From Instances to Rules: A Comparison of Biases

Pedro Domingos

Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717, U.S.A.
pedrod@ics.uci.edu
http://www.ics.uci.edu/~pedrod

Abstract

RISE is an algorithm that combines rule induction and instance-based learning (IBL). It has been empirically verified to achieve higher accuracy than state-of-the-art representatives of its parent approaches in a large number of benchmark problems. This paper investigates the conditions under which RISE's bias will be more appropriate than that of the pure approaches, through experiments in carefully controlled artificial domains. RISE's advantage compared to pure rule induction increases with increasing concept specificity. RISE's advantage compared to pure IBL is greater when the relevance of features is context-dependent (i.e., when some of the features used to describe examples are relevant only given other features' values). The paper also reports lesion studies and other empirical observations showing that RISE's good performance is indeed due to its combination of rule induction and IBL, and not to the presence of either component alone.

Introduction

Rule induction (either performed directly (Michalski 1983) or by means of decision trees (Quinlan 1993a)) and instance-based learning (Aha, Kibler, & Albert 1991) (forms of which are also known as case-based, memory-based, exemplar-based, lazy, local, and nearest-neighbor learning) constitute two of the leading approaches to concept and classification learning. Rule-based methods discard the individual training examples, and remember only abstractions formed from them. At performance time, rules are typically applied by logical match (i.e., only rules whose preconditions are satisfied by an example are applied to it). Instance-based methods explicitly memorize some or all of the examples; they generally avoid forming abstractions, and instead invest more effort at performance time in finding the most similar cases to the target one.

The two paradigms have largely complementary strengths and weaknesses. Rule induction systems often succeed in identifying small sets of highly predictive features, and, crucially, these features can vary from example to example. However, these methods can have trouble recognizing exceptions, or in general small, low-frequency sections of the space; this is known as the "small disjuncts problem" (Holte, Acker, & Porter 1989). Further, the general-to-specific, "separate and conquer" search strategy they typically employ causes them to suffer from the "fragmentation 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 methods are well suited to handling exceptions, but can be very vulnerable to irrelevant features. If many such features are present in the example descriptions, IBL systems will be confused by them when they compare examples, and accuracy may suffer markedly. Unsurprisingly, in classification applications each approach has been observed to outperform the other in some, but not all, domains.

We believe that rule induction and instance-based learning have much more in common than a superficial examination reveals, and can be unified into a single, simple and coherent framework for classification learning, one that draws on the strengths of each to combat the limitations of the other. This unification rests on two key observations. One is that an instance can be regarded as a maximally specific rule (i.e., a rule whose preconditions are satisfied by exactly one example). Therefore, no syntactic distinction need be made between the two. The second observation is that rules can be matched approximately, as instances are in an instance-based classifier (i.e., a rule can match an example if it is the closest one to it according to some similarity-computing procedure, even if the example does not logically satisfy all of the rule's preconditions; see (Michalski *et al.* 1986)). A rule's extension, like an instance's, then becomes the set of examples that it is the most similar rule to, and thus there is also no necessary semantic distinction between a rule and an instance.

The RISE algorithm (Domingos 1995b) is a practical, computationally efficient realization of this idea.[1]

[1]Obviously, it is not the only possible approach to uni-

RISE starts with a rule base that is simply the training set itself, and gradually generalizes each rule to cover neighboring instances, as long as this does not increase the rule base's error rate on the known cases. If no generalizations are performed, RISE acts as a pure instance-based learner. If all cases are generalized and the resulting set of rules covers all regions of the instance space that have nonzero probability, it acts as a pure rule inducer. More generally, it will produce rules along a wide spectrum of generality; sometimes a rule that is logically satisfied by the target case will be applied, and in other cases an approximate match will be used. RISE's bias, which is in effect intermediate between that of pure rule inducers and that of pure instance-based learners, has been observed to lead to improvements in accuracy in a large number of domains from the UCI repository (Murphy & Aha 1995), resulting in significantly better overall results than either "parent" bias (with C4.5RULES (Quinlan 1993a) and CN2 (Clark & Boswell 1991) being used as representatives of rule induction, and PEBLS (Cost & Salzberg 1993) as a representative of IBL). RISE is described in greater detail in the next section.

