Zoomed Ranking: Selection of Classification Algorithms Based on Relevant Performance Information

Carlos Soares and Pavel B. Brazdil

LIACC/FEP, University of Porto, R. Campo Alegre 823, 4150-180 Porto, Portugal
{csoares,pbrazdil}@ncc.up.pt

Abstract. Given the wide variety of available classification algorithms and the volume of data today's organizations need to analyze, the selection of the right algorithm to use on a new problem is an important issue. In this paper we present a combination of techniques to address this problem. The first one, *zooming*, analyzes a given dataset and selects relevant (similar) datasets that were processed by the candidate algoritms in the past. This process is based on the concept of "distance", calculated on the basis of several dataset characteristics. The information about the performance of the candidate algorithms on the selected datasets is then processed by a second technique, a ranking method. Such a method uses performance information to generate advice in the form of a ranking, indicating which algorithms should be applied in which order. Here we propose the *adjusted ratio of ratios* ranking method. This method takes into account not only accuracy but also the time performance of the candidate algorithms. The generalization power of this ranking method is analyzed. For this purpose, an appropriate methodology is defined. The experimental results indicate that on average better results are obtained with zooming than without it.

1 Introduction

The need for methods which would assist the user in selecting classification algorithms for a new problem has frequently been recognized as an important issue in the fields of Machine Learning (ML) [13, 5] and Knowledge Discovery in Databases (KDD) [3].

Previous meta-learning approaches to algorithm selection consist of suggesting one algorithm or a small group of algorithms that are expected to perform well on the given problem [4, 21, 10]. We believe that a more informative and flexible solution is to provide rankings of the candidate algorithms [15, 19, 5]. A ranking can be used to select just one algorithm, i.e. the one for which the best results are expected. However, if enough resources are available, more than one algorithms may be applied on the given problem.

The problem of constructing rankings can be seen as an alternative to other ML methods, such as classification and regression. Therefore, we must develop

D.A. Zighed, J. Komorowski, and J. Zytkow (Eds.): PKDD 2000, LNAI 1910, pp. 126−135, 2000.

methods to generate rankings and also methodologies to evaluate and compare such methods [5].

Recently, several methods that generate rankings of algorithms based on their past performance have been developed with promising results. Some are based only on accuracy [5], others on accuracy and time [19]. So far, these methods were used without taking the dataset which the ranking was intended for into account. That is, given a new dataset, a ranking was generated by processing all available performance information. However, considering the NFL theorem we cannot expect that all that information is relevant for the problem at hand. Therefore, we do not expect that rankings generated this way accurately represent the relative performance of the algorithms on the new problem.

We, therefore, address the problem of algorithm selection by dividing it into two distinct phases. In the first one we identify a subset of relevant datasets. For that purpose we present a technique called *zooming*. It employs the k-Nearest Neighbor algorithm with a distance function based on a set of statistical, information theoretic and other dataset characterization measures to identify datasets that are similar to the one at hand. More details concerning this are in Section 2.

In the second phase we proceed to construct a ranking on the basis of the performance information of the candidate algorithms on the selected datasets. In Section 3 we present the *adjusted ratio of ratios* ranking method. This method processes performance information on accuracy and time. In Section 4 we evaluate this approach by assessing the gains that can be attributed to zooming. In this analysis we assess the effect of varying the number of neighbors and adopting different compromises between the importance of accuracy and time. Finally, we describe some related work (Section 6) and present conclusions (Section 7).

2 Selection of Relevant Datasets

As explained earlier, the ranking of the candidate algorithms is preceded by selecting, from a set of previously processed datasets, those whose performance information is expected to be relevant for the dataset at hand. The ranking is based on that information. We refer to the selection process as *zooming*, because, given the space of all previously processed datasets, it enables us to focus on the "neighborhood" of the new one.

