Predicting Relative Performance of Classifiers from Samples

Rui Leite RLEITE@LIACC.UP.PT
Pavel Brazdil PBRAZDIL@LIACC.UP.PT
LIACC-NIAAD/FEP, University of Porto, Rua de Ceuta, 118-6^o, 4050-190 Porto - Portugal

Abstract

This paper is concerned with the problem of predicting relative performance of classification algorithms. It focusses on methods that use results on small samples and discusses the shortcomings of previous approaches. A new variant is proposed that exploits, as some previous approaches, meta-learning. The method requires that experiments be conducted on few samples. The information gathered is used to identify the *nearest learning curve* for which the sampling procedure was carried out fully. This in turn permits to generate a prediction regards the relative performance of algorithms. Experimental evaluation shows that the method competes well with previous approaches and provides quite good and practical solution to this problem.

1. Introduction

The problem of predicting relative performance of algorithms continues to be an issue that is worth investigating further. There are many algorithms that can in principle be used on any given problem. The user can make a direct comparison of the considered algorithms on any given problem using a cross-validation evaluation scheme. However the computational costs of this approach are significant. If it is desirable to avoid running every algorithm.

It is thus useful to have a principled way that would help us to determine which algorithms are likely to lead to the best results on a new problem. The common thread of many previous methods is to store previous experimental results on different datasets. The datasets are characterized using a set of measures, including the dataset in question for which we seek an advice. A (meta-)learning method is used to generate a prediction (i.e relative ordering of algorithms) for the new case.

Some methods rely on dataset characteristics in the form of statistical and information-theoretic measures (D. Michie, 1994; Brazdil et al., 2003). These measures need to be identified beforehand, which is a nontrivial task. Even if we can come up with a set of supposedly good *candidate measures*, it is not guaranteed that these will be useful in the end. Typically we need to verify whether these are useful in the task of predicting the relative performance of algorithms.

These difficulties have led some researchers to explore alternative ways to achieve the same aim. Some have used simplified versions of some of the algorithms referred to as *landmarks* (Bensussan & Giraud-Carrier, 2000; Pfahringer et al., 2000). The results of these landmark algorithms are then used as *measures* to estimate the relative performance of algorithms. Although the initial results were promissing, the method has not been extensively used afterwards.

Other researchers have proposed to use simplified versions of the data, which are sometimes referred to as *sampling landmarks*. The performance of algorithms on samples can be used again to estimate their relative performance. However, the methods that exploited information from sampling landmarks were on the whole inconclusive. The results did not show a clear advantage of using this kind of information (Fürnkranz & Petrak, 2001; Soares et al., 2001). Somewhat surprisingly, using more information in the form of more samples did not lead to marked improvement.

This somewhat startling finding motivated us to investigate this issue further, which in turn enabled us to come up with a new solution. The method described here exploits information about learning curves which has already proved to be useful in other contexts (Leite & Brazdil, 2004). The method requires that experiments be conducted on few samples for the algorithms

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005. Copyright 2005 by the author(s)/owner(s).

in question. The information gathered is used to identify the *nearest learning curves*, for which the sampling procedure was carried out fully. This in turn permits to generate the prediction regards the relative performance of the algorithms. Experimental evaluation shows that the method competes well with previous approaches and provides quite good and practical solution to this problem.

The rest of the paper is organized as follows. The next section provides more details regards the sampling method and how we can predict the outcome of learning. Section 3 discusses the experiments and the results obtained. The final section describes our conclusions.

2. Using Sampling to Predict the Outcome of Learning

As we have mentioned earlier, the method described relies on performance estimates obtained on samples. The method proceeds in two phases. In the first one, we generate a model by training the given algorithm on a small sample of the data. After this, another small sample (from the same data) is used to carry out tests and to obtain the required estimates of performance.

