Predictive Performance of Weighted Relative Accuracy

Ljupčo Todorovski[1], Peter Flach[2], and Nada Lavrač[1]

[1] Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
{Ljupco.Todorovski, Nada.Lavrac}@ijs.si
[2] Department of Computer Science, University of Bristol
The Merchant Venturers Building, Woodland Road, Bristol, UK
Peter.Flach@cs.bris.ac.uk

Abstract. Weighted relative accuracy was proposed in [4] as an alternative to classification accuracy typically used in inductive rule learners. Weighted relative accuracy takes into account the improvement of the accuracy relative to the default rule (i.e., the rule stating that the same class should be assigned to all examples), and also explicitly incorporates the generality of a rule (i.e., the number of examples covered). In order to measure the predictive performance of weighted relative accuracy, we implemented it in the rule induction algorithm CN2. Our main results are that weighted relative accuracy dramatically reduces the size of the rule sets induced with CN2 (on average by a factor 9 on the 23 datasets we used), at the expense of only a small average drop in classification accuracy.

1 Introduction

In a recent study of different rule evaluation measures in inductive machine learning [4], a new measure, named weighted relative accuracy (*WRAcc*), was proposed. This measure can be understood from different perspectives. On one hand, *WRAcc* takes into account the improvement of the accuracy relative to the default rule (i.e., the rule stating that the same class should be assigned to all examples), and also explicitly incorporates the generality of a rule (i.e., the number of examples covered). Secondly, it can be seen as a single measure trading off several accuracy-like measures such as precision and recall in information retrieval, or sensitivity and specificity as used in medical applications. Finally, we showed that weighted relative accuracy is equivalent to the novelty measure used in a descriptive induction framework.

In this paper we further investigate the first perspective, i.e., whether weighted relative accuracy is a viable alternative to standard accuracy in rule learning. We experimentally tested the hypothesis that *WRAcc* directs the learner to fewer and more general rules (because it explicitly trades off generality and accuracy) without sacrificing too much on accuracy (because it will concentrate on the classes with high default accuracy). To this end, we implemented weighted

D.A. Zighed, J. Komorowski, and J. Żytkow (Eds.): PKDD 2000, LNAI 1910, pp. 255–264, 2000.

relative accuracy measure in the rule induction algorithm CN2 [2]. The original version of CN2 uses classification accuracy as a rule evaluation measure. In order to compare the predictive performance of accuracy and weighted relative accuracy, we performed experiments in twenty-three classification data sets. The results show a dramatic decrease in the number of induced rules, with only a relatively small drop in classification accuracy.

The paper is organised as follows. In Section 2 the weighted relative accuracy measure is defined. CN2 rule induction algorithm along with the changes made to incorporate weighted relative accuracy is presented in Section 3. The results of an experimental comparison of CN2 and CN2 with weighted relative accuracy on twenty-three data sets are presented and discussed in Section 4. The final Section 5, summarises the experimental results and proposes possible directions for further work.

2 Weighted Relative Accuracy

Although weighted relative accuracy can also be meaningfully applied in a descriptive induction framework, in this paper we restrict attention to classification-oriented predictive induction. Classification rules are of the form $H \leftarrow B$, where B is a condition and H is a class assignment to instances satisfying the condition. In propositional predictive rules, B is (typically) a conjunction of attribute-value pairs, and H is a class assignment. In first-order learning, frequently referred to as *inductive logic programming*, predictive rules are Prolog clauses, where H is a single positive literal and B is a conjunction of positive and/or negative literals. The difference with propositional predictive rules is that first-order rules contain variables that are shared between literals and between H and B. In this paper we restrict attention to propositional rules.

We use the following notation. $n(B)$ stands for the number of instances covered by a rule $H \leftarrow B$, $n(H)$ stands for the number of examples of class H, and $n(HB)$ stands for the number of correctly classified examples (true positives). We use $p(HB)$ etc. for the corresponding probabilities. We then have that rule accuracy can be expressed as $Acc(H \leftarrow B) = p(H|B) = \frac{p(HB)}{p(B)}$. Weighted relative accuracy is defined as follows.

