Rule Induction and Instance-Based Learning: A Unified Approach

Pedro Domingos
Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717, U.S.A.
pedrod@ics.uci.edu

Abstract

This paper presents a new approach to inductive learning that combines aspects of instance-based learning and rule induction in a single simple algorithm. The RISE system searches for rules in a specific-to-general fashion, starting with one rule per training example, and avoids some of the difficulties of separate-and-conquer approaches by evaluating each proposed induction step globally, i.e., through an efficient procedure that is equivalent to checking the accuracy of the rule set as a whole on every training example. Classification is performed using a best-match strategy, and reduces to nearest-neighbor if all generalizations of instances were rejected. An extensive empirical study shows that RISE consistently achieves higher accuracies than state-of-the-art representatives of its "parent" paradigms (PEBLS and CN2), and also outperforms a decision-tree learner (C4.5) in 13 out of 15 test domains (in 10 with 95% confidence).

1 Introduction

Several well-developed approaches to inductive learning currently exist, among them induction of decision trees [Quinlan, 1993a], rule induction [Clark and Niblett, 1989], and instance-based learning [Aha *et al.*, 1991]. While accuracy in many practical domains is still far from 100%, it is unclear how much, if any, improvement is still possible with current methods. Empirical studies have also shown repeatedly that each approach works best in some, but not all, domains; this has been termed the *selective superiority problem* [Brodley, 1993]. Ideally, we would like to have an algorithm that in each domain of interest performs as well as the best of the algorithms above, or better. While it is now clearly understood that induction is a "zero-sum game", and thus this goal is unachievable for the set of all mathematically possible domains [Schaffer, 1994], it may well be possible to produce learners that perform better on a wide variety of real-world domains, at the cost of worse performance in domains that never occur in practice. One way to attempt this is by combining two or more of the basic approaches into an algorithm that will behave as the most appropriate of them in each situation. This line of research may be termed "empirical multi-strategy learning" [Michalski and Tecuci, 1994]. MCS [Brodley, 1993], KBNGE [Wettschereck, 1994] and ITRULE [Smyth *et al.*, 1990] are examples of systems of this type.

Two induction paradigms with largely complementary strengths and weaknesses are rule induction and instance-based learning (IBL). Rule induction systems often succeed in identifying small sets of highly predictive features, and can make effective use of statistical measures to combat noise. However, they can only form axis-parallel frontiers in the instance space, and they have trouble recognizing exceptions, or in general small, low-frequency sections of the space; this is known as the *small disjuncts problem* [Holte *et al.*, 1989]. Further, their general-to-specific, "separate and conquer" search strategy causes them to suffer from the *splintering problem*: as induction progresses, the amount of data left for further learning dwindles rapidly, leading to wrong decisions or insufficient specialization due to lack of adequate statistical support. On the other hand, IBL (also known as nearest-neighbor) methods can form complex, non-axis-parallel frontiers, and are well suited to handling exceptions, but can be very vulnerable to noise and irrelevant features.

This paper describes an approach to induction that attempts to overcome each of these methods' limitations by combining it with the other. The approach is implemented in the RISE system. Unlike many other empirical multi-strategy systems, RISE does not consist of a global procedure calling the different algorithms as subprocedures, but rather of a single, simple algorithm that can behave both as a nearest-neighbor classifier and a rule induction system. Hence its being termed a "unified" approach, instead of a "combined" one; many other unification schemes are of course possible.

The next section presents the RISE algorithm, followed by the derivation of an upper bound for its time complexity. An empirical study is then reported, comparing RISE and several other systems on 30 domains from the UCI repository, and the results are interpreted. Finally, related work is discussed, and some directions for future research are outlined.

2 The RISE Algorithm

"RISE" stands for "Rule Induction from a Set of Exemplars." This paper describes version 2.0 of the algorithm. The learning and classification procedures will be considered in turn.

