A Comparison of Ranking Methods for Classification Algorithm Selection

Pavel B. Brazdil and Carlos Soares

LIACC/Faculty of Economics, University of Porto
R. Campo Alegre, 823, 4150-800 Porto, Portugal
{pbrazdil,csoares}@ncc.up.pt

Abstract. We investigate the problem of using past performance information to select an algorithm for a given classification problem. We present three ranking methods for that purpose: average ranks, success rate ratios and significant wins. We also analyze the problem of evaluating and comparing these methods. The evaluation technique used is based on a leave-one-out procedure. On each iteration, the method generates a ranking using the results obtained by the algorithms on the training datasets. This ranking is then evaluated by calculating its distance from the ideal ranking built using the performance information on the test dataset. The distance measure adopted here, average correlation, is based on Spearman's rank correlation coefficient. To compare ranking methods, a combination of Friedman's test and Dunn's multiple comparison procedure is adopted. When applied to the methods presented here, these tests indicate that the success rate ratios and average ranks methods perform better than significant wins.

Key Words: classifier selection, ranking, ranking evaluation

1 Introduction

The selection of the most adequate algorithm for a new problem is a difficult task. This is an important issue, because many different classification algorithms are available. These algorithms originate from different areas like Statistics, Machine Learning and Neural Networks and their performance may vary considerably [12]. Recent interest in combination of methods like bagging, boosting, stacking and cascading has resulted in many new additional methods. We could reduce the problem of algorithm selection to the problem of algorithm performance comparison by trying all the algorithms on the problem at hand. In practice this is not feasible in many situations, because there are too many algorithms to try out, some of which may be quite slow., especially with large amounts of data, as it is common in Data Mining. An alternative solution would be to try to identify the single best algorithm, which could be used in all situations. However, the *No Free Lunch* (NFL) theorem [19] states that if algorithm A outperforms algorithm

B on some cost functions, then there must exist exactly as many other functions where B outperforms A.

All this implies that, according to the problem at hand, specific recommendation should be given concerning which algorithm(s) should be used or tried out. Brachman et al. [3] describe algorithm selection as an exploratory process, highly dependent on the analyst's knowledge of the algorithms and of the problem domain, thus something which lies somewhere on the border between engineering and art.

As it is usually difficult to identify a single best algorithm reliably, we believe that a good alternative is to provide a ranking. In this paper we are concerned with ranking methods. These methods use experimental results obtained by a set of algorithms on a set of datasets to generate an ordering of those algorithms. The ranking generated can be used to select one or more suitable algorithms for a new, previously unseen problem. In such a situation, only the top algorithm, i.e. the algorithm expected to achieve the best performance, may be tried out or, depending on the available resources, the tests may be extended to the first few algorithms in the ranking.

Considering the NFL theorem we cannot expect that a single best ranking of algorithms could be found and be valid for all datasets. We address this issue by dividing the process into two distinct phases. In the first one, we identify a subset of relevant datasets that should be taken into account later. In the second phase, we proceed to construct a ranking on the basis of the datasets identified. In this paper we restrict our attention to the second phase only. Whatever method we use to identify the relevant datasets, we still need to resolve the issue concerning which ranking method is the best one.

Our aim is to examine three ranking methods and evaluate their ability to generate rankings which are consistent with the actual performance information of the algorithms on an unseen dataset. We also investigate the issue whether there are significant differences between them, and, if there are, which method is preferable to the others.

2 Ranking Methods

The ranking methods presented here are: *average ranks* (AR), *success rate ratios* (SRR) and *significant wins* (SW). The first method, AR, uses, as the name suggests, individual rankings to derive an overall ranking. The next method, SRR, ranks algorithms according to the relative advantage/disadvantage they have over the other algorithms. A parallel can be established between the ratios underlying SRR and performance scatter plots that have been used in some empirical studies to compare pairs of algorithms [14]. Finally, SW is based on pairwise comparisons of the algorithms using statistical tests. This kind of tests is often used in comparative studies of classification algorithms.