Definition 1 (Weighted relative accuracy).

$$WRAcc(H \leftarrow B) = p(B)(p(H|B) - p(H)).$$

Weighted relative accuracy consists of two components: relative accuracy $p(H|B) - p(H)$ and generality $p(B)$. Relative accuracy is the accuracy gain relative to the fixed rule $H \leftarrow true$. The latter rule predicts all instances to satisfy H; a rule is only interesting if it improves upon this 'default' accuracy. Another way of viewing relative accuracy is that it measures the utility of connecting body B with a given head H. However, it is easy to obtain high relative accuracy with highly specific rules, i.e., rules with low generality $p(B)$. To this end, generality is used as a 'weight', so that weighted relative accuracy trades off generality and relative accuracy.

3 Rule Induction Algorithm CN2

CN2 [2,1] is an algorithm for inducing propositional classification rules. The list of induced classification rules can be ordered or unordered. Ordered rules are interpreted in a straight-forward manner: when classifying a new example, the rules are sequentially tried and the first rule that covers the example is used for prediction. In unordered case, all the rules are tried and predictions of those that cover the example are collected. A voting mechanism is used to obtain the final prediction. CN2 consists of two main procedures: search procedure that performs beam search in order to find a single rule and control procedure that repeatedly execute the search.

The search procedure performs beam search using classification accuracy of the rule as a heuristic function. The accuracy of the propositional classification rule `if` *cond* `then` c is equal to the conditional probability of class c, given that the condition *cond* is satisfied:

$$Acc(\texttt{if}\ cond\ \texttt{then}\ c) = p(c|cond).$$

We replaced the accuracy measure with the weighted relative accuracy (see Definition 1):

$$WRAcc(\texttt{if}\ cond\ \texttt{then}\ c) = p(cond)(p(c|cond) - p(c)), \tag{1}$$

where $p(cond)$ is the generality of the rule and $p(c)$ is the prior probability of class c. The second term in Equation 1 is the accuracy gain relative to the "default" rule `if` *true* `then` c. Weighted relative accuracy measure trades off this relative accuracy gain with generality of the rule $p(cond)$.

Different probability estimates, like Laplace [1] or m-estimate [3], can be used in CN2 for calculating the conditional probability $p(c|cond)$. Each of them can be also used for calculating the same term in the formula for weighted relative accuracy.

Additionally, CN2 can apply significance test to the induced rule. The rule is considered to be significant, if it locates regularity unlikely to have occurred by chance. To test significance, CN2 uses likelihood ratio statistic [2] that measures the difference between the class probability distribution in the set of examples covered by the rule and the class probability distribution in the set of all training examples. Empirical evaluation in [1] shows that applying a significance test reduces the number of induced rules and also slightly reduces the predictive accuracy.

Two different control procedures are used in CN2: one for inducing ordered list of rules and the other for unordered case. When inducing an ordered list of rules, the search procedure looks for the best rule, according to the heuristic measure, in the current set of training examples. The rule predicts the most frequent class in the set of examples, covered by the induced rule. Before starting another search procedure, all examples covered by the induced rule are removed. The control procedure invokes new search, until all the examples are covered (removed).

In unordered case, the control procedure is iterated, inducing rules for each class in turn. For each induced rule, only covered examples belonging to that class are removed, instead of removing all covered examples, like in ordered case. The negative training examples (i.e., examples that belong to other classes) remain and positives are removed in order to prevent CN2 finding the same rule again. The distribution of covered training examples among classes is attached to each rule. When classifying a new example, all rules are tried and those covering the example are collected. If a clash occurs (several rules with different class predictions cover the example), the class distributions attached to the rules are summed to find the most probable class.

