



# The University of Sydney

## **A Proposed Meta-learning Framework for Algorithm Selection Utilising Regression-based Landmarkers**

Technical Report Number 569

May 2005

Daren Ler, Irena Koprinska and Sanjay Chawla

ISBN 1 86487 721 9

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

---

# A Proposed Meta-learning Framework for Algorithm Selection Utilising Regression-based Landmarkers

---

Daren Ler  
Irena Koprinska  
Sanjay Chawla

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

LER@IT.USYD.EDU.AU  
IRENA@IT.USYD.EDU.AU  
CHAWLA@IT.USYD.EDU.AU

## Abstract

In this paper, we present a framework for meta-learning that adopts the use of regression-based landmarks. Each such landmarker exploits the correlations between the various patterns of performance for a given set of algorithms so as to construct a regression function that represents the pattern of performance of one algorithm from that set. The idea is that the independents utilised by these regression functions – i.e. landmarks – correspond to the performance of a subset of the given algorithms. In this manner, we may control the number of algorithms being landmarked; the more that are landmarked, the fewer independents or evidence we have to make those approximations, and less accurate the landmarks are. We investigate the ability of such landmarks in combination with meta-learners to learn how to predict the most accurate algorithm from a given set. While our results show that the accuracy of the meta-learning solutions increases as the quality of the meta-attributes improves; i.e. when less algorithm performance measurements are landmarked and instead evaluated as independents, we find that in general, the results are still poor. However, we find that when a simple sorting mechanism is instead employed, the results are quite promising.

## 1. Introduction

The selection of the most adequate learning algorithm for a given dataset is an important problem. If unlimited time were available to make this decision, hold-out testing (e.g. cross-validation or bootstrapping) could be used to evaluate the performance of all applicable algorithms and thus determine which should be utilised – e.g. (Schaffer, 1993). However, such evaluation is computationally

unfeasible due to the large number of available algorithms. To overcome this limitation, various algorithm selection methods have been proposed.

Typically referred to as a kind of meta-learning (Giraud-Carrier et al., 2004; Vilalta & Drissi, 2002), these algorithm selection solutions utilise experience on previous datasets (i.e. meta-knowledge) to learn how to characterise the areas 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.) Predominantly, such solutions involve the mapping of dataset characteristics to the domains of expertise of some set of candidate algorithms.

Recently, the concept of landmarking (Fürnkranz & Petrak, 2001; Ler et al., 2004a; Pfahringer et al., 2000) has emerged as a technique that characterises a dataset by directly measuring the performance of simple and fast learning algorithms, called landmarks.

In (Ler et al., 2004b) we proposed landmarker selection criteria based on efficiency and correlativity, and based on them a landmarker generation approach. This approach exploits the correlations between the various patterns of performance of a given set of algorithms to construct landmarks that each correspond to a regression function, with the independents for these regression functions consisting of the performance measurements of a subset of the given algorithms. Accordingly, we refer to these as regression-based landmarks.

In this paper, we consider the use of these landmarks in a meta-learning framework for algorithm selection; we evaluate and discuss the proposed framework on the algorithm selection task of predicting the most accurate candidate algorithm from a given set.

## 2. Meta-learning via Landmarking

Various meta-learning approaches have been proposed to perform algorithm selection (Aha, 1992; Brazdil et al., 2003; Gama & Brazdil, 1995; Kalousis and Hilario, 2001; Lindner & Studer, 1999; Michie et al., 1994; Pfahringer et al., 2000; Todorovski et al., 2002). The predominant

---

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

strategy is to describe learning tasks in terms of a set of meta-attributes and classify them based on some aspect of the performance of the set of candidate algorithms (e.g. which among the candidate algorithms will give the best performance on the learning problem in question).

To date, three types of meta-attributes have been suggested: (i) dataset characteristics, including basic, statistical and information theoretic measurements (Brazdil et al., 2003; Gama & Brazdil, 1995; Kalousis and Hilario, 2001; Lindner & Studer, 1999; Michie et al., 1994); (ii) properties of induced classifiers over the dataset in question (Bensusan, 1998; Peng et al., 2002); and (iii) measurements that represent the performance or other output of representative classifiers (to the candidate algorithms); i.e. landmarks (Fürnkranz & Petrak, 2001; Ler et al., 2004a; Pfahringer et al., 2000). Correspondingly, the types of algorithm selection problems that have been suggested include: (i) classifying an algorithm as appropriate or inappropriate on the learning task in question (given that an algorithm is appropriate if it is not considered worse than the best performing candidate algorithm) (e.g. see Gama & Brazdil, 1995; Michie et al., 1994), (ii) classifying which of two specific candidate algorithms is superior (e.g. see Fürnkranz & Petrak, 2001; Pfahringer et al., 2000), (iii) classifying the best performing algorithm for the learning task in question (e.g. see Pfahringer et al., 2000), and (iv) classifying the set of rankings for all the candidate algorithms (e.g. see Brazdil et al., 2003).

