Improving C4.5-rules by means of a partial-matching strategy*

J. Ranilla[1], O. Luaces[1], A. M. Peña[2], and A. Bahamonde[1]

[1] Centro de Inteligencia Artificial. Universidad de Oviedo en Gijón.
Campus de Viesques, E-33271 Gijón, Spain.
{ranilla,oluaces,antonio}@aic.uniovi.es
[2] Facultad de Ingeniería.Universidad Distrital Francisco José de Caldas.
Bogotá, Colombia.

Abstract. In this paper we present a new pruning mechanims for C4.5 decision trees. The idea is to select rules that are going to be applied by means of a minimal distance (or partial-matching) criterion. To illustrate its advantages, we have built an algorithm based on the skeleton of the C4.5-rules, though including several modifications to induce partial-matching rules. The modifications consist in replacing the MDL-based method with a pruning process whose performance relies on an estimation of the quality of the rules. Empirical results show that, in general, inducing partial-matching rules yields more compact rule sets without degrading performance, no matter which estimation is used. If this estimation is done by means of the impurity level or the Laplace correction, our experiments show that both the accuracy and size of the rule sets are significantly improved.

1 Introduction

There are two possibilities when using a set of classification rules. Given a new case, to return a class label we can use the conclusion of the rule whose conditions are completely fulfilled by case values, or instead we can follow the rule whose conditions are the nearest to case values. Usually we refer to these procedures as **full** and **partial matching** respectively.

In this paper, we illustrate the advantages of partial matching. First, it should be mentioned that the kind of knowledge induced when we are planning to use distances is distinct. Learning algorithms must concentrate on selecting clusters of near training examples that belong to the same class. In contrast, when full matching is involved, learning algorithms must cover all the attribute space.

An advantage of partial matching is the possibility of adapting the decision areas of rules to regions with wavy frontiers. In general, although it is possible to find rule learners that obtain oblique rule conditions, such as OC1 [19], the geometry of Voronoi regions is much richer.

* The research reported in this paper has been supported in part under MCyT and Feder grant TIC2001-3579

Moreover, when we are supported by a distance criterion, we generally obtain less classification rules than when we require a full match to apply them. We can simply observe that full matching is a particular case of partial matching, where all minimal distances are zero. Therefore, the output of partial-matching learners seems to be more readable for human users, since it is generally acknowledged that a reduced size of rule sets improves comprehensibility.

We used a divide-and-conquer template to implement a family of partial-matching learners and discuss the scores obtained with a number of experiments carried out to compare their performance with respect to the accuracy and size of the rule sets induced.

The basic structure used is Quinlan's well-known and well-reputed C4.5-rules [21], which produces rules passing through decision trees. Breslow and Aha [4] present a framework to categorize different approaches for simplifying decision trees. On the basis of their framework, our work can be placed into two categories: algorithms that *modify the test search* by means of new selection measures; we used a number of different heuristics, such as the Laplace correction, Gini index [3], G index [9] or impurity level [7, 15, 16] instead of using Shannon's information-based measures.

The other category is composed of algorithms using *alternative data structures*, specifically rules, obtained by new pruning methods [15] applied to induced trees. Worth of mention are some previous works related to pruning, like Wilson and Martinez's [25] survey on reduction techniques for instance-based learners, including their original contribution, the family of DROPx algorithms. Smyth and McKenna [23] stressed the need to reach a trade-off between efficiency and competence, that is a reduced set of selected cases, to correctly solve problems in CBR-based approaches. Pfahringer [20] also suggests an alternative MDL-based formula to reduce the size of the rule sets produced by C4.5-rules without degrading performance.

We define the alternative measures for test selection used in this paper in Section 3, while Section 4 details the algorithm template based on these measures. To close the paper, we report the results found when comparing the partial-matching learners amongst themselves and with respect to their full matcher counterpart, C4.5-rules. The conclusion we reach is that simple heuristic measures give rise to quite acceptable learners with regard to the accuracy and size of rule sets. However, in order to achieve significant differences with respect to C4.5-rules, we must use heuristics like the Laplace correction to improve size with good accuracy scores, or our impurity level to significantly improve both the accuracy and size of rule sets.