2.1 Representation and Search

Each example is a vector of attribute-value pairs, together with a specification of the class to which it belongs; attributes can be either nominal (symbolic) or numeric. Each rule consists of a conjunction of antecedents and a predicted class. Each antecedent is a condition on a single attribute, and there is at most one antecedent per attribute. Conditions on nominal attributes are equality tests of the form $a_i = v_j$, where a_i is the attribute and v_j is one of its legal values. Conditions on numeric attributes take the form of allowable intervals for the attributes, i.e., $a_i \in [v_{j1}, v_{j2}]$, where v_{j1} and v_{j2} are two legal values for a_i. Exemplars (i.e., examples used as prototypes for classification) are viewed as maximally specific rules, with conditions on all attributes and degenerate (point) intervals for numeric attributes. A rule is said to *cover* an example if the example satisfies all of the rule's conditions; a rule is said to *win* an example if it is the nearest rule to the example according to the distance metric that will be described below.

The RISE algorithm is summarized in Table 1. RISE searches for "good" rules in a specific-to-general fashion, starting with a rule set that is the training set of examples itself. RISE looks at each rule in turn, finds the nearest example of the same class that it does not already cover (i.e., that is at a distance greater than zero from it), and attempts to minimally generalize the rule to cover it. The generalization procedure is outlined in Table 2. If the change's effect on global accuracy is positive, it is retained; otherwise it is discarded. Generalizations are also accepted if they appear to have no effect on accuracy; this reflects a simplicity bias. This procedure is repeated until, for each rule, attempted generalization fails. In the worst case, no generalizations are performed, and the end result is simply a nearest-neighbor classifier using all the training examples as exemplars.

Accuracy is measured using a leave-one-out methodology: when attempting to classify an example, the corresponding rule is left out, unless it already covers other examples as well. A potential difficulty with this procedure is that it requires matching all rules (or all but one) with all training examples, and this would entail a high computational cost if it was repeatedly done as outlined. Fortunately, at each step only the *change* in accuracy needs to be computed. Each example memorizes the distance to its nearest rule and its assigned class. When a rule is generalized, all that is necessary is then to match that single rule against all examples, and check if it wins any that it did not before, and what its effect on these is. Previously misclassified examples that are now correctly classified add to the accuracy, and previously correctly classified examples that are now misclassified subtract from it. If the former are more numerous than the latter, the change in accuracy is positive, and the generalization is accepted.

Table 1: The RISE algorithm.

Input: ES is the training set.

Procedure RISE (ES)

Let RS be ES.
Compute $Acc(RS)$.
Repeat
  For each rule R in RS,
    Find the nearest example E to R not already
      covered by it (and of the same class).
    Let R' = Most_Specific_Generalization(R, E).
    Let RS' = RS with R replaced by R'.
    If $Acc(RS') \geq Acc(RS)$
    Then Replace RS by RS',
      If R' is identical to another rule in RS,
      Then delete R' from RS.
Until no increase in $Acc(RS)$ is obtained.
Return RS.

Table 2: Generalization of a rule to cover an example.

Inputs: $R = (a_1, a_2, \ldots, a_A, c_R)$ is a rule,
  $E = (e_1, e_2, \ldots, e_A, c_E)$ is an example.
a_i is either True, $x_i = r_i$, or $r_{i,lower} \leq x_i \leq r_{i,upper}$.

Function Most_Specific_Generalization (R, E)

For each attribute i,
  If a_i = True then Do nothing.
  Else if i is symbolic and $e_i \neq r_i$ then a_i = True.
  Else if $e_i > r_{i,upper}$ then $r_{i,upper} = e_i$.
  Else if $e_i < r_{i,lower}$ then $r_{i,lower} = e_i$.

2.2 Classification

At performance time, classification of each test example is performed by finding the nearest rule to it, and assigning the example to the rule's class. The distance measure used is a combination of Euclidean distance for numeric attributes, and a simplified version of Stanfill and Waltz's value difference metric for symbolic attributes [Stanfill and Waltz, 1986].