The question now arises of exactly what factors RISE's comparative advantage is due to, and thus of when it will be appropriate to apply this algorithm instead of a pure IBL or a pure rule induction one. This will first be approached by showing through lesion studies that RISE's strength derives from the simultaneous presence of the two components, and not from either one alone. We will then consider rule induction and IBL in turn, formulating hypotheses as to the factors that favor RISE over the "atomic" approach, and testing these hypotheses through empirical studies in artificial domains, where these factors are systematically varied.

The RISE Algorithm

RISE's learning and classification procedures will be considered in turn. More details can be found in (Domingos 1995b; 1995a).

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. Instances (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.

fying the two paradigms (cf. (Branting & Porter 1991; Golding & Rosenbloom 1991; Quinlan 1993b), etc.).

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.

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.

A potential difficulty is that measuring the accuracy of a rule set on the training set requires matching all rules 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

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 $r_{i,lower} = e_i$.

---

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. With this optimization, RISE's worst-case time complexity has been shown to be quadratic in the number of examples and the number of attributes, which is comparable to that of commonly-used rule induction algorithms (Domingos 1995b).

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 & 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:

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

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

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

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

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

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. 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 & 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:

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

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 distance to all other values of the attribute is computed and used. When coupled with SVDM, 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 does not.

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):

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

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.

Lesion Studies

Lesion studies were conducted using 30 datasets from the UCI repository (Murphy & Aha 1995). Several aspects of the algorithm's performance were also measured. The results are shown in Table 3. Superscripts indicate significance levels for the accuracy differences between systems, using a one-tailed paired t

Table 3: Results of lesion studies, and performance monitoring. Superscripts denote significance levels: 1 is 0.5%, 2 is 1%, 3 is 2.5%, 4 is 5%, 5 is 10%, and 6 is above 10%.