In this paper, we focus primarily on the type of meta-attributes used for algorithm selection problems, and in particular, on solutions employing landmarks.

## 2.1 Landmarkers and Landmarking

Traditionally, a landmarker is associated with a single algorithm with low computational complexity. The general idea is that the performance of a learning algorithm on a dataset reveals some characteristics of that dataset. However, in the initial landmarking work (Fürnkranz & Petrak, 2001; Pfahringer et al., 2000), despite the presence of two landmarker criteria (i.e. efficiency and bias diversity), no actual mechanism for generating appropriate landmarks were defined, and the choice of landmarks was made in an ad hoc fashion. Subsequently, Fürnkranz & Petrak (2001) proposed to generate landmarks using: (i) the candidate algorithms themselves, but only on a sub-sample of the given data (called sampling-based landmarks), (ii) the relative performance of each pair of candidate algorithms (called relative landmarks), and (iii) a combination of both (i) and (ii). However, we note that to compute relative landmarks (in the absence of sampling), we are required to evaluate the performance of the candidate algorithms themselves, making the meta-learning task(s) redundant. Also, when adopting sampling-based landmarks, the question of appropriate sample size is difficult to solve – one might even assume that some sub-samples might not

be indicative enough of the learning task in question and thus not capture the right dataset characteristics.

The fundamental difficulty with landmarks is that we must find algorithms that are: (i) efficient (i.e. more efficient than the set of candidate algorithms), and (ii) able to describe the datasets so that the regions of expertise of the set of candidate algorithms are well represented. The sample-based and relative landmarks proposed in (Fürnkranz & Petrak, 2001) take a step closer toward this goal since those landmarks would potentially map similar regions of expertise.

## 2.2 Regression-based Landmarkers

Consequently, in (Ler et al., 2004a), we proposed alternate landmarker selection criteria (i.e. efficiency and correlativity) and correspondingly propose landmarks based on the regression estimators whose independents correspond to a subset of the set of candidate algorithms.

Essentially, we wish the chosen landmarks (i.e. meta-attributes) to capture the patterns of performance of the given set of candidate algorithms; in other words, to be correlated to the fluctuations in performance of the candidate algorithms. As such, we propose that each candidate algorithm be represented by a regression function that would be indicative of the performance of the candidate algorithm being landmarked. Further, in order to preserve efficiency (and thus the benefit of this type of meta-learning) we require that the independents of these regression functions correspond to a subset of the candidate algorithms. In this manner, we utilise meta-knowledge regarding the correlativity between the candidate algorithms to infer the performance of the subset that is not evaluated.

More specifically, given a set of candidate algorithms  $\mathcal{A} = \{a_1, \dots, a_m\}$ , and a set of datasets  $\mathcal{S} = \{s_1, \dots, s_n\}$ , let the pattern of performance of each  $a_i \in \mathcal{A}$  be the vector  $PP(a_i) = \{performance(a_i, s_1), \dots, performance(a_i, s_n)\}$ , which describes the performance of  $a_i$  over each  $s_j \in \mathcal{S}$ . Thus, the landmarker for an algorithm  $a_j$  is an estimate of  $PP(a_j)$  based on some (regression) function  $f(a_k | a_k \in \mathcal{B} \subset \mathcal{A}')$ , where  $\mathcal{A}' \subset \mathcal{A} \setminus a_j$ . Consequently, we only require landmarks for a subset of  $\mathcal{A}$ , as the complement set is actually evaluated and used to by the landmarks. As such, depending on the amount of computation we wish to save, we may adjust the sizes of either set; the more computation we save (i.e. the less algorithm performance measurements that are evaluated), the less evidence we provide for the landmarks and the less accurate the predictions of the performance of the corresponding landmarked candidate algorithms. This conceptualisation of landmarking is depicted in Figure 1.

Thus far, we have only utilised linear regression functions. This is because they represent the simplest relationships between the patterns of performance possible, and because they are relatively inexpensive.