Let $E = (e_1, e_2, \ldots, e_A, c_E)$ be an example with value e_i for the ith attribute and class c_E. Let $R = (a_1, a_2, \ldots, a_A, c_R)$ be a rule with class c_R and condition a_i on the ith attribute, where a_i = True if there is no condition on i, otherwise a_i is $x_i = r_i$ if i is symbolic and a_i is $r_{i,lower} \leq x_i \leq r_{i,upper}$ if i is numeric. The distance $\Delta(R, E)$ between R and E is then defined as:

$$\Delta(R, E) = \sum_{i=1}^{A} \delta^2(i) \qquad (1)$$

where the component distance $\delta(i)$ for the ith attribute is:

$$\delta(i) = \begin{cases} 0 & \text{if } a_i = \text{True} \\ SVDM(r_i, e_i) & \text{if } i \text{ is symbolic and } a_i \neq \text{True} \\ \delta_{num}(i) & \text{if } i \text{ is numeric and } a_i \neq \text{True} \end{cases} \quad (2)$$

$SVDM(r_i, e_i)$ is the simplified value difference metric, defined as:

$$SVDM(x_i, x_j) = \sum_{h=1}^{C} |P(c_h|x_i) - P(c_h|x_j)| \quad (3)$$

where x_i and x_j are any legal values of the attribute, C is the number of classes, c_h is the hth class, and $P(c_h|x_i)$ denotes the probability of c_h conditioned on x_i. If a value x_i dos not occur in training, its distance to all other values is taken to be 1. The essential idea behind VDM-type metrics is that two values should be considered similar if they make similar class predictions, and dissimilar if their predictions diverge. This has been found to give good results in several domains [Cost and Salzberg, 1993]. Notice that, in particular, $SVDM(x_i, x_j)$ is always 0 if $i = j$.

The component distance for numeric attributes is defined as:

$$\delta_{num}(i) = \begin{cases} 0 & \text{if } r_{i,lower} \leq e_i \leq r_{i,upper} \\ \frac{e_i - r_{i,upper}}{x_{max} - x_{min}} & \text{if } e_i > r_{i,upper} \\ \frac{r_{i,lower} - e_i}{x_{max} - x_{min}} & \text{if } e_i < r_{i,lower} \end{cases} \quad (4)$$

x_{max} and x_{min} being respectively the maximum and minimum observed values for the attribute.

The distance from a missing numeric value to any other is defined as 0. If a symbolic attribute's value is missing, it is assigned the special value "?". This is treated as a legitimate symbolic value, and its SVDM to all other values of the attribute is computed and used. In the present framework, this is a sensible policy: a missing value is taken to be roughly equivalent to a given possible value if it behaves similarly to it, and inversely if it doesn't.

When two or more rules are equally close to a test example, the rule that was most accurate on the training set wins. So as to not unduly favor more specific rules, the Laplace-corrected accuracy is used [Niblett, 1987]:

$$LAcc(R) = \frac{N_{corr}(R) + 1}{N_{won}(R) + C} \quad (5)$$

where R is any rule, C is the number of classes, $N_{won}(R)$ is the total number or examples won by R, $N_{corr}(R)$ is the number of examples among those that R correctly classifies, and C is the number of classes. The effect of the Laplace correction is to make the estimate of a rule's accuracy converge to the "random guess" value of $1/C$ as the number of examples won by the rule decreases. Thus rules with high apparent accuracy are favored only if they also have high statistical support, i.e., if that apparent accuracy is not simply the result of a small sample.

3 Time Complexity of RISE

This section shows that RISE is a computationally efficient algorithm by deriving an upper bound for its time complexity. Let E be the number of examples in the training set, A the number of attributes, V_S (V_N) the average number of observed values per symbolic (numeric) attribute, R the number of rules, and C the number of classes. Assume for now that all attributes are symbolic. RISE can be divided into two parts: initialization and search. The initialization phase of the algorithm consists of three operations. The first is copying the examples to the rules, and takes $O(EA)$ time. The second is compiling a table of VDM distances, taking $O(EA + AV_S^2C)$ (EA to run through all the examples, for each one noting the correspondence between each attribute's value and the class, and AV_S^2C to sum the results for all classes, for each pair of values of each attribute). The third operation is finding each example's closest rule and computing the initial accuracy of the rule set, which involves matching all rules against all examples. Since there are initially E rules, this takes $O(E^2A)$ time. The total time necessary for initialization is therefore $O(E^2A + AV_S^2C)$.

The search part of the algorithm consists of repeating four steps for each rule (see Table 1): finding its nearest example, generalizing it to cover the example, comparing the altered rule to all examples to see if any are newly won, and (if the change is adopted) comparing the rule to all other rules to check for duplications. These operations consume respectively $O(EA)$, $O(A)$, $O(EA)$ and $O(RA)$ time, for a total of $O(EA + RA)$. Doing this for all rules thus takes $O[R(EA + RA)]$ time. In RISE each example produces at most one rule, therefore $R \leq E$ and this time is at worst $O(E^2A)$.