The relevance of a processed dataset to the one at hand is defined in terms of similarity between them, according to a set of measures (meta-attributes). It is given by function $dist(d_i, d_j) = \sum_x \delta(v_{x,d_i}, v_{x,d_j})$ where d_i and d_j are datasets, v_{x,d_i} is the value of meta-attribute x for dataset d_i, and $\delta(v_{x,d_i}, v_{x,d_j})$ is the distance between the values of meta-attribute x for datasets d_i and d_j. In order to give all meta-attributes the same weight, they are normalized in the following way: $\delta(v_{x,d_i}, v_{x,d_j}) = \frac{|v_{x,d_i} - v_{x,d_j}|}{\max_{k \neq i}(v_{x,d_k}) - \min_{k \neq i}(v_{x,d_k})}$, where $\max_{k \neq i}(v_{x,d_k})$ calculates the maximum value of meta-attribute x for all datasets except d_i and $\min_{k \neq i}(v_{x,d_k})$ calculates the corresponding minimum. Note that, it may be the case that a meta-attribute is not applicable on a dataset. For instance, if dataset d_i has no numerical attributes then it makes no sense to calculate mean skew,

which is a statistical meta-attribute. It seems reasonable to say that, with respect to this attribute, dataset d_i is very close to dataset d_j if d_j does not have any numerical attributes either. We have, thus, determined that δ is 0 in this case. Furthermore, we can say that dataset d_i is quite different from dataset d_k if the latter has some numerical attributes. In this case δ is assigned the maximum distance, 1.

The meta-attributes used were obtained with the Data Characterization Tool (DCT) [11]. They can be grouped into three categories: general, statistical and information theoretic measures. Examples of general measures used are number of attributes and number of cases. As for the statistical measures, we included mean skew and number of attributes with outliers, among others. Finally, some of the information theoretic measures are class entropy and noise-signal ratio. A full listing of the measures used is given in the appendix.

The meta-attributes used were chosen simply because they are provided by DCT and because they were used before for the same purpose [11]. We do not investigate whether they are appropriate or not, and if different weights should be assigned to them in the distance function, although these are questions that we plan to address in the future.

The distance function defined is used as part of the k-Nearest Neighbor (kNN) algorithm to identify the datasets that are most similar to the one at hand. The kNN algorithm is a simple instance-based learner [13]. Given a case, this algorithm simply selects k cases which are nearest to it according to some distance function.

Performance information for the given candidate algorithms on the selected datasets is then used to construct a ranking. Several methods can be used for that purpose [19, 5]. Details of one of them are given in the next section.

3 Ranking Based on Accuracy and Time

In the previous section we have explained how to select performance information that is relevant to the problem at hand. Here we explain how that information can be used to generate a ranking of the corresponding algorithms. Since the datasets selected are similar to the one at hand, it is expected that algorithms perform similarly. In other words, the method should provide us with a good advice for the selection of algorithms to apply on the dataset at hand.

The ranking method presented here is referred to as the *adjusted ratio of ratios* (ARR) ranking method [19]. This method uses information about accuracy and total execution time to rank the given classification algorithms. We start by defining the measure underlying the method and the parameter that determines the relative importance of time and accuracy. Next we describe how the method works. Finally we describe the experimental setup and give an example.

Weighing Success Rates and Time: The ARR method is based on the ratio of success rate ratio and an adjusted time ratio:

$$ARR^{d_i}_{a_p,a_q} = \frac{\frac{SR^{d_i}_{a_p}}{SR^{d_i}_{a_q}}}{1 + \frac{\log\left(\frac{T^{d_i}_{a_p}}{T^{d_i}_{a_q}}\right)}{K_T}} \quad (1)$$

where $SR^{d_i}_{a_p}$ and $T^{d_i}_{a_p}$ are the success rate and time of algorithm a_p on dataset d_i, respectively, and K_T[1] is a user-defined value that determines the relative importance of time.

The formula may seem ad-hoc at first glance, but its form can be related to the ones used in other areas of science. We can look at the ratio of success rates, $SR^{d_i}_{a_p}/SR^{d_i}_{a_q}$, as a measure of the advantage, and the ratio of times, $T^{d_i}_{a_p}/T^{d_i}_{a_q}$, as a measure of the disadvantage of algorithm a_p relative to algorithm a_q on dataset d_i. The former can be considered a *benefit* while the latter a *cost*. Thus, by dividing a measure of the benefit by a measure of the cost, we assess the overall quality of an algorithm. A similar philosophy underlies the efficiency measure of Data Envelopment Analysis (DEA) that has been proposed for multicriteria evaluation of data mining algorithms [15].

Furthermore, the use of ratios of a measure, namely success rate, has been shown earlier to lead to competitive rankings overall when compared to other ways of aggregating performance information [19, 5]. A parallel can be established between the ratio of success rates and performance scatterplots that have been used in some empirical studies involving comparisons of classification algorithms [17].