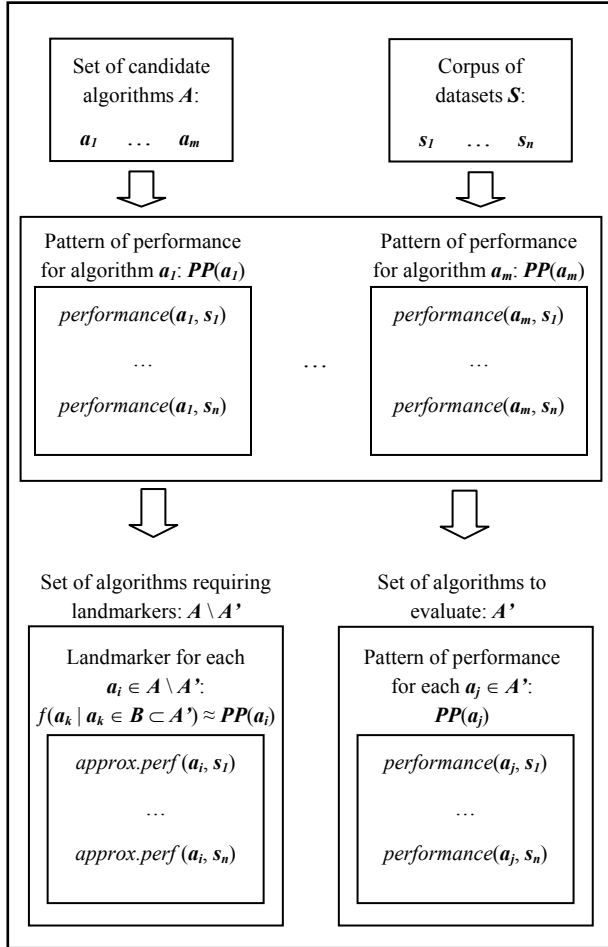


Figure 1. The proposed regression-based landmarks. For the set of candidate algorithms  $A$ , only the algorithms in the subset  $A \setminus A'$ , require landmarks; the patterns of performance from the algorithms in the complement subset  $A'$  serve as potential independents for the regression functions  $f(a_k | a_k \in B \subset A')$  that estimate the patterns of performance of the landmarked algorithms (i.e.  $f(a_k | a_k \in B \subset A') \approx PP(a_i)$ ).

The actual generation method is discussed in greater detail in (Ler et al., 2004a, 2004b), while a more efficient hill-climbing version of the proposed landmarker generation algorithm is described in (Ler et al., 2005).

### 2.3 Meta-learning via Regression-based Landmarkers

Landmarkers by themselves may provide adequate meta-knowledge regarding the regions of expertise of the candidate algorithms; as shown in (Fürnkranz & Petrak, 2001; Ler et al., 2004a, 2004b; Pfahringer et al., 2000). However, the quality of such results could potentially be further enhanced by applying them within a meta-learning framework.

In essence, the proposed meta-learning framework is expected to perform the following: (a) the base-learning task(s) – learn to approximate the patterns of performance of a subset of candidate algorithms via regression-based

landmarkers, which use (i.e. whose independents are) the evaluated performance scores from the complement subset, (b) the meta-learning task(s) – learn to map the various evaluated and estimated algorithm performance measurements to classes concerning the comparisons between the candidate algorithms (e.g. the targets described in Section 2.1, such as the best performing algorithm among the set of candidate algorithms, or the superiority of algorithm  $a_1$  versus  $a_2$ ). The proposed meta-learning framework is depicted in Figure 2.

For both the base and meta-learning (algorithm selection) tasks, more than one task may be defined. For the base-learning task, the number of tasks corresponds to the number of candidate algorithms whose performance we wish to estimate (i.e. to landmark). The number of meta-learning tasks on the other hand, depends on how we decide to structure the algorithm selection solution. For example, if we decide to learn the superiority between each pair of candidate algorithms, then  ${}^{|A|}C_2$  meta-learning tasks must be solved; alternatively, should we decide to simply learn which algorithm is overall superior, then we could adopt a single meta-learning task.

Obviously, the actual learning mechanism utilised for any base or meta-learning (algorithm selection) task may correspond to any applicable learning solution, including (model combination) meta-learning ones (e.g. stacking). For the sake of clarity, we abstract the learner involved, and reserve the term meta-learner (i.e.  $ML$  in Figure 2) for solution(s) to the meta-learning task(s).

Following Figure 2, given a performance generation mechanism (e.g. stratified ten-fold cross validation) to measure one or more performance measurements (e.g. accuracy, precision, recall,  $F$ -score, etc), we may obtain the raw meta-data characterising the dataset under scrutiny in terms of the set of candidate algorithms. This raw meta-data may then be used to generate the meta-class  $MC$  and (indirectly) the meta-attributes  $MA$  for each algorithm selection meta-learning task.  $MA$  are generated via the base-learning tasks solved by the generated regression-based landmarks.

To utilise the solutions (i.e. meta-classifiers) on a new dataset  $s_{new}$ , we require that: (i) the performance of the algorithms in  $A'$  be evaluated on  $s_{new}$ , then (ii) these measurements (i.e. the  $performance(a_k, s_{new})$  score for each  $a_k$  in  $A'$ ) will then be used to infer the approximate performance values of the remaining algorithms, and finally (iii) all the performance measurements and approximations (or some subset of them) are input into the meta-classifiers to generate the meta-classes, or rather, the algorithm selection predictions.