How many "repeat" cycles can the algorithm perform in the worst case? Two answers are possible, depending on how the stopping criterion is interpreted. If it is applied individually, i.e., generalization of a given rule stops as soon as covering the nearest example produces no improvement, then the "repeat" cycle is performed at worst $O(A)$ times, since each cycle must remove at least one condition, and a rule contains at most A conditions, this being true for each rule. On the other hand, if the stopping criterion is applied globally, i.e. generalization of a given rule stops only when no change to *any* rule produces an improvement, the "repeat" cycle can in theory be performed up to $O(EA)$ times, because in the worst case only one condition of one rule will be dropped in each entire cycle, each time causing some currently-unprofitable change in another rule to become profitable in the next round. This, however, is extremely unlikely. The two policies were empirically compared, showing no appreciable difference between the two in accuracy or time. (By default, global stopping is used.) Multiplying the values above by the cost of a single cycle yields a total time complexity of $O(E^2A^2)$ or $O(E^3A^2)$ respectively. Since $E \geq V_S$, and assuming that $A \geq C$, which is generally the case, the smaller of those values dominates the complexity of the initialization phase, and both therefore constitute upper bounds on the time complexity of the whole algorithm in their respective situations.

Results in [Clark and Niblett, 1989] show that the ba-

Table 3: Experimental results.

Domain	RISE	Default	Conf.	PEBLS	Conf.	CN2	Conf.	C4.5	Conf.
Audiology	76.9±5.3	20.8±3.6	99.5	75.8±5.2	90.0	69.4±6.6	99.5	70.9±6.0	99.5
Annealing	97.4±0.9	76.2±2.2	99.5	99.0±0.7	99.5	82.1±6.6	99.5	93.6±1.6	99.5
Echocardiogram	66.2±5.9	68.1±6.9	90.0	63.1±6.6	99.5	68.3±6.9	95.0	65.8±7.0	-
Glass	70.6±5.6	31.4±5.0	99.5	58.6±5.9	99.5	64.2±6.0	99.5	64.4±7.7	99.5
Heart (Cleve.)	79.8±3.7	55.1±3.8	99.5	78.2±4.3	99.0	78.5±3.6	97.5	76.2±4.3	99.5
Horse colic	82.0±3.0	64.1±3.8	99.5	76.1±3.7	99.5	83.0±3.6	97.5	81.0±3.8	95.0
Hypothyroid	96.8±0.7	95.1±0.9	99.5	94.3±1.6	99.5	98.0±0.7	99.5	98.7±0.7	99.5
LED	59.8±7.1	7.3±3.2	99.5	53.0±7.0	99.5	55.8±9.5	99.5	58.8±8.7	-
Mushroom	99.7±0.4	48.3±1.4	99.5	99.8±0.3	-	99.6±0.4	97.5	99.3±0.6	99.5
Post-operative	64.0±7.5	71.0±6.2	99.5	58.3±7.9	99.5	62.1±7.5	97.5	67.5±8.3	99.5
Promoters	86.9±5.3	42.1±5.3	99.5	90.6±5.2	99.5	74.2±9.7	99.5	79.0±9.8	99.5
Solar flare	71.5±3.7	25.4±4.1	99.5	68.2±3.5	99.5	69.8±4.5	97.5	71.3±4.1	-
Sonar	77.9±9.2	51.0±7.0	99.5	76.9±6.1	-	68.8±7.6	99.5	66.4±6.5	99.5
Splice junctions	91.4±1.7	54.0±3.4	99.5	92.9±1.2	99.5	83.8±3.2	99.5	90.5±2.0	99.0
Zoology	93.9±3.7	40.5±6.3	99.5	94.9±4.2	97.5	91.9±5.2	99.5	90.8±5.4	99.5

sic step of the CN2 and AQ rule induction algorithms takes $O(EAS)$ time plus a logarithmic term, where S is a user-set parameter related to the width of the search. This step is embedded in loops that may run up to $O(EA)$ times, yielding a worst-case total time in excess of $O(E^2A^2S)$. RISE's worst-case complexity is thus competitive with that of standard rule induction algorithms.