Before presenting the ranking methods, we describe the experimental setting. We have used three decision tree classifiers, C5.0, C5.0 with boosting [15] and Ltree, which is a decision tree which can introduce oblique decision surfaces [9]. We have also used an instance based classifier, TiMBL [6], 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 on 16 datasets. Seven of those (australian, diabetes, german, heart, letter, segment and vehicle) are from the StatLog repository[1] and the rest (balance-scale, breast-cancer-wisconsin, glass, hepatitis, house-votes-84, ionosphere, iris, waveform and wine) are from the UCI repository[2] [2]. The error rate was estimated using 10-fold cross-validation.

2.1 Average Ranks Ranking Method

This is a simple ranking method, inspired by Friedman's M statistic [13]. For each dataset we order the algorithms according to the measured error rates[3] and assign ranks accordingly. The best algorithm will be assigned rank 1, the runner-up, 2, and so on. Let r_j^i be the rank of algorithm j on dataset i. We calculate the *average rank* for each algorithm $\bar{r}_j = \left(\sum_i r_j^i\right)/n$, where n is the number of datasets. The final ranking is obtained by ordering the average ranks and assigning ranks to the algorithms accordingly. The average ranks based on all the datasets considered in this study and the corresponding ranking are presented in Table 1.

Table 1. Rankings generated by the three methods on the basis of their accuracy on all datasets.

	AR		**SRR**		**SW**	
Algorithm (j)	$\bar{r}_j$	**Rank**	SRR_j	**Rank**	pw_j	**Rank**
c5	3.9	4	1.017	4	0.225	4
ltree	2.2	1	1.068	2	0.425	2
timbl	5.4	6	0.899	6	0.063	6
discrim	2.9	3	1.039	3	0.388	3
nbayes	4.1	5	0.969	5	0.188	5
c5boost	2.6	2	1.073	1	0.438	1

2.2 Success Rate Ratios Ranking Method

As the name suggests this method employs ratios of success rates between pairs of algorithms. We start by creating a *success rate ratio table* for each of the datasets. Each slot of this table is filled with $SRR_{j,k}^i = \left(1 - ER_j^i\right) / \left(1 - ER_k^i\right)$,

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

[2] 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.

[3] The measured error rate refers to the average of the error rates on all the folds of the cross-validation procedure.

where ER_j^i is the measured error rate of algorithm j on dataset i. For example, on the `australian` dataset, the error rates of `timbl` and `discrim` are 19.13% and 14.06%, respectively, so $SRR_{\text{timbl,discrim}}^{\text{australian}} = (1 - 0.1913)/(1 - 0.1406) = 0.941$, indicating that `discrim` has advantage over `timbl` on this dataset. Next, we calculate a *pairwise mean success rate ratio*, $SRR_{j,k} = \left(\sum_i SRR_{j,k}^i\right)/n$, for each pair of algorithms j and k, where n is the number of datasets. This is an estimate of the general advantage/disadvantage of algorithm j over algorithm k. Finally, we derive the *overall mean success rate ratio* for each algorithm, $SRR_j = \left(\sum_k SRR_{j,k}\right)/(m - 1)$ where m is the number of algorithms (Table 1). The ranking is derived directly from this measure. In the current setting, the ranking obtained is quite similar to the one generated with AR, except for `c5boost` and `ltree`, which have swapped positions.

2.3 Significant Wins Ranking Method

This method builds a ranking on the basis of results of pairwise hypothesis tests concerning the performance of pairs of algorithms. We start by testing the significance of the differences in performance between each pair of algorithms. This is done for all datasets. In this study we have used paired t tests with a significance level of 5%. We have opted for this significance level because we wanted the test to be relatively sensitive to differences but, at the same time, as reliable as possible. A little less than 2/3 (138/240) of the hypothesis tests carried out detected a significant difference. We denote the fact that algorithm j is significantly better than algorithm k on dataset i as $ER_j^i \ll ER_k^i$. Then, we construct a *win table* for each of the datasets as follows. The value of each cell, $W_{j,k}^i$, indicates whether algorithm j wins over algorithm k on dataset i at a given significance level and is determined in the following way:

$$W_{j,k}^i = \begin{cases} 1 & \text{iff } ER_j^i \ll ER_k^i \\ -1 & \text{iff } ER_k^i \ll ER_j^i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Note that $W_{j,k}^i = -W_{k,j}^i$ by definition. Next, we calculate the *pairwise estimate of the probability of winning* for each pair of algorithms, $pw_{j,k}$. This is calculated by dividing the number of datasets where algorithm j is significantly better than algorithm k by the number of datasets, n. This value estimates the probability that algorithm j is significantly better than algorithm k. For instance, `ltree` is significantly better than `c5` on 5 out of the 16 datasets used in this study, thus $pw_{\text{ltree,c5}} = 5/16 = 0.313$. Finally, we calculate the *overall estimate of the probability of winning* for each algorithm, $pw_j = \left(\sum_k pw_{j,k}\right)/(m-1)$ where m is the number of algorithms (Table 1). The values obtained are used as a basis for constructing the overall ranking. In our example, $pw_{\text{c5boost}} = 0.438$, which is the largest one and, thus, `c5boost` appears first in the ranking, closely followed by `ltree`, as happened in the ranking generated with SRR.

3 Evaluation

Having considered three ranking methods, we would like to know whether their performances differ, and, if they do, which is the best one. For that purpose we use a leave-one-out procedure. For each dataset (*test dataset*), we do the following:

1. Build a *recommended ranking* by applying the ranking method under evaluation to all but the test dataset (*training datasets*).
2. Build an *ideal ranking* for the test dataset.
3. Calculate the distance between the two rankings using an appropriate measure.

The score of each of the ranking methods is expressed in terms of the mean distance.

The ideal ranking represents the correct ordering of the algorithms on a test dataset, and it is constructed on the basis of their performance (measured error rate) on that dataset. Therefore, the distance between the recommended ranking and the ideal ranking for some dataset is a measure of the quality of the former and thus also of the ranking method that generated it.

Creating an ideal ranking is not a simple task, however. Given that only a sample of the population is known, rather than the whole population, we can only *estimate* the error rate of algorithms. These estimates have confidence intervals which may overlap. Therefore, the ideal ranking obtained simply by ordering the estimates may often be quite meaningless. For instance, Table 2 shows one ranking for the `glass` dataset, where `c5` and `ltree` are ranked in 2nd and 3rd, respectively. The performance of these algorithms on this dataset is, however, not significantly different, according to a paired t test at a 5% significance level. Thus, we would not consider a ranking where the position of `c5` and `ltree` is interchanged worse than the one we show. In such a situation, these algorithms often swap positions in different folds of the N-fold cross-validation procedure (Table 2). Therefore, we use N orderings to represent an ideal ordering.

To calculate the distance between the recommended ranking and each of the N orderings that represent the ideal ranking, we use Spearman's rank correlation coefficient [13]. The score of the recommended ranking is expressed as the average of the N correlation coefficients. This measure is referred here as *average correlation, $\bar{C}$*.

To illustrate this performance measure, we evaluate the ranking recommended by SW for the `glass` dataset, focusing on the first fold (Table 2). Note that `c5` and `c5boost` share the first place in the ordering obtained in this fold, so they are both assigned rank $\frac{1+2}{2} = 1.5$, following the method in [13]. A similar situation occurs with `c5` and `nbayes` in the recommended ranking[4]. To calculate Spearman's rank correlation coefficient we first calculate $D^2 = \sum D_i^2$, where D_i is the difference between the recommended and the ideal rank for algorithm i. The correlation coefficient is $r_s = 1 - \frac{6D^2}{n^3 - n}$, where n is the number of datasets.

[4] The same reasoning is applied when more than two algorithms are tied.

In our example, $D^2 = 17.5$ and $r_s = 0.5$, where n is the number of algorithms. These calculations are repeated for all the folds, permitting to calculate the score of the recommended ranking, $\bar{C}$, as the average of the individual coefficients.