## 3. Experimental Setup

For our experiments we utilise a set of candidate algorithms (i.e.  $A$ ) consisting of 6 classification learning algorithms from WEKA (Witten & Frank, 2000) (i.e.

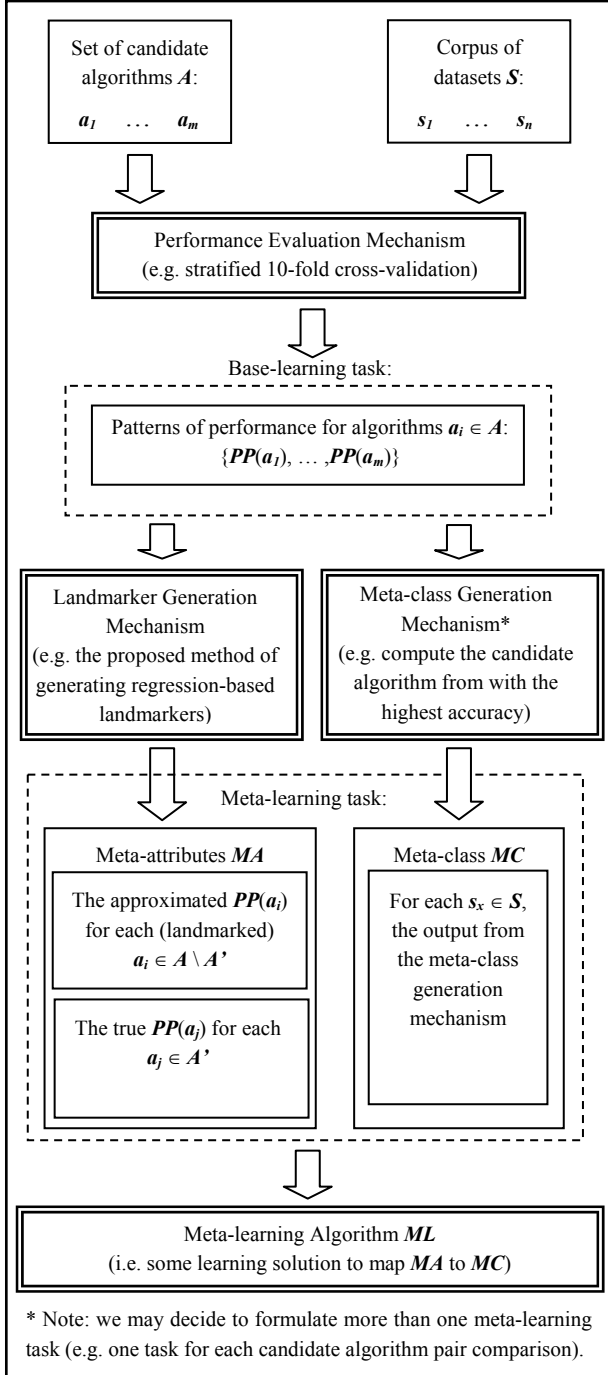


Figure 2. The proposed meta-learning framework that adopts the regression-based landmarks. The landmarks are employed to generate part of the meta-attributes  $MA$ , which are then combined to some meta-class  $MC$  to form the meta-learning task.

naive Bayes –  $N.Bayes$ , k-nearest neighbour (with  $k = 5$ ) –  $IB5$ , a polynomial kernel SVM –  $SMO$ , a RBF kernel SVM –  $SMO-R$ , a WEKA implementation of C4.5 –  $J4.8$ , and Ripper –  $JRip$ ); and as our corpus of datasets (i.e.  $S$ ) 80 classification datasets from the UCI repository (Blake

& Merz, 1998). To evaluate the performance of each candidate algorithm on each dataset (i.e. as our performance evaluation mechanism), accuracy based on stratified ten-fold cross-validation was employed.

The effectiveness of the proposed meta-learning framework 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 80, each pertaining to one UCI dataset.

For each fold we use 79 of the datasets to generate the regression-based landmarks as described Section 2.2. The resultant set of landmarks indicates which algorithms must be evaluated and which will be estimated (i.e.  $A'$  and  $A \setminus A'$  respectively). Recall from Section 2.1 that we may vary the number of candidate algorithm performance measurements that are estimated by the regression-based landmarks and thus, the number of candidate algorithm performance measurements that are actually evaluated. Given  $|A| = 6$ , we may generate the regression-based landmarks utilising a subset of independents whose size ranges from 1 to 5 (i.e. have  $1 \leq |A'| < 6$ ). Additionally, we evaluate the outputs from two different regression-based landmarker sets, each generated utilising two different criteria (i.e.  $r^2$  and  $r^2 + efficiency.gained$  – see (Ier et al., 2004b) for details). Thus, in our experiments, we generate 10 sets of meta-attributes (i.e.  $MA_{r^2,1}, \dots, MA_{r^2,5}$  and  $MA_{r^2+EG,1}, \dots, MA_{r^2+EG,5}$ ), each utilising the outputs from the regression-based landmarks generated based on one of the two criteria, and using one of the available independent sets.