Relative Importance of Accuracy and Time: The reason behind the adjustment of the time ratio is concerned with the fact that time ratios have, in general, a much wider range of possible values than success rate ratios. Therefore, if a simple time ratio were used, it would dominate the ratio of ratios. By using $\log\left(T^{d_i}_{a_p}/T^{d_i}_{a_q}\right)$, i.e. the order of magnitude of the difference between the times of algorithms a_p and a_q, this effect is minimized. We, thus, obtain values that vary around 1, as happens with the success rate ratio. The parameter K_T enables us to determine the relative importance of the two criteria, which is expected to vary for different applications.

However, the use of the K_T parameter is not very intuitive and would present an obstacle if the method were to be used by non-expert users. We have therefore devised a way to obtain K_T in a way that is more user-friendly. We need an estimate of how much accuracy we are willing to trade for a 10 times speedup or slowdown. The defined setting is represented as 10x≅X%. The parameter K_T is then approximated by $1/X\%$. For instance, if the user is willing to trade 10% of accuracy for 10 times speedup/slowdown (10x≅10%), then $K_T = 1/10\% = 10$.

[1] Here, to avoid confusion with the number of nearest-neighbors (k), we refer to the compromise between time and accuracy as K_T, rather then K, as in [19].

Aggregating ARR Information: The method aggregates the given performance information as follows. First, we create an *adjusted ratio of ratios table* for each dataset. The table for dataset d_i is filled with the corresponding values of adjusted ratio of ratios, $ARR^{d_i}_{a_p,a_q}$. Next, we calculate a *pairwise mean adjusted ratio of ratios* for each pair of algorithms, $ARR_{a_p,a_q} = \left(\sum_{d_i} ARR^{d_i}_{a_p,a_q}\right)/n$ where n is the number of datasets. This represents an estimate of the general advantage/disadvantage of algorithm a_p over algorithm a_q. Finally, we derive the *overall mean adjusted ratio of ratios* for each algorithm, $ARR_{a_p} = \left(\sum_{a_q} ARR_{a_p,a_q}\right)/(m-1)$ where m is the number of algorithms. The ranking is derived directly from this measure. The higher the value an algorithm obtains, the higher the corresponding rank.

Experimental Setup: Before presenting an example, we describe the experimental setting. We have used three decision tree classifiers, C5.0, C5.0 with boosting [18] and Ltree, which is a decision tree that can introduce oblique decision surfaces [8]. We have also used an instance based classifier, TiMBL [7], a linear discriminant and a naive bayes classifier [12]. We will refer to these algorithms as `c5`, `c5boost`, `ltree`, `timbl`, `discrim` and `nbayes`, respectively. We ran these algorithms with default parameters on 16 datasets. Seven of those (`australian`, `diabetes`, `german`, `heart`, `letter`, `segment` and `vehicle`) are from the StatLog repository[2] and the rest (`balance-scale`, `breast-cancer-wisconsin`, `glass`, `hepatitis`, `house-votes-84`, `ionosphere`, `iris`, `waveform` and `wine`) are from the UCI repository[3] [2]. The error rate and time were estimated using 10-fold cross-validation.[4]

Example: Supposing that we want to obtain a ranking of the given algorithms to use on the `segment` dataset (*test dataset*), without having tested any of them on that dataset. We must, thus, exclude the dataset in question from consideration and use only the remaining datasets (*training datasets*) in the process. In Table 1 we present two rankings. The first is generated by ARR based on all training datasets while the second is based only on the two datasets that are most similar to `segment`. Here we refer to zooming with a given k followed by the application of ARR on the selected datasets as $Z_k(ARR)$. We note that ARR can be considered as a special case of $Z_k(ARR)$, where k spans across all training datasets. In our meta-data, the two datasets that are most similar to `segment` are `ionosphere` ($dist = 4.99$) and `glass` ($dist = 8.28$). The results presented are obtained with 10x≅1% or $K_T = 100$.

[2] See `http://www.liacc.up.pt/ML/statlog/`.

[3] Some preparation was necessary in some cases, so some of the datasets may not be exactly the same as the ones used in other experimental work.

[4] It must be noted that this is not a comparative study of the algorithms involved. Not all of them were executed on the same machine. However, this does not conflict with the purpose of this work because, in a real-world setting, not all algorithms may be available on the same machine.

