Estimating the Predictive Accuracy of a Classifier

Hilan Bensusan[1] and Alexandros Kalousis[2]

[1] Department of Computer Science, University of Bristol
The Merchant Venturers Building, Woodland Road
Bristol, BS8 1UB, England
hilanb@cs.bris.ac.uk
[2] CSD University of Geneva,
CH-1211 Geneva 4, Switzerland
kalousis@cui.unige.ch

Abstract. This paper investigates the use of meta-learning to estimate the predictive accuracy of a classifier. We present a scenario where meta-learning is seen as a regression task and consider its potential in connection with three strategies of dataset characterization. We show that it is possible to estimate classifier performance with a high degree of confidence and gain knowledge about the classifier through the regression models generated. We exploit the results of the models to predict the ranking of the inducers. We also show that the best strategy for performance estimation is not necessarily the best one for ranking generation.

1 Introduction

The practice of machine learning often involves the estimation of how well a classification learning algorithm would perform in a dataset. There is no classifier that can be predictively successful in every dataset. In addition, any device that selects the most appropriate classifier for a dataset based on its properties is bound to fail in some area of the dataset space [23]. However, there may exist a sub-area of the dataset space that is small enough for such a device to have high performance, compensated by bad performance elsewhere, and yet large enough to include all the datasets we are actually interested [22,17]. Machine learning is possible only if this sub-area exists. Within this sub-area it is possible to estimate the performance of different learners according to the nature of the dataset.

Currently, this estimation is done through the expertise of the machine learning practitioner. The practitioner, of course, brings her previous experience with classifiers as well as her preferences to the estimation process. As a consequence, the estimation is often vague, in many cases unprincipled and always relying on the human expert. Often, the practitioner can appeal to a well established technique in the field, cross-validation, to help the estimation or at least to establish which classifiers are likely to work best. Cross-validation is a priori justifiable because it works for a large enough area of the dataset space [22]. It is, however, too costly. Moreover, it provides no insight concerning the relations between

L. De Raedt and P. Flach (Eds.): ECML 2001, LNAI 2167, pp. 25–36, 2001.

the performance of a classifier and the properties of the domain. In this sense it provides no principled basis for an analysis of what lies behind successful performance estimations.

Meta-learning is the endeavour to learn something about the expected performance of a classifier from previous applications. In any scenario, it depends heavily on the way we choose to characterize the datasets. Meta-learning has often concentrated on predicting whether a classifier is suitable for a dataset [11] and on selection of the best option from a pool of classifiers[1,13]. In the former, we ask whether a good performance is to be expected from a classifier given the properties of a dataset. In the latter, given a pool of classifiers, we attempt to establish which ones are the best. In both cases meta-learning is constructed as a classification task. Little work has been reported on direct estimation of the performances of classifiers.

In this paper we propose and examine an approach to the direct estimation of classifiers' performances via meta-learning. We face meta-learning tasks as regression tasks whereby we look for relationships between the properties of a dataset and the performance of the classifier. This direct approach is more flexible than meta-learning for model selection since the estimations are not relative to a specific pool of classifiers.

2 How Can We Predict Accuracies

The idea of using regression to predict the performance of learning algorithms was first used by Gama and Brazdil in [7], were they continued on the framework adopted in STATLOG. They tried to directly predict the error of an algorithm for a specific dataset based on the characteristics of the dataset, as these were defined in the STATLOG project. For each of the learners they evaluated various regression models like linear regression, instance based regression and rule based regression. They report poor results in terms of the Normalised Mean Squared Error (NMSE).

Sohn in [19] uses the results of STATLOG (i.e. same data characterization, same learning algorithms and same datasets) and constructs linear regression models that predict the errors of the learning algorithms on unseen datasets. As she is using the results of STATLOG, the study is limited to 19 datasets To overcome the small number of datasets she used bootstraping resampling to estimate the parameters of the regression models. The models were used to provide a ranking of the available learning algorithms. The results show that the statistical models produced, exhibit high performance. However they must be interpreted cautiously because of the limited number of datasets used in the study.