2 Partial-matching rules

The materials that we will use throughout the paper are examples and rules; all are described by set of attributes whose values can be either numeric or symbolic. While the examples have a fixed number of attributes plus the class, rules have antecedents or conditions represented by a variable number of literals, and a

conclusion naming one class. Our rules look like those found by C4.5-rules; for instance,

$$R: \quad \text{class} \leftarrow (x \leq 5.6) \wedge (y > 89.34) \wedge (color = \text{blue}). \tag{1}$$

In partial-matching environments, a case is classified following a nearest-neighbor principle, so we need a function to compute distances between rules and cases. For this purpose we use a HEOM-like [24] metric defined as:

$$\text{distance}(R, c) := \sqrt{\sum_{a=1}^{m} \text{difference}_a^2(R_a, c_a)} \tag{2}$$

where m is the number of attributes describing the examples, R_a is the condition on attribute a in rule R and c_a is the actual value of attribute a in case c. For every attribute a, in turn, differences are calculated using the normalized Euclidean distance if a takes continuous values, or using the *overlap* function (likewise HEOM) if a is a symbolic attribute.

$$\text{difference}_a(R_a, c_a) = \begin{cases} \text{overlap}(R_a, c_a), & \text{if } a \text{ is symbolic} \\ \text{norm\_eucl}(R_a, c_a), & \text{if } a \text{ is numeric} \end{cases} \tag{3}$$

The *overlap* metric [24] yields a difference of 1 when the symbolic value of the attribute is different than the value mentioned in the condition of the rule, and 0 otherwise. For numerical attributes we use

$$\text{norm\_eucl}(R_a, c_a) = \begin{cases} 0, & \text{if } c_a \text{ fulfills } R_a \\ \frac{|c_a - value_a|}{4\sigma_a}, & \text{otherwise} \end{cases} \tag{4}$$

where R_a may be of the form "$a \leq value_a$", or "$a > value_a$", or "$a \in I$", and $value_a$ is the nearest border of interval I to c_a. Differences are normalized by means of a commonly used [24] large value: four times the standard deviation, σ_a, of the observed attribute values.

To completely specify the distance function we must define how to deal with missing values. Whenever R_a is missing means that no particular value of a is required to apply R, i.e. the value of a makes no difference. A missing c_a means that the value of a is unknown in case c. In both cases our difference function will return a value of 0 to make the value of a have no influence in the distance computation. RISE [10] deals with missing numerical values in the exact same way.

3 Purity measures

The core for building a partial-matching rules learner is a measure capable of testing whether a selected group of training examples is coherent enough to somehow become a classification rule. Usually, these measures are called *purity measures*, and they will act as a heuristic to build our partial-matching learners.

To define these measures formally, let us consider a subset of training examples E (the whole set is TS) and a class C.

For ease of reference, we call e^+ the number of examples in E of class C (positive examples), e^- the number of examples in E of a class different than C (negative examples), p the success probability, i.e. $p = \frac{e^+}{e^+ + e^-}$, n the number of examples in subset E, i.e. $n = e^+ + e^-$ and $\#c$ the number of classes in training set TS. Then the heuristic purity functions are the following:

Trivial	$T = p$;	Minimum	$M = \min\{p, 1-p\}$;
Difference	$D = e^+ - e^-$;	Gini index	$\text{Gini} = 1 - (p^2 + (1-p)^2)$;
Laplace	$\text{LAP} = \frac{e^+ + 1}{e^+ + e^- + \#c}$;	G index	$G = 2\sqrt{p(1-p)}$;

The most basic heuristic is the *trivial* one, which coincides with the success probability when we predict class C for all the examples in the subset E. The *difference* is just the balance of positive and negative examples in E when we are concluding class C too. This heuristic can be considered as a simplification of the *accuracy* function used by Muggleton [18] and by Fürnkranz and Widmer [12]; see [11] for details. The Laplace measure was used in CN2 [5] and RISE [10]. The Gini index is the heuristic of CART [3]. The G index was proposed by Diettrich et al. in [9] as an alternative to the information-based measures of C4.5. The *minimum* heuristic is also mentioned in this last paper.

