Meta-Learning by Landmarking Various Learning Algorithms

Bernhard Pfahringer BERNHARD@CS.WAIKATO.AC.NZ
Department of Computer Science, University of Waikato, Hamilton, New Zealand

Hilan Bensusan HILANB@CS.BRIS.AC.UK
Christophe Giraud-Carrier CGC@CS.BRIS.AC.UK
Department of Computer Science, University of Bristol, Bristol BS8 1UB, UK

Abstract

Landmarking is a novel approach to describing tasks in meta-learning. Previous approaches to meta-learning mostly considered only statistics-inspired measures of the data as a source for the definition of meta-attributes. Contrary to such approaches, landmarking tries to determine the location of a specific learning problem in the space of all learning problems by directly measuring the performance of some simple and efficient learning algorithms themselves. In the experiments reported we show how such a use of landmark values can help to distinguish between areas of the learning space favouring different learners. Experiments, both with artificial and real-world databases, show that landmarking selects, with moderate but reasonable level of success, the best performing of a set of learning algorithms.

1. Introduction

After the seminal work describing the so-called *no free lunch* theorems (Wolpert & Macready, 1995; Schaffer, 1994) it is now generally accepted that no single learning algorithm can dominate some other algorithm over all possible learning problems. Still, in daily practise we experience that some algorithm or some class of algorithms seems to perform much better than some other algorithm on a *specific* problem or a class of problems sharing some kind of properties. For instance, it is common practise to compare algorithms over subsets of the UCI data sets (Blake & Merz, 1998), as it is commonly believed that these data sets share some mythical properties making them *real-world* data sets. So the basic wish is that these data sets share some unspecified properties with any newly given *real-world* data set and that therefore any learning algorithm which performs well on the UCI sets will also perform well on these new sets.

A somewhat more sophisticated approach is to describe different classes of problems and search for correlations between these different classes and the different *optimal* learners. This is what happens - sometimes unconsciously - in the head of an ML or KDD practitioner. Faced with a new application, they usually perform some preliminary exploration and will quickly come up with some reasonable small set of potentially promising learning algorithms. As this kind of expertise develops with experience, it is only natural to try to construct either a knowledge base manually - the expert system's approach - or to apply some learning algorithm to acquire such rules - the meta-learning approach.

Successful meta-learning would be of great practical importance, as in any real-world application the brute-force approach of simply comparing all applicable algorithms is just not possible. Due to resource limitations - mostly limited computing time, but also limited space - only a handful of algorithms can ever be applied to a specific problem. Furthermore, usually several different representations are reasonable for a given application, which adds a further dimension to this already vast search space for the optimal algorithm. So successful meta-learning could guide exploration by supplying priorities for the allocation of limited resources, hopefully in a sound decision-theoretical way.

As to guide such a selection or ranking of learning algorithms, several approaches to meta-learning have been proposed (Bensusan, 1999; Chan & Stolfo, 1996; Giraud-Carrier & Hilario, 1998; Giraud-Carrier & Pfahringer, 1999; Lindner & Studer, 1999; Widmer, 1997). Perhaps, the most popular strategy consists of describing learning tasks in terms of a set of (meta-)attributes and classifying them according to the performance of one or more learners (e.g., which, of a set of candidate learners, gives the best performance on the data set). As is the case with standard machine learning, the performance of meta-learning is greatly dependent upon the description of tasks. A number of strategies have been proposed to describe learning tasks, including ones based on statistics, information theoretic considerations and properties of induced decision trees (Brodley, 1995; Bensusan, 1998; Bensusan, 1999; Michie et al., 1994; Lindner & Studer, 1999). There is however, to this day, no consensus on what strategy is most suitable for meta-learning.

In this paper we want to put forward the notion of *landmarking*. All the above described attempts are kind of indirect approaches in characterising domains. *Landmarking* tries to directly characterise a domain by relating the performance of some learners - the *landmarkers* - to the performance of some other algorithm. Essentially, we are looking for advice along the lines of "if landmarker A outperforms landmarker B on a given task, then learner X will also outperform learner Y on this task".

The next section will further develop the idea of landmarking. In section 3 we report on some exploratory studies we have conducted. Finally, section 4 summarises the paper and points to future work.