Table 1. Recommended rankings for the `segment` dataset based on all the other datasets and on its two nearest neighbors (*left*). The ideal ranking and part of the calculation of Spearman's correlation for the latter recommended ranking (*right*)

	Recommended Ranking				**Ideal**		**Spearman**
	ARR		$Z_2(ARR)$		**Ranking**		
Rank	a_p	ARR_{a_p}	a_p	ARR_{a_p}	a_p	ARR_{a_p}	$D^2_{a_p}$
1	ltree	1.066	c5boost	1.151	c5boost	1.151	0
2	c5boost	1.057	c5	1.075	c5	1.088	0
3	discrim	1.046	ltree	1.049	ltree	1.088	0
4	c5	1.009	discrim	0.991	discrim	1.031	0
5	nbayes	0.974	timbl	0.902	nbayes	1.008	1
6	timbl	0.919	nbayes	0.900	timbl	0.769	1

We observe that the rankings generated are quite different. The obvious question is which one is the best, i.e. the one that most accurately reflects the actual performance of the algorithms on the test dataset? We try to answer it in the next section.

4 Assessment of Generalization Power

A ranking should naturally be evaluated by comparing it to the actual performance of the algorithms on the dataset the ranking is generated for. Our approach consists of using that performance information to generate an *ideal ranking* [5]. The quality of the ranking being evaluated (*recommended ranking*) is assessed by measuring the distance to the ideal ranking.

The ideal ranking represents the correct ordering of the algorithms on a test dataset. Here, it is based on the assumption that the ARR measure (Eq. 1) appropriately represents the criteria to be used to evaluate the results and that the measured accuracies and times are good estimates of the corresponding true accuracies and times.

The distance between two rankings is best calculated using correlation. Here we use Spearman's rank correlation coefficient [16]. To illustrate this measure, we show how we evaluate the ranking recommended by $Z_2(ARR)$ for the `segment` dataset with 10x≅1% (Table 1). First we calculate the squared differences, $D^2_{a_p}$, between the recommended and the ideal ranks for algorithm a_p. Then we calculate $D^2 = \sum_{a_p} D^2_{a_p}$. The score of the recommended ranking is the correlation coefficient, $r_s = 1 - \frac{6D^2}{n^3 - n}$, where n is the number of algorithms. In the example used $D^2 = 2$ and $r_s = 0.943$, while the correlation for the ranking recommended by ARR is 0.714.

It is not possible to draw any conclusion based on one dataset only. We have, therefore, carried a leave-one-out procedure. As the name suggests, in each iteration one dataset is selected as the test dataset. The rankings generated based on the corresponding training datasets are then evaluated in the way described in the previous paragraphs. The methods compared were $Z_2(ARR)$, $Z_4(ARR)$

Table 2. Mean correlations obtained by $Z_2(ARR)$, $Z_4(ARR)$ and ARR for different values of K_T. The +/- column indicates the number of datasets where the corresponding method has higher/lower correlation than ARR

	$Z_2(ARR)$		$Z_4(ARR)$		**ARR**
10x≅	**mean** r_S	+/-	**mean** r_S	+/-	**mean** r_S
10%	0.64	9/6	*0.68*	9/2	0.64
1%	*0.63*	10/6	0.56	9/6	0.54
0.1%	*0.64*	10/6	0.57	8/6	0.54

and also ARR without zooming. The later method serves as a baseline to assess whether zooming is really advantageous and if it is, quantify that advantage. The values for k were determined in order for the rankings to be built based on 10% and 25%, respectively, of the datasets available. These seem to be sensible values. As for K_T, values were chosen for the time to have large, medium and small importance, respectively 10x≅10%, 1% and 0.1% ($K_T = 10, 100$ and 1000).

If we analyze the mean correlations in Table 2, we observe that when time is predominant (10x≅10%), $Z_4(ARR)$ performs better than ARR while the mean correlations of $Z_2(ARR)$ and ARR are equal. The situation changes when time is given less importance. The advantage of $Z_4(ARR)$ over ARR remains, although in a smaller scale. On the other hand, $Z_2(ARR)$ is now considerably better than ARR, exceeding also the performance of $Z_4(ARR)$. Note that the performance of $Z_2(ARR)$ seems to be quite robust with respect to the variation of K_T.

