meta-learning* [6, 13], such solutions employ experience on previous datasets (i.e. meta-knowledge) to learn concepts 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 superior to others.

As with 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 [3, 9, 1, 7]. However, to date, there is no consensus on how good meta-features should be chosen. *Landmarking* [11, 5] is an alternative and promising approach that characterises datasets by directly measuring the performance of simple and fast learning algorithms, called landmarks. However, the selection of landmarks is typically done in an ad hoc

fashion, with the landmarks generated focused on characterising an arbitrary set of algorithms.

In this paper, we reinterpret the role of landmarks, defining each as 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 landmarks (one landmarker for each candidate algorithm), such that each landmarker: (1) is more *efficient* than its counterpart candidate algorithm, 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). 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 landmarks and introduce the new landmarker criteria. In Section 3, we introduce a new method for landmarker generation based on the new criteria, while Section 4 describes and discusses the experiments, and corresponding results. The last section concludes the paper and suggests paths for future work.

## 2 Landmarking and Landmarkers

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 the problem. Thus, appropriate meta-features, or in our case, appropriate landmarks, must be found.

### 2.1 Redefining Landmarkers

In the literature [11, 5], 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 a meta-feature whose purpose is to help define the generic expertise space, in which the domain of expertise of *any* candidate algorithm may be defined. We introduce a new

perception of landmarks that defines *each landmarker* to be:

1. A *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 a function of the accuracies of several learning algorithms such as a decision stump, a naïve Bayes learner, etc. This 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) [11, 7]. 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 [8].

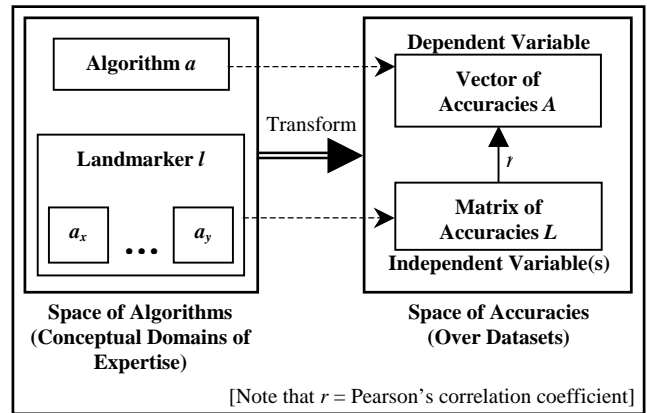
Intuitively, while one can consider an overall concept regarding the generic expertise space, this requires that the dataset characteristics be relevant in defining the domain of expertise of *any* algorithm. This meta-feature space would most likely be very complex, and thus difficult to define. Additionally, 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 [11]. 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 bias diversity criterion seeks to find a set of landmarkers that is able to characterise the space of all datasets well enough so that *any* domain of expertise (i.e. a generic expertise space) may be defined. However, the bias diversity criterion does not ensure the generation of a set of landmarkers that would characterise the right dataset features, or possess sufficient generality required to locate the domains of expertise of the given set of candidate algorithms. What landmarker criteria should we then use?

In meta-learning, we are primarily interested in the performance measurements<sup>1</sup> 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 candidate 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  $a$  and  $l$  can be regarded as  $\|a - l\|$ . Also, we may express  $\|a - l\|^2$  as  $\|a\|^2 + \|l\|^2 - 2a.l$ . Thus, if  $a$  is close to  $l$ , this implies that  $\|a - l\|^2$  is small, and thus that  $a.l$  is relatively large. This pertains to the correlativity criterion we use to check if a landmarker (e.g.  $l$ ) is representative of some candidate algorithm (e.g.  $a$ ). However, in order to operationalise  $a.l$ , we must move into the space of datasets, as depicted in Figure 1. Thus, we measure correlativity based on  $r$  [14] as:

$$r = a.l = \cos \angle(a, l) = \frac{a.l}{\|a\| \|l\|} = \frac{(\sum a_i l_i)}{\sqrt{(\sum a_i^2)(\sum l_i^2)}}$$



**Figure 1. An example depicting of how  $a.l$  may be operationalised**

It should 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-

<sup>1</sup> Essentially, all the meta-target types are functions of the performance measurements over the candidate algorithms.

validation! Thus, we define the following criteria for landmarkers:

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

## 2.3 Landmarker Generation

The landmarker criteria by themselves do not indicate how landmarkers may be generated, but merely constitute the conditions under which any landmarker should be selected. How does one then generate a set of landmarkers that satisfies these criteria?