As our meta-learning task, we attempt to map the meta-attributes to a meta-class (i.e.  $MC$ ) indicating the candidate algorithm with the highest accuracy. More specifically, each instance in our meta-learning problem consists of the performance evaluations/estimations of the 6 candidate algorithms on one of the UCI datasets ( $MA$ ), and the index of the candidate algorithm that attained the highest accuracy on that dataset ( $MC$ ). It should be noted that the determination of this best performing algorithm is done in simplistic fashion by directly comparing the stratified ten-fold cross-validation accuracies of the candidate algorithms.

For each (leave-one-out) fold, we thus have 5 datasets for each criterion; one for each meta-attribute set, meta-class pairing (i.e.  $(MA_1, MC), \dots, (MA_5, MC)$ ), and thus 10 datasets in total.

For potential meta-learning algorithms, we employ the same 6 WEKA algorithms, and the default class (i.e. the ZeroR WEKA algorithm). This means that we have 7 meta-classifiers for each dataset, and thus, a total of 35 classifiers per (leave-one-out) fold.

These classifiers are then tested on the instance (i.e. representing the UCI dataset) that was left out. Note that to obtain either some  $MA_{r^2,i}$  or  $MA_{r^2+EG,i}$  for this instance,

we use the performance measurements obtained via evaluation (for the algorithms in the respective  $\mathcal{A}'$ ) and estimation (for the algorithms in the corresponding  $\mathcal{A} \setminus \mathcal{A}'$ ). In the latter case, it should further be noted that the performance measurements of the current test instance (i.e. UCI dataset) were not used to train the regression-based landmarks that are used.

Let  $best.acc(i)$ , and  $worst.acc(i)$  denote the stratified ten-fold cross-validation accuracies of the most and least accurate candidate algorithms respectively on the test dataset used in fold  $i$ . Also, let  $prediction.acc(i)$  be the accuracy of the algorithm that is predicted by the meta-classifier to be the most accurate candidate algorithm on the test dataset used in fold  $i$ .

To grade the success of these classifiers, we measure the following:

1. Classification accuracy (*Acc*): the proportion of test instances (i.e. leave-one-out folds) in which the classifier made a correct prediction of the most accurate algorithm.
2. Average rank (*Rank*): the average rank of the candidate algorithm predicted to be most accurate (i.e. an indication of the rank of the algorithm predicted as the most accurate).
3.  $E[prediction.acc(i) - best.acc(i)]$  for  $i = 1 \dots 80$  (*PtoB*): the mean difference between the accuracy of the predicted most accurate candidate algorithm and actual most accurate algorithm over all the test datasets.
4.  $E[prediction.acc(i) - worst.acc(i)]$  for  $i = 1 \dots 80$  (*PtoW*): the mean difference between the accuracy of the predicted most accurate candidate algorithm and actual least accurate algorithm over all the test datasets.

#### 4. Results and Discussion

The results of the experiments are described in Table 1 through to Table 5. Each table reports the success of the various meta-learners trained using meta-attributes generated via a specific number of regression-based landmarks. That is, Table 1 reports the results based on meta-attributes obtained by evaluating the performance measurements of one candidate algorithm and estimating the remaining five, while Table 2 reports the results based on two evaluated performance measurements and four estimated ones, and so forth. In addition to the results of the meta-learners, we also list the results obtained by directly sorting the meta-attributes in question (listed as *sorted best*). Also note that in each table, we report the results obtained via the meta-attributes generated based on both the  $r^2$  (*crit.1*) and  $r^2 + efficiency.gained$  (*crit.2*) criteria (listed in rows 2 through 8 and 9 through 15 respectively). As a baseline, the accuracy based on the default class is also listed in row 1 in each of these tables.

Table 1. The results based on the meta-attributes generated via 5 regression-based landmarks. The remaining 1 accuracy measurement was directly evaluated.

META-LEARNER	ACC	RANK	PTOW	PTOB
ZEROR (DEF. CLASS)	32.5	2.96	11.2	-3.6
IB5 (CRIT.1)	41.3	2.56	11.9	-2.9
J4.8 (CRIT.1)	38.8	2.58	12.0	-2.8
JRIP (CRIT.1)	40.0	2.54	12.4	-2.4
N.BAYES (CRIT.1)	33.8	2.76	11.5	-3.3
SMO-R (CRIT.1)	42.5	2.32	12.3	-2.5
SMO (CRIT.1)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.1)	28.8	2.63	12.4	-2.4
IB5 (CRIT.2)	28.8	3.13	11.1	-3.7
J4.8 (CRIT.2)	27.5	2.99	11.2	-3.6
JRIP (CRIT.2)	31.3	2.94	11.0	-3.8
N.BAYES (CRIT.2)	18.8	3.63	8.8	-6.0
SMO-R (CRIT.2)	26.3	3.12	10.9	-3.9
SMO (CRIT.2)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.2)	31.3	2.83	11.7	-3.1

Table 2. The results based on the meta-attributes generated via 4 regression-based landmarks. The remaining 2 accuracy measurements were directly evaluated.

