Multimodal Inductive Reasoning: Combining Rule-Based and Case-Based Learning

Pedro Domingos
Sec. Sistemas, Dept. Eng. Mecânica
Instituto Superior Técnico
Av. Rovisco Pais
Lisbon 1096, Portugal
pedrod@gia.ist.utl.pt
http://www.gia.ist.utl.pt/~pedrod

Abstract

Multimodal inductive reasoning is the combination of multiple learning paradigms in a single system. This article describes RISE, a combination of rule induction and case-based (or instance-based) learning, and uses experiments with synthetic datasets to investigate what is gained by the multimodal approach relative to the individual ones.

Introduction

In recent years, multimodal reasoning has become popular in machine learning under the name of "multistrategy learning." This interest has led to a series of workshops (Michalski & Tecuci 1991; 1993; Michalski & Wnek 1996), an edited volume (Michalski & Tecuci 1994), and two special issues of the *Machine Learning* journal (Michalski 1993; Michalski & Tecuci 1997). As a result, a considerable body of knowledge has been built up in what might be called *multimodal inductive reasoning.*

Initial work in this area concentrated on combining inductive and deductive reasoning (e.g., (Pazzani & Kibler 1992)). More recently, the focus has shifted increasingly to combining different modes of inductive reasoning. For example, the RISE system (Domingos 1996) unifies rule induction (Clark & Niblett 1989) and case-based or instance-based learning (Aha, Kibler, & Albert 1991). 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, case-based/instance-based 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, instance-based 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). 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 1996) is a practical, computationally efficient realization of this idea.[1] 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 (Merz, Murphy, & Aha 1997), resulting in significantly better overall results than either "parent" bias (with C4.5RULES (Quinlan 1993a) and CN2 (Clark & Niblett 1989) 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 be approached by considering 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 1996).

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.

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') ≥ 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. 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.

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

[1] Obviously, it is not the only possible approach to combining the two paradigms (cf. (Branting & Porter 1991; Golding & Rosenbloom 1991; Quinlan 1993b), etc.).

of Stanfill and Waltz's value difference metric for symbolic attributes (Stanfill & Waltz 1986).

Given a training set, RISE can form frontiers that are equivalent to those produced by IBL in some regions and equivalent to those produced by rule induction in others, and transition smoothly between the two. It can also create nonlinear frontiers, which neither method in its basic form is able to do. The following simple example illustrates this. Let the instance space be the plane, and consider a training set composed of the three positive and two negative training examples in Figure 1. From this training set, a typical rule learner will induce a frontier similar to the one shown in Figure 2: a horizontal straight line, with the region below it labeled positive, and the region above it labeled negative. A typical CBL algorithm like the one-nearest-neighbor classifier with Euclidean distance will produce the zigzag frontier shown in Figure 3. RISE will generalize the two negative examples to a straight-line segment with the two examples as the endpoints, and will generalize the three positive examples to another straight line segment, with the rightmost and leftmost examples as endpoints. The resulting frontier is shown in Figure 4. Between points b and c it is a straight line, identical to the frontier produced by the rule learner. To the left of a it is a diagonal line, identical to the frontier produced by the nearest-neighbor classifier; and similarly to the right of d. Between a and b it is an arc of a parabola with it focus at the leftmost negative example. This arc transitions smoothly between the nearest-neighbor frontier to its left and the rule-induction frontier to its right. Between c and d a similar transition occurs. Thus RISE behaves like a case-based classifier in some parts of the instance space, and like a rule learner in others; and it transitions smoothly between the two, creating non-linear frontiers in the process.

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 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 accura-

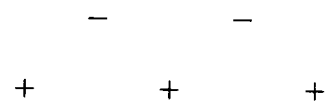

Figure 1: A training set.

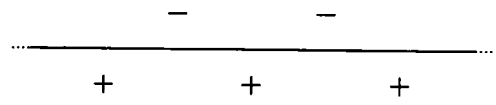

Figure 2: Frontier induced by a rule learner.

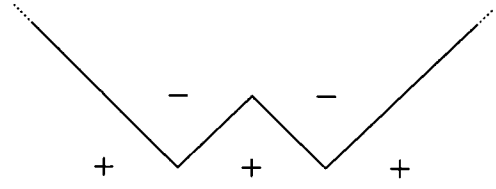

Figure 3: Frontier induced by an instance-based learner.