### 2.3.1 Identifying Potential Landmarkers

The generation of a set of landmarkers satisfying the defined criteria is essentially a search problem, whose initial space includes all possible algorithms. Under this interpretation, a first logical step – as per any other search problem, would be to somehow limit this search space. One simple method is to only consider algorithms that are less complex versions of the candidate algorithms. Such modifications may be categorised into two groups:

- **Algorithm specific reductions**, which relate to the actual inner workings of one particular algorithm. This includes: limiting the structure formed (e.g. decision stumps for decision trees), and limiting the internal search mechanisms within the algorithm (e.g. by employing randomness [4]).
- **Algorithm generic reductions**, which conversely relate to generic modifications that may be applied to any learning algorithm. As done in [5], such modifications could be similar to the sub-sampling techniques found in ensemble literature [4].

### 2.3.2 Comparing Potential Landmarkers

Once a potential set of landmarkers has been generated, the landmarker criteria may be utilised to compare and eventually select the final set of landmarkers to employ. The question though, is what heuristic should be utilised to compare the landmarkers?

In terms of the efficiency criteria, simply utilising the (computational) time taken to run a given landmarker would suffice. However, measuring correlation is not as straightforward. We must choose one of the many different forms of correlation and association measures available (e.g. Pearson’s  $r$ , the coefficient of determination, etc).

Furthermore, we would like to maximise both criteria; i.e. we would like to maximise the correlation measurement (e.g. the coefficient of determination)

between the landmarker and the target algorithm being landmarked, while also maximising the efficiency or minimising the computational cost of this landmarker. In order to do so, as a heuristic, we must take some function of the two. An intuitive measure would be to take the ratio of the correlation over the computational cost. However, as has been noted in [3], straightforward measures of efficiency tend to have a much wider range of possible values than accuracy or in this case correlation, and would probably dominate the ratio. To cope with this, one proposed method is to take the logarithm of the efficiency measured, thereby dulling its effect [3]. Ultimately, any heuristic that normalises the effect of efficiency to that of correlation may be utilised.

It should also be noted that any heuristic that takes into account both correlation and efficiency makes two different inferences. The first concerns the performance of the algorithm being landmarked, while the other concerns the cost of the potential landmarkers, or rather, the computational cost of the algorithms that comprise the various potential landmarkers. The latter is an initial inference that is made to estimate the efficiency of the various potential landmarkers (i.e. sets of algorithms) on the new unseen dataset.

## 3 The Proposed Landmarker Generation Approach

A description of the proposed landmarker generation algorithm is given in Figure 2. 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 landmarkers  $L' = \{l'_1, \dots, l'_n\}$ , such that each  $l'_i$  is the landmarker selected for the algorithm  $a_i$ , where  $l'_i \subset A$ , 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 selected set of landmarkers is  $L' = \{l'_1, l'_2, l'_3, l'_4\}$ , where  $l'_1 = \{a_1\}$ ,  $l'_2 = \{a_2\}$ ,  $l'_3 = \{a_1, a_2\}$ ,  $l'_4 = \{a_2\}$ .

To choose between the various potential landmarkers, we propose evaluate the following heuristics:

- **Variante 1:**  $r^2(l_i)$ , where  $r^2(l_i)$  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  $a_j \in A$ ). 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.
- **Variante 2:**  $r^2(l_i) + (eff(A) - eff(l_i)) / eff(A)$ , where  $eff(A)$  represents the computational cost of running

each  $a_j \in A$  (i.e. this heuristic is essentially correlation plus gain in efficiency).

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 landmarker for  $a_i$ .