META-LEARNER	ACC	RANK	PTOW	PTOB
ZEROR (DEF. CLASS)	32.5	2.96	11.2	-3.6
IB5 (CRIT.1)	35.0	2.65	12.2	-2.6
J4.8 (CRIT.1)	33.8	2.68	12.1	-2.7
JRIP (CRIT.1)	40.0	2.54	12.4	-2.4
N.BAYES (CRIT.1)	36.3	2.73	11.0	-3.8
SMO-R (CRIT.1)	38.8	2.48	11.7	-3.1
SMO (CRIT.1)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.1)	30.0	2.51	12.5	-2.3
IB5 (CRIT.2)	43.8	2.51	11.7	-3.1
J4.8 (CRIT.2)	47.5	2.33	11.7	-3.1
JRIP (CRIT.2)	45.0	2.46	12.3	-2.5
N.BAYES (CRIT.2)	31.3	2.91	11.4	-3.4
SMO-R (CRIT.2)	27.5	2.86	11.5	-3.3
SMO (CRIT.2)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.2)	42.5	2.07	13.6	-1.2

From these results, we notice that in general, the accuracy of the meta-learning solutions tends to increase as the quality of the meta-attributes increase (i.e. when more performance measurements are evaluated instead of estimated). However, this improvement is far more pronounced in the solutions where the meta-attributes are simply sorted, and the best chosen from that ranking.

For the meta-attribute sets generated using more estimated than evaluated accuracy measurements (i.e.

Table 3. The results based on the meta-attributes generated via 3 regression-based landmarks. The remaining 3 accuracy measurements were directly evaluated.

META-LEARNER	ACC	RANK	PTOW	PTOB
ZEROR (DEF. CLASS)	32.5	2.96	11.2	-3.6
IB5 (CRIT.1)	47.5	2.35	12.7	-2.1
J4.8 (CRIT.1)	37.5	2.74	11.5	-3.3
JRIP (CRIT.1)	40.0	2.59	11.7	-3.1
N.BAYES (CRIT.1)	33.8	2.86	11.3	-3.5
SMO-R (CRIT.1)	28.8	2.83	11.5	-3.3
SMO (CRIT.1)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.1)	41.3	2.19	13.0	-1.8
IB5 (CRIT.2)	50.0	2.26	13.1	-1.7
J4.8 (CRIT.2)	50.0	2.26	13.1	-1.7
JRIP (CRIT.2)	43.8	2.53	12.3	-2.5
N.BAYES (CRIT.2)	26.3	2.95	11.1	-3.7
SMO-R (CRIT.2)	27.5	2.78	11.5	-3.3
SMO (CRIT.2)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.2)	55.0	1.85	13.9	-0.9

Table 4. The results based on the meta-attributes generated via 2 regression-based landmarks. The remaining 4 accuracy measurements were directly evaluated.

META-LEARNER	ACC	RANK	PTOW	PTOB
ZEROR (DEF. CLASS)	32.5	2.96	11.2	-3.6
IB5 (CRIT.1)	45.0	2.36	12.5	-2.3
J4.8 (CRIT.1)	42.5	2.46	12.2	-2.6
JRIP (CRIT.1)	42.5	2.51	12.2	-2.6
N.BAYES (CRIT.1)	32.5	2.69	11.8	-3.0
SMO-R (CRIT.1)	26.3	2.91	11.0	-3.8
SMO (CRIT.1)	32.5	2.96	11.2	-3.5
SORTED BEST (CRIT.1)	52.5	1.82	14.1	-0.8
IB5 (CRIT.2)	48.8	2.23	12.8	-2.0
J4.8 (CRIT.2)	33.8	2.74	11.0	-3.8
JRIP (CRIT.2)	38.8	2.63	12.0	-2.8
N.BAYES (CRIT.2)	30.0	2.74	11.8	-3.0
SMO-R (CRIT.2)	27.5	2.89	11.4	-3.4
SMO (CRIT.2)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.2)	65.0	1.62	14.2	-0.6

Table 1 and 2), we find that the performance of the meta-learning solutions tends to fair better than directly sorting the meta-attributes. When there are an equal number of evaluated and estimated accuracy meta-attributes (i.e. Table 3), the solution based on direct sorting approaches the performance of the best performing meta-learning solution. And when more evaluated than estimated meta-attributes are utilised (i.e. Table 4 and 5), the sorting solution clearly outperforms the meta-learning solutions.