2. Theory of Landmarking

The main idea of landmarking - using learners themselves to characterise a learning problem - has already been mentioned by the Statlog project (Michie et al., 1994), but has not been explored any further in that project. The idea can be seen as arising from the experience with ML applications. After an initial exploration using statistical tools and visualization methods usually a few very efficient learning algorithms are applied. Afterwards an iterative process starts which, building on all the facts and measurements gathered from the previous phases, tries to employ an ever more sophisticated set of learners and appropriate parameter settings.

Landmarking focusses on the step which replaces the first set of simple learners by a set of more sophisticated ones guided by the results of the simple learners. In landmarking we try to generalize over a set of related domains to come up with some general rules for selecting promising learners. *Prima facie*, there are two essential questions to be answered for landmarking: 1) Is it possible at all? and 2) What are useful landmarkers?

The doubts concerning the possibility of landmarking are entangled with the question of whether meta-learning is feasible. Firstly, it can be pointed out that the possible space of learning problems - the examples of the meta-learning task - is vast. Secondly, the number of such meta-examples is necessarily small, simply because of the resource consumption of preparing a single meta-example: we have to estimate the performance of *all* learners we are interested in, usually by some time-consuming cross-validation process. Still, as we shall see in the next section, there is some hope for the feasibility of the approach.

Now let us turn to the second question, which landmarkers should we choose. One of the negative results of the StatLog Esprit project concerned the use of very sophisticated statistical measurements. The problems encountered with these was their computational complexity. Some of these tests have a complexity of the order of $O(n^3)$, where n is the number of examples. There are two major difficulties when using such a test. First of all it scales badly for large data-bases, and secondly, the same amount of CPU time could have been alloted to a sophisticated learner of equal or even better complexity already. Thus we probably should limit ourselves to a bound of $O(n\ logn)$.

A further consideration for landmarkers is their respective biases. If we hope to be able to successfully chart a territory of learning problems, we have to make sure that different landmarkers measure different properties, at least implicitly. Using several small variants of the same kind of landmarker would not yield a lot of useful additional information, and might also give the meta-learner more of an opportunity of fitting spurious, meaningless patterns.

3. Practise of Landmarking

In this section we will describe some exploratory case studies which intended to demonstrate that meta-learning based on landmarkers is possible at all - countering the doubts expressed above. Additionally, these studies should also provide some insight into useful choices for landmarkers.

One problem we have not yet discussed is the question of how we declare one algorithm the winner over another for some problem. The following studies will always describe what criterion they have chosen. But let us shortly discuss some interesting problem encountered here: basing the decision on some statistical test and some certain p-value will define a three-class meta-learning problem: algorithm 1 wins significantly, algorithm 2 wins significantly, or there is no significant difference between the two algorithms, i.e. they are *tied*.

This case of ties is interesting from a practical point of view: if the expected performance is the same, then it does not matter, which algorithm is chosen.

3.1 Artificial Rule Lists and Sets

To perform the experiments of this subsection we have implemented a simple-minded generator for artificial data based on decision lists. Additionally, we used rule sets where the class of each example was determined by voting all applicable rules. The set of landmarkers consisted of a linear discriminant learner, a naive bayes learner and the C5.0 tree learner.

The learners whose relative performance to each other should be predicted, consisted of boosted C5.0 trees, the rule learner Ripper (Cohen, 1995), the discriminant tree learner Ltree (Gama, 1999) and a nearest-neighbor learner. As it most closely matches the bias of the way the artificial data was generated, we would expect Ripper to perform best for at least the decision *list* type of data. But before looking at the results, let us first describe the inputs.

3.1.1 Landmarkers

This specific choice of landmarkers was made rather pragmatically. Of all learners available to us these were the ones which more or less fulfilled the criteria defined above: their computational complexity is within reasonable bounds (for the decision tree we have to assume reasonably balanced trees of $O(nlogn)$ size) and their biases are reasonably different (which might be called into question for the pair of naive bayes and linear discriminant).

Landmark values themselves were determined by tenfold stratified cross-validation. Whereas for naive bayes and linear discriminant the resubstitution error only slightly differed from the cross-validation estimate, the difference was more pronounced for C5.0, which more easily overfits a given data set. As we found the cross-validation estimates more reliable than the simple resubstitution error, we only report these results. From an application's point of view, landmark values computed by cross-validation are of course more *expensive* in terms of invested computation time. Resubstitution error would be *cheaper* and practical for both naive bayes and linear discriminant, and maybe even for C5.0 when coupled with rather strong pruning. This will be a line for further investigation.