A recent paper provided some initial results related to the use of estimated performances for model selection [10]. It shows that estimating performances leads to a better result in selecting a learner from a pool than learning through a repository of datasets classified in terms of the best performing algorithm in the pool. Using a pool composed by three classifiers, the paper indicates that regression (by M6), when used to estimate the error of the three classifiers, selects the classifier with least error with better performance than using classification (with C5.0) to decide the best algorithm for a dataset. The experiments, however, were preliminary and concentrated only on one strategy of dataset characterization, on only three classifiers and were performed on artificially generated datasets.

A work on a similar direction is that of instance-based ranking of classifiers through meta-learning, also called zooming[18]. The goal there is to determine a preference order over a pool of classifiers, based on predictive accuracy and computational cost. The ranking for a new dataset is built by inspecting a number of k-nearest neighbours in a collection of reference datasets, that form a meta-dataset. The produced ranking is based on a preference function that weights cost and accuracy. This approach cannot be used as it is to estimate accuracies of learners, but only to provide a relative ranking of them.

Our goal was to broaden this research by considering a greater number of classifiers and different strategies of dataset characterization. We therefore concentrate primarily on performance estimation. Our approach was to construct meta-datasets for regression so that any technique could then be used for performance estimation. A meta-dataset is constructed for each classifier. In order to do that, each dataset needs to be characterised by a dataset characterization stategy that produces meta-attributes. The meta-attributes produced are then the attributes for the meta-learning problem where each data-point corresponds to a dataset. Each data-point contains the description of the dataset by the meta-attributes and the performance of the classifier in the dataset. The meta-dataset can then be treated as an ordinary regression dataset.

In this paper we concentrate on 8 classifiers: two decision tree learning methods (Ltree [6] and C5.0tree [14]), Naive Bayes [4], two rule methods (Ripper [3] and c5.0rules [14]), linear discriminant, nearest neighbor [4] and a combination method, c5.0boost [15,5]. These classifiers are representative of different types of induction procedures and are among the most popular non-parametric classifiers in machine learning.

3 Strategies of Dataset Characterization

The characterization of datasets is the touchstone of meta-learning. Its success depends on how well can the meta-attributes support generalization. We aim at dataset characteristics that produce accurate estimates and insightful regression models. We will make use of three different strategies of dataset characterization:

- A set of information-theoretical and statistical features of the datasets that were developed as a sequel to the work done in STATLOG [11]. We refer to this strategy as DCT. We used the extended set of characteristics given in [20].
- A finer grained development of the STATLOG characteristics, where histograms were used to describe the distributions of features that are computed for each attribute of a dataset, thus preserving more information than

the initial mean based approach of STATLOG. We refer to this strategy as HISTO. A detailed description of the histograms can be found in [9].

- Landmarking, a characterization technique where the performance of simple, bare-bone learners in the dataset is used to characterise it [13]. In this paper we use seven landmarkers: *Decision node*, *Worst node*, *Randomly chosen node*, *Naive Bayes*, *1-Nearest Neighbour*, *Elite 1-Nearest Neighbour* and *Linear Discriminant*. We refer to this dataset characterisation strategy as LAND. Obviously, when we want to predict the error of an algorithm that is also a landmarker, this landmarker is omited from the description of the dataset.

Of the three approaches landmarking has a completely different philosophy since it is using the performance of simple learners to characterize datasets. DCT and HISTO are both based on a description of datasets in terms of their statistical and information based properties although using a different set of characteristics, with HISTO trying to exploit the full information contained in the distribution of these characteristics.

4 Regression on Accuracies

Regression was used to estimate the performance of classifiers using the different strategies of dataset characterization. Since the quality of the estimate depends on its closeness to the actual accuracy achieved by the classifier, the meta-learning performance is measured by the Mean Absolute Deviation (MAD). MAD is defined as the sum of the absolute differences between real and predicted values divided by the number of test items. It can be seen as measure of the distance between the actual values and the predicted ones.