Our modification of CN2 affects the heuristic function only. Modified CN2 uses weighted relative accuracy as search heuristic. Everything else remain the same: in particular, the quality of learned rule sets is still evaluated by means of accuracy on a test set. Our experiments, described in the next section, demonstrate that this simple change results in a dramatic decrease of the number of induced rules, at the expense of (on average) a small drop in accuracy.

4 Experiments

We performed experiments on a collection of twenty-one domains from the UCI Repository of Machine Learning Databases and Domain Theories [6] and two data sets originating from mutagenesis domain [5]. These domains have been widely used in other comparative studies. The domains properties (number of examples, number of discrete and continuous attributes and class distribution) are given in Table 1.

Performance of the rule inducing algorithms were measured using 10-fold stratified cross validation. Cross validation is repeated 10 times using a different random reordering of the examples in the data set. The same set of re-orderings were used for all experiments. The average and standard deviation (over the ten cross validations) of the classification error on unseen examples are reported.

Two series of experiments with both versions of CN2 algorithm (unordered and ordered one) were performed. In both cases, we compared CN2 using accuracy measure (CN2-acc) and CN2 with significance test applied to the induced rules (CN2-acc-99) with CN2 using weighted relative accuracy as a search heuristic (CN2-wracc). The experimental results are presented and discussed in the following two subsections.

4.1 Unordered Rules

The classification errors of unordered rules induced with CN2-acc, CN2-acc-99 and CN2-wracc are presented in Table 2. The statistical significance of the differences is tested using paired t-tests with significance level of 99%: $+/-$ to the left of a figure in CN2-acc or CN2-acc-99 columns means that CN2-wracc is significantly better/worse than CN2-acc or CN2-acc-99, respectively.

Table 1. Properties of the experimental data sets

Data set	#exs	#da	#ca	Class distribution
australian	690	8	6	56:44
balance	625	0	4	08:46:46
breast-w	699	9	0	66:34
bridges-td	102	4	3	15:85
car	1728	6	0	22:04:70:04
chess	3196	36	0	48:52
diabetes	768	0	8	65:35
echo	131	1	5	67:33
german	1000	13	7	70:30
glass	214	0	9	33:36:08:06:04:14
heart	270	6	7	56:44
hepatitis	155	13	6	21:79
hypothyroid	3163	18	7	05:95
image	2310	0	19	14:14:14:14:14:14:14
ionosphere	351	0	34	36:64
iris	150	0	4	33:33:33
mutagen	188	57	2	34:66
mutagen-f	188	57	0	34:66
soya	683	35	0	02:13:06:03:03:13:06:03:02:02:03:03:13:01:03:13:03:03:03
tic-tac-toe	958	9	0	35:65
vote	435	16	0	61:39
waveform	5000	0	21	33:33:34
wine	178	0	13	33:40:27

The overall predictive accuracy of CN2-wracc is smaller than the one of CN2-acc, the difference being almost 5%. CN2-wracc is significantly better in four domains, but also significantly worse in eleven out of twenty-three domains. Experiments with CN2-acc-99 confirms the results of previous empirical study in [1]: applying a significance test reduce the predictive accuracy of CN2. However, the overall predictive accuracy of CN2-wracc is still 3% smaller than the one of CN2-acc-99.

One of the bad scenarios for weighted relative accuracy is to have skewed class probability distribution. Namely, when using the *WRAcc* measure, the bigger the set of covered examples, the better is the rule. It is easier to achieve big coverage for the rules predicting the majority class, than for the rules predicting the minority classes. There is yet another handicap of *WRAcc* for rules predicting minority classes: these rules can have poor accuracy, which is still good relative to the prior probability of the minority class. Therefore, the rules (induced using CN2-wracc) predicting the minority classes tend to have very "impure" class probability distribution of the covered examples (i.e., they tend to be inaccurate). The problem with minority classes is also obvious from the empirical results. There are six domains with at least one class having prior probability below

Table 2. Classification errors (in %) of unordered rules induced with CN2-acc, CN2-acc-99 and CN2-wracc