5 Discussion

According to the previous section, it appears that zooming improves the quality of the rankings generated. However, we would like to obtain statistical support for this claim. For that purpose, we have applied Friedman's test to the results obtained [16]. The values obtained for M, after correction due to the occurrence of ties, are 3.37, 1.34 and 1.24, for 10x≅10%, 1% and 0.1%, respectively. The critical value is 6.5 for $k = 3$[5] and $n = 16$ at a 5% significance level, so we do not reject the null hypothesis that their performance is not significantly different. We, thus, have no statistical evidence of the difference in mean correlation of the methods compared. We must note, however, that, although Friedman's test was used before for the same purpose [19, 5] with a conclusive result, it is a distribution-free test, which implies that it is not expected to be very powerful. Also we have restricted our experiments to 6 algorithms and 16 datasets. We expect that by increasing the number of datasets and algorithms used, we are able to obtain statistical evidence of the improvement brought by zooming. The extended study is currently being carried out.

One drawback of the ideal ranking used is that it is built with average accuracies and times. Given that these are only estimates, the ranking generated may

[5] This k is a parameter of Friedman's test representing the number of methods being compared.

not be reliable. To minimize this problem, the ideal ranking can be generated as a set of n orderings, one for the results in each fold of the cross-validation procedure used to estimate the performance of the algorithms. A similar procedure as been used before with satisfactory results [20, 5].

As for the measure of distance between rankings used here, it has been shown that correlation is appropriate for that purpose [20]. One drawback is, though, the lack of distinction between rank importance. For instance, it is obvious that the switch made between the 5th and 6th algorithm by the $Z_2(ARR)$ on the `segment` dataset (Table 1) is less important than if it would involve the 1st and the 2nd (`c5boost` and `c5`). We have previously developed a measure to solve this problem, *weighted correlation* [20]. However, it has not yet been thoroughly analyzed, and, thus was not used here. An alternative to Spearman's correlation coefficient that could be tried is Kendall's tau [16].

6 Related Work

Meta-knowledge as been used before for the purpose of algorithm selection. This knowledge can be either of theoretical or of experimental origin, or a mixture of both. The rules described by Brodley [6] for instance, captured the knowledge of experts concerning the applicability of certain classification algorithms. The meta-knowledge of [1], [4] and [9] was of experimental origin and was obtained by *meta-learning* on past performance information of the algorithms. Its objective is to capture certain relationships between the measured dataset characteristics and the relative performance of the algorithms. As was demonstrated, meta-knowledge can be used to predict the errors of individual algorithms or construct a ranking with a certain degree of success.

Not much work exists in the areas of Machine Learning or KDD concerning multicriteria ranking and evaluation. A noteworthy exception is the work of Nakhaeizadeh et al. [15, 14], who have applied a technique that originated in the area of Operations Research, Data Envelopment Analysis (DEA). It remains to be seen how this approach compares with the method described here.

7 Conclusions

We have presented a combination of techniques that uses past performance information to assist the user in the selection of a classification algorithm for a given problem. The first technique, zooming, works by selecting datasets and associated performance information that is relevant to the problem at hand. This process is based on the distance between datasets, according to a set of statistical, information theoretic and other measures. Here, it is performed using the k-Nearest Neighbor algorithm. We have selected dataset measures that were previously used for the same purpose. Work is under way to select the most predictive subset of those measures.

The ranking method used here is the Adjusted Ratio of Ratios (ARR) method. This is a multicriteria method that takes into account both accuracy and total

execution time information. It has a parameter that enables us to determine the relative importance of each criteria. One of the main advantages is its intuitiveness, which is essential to enable its use by non-experts.

We have reported experiments varying the number of neighbors and the relative importance of accuracy and time. The results obtained are compared to results obtained by ARR without zooming. It appears that zooming improves the quality of the generated rankings, although the results obtained are not significantly different according to the Friedman's test.

In summary, our contributions are (1) exploiting rankings rather then classification or regression, (2) providing a general evaluation methodology for ranking, (3) providing a way of combining success rate and time and (4) exploiting dataset characteristics to select relevant performance information prior to ranking.

Acknowledgments We would like to thank the METAL partners for useful discussions. Also thanks to João Gama for providing his implementations of Linear Discriminant and Naive Bayes and to Rui Pereira for implementing an important part of the methods. The financial support from ESPRIT project METAL, project ECO under PRAXIS XXI, FEDER, Programa de Financiamento Plurianual de Unidades de I&D and Faculty of Economics is gratefully acknowledged.