Table 5. The results based on the meta-attributes generated via 1 regression-based landmark. The remaining 5 accuracy measurements were directly evaluated.

META-LEARNER	ACC	RANK	PTOW	PTOB
ZEROR (DEF. CLASS)	32.5	2.96	11.2	-3.6
IB5 (CRIT.1)	45.0	2.35	13.1	-1.7
J4.8 (CRIT.1)	50.0	2.21	13.0	-1.8
JRIP (CRIT.1)	43.8	2.26	13.2	-1.6
N.BAYES (CRIT.1)	36.3	2.72	11.9	-2.9
SMO-R (CRIT.1)	27.5	2.94	10.8	-4.0
SMO (CRIT.1)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.1)	77.5	1.42	14.6	-0.2
IB5 (CRIT.2)	48.8	2.23	12.9	-1.9
J4.8 (CRIT.2)	43.8	2.54	12.0	-2.8
JRIP (CRIT.2)	46.3	2.40	12.3	-2.5
N.BAYES (CRIT.2)	30.0	2.80	11.7	-3.1
SMO-R (CRIT.2)	26.3	3.07	10.7	-4.1
SMO (CRIT.2)	32.5	2.96	11.2	-3.6
SORTED BEST (CRIT.2)	86.3	1.19	14.6	-0.3

The accuracy of the meta-learning solutions range from 26.3% to 50%, which suggests that none of the meta-classifiers generated over the various meta-attribute sets are able to sufficiently learn how to classify the candidate algorithm with the highest accuracy. However, in comparison to the baseline accuracy afforded by the default class (i.e. ZeroR accuracy of 32.5%), we find that the IB5, J4.8, and JRip meta-learners consistently perform better. The performance of the Naive Bayes meta-learner on the other hand, is close to that achieved by the default class, while the SVMs perform quite poorly. One possible reason for this is that the amount of meta-knowledge regarding the patterns of performance of the candidate algorithms is insufficient for any meta-learner to satisfactorily learn to distinguish the most accurate candidate algorithm.

Essentially, the meta-learner must attempt to decode the tangle of partially approximated performance patterns (i.e. potentially noisy accuracy measurements) and then predict which is likely to be superior. This problem seems to be too complex for the meta-learner given only the accuracy measurements of the candidate algorithms over 80 UCI datasets. To clarify this point, we also evaluate the meta-learners using the actual stratified ten-fold cross-validation results as meta-attributes. These results, which are described in Table 6, suggest that even when the *true* accuracy scores of the candidate algorithms are provided, there is still insufficient data (i.e. UCI datasets) for the meta-learners to learn how to choose the highest score from among the accuracies input (i.e. to learn to perform an *argmax*).

Table 6. The meta-learning task results based on meta-attributes corresponding to the stratified 10-fold cross-validation accuracies.

META-LEARNER	ACC	RANK	PTOW	PTOB
ZEROR (DEF. CLASS)	32.5	2.96	11.3	-1.9
IB5	43.8	2.27	12.9	-1.9
J4.8	45.0	2.51	12.4	-2.4
JRIP	46.3	2.54	12.1	-2.7
N.BAYES	38.8	2.62	12.0	-2.6
SMO-R	32.5	2.96	11.3	-1.9
SMO	27.5	2.89	11.1	-1.8
SORTED BEST*	100.0	1.00	14.8	0.0

\* This corresponds to the procedure to compute the target meta-classes.

In comparison, the accuracy achieved by simply sorting the generated meta-attributes improves more significantly as the number of evaluated accuracy measurements increases, and eventually outperforms the default class and other meta-learners. This may be because of the bias behind by the sorting operation is directly representative of the (*argmax*) task, and thus only requires induction over the accuracy score approximations. This also means that the as the number of accuracy measurements are evaluated, the required induction is correspondingly lessened. Consider the following. Given  $|\mathcal{A}|$  candidate algorithms, and assuming that each  $a_i \in \mathcal{A}$  has an equal chance of being the most accurate on a given dataset  $s_j$ , there is thus a 1 in  $|\mathcal{A}|$  chance of selecting the most accurate algorithm from among  $\mathcal{A}$  for  $s_j$ . If  $k$  algorithms are evaluated (and thus the accuracy of  $|\mathcal{A}| - k$  remain unknown), then the chance of selecting the most accurate algorithm falls to 1 in  $(|\mathcal{A}| - k + 1)$  – i.e. we know the most accurate algorithm over the  $k$  that are run, but not any that were not are actually even more accurate). Essentially, when evaluating all but one candidate algorithm, there is a 1 in 2 chance of picking the most accurate one (i.e. from between the best of those evaluated, and the one that was not). However, the meta-learning solutions cannot take advantage of this, and the difficulty of the induction task that is faced (i.e. to determine the most accurate candidate algorithm) persists despite this potential discount.