Data set	**CN2-acc**	**CN2-acc-99**	**CN2-wracc**
australian	+ 17.41 ±0.50	+ 17.91 ±0.85	14.78 ±0.33
balance	- 19.95 ±0.91	- 22.83 ±1.49	28.43 ±1.50
breast-w	- 6.43 ±0.42	- 7.17 ±0.73	10.11 ±0.64
bridges-td	- 14.68 ±1.11	- 16.02 ±1.46	20.03 ±2.48
car	- 5.01 ±0.57	- 7.82 ±0.56	30.05 ±0.22
chess	- 1.58 ±0.25	- 3.25 ±0.20	5.91 ±0.01
diabetes	25.89 ±0.52	26.91 ±0.90	26.22 ±0.49
echo	33.46 ±1.86	- 31.91 ±1.94	35.24 ±2.78
german	- 26.59 ±0.59	28.34 ±0.93	28.69 ±0.91
glass	- 32.91 ±1.73	35.19 ±2.40	35.22 ±1.39
heart	22.46 ±1.40	+ 25.64 ±1.94	23.09 ±1.20
hepatitis	19.59 ±1.16	20.23 ±1.80	18.14 ±1.19
hypothyroid	1.42 ±0.25	1.53 ±0.23	1.39 ±0.10
image	+ 14.05 ±0.46	+ 14.62 ±0.52	9.32 ±0.51
ionosphere	9.29 ±0.92	10.26 ±0.88	9.90 ±0.20
iris	+ 7.48 ±0.82	+ 9.54 ±1.86	5.68 ±0.72
mutagen	- 17.13 ±1.69	18.85 ±0.86	19.03 ±0.66
mutagen-f	- 15.22 ±1.70	+ 25.35 ±1.55	20.86 ±1.78
soya	- 8.13 ±0.36	- 10.71 ±0.50	46.29 ±0.67
tic-tac-toe	- 1.49 ±0.17	- 1.98 ±0.19	26.40 ±0.65
vote	4.74 ±0.40	+ 5.79 ±0.56	4.38 ±0.01
waveform	+ 30.81 ±0.31	+ 30.39 ±0.48	27.17 ±0.35
wine	6.95 ±1.51	7.00 ±1.93	6.97 ±0.61
Average	**14.90** ±0.31	**16.49** ±1.08	**19.71** ±0.84

20%: balance, bridges-td, car, glass, hypothyroid and soya. CN2-wracc performs significantly worse than CN2-acc in five and significantly worse than CN2-acc-99 in four of them. CN2-wracc performs slightly (and insignificantly) better only in hypothyroid domain. If these six domains are discarded and the averages are calculated again, the overall predictive accuracy of CN2-wracc is only 1.5% smaller than the overall accuracy of CN2-acc and 0.5% better than the overall accuracy of CN2-acc-99.

On the other hand, there are seven domains with almost uniform class distributions: australian, chess, heart, image, iris, waveform and wine. CN2-wracc significantly outperforms CN2-acc in four and CN2-acc-99 in five domains out of seven. The differences in the heart and wine domains are small and insignificant, but CN2-wracc performs significantly worse in the chess domain.

The reason for being worse in the chess domain can be in the search strategy used in CN2. When searching for the condition of the rule, CN2 starts with empty condition, specialising it with conjunctively adding the literals of the form $(A_d = v)$ for a discrete attribute A_d and $(v_1 \leq A_c \leq v_2)$ for a continuous attribute A_c. Several best conditions (the number of them provided by the beam size) are kept in the beam. Intuitively, the number of examples covered drops

Table 3. Number of unordered rules induced with CN2-acc, CN2-acc-99 and CN2-wracc