Different sampling strategies have been described in literature. Simple strategies use a fixed number or fixed fraction of cases. Some more elaborate strategies try to find an optimum sample size using a succession of models generated by a given learning algorithm on the basis of a sequence of samples. Some authors have used samples that increase by a fixed amount (John & Langley, 1996), while others have used progressively increasing samples (Provost et al., 1999; Leite & Brazdil, 2004). Usually the aim is to determine the sample size in which the accuracy does not increase any more, called a *optimal sample size.* Fig. 1 shows a typical learning curve and the optimal sample size is represented by S^*. The corresponding accuracy is represented by a^*.

Our aim is to predict the accuracy a_i at a particular sample size S_i, on the basis of known measurements, corresponding to an initial segment consisting of #S samples. Let us examine again the learning curve represented in Fig. 1. Suppose the points a_1, a_2 and a_3 constitute the initial segment. So, our aim is to estimate what happens afterwards.

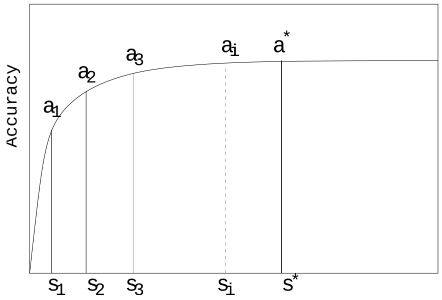

Figure 1. Learning Curve

The prediction of a_i is done on the basis of previous knowledge about the algorithm in question. The knowledge is stored in the form of learning curves obtained earlier on other (similar) datasets. The aim is to use these curves to make predictions of accuracy on a curve that is only partly known (we have information about the initial segment only).

The details of this method are described in the following. First, we will discuss how the learning curves are represented. Then, we will show how certain learning curves are identified on the basis of existing information for the purpose of prediction. Finally, we show how the prediction is generated and how this can determine the relative ordering of pairs of algorithms [1]. The reader can consult Fig. 2 for an overview of the method.

```
Input:
   $A_p$, $A_q$ (2 algorithms)
   $d$ (dataset in question), $L$ (database of $n$ learning curves)
Parameters:
   $\#S$ (size of the initial segment)
   $k$ (number of neighbors)
   $S_i$ (size of dataset $d$ expressed as sample size)
Output:
    Decision (1 or 0), regards $A_p > A_q$.

Run the algorithms $A_p$ and $A_q$ on dataset $d$
   while varying samples from $m = 1$ to $\#S$.
Calculate accuracies $A_{p,d,m}$ and $A_{q,d,m}$
    (construct partial learning curves)
Analyze the learning curves stored and
   calculate distances $d_{p,q}(d, j)(j = 1..n)$
Identify k curves for algorithms $A_p$ and $A_q$
    with the smallest distance.
Retrieve the corresponding accuracies of $A_p$ for size $S_i$,
    $a_{p,j,n1}...a_{p,j,nk}$, and combine them.
    Repeat this also for $A_q$.
Return value 1 if the combined accuracy (mean) of $A_p$
   is higher than the combined accuracy of $A_q$.
   Otherwise return 0.
```

Figure 2. The basic algorithm for predicting relative performance

2.1. Representation of Learning Curves

Suppose we have datasets $\{D_1, D_2, ..., D_n\}$ and for each one we have a learning curve available (later we

[1] This goal is similar to the one used by others (Kalousis & Theoharis, 1999), but the method used here is different.

will discuss a variant of this basic method which uses N learning curves per dataset). Each learning curve is represented by a vector $< a_{i,1}, a_{i,2}, .., a_{i,z} >$, where $a_{i,m}$ represents the accuracy of the given algorithm on dataset D_i on m-th sample in the sequence. Following previous work (Provost et al., 1999) the sizes follow a geometric progression. The sequence spans across the whole dataset.

2.2. Identification of Appropriate Learning Curves for the Purpose of Prediction

Suppose we are interested in dataset D and we have information about the initial segment of the learning curve (e.g. the first #S=3 points). We employ a nearest neighbor algorithm (k-NN) to identify similar datasets and retrieve the appropriate learning curves. Here the k-NN algorithm represents a meta-learner that helps us to resolve the issue of predicting the accuracy for a particular sample. As k-NN uses a distance measure to identify k similar cases, we need to adapt the method to our problem.