Let:

- the powerset of  $A$  (excluding  $\emptyset$  and  $A$  itself) be  $L = \{\text{union}(L_1), \dots, \text{union}(L_m)\}$ , where  $m = 2^n - 2$ ,
- the powerset of  $\text{union}(L_i)$  (excluding  $\emptyset$ ) be  $L_i = \{l_1, \dots, l_p\}$ , where  $p = 2^{|\text{union}(L_i)|} - 1$ ,
- $r^2(l_i)$  be the coefficient of determination of the linear regression function whose dependent =  $a_j$  and independent(s) =  $\{a_x \mid a_x \in l_i\}$ .
- the computational cost of each  $l_i$  be denoted by  $\text{eff}(l_i)$ , and the computational cost of  $A$  (via ten-fold cross-validation) be denoted by  $\text{eff}(A)$ . Note that these are evaluated by either: (i) taking the mean computational cost over the training datasets, or (ii) inferring the computational cost of  $l_i$  and  $A$  based on a k-nearest neighbour algorithm.
- the heuristic employed be either:
  - $H_1: r^2(l_i)$
  - $H_3: r^2(l_i) + ((\text{eff}(A) - \text{eff}(l_i)) / \text{eff}(A))$

Landmarker generation algorithm:

- For each  $\text{union}(L_i)$ :
- For each  $a_j \in A$ :
- If  $a_j \notin L_i$  then:
- For each  $l_k \in L_i$ :
  - Find  $\psi(a_j, l_k) = H_*$ , where  $H_* \in \{H_1, H_2, H_3\}$ .
- Let  $\text{num\_landmarkers}(L_i) = \text{num\_landmarkers}(L_i) + 1$
- Find  $\text{best\_landmarker}(a_j, L_i) = l_y \mid \arg \max_{\forall l_z \in L_i} \psi(a_j, l_z) = \psi(a_j, l_y)$ .
- Find  $\text{mean\_r2}(L_i) = 1 / \text{num\_landmarkers}(L_i) * \sum_{\forall a_j \in A} \psi(a_j, l_x) \mid l_x = \text{best\_landmarker}(a_j, L_i)$ .
- Let  $L' = \{l_i' \mid l_i' = \text{best\_landmarker}(a_i, L_i), L_j = \arg \max_{\forall \text{union}(L_k) \in L} \text{mean\_r2}(L_k)\}$ .

**Figure 2. Pseudo code for the proposed landmarker generation algorithm**

Additionally, we propose two different methods of evaluating  $\text{eff}(l_i)$  and  $\text{eff}(A)$ :

- Variants 2a:** Calculate  $\text{eff}(..)$  by computing the mean computational cost of the algorithms involved over all the training datasets.
- Variants 2b:** Utilise a weighted k-nearest neighbour (K-NN) algorithm [10] over the training datasets to infer the efficiency (i.e. in this case, the computational cost) of the dataset(s) being evaluated. In terms of the features employed, we simply use the number of instances and number of attributes. Note that as this method is a lazy learning one, it requires that all the potential landmarks are stored as we may only make our landmarker choice when the new dataset is encountered. The construction of the potential landmarks is still done during training. This sub-inference can be seen as a modular

component of the algorithm, which may be replaced by another (perhaps, more competent) mechanism (recall Section 2.3.2).

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 [15] (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 [2]. 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 then 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 the various sets of landmarks (i.e. the landmarks generated with the 3 variants of the landmarker generation algorithm) as described in Section 3. With each variant, the resultant set of landmarks indicates which algorithms must be evaluated and which will be estimated (i.e. the algorithms in  $\text{union}(L')$ , and the remaining  $A \setminus \text{union}(L')$  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.

Each version of the algorithm described in Section 3 will attempt to find a set of landmarks  $L'$  such that  $|\text{union}(L')| < |A|$ . We have modified the algorithm so that the maximum number of candidate algorithms used by a chosen set of landmarks (i.e.  $|\text{union}(L')|$ , 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 ( $9 \geq |\text{union}(L')| \geq 1$ , 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  $acc(a_1) > acc(a_2)$ ) 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  $\binom{10}{2} = 45$  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,  $\binom{x}{2}$  algorithm pairs will mimic the ten-fold cross-validation orderings. We denote this as *Assured AP*, which is the accuracy associated with pairings that are guaranteed to be correct.