## 5. Conclusion and Future Work

In this paper we present a new meta-learning framework for algorithm selection utilising regression-based landmarks. In essence, we seek to solve the algorithm selection task of identifying the more accurate algorithm from a given set of candidate algorithms by: 1) generating meta attributes that correspond to the performance patterns either directly evaluated or estimated via regression-based landmarks; 2) attempting to (meta-) learn the mapping between these meta-attributes and

meta-classes corresponding to the most accurate algorithm in the set. From our experiments using 80 UCI datasets and 6 WEKA algorithms, we discover that for the meta-knowledge employed, learning to predict the most accurate algorithm given some new dataset is a task that is too complex, and simply sorting the outputs of the predicted accuracy measurements via the regression-based landmarks achieves more satisfactory results.

There are several possible avenues for future work, including:

- Developing theory regarding the difficulty of meta-learning tasks and which of these to solve given some finite amount of meta-knowledge.
- Generating and meta-learning with more universally representative datasets, or perhaps datasets that are attuned to the failings of specific learning algorithms.
- Experimenting with different meta-learning tasks. For example, learning how to classify the  ${}^{\lfloor \mathcal{A} \rfloor}C_2$  pairwise comparisons from among the given  $\mathcal{A}$  candidate algorithms.
- The generation of use of more accurate meta-class data. In particular, this corresponds to the use of more statistically sound methods of algorithm evaluation and comparison (e.g. using stratified 10x10-fold cross-validation, paired t-tests, McNemar tests, etc).
- Experimentation with other meta-attributes. Other types of landmarks (e.g. regression-based relational landmarks) and dataset characteristics.
- Considering more complicated performance indicators (e.g. F-score, or other cost/utility functions) to either landmark or use as meta-classes.
- The development of landmarker theory.

Landmarking remains a new and relatively unexplored facet of meta-learning for algorithm selection, and should be further explored.

## Acknowledgments

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

## References

- Aha, D. (1992). Generalizing from case studies: a case study. *Proceedings of the 9th International Conference on Machine Learning*, (pp. 1-10).
- Blake, C., and Merz, C. (1998). *UCI repository of machine learning databases*. University of California,

- Irvine, Department of Information and Computer Sciences.
- Bensusan, H. (1998). God doesn't always shave with Occam's Razor: learning when and how to prune. *Proceedings of the 9th European Conference on Machine Learning*, (pp. 119-124).
- Brazdil, P., Soares, C., & Costa, J. (2003). Ranking learning algorithms: using IBL and meta-learning on accuracy and time results. *Machine Learning*, 50(3), 251-277.
- Fürnkranz, J., & Petrak, J. (2001). An evaluation of landmarking variants. *Proceedings of the 10th European Conference on Machine Learning, Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-learning*, (pp. 57-68).
- Gama, J., & Brazdil, P. (1995). Characterization of classification algorithms. *Proceedings of the 7th Portuguese Conference in AI*, (pp. 83-102).
- Giraud-Carrier, C., Vilalta, R., & Brazdil, P. (2004). Introduction to the special issue on meta-learning. *Machine Learning*, 54(3), 187-193.
- Kalousis, A., & Hilario, M. (2001). Model selection via meta-learning: a comparative study. *International Journal on Artificial Intelligence Tools*, 10(4), 525-554.
- Ler, D., Koprinska, I., and Chawla, S. (2005). A hill-climbing landmarker generation algorithm based on efficiency and correlativity criteria. *Proceedings of the 18th International FLAIRS Conference, Machine Learning Track*, (in press).
- Ler, D., Koprinska, I., & Chawla, S. (2004a). A landmarker selection algorithm based on correlation and efficiency criteria. *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence*, (pp. 296-306).
- Ler, D., Koprinska, I., & Chawla, S. (2004b). Comparisons between heuristics based on correlativity and efficiency for landmarker generation. *Proceedings of the 4th International Conference on Hybrid Intelligent Systems*, (pp. 32-37).
- Lindner, G., & Studer, R. (1999). AST: support for algorithm selection with a CBR approach. *Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases*, (pp. 418-423).
- Michie, D., Spiegelhalter, D., & Taylor, C. (1994). *Machine learning, neural and statistical classification*. Ellis Horwood.
- Pfahring, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. *Proceedings of the 17th International Conference on Machine Learning*, (pp. 743-750).
- Schaffer, C. (1993). Technical note: selecting a classification method by cross-validation. *Machine Learning*, 13(1), 135-143.
- Todorovski, L., Blockeel, H., & Dzeroski, S. (2002). Ranking with predictive clustering trees. *Proceedings of the 13th European Conference on Machine Learning*, (pp. 444-455).
- Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. *Journal of Artificial Intelligence Review*, 18(2), 77-95.
- Witten, I., & Frank, E. (2000). *Data mining: practical machine learning tools with Java implementations*. Morgan Kaufmann.