In order to compare the estimation capabilities of the three strategies of dataset characterization we used Cubist [16] and a kernel method [21] to perform regression on the meta-dataset. Kernel methods work in an instance-based principle and they fit a linear regression model to a neighborhood around the selected instance. It is straightforward to alter their distance metric in order to make better use of the semantics of the non-applicable values that occur in meta-attributes of DCT and HISTO. The drawback of kernel methods is that they do not produce a model that can be used to improve our knowledge of the relationships between performance and dataset. Cubist, on the other side, produces models in the form of rulesets and therefore is more suitable for our analysis of the insight gained about a classifier by the process of estimating its accuracy. The disadvantage of Cubist, on the other hand, is that it can make no direct use of the non-applicable values found in the meta-features of DCT and HISTO. To be able to use the algorithm we had to recode those values to new ones that lie outside the domain of definition of the dataset characteristics we use. Since at least one of these limitations would apply to every existing regression system, we remedy the situation by applying both the kernel method and Cubist to all meta-datasets. We do that while bearing in mind that the kernel methods

Table 1. Kernel on estimating performance.

Classifier	dct	histo	land	dMAD
c50boost	0.112	0.123	0.050	0.134
c50rules	0.110	0.121	0.051	0.133
c50tree	0.110	0.123	0.054	0.137
lindiscr	0.118	0.129	0.063	0.137
ltree	0.105	0.113	0.041	0.132
mlcib1	0.120	0.138	0.081	0.153
mlcnb	0.121	0.143	0.064	0.146
ripper	0.113	0.128	0.056	0.145

are more likely to produce good results for dct and histo meta-datasets while Cubist is being run for the rulesets it produces.

Experiments were done with 65 datasets from the UCI repository [2] and from the METAL project [8]. For each classifier meta-dataset, we run a 10-fold cross-validation to assess the quality of performance estimations. The quality of the estimation is assessed by the MAD in the 10 folds, and it is compared with the default MAD (dMAD). The latter is the MAD obtained by predicting that the error of a classifier in a test dataset is the mean of the error obtained in the training datasets. dMAD is a benchmark for comparison. We expect regression to produce a smaller MAD than the dMAD. We have to note here that, in the case of landmarkers, when ever we build a model to predict the performance of a classifier that is a member of the set of landmarkers the corresponding landmarker is removed.

The quality of the estimation with the kernel method using different dataset characterization strategies is shown in table 1. The table presents the MAD in the 10 folds for every regression problem and the dMAD. Using the three dataset characterization strategies, the MAD obtained by the kernel method is smaller than the dMAD, showing that regression is worth trying. Landmarking outperforms the other two strategies by far and produces estimated accuracies with a MAD smaller than 0.081 for every classifier. This means that the average error of the estimated accuracy in unseen datasets will be in the worst case (that of mlcib1) 8.1%. The table shows that a great deal of meta-learning is taking place. Histo and dct do not produce estimates as good as the ones produced by landmarking. One could suspect that this is because the meta-dataset is relatively small when compared to the large number of meta-attributes used by these two strategies of dataset characterization. To check whether reducing the dimensionality of the problem would significantly improve the estimates, we performed feature selection through wrapping in the dct and the histo meta-dataset. The estimates, however, were not greatly improved. We conclude that landmarking performs best in performance estimation using kernel.

With Cubist, the situation is similar. Table 2 shows that land still performs better than dct and histo. The table also gives figures for land-, a strategy

Table 2. Cubist on estimating performance.

Classifier	dct	histo	land	land-	dMAD
c50boost	0.103	0.128	0.033	0.079	0.134
c50rules	0.121	0.126	0.036	0.077	0.133
c50tree	0.114	0.130	0.044	0.078	0.137
lindiscr	0.118	0.140	0.054	0.127	0.137
ltree	0.114	0.121	0.032	0.054	0.132
mlcib1	0.150	0.149	0.067	0.077	0.153
mlcnb	0.126	0.149	0.044	0.052	0.146
ripper	0.128	0.131	0.041	0.061	0.145

Table 3. P-values of paired T-tests of significance comparing with the dMAD (Kernel).