Table 2. Some steps in the calculation of the correlation coefficient between recommended and ideal ranking for the `glass` dataset.

		Average		**Fold 1**			**Fold 5**		
Algorithm (i)	**Rec. rank**	**ER (%)**	**rank**	**ER (%)**	**rank**	D_i^2	**ER (%)**	**rank**	D_i^2
c5	4.5	29.9	2	28.6	1.5	9	47.6	3.5	1
ltree	1	31.8	3	31.8	3	4	42.9	2	1
timbl	6	45.2	5	50.0	5	1	52.4	5	1
discrim	3	36.9	4	36.4	4	1	47.6	3.5	0.25
nbayes	4.5	48.7	6	59.1	6	2.25	71.4	6	2.25
c5boost	2	23.8	1	28.6	1.5	0.25	23.8	1	1

Table 3 presents the results of the evaluation of the three ranking methods presented earlier. These results indicate that AR is the best method as the mean $\bar{C}$ has the highest value (0.426). It is followed by SRR (0.411) and SW (0.387). However, when looking at the standard deviations, the differences do not seem to be too significant. A comparison using an appropriate statistical test needs to be carried out. It is described in the next section.

Table 3. Average correlation scores for the three ranking methods.

Test dataset	**AR**	**SRR**	**SW**
australian	0.417	0.503	0.494
balance-scale	0.514	0.440	0.651
breast-cancer-wisconsin	0.146	0.123	0.123
diabetes	0.330	0.421	0.421
german	0.460	0.403	0.403
glass	0.573	0.573	0.413
heart	0.324	0.339	0.339
hepatitis	0.051	0.049	0.049
house-votes-84	0.339	0.307	0.307
ionosphere	0.326	0.326	0.120
iris	0.270	0.167	0.167
letter	0.086	0.086	-0.086
segment	0.804	0.853	0.804
vehicle	0.800	0.731	0.731
waveform	0.714	0.663	0.663
wine	0.621	0.587	0.587
Mean $\bar{\mathbf{C}}$	**0.426**	**0.411**	**0.387**
StdDv	0.235	0.235	0.262

4 Comparison

To test whether the ranking methods have significantly different performance we have used a distribution-free hypothesis test on the difference between more than two population means, Friedman's test [13]. This hypothesis test was used because we have no information about the distribution of the correlation coefficient in the population of datasets, the number of samples is larger than 2 and also because the samples are related, i.e. for each ranking method the correlation coefficients are calculated for the same part of each dataset. According to Neave and Worthington [13] not many methods can compete with Friedman's test with regard to both power and ease of computation.

Here, the hypotheses are:

> H_o: There is no difference in the mean average correlation coefficients for the three ranking methods.
> H_1: There are some differences in the mean average correlation coefficients for the three ranking methods.

We will use results for fold 1 on datasets `australian` and `ionosphere` to illustrate how this test is applied (Table 4). First, we rank the correlation coefficients of all the ranking methods for each fold on each dataset. We thus obtain $R_j^{d,f}$, representing the rank of the correlation obtained by ranking method j on fold f of dataset d, when compared to the corresponding correlations obtained by the other methods. Next, we calculate the mean rank for each method, $\bar{R}_j$, and the overall mean rank across all methods, $\bar{R}$. As each method is ranked from 1 to k, where k is the number of methods being compared (3 in the present case), we know that $\bar{R} = \frac{k+1}{2} = 2$. Then we calculate the sum of the squared differences between the mean rank for each method and the overall mean rank, $S = \sum_{j=1}^{k}(\bar{R}_j - \bar{R})$. Finally, we calculate Friedman's statistic, $M = \frac{12nS}{k(k+1)}$, where n is the number of points being compared, which in this case is the total number of folds. In this simple example where $n = 2$, $S = 0.5$ and $M = 1$. The critical region for this test has the form $M \geq$ *critical value*, where the critical value is obtained from the appropriate table, given the number of methods (k) and the number of points (n).