	Accuracy of subsystem				Match type frequency			
Domain	RISE	IBL	Rules	No tie-b.	No/One	No/Multi	One	Multi
Audiology	77.0	75.8^{5}	55.1^{1}	76.2^{1}	53.6	1.6	43.0	1.8
Annealing	97.4	97.7^{4}	77.7^{1}	97.2^{1}	24.1	0.0	75.6	0.2
Breast cancer	67.7	65.1^{1}	69.0^{4}	68.7^{1}	33.4	1.5	59.1	6.0
Credit screening	83.3	81.3^{1}	66.9^{1}	83.2^{6}	51.9	0.0	46.9	1.1
Chess endgames	98.2	91.9^{1}	91.9^{1}	98.0^{1}	27.6	0.1	71.4	0.9
Pima diabetes	70.4	70.3^{6}	66.2^{1}	70.5^{1}	70.9	0.1	27.4	1.6
Echocardiogram	64.6	59.2^{1}	65.7^{6}	64.5^{6}	70.1	0.1	26.8	3.0
Glass	70.6	68.3^{2}	47.3^{1}	70.4^{2}	71.5	0.0	27.2	1.3
Heart disease	79.7	77.8^{1}	64.7^{1}	79.7^{6}	63.1	0.0	34.8	2.1
Hepatitis	78.3	78.4^{6}	79.6^{5}	78.5^{5}	55.3	0.1	43.1	1.5
Horse colic	82.6	76.6^{1}	79.0^{1}	81.7^{1}	39.7	0.2	55.4	4.6
Thyroid disease	97.5	94.1^{4}	84.8^{1}	97.5^{6}	40.7	0.1	58.3	0.9
Iris	94.0	94.7^{3}	71.0^{1}	94.0^{6}	45.9	0.0	54.0	0.2
Labor neg.	87.2	90.8^{1}	73.7^{1}	87.1^{6}	51.8	0.2	46.6	1.4
Lung cancer	44.7	42.0^{6}	26.5^{1}	44.2^{5}	88.2	0.4	9.1	2.4
Liver disease	62.4	60.9^{4}	62.1^{6}	62.3^{6}	72.7	0.2	24.0	3.0
Contact lenses	77.2	72.5^{1}	64.8^{1}	75.8^{3}	13.2	1.5	83.0	2.2
LED	59.9	55.9^{1}	52.7^{1}	49.7^{1}	14.7	4.5	47.3	33.5
Lymphography	78.7	82.0^{1}	70.1^{1}	78.2^{3}	42.9	0.4	53.9	2.8
Mushroom	100.0	97.5^{6}	100.0^{6}	99.8^{1}	7.3	0.0	92.5	0.2
Post-operative	64.1	59.1^{1}	70.9^{1}	65.9^{1}	36.5	12.2	44.7	6.6
Promoters	86.8	90.6^{1}	68.4^{1}	85.9^{4}	59.7	0.0	35.8	4.5
Primary tumor	40.3	34.3^{1}	33.5^{1}	37.0^{1}	34.2	4.1	41.6	20.1
Solar flare	71.6	71.1^{6}	65.5^{1}	68.1^{1}	16.7	2.9	59.9	20.4
Sonar	77.9	83.8^{1}	52.9^{1}	77.9^{6}	95.6	0.0	4.3	0.1
Soybean	100.0	100.0^{6}	85.1^{1}	100.0^{6}	16.9	0.0	83.1	0.0
Splice junctions	93.1	87.8^{1}	75.9^{1}	92.1^{1}	67.4	0.0	29.9	2.7
Voting records	95.2	94.6^{3}	83.7^{1}	94.2^{1}	7.4	0.1	90.3	2.1
Wine	96.9	95.1^{1}	51.5^{1}	96.9^{6}	78.1	0.0	21.9	0.0
Zoology	93.9	94.5^{4}	81.0^{1}	93.7^{4}	17.5	0.0	81.9	0.5

test.[2] The first four columns compare the full system's accuracy ("RISE") with that obtained using lesioned versions: the IBL component alone ("IBL"), the rule induction component alone ("Rules"), and disabling the tie-breaking procedure ("No tie-b."). The last four columns all refer to the full RISE system, and show, respectively: the percentage of test cases that were not matched by any rule, but had a single closest rule, or for which all equally close rules were of the same class ("No/One"); the percentage not matched by any rule, and for which there were equally close rules of more than one class ("No/Multi"); the percentage matched by only one rule, or rules of only one class ("One"); and the percentage matched by rules of more than one class ("Multi"). These observations aid in interpreting the lesion study results.

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. The second line considers only those domains in which the observed difference is significant at the 5% level or lower. The third line shows the global significance levels obtained by applying a Wilcoxon signed-ranks test (DeGroot 1986) to the 30 accuracy differences observed. The average accuracy across all domains is a measure of debatable significance, but it is often reported, and is shown on the last line.

The first specific question addressed was whether

[2] Since, in each case, the goal is to determine whether RISE's accuracy is *higher* than that of the lesioned system (and not just different from it in either direction), a one-tailed test is the appropriate choice (rather than a two-tailed one).

Table 4: Summary of lesion study results.

Measure	RISE	IBL	Rules	No t.-b.
No. wins	-	21-8	25-4	20-4
No. sig. wins	-	16-7	24-2	15-3
Wilcoxon test	-	0.5%	0.1%	0.2%
Average	79.7	78.1	67.9	79.0

there is any gain in the rule induction process (i.e., whether RISE constitutes an improvement over pure instance-based learning). The "IBL" column in Table 3 reports the accuracies obtained by the initial, ungeneralized instance set, and shows that generalization often produces significant gains in accuracy, while seldom having a negative effect.