Classifier	dct	histo	land
c50boost	*tie* (0.112)	*tie* (0.361)	+ (0.00)
c50rules	*tie* (0.075	*tie* (0.319)	+ (0.00)
c50tree	+ (0.045)	*tie* (0.242)	+ (0.00)
lindiscr	*tie* (0.110)	*tie* (0.548)	+ (0.00)
ltree	+ (0.030)	*tie* (0.115)	+ (0.00)
mlcib1	+ (0.037)	*tie* (0.269)	+ (0.00)
mlcnb	+ (0.036)	*tie* (0.807)	+ (0.00)
ripper	+ (0.024)	*tie* (0.217)	+ (0.00)

of dataset description where only decision node, random node, worst node, elite nearest neighbour and linear discriminants are used as landmarkers. The reason for this is that with less landmarkers we obtain more insightful models. Also, the loss in MAD is not extreme and land- still performs well when compared to the dMAD.

To examine whether the results presented are significant we performed t-paired tests of significance. In Table 3 we give the results of the t-paired test between each model and the dMAD. We present results only for the kernel based models, but the situation is similar for the cubist based ones. In this table and in the following ones, "+" indicates that the method is significantly better than the default, *tie* signifies that there is no difference, "−" that the method is significantly worse then the default. The table shows that the performance of landmarkers is always significantly better then the default. dct is significantly better in 5 out of the 8 cases, and histo performance is not statistically different than the default. Furthermore, landmarking is always significantly better than dct and histo for all the eight different learning algorithms. Between dct and histo the differences are not significant for any of the 8 learners.

Concluding we can say that the use of landmarkers to perform accuracy estimation is a method with very good performance and low estimation error, significantly better from histo and dct. The reason for that is: landmark based

Table 4. Average Spearman's Correlation Coefficients with the True Ranking

	MODELS		
Rankings	Kernel	Cubist	Zooming
DEFAULT	0.330	0.330	0.330
DCT	0.435	0.083	0.341
HISTO	0.405	0.174	0.371
LAND	0.180	0.185	
LAND-		0.090	0.190

characteristics are better suited for that type of task. They provide a direct estimation of the hardness of the problem since they are themselves performance estimations. On the other side DCT and HISTO, give an indirect description of the hardness of the problem, through the use of characteristics like attributes correlations, which are more difficult to directly associate with accuracy.

5 Using Estimates to Rank Learners

An obvious way to use the accuracies predicted by regression is to provide a ranking of the learners based on these predictions. In this section we give results for various ways of predicting rankings. We validate their usefulness by comparing them with the true ranking, and the performance of a default ranking.

To evaluate the different approaches, the rankings produced for a dataset are compared to the true ranking of the learners on this dataset. The true ranking is known since we know the accuracies of all the learners on the 65 datasets that we are using. As a measure of similarity of the rankings, we used Spearman's rank correlation coefficient [12]. We also compare our method with zooming [18]. The results in terms of the average Spearman rank correlation coefficient are given in table 4. Zooming cannot be applied to the full set of landmarkers, since that will mean using the performance of lindiscr, mlcib and mlcnb to predict their ranking. This is why the corresponding combination, (zooming+land) is not included in the table. We also give the average Spearman's rank correlation coefficient of the *default ranking* with the true ranking. The default ranking is a ranking that remains the same no matter what the dataset under examination is. It is computed on the basis of the mean accuracies that the learners achieve over all the datasets. The default ranking, starting from the best learner, is : c50boost, c50rules, c50tree, ltree, ripper, mlcib1, mlcnb, lindiscr. A ranking method is interesting if it performs significantly better than this default ranking: in this case it is worth applying the meta-learning method to discover a suitable ranking for a given dataset. Table 5 gives the results of the statistical significance tests, between the different models and the default ranking. We concentrate on kernel since the rankings produced by cubist perform always worse than the default.