Table 4. Some steps in the application of Friedman's test and Dunn's Multiple Comparison procedure on folds 1 of the `australian` and `ionosphere` datasets.

	australian		**ionosphere**				
Method (j)	r_s	$R_j^{\text{australian},1}$	r_s	$R_j^{\text{ionosphere},1}$	$\bar{R}_j$	$(\bar{R}_j - \bar{R})^2$	$\sum_{d,f} R_j^{d,f}$
SW	0.357	2	-0.371	3	2.5	0.25	5
SRR	0.314	3	-0.086	1	2	0	4
AR	0.371	1	-0.214	2	1.5	0.25	3

Dealing with Ties. When applying this test ties may occur, meaning that two ranking methods have the same correlation coefficient on a given fold of a given dataset. In that case, the average rank value is assigned to all the methods involved, as explained earlier for Spearman's correlation coefficient. When the number of ties is significant, the M statistic must be corrected [13]. First, we calculate Friedman's statistic as before, M. Then, for each fold of each dataset, we calculate $t^* = t^3 - t$, where t is the number of methods contributing to a tie. Next, we obtain T by adding up all t^*'s. The correction factor is $C = 1 - \frac{T}{n(k^3-k)}$ where k and n are the number of methods and the number of points, as before. The modified statistic is $M^* = M/C$. The critical values for M^* are the same as for M. More details can be found in [13, 16].

Results. With the full set of results available, $\bar{R}_{\mathrm{AR}} = 1.950$, $\bar{R}_{\mathrm{SRR}} = 1.872$ and $\bar{R}_{\mathrm{SW}} = 2.178$. Given that the number of ties is high (55%), the statistic is appropriately corrected, yielding $M^* = 13.39$. The critical value for the number of methods being compared ($k = 3$) and the number of points in each ($n = \#datasets * \#folds = 160$) is 9.210 for a significance level of 1%[5]. As $M^* > 9.210$, we are 99% confident that there are some differences in the $\bar{C}$ scores for the three ranking methods, contrary to what could be expected.

Which Method is Better? Naturally, we must now determine which methods are different from one another. To answer this question we use Dunn's multiple comparison technique [13]. Using this method we test $p = \frac{1}{2}k(k-1)$ hypotheses of the form:

> $\mathrm{H}_o^{(i,j)}$: There is no difference in the mean average correlation coefficients between methods i and j.
>
> $\mathrm{H}_1^{(i,j)}$: There is some difference in the mean average correlation coefficients between methods i and j.

We use again the results for fold 1 on datasets `australian` and `ionosphere` to illustrate how this procedure is applied (Table 4). First, we calculate the rank sums for each method. Then we calculate $T_{i,j} = D_{i,j}/stdev$ for each pair of ranking methods, where $D_{i,j}$ is the difference in the rank sums of methods i and j, and. $stdev = \sqrt{\frac{nk(k+1)}{6}}$. As before, k is the number of methods and n is the number of points in each. In our simple example, where $n = 2$ and $k = 3$, $stdev = 2$, $D_{\mathrm{SRR,AR}} = D_{\mathrm{SW,SRR}} = 1$ and $D_{\mathrm{SW,AR}} = 2$, and then $|T_{\mathrm{SW,SRR}}| = |T_{\mathrm{SRR,AR}}| = 0.5$ and $|T_{\mathrm{SW,AR}}| = 1$.

The values of $|T_{i,j}|$, which follow a normal distribution, are used to reject or accept the corresponding null hypothesis at an appropriate confidence level. As we are doing multiple comparisons, we have to carry out the Bonferroni adjustment to the chosen overall significance level. Neave and Worthington [13] suggest a rather high overall significance level (between 10% and 25%) so that we

[5] We have used the critical value for $n = \infty$, which does not affect the result of the test.

could detect any differences at all. The use of high significance levels naturally carries the risk of obtaining false significant differences. However, the risk is somewhat reduced thanks to the previous application of the Friedman's test, which concluded that there exist differences in the methods compared. Here we use an overall significance level of 25%. Applying the Bonferroni adjustment, we obtain $\alpha = \mathit{overall}\ \alpha / k\,(k-1) = 4.17\%$ where $k = 3$, as before. Consulting the appropriate table we obtain the corresponding critical value, $z = 1.731$. If $|T_{i,j}| \geq z$ then the methods i and j are significantly different.