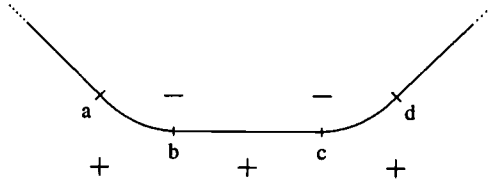

Figure 4: Frontier induced by RISE.

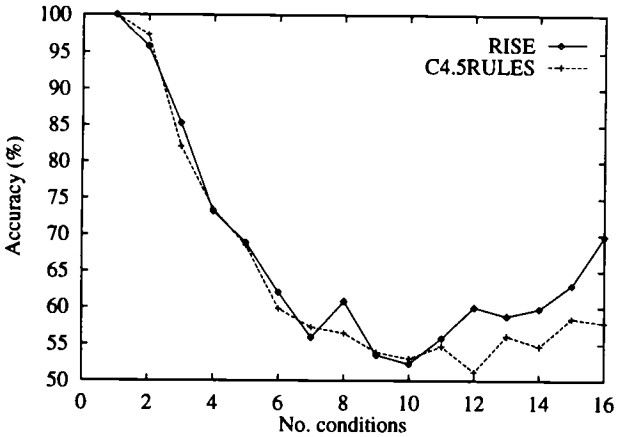

Figure 5: Accuracy as a function of concept specificity.

cies 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 $\text{Min}\{2^{C-1}, 25\}$. 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. 5. 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 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.

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 show that 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, of which two of the most widely known are forward sequential search (FSS) and backward sequential search (BSS) (Devijver & Kittler 1982). 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:

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

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 procedure, 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,[2] 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.

[2]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.

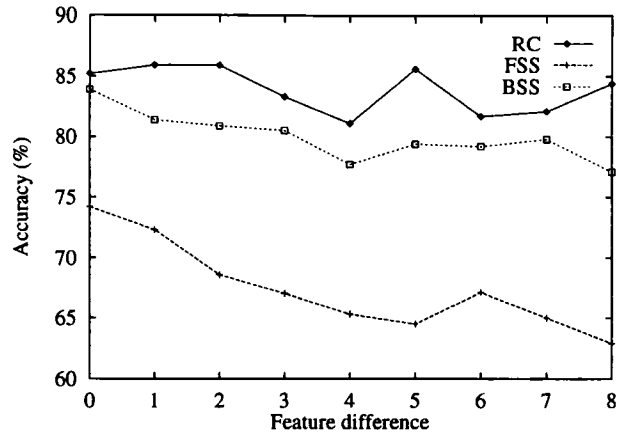

Figure 6: Accuracy as a function of context dependency.

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 6.

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.

Conclusion

In this paper we investigated a form of multimodal inductive reasoning, embodied in the RISE algorithm. RISE combines rule induction and case-based learning, and has been observed to achieve higher accuracies than state-of-the-art representatives of either approach. 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 case-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 case-based learning; 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 paper describes research conducted at the University of California, Irvine, and partly supported by PRAXIS XXI and NATO scholarships. 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.; 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 Niblett, T. 1989. The CN2 induction algorithm. *Machine Learning* 3:261–283.

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

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

Domingos, P. 1996. Unifying instance-based and rule-based induction. *Machine Learning* 24:141–168.

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.

Merz, C. J.; Murphy, P. M.; and Aha, D. W. 1997. UCI repository of machine learning databases. Machine-readable data repository, Department of Information and Computer Science, University of California at Irvine, Irvine, CA.

Michalski, R. S., and Tecuci, G., eds. 1991. *Proceedings of the First International Workshop on Multistrategy Learning.* Harpers Ferry, VA: Center for Artificial Intelligence, George Mason University.

Michalski, R. S., and Tecuci, G., eds. 1993. *Proceedings of the Second International Workshop on Multistrategy Learning.* Harpers Ferry, VA: Office of Naval Research/George Mason University.

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

Michalski, R. S., and Tecuci, G. 1997. Introduction, second special issue on multistrategy learning. *Machine Learning* 27(3):205–208.

Michalski, R. S., and Wnek, J., eds. 1996. *Proceedings of the Third International Workshop on Multistrategy Learning.* Harpers Ferry, VA: AAAI Press.

Michalski, R. S. 1993. Introduction, special issue on multistrategy learning. *Machine Learning* 11(2-3):109–110.

Pazzani, M., and Kibler, D. 1992. The utility of knowledge in inductive learning. *Machine Learning* 9:57–94.

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.