The converse question is whether the instance-based component is really necessary. Simply assigning examples not covered by any rule to a default class, as done in most rule induction systems, might be sufficient. The "Rules" column in Table 3 shows the results obtained using this policy, and confirms the importance of "nearest-rule" classification in RISE. The sum of the "No" columns in the right half of Table 3 is the percentage of test cases assigned to the default class. This is often very high, the more so because RISE tends to produce rules that are more specific than those output by general-to-specific inducers. The use of nearest-rule is thus essential. Note that the results reported in the "Rules" column are for applying the default rule during both learning and classification; applying it exclusively during classification produced only a slight improvement.

Another important component of RISE whose utility needs to be determined is the conflict resolution policy, which in RISE consists of letting the tied rule with the highest Laplace accuracy win. This was compared with simply letting the most frequent class win ("No tie-b." column in Table 3). The sum of the "Multi" columns in the right half of Table 3 is the percentage of cases where tie-breaking is necessary. This is typically small, and the increase in accuracy afforded by RISE's conflict resolution strategy is correspondingly small (0.7% on average, for all datasets). However, this increase is consistently produced, as evinced by the fact that RISE is more accurate than its lesioned version with a 0.2% significance by the Wilcoxon test.

Taken together, the lesion studies show that each of RISE's components is essential to its performance, and that it is their combination in one system that is responsible for the excellent results obtained by RISE *vis-à-vis* other approaches.

RISE as Rule Induction

Our hypothesis is that RISE's advantage relative to "divide and conquer" rule induction algorithms is at least in part due to its greater ability to identify small

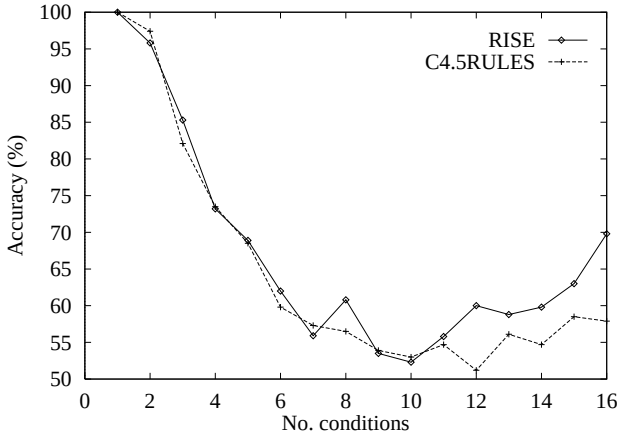

Figure 1: Accuracy as a function of concept specificity (16 features).

regions in the instance space (i.e., regions that are represented by few examples in the training set). Thus RISE should be more accurate than a "divide and conquer" algorithm when the target concepts are fairly to very specific, with the advantage increasing with specificity. Thus the independent variable of interest is the specificity of the target concept description. A good operational measure of it is the average length of the rules comprising the correct description: rules with more conditions imply a more specific concept. The dependent variables are the out-of-sample accuracies of RISE and of a "divide and conquer" algorithm; C4.5RULES (Quinlan 1993a) was used as the latter. Concepts defined as Boolean functions in disjunctive normal form were used as targets. The datasets were composed of 100 examples described by 16 attributes. The average number of literals C in each disjunct comprising the concept was varied from 1 to 16. The number of disjuncts was set to $\mathrm{Min}\{2^{C-1}, 25\}$. This attempts to keep the fraction of the instance space covered by the concept roughly constant, up to the point where it would require more rules than could possibly be learned. Equal numbers of positive and negative examples were included in the dataset, and positive examples were divided evenly among disjuncts. In each run a different target concept was used, generating the disjuncts at random, with length given by a binomial distribution with mean C and variance $C(1-\frac{C}{16})$; this is obtained by including each feature in the disjunct with probability $\frac{C}{16}$. Twenty runs were conducted, with two-thirds of the data used for training and the remainder for testing.

