Learning Multiple Relational Rule-based Models

Kamal Ali, Clifford Brunk and Michael Pazzani
Department of Information and Computer Science,
University of California, Irvine, CA, 92717
{ali,brunk,pazzani}@ics.uci.edu
714-725-3491, 714-856-5888

Abstract

We present a method for learning multiple relational models for each class in the data. Bayesian probability theory offers an optimal strategy for combining classifications of the individual concept descriptions. Here we use a tractable approximation to that theory. Previous work in learning multiple models has been in the attribute-value realm. We show that stochastically learning multiple relational (first-order) models consisting of a ruleset for each class also yields gains in accuracy when compared to the accuracy of a single deterministically learned relational model. In addition we show that learning multiple models is most helpful when the hypothesis space is "flat" with respect to the gain metric used in learning.

1 Introduction

There has been much work in learning relational models of data in recent years (e.g. FOIL: Quinlan, 1990; FOCL: Pazzani & Kibler, 1992; CLINT: De Raedt, 1992). Here we present results that combine learning first-order concept descriptions with Bayesian probability theory (e.g. Buntine, 1990) which stipulates that in order to maximize accuracy one should use all hypotheses in the hypothesis space not just a single description. The descriptions vote with weight equal to their posterior probability (given the data). Descriptions that are highly probable represent a "good fit" to the data and so are given higher weight. Although learning multiple descriptions reduces human comprehensibility, it is important in situations where additional data are hard to obtain and each percentage point in accuracy is important.

There are two results of this paper. The first is that multiple concept descriptions are particularly helpful in searching "flat" hypothesis spaces. Briefly, a search space if flat with respect to a learning metric (e.g. information gain) if there are many equally good ways to grow a rule, each candidate being ranked similarly by that learning metric. The second result is experimental evidence that learning multiple *rule sets* yields more accurate classifications than learning multiple rules (Kononenko and Kovacic, 1992). We also present results on learning recursive concepts in the context of learning multiple models. To demonstrate these results, we adapt a relational learning algorithm (HYDRA, Ali & Pazzani, 1993) to learn multiple models, yielding HDYRA-MM. HYDRA-MM learns multiple models, each

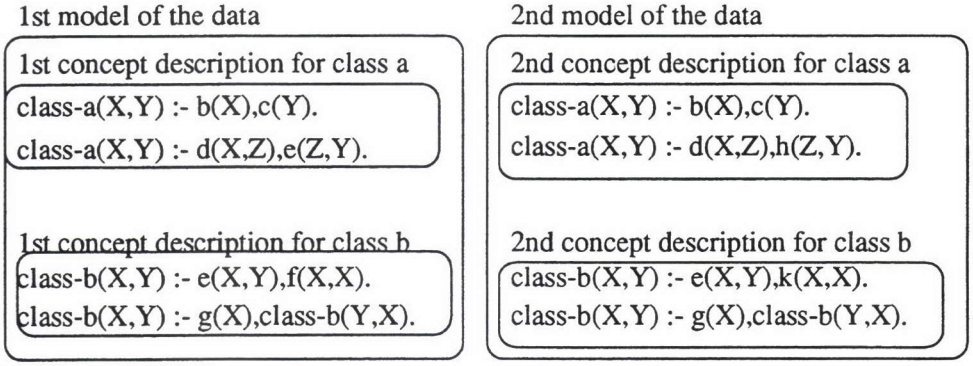


Figure 1: HYDRA-MM learns many concept descriptions for each class and combines their classifications to produce a classification for the class.