The introduction of numeric values simply increases the values above by a factor of V_N, since the single-step removal of a condition may now be replaced by at most $O(V_N)$ steps of expanding the corresponding interval. In practice only a small number of steps may actually be required.

4 Experiments and Results

4.1 Experimental Design

In order to verify if RISE's expected benefits are observed in practice, experiments were carried out on 30 datasets from the UCI repository [Murphy and Aha, 1995]. Half of the domains were first used to select a default version of RISE by 10-fold cross-validation. The domains used in this phase were: breast cancer, credit screening (crx), chess endgames (kr-vs-kp), Pima diabetes, hepatitis, iris, labor negotiations, lung cancer, liver disease, contact lenses, lymphography, primary tumor, soybean (small), voting records, and wine. The version of RISE thus selected is the one described in previous sections. RISE was then compared with state-of-the-art representatives of other approaches on the remaining 15 domains: PEBLS 2.1 for IBL [Cost and Salzberg, 1993], CN2 6.1 for rule induction [Clark and Boswell, 1991], and C4.5 for induction of decision trees [Quinlan, 1993a]. PEBLS 2.1's inability to deal with missing values was overcome by grafting onto it an approach similar to the one used by RISE, and numeric values were discretized as directed in the manual. C4.5RULES, the version of C4.5 that converts trees to rules, was chosen because rules have been observed to achieve the highest accuracies [Quinlan, 1987], and because they are more directly comparable to RISE. The default classifier, which assigns all test examples to the most frequent class, was also included in the study to provide a baseline.

The default versions of all algorithms were used. This ensures a fair comparison: RISE is tested on domains for which it was not fine-tuned, and its default version is compared with other default versions. (If anything, this procedure is unfavorable to RISE, because the test suite includes domains which were used in the development of the other algorithms, as reported in the references above).

Each dataset was randomly divided 50 times into a training set containing two-thirds of the examples, and a testing set containing the remainder. To speed the experiments, datasets with more than 1000 examples were first reduced to this size by random selection. Each algorithm was then trained on each of the 50 training sets, and its accuracy on the corresponding testing set recorded.

4.2 Results

Table 3 shows the average accuracy and sample standard deviation for each algorithm in each domain. The accuracy column for each algorithm is followed by a column showing the confidence level for the difference in accuracy between RISE and that algorithm, using a one-tailed paired t test. A dash denotes less than 90% confidence.

These results are more easily understood by summarizing them in a few comparative measures. These are shown in Table 4. The first line shows the number of domains in which RISE achieved higher accuracy than the corresponding system, vs. the number in which the reverse happened. In each case the comparison is highly favorable to RISE. The second line considers only those domains in which the observed difference is significant with at least 95% confidence, and shows that most of the previous "wins" were indeed significant. The third line shows the results of applying a sign test to the values of line one. This consists of considering the number of wins as a binomial variable, and asking how unlikely

Table 4: Summary of experimental results.

Measure	RISE	PEBLS	CN2	C4.5
No. wins	-	10-5	12-3	13-2
No. signif. wins	-	8-4	12-3	10-2
Sign test	-	85.0	98.0	99.5
Wilcoxon test	-	98.0	99.6	99.4
Average	81.0	78.6	76.6	78.3
Score	48.0	32.0	31.0	31.0

the value obtained is (e.g., 13 wins in 15 trials vs. C4.5). This results in a confidence greater than 95% that RISE is a more accurate learner than CN2 and C4.5, if the set of domains used is assumed to be representative of real-world tasks. The comparison with PEBLS in not conclusive (only 85% confidence). The sign test, however, can be misled by very small, insignificant differences; a more sensitive procedure is the Wilcoxon signed-ranks test [DeGroot, 1986], which also takes into account the relative magnitudes of the differences between each pair of accuracies being compared. This produces confidences greater than 95% for all comparisons, reflecting that the larger, more significant differences in accuracy tend to be in RISE's favor, and inversely for the smaller, more uncertain ones.