Here we just use the information concerning the initial segment. The distance function between datasets D_i and D_j takes into acount the results of both algorithms considered (A_p and A_q) and is defined by

$$d_{p,q}(i,j) = \sum_{m=1}^{\#S} (a_{p,i,m} - a_{p,j,m})^2 + \sum_{m=1}^{\#S} (a_{q,i,m} - a_{q,j,m})^2 \tag{1}$$

where m spans across the initial segment. In other words, the method takes into acount a given pair of algorithms and tries to identify cases (i.e. datasets) which are most similar to the current one.

2.3. Predicting Accuracy and Determining which Algorithm is Better

Once k learning curves have been identified, we generate the prediction regards the accuracy for a given sample size S_i [2]. That is, if the dataset in question can be described, say, using 12 samples, we would try to predict the accuracy of the 12-th sample(S_{12}).

If we use k>1 curves, then, in general, the retrieved values will differ. One obvious way to estimate the accuracy a_i on the basis of this information is by using the *mean* value.

Our task is to use this information to resolve the following decision problem. Suppose we have 2 algorithms, A_p and A_q, then our aim is to determine which one of the two is likely to give better results. The decision is easy to make, as we can just compare the predicted accuracies (the means) and select the one that is better.

Some alternatives as to how the problem can be formulated are discussed further on.

2.4. Using Aggregated Learning Curves

It is a well known fact that the performance of many algorithms may vary substantially when different portions of data are selected from a given source. This phenomenon is usually referred to as *variance* (Breiman, 1996). The problem is even more apparent if we use small samples. As a consequence, the learning curves do not always look like the one shown in Fig. 1 which is monotonically increasing. The curves obtained from real data often include points that appear to jump up and down. This has an adverse affect on the method described.

To minimize this problem we have decided to generate a smoothed-out curve on the basis of N learning curves per dataset. Each individual learning curve is obtained using a different portion of the data, using a method similar to N cross-validation. Each point $A_{i,m}$, the m-th point of smoothed curve for dataset i, represents the mean of the corresponding points of the individual learning curves.

2.5. Adaptation of the Retrieved Curves

As was described earlier, the retrieved learning curve is used to generate a prediction. We note that even if we retrieved the nearest curve to the one given, in general, the accuracies will not coincide. It may happen that all the accuracies will be above (below) the given curve. So if we use this curve directly for prediction, we can expect that the accuracy a_i for sample S_i will be off the target.

We note that even if we used a larger segment in the matching process, we may retrieve exactly the same curve, without being able to affect the final prediction. In other words, the predicted accuracy will be biased by form of the retrieved learning curves. This fact explains the somewhat surprising finding that if we use more information (i.e. larger segment in matching), this may not always lead to improvements. This could be one of the reasons for the somewhat discouraging results reported in literature (Fürnkranz & Petrak, 2001; Soares et al., 2001).

We are interested to correct this shortcoming. If we

[2] Apart from this we use also an additional mechanism of *adaptation* described in a separate section later.

provide a system with more information it should work better! We have adopted the following strategy to overcome (or to mitigate) this problem.

The strategy exploits the notion of *adaptation* as in Case-based Reasoning (Kolodner, 1993; Leake, 1996). The main idea behind this is not only to retrieve a partial solution (i.e. a curve), but to *adapt* it to new circumstances. One straightforward way of doing this is by *moving* each retrieved curve to the partial curve available for the new dataset (see Figure 3). This adaptation can be seen as a way of combining the information of the retrieved curve with the information available for the new dataset.

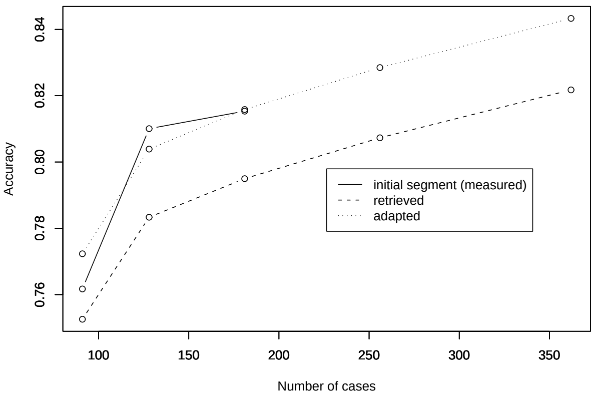

Figure 3. Adaptation of the retrieved learning curve

We have designed a simple adaptation procedure which modified the retrieved curve using a *scale coefficient* (named f).