Given that three methods are being compared, the number of hypotheses being tested is, $p = 3$. We obtain $|T_{\mathrm{SRR,SW}}| = 1.76$, $|T_{\mathrm{AR,SW}}| = 3.19$ and $|T_{\mathrm{SRR,AR}}| = 1.42$. As $|T_{\mathrm{SRR,SW}}| > 1.731$ and $|T_{\mathrm{AR,SW}}| > 1.731$, we conclude that both SRR and AR are significantly better than SW.

5 Discussion

Considering the variance of the obtained $\bar{C}$ scores, the conclusion that the SRR and AR are both significantly better than SW is somewhat surprising.

We have observed that the three methods generated quite similar rankings with the performance information on all the datasets used (Table 1). However, if we compare the rankings generated using the leave-one-out procedure, we observe that the number of differently assigned ranks is not negligible. In a total of 96 assigned ranks, there are 33 differences between AR and SRR, 8 between SRR and SW, and 27 between SW and AR.

Next, we analyze the ranking methods according to how well they exploit the available information and present some considerations concerning sensitivity and robustness.

Exploitation of Information. The aggregation methods underlying both SRR and AR exploit to some degree the magnitude of the difference in performance of the algorithms. The ratios used by the method SRR indicate not only which algorithm performs better, but also exploit the magnitude of the difference. To a smaller extent, the difference in ranks used in the AR method, does the same thing. However, in SW, the method is restricted to whether the algorithms have different performance or not, therefore exploiting no information about the magnitude of the difference. Therefore, it seems that methods that exploit more information generate better rankings.

Sensitivity to the Significance of Differences. One potential drawback of the AR method is that it is based on rankings which may be quite meaningless. Two algorithms j and k may have different error rates, thus being assigned different ranks, despite the fact that the error rates may differ only slightly. If we were to conduct a significance test on the difference of two averages, it could show they are *not* significantly different.

With the SRR method the ratio of the success rates of two algorithms which are not significantly different is close to 1, thus, we expect that this problem has small impact. The same problem should not happen with SW, although the statistical tests on which it is based are liable to commit errors [7].

The results obtained indicate that none of the methods seem to be influenced by this problem. However, it should be noted that the $\bar{C}$ measure used to evaluate the ranking methods equally does not take the significance of the differences into account, although, as was shown in [17], the problem does not seem to affect the overall outcome.

Robustness. Taking the magnitude of the difference in performance of two algorithms into account makes SRR liable to be affected by outliers, i.e. datasets where the algorithms have unusual error rates. We, thus, expect this method to be sensitive to small differences in the pool of the training datasets. Consider, for example, algorithm `ltree` on the `glass` dataset. The error rate obtained by `ltree` is higher than usual. As expected, the inclusion of this dataset affects the rankings generated by the method, namely, the relative positions of `ltree` and `c5boost` are swapped.

This sensitivity does not seem to significantly affect the rankings generated, however. We observe that identical rankings were generated by SRR in 13 experiments of the leave-one-out procedure. In the remaining 3, the positions of two algorithms (`ltree` and `c5boost`) were interchanged. Contrary to what could be expected, the other two methods show an apparently less stable behavior: AR has 4 variations on 4 datasets and SW has 13 across 5 datasets.

6 Related Work

The interest in the problem of algorithm selection based on past performance is growing[6]. Most recent approaches exploited *Meta-knowledge* concerning the performance of algorithms. This knowledge can be either theoretical or of experimental origin, or a mixture of both. The rules described by Brodley [5] captured the knowledge of experts concerning the applicability of certain classification algorithms. Most often, the meta-knowledge is of experimental origin [1, 4, 10, 11, 18]. In the analysis of the results of project StatLog [12], the objective of the meta-knowledge is to capture certain relationships between the measured dataset characteristics (such as the number of attributes and cases, skew, etc.) and the performance of the algorithms. This knowledge was obtained by *meta-learning* on past performance information of the algorithms. In [4] the meta-learning algorithm used was c4.5. In [10] several meta-learning algorithms were used and evaluated, including rule models generated with c4.5, IBL, regression and piecewise linear models. In [11] the authors used IBL and in [18], an ILP framework was applied.