The average accuracy across all domains is a measure of questionable significance, but it is often reported, and is also included here. RISE achieves the highest average accuracy. Finally, a comparison of all the algorithms was carried out by, for each domain, assigning 4 points to the most accurate algorithm, 3 to the second most accurate one, and so on. RISE obtains the highest score by a wide margin. This reflects the fact that RISE performs consistently well, i.e., even when it is not the most accurate algorithm, it is almost always the second best one.

Many widely-used domains were not included in this study, because they had previously been used to fine-tune RISE. As a further check that the results obtained were not a fortuitous consequence of the choice of domains, the same experimental procedure was applied to the "training" domains, and the results merged with the above ones. The percentage of domains in which RISE wins vs. CN2 and C4.5 is now somewhat smaller, and vs. PEBLS it is higher. Overall the differences and confidences obtained are substantially higher than before, reflecting the fact that twice as many domains were used. The sign and Wilcoxon tests both yield confidences in RISE's superiority in excess of 99% vs. all algorithms. Thus there is strong evidence that, if the domains used are considered representative, RISE is the most accurate of the algorithms tested.

Another significant observation is that, in approximately half of the test domains used (and similarly for the training ones), RISE's accuracy exceeds the highest of PEBLS's and CN2's. This shows that a multistrategy learning approach can not only often match the results of the best of its "parent" paradigms, but also achieve new synergies between them.

4.3 Discussion

The results above can be interpreted as follows.

Compared with IBL algorithms, RISE has the crucial advantage of being able to select different sets of relevant features in different sections of the instance space. Several well-known methods for removing irrelevant features in nearest-neighbor classifiers exist [Aha and Bankert, 1994], but the decisions they make are very coarse, applying to the whole instance space at once. The same consideration holds for the many feature-weighting schemes that have been proposed [Mohri and Tanaka, 1994]. Rule induction systems are able to detect that certain features are relevant only in the context of others, and RISE shares this ability: a feature may be dropped in some rules, but not others. Also, the VDM-type metrics used by PEBLS and other systems are effective in reducing the influence of irrelevant features, but require numeric attributes to be discretized, losing the ordering information they contain. If a Euclidean-type distance is used for numeric attributes, irrelevant ones may seriously affect the results. RISE is able to use Euclidean distance for numeric features, and generalize or drop them if they are irrelevant. Additionally, RISE's search strategy is such that it only diverges from a nearest-neighbor classifier if this causes an estimate of accuracy to improve, and thus as long as this estimate is good RISE should be able to not perform worse in general than such a classifier.

Compared with rule induction algorithms (and decision tree ones, which share many biases with them), RISE has several notable advantages. First, due to the best-match policy it employs for classification, it is able to form complex, non-axis-parallel frontiers in the search space, and is thus at an advantage when these are appropriate. Second, because of its specific-to-general search direction, and the Laplace accuracy measure used to choose a winner when several rules cover an example, RISE is better able to deal with exceptions and small disjuncts: they will often be retained when generalizing, because absorbing or expanding them into larger rules would decrease accuracy; and at classification time, they will prevail over those larger rules if their Laplace accuracy is higher. Finally, RISE's "conquering without separating" search strategy avoids some of the difficulties of "separate and conquer" ones: each induction step is evaluated with respect to how it affects the accuracy of the rule set as a whole on the entire training set, mitigating the splintering problem.

RISE has some disadvantages. It is on average the slowest of the systems compared, although this is only of any significance in the largest domains (in all others, every algorithm runs in seconds on a Sun 670). The two slowest domains were hypothyroid and splice junctions, where RISE took respectively 119 minutes and 20 minutes. RISE has not been optimized, however, and several important components of the system are amenable to such optimization. Beyond that, windowing and other sampling techniques can be used without expected loss in accuracy [Catlett, 1991]. Also, even though RISE's memory cost is much smaller than that of a simple nearest-neighbor classifier, the rule sets it

produces are not as compact as those output by C4.5 or CN2. RISE's greater costs will generally be a price well worth paying for the additional accuracy obtained. However, for domains of very large size, and/or when comprehensibility is paramount, a system like C4.5 will still be the first choice.