In addition to these three numerical attributes the meta-data comprised the following: the total number of attributes, the number of symbolic attributes, the number of numeric attributes, the number of classes, and the frequency of the most common class. This default accuracy could also be interpreted as the most simple landmark. Meta-learners which are not numerically-challenged might be able to make use of this information relating it to the values of the other *true* landmarkers.

3.1.2 Meta-Learners

The *test-battery* of learning algorithms commonly employed by the Metal Esprit project was also utilized in these experiments. Therefore we have run the following learners on the meta-data: C5.0 trees, rules and boosted C5.0 trees, Ripper, Ltree, linear discriminant, naive bayes, and nearest neighbor. Each learning task comprised a two-class problem: for each pair of the algorithms from the set {Ripper, boosted C5.0 trees, nearest neighbor, Ltree}, predict which one will win. Wins are defined in an absolutely simple-minded manner here: the winner achieves a smaller predictive error which was estimated by cross-validation. Statistical significance was ignored here.

3.1.3 Data Generation

The data sets used were generated in the following again simple-minded manner. Quite a few parameters influence the generation of a data set and its associated decision rule list. We have chosen the settings given in Table 1.

Table 1. Experimental parameter settings.

Parameter	Values
num of examples	$[1000, 5000, 10000]$
num of attributes	always 20
num of numeric attributes	$[0, 10, 20]$
num of irrelevant attributes	$[0, 5, 10]$
max num of rule-conditions	always 3
num of different classes	$[2, 5]$
noise level	$[0.0, 0.05, 0.1, 0.2]$

Altogether there are 216 different parameter value combinations. For each such combination one data set was generated in the following way. Attributes values are generated randomly according to a uniform distribution: symbolic values are drawn to be true or false with a probability $p = 0.5$ and numeric values are drawn from the interval $[0, 1)$ uniformly. Decision rule lists are computed dynamically: either a new example is matched by some already present rule, then this rule's classification is used; or a new rule is dynamically created such that it matches the new example and the rule is appended to the end of the decision rule list. The predicted classes are assigned to newly created rules in a round-robin fashion. Finally the predicted class of some example is switched to a different class with a probability equal to the chosen *noise level*. The rationale for this kind of generator was the desire to have some guarantee on the learnability of the generated data set, as most concepts are not learnable at all according to algorithmic complexity theory (Li & Vitanyi, 1993).

Additionally, the decision rule lists generated along with the data were also used in a different way: if they are interpreted as sets instead of ordered lists, voting can be used to determine classification as well. So the same structure can define two different concepts depending on the interpretation. It was hoped that this second class of voted concepts would not as obviously favour Ripper as the best learner.

3.1.4 Results

Let us look at the results for the 216 original artificial problems first. Table 2 depicts the error-rates of each meta-learner when predicting pair-wise winners for all combinations of C5.0boost (abbreviated as C), Ripper (R), and Ltree (L). Generally we can notice that for this kind of domain Ripper seems to be quite a good meta-learner. This could be attributed to its strong over-fitting avoidance, which might be appropriate for such small data sets. We have not shown any pair-wise comparisons involving the nearest neighbor learner, as it performed worse than the other three learners on every single of the 216 data sets. Due to this unexpected result we will have to scrutinize our nearest neighbor algorithm more closely in the near future, as this finding probably indicates some deficiency or bug in this specific implementation.

Table 2. Meta-learner error-rates for pairwise predictions over decision-list problems.