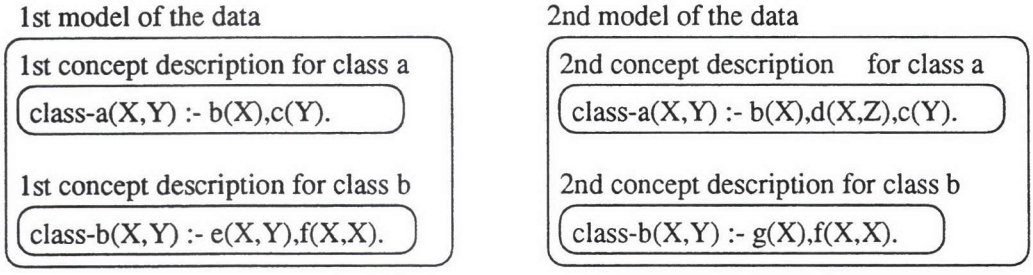
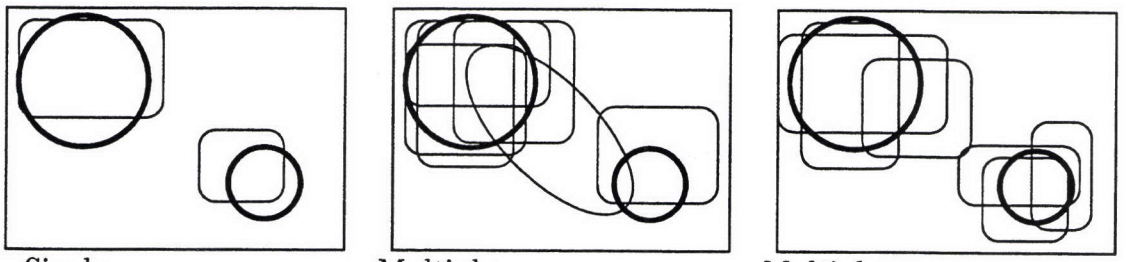


Figure 2: The multiple rules approach learns several approximations for each class, each approximation consisting of a single rule.

consisting of a rule set for each class. Because HYDRA is a *relational* learning algorithm, it can learn the more expressive first-order rules rather than just attribute-value rules.

The learning task requires as input (1) a collection of examples belonging to a set of specified classes (e.g. *class-a, class-b*) which partition the example space and (2) a set of *background relations* (e.g. *a..k*) for which full definitions are provided to the learning algorithm. The task then is to build a concept description for each class using combinations of the background relations as in Figure 1.

Although previous work in machine learning (e.g. Buntine (1990); Smyth and Goodman (1992)) has shown that the predictive accuracy of a classifier can be increased by learning multiple models from the data, there has been no work in learning concept descriptions consisting of rule sets or in learning relational descriptions. Previous work in learning multiple models, each consisting of a single rule (Kononenko & Kovacic) is limited by the basic assumption that that each class can be accurately described by a single, purely conjunctive rule (Figure 2). In our approach each concept description consists of many rules for each class that describe different modes or subclasses within the class (Figure 1). Past success of systems like FOIL (Quinlan, 1990) which learns many rules for a concept suggests such descriptions are appropriate for many problems. Figure 3 shows the problem with the multiple rules approach in trying to learn a concept with multiple modes (subclasses or disjuncts): a rule for the minor class cannot be learned well because examples of the major disjunct have not been removed from the training set.



Single Description | Multiple Rules | Multiple Rulesets

Figure 3: Comparison of 3 algorithms trying to learn on a domain where the first class consists of 2 disjuncts (dark circles). The area outside the dark circles corresponds to the other class. Light lines show coverage of rules learned by the 3 algorithms.

Table 1: Pseudo-code for FOIL.

```
FOIL(POS,NEG,Metric):
 Let POS be the positive examples.
 Let NEG be the negative examples.
 Set NewClause to empty.
 Until POS is empty do:
  Separate: (begin a new clause)
  Remove from POS all positive examples that
        satisfy NewClause.
  Reset NewClause to empty.
  Until NEG is empty do:
   Conquer: (build a clause body)
    Choose a literal L using Metric
    Conjoin L to NewClause.
    Remove from NEG examples that dont satisfy
          NewClause.
```