The heuristic that achieves the best results is the impurity level [7, 15, 16]. It explicitly takes into account not only the success probability p, but also the difficulty of attaining that amount of examples of class C.

To define the impurity level we previously need to compute the confidence interval of p when predicting class C in subset TS. For this purpose we use the following expresion

$$\text{CI}(TS, C) = \left[\underbrace{\frac{p + \frac{z^2}{2n} - z\sqrt{\frac{p(1-p)}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}}_{\text{CI}_l(TS,C)}, \underbrace{\frac{p + \frac{z^2}{2n} + z\sqrt{\frac{p(1-p)}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}}_{\text{CI}_h(TS,C)} \right] \tag{5}$$

where n is the number of cases in TS and z is a constant obtained from a normal distribution table which depends on the confidence level used (by default 95%, hence z is 1.96). An analogous calculation is done for $CI(E, C)$. The impurity level of E is defined as the overlapping percentage of both confidence intervals,

$$\text{IL}(R) = \frac{\text{CI}_h(TS, C) - \text{CI}_l(E, C)}{\text{CI}_h(E, C) - \text{CI}_l(E, C)} \times 100 \tag{6}$$

This heuristic is based on Aha's mechanism in IB3 [1] for selecting a set of representative instances from a set of training examples.

4 The learner purity-rules

In this section, we describe the template used in the experiments reported in the next section. As previously stated, this is a sketch of C4.5-rules in which we

substitute the information gain ratio by a heuristic of those listed in the previous section. These heuristics are used to evaluate the quality of different pieces of induced knowledge and, in general, this quality is better with higher values of the heuristic (i.e. the gain ratio). However some other measures, likewise the impurity level, point to a better quality with lower values. Thus, for the purpose of simplification, we will use the term *purity* in the following, provided that a higher purity value means, in some cases, a lower value of the actual heuristic.

The learner that will be called **purity-rules** has two stages. In the first one, it builds a decision tree that will become a rule set at the end of the second stage. To build the tree, the algorithm follows a greedy process, trying in each step to discover the test about the values of an attribute that leads more directly to a decision about the class of the examples involved.

Thus, if E is a subset of training examples, and X is an attribute whose values in E split the set into subsets $(E_i : i = 1, \ldots, n)$, a measure of the convenience of the test concerning X is given by

$$\text{convenience}(E, X) = \sum_{i=1}^{n} \frac{|E_i|}{|E|} \cdot \text{purity}(E_i, X) \tag{7}$$

where *purity* is the gain ratio in C4.5; purity-rules, however, use any of the heuristics mentioned in Section 3.

Decision trees thus obtained are usually too big and complex. This yields an overfitting of training data and consequently leads to poor classification accuracy on unseen cases. Additionally, decision trees are not as intuitive and human-readable as classification rules [21]. Hence, C4.5-rules transforms the decision trees induced by C4.5 into pruned rule sets by means of the MDL (Minimum Description Length) principle. These induced rules are applied following a full-matching strategy.

Our purity-rules differs in the purity measure used in building the trees, as stated above, but the main difference from the master full-matching learner is in the second stage, the rule generation process. Here, we follow the steps of our previous systems: FAN [22], INNER [17], and BETS [8]. The idea is to explicitly use the guidelines of the heuristic measure. First, purity-rules tries to clean the antecedents or conditions list from each rule; this is called *qualification*. Then, the algorithm *selects* the most promising subset of classification rules. These processes have some similarities with the approach followed by IREP* [6], a modified version of Fürnkranz and Widmer's IREP [12], although using different quality estimators and stopping criteria. In the following we briefly describe our pruning processes.