Meta-learner	C-R	L-R	L-C
Default class	0.500	0.194	0.199
c5.0boost	0.231	0.181	0.157
c5.0rules	0.264	0.162	0.199
c5.0tree	0.259	0.167	0.185
lindiscr	0.278	0.194	0.199
ltree	0.264	0.222	0.199
mlcib1	0.310	0.236	0.227
mlcnb	0.260	0.204	0.222
ripper	0.208	0.153	0.139

Another advantage of Ripper as a meta-learner is the format of its output. The induced decision lists are rather terse and therefore easy to read and interpret. Here are the three rule lists induced using *all* 216 examples for the three pairwise comparisons:

```
C5.0BOOST versus RIPPER:
 c5.0boost :- c5<=0.09 (72/26).
 c5.0boost :- classes>=5, ex<=1000 (20/5).
 default ripper (77/16).

LTREE versus RIPPER:
 ltree :- lind<=0.0652 (7/3).
 ltree :- num>=10, classes>=5,
          maxclass>=0.547 (15/10).
 default ripper (161/20).

LTREE versus C5.0BOOST:
 ltree :- maxclass>=0.557, c5>=0.1978,
          lind<=0.28 (15/0).
 default c5.0boost (173/28).
```

First of all we notice that the landmark values are really being used: C5.0boost is predicted to win over Ripper in case the C5-estimate is already smaller or equal to 9%, or if both the number of classes is five or more and then the number of examples does not exceed 1000. Ltree is predicted to win over Ripper, if the linear discriminant error estimate is already small, or if both the number of numerical attributes and of different classes is large, but the distribution of the classes is rather skewed (as pointed out by the minimal `maxclass`-value of 0.547). Ltree is predicted to win over C5.0boost only in those rare cases that enjoy a similar skewed class distribution, and where the error estimate for C5 proper is rather high, and at the same time the linear discriminant based estimate is less than 28%. All these findings nicely parallel our own intuitions and experiences with these three learners in general.

If we look at the results for the *voted* rule sets reported in Table 3, there are two observations to make. Firstly, looking at the base level learners, C5.0boost now beats Ripper on about two third of all data sets, and also Ltree is approaching Ripper, beating it on about 40% of all data sets. At the meta-level we see that only for the pair of Ltree and Ripper we are able to predict the winner considerably better than the default would. One fact that might partly explain these different results is that the voted rule set problems are simply harder to learn than the decision-list problems: e.g. C5.0boost achieved an average error rate of 11.3% over the 216 decision-list problems compared to an average error of 18.12% for the same number of voted rule set problems.

Now the really interesting question is what is going to happen when we fuse the meta-data of the previous

Table 3. Meta-learner error-rates for pairwise predictions over voted rule set problems.

Meta-learner	C-R	L-R	L-C
Default class	0.260	0.372	0.240
c5.0boost	0.240	0.235	0.168
c5.0rules	0.220	0.245	0.200
c5.0tree	0.224	0.270	0.200
lindiscr	0.260	0.260	0.219
ltree	0.265	0.306	0.168
mlcib1	0.296	0.260	0.224
mlcnb	0.270	0.250	0.252
ripper	0.240	0.255	0.240

two experiments. Will the meta-learners still be able to predict at least some of the winners, i.e., will they be able to implicitly distinguish between the two problem classes? Table 4 answers this question, and the answer is clearly positive for at least some of the pairings. So we are quite confident that landmarkers enable meta-learning.

Table 4. Meta-learner error-rates for pairwise predictions over the union of both problem domains.

Meta-learner	C-R	L-R	L-C
Default class	0.386	0.279	0.218
c5.0boost	0.297	0.214	0.158
c5.0rules	0.311	0.248	0.170
c5.0tree	0.316	0.243	0.177
lindiscr	0.354	0.291	0.218
ltree	0.345	0.262	0.180
mlcib1	0.386	0.282	0.214
mlcnb	0.316	0.267	0.192
ripper	0.303	0.255	0.150

3.2 Selecting Learning Models

In another set of experiments, we attempted to investigate the potential of landmarking to decide whether a learning algorithm or a learning model, involving more than one learning algorithm, is better than the others being considered. The idea was to use meta-learning to determine whether the task should be assigned to a particular learning model. Landmarkers were tested for their capacity to establish if the task falls close to the area of expertise of a learning model. If the task belongs to this area, the learner most equipped to tackle it should be in the vicinity.

In these experiments, the landmarkers' performance values for describing the tasks are calculated by re-substitution error over ten randomly drawn training sets. Here, 222 artificial Boolean data sets, together with 18 Uci ones (Blake & Merz, 1998) were used. The Boolean data sets had from 5 to 12 attributes and were classified by simple parity, DNF and CNF rules as well as at random. The 18 Uci data sets were: mushrooms, abalone, crx, sat, acetylation, titanic, waveform, yeast, car, chess(king-rook-vs-king), led7, led24, tic-tac-toe, monk1, monk2, monk3, satimage, quisclas. The performance of every learner in each data set is calculated by ten-fold stratified cross-validation.