Data set	**CN2-acc**	**CN2-acc-99**	**CN2-wracc**
australian	35.51 ±0.72	13.28 ±0.63	2.00 ±0.00
balance	106.27 ±1.08	40.85 ±2.21	6.85 ±0.15
breast-w	37.89 ±0.74	14.30 ±0.23	5.66 ±0.12
bridges-td	15.35 ±0.45	2.73 ±0.22	3.05 ±0.08
car	120.13 ±0.69	95.05 ±0.77	6.69 ±0.13
chess	29.47 ±0.32	16.61 ±0.21	5.00 ±0.00
diabetes	46.18 ±1.43	16.39 ±0.75	2.95 ±0.11
echo	19.48 ±0.31	5.51 ±0.28	3.60 ±0.18
german	83.14 ±1.22	21.52 ±0.75	3.96 ±0.07
glass	22.40 ±0.48	15.19 ±0.40	7.45 ±0.16
heart	21.90 ±0.46	8.48 ±0.52	3.01 ±0.10
hepatitis	17.48 ±0.46	4.44 ±0.64	2.73 ±0.37
hypothyroid	23.94 ±1.20	10.65 ±0.72	2.00 ±0.00
image	39.22 ±0.72	33.21 ±0.70	7.03 ±0.05
ionosphere	17.32 ±0.19	8.71 ±0.23	3.00 ±0.00
iris	6.75 ±0.22	3.82 ±0.06	3.00 ±0.00
mutagen	16.78 ±0.35	5.28 ±0.16	3.70 ±0.12
mutagen-f	25.12 ±0.30	7.53 ±0.19	3.78 ±0.12
soya	37.70 ±0.25	35.27 ±0.23	18.20 ±0.00
tic-tac-toe	28.33 ±0.77	22.08 ±0.43	7.23 ±0.12
vote	18.41 ±0.32	7.70 ±0.15	2.00 ±0.00
waveform	208.66 ±3.35	84.39 ±2.82	3.09 ±0.09
wine	8.28 ±0.18	5.89 ±0.09	3.81 ±0.09
Average	**42.86** ±0.70	**20.82** ±0.58	**4.77** ±0.09

quicker when adding a condition involving discrete attribute when compared to a condition involving continuous attribute. Therefore, conditions involving continuous attributes can be expected to be more "WRAcc friendly".

This intuition can be empirically confirmed in seven domains consisting of discrete attributes only: breast-w, car, chess, mutagen-f, soya, tic-tac-toe and vote. In six (five) of them CN2-acc (CN2-acc-99) performs significantly better than CN2-wracc. Consider also the mutagen and mutagen-f data sets: the only difference between them is that mutagen data set has two extra continuous attributes (logP and LUMO). CN2-acc achieves better accuracy for both data sets, the difference being bigger for the one without continuous attributes (mutagen-f).

Consider now the number of rules induced by CN2-acc, CN2-acc-99 and CN2-wracc in Table 3. The unordered rule lists induced with CN2-wracc are, on average, nine times smaller than the rule lists induced with CN2-acc. Removing insignificant rules has already been used for reducing the rule lists size in CN2. However, we can see from Table 3 that CN2-wracc outperforms CN2-acc-99 in terms of rule list size: rule lists induced with CN2-wracc are, on average, four times smaller. Note also that the reduction of number of rules does not cause

Table 4. Classification errors (in %) of ordered rules induced with CN2-acc, CN2-acc-99 and CN2-wracc