[6] Recently, an ESPRIT project, METAL, involving several research groups and companies has started (`http://www.cs.bris.ac.uk/~cgc/METAL`).

7 Conclusions and Future Work

We have presented three methods to generate rankings of classification algorithms based on their past performance. We have also evaluated and compared them. Unexpectedly, the statistical tests have shown that the methods have different performance and that SRR and AR are better than SW.

The evaluation of the scores obtained does not allow us to conclude that the ranking methods produce satisfactory results. One possibility is to use the statistical properties of Spearman's correlation coefficient to assess the quality of those results. This issue should be further investigated.

The algorithms and datasets used in this study were selected according to no particular criterion. We expect that, in particular, the small number of datasets used has contributed to the sensitivity to outliers observed. We are planning to extend this work to other datasets and algorithms.

Several improvements can be made to the ranking methods presented. In particular paired t tests, which are used in SW, have been shown to be inadequate for pairwise comparisons of classification algorithms [7].

Also, the evaluation measure needs further investigation. One important issue is the difference in importance between higher and lower ranks into account, which is addressed by the Average Weighted Correlation measure [16, 17].

The fact that some particular classification algorithm is generally better than another on a given dataset, does not guarantee that the same relationship holds on a new dataset in question. Hence datasets need to be characterized and some metric adopted when generalizing from past results to new situations. One possibility is to use an instance based/nearest neighbor metric to determine a subset of relevant datasets that should be taken into account, following the approach described in [10]. This opinion is consistent with the NFL theorem [19] which implies that there may be *subsets of all possible applications* where the the same ranking of algorithms holds.

In the work presented here, we have concentrated on accuracy. Recently we have extended this study to two criteria — accuracy and time — with rather promising results [16]. Other important evaluation criteria that could be considered are the simplicity of its use [12] and also some knowledge-related criteria, like novelty, usefulness and understandability [8].

Acknowledgements We would like to thank Joaquim Costa, Joerg Keller, Reza Nakhaeizadeh and Iain Paterson for useful discussions, and the reviewers for their comments. Also 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 PRAXIS XXI project ECO, ESPRIT project METAL, Ministry of Science and Technology (plurianual support) 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] 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.

[6] W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van Den Bosch. TiMBL: Tilburg memory based learner. Technical Report 99-01, ILK, 1999.

[7] T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998. ftp://ftp.cs.orst.edu/pub/tgd/papers/nc-stats.ps.gz.

[8] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 1, pages 1–34. AAAI Press/The MIT Press, 1996.

[9] J. Gama. Probabilistic linear tree. In D. Fisher, editor, *Proceedings of the 14th International Machine Learning Conference (ICML97)*, Morgan Kaufmann, 1997.

[10] 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.

[11] 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.

[12] D. Michie, D.J. Spiegelhalter, and C.C. Taylor. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, 1994.

[13] H.R. Neave and P.L. Worthington. *Distribution-Free Tests.* Routledge, 1992.

[14] F. Provost and D. Jensen. Evaluating knowledge discovery and data mining. Tutorial Notes, Fourth International Conference on Knowledge Discovery and Data Mining, 1998.

[15] R. Quinlan. *C5.0: An Informal Tutorial.* RuleQuest, 1998.

[16] 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.

[17] C. Soares, P. Brazdil, and J. Costa. Measures to compare rankings of classification algorithms. In *Proceedings of the 7th IFCS*, 2000.

[18] L. Todorovski and S. Dzeroski. Experiments in meta-level learning with ILP. In *Proceedings of PKDD99*, 1999.

[19] D.H. Wolpert and W.G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, The Santa Fe Institute, 1996.