Supose that the new dataset is D_i and its partial learning curve $c_i :< a_{i,1}, a_{i,2}, ..., a_{i,\#S} >$. Supose also that $c_r :< a_{r,1}, a_{r,2}, ..., a_{r,m} >$ is one of the retrieved curves. We adapt the retrieved curve c_r to c_i by multiplying each point of c_r by f. The adapted version of c_r will be given by $c'_r : a'_{r,j} = f \times a_{r,j}$.

The scale coeffcient f is the one that minimizes the euclidean distance between the two curves on the initial segment $< 1, ..., \#S >$. Besides we consider that each point has a different weight. The idea is to give more importance to points occuring later on the learning curve. The weigths increases as the sample sizes increases. An obvious way to express this idea is to define the weight as the sample size associated to the considered point ($w_j = \#S_j$).

The following equation determines f:

$$f = \frac{\sum_{j=1}^{\#S} \left(a_{i,j} \times a_{r,j} \times w_j^2 \right)}{\sum_{j=1}^{\#S} \left(a_{r,j}^2 \times w_j^2 \right)} \qquad (2)$$

The inclusion of the adaptation in our algorithm is straightforward. After retrieving the curves we adapt them to the partial learning curves.

3. Empirical Evaluation

As we have explained earlier in this paper, our aim is to devise a method for a simple decision problem: Suppose we have 2 algorithms, which one will give better accuracy on a given dataset?

In our experimental study we have used the following two algorithm, C5 (Quinlan, 1998) and SVM [3]. We could have chosen others. Which algorithm we use is not really so important, as long as one competes well with the other. This condition was satisfied with the datasets used (the default accuracy is discussed further on). Both algorithms were used with default settings for similar reasons to the ones mentioned. Our aim was not to achieve the highest possible accuracy, but predict which of the *two given algorithms* will be better.

This decision problem is represented as classification task as follows: If C5 is better than SVM (shortly C5>SVM), then we say the class is 1, while in the opposite case the class is 0. To get the true classification we have used the usual cross-validation evaluation procedure on each dataset for the two given algorithms.

Our first aim is to determine the accuracy of our approach. That is, can the samples be used to obtain high accuracy on our classification task?

Besides, our other aim was to compare these results to a previous method which relies on dataset characteristics instead. So, we have elaborated a variant of the method that would enable us to evaluate this. Instead of using results on samples when searching for nearest learning curves, we would use the seven characteristics used earlier (Brazdil et al., 2003).

The first approach is identified here as MDS (meta-learning on data samples) and the second one as MDC (meta-learning with data characteristics).

We have used 30 datasets in the evaluation. Some come from UCI (Blake & Merz, 1998), others were

[3] We have used the libsvm (Chang & Lin, 2001) implementation provided by the e1071 (Dimitriadou et al., 2004) package of R (R Development Core Team, 2004). We have used a radial basis kernel.

used within project METAL (MetaL, 1999).

The samples were generated using a geometric progression as follows. The size of m_i-th sample is set to the rounded value of $2^{6+0.5\times m_i}$. Thus the size of the first sample is $2^{6.5}$, giving 91 after rounding, and the second sample is 2^7, giving 128 etc. Table 1 shows the relationship between the sample number and the actual sample size.

Table 1. Relationship between the sample number and the actual sample size

m	1	2	3	...	20	25
size	91	128	181	...	65536	370728

In this experiment the number of samples used is 1. We have just used the first sample (91 cases). The results are shown in Table 2. As we can see, the initial problem has a default accuracy of 53%. Our approach (MDS) achieves accuracy of 77% and outperforms the one that uses dataset characteristics (MDC).