3.2.1 Landmarkers

In contrast with the previous experiments, in the experiments of this section we used *only* the landmarkers' performance values to describe the task. Four landmark learners were used, all of which were learners producing single node classifiers, i.e. decision trees of minimal node complexity. These learners are efficient and often reveal what bias is required for the task. All of them attempt to establish in one way or another whether relations between the attributes that cannot easily be captured by linear separation are present in the task.

The first landmarker is a *decision node* learner. Here, a single decision node is chosen according to C5.0's information gain-ratio (Quinlan, 1993). The node is then used to classify test examples. This landmark learner aims to establish how much the task is amenable to linear separability since the performance of the decision node learner tells us whether some progress is made by dividing the examples according to the most informative attribute.

The second landmarker is a *randomly chosen node* learner. A randomly chosen attribute is used to split the training set and classify the test examples. This landmark learner, together with the next one, aims to inform about irrelevant attributes.

The third landmarker is a *worst node* learner where the gain-ratio information criterion is used to pick up the least informative attribute to make the single split. This landmarker, together with the first one, is supposed to tell us something else about linear separability: if neither the best nor the worst attribute produce a single well performing separation, it is likely that linear separation is not an adequate learning strategy.

The last landmarker is an *elite 1-nearest neighbor* learner that acts as follows. It computes 1-Nearest Neighbor, where the test set is classified based on the classification of the closest training example on a subset of all attributes. This elite subset is composed by the most informative attributes if the gain-ratio difference between them is smaller than 0.1[1]. Otherwise, the elite subset is a singleton and the learner acts like a decision node learner. This landmark learner intends to establish whether the task is a relational one, that is, if it involves parity-like relationships between the attributes (Clark & Thornton, 1997). In relational tasks, no single attribute is considerably more informative than others.

3.2.2 Meta-Learners

In some of the experiments of this section, the 10 learners chosen to be part of the Metal project were used, that is: C5.0 trees, rules and boosted C5.0 trees, Ripper, Ltree, linear discriminant, naive bayes, and nearest neighbor. In other experiments, C4.5 was used as a meta-learner trained with artificial Boolean data and tested with real, Uci data.

3.2.3 Results

The task was to classify problems as either problems for a specific learning model or problems where none of the learners being considered is significantly better than any other.

The meta-learners considered for the experiments were exactly the same 10 Metal learners. The problems were classified therefore either as suitable for a learning model or equally adequate for all 10 learners. A problem is said to be better suitable for a given learning strategy if the performance of this strategy is at least 10% better than the average performance of all 10 learners. We considered three specific learning algorithms, nearest neighbor, naive Bayes and C5.0 with boosting, and three learning models, neural network learning, rules learning, decision tree learning and boosted decision tree learning. A problem is considered to be suitable for neural networks if either backpropagation on a multi-layered perceptron or radial-basis function network learning perform better than average. It is suitable for rules learning if either Ripper or C5.0rules is better than average. It is suitable for decision tree learning if Ltree, C5.0 or C5.0 with boosting is better than average.

In the first experiments, we used a ten-fold stratified cross-validation on all the 240 data sets (222 Boolean and 18 Uci data sets). Error rates are reported in Tables 5 and 6.

Table 5. Meta-learner error rates for predicting nearest neighbor (kNN), naive bayes (NB), and boosted C5.0 (C5B) suitabilities.

Meta-Learner	kNN	NB	C5B
Default class	0.420	0.380	0.510
c5.0boost	0.283	0.304	0.492
c5.0rules	0.250	0.258	0.475
c5.0tree	0.254	0.263	0.488
clemMLP	0.313	0.358	0.458
clemRBFN	0.304	0.288	0.429
lindiscr	0.304	0.371	0.421
ltree	0.246	0.275	0.413
mlcib1	0.388	0.308	0.392
mlcnb	0.342	0.329	0.513
ripper	0.292	0.233	0.417

Table 6. Meta-learner error rates for predicting neural network (NN), rule (R), and decision tree (DT) suitabilities.