2 Learning in HYDRA

HYDRA uses a separate and conquer control strategy based on FOIL (Quinlan, 1990) in which a rule is learned and then the training examples covered by that rule are separated from the training set. Subsequent rules are learned on the remaining data. The pseudo-code for FOIL is presented in Table 1. FOIL begins to learn a rule for say *class-a(X, Y)* by starting with the rule with the empty body (this rule is true for all positive and negative examples):

$$class\text{-}a(X,Y) \leftarrow$$

Then, it ranks each literal by the information that would be gained if the empty body was replaced by that literal (Table 2). ϵ in Table 2 denotes the empty rule body. If p and n denote the numbers of positive and negative training examples covered by some rule, their information content (Quinlan, 1990) is defined to be

$$I(p,n) = -log_2 \frac{p}{p+n}$$

Table 2: Ranking literals by information gain.

Rule	Num. positive covered	Num. negative covered	Information content	Gain
class-a$(X,Y) \leftarrow \epsilon$	10	10	1.0	0.0
class-a$(X,Y) \leftarrow b(X)$	9	3	0.415	5.26
class-a$(X,Y) \leftarrow c(Y)$	8	2	0.322	5.42

and the information gain is defined to be

$$gain = p \times (I(p_0, n_0) - I(p, n))$$

where p_0 and n_0 denote the numbers of positive and negative examples covered by the rule before addition of the current literal. (justifications for these measures and more details can be found in (Quinlan, 1990)). FOIL adds the literal which yields the largest gain and resets p_0 and n_0 to reflect the numbers of positive and negative examples covered (after addition of the new literal). FOIL contines to add literals until the rule covers no negative examples. Then, it removes positive examples covered by the rule and learns subsequent rules with the reduced set of examples. The process terminates when no positive examples are left in the training set.

HYDRA calls FOIL for each class in the data, treating positive examples of other classes as negative for the current class. The major difference between HDYRA and FOIL is that HYDRA attaches a reliability measure (such as coverage over the training data) to each rule. In order to classify a test example if that test example satisfies rules from more than one class, HYDRA chooses the class corresponding to the satisfied rule with the highest reliability. Ali & Pazzani (1993) show that this modification to FOIL makes the system much more accurate in noisy domains. In this paper, we use a Laplace estimate of the accuracy of a rule (estimated from the training examples) as its estimate of reliability. The Laplace accuracy of a rule covering p positive and n negative training examples is $(p+1)/(p+n+2)$. The advantage of using this rather than the maximum likelihood estimate $(p/(p+n))$ is that it does not assign an accuracy of 1 to a rule covering just a small number of positive examples and no negative examples. The reliability of such rules is quite different from that of a rule covering say a hundred positive and no negative examples. Yet, both would have the same maximum likelihood accuracy estimate. Among rules covering no negative examples, the Laplace estimate rises monotonically with increasing coverage of positive examples.

3 Learning multiple models with HYDRA

Bayesian theory recommends making classifications based on voting from all concept descriptions in the hypothesis space but in practice we want to find a few highly probable concept descriptions. Using the notation in Buntine (1990), let c be a class, $\mathcal{T}$ be the set of hypotheses (descriptions) the learning algorithm has produced, x be a test example and $\vec{x}$ denote the training examples. Then, we should assign x to c that maximizes

$$pr(c|x,\mathcal{T}) = \sum_{T in \mathcal{T}} pr(c|x,T)pr(T|\vec{x}) \quad (1)$$

Stochastic search is used by HDYRA-MM to find such descriptions. Ideally, one would want the n most probable concept descriptions with their probability evaluated globally, but the search would be intractable. HYDRA searches for literals whose addition to the rule currently being learned would maximize the posterior probability of the new rule. The posterior probability of a rule covering p positive examples out of P positive training examples, and n negative examples out of N negative training examples is

$$pr(p, n, P-p, N-n) \propto pr(T) \times \frac{B(p+\alpha_1, n+\alpha_2)}{B(\alpha_1, \alpha_2)} \times \frac{B(P-p+\alpha_1, N-n+\alpha_2)}{B(\alpha_1, \alpha_2)} \quad (2)$$

where $pr(T)$ is the prior probability of the clause, B is the beta function and α_1 and α_2 are parameters. The posterior probability of the concept description is the product of the middle terms of the right side of equation 2.