Besides, we have observed that our approach is much faster to execute than the alternative one (MDC). It needs to examine only a part of the data to obtain the estimate. The process is fast, despite the fact that we need to train a classifier in each case. This is because training on small samples is relatively fast. Characterisation of the entire dataset is much slower, particularly if the dataset is large.

3.1. The Effect of Curve Adaptation on Relative Performance Prediction

In this section we describe an experiment whose aim was to evaluate the adaptation procedure described earlier. Here it is referred to as A_MDS, while the method without adaptation is called MDS. The difference between A_MDS and MDS relies in the adaptation of the retrieved learning curves to the partial learning curves.

Our aim here was to determine which of the versions (A_MDS or MDS) has better performance. Besides we also compare the computational costs of MDS (which has a very similar cost to A_MDS) to a cross-validation approach to our decision problem. For each dataset we consider the time cost of MDS as the time spent on training all the classifiers needed to obtain the two partial learning curves (one for A_p and other for A_q).

The experimental setup was the same employed in the experiment described previously. The only difference is that we have varied the size of the initial segment to test the adaptation procedure.

Table 2. Classification results for C5 > SVM

dataset	MDS	MDC	true class	win/loss
acetylation	0	0	0	
adult (METAL)	1	0	1	+
contraceptive	0	0	0	
musk	0	0	1	
parity	0	1	1	−
quisclas	0	0	0	
recljan2jun97	1	1	1	
adult (UCI)	1	0	1	+
allbp	1	1	1	
allhyper	1	1	1	
ann	1	1	1	
car	1	0	0	−
cmc	0	0	0	
krkopt	0	0	0	
mfeat	0	0	0	
nursery	1	0	1	+
optdigits	0	0	0	
pendigits	1	0	0	−
pyrimidines	1	0	1	+
quadrupeds	1	0	0	−
sat	1	0	0	−
segmentation	1	0	1	+
shuttle	1	0	1	+
sick	1	1	1	
spambase	1	1	0	
splice	0	0	0	
thyroid0387	1	1	1	
waveform21	0	1	0	+
waveform40	0	1	0	+
yeast	0	0	0	
Correctly Classified	23	20	–	
Accuracy	77% (23/30)	67% (20/30)	–	
Default Accuracy	–	–	53% (16/30)	

The results concerning the accuracies of A_MDS and MDS are shown in Figure 4.

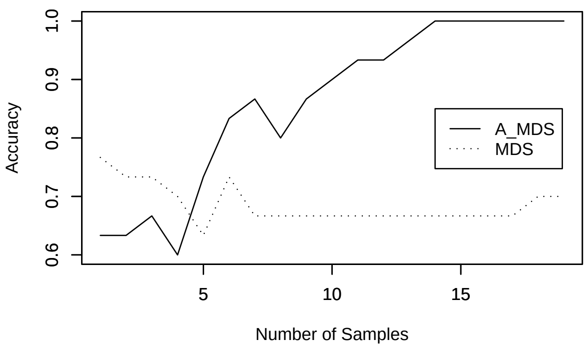

Figure 4. The effect of curve adaptation on the accuracy

We can see that for MDS the results do not improve if we use more samples. In fact, we observe a drop in the accuracy as the initial segment sizes increases. In contrast, the accuracy of A_MDS as we use more samples. If we use all the data it reaches almost 100% accuracy. In this sence it is equivalent to a cross-validation evaluation scheme.

We observe that A_MDS outperforms MDS if the initial segment sizes is greater than 5.

This means that if we use only few samples the adaptation procedure harms the performance.

This result is in our view of interest to others. Adaptation should be used if the dataset is relatively large and so constructing an initial segment with more than 5 points is justified.

As for the time computational costs the results are presented in Figure 5.

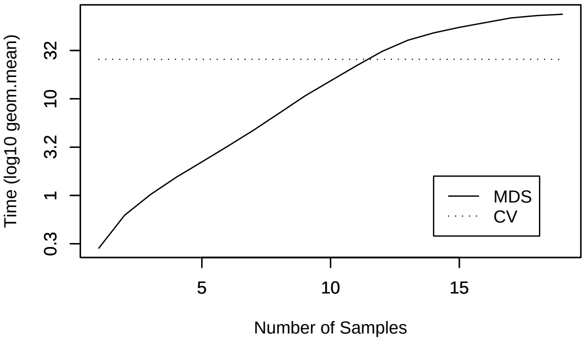

Figure 5. Time comparison between MDS and cross-validation

The time costs are presented in the y axis using a log_{10} scale. The results for the 30 datasets are summarized using a geometric mean [4].