Data set	**CN2-acc**	**CN2-acc-99**	**CN2-wracc**
australian	+ 18.93 ±0.93	+ 18.29 ±1.20	15.12 ±0.42
balance	- 17.48 ±1.30	- 22.11 ±1.47	27.68 ±0.80
breast-w	- 5.46 ±0.51	- 5.41 ±0.49	6.75 ±0.93
bridges-td	- 19.14 ±3.32	- 16.30 ±1.38	23.01 ±2.33
car	- 3.59 ±0.52	- 4.24 ±0.54	18.73 ±0.30
chess	- 0.72 ±0.11	- 1.14 ±0.17	5.72 ±0.02
diabetes	+ 29.40 ±1.17	27.70 ±1.06	27.35 ±1.25
echo	38.58 ±3.40	39.57 ±3.72	35.58 ±4.13
german	29.55 ±1.48	- 29.45 ±0.79	30.39 ±0.24
glass	- 30.94 ±2.45	- 34.52 ±1.72	37.89 ±1.36
heart	24.05 ±1.60	+ 25.79 ±1.97	23.61 ±1.50
hepatitis	22.07 ±1.71	21.45 ±2.57	23.23 ±1.23
hypothyroid	- 1.50 ±0.18	- 1.65 ±0.20	2.94 ±0.15
image	- 3.67 ±0.25	- 4.18 ±0.27	9.78 ±0.16
ionosphere	12.71 ±1.45	+ 13.13 ±1.43	12.03 ±0.65
iris	6.41 ±0.64	- 5.14 ±1.09	6.21 ±0.71
mutagen	19.84 ±2.24	19.97 ±1.47	20.37 ±1.94
mutagen-f	- 18.05 ±1.92	+ 28.30 ±2.66	22.29 ±2.37
soya	- 9.38 ±0.58	- 11.52 ±0.55	49.28 ±0.21
tic-tac-toe	- 2.70 ±0.71	- 4.31 ±1.30	30.35 ±0.83
vote	5.38 ±0.79	+ 6.48 ±0.69	4.74 ±0.25
waveform	- 22.02 ±0.58	- 22.35 ±0.63	24.06 ±0.35
wine	+ 6.37 ±1.01	5.59 ±0.79	4.90 ±1.33
Average	**15.13** ±1.25	**16.03** ±1.22	**20.09** ±1.02

significant increase of rule length in terms of number of conditions in rule. The average length of rules induced with CN2-acc is 2.99 and with CN2-wracc is 3.51.

4.2 Ordered Rules

The classification errors of ordered rules induced with CN2-acc, CN2-acc-99 and CN2-wracc are presented in Table 4. The experimental results show that weighted relative accuracy is better suited towards the unordered version of the CN2 algorithm, than to the ordered version. The reason is that in each iteration of the ordered algorithm, the number of uncovered examples drops, so the coverage term in Equation 1 gets smaller. The small value of the coverage term diminishes the difference between "good" and "bad" rules. This is not the case with the covering algorithm used in unordered version of CN2, where negative covered examples are not removed.

The experimental results regarding accuracy of ordered rules more or less follow the pattern already seen in the results for unordered ones. The two notable exceptions are image and waveform domains, where the significant accuracy improvement (for unordered rules) obtained with *WRAcc* changed to significant accuracy deterioration.

Table 5. Number of ordered rules induced with CN2-acc, CN2-acc-99 and CN2-wracc

Data set	**CN2-acc**	**CN2-acc-99**	**CN2-wracc**
australian	34.99 ±0.84	17.43 ±1.18	2.72 ±0.11
balance	60.43 ±0.98	27.95 ±0.92	5.63 ±0.23
breast-w	30.23 ±0.46	12.87 ±0.42	11.32 ±0.19
bridges-td	9.81 ±0.35	2.55 ±0.33	2.87 ±0.07
car	51.61 ±0.89	37.58 ±0.57	9.18 ±0.38
chess	28.16 ±0.51	22.27 ±0.34	2.00 ±0.00
diabetes	49.48 ±0.56	19.65 ±1.01	4.22 ±0.24
echo	15.80 ±0.39	4.99 ±0.54	3.66 ±0.25
german	74.22 ±0.67	21.64 ±1.51	4.65 ±0.31
glass	16.23 ±0.27	10.36 ±0.28	4.00 ±0.00
heart	16.25 ±0.40	10.43 ±0.58	3.29 ±0.07
hepatitis	11.54 ±0.39	5.05 ±0.37	3.04 ±0.13
hypothyroid	20.28 ±0.61	12.92 ±0.75	2.00 ±0.00
image	29.05 ±0.37	23.89 ±0.52	8.00 ±0.00
ionosphere	10.03 ±0.32	8.74 ±0.25	2.90 ±0.09
iris	5.80 ±0.13	2.20 ±0.09	3.01 ±0.09
mutagen	11.78 ±0.39	6.02 ±0.34	3.03 ±0.13
mutagen-f	21.94 ±0.36	6.84 ±0.39	3.74 ±0.11
soya	28.19 ±0.40	21.58 ±0.16	6.00 ±0.00
tic-tac-toe	36.73 ±2.80	32.86 ±2.09	5.23 ±0.20
vote	17.89 ±0.19	7.43 ±0.23	5.17 ±0.11
waveform	175.92 ±1.32	168.41 ±3.02	6.16 ±0.13
wine	3.97 ±0.11	3.20 ±0.11	2.96 ±0.10
Average	**33.06** ±0.60	**21.17** ±0.70	**4.56** ±0.13