Surprisingly enough the only case that a model is significantly better than the default ranking is the combination of Kernel and DCT, even ranking with zoom-

Table 5. P-values of paired t-tests, between the rank correlation coefficients of the models and the rank correlation coefficient of the default ranking

Dataset	models	
Characterization	Kernel	Zooming
dct	+(0.05)	*tie*(0.862)
histo	*tie*(0.147)	*tie*(0.482)
land	−(0.010)	
land-		−(0.018)

ing a method specifically created to provide rankings is not significantly better then the default. Although landmarking constructs regression models that have a very low MAD error, it fails to provide a good ranking of the classifiers. The predictions provided by Kernel and dct, while worse than the ones provided by landmarking based models, systematically keep the relative order of the accuracies of the classifiers. So although they do not estimate the performances accurately, they do rank the classifiers well. A reason for the poor performance of landmarking in ranking is that landmarking based regression models give the error as a function of the error of simple learners. This can lead to models where the error of an inducer is proportional to the error of another inducer resulting in a more or less fixed ranking of the available inducers, a fact that explains the poor performance of landmarkers when it comes to ranking inducers.

6 What Regression Models Can Tell Us about Classifiers

An advantage of using regression for meta-learning is that we can extract useful knowledge about the classifiers whose performance we are trying to estimate. The generated models refer to a specific classifier. They associate characteristics of the datasets, classifier error. Examining them will give us an idea of what are the dataset characteristics that affect the performance of classifiers.

The main motivation behind the application of Cubist, was the construction of a set of rules in order to see how datasets characteristics associate with learners' performances, and more important, whether the associations that pop up make sense. Below we can see typical examples of rules constructed by Cubist to estimate the error of Naive Bayes. For each rule, we give the number of the cases (i.e. datasets) that are covered by the rule, and the mean error of Naive Bayes on the cases covered by the rule. Every rule consists of two parts, an *IF* and a *THEN* part. The *IF* part specifies the precondictions that must hold in order for the rule to apply, these preconditions are conditions on specific dataset characteristics which are determined by cubist. The *THEN* part of a rule is the regression model constructed by cubist to predict the error of Naive Bayes on the datasets covered by the preconditions of the rule. Below we are going to review the produced rules in terms of their agreement with expert knowledge.

```
*DCT rules:
a)7 cases, mean error 0.06          b)17 cases, mean error 0.10
  IF MAXAttr_Gini_sym > 0.245        IF MAXAttr_Gini_sym <= 0.245
     AVGAttr_gFunction > -0.91          AVGAttr_gFunction > -0.91
   THEN mlcnb_error = -0.0748        THEN mlcnb_error = -0.0965
     - 0.1528 Cancor                   - 0.252 AVGAttr_gFunction
     + 0.1479 Frac                     - 0.1706 Cancor
     - 0.1179 AVGAttr_gFunction        + 0.1613 Fract

* HISTO rules:
a)29 cases, mean error 0.22       b)11 cases, mean error 0.46
  IF conc_hist_4 <= 0.0032 AND      IF num_attributes > 7
     conc_hist_with_class_0 <= 0.68   con_histo_4 <= 0.0032
   THEN mlcnb error = 0.2074          conc_histo_class_0 > 0.68
     - 0.25 correl_hist_4           THEN mlcnb_error = 0.4235
     + 0.21 correl_hist_5           - 0.25 correl_hist_4
     + 0.04 correl_hist_2           + 0.21 correl_hist_5
     + 0.02 conc_hist_class_0       + 0.04 correl_hist_2
     - 0.01 conc_hist_class_1       + 0.026 conc_histo_class_0
     - 0.01 con_histo_4             - 0.021 con_histo_4
                                    - 0.016 conc_histo_class_1
* LAND rules:
a)34 cases, mean error 0.218     b)16 cases, mean error 0.385
  IF Rand_Node <= 0.57           IF Rand_Node > 0.57
     Elite_Node > 0.084           THEN mlcnb = 0.339
   THEN mlcnb = 0.167             - 1.099 Worst_Node
     + 0.239 Rand_Node            + 0.792 Dec_Node
     - 0.18 Worst_Node            + 0.292 Rand_Node
     + 0.105 Elite_Node           + 0.105 Elite_Node
```