5 Related Work

The RISE approach should be seen in the context of previous work in inductive learning. Several algorithms proposed in the literature can be seen as empirical multi-strategy learners, but combining different paradigms from RISE's: decision trees, IBL and linear machines [Brodley, 1993], decision trees and rules [Quinlan, 1987], decision trees and perceptrons [Utgoff, 1989], rules and Bayesian classification [Smyth *et al.*, 1990], back-propagation and genetic algorithms [Belew *et al.*, 1992], etc. Quinlan [Quinlan, 1993b] has successfully combined IBL with trees and other methods, but for the purpose of regression as opposed to classification, performing this combination only at classification time, and in a way that depends critically on the predicted value being continuous.

AQ15 [Michalski *et al.*, 1986] is a rule induction system that employs best-match classification. Its approach was carried further in the FCLS system [Zhang, 1990], which combines rules with exemplars in an attempt to alleviate the small disjuncts problem. Unlike RISE, FCLS employs different representations for rules and exemplars, and uses a separate-and-conquer strategy similar to that of its AQ ancestors.

Golding and Rosenbloom's Anapron system [Golding and Rosenbloom, 1991] combines case-based and rule-based reasoning in a name-pronunciation task. It differs substantially from RISE in that it does not learn rules, but rather makes use of a pre-existing knowledge base. It also treats cases and rules separately, and employs a different matching procedure.

A system more similar to RISE is EACH/NGE [Salzberg, 1991], which produces and uses hyperrectangles generalized from specific instances. It differs from RISE in many ways, however: it is applicable only in purely numerical domains, is an incremental algorithm, never drops attributes, uses different heuristics and search strategies, allows only nested hyperrectangles as opposed to arbitrary intersecting ones, always prefers the most specific hyperrectangle, etc. Recently Wettschereck and Dietterich [1995] have carried out a detailed comparison of NGE and k-nearest-neighbor (kNN), and designed an algorithm that combines the two [Wettschereck, 1994], but does not achieve greater accuracy than kNN alone. They found that NGE performs substantially worse than kNN, and that the chief cause of this is NGE's use of overlapping rectangles. The fact that RISE performs better than nearest-neighbor, while NGE performs worse with a representation that is similar in the case of numeric attributes, deserves some attention.

In the cross-validation studies reported above, RISE's tie-breaking policy based on Laplace accuracy was compared with one selecting the most specific rule as in NGE, and found to be clearly superior. This can be understood as follows. In regions of overlap, NGE arbitrarily assigns all examples to the class of the most specific hyperrectangle. In contrast, RISE's learning strategy approximates the optimal decision rule of placing the boundary between two classes at the point where the density of examples from one overtakes that of the other [Duda and Hart, 1973]. This is because a rule is started from each example, and its generalization halts when it would include more examples of other classes than of the example's one. The use of Laplace accuracy then implies that, given similar-sized samples, each rule prevails in areas where the density of examples of its class is greater. RISE's batch-learning approach also avoids the problems that NGE's incremental learning one was observed to suffer from. These factors, and the experimental results reported above, support the conclusion that generalizing instances to rules can indeed produce substantial improvements in accuracy, if done in an appropriate manner.

6 Future Work

A priority area for future research is carrying out lesion studies and experiments in artificial domains to verify whether the interpretation of results in the previous discussion section is indeed correct. More detailed observation of RISE's workings is also needed to find out how well and how RISE is dealing with the splintering and small disjuncts problems.

Another important direction for research is extending RISE to make use of domain knowledge, bringing the analytical component into the current multistrategy approach.

7 Conclusions

This paper presented an approach to inductive learning that attempts to combine the best features of IBL and rule induction. The RISE algorithm searches for rules in a specific-to-general fashion, avoiding some of the pitfalls of "separate and conquer" methods, and uses a best-match classification procedure, enabling it to form non-axis-parallel frontiers in the instance space. At the same time it shares with rule induction methods the ability to find small sets of highly predictive features in a context-sensitive manner. In experiments on a large number of practical domains, RISE achieved significantly higher accuracies than either of its parent approaches alone, and also compared favorably with a decision-tree algorithm. These results show that multistrategy learning can create significant synergies between the methods it combines, and thus produce improved classifiers.

Acknowledgments

This work was partly supported by JNICT/Programa Ciência and Fulbright scholarships. The author is grateful to Dennis Kibler and Mike Pazzani for many helpful comments and suggestions, to the creators of the CN2, PEBLS and C4.5 systems, and to all the people who provided the datasets used in the empirical study. Please

see the documentation in the UCI Repository for detailed information.