References

1. D.W. Aha. Generalizing from case studies: A case study. In D. Sleeman and P. Edwards, editors, *Proceedings of the Ninth International Workshop on Machine Learning (ML92)*, pages 1–10. Morgan Kaufmann, 1992.
2. C. Blake, E. Keogh, and C.J. Merz. Repository of machine learning databases, 1998. http:/www.ics.uci.edu/~mlearn/MLRepository.html.
3. R.J. Brachman and T. Anand. The process of knowledge discovery in databases. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 2, pages 37–57. AAAI Press/The MIT Press, 1996.
4. P. Brazdil, J. Gama, and B. Henery. Characterizing the applicability of classification algorithms using meta-level learning. In F. Bergadano and L. de Raedt, editors, *Proceedings of the European Conference on Machine Learning (ECML-94)*, pages 83–102. Springer-Verlag, 1994.
5. P. Brazdil and C. Soares. A comparison of ranking methods for classification algorithm selection. In R.L. de Mántaras and E. Plaza, editors, *Machine Learning: Proceedings of the 11th European Conference on Machine Learning ECML2000*, pages 63–74. Springer, 2000.
6. C.E. Brodley. Addressing the selective superiority problem: Automatic Algorithm/Model class selection. In P. Utgoff, editor, *Proceedings of the 10th International Conference on Machine Learning*, pages 17–24. Morgan Kaufmann, 1993.
7. W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van Den Bosch. TiMBL: Tilburg memory based learner v2.0 guide. Technical Report 99-01, ILK, 1999.
8. J. Gama. Probabilistic linear tree. In D. Fisher, editor, *Proceedings of the 14th International Machine Learning Conference (ICML97)*, pages 134–142. Morgan Kaufmann, 1997.

9. J. Gama and P. Brazdil. Characterization of classification algorithms. In C. Pinto-Ferreira and N.J. Mamede, editors, *Progress in Artificial Intelligence*, pages 189–200. Springer-Verlag, 1995.
10. 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, November 1999.
11. G. Lindner and R. Studer. AST: Support for algorithm selection with a CBR approach. In C. Giraud-Carrier and B. Pfahringer, editors, *Recent Advances in Meta-Learning and Future Work*, pages 38–47. J. Stefan Institute, 1999.
12. D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.
13. T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
14. G. Nakhaeizadeh and A. Schnabl. Towards the personalization of algorithms evaluation in data mining. In R. Agrawal and P. Stolorz, editors, *Proceedings of the Third International Conference on Knowledge Discovery & Data Mining*, pages 289–293. AAAI Press, 1997.
15. G. Nakhaeizadeh and A. Schnabl. Development of multi-criteria metrics for evaluation of data mining algorithms. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proceedings of the Fourth International Conference on Knowledge Discovery in Databases & Data Mining*, pages 37–42. AAAI Press, 1998.
16. H.R. Neave and P.L. Worthington. *Distribution-Free Tests*. Routledge, 1992.
17. F. Provost and D. Jensen. Evaluating knowledge discovery and data mining. Tutorial Notes, Fourth International Conference on Knowledge Discovery and Data Mining, 1998.
18. R. Quinlan. *C5.0: An Informal Tutorial*. RuleQuest, 1998. http://www.rulequest.com/see5-unix.html.
19. C. Soares. Ranking classification algorithms on past performance. Master's thesis, Faculty of Economics, University of Porto, 1999. http://www.ncc.up.pt/∼csoares/miac/thesis_revised.zip.
20. C. Soares, P. Brazdil, and J. Costa. Measures to compare rankings of classification algorithms. In *Proceedings of the Seventh Conference of the International Federation of Classification Societies IFCS (to Be Published)*, 2000.
21. L. Todorovski and S. Dzeroski. Experiments in meta-level learning with ILP. In *Proceedings of the Third European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD99)*, pages 98–106, 1999.

Appendix The dataset characterization measures used in this study were obtained with the DCT program. They consist of simple (number of attributes, number of symbolic and numerical attributes, number of cases and classes, default accuracy, standard-deviation of classes, number of missing values and cases with missing values), statistical (mean skew and kurtosis, number of attributes with outliers, M statistic, degrees of freedom of the M statistic, chi-square M statistic, SD ratio, relative importance of the most important eigenvalue, canonical correlation for the most discriminant function, Wilks Lambda and Bartlett's V statistics, chi square V statistic and number of discriminant functions) and information theoretic measures (minimum, maximum and average symbolic attributes, class entropy, attributes entropy, average mutual information, joint entropy, equivalent number of attributes and noise signal ratio). More details can be found in [11, 12].