The dotted line, which remains constant as the initial segment sizes increases, represents the average time needed to run a cross 10 fold cross-validation on each dataset. The value for this average is 25.7 seconds.

The solid line represents the time costs for A_MDS. Both lines cross when the number of samples is between 11 and 12. This means that the time of A_MDS is the same as in the cross-validation approach.

When the number of samples is between 5 and 6 A_MDS is about 10 times faster than cross-validation. A number of samples equal to 7 corresponds to a point when A_MDS is 5.4 times faster than cross-validation and reaches still quite good accuracy (86.6%).

4. Parameters of Our Method: The Values Used and Future Work

The method described involves various parameters. In this section we briefly justify why certain choices were made and discuss other options that could be taken.

Choice of Algorithms and Datasets: In this study we have used a pair of algorithms (C5 and SVM) and 30 datasets. Further work could be carried out to verify that the results hold in other settings.

Using significance tests: The decision problem regards whether we should use algorithms A_p or A_q could be formulated as 3-class problem as other authors have sugested (Kalousis & Theoharis, 1999). Class 1 (-1) would be attributed to cases when Ap is *significantly better (worse)* than A_q. All other cases would be classified as 0. We plan to verify whether the main result holds also for this scenario.

Representation of the Learning Curve: Each learning curve is represented by a sequence of points. The sample sizes follow a geometric progression. Both the initial size (91 cases) and the increment represent parameters of the method are considered fixed. Other settings could be tried in future, although we do not think the results could be improved dramatically this way. Besides, instead of saving point-to-point information about learning curves, one could take a model-based approach. In principle it would be possible to fit a predefined type of curve through the points and save the curve parameters. The distance measure could then be redefined accordingly. As the curve fitting is subject to errors, it remains to be seen whether this approach would lead to better results.

Number of Curves Constructed per Dataset: We have used both a single curve and N=10 curves per dataset. As has been pointed out earlier, the N=10 curves were compacted into a single aggregated smoothed-out curve. The results with this curve were much better than the results with a single curve. Further work could be done to determine the advantages / disadvantages of using other values than 10.

Size of the initial segment of the learning curve used in matching: It appears that the segment consisting of only one sample provides already quite a lot of information for a good decision. We have experimented with other values and should quantify what the net benefit is of using a larger segment.

[4] We have chosen the geometric mean because it correspond to arithmetic mean when using a log scale.

Using data characteristics: In the work on predicting the stopping point on a learning curve, one particular dataset characteristic - the dataset size (i.e. number of cases) - was shown to be useful (Leite & Brazdil, 2004). Addition of this attribute led to better performance. We plan to examine whether this could also improve the method described here and whether some other characteristics could be exploited (e.g. number or entropy of classes etc.).

Value of k in the k-NN procedure: In our experiments we have used the value k=3. We have experimented with other different values (both lower and higher than 3), but the results were on the whole comparable. These results could be validated further by conducting further experiments.

5. Discussion

In this section discusses some related work which has not been covered in the earlier sections of this paper.

The issue of accuracy prediction has been addressed for by others (Bensussan & Kaloussis, 2001). There are several differences between this work and the one presented here. The most important one is that the authors did not use sampling landmarks, but other methods, including for instance landmarks. Although landmarking enabled to construct regression models with rather low MAD error, it failed to provide a good ranking of classifiers. As we have demonstrated, the method proposed here does not suffer from this shortcoming.

6. Conclusions

We have described a method that exploits the information about learning curves to determine which of two learning algorithm is likely to be better on a new problem.

The method requires that experiments be conducted on few samples. The information gathered is used to identify the *nearest learning curves*, for which the sampling procedure was carried out fully. This in turn permits to generate the prediction regards the relative performance of the given algorithms.