Meta-Learner	NN	R	DT
default class	0.440	0.370	0.470
c5.0boost	0.438	0.229	0.379
c5.0rules	0.367	0.229	0.371
c5.0tree	0.358	0.233	0.371
clemMLP	0.413	0.392	0.454
clemRBFN	0.333	0.225	0.375
lindiscr	0.371	0.379	0.467
ltree	0.396	0.221	0.346
mlcib1	0.388	0.258	0.354
mlcnb	0.433	0.421	0.421
ripper	0.363	0.221	0.363

Results show that most meta-learners produce error levels smaller than the default error class and often the difference is substantial. The error figures show that landmarking successfully meta-learns but they do not reflect the overall performance of a system whose end result is the accuracy of the selected learning model. In further experiments, we tried to estimate this by using the 222 Boolean problems as training examples and the 18 Uci problems for testing. The results reported for these experiments are the average error difference between the best choice and the selected choice in the 18 Uci problems. If the average is in fact better than the chosen model, we consider the error difference between the chosen model and the average. We proceed similarly if the meta-learner had chosen against the model that in fact is better than the average of the 10 learners. Here we used only C4.5 as meta-learner. Average error differences are given in Table 7, which compares the average error loss between the best option available and that picked by landmarking, and the maximum error loss. The latter would be the maximum error difference if the worst learner were always chosen. The results indicate that landmarking meta-

[1] This threshold is based on previous work by the second author (Bensusan, 1999).

learns with a considerable level of overall accuracy.

Table 7. Error loss of landmarking and maximum error loss in the average of the 18 Uci data sets

Learner	loss	Maximum loss
nearest neighbor	0.036	0.067
naive Bayes	0.016	0.063
c5.0boost	0.044	0.065
neural networks	0.031	0.081
rules	0.036	0.088
decision trees	0.021	0.096

3.3 Landmarking vs. the Traditional Approach

The following experiment was designed to compare landmarking with the traditional information-theoretical meta-characterisation of tasks for meta-learning. We considered six of the commonly used meta-attributes defined in the literature (Michie et al., 1994; Engels & Theusinger, 1998): class entropy, average entropy of the attributes, mutual information, joint entropy, equivalent number of attributes and signal-to-noise ratio. In this experiment, the goal was to select one of the 10 learners of the Metal project. We used 320 artificially generated binary classification tasks with 5 to 12 Boolean attributes. We used the same landmarkers as in section 3.2. Table 8 shows the error rates obtained by 10 meta-learners using landmarking, the traditional approach or both.

Table 8. Error rates of the landmarking approach (land), the information-based approach (dct) and combination of both (both).

Meta-Learner	land	dct	both
default class	0.460	0.460	0.460
c5.0boost	0.248	0.360	0.295
C5.0rules	0.239	0.333	0.301
C5.0tree	0.242	0.342	0.314
clemMLP	0.301	0.317	0.320
clemRBFN	0.289	0.323	0.304
lindiscr	0.335	0.311	0.301
ltree	0.270	0.317	0.286
mlcib1	0.329	0.366	0.342
mlcnb	0.429	0.407	0.363
ripper	0.292	0.314	0.295

Notice that landmarking outperforms the information-based approach for most of the 10 meta-learners. Furthermore, it seems that adding the information-based features to landmarking impairs landmarking performance. Although further experiments are required, this comparison suggests that landmarking offers a promising approach to meta-learning.

4. Conclusions

Landmarking can only be successful when the landmarkers are able to measure properties relevant to the success or failure of some learning algorithm for some specific problem. Thus the choice of learners to distinguish between will also influence the selection of landmarkers. Meta-learning in general might assist the further development and improvement of learning algorithms. Pinpointing weaknesses of certain learning algorithm relative to their competitors can show directions for this algorithm's further improvement. Other further work will address, amongst others, the following points:

- Generation of artificial data: for large-scale meta-learning experiments the generation of artificial data is tantamount. There are simply not enough real-world data sets at our disposal. Furthermore, artificial data allows for careful variation of properties which are deemed essential. We will have to acquire a better understanding of what and how data can be generated as to be both learnable and relevant for real-world problems. Then the scarce real-world problems could be set aside for use as truly independent test examples in evaluating meta-learning.