Tables 1, 2 and 3 present the results from our experiments for Variant 1, 2a and 2b respectively. In each table, the mean efficiency gained ( $EG$ ), 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  $r^2$  are reported. Each  $i$ -th column of each table presents the results of the landmarker set(s) of set size  $i$  from each fold.

For all the variants evaluated, 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.  $|\text{union}(L')| = 1$ ), the chosen set of landmarkers is still able to produce a reasonable result (i.e. mean  $r_s = 0.55, 0.54, 0.53$  and mean  $AP = 71.5, 71.31, 70.78$  for Variant 1, 2a and 2b respectively). The efficiency gained is also substantial (i.e. mean  $EG = 92.42, 92.92, 96.83$  for Variant 1, 2a and

2b respectively, which means the landmarkers in each version incur less than 10% of the computational cost that is observed with ten-fold cross-validation on all ten candidate algorithms!).

**Table 1. Results of Variant 1**

	No. Algorithms Used, $ L' $								
	1	2	3	4	5	6	7	8	9
Mean $EG$	92.42	85.33	83.63	83.09	82.73	44.10	52.11	23.83	8.44
Mean $r_s$	0.55	0.59	0.68	0.73	0.79	0.81	0.86	0.92	0.97
Mean $AP$	71.50	74.05	78.04	80.52	82.94	85.23	88.69	93.14	96.34
Assured $AP$	0.00	2.22	6.67	13.33	22.22	33.33	46.67	62.22	80.00
Mean $r^2$	0.64	0.81	0.89	0.92	0.94	0.95	0.96	0.97	0.98

**Table 2. Results of Variant 2a**

	No. Algorithms Used, $ L' $								
	1	2	3	4	5	6	7	8	9
Mean $EG$	92.92	85.44	84.57	83.08	82.60	66.98	69.45	23.83	8.47
Mean $r_s$	0.54	0.59	0.68	0.73	0.77	0.83	0.88	0.92	0.97
Mean $AP$	71.31	74.05	77.71	80.59	82.35	86.08	89.61	92.75	96.34
Assured $AP$	0.00	2.22	6.67	13.33	22.22	33.33	46.67	62.22	80.00
Mean $r^2$	0.64	0.81	0.89	0.92	0.94	0.94	0.95	0.96	0.97

**Table 3. Results of Variant 2b**

	No. Algorithms Used, $ L' $								
	1	2	3	4	5	6	7	8	9
Mean $EG$	96.83	96.86	96.34	88.78	85.53	80.19	60.18	25.87	12.50
Mean $r_s$	0.53	0.60	0.70	0.74	0.78	0.83	0.87	0.92	0.96
Mean $AP$	70.78	74.05	78.82	81.50	83.66	86.47	89.02	92.55	95.69
Assured $AP$	0.00	2.22	6.67	13.33	22.22	33.33	46.67	62.22	80.00
Mean $r^2$	0.64	0.81	0.87	0.90	0.92	0.93	0.93	0.94	0.95

Correspondingly, and as expected, when we allow the generation algorithm to utilise larger sets of candidate algorithms as landmarkers (i.e. as we allow larger  $|\text{union}(L')|$ ), the  $r^2$ ,  $r_s$ , and  $AP$  all increase, while the  $EG$  decreases. However, it should be noted that the high saving in computational cost experienced with Variant 1 of the landmarker generation algorithm is surprising as we do not specifically attempt to select more efficient algorithms (i.e. the heuristic used in Variant 1 does not take into account efficiency; thus any savings can only be attributed to the smaller number of algorithms evaluated).

In fact, the computational costs of the algorithms used vary quite drastically. From the three tables, we see larger reductions in  $EG$  as more algorithms are employed, indicating that more expensive algorithms are added only after the more efficient versions. Given the similarity of the results from Variant 1 to the other 2 variants (which incorporate efficiency into the heuristic), we may also

conclude that the algorithms that achieve high correlation also seem to be more efficient.