We have carried out experimental evaluation of the method using 30 datasets. Our approach (MDS) achieves accuracy of about 77% and outperforms the method that uses dataset characteristics only (MDC).

Besides being much faster, it does not require to come up with data characteristics that are suitable for prediction. The information of the algorithms on data samples provides that information. The method is thus usable in other circumstances where we have limited knowledge about why some algorithms work, while others don't.

An interesting issue arises why previous attempts in this direction were inconclusive. Previous approaches did not exploit the information regards learning curves as we have done. This we believe is an important aspect that makes a difference.

Besides, we have re-used the idea of *case adaptation* to ovecome one fundamental problem which the previous methods did not manage to resolve. That is, how can we explain the fact if we use more samples, the performance does not really improve. If we use adaptation, this problem is resolved.

With a larger number of samples we get more accuracte predictions. When the number of samples is 7 whe obtain quite a good compromise between speed and accuracy. The method is about 5.4 times faster than cross-validation, but still achieves quite good accuracy.

In conclusion, the approach presented provides quite good and practical solution to the problem of estimating which algorithm is better than another.

Acknowledgments

The authors gratefully acknowledge the financial support from FCT within pluri-annual funding (Financiamento Plurianual) attributed to the Portuguese R&D Units. Thanks also to Carlos Soares for providing us with the scripts for dataset characterization.

References

Bensussan, H., & Giraud-Carrier, C. (2000). Discovering task neighbourhoods through landmark learning performances. *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2000)* (pp. 325–330). Springer.

Bensussan, H., & Kaloussis, A. (2001). Estimating the predictive accuracy of a classifier. in machine learning. *Proceedings of the 12th European Conference on Machine Learning*. Springer.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/mlrepository.html.

Brazdil, P., Soares, C., & Costa, J. (2003). Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, *50*, 251–277.

Breiman, L. (1996). *Bias, variance, and arcing classifiers* (Technical Report 460). Statistics Department, University of California.

Chang, C.-C., & Lin, C.-J. (2001). Libsvm: a library for support vector machines (version 2.31).

D. Michie, D. e. C. (1994). *Machine learning, neural and statistical classification*. Ellis Horwood.

Dimitriadou, E., Hornik, K., Leisch, F., Meyer, D., & Weingessel, A. (2004). *e1071: Misc functions of the department of statistics (e1071), tu wien*. R package version 1.5-1.

Fürnkranz, J., & Petrak, J. (2001). An evaluation of landmarking variants. *Proceedings of the ECML/PKDD Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2001)* (pp. 57–68). Springer.

John, G., & Langley, P. (1996). Static versus dynamic sampling for data mining. *Proceedings of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*. AAAI Press.

Kalousis, A., & Theoharis, T. (1999). Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*, *3(50)*, 319–337.

Kolodner, J. (1993). *Case-based reasoning*. Morgan Kaufmann Publishers.

Leake, D. B. (1996). *Case-based reasoning: Experiences, lessons & future directions*. AAAI Press.

Leite, R., & Brazdil, P. (2004). Improving progressive sampling via meta-learning on learning curves. *Proceedings of the 15th European Conference on Machine Learning - ECML 2004* (pp. 250–261). Springer.

MetaL (1999). Metal project site. http://www.metal-kdd.org/.

Pfahringer, B., Bensusan, H., & Giraud-Carrier, C. (2000). Meta-learning by landmarking various learning algorithms. *Proceedings of the 17th International Conference on Machine Learning (ICML-2000)* (pp. 743–750). Stanford, CA.

Provost, F., Jensen, D., & Oates, T. (1999). Efficient progressive sampling. *Proceedings of Fifth Int. Conf. on Knowledge Discovery and Data Mining*. AAAI Press.

Quinlan, R. (1998). C5.0 : 'an informal tutorial'.

R Development Core Team (2004). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Soares, C., Petrak, J., & Brazdil, P. (2001). Sampling-based relative landmarks: Systematically test-driving algorithms before choosing. *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA 2001)* (pp. 88–94). Springer.