- Feature engineering for meta-learning: the choice of our landmarkers was rather ad-hoc. Such choices must be scrutinized more thoroughly, as must be the combination of landmark values with other meta-attributes that are based on statistical or information-theoretic grounds.

- Suitable meta-learners: The choice of meta-learners used in this paper was ad-hoc as well. One possibly fruitful direction would be the use of some first-order learning algorithm at the meta-level (Todorovski & Dzeroski, 1999). That would enable a more detailed meta-level description of a learning task including not only summary statistics, but also information about specific single attributes. Another direction would investigate learners of a more characterising nature, being able to indicate their area of expertise and consequently being able to abstain for examples outside these areas.

- Quantification of the gain due to meta-learning: the cost of meta-learning itself can be neglected if we assume that this cost will be spread-out over a number of applications and therefore amortize itself nicely. Even the measurement cost of landmarking could be neglected if the time complexity of the land-marking algorithms is considerably smaller than the complexity of the predicted algorithms. For such an idealized setting predictive accuracy would suffice for discriminating between competing meta-learners. More general settings will demand the development of more sophisticated measures trading off meta-level accuracy and meta-level time complexity.

In summary, we have empirically shown landmarking to be of potential value for meta-learning. Naturally, the initial studies reported in this paper have but slightly scratched the surface of a huge body of promising research work.

Acknowledgements

This work is supported by an ESPRIT Framework IV LTR Grant (Nr 26.357). We thank the members of the project consortium for useful comments and discussion. This research was conducted while the first author was affiliated with the Austrian Research Institute for Artificial Intelligence.

References

Bensusan, H. (1998). God doesn't always shave with Occam's Razor – learning when and how to prune. *Proceedings of the Tenth European Conference on Machine Learning* (pp. 119–124).

Bensusan, H. (1999). *Automatic bias learning: an inquiry into the inductive basis of induction.* Doctoral dissertation, School of Cognitive and Computing Sciences, University of Sussex.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.

Brodley, C. E. (1995). Recursive automatic bias selection for classifier construction. *Machine Learning*, *20*, 63–94.

Chan, P., & Stolfo, S. (1996). On the accuracy of meta-learning for scalable data mining. *Journal of Intelligent Information Systems*, *8*, 3–28.

Clark, A., & Thornton, C. (1997). Trading spaces: computation, representation and the limits of uninformed learning. *Behaviour and Brain Sciences*, *20*, 57–90.

Cohen, W. (1995). Fast effective rule induction. *Proceedings of the Twelfth International Machine Learning Conference* (pp. 115–123).

Engels, R., & Theusinger, C. (1998). Using a data metric for offering preprocessing advice in data mining applications. *Proceedings of the Thirteenth European Conference on Artificial Intelligence* (pp. 430–434).

Gama, J. (1999). Discriminant trees. *Proceedings of the Sixteenth International Machine Learning Conference* (pp. 134–142).

Giraud-Carrier, C., & Hilario, M. (Eds.). (1998). *ECML'98 Workshop Notes - Upgrading Learning to the Meta-Level: Model Selection and Data Transformation.* Technical Report CSR-98-02, Fakultät für Informatik, Technische Universität Chemnitz.

Giraud-Carrier, C., & Pfahringer, B. (Eds.). (1999). *Proceedings of the ICML'99 Workshop on Recent Advances in Meta-Learning and Future Work.* Ljubljana, Jozef Stefan Institute.

Li, M., & Vitanyi, P. (1993). *An introduction to kolmogorov complexity and its applications.* New York, Springer.

Lindner, G., & Studer, R. (1999). AST: Support for algorithm selection with a CBR approach. *Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 418–423).

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (Eds.). (1994). *Machine learning, neural and statistical classification.* Ellis Horwood.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann.

Schaffer, C. (1994). A conservation law for generalization performance. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 259–265).

Todorovski, L., & Dzeroski, S. (1999). Experiments in meta-level learning with ILP. *Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases* (pp. 98–106).

Widmer, G. (1997). Tracking context changes through meta-learning. *Machine Learning*, *27*, 259–286.

Wolpert, D., & Macready, W. (1995). *No free lunch theorems for search* (Technical Report SFI-TR-95-02-010). Santa Fe Institute.