The qualification tries to drop the redundant or unnecessary conditions. In order to reduce the number of possibilities to consider, we first order the conditions according to the purity heuristic. Thus, if purity returns higher values to better rules, we have

$$R: \quad \text{C} \leftarrow \text{cond}_1 \wedge \text{cond}_2 \wedge \ldots \wedge \text{cond}_n \tag{8}$$

$$\text{purity}(\text{C} \leftarrow \text{cond}_1) \geq \text{purity}(\text{C} \leftarrow \text{cond}_2) \geq \ldots \geq \text{purity}(\text{C} \leftarrow \text{cond}_n) \tag{9}$$

Once the conditions have been ordered, the process starts with an empty set of descriptions and progressively adds partial descriptions of the original rule being qualified. These descriptions are of the form:

$$R_i : C \leftarrow \text{cond}_1 \wedge \ldots \text{cond}_i; \quad i = 1 \ldots n \tag{10}$$

Only those descriptions with a success probability higher than an *acceptance threshold* are saved. Furthermore, if a partial description with no errors is found, no more descriptions are added.

Once a set of partial descriptions has been obtained, the best of them is selected, namely R_{best}; more partial descriptions are added by sequentially deleting each antecedent, from the penultimate to the first one. Only descriptions with higher success probability than R_{best} are added. The whole set of obtained descriptions is finally filtered, removing those with purity lower than 90% of the highest purity found.

Rule qualification is followed by a selection process aimed at reducing the total number of rules induced so far, since compact rule sets are more comprehensible and they usually provide more accurate generalizations.

The selection starts detecting and deleting those rules classifying too specific peculiarities of data caused by noisy examples. This procedure deletes rules whose success probability is lower than a noise threshold. Then, the algorithm determines the purity threshold that yields the better subset of rules, in terms of accuracy. At this stage rules compete to classify examples, being applied on the basis of a minimum distance criterion. The resulting subset is revised to eliminate useless rules still undeleted in prior steps. Each rule is considered useless if there is no accuracy loss when eliminated.

There is a final stage in the rule selection process related to the comprehensibility of the resulting rule set. Whenever all the attributes describing the training examples have symbolic values, the selection is allowed to include a *default rule* that will be applied when no other rule is at distance zero. In data sets with some continuous attributes, the default rule would destroy the benefits of the application by means of a minimal distance criterion, so purity-rules never includes a default rule in these cases.

5 Experimental results

In this section we present the scores reached by learners of the type purity-rules, which apply rules following a partial-matching strategy, in comparison with those obtained by C4.5-rules (release 8), which uses a full-matching strategy.

To carry out the experiments, we chose the Holte's [13] problems, a well-known set of 16 data sets downloaded from the UCI Machine Learning Repository [2]. Following the recommendations in [14], we used a 10-fold stratified cross validation repeated 5 times, ensuring that the algorithms were run on identical training and test sets.

We tested the accuracy (see Table 1) and size of the induced rule sets, distinguishing between the number of rules and the average size of these rules, i.e. the average number of antecedents (see Table 2).

Table 1. Average classification errors for each learner in Holte's data sets. The partial-matching learners are named in accordance with the purity measures (see Section 3) used to build them from the template describe in Section 4. The last row shows the average errors for all data sets.