If we examine the two rules of DCT, we see that they define two sets of datasets based on the MAXAttr_Gini_Sym. The first one containing seven datasets, with the mean error of Naive Bayes on these to be 6%, and the second set containing seventeen datasets, with a mean error of 10%. MAXAttr_Gini_Sym is the maximum gini index, that we compute between one attribute and the class variable. If this characteristic has a high value, this means that there exists one attribute on the dataset that has high information content for the class attribute. The higher the value of this characteristic is, the easier the prediction of the class. By examining the two rules we can see that on the datasets with a smaller value of MAXAttr_Gini_Sym, Naive Bayes has higher mean error. Examining now the regression models constructed for the two rules we see that they are quite similar, they have the same set of dataset characteristics and the same signs on coefficients, with just a slight difference on the magnitute of the coefficients. Cancor is the first canonical correlation between the attributes and the class. As this increases we expect the error to decrease and this is indeed the case, since on the regression models it has a negative sign. Frac is the proportion of total class variation explained by the first canonical discriminant. It seems that this feature has a negative effect on the error of Naive Bayes, since the corresponding regression coefficient has a positive sign. The gFunction is a measurement for the probability of finding the joint distribution of classes and attribute values for a specific attribute. AVGAttr_gFunction is the average overall the attributes, the higher the value of the characteristic the easier the problem it is, and indeed the error of Naive Bayes decreases when AVGAttr_gFunction increases.

In HISTO, datasets separated according to the value of con_histo_with_class_0. This characteristic gives the percentage of attributes that have a concentration coefficient with the class attribute, between 0 and 0.1, (the concentration coefficient is a measure of association between two discrete characteristics, with 0 for no association at all, and 1 for the highest association). A high value of it implies low association between the attributes and the class attribute, thus one would expect a high error, for the datasets that exhibit that property. The two rules comply with this expectation. Datasets covered by the first rule (i.e. datasets were less than 68% of the attributes have a concentration coefficient with the class variable smaller than 0.1) have lower error, 0.22, than the datasets covered by the complementary rule, 0.46.

In the case of landmarking, we used the smaller set of landmarkers to generate the regression model with Cubist (that is, LAND-). In the two rules given above using landmarking, two disjoint sets of datasets are established based on the value of the Rand_Node. Rand_Node is the error of a single decision node created from a randomly chosen attribute. If this node exhibits a high error, chances are that the attributes have poor discriminating power. This situation is indeed captured by the rules. As we may see the datasets covered from the first rule (lower error of Rand_Node), exhibit a lower mean error from the datasets covered from the second rule.

7 Computational Cost

Setting up the different regression models, is a task that is performed only once. After that the produced models can be used on new unseen cases. The main cost of the method comes on the exploitation phase, from the characterization of a new dataset. DCT and HISTO are information and statistical based approaches. Their main cost is on the construction of the contigency tables and the covariance matrixes of the datasets. These have a complexity of $O(n)$, where n is the number of examples of a dataset. Some of the characteristics used require the computation of eigen values of the covariance matrixes, this has a complexity of $O(p^3)$, where p is the number of attributes, so the computational complexity of DCT and HISTO is $O(n+p^3)$ which is much smaller of that of cross validation. In the case of landmarking the computational complexity is that of the cross validation of the landmarkers. The most expensive landmarker used here is the 1-nearest neighbor, whose complexity is $O(n^2)$. We can reduce this complexity, obtaining similar results, if we omit the nearest neighbor landmarker. In this case the complexity is determined by linear discriminants that have a complexity of $O(n+p^3)$.

8 Conclusions

In this paper we invistigated the use of regression learners in order to directly estimate the errors of classification inducers on specific datasets through the use of three different ways of characterizing the datasets. Landmarking provided by far the best predictions among the three different approaches. Using these predictions we were able to provide a ranking of the inducers, with the results being acceptable, on the limit, in only one of the examined cases.

Two are the main prerequisites for meta-learning to be effective, first a good characterization of the datasets and second the morphological similarity of new datasets to the ones used to construct the meta-models. This is not different from the fact that in order, for learned models to be useful, the examples on which they are applied should come from the same distribution as the training examples. The ideal environment for successfull utilisation of the approach is one where the analyst normally faces datasets of similar nature.