References

[Aha and Bankert, 1994] D. W. Aha and R. L. Bankert. Feature selection for case-based classification of cloud types: An empirical comparison. In *Proc. AAAI-94 Workshop on Case-Based Reasoning*, pages 106–112, 1994.

[Aha *et al.*, 1991] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[Belew *et al.*, 1992] R. K. Belew, J. McInerney, and N. N. Schraudolph. Evolving networks: Using the genetic algorithm with connectionist learning. In C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pages 511–547. Addison-Wesley, Redwood City, CA, 1992.

[Brodley, 1993] C. E. Brodley. Addressing the selective superiority problem: Automatic algorithm/model class selection. In *Proc. 10th Machine Learning Conf.*, pages 17–24, 1993.

[Catlett, 1991] J. Catlett. Megainduction: A test flight. In *Proc. 8th Machine Learning Conf.*, pages 589–604, 1991.

[Clark and Boswell, 1991] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proc. EWSL-91*, pages 151–163, 1991.

[Clark and Niblett, 1989] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[Cost and Salzberg, 1993] S. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.

[DeGroot, 1986] M. H. DeGroot. *Probability and Statistics*. Addison-Wesley, Reading, MA, 2nd edition, 1986.

[Duda and Hart, 1973] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[Golding and Rosenbloom, 1991] A. R. Golding and P. S. Rosenbloom. Improving rule-based systems through case-based reasoning. In *Proc. AAAI-91*, pages 22–27, 1991.

[Holte *et al.*, 1989] R. C. Holte, L. E. Acker, and B. W. Porter. Concept learning and the problem of small disjuncts. In *Proc. IJCAI-89*, pages 813–818, 1989.

[Michalski and Tecuci, 1994] R. Michalski and G. Tecuci, editors. *Machine Learning: A Multistrategy Approach*. Morgan Kaufmann, San Mateo, CA, 1994.

[Michalski *et al.*, 1986] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proc. AAAI-86*, pages 1041–1045, 1986.

[Mohri and Tanaka, 1994] T. Mohri and H. Tanaka. An optimal weighting criterion of case indexing for both numeric and symbolic attributes. In *Proc. AAAI-94 Workshop on Case-Based Reasoning*, pages 123–127, 1994.

[Murphy and Aha, 1995] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. Machine-readable data repository, Department of Information and Computer Science, University of California at Irvine, Irvine, CA, 1995.

[Niblett, 1987] T. Niblett. Constructing decision trees in noisy domains. In *Proc. EWSL-87*, pages 67–78, 1987.

[Quinlan, 1987] J. R. Quinlan. Generating production rules from decision trees. In *Proc. IJCAI-87*, pages 304–307, 1987.

[Quinlan, 1993a] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Quinlan, 1993b] J. R. Quinlan. Combining instance-based and model-based learning. In *Proc. 10th Machine Learning Conf.*, pages 236–243, 1993.

[Salzberg, 1991] S. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:251–276, 1991.

[Schaffer, 1994] C. Schaffer. A conservation law for generalization performance. In *Proc. 11th Machine Learning Conf.*, pages 259–265, 1994.

[Smyth *et al.*, 1990] P. Smyth, R. M. Goodman, and C. Higgins. A hybrid rule-based/Bayesian classifier. In *Proc. ECAI-90*, pages 610–615, 1990.

[Stanfill and Waltz, 1986] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29:1213–1228, 1986.

[Utgoff, 1989] P. E. Utgoff. Perceptron trees: A case study in hybrid concept representations. *Connection Science*, 1:377–391, 1989.

[Wettschereck and Dietterich, 1995] D. Wettschereck and T. Dietterich. An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. *Machine Learning*, 1995. To appear.

[Wettschereck, 1994] D. Wettschereck. A hybrid nearest-neighbor and nearest-hyperrectangle algorithm. In *Proc. 7th ECML*, 1994.

[Zhang, 1990] J. Zhang. A method that combines inductive learning with exemplar-based learning. In *Proc. 2nd International IEEE Conf. on Tools for Artificial Intelligence*, pages 31–37, 1990.