data set	IL	LAP	T	C4.5-rules	D	Gini	M	G
BC	27.63	30.32	30.41	30.84	25.59	29.93	32.44	30.82
CH	2.10	1.93	1.40	0.97	9.57	1.21	1.50	1.30
G2	18.31	21.96	19.21	21.87	23.35	23.10	20.85	21.40
GL	30.57	32.97	30.63	32.22	36.43	32.12	38.63	35.43
HD	17.73	22.15	23.26	20.98	21.91	23.15	25.73	21.88
HE	19.33	15.96	20.87	20.43	17.37	23.19	19.12	18.95
HO	15.82	15.82	14.83	17.45	14.68	19.34	19.13	26.07
HY	0.99	1.30	1.54	0.78	1.08	0.89	1.52	0.95
IR	5.07	5.33	5.33	4.40	7.33	5.47	5.33	5.73
LA	17.80	20.67	18.20	17.00	17.13	22.67	23.00	25.80
LY	22.85	21.09	23.62	23.25	24.60	25.57	28.42	27.43
MU	1.52	0.32	0.55	0.03	4.42	0.57	0.57	0.57
SE	2.25	2.29	2.61	2.35	5.39	2.49	2.57	2.54
SO	0.00	0.00	1.60	2.90	0.00	2.90	2.10	2.90
VO	4.78	4.41	4.74	4.37	4.73	5.80	5.52	5.47
V1	10.71	10.48	9.92	10.16	11.95	10.43	10.15	9.33
Av.	12.34	12.94	13.05	13.13	14.10	14.30	14.79	14.79

The scores shown in Table 1 reflect quite similar learner behavior with respect to data sets, though with different final results. The correlations between the error columns are very high, with an average of 0.98 and a standard deviation of 0.02, and with a minimum of 0.92 between D and G purity measures. Another surprising issue is the difference in accuracy between C4.5-rules (13.13) and trivial-rules (13.05), built with the simple heuristic given by the success probability.

The scores obtained with respect to the size of the induced rule set exhibit quite different behavior among learners; with respect to the number of rules, the average of correlations is 0.61 with a standard deviation of 0.32, which indicates important differences between some learners. Worth of mention are the scores obtained by D-rules; said learner achieves a very small size of induced knowledge, but the price is a lower degree of accuracy: 14.10, almost one point up on C4.5-rules. The best balance is clearly achieved by the impurity level.

Respect to the percentage of cases classified by rules at distance greater than zero (uncovered cases), the average for the whole data sets is about 10% for each purity measure. Obviously, the percentage of uncovered cases mostly depends on the problem itself, so we have noticed large differences among problems. For example, there is an average percentage of 21.41% of uncovered cases for the HO problem and the impurity-rules, while there is only a 0.74% for the CH problem

Table 2. This table shows the average number of rules and antecedents for each learner in Holte's data sets.

data	IL		LAP		T		C4.5-rules		D		Gini		M		G	
set	rul.	ants.	rul.	ants.	rul.	ants.	rul.	ants.	rul.	ants.	rul.	ants.	rul.	ants.	rul.	ants.
BC	4.6	7.7	7.6	14.4	21.4	53.5	8.2	17.1	3.4	4.3	23.5	66.7	23.5	69.4	22.5	63.6
CH	9.8	36.4	18.4	85.0	23.9	103.1	26.8	100.0	3.0	4.0	26.5	113.3	23.7	110.3	26.0	114.9
G2	6.2	11.4	6.7	13.4	9.7	18.4	8.1	21.0	2.3	3.9	13.8	37.9	12.5	27.5	13.2	35.1
GL	8.7	26.4	6.8	21.3	12.2	34.9	14.1	50.8	6.5	18.0	28.0	87.6	30.8	76.4	30.7	97.6
HD	7.1	17.5	11.5	23.5	19.3	53.6	13.3	35.8	2.1	4.0	23.3	76.3	22.0	72.1	24.1	81.8
HE	5.6	15.3	3.4	8.3	3.7	7.3	7.9	20.5	3.4	6.9	4.6	10.5	5.1	11.4	5.4	12.6
HO	3.8	10.1	4.6	9.1	6.4	15.2	6.0	11.4	2.0	3.0	5.4	15.1	6.7	18.1	4.7	11.9
HY	4.6	10.6	4.2	10.6	2.8	6.5	6.3	13.1	6.8	16.3	3.8	10.6	4.2	14.4	7.8	22.8
IR	3.3	4.2	3.1	3.2	4.8	6.9	4.0	6.1	3.7	3.9	4.1	5.7	3.8	5.4	4.6	6.9
LA	3.6	7.2	2.6	4.4	2.7	5.0	4.0	5.8	2.0	2.3	2.4	6.4	2.5	6.7	1.9	4.8
LY	7.8	14.9	7.4	13.8	17.0	42.8	10.6	23.6	4.1	7.4	16.5	42.6	22.5	63.6	16.9	43.1
MU	4.7	3.9	8.1	11.8	8.9	11.9	17.7	26.4	3.9	4.2	9.0	16.0	9.0	16.0	9.0	16.0
SE	4.5	14.0	3.0	6.1	4.1	12.8	12.7	41.7	3.1	8.7	4.2	11.5	6.0	23.5	18.8	80.9
SO	4.0	4.0	4.0	4.0	4.0	3.9	4.0	5.9	4.0	4.0	4.2	4.5	4.2	4.4	4.2	4.5
VO	2.3	4.3	4.3	8.1	4.8	12.2	6.1	13.8	2.4	2.9	5.3	12.8	4.4	13.7	5.5	14.4
V1	4.3	11.4	6.8	18.4	9.5	29.2	11.2	29.3	3.4	3.9	9.2	29.7	8.5	27.6	10.5	34.1
Av.	5.3	12.5	6.4	16.0	9.7	26.1	10.1	26.4	3.5	6.1	11.5	34.2	11.8	35.0	12.9	40.3