This is still an initial study and there is lot of work that has to be done, especially in the area of datasets characterization, possibly with the use of new characteristics, or with the combination of the existing ones to a single characterization.

Acknowledgements. This work is part of the METAL project supported by an ESPRIT Framework IV LTR Grant (Nr 26.357). Thanks to Carlos Soares for providing with the results on zooming.

Appendix Datasets Used

abalone, acetylation, agaricus-lepiota, allbp, allhyper, allhypo, allrep, australian, balance-scale, bands, breast-cancer-wisc,breast-cancer-wisc_nominal, bupa, car, contraceptive, crx, dermatology, dis, ecoli, flag_language, flag_religion, flare_c, flare_c_er, flare_m, flare_m_er, flare_x, flare_x_er, fluid, german, glass, glass2, heart, hepatitis, hypothyroid, ionosphere, iris, kp, led24, led7, lymphography, monk1, monk2, monk3-full, mushrooms, new-thyroid, parity5_5, pima-indians-diabetes, proc-cleveland-2, proc-cleveland-4, proc-hungarian-2, proc - hungarian-4, proc-switzerland-2, proc-switzerland-4, quisclas, sick-euthyroid, soybean-large, tic-tac-toe, titanic, tumor-LOI, vote, vowel, waveform40, wdbc, wpbc, yeast.

References

1. H. Bensusan. God doesn't always shave with Occam's Razor – learning when and how to prune. In *Proceedings of the 10th European Conference on Machine Learning*, pages 119–124, 1998.
2. C. Blake and C. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/.
3. W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

4. R. O. Duda and P. E. Hart. *Pattern classification and scene analysis.* John Wiley, 1973.
5. Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings 13th International Conference on Machine Learning*, pages 148–146, 1996.
6. J. Gama. Discriminant trees. In *Proceedings of the 16th International Machine Learning Conference (ICML'99)*, pages 134–142, 1999.
7. J. Gama and P. Brazdil. Characterization of classification algorithms. In *Proceedings of the 7th Portugese Conference in AI, EPIA 95*, pages 83–102, 1995.
8. C. Giraud-Carrier et al. A meta-learning assistant for providing user support in data mining and machine learning, 1999-2001. http://www.metal-kdd.org.
9. A. Kalousis and T. Theoharis. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intelligent Data Analysis*, 3(5):319–337, 1999.
10. C. Köpf, C. Taylor, and J. Keller. Meta-analysis: from data characterisation for meta-learning to meta-regression. In P. Brazdil and A. Jorge, editors, *PKDD'2000 Workshop on Data Mining, Decision Support, Meta-Learning and ILP*, 2000.
11. D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, 1994.
12. H. Neave and Worthington P. *Distribution Free Tests.* Unwin Hyman, London, UK, 1992.
13. B. Pfahringer, H. Bensusan, and C. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 743–750, 2000.
14. J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
15. J. R. Quinlan. Bagging, boosting, and C4. 5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pages 725–730, 1996.
16. R. Quinlan. An overview of cubist. Rulequest Research, November 2000. http://www.rulequest.com/.
17. R.B. Rao, D. Gordon, and W. Spears. For every generalization action, is there really an equal and opposite reaction? In *Proceedings of the 12th International Conference on Machine Learning*, 1995.
18. C. Soares and P. Brazdil. Zoomed ranking: Selection of classification algrorithms based on relevant performance information. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 126–135. Springer, 2000.
19. S. Y. Sohn. Meta analysis of classification algorithms for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11), 1999.
20. L. Todorovski, P. Brazdil, and C. Soares. Report on the experiments with feature selection in meta-level learning. In *Proceedings of the Workshop on Data Mining, Decision Support, Meta-learning and ILP at PKDD'2000*, pages 27–39, 2000.
21. L. Torgo. *Inductive Learning of Tree-based Regression Models.* PhD thesis, Department of Computer Science, Faculty of Sciences, University of Porto, Porto, Portugal, September 1999.
22. D Wolpert. The existence of a priori distinctions between learning algorithms. *Neural Computation*, 8:1391–1420, 1996.
23. D Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, 1996.