The results are shown graphically in Fig. 1. The most salient aspect is the large difference in difficulty between short and long rules for both learners. Concepts with very few (approx. three or less) conditions

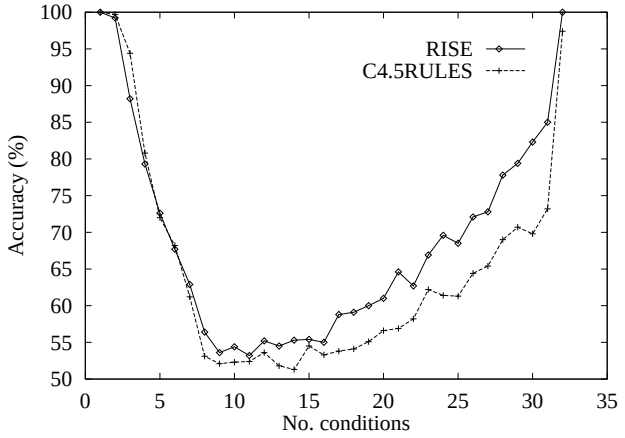

Figure 2: Accuracy as a function of concept specificity (32 features).

per rule are so simple that both RISE and C4.5RULES are able to learn them easily. In separate experiments, corrupting the data with 10% and 20% noise degraded the performance of the two algorithms equally, again giving no advantage to C4.5RULES. At the other end, however, RISE has a clear advantage for concepts with 12 or more conditions per rule; all differences here are significant at the 5% level using a one-tailed paired t test.[3]

The slight upward trend in C4.5RULES's curve for $C > 10$ was investigated by repeating the experiments with 32 attributes, 400 examples, a maximum of 50 rules and $C = 1, \ldots, 32$. The results are shown in Fig. 2. C4.5RULES's lag increases, but the upward trend is maintained; on inspection of the rules C4.5RULES produces, this is revealed to be due to the fact that, as the concept rules become more and more specific, it becomes possible to induce short rules for its negation. The hardest concepts, for which both the concept and its negation have necessarily long rules, are for intermediate values of C.

In summary, the results of this study support the hypothesis that the specificity of the regions to be learned is a factor in the difference in accuracy between RISE and "divide and conquer" rule induction systems, with greater specificity favoring RISE.

RISE as IBL

High sensitivity to irrelevant features has long been recognized as IBL's main problem. A natural solution is identifying the irrelevant features, and discarding them before storing the examples for future use. Several algorithms have been proposed for this purpose (see (Kittler 1986) for a survey), of which two of the most widely known are forward sequential search (FSS) and backward sequential search (BSS) (Devijver & Kittler 1982). Many variations of these exist (e.g., (Aha & Bankert 1994)). Their use can have a large positive impact on accuracy. However, all of these algorithms have the common characteristic that they ignore the fact that some features may be relevant only in context (i.e., given the values of other features). They may discard features that are highly relevant in a restricted sector of the instance space because this relevance is swamped by their irrelevance everywhere else. They may retain features that are relevant in most of the space, but unnecessarily confuse the classifier in some regions.

Consider, for example, an instance space defined by a set of numeric features $\mathbf{F}$, and a class composed of two hyperrectangles, one of which is defined by intervals $f_i \in [a_i, b_i]$ in a subset $\mathbf{F_1}$ of the features, and the other by intervals in a subset $\mathbf{F_2}$ disjoint from the first. Current feature selection algorithms would retain all features in $\mathbf{F_1}$ and $\mathbf{F_2}$, because each of those features is relevant to identifying examples in one of the hyperrectangles. However, the features in $\mathbf{F_2}$ act as noise when identifying examples defined by $\mathbf{F_1}$, and vice-versa. Instead of storing the same set of features for all instances, a better algorithm would discard the features in $\mathbf{F_2}$ from the stored instances of the first hyperrectangle, and the features in $\mathbf{F_1}$ from those of the second one. RISE has this capability.