and the same purity measure. Due to the lack of space we can not show a full comparison table in this paper.

To appreciate the significance of the aforementioned scores, we elaborated Tables 3A and 3B with the results of one-tail paired t-tests. In these comparisons, we can observe that the first place in accuracy is obtained by the impurity level. The difference with respect to LAP-rules and T-rules is not statistically significant, although the significance with the latter is in the borderline (94.65%). We can also see that there are two other groups of measures which present no significant differences among themselves. The former is composed of LAP, T, C4.5-rules and D and the latter by D, Gini and M.

With respect to size considerations, we observe that D-rules presents an obvious advantage with respect to all other learners, but suffers from a decrease in accuracy. In the group of more accurate learners, LAP and impurity level produce smaller rule sets (with no significant differences between them), followed by T and then the group composed of C4.5-rules, Gini, M and G, with no significant differences among them. To save space we do not show the table corresponding to the t-test for antecedents but the results are very similar to those for number of rules, except that the difference among C4.5-rules and G-rules is significant at 99.31% for antecedents.

6 Concluding remarks

We have presented a family of rule learners whose application is carried out according to a partial-matching mechanism based on minimal distance from rule conditions and case values. All the learners have a common template based on Quinlan's algorithm C4.5-rules; instead of using the information gain ratio to

Table 3. Significant levels of differences found with one-tail paired t-tests. The label n.s., for non-significant, means that the level found is below 95%. The first table refers to accuracy while the second considers the number of rules.

	Avg.	IL	LAP	T	C4.5-rules	D	Gini	M
IL	12.34	···	···	···	···	···	···	···
LAP	12.94	n.s.	···	···	···	···	···	···
T	13.05	n.s.	n.s.	···	···	···	···	···
C4.5-rules	13.13	95.65%	n.s.	n.s.	···	···	···	···
D	14.10	98.69%	n.s.	n.s.	n.s.	···	···	···
Gini	14.30	99.89%	98.70%	99.44%	99.54%	n.s.	···	···
M	14.79	99.71%	99.68%	99.24%	99.09%	n.s.	n.s.	···
G	14.79	99.47%	98.46%	96.64%	97.60%	n.s.	n.s.	n.s.

A) Accuracy

	Avg.	D	IL	LAP	T	C4.5-rules	Gini	M
D	3.50	···	···	···	···	···	···	···
IL	5.31	99.74%	···	···	···	···	···	···
LAP	6.41	99.12%	n.s.	···	···	···	···	···
T	9.71	99.80%	99.60%	99.70%	···	···	···	···
C4.5-rules	10.06	99.97%	99.95%	99.98%	n.s.	···	···	···
Gini	11.48	99.85%	99.74%	99.64%	95.02%	n.s.	···	···
M	11.85	99.88%	99.80%	99.57%	95.54%	n.s.	n.s.	···
G	12.87	99.96%	99.94%	99.88%	98.01%	n.s.	n.s.	n.s.