The comparison of the size of the ordered rule list induced with CN2-acc, CN2-acc-99 and CN2-wracc is presented in Table 5. Ordered rule lists induced with CN2-wracc are, on average, seven times smaller than the ones induced with CN2-acc and four times smaller than the ones induced with CN-acc-99.

5 Summary

In order to measure the predictive performance of a a new rule evaluation measure, weighted relative accuracy (*WRAcc*), we implemented it in the rule induction algorithm CN2. The empirical evaluation of CN2-wracc shows that *WRAcc* clearly outperforms ordinary accuracy measure used in CN2, when the size of the induced rule sets is considered. The rule sets induced with CN2-wracc are, on average, about nine times smaller in unordered case and seven times smaller in ordered case. The factor of nine (or seven) is actually huge, and can be regarded as a step towards higher comprehensibility of the rule lists induced with CN2.

Significance tests [1] of the rules have already been used in CN2 to reduce the size of the induced rule lists. However, the experiments show that the rule lists

induced using *WRAcc* are on average four times smaller than the ones induced using significance test in CN2.

The price to be paid for the reduction of the rule list size is an overall drop of the predictive classification accuracy of the induced rules by 5%. When compared to the predictive accuracy of rule lists induced using significance test, the drop is about 3.5%. The drop of the accuracy is mostly due to the domains with skewed class distributions, having classes with small number of examples. In seven experimental domains with almost uniform class distributions CN2-wracc significantly outperforms CN2 also in terms of predictive accuracy.

The predictive performance of weighted relative accuracy can be further evaluated by implementing it in other rule induction algorithms. In order to confirm the improvement of the comprehensibility of the rules induced with CN2-wracc, an expert evaluation of the rules should be obtained by a human domain experts. This would be also a step towards evaluating the descriptive performance of the *WRAcc* measure.

Acknowledgements

The work reported was supported in part by the Slovenian Ministry of Science and Technology, the Joint project with Central/Eastern Europe funded by the British Royal Society and by the EU-funded project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise (IST-1999-11495).

References

1. Clark, P. and Boswell, R. (1991) Rule induction with CN2: Some recent improvements. In *Proceedings of the Fifth European Working Session on Learning*: 151–163. Springer-Werlag.
2. Clark, P. and Niblett, T. (1989) The CN2 induction algorithm. *Machine Learning Journal*, 3(4): 261–283.
3. Džeroski, S., Cestnik, B. and Petrovski, I. (1993) Using the m-estimate in rule induction. Journal of Computing and Information Technology, 1(1):37 – 46.
4. Lavrač, N., Flach, P. and Zupan, B. (1999) Rule Evaluation Measures: A Unifying View. In *Proceedings of the Ninth International Workshop on Inductive Logic Programming, volume 1634 of Lecture Notes in Artificial Intelligence*: 74–185. Springer-Verlag.
5. Muggleton, S., Srinivasan, A., King R. and Sternberg, M. (1998) Biochemical knowledge discovery using Inductive Logic Programming. In Motoda, H. (editor) *Proceedings of the first Conference on Discovery Science*. Springer-Verlag.
6. Murphy, P. M. and Aha, D. W. (1994) *UCI repository of machine learning databases* [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.