Our hypothesis is that, viewed as an instance-based learner, RISE derives strength from its ability to perform context-sensitive feature selection (since different examples may be covered by different rules, and thus different features will be used in their classification). Thus, RISE's advantage relative to IBL using conventional feature selection methods should increase with the degree of context sensitivity of feature relevance. To empirically investigate this hypothesis, a concrete measure of the latter is required. If the target concept description is composed of a set of prototypes, one such possible measure is the average D for all pairs of prototypes of the number of features that appear in the definition of one, but not the other:

$$D = \frac{2}{P(P-1)} \sum_{i=1}^{P} \sum_{j=1}^{i-1} \sum_{k=1}^{F} d_{ijk} \qquad (6)$$

where P is the number of prototypes, F is the total number of features, and d_{ijk} is 1 if feature k appears in the definition of prototype i but not in that of prototype j or vice-versa, and 0 otherwise. This "feature difference" measure was taken as the independent variable in the study.

RISE's pure IBL component (see the section on lesion studies) was taken as the basic instance-based learner, and FSS and BSS were applied to it. For comparison, RISE's generalization procedure was also applied, but in order to ensure the fairness of this pro-

[3] See the previous footnote regarding this test.

cedure, all aspects of RISE that do not relate to feature selection were disabled: numeric features were not generalized to intervals, but either retained as point values or dropped altogether,[4] generalization for each rule stopped as soon as an attempted feature deletion for that rule failed (as opposed to only when attempts failed for all rules simultaneously), and duplicate rules were not deleted. The resulting simplified algorithm will hereafter be referred to as "RC". Thus the dependent variables of interest were the accuracies of RC, FSS and BSS.

Two-class problems were considered, with 100 examples in each dataset, described by 32 features. In each domain, each feature was chosen to be numeric or Boolean with equal probability (i.e., the number of numeric features is a binomial variable with expected value $F/2$ and variance $F/4$). Class 1 was defined by ten clusters, and class 0 was the complement of class 1. Each prototype or cluster was defined by a conjunction of conditions on the relevant features. The required value for a Boolean feature was chosen at random, with 0 and 1 being equally probable. Each numeric feature i was required to fall within a given range $[a_i, b_i]$, with a_i being the smaller of two values chosen from the interval [-1, 1] according to a uniform distribution, and b_i the larger one. A cluster was thus a hyperrectangle in the relevant numeric subspace, and a conjunction of literals in the Boolean one.

The choice of relevant features for each prototype was made at random, but in a way that guaranteed that the desired value of D for the set of prototypes was maintained on average. The feature difference D was varied from 0 to 8, the latter being the maximum value that can be produced given the number of features and prototypes used. Twenty domains were generated for each value of D, and two-thirds of the examples used as the training set. The average accuracy of RC, FSS and BSS on the remaining examples is shown graphically as a function of D in Figure 3.

All differences in accuracy between RC and FSS are significant at the 5% level, as are those between RC and BSS for $D = 1$, 2, 4, 5, and 8. The smallest difference occurs when $D = 0$, as our hypothesis would lead us to expect. All accuracies are negatively correlated with D, but the absolute value of the correlation is much smaller for RC (0.49) than for FSS and BSS (0.89 and 0.82, respectively). The downward slope of the regression line for RC's accuracy as a function of D (–0.35) is also much smaller than that for FSS (–1.21) and BSS (–0.61). We thus conclude that RC's higher performance is indeed at least partly due do its context sensitivity.

[4] The policy adopted was to compute the mean and standard deviation of each numeric feature from the sample in the training set, and attempt dropping the feature only when the values for the rule and the example to which its generalization is being tried differ by more than one standard deviation.