B) Number of rules

construct decision trees, our learners use different purity measures. The same measure is actively used for pruning the tree to obtain a rule set.

The purity measures used can be described as heuristics capable of quantifying the classification quality of a rule. They range from extremely simple, such as the success probability (called T, for trivial) or the difference (D) between the number of positive and negative examples, to slightly more complex heuristics, such as the Laplace correction, the Gini index or the impurity level. However, all our purity measures can be computed with simple arithmetic expressions.

A total of seven learners thus built were compared, together with C4.5-rules, with regard to their accuracy and the size of their induced rule sets. The scores show that the partial-matching learner built with the assistance of the impurity level gives rise to the best results in accuracy and in size measures, if we exclude the learner generated by D, which produces very small rule sets, but with a substantially inferior degree of accuracy.

References

[1] D. Aha. *A Study of Instance-based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, and Psychological Evaluations*. PhD thesis, University of California at Irvine, 1990.

[2] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[3] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984. Belmont, CA.

[4] L. A. Breslow and D. W. Aha. Simplifying decision trees: a survey. *Knowledge Engineering Review*, 12(1):1–40, 1997.
[5] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
[6] W. Cohen. Fast effective rule induction. In *International Conference on Machine Learning*, pages 115–123, 1995.
[7] J. del Coz, O. Luaces, J.R. Quevedo, J. Alonso, J. Ranilla, and A. Bahamonde. Self-organizing cases to find paradigms. In *Proc. of the IWANN '99*, volume 1606 of *LNCS*, pages 527–536. Springer-Verlag, 1999.
[8] J. del Coz. BETS*: Sistema de aprendizaje basado en la selección de ejemplos paradigmáticos*. PhD thesis, University of Oviedo at Gijón, 2000.
[9] T. Dietterich, M . Kearns, and Y . Mansour. Applying the weak learning framework to understand and improve C4.5. In *Proc. 13th International Conference on Machine Learning*, pages 96–104. Morgan Kaufmann, 1996.
[10] P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168, 1996.
[11] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.
[12] J. Fürnkranz and G. Widmer. Incremental reduced error pruning. In *International Conference on Machine Learning*, pages 70–77, 1994.
[13] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.
[14] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, pages 1137–1145, 1995.
[15] O. Luaces, J. Alonso, E. de la Cal, J. Ranilla, and A. Bahamonde. Machine learning usefulness relies on accuracy and self-maintenance. In *Proc. of the* 11^{th} *IEA & AIE*, volume 1416 of *LNAI*, pages 448–457. Springer-Verlag, 1998.
[16] O. Luaces, J. del Coz, J.R. Quevedo, J. Alonso, J. Ranilla, and A. Bahamonde. Autonomous clustering for machine learning. In *Proc. of the IWANN '99*, volume 1606 of *LNCS*, pages 497–506, Alicante, Spain, 1999. Springer-Verlag.
[17] O. Luaces and A. Bahamonde. Inflating examples to obtain rules. Technical report, Artificial Intelligence Center, University of Oviedo at Gijón, 2000.
[18] S. Muggleton. Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4):245–286, 1995.
[19] S.K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.
[20] B. Pfahringer. Compression-based pruning of decision lists. In *Proc. of the European Conference on Machine Learning*, LNAI-1224, pages 199–212. Springer-Verlag, 1997.
[21] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, 1993.
[22] J. Ranilla and A. Bahamonde. FAN: Finding Accurate iNductions. *International Journal of Human Computer Studies*, 56(4):445–474, June 2002.
[23] B. Smyth and E. McKenna. Building compact competent case-bases. In *ICCBR*, pages 329–342, 1999.
[24] D. R. Wilson and T. R. Martínez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6(1):1–34, 1997.
[25] D.R. Wilson and T.R. Martinez. Reduction techniques for exemplar-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.