Further, when comparing Table 2 and 3 we see that there is an obvious improvement in the efficiency gained (i.e. mean  $EG$ ). This is due to the more precise estimation of the efficiency of the unseen dataset using 1-NN. At the same time the correlation results (i.e. mean  $r_s$  and  $AP$ ) remain similar. We may conclude that use of the 1-NN algorithm to estimate the efficiency of the unseen dataset leads to better landmarks being selected.

In another experiment, we also attempted to use a 5-NN mechanism to estimate efficiency. Although these results outperformed Variant 2a, they were not as good as the 1-NN version. This is because we only employ the number of instances and attributes as our features. As indicated in Section 2, many other meta-features may serve to enhance this estimation.

## 6 Conclusions

In this paper, we have provided a new definition of landmarks, specifying each to be: (i) 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 landmarks, in that each should be: (i) efficient, and (ii) correlated as compared with its associated algorithm. Via a landmarker generation algorithm that considers subset combinations of the set of candidate algorithms as landmarks, we compare three heuristics based on the proposed criteria. The experimental results show that even when the set of landmarks selected consists of a single algorithm, the lowest rank order correlation among the 3 variants is 0.53, which is a very promising result. Furthermore, for each heuristic employed, as the number of algorithms used by the set of landmarks increases, the accuracy of the performance estimations approaches that of ten-fold cross-validation, even though the gain in efficiency mostly sustained. Also, when a more precise estimation of efficiency is made (i.e. via 1-NN), the results (in particular, efficiency) are enhanced. As future work, we would like to: (i) enhance the efficiency of the landmarker generation algorithm via some form of stepwise regression, (ii) explore a wider variety of potential landmarks, (iii) explore the use of different heuristics, (iv) conduct more extensive experimentation, and (v) ground this approach in a theoretical framework.

## Acknowledgements

The authors would like to acknowledge the support of the Smart Internet Technology CRC in this research.

## References

- [1] Bensusan, H., "God doesn't always shave with Occam's Razor: learning when and how to prune," *Proc. ECML*, 119-124, 1998.
- [2] Blake, C., & Merz, C., "UCI repository of machine learning databases," *University of California, Irvine, Dept. of Information and Computer Sciences*, 1998.
- [3] Brazdil, P., Soares, C., & Costa, J., "Ranking learning algorithms: Using IBL and meta-learning on accuracy & time results," *Machine Learning*, 50(3), 251-277, 2003.
- [4] Dietterich, T., "Machine-learning research: 4 current directions," *AI Magazine*, 18(4), 97-136, 1997.
- [5] Fürnkranz, J., and Petrak, J., "An evaluation of landmarking variants," *Proc. ECML, Wkshop. on integrating aspects of data mining, decision support and meta-learning*, 57-68, 2001.
- [6] Giraud-Carrier, C., Vilalta, R., and Brazdil, P., "Introduction to the special issue on meta-learning," *Machine Learning*, 54(3), 187-193, 2004.
- [7] Kalousis, A., and Hilario, M., "Feature selection for meta-learning," *Proc. PAKDD*, 222-233, 2001.
- [8] Köpf, C., Taylor, C., and 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.
- [9] Michie, D., Spiegelhalter, D., and Taylor, C., "Machine learning, neural and statistical classification," *Ellis Horwood*, 1994.
- [10] Mitchell, T., "Machine Learning," McGraw-Hill, 1997.
- [11] Pfahringer, B., Bensusan, H., and Giraud-Carrier, C., "Meta-learning by landmarking various learning algorithms," *Proc. ICML*, 743-750, 2000.
- [12] Schaffer, C., "Technical note: selecting a classification method by cross-validation," *Machine Learning*, 13(1), 135-143, 1993.
- [13] Vilalta, R., and Drissi, Y., "A perspective view and survey of meta-learning," *Journal of Artificial Intelligence Review*, 18(2), 77-95, 2002.
- [14] Wickens, T., "The geometry of multivariate statistics," *LEA Publishers*, 1995.
- [15] Witten, I., and Frank, E., "Data mining: practical machine learning tools with Java implementations," *Morgan Kaufmann*, 2000.
- [16] Wolpert, D., "The supervised learning no-free-lunch theorems," *Proc. Soft Computing in Industry - Recent Applications*, 25-42, 2001.