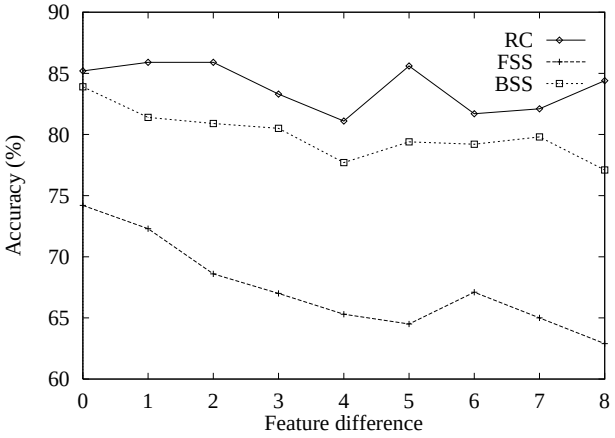

Figure 3: Accuracy as a function of context dependency.

Conclusion

In this paper we investigated the bias of RISE, an algorithm that combines rule induction and instance-based learning, and has been observed to achieve higher accuracies than state-of-the-art representatives of either approach. Lesion studies using benchmark problems showed that each of the two components is essential to RISE's high performance. Studies in carefully controlled artificial domains provided evidence for the hypothesis that, compared to rule inducers, RISE's strength lies in its ability to learn fairly to highly specific concepts, and, compared to instance-based learners, in its ability to detect context dependencies in feature relevance.

Directions for future research include: elucidating further factors in the differential performance of RISE relative to rule induction and IBL; repeating the experiments described here with a wider variety of rule and instance-based learners and artificial domains; and bringing further types of learning into RISE's framework, including in particular the use of analytical learning from expert-supplied domain knowledge.

Acknowledgments

This work was partly supported by a JNICT/PRAXIS XXI scholarship. The author is grateful to Dennis Kibler for many helpful comments and suggestions, and to all those who provided the datasets used in the lesion studies. Please see the documentation in the UCI Repository for detailed information.

References

Aha, D. W., and Bankert, R. L. 1994. Feature selection for case-based classification of cloud types: An empirical comparison. In *Proceedings of the 1994*

AAAI Workshop on Case-Based Reasoning, 106–112. Seattle, WA: AAAI Press.

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

Branting, L. K., and Porter, B. W. 1991. Rules and precedents as complementary warrants. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 3–9. Anaheim, CA: AAAI Press.

Clark, P., and Boswell, R. 1991. Rule induction with CN2: Some recent improvements. In *Proceedings of the Sixth European Working Session on Learning*, 151–163. Porto, Portugal: Springer-Verlag.

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

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

Devijver, P. A., and Kittler, J. 1982. *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, N.J.: Prentice/Hall.

Domingos, P. 1995a. The RISE 2.0 system: A case study in multistrategy learning. Technical Report 95-2, Department of Information and Computer Science, University of California at Irvine, Irvine, CA.

Domingos, P. 1995b. Rule induction and instance-based learning: A unified approach. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1226–1232. Montréal, Canada: Morgan Kaufmann.

Golding, A. R., and Rosenbloom, P. S. 1991. Improving rule-based systems through case-based reasoning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 22–27. Menlo Park, CA: AAAI Press.

Holte, R. C.; Acker, L. E.; and Porter, B. W. 1989. Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 813–818. Detroit, MI: Morgan Kaufmann.

Kittler, J. 1986. Feature selection and extraction. In Young, T. Y., and Fu, K. S., eds., *Handbook of Pattern Recognition and Image Processing*. New York, NY: Academic Press.

Michalski, R. S.; Mozetic, I.; Hong, J.; and Lavrac, N. 1986. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 1041–1045. Philadelphia, PA: AAAI Press.

Michalski, R. S. 1983. A theory and methodology of inductive learning. *Artificial Intelligence* 20:111–161.

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

Niblett, T. 1987. Constructing decision trees in noisy domains. In *Proceedings of the Second European Working Session on Learning*, 67–78. Bled, Yugoslavia: Sigma.

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

Quinlan, J. R. 1993b. Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, 236–243. Amherst, MA: Morgan Kaufmann.

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