During the process of deciding which literal (test) to add to the body of a rule being learned, HYDRA-MM stores the top *MAX-BEST* literals (ranked in terms of how much they contribute to the posterior probability of the current rule). HYDRA-MM then chooses a literal stochastically from this set; the probability of a literal being chosen is equal to the amount of contribution that literal makes to the posterior probability of the rule. Thus, HYDRA-MM conducts a greedy search; searching for the rules with the highest posterior probability given the data.

Classification in HYDRA-MM: To compute the degree of belief that a test example belongs to some class, all concept descriptions of that class that had at least one rule satisfied by the example are considered. From each description, the training accuracy of the most reliable rule (from that description) is multiplied by the probability of that description. These products are summed over all descriptions of that class to yield a degree of belief for that class. Finally, the example is classified to the class with the highest degree of belief. If no rules from any description are satisfied by the example, HYDRA-MM classifies the example to the most frequent class (estimated from the training data).

4 Experimental Results

The goals of this research are to determine the conditions under which learning multiple descriptions yields the greatest increase in accuracy and to provide some evidence that learning multiple rulesets yields more accurate classifications than learning multiple rules. Our hypothesis is that multiple concept descriptions help in domains where no single literal has much higher gain than others. This "flatness" is defined to be the percentage of attempts at adding a literal during learning in which more than one literal had the highest posterior probability. In such a situation, the greedy deterministic learner is at a disadvantage because it cannot explore both search paths: it must choose one literal and forget about the other equally good alternative. Table 3 indicates that multiple concept descriptions are most helpful in the DNA promoters domain which also has a very "flat" hypothesis space.

We tested on problems that are traditionally tackled using attribute-value learners (e.g. Promoters, Lymphography) as well as relational problems (King Rook King (Muggleton *et al.*, 1989), Students (Pazzani & Brunk, 1991) and Document (Esposito *et al.*, 1992).

Recursive concept descriptions- In the document task (Esposito *et al.*, 1992) the problem is to determine whether an example representing a document or a part of a document is a "date-block". A "date-block" is that part of a document which contains the date (other parts being the body, the signature-block etc). This domain is used for automatic

Table 3: Accuracies obtained by learning multiple models. The numbers in the 3rd-last column give the significance level (SIG) at which 11 concept descriptions (CDs) accuracy differs from that of 1 deterministic description according to the paired 2-tailed t-test. NS - not significant, NA - t-test not applicable. KRK 160,20 denotes learning from 160 training examples with 20% artificial class noise on the King-Rook-King task. *MAX-BEST* is 2.

Domain	Deter-ministic	1 CD	2 CDs	5 CDs	11 CDs	SIG.	ties	# Train eg.s
Promoters	65.1	75.0	77.2	85.9	84.8	NA	24.0%	105
Lymphography	79.3	80.5	80.5	82.0	82.8	98	21.1%	99
Cancer	71.2	71.5	71.1	71.1	71.1	NS	9.3%	181
KRKP	94.6	94.3	94.8	94.8	94.9	NS	18.7%	200
Document	98.3	97.4	98.4	99.0	99.5	NA	25.2%	220
Students	86.1	85.4	86.9	88.9	90.4	99	7.3%	100
KRK 160,20	92.3	91.6	91.8	91.9	92.0	NS	7.2%	160
KRK 320,20	95.5	94.9	95.3	95.6	95.5	NS	5.8%	320

classification (using optical character recognition and a learned relational rule-base) of documents into types of documents such as letters, orders, etc. An example is a symbol (such as `block-29`) and its attributes can be obtained using background relations provided to the learner (such as the 1-arity relation `height-small(Block)`). For this domain, HYDRA and HYDRA-MM learned recursive rules such as (in Prolog notation):

$$\text{date-block(X)} \leftarrow \text{to-the-right(X,Y)}, \text{date-block(Y)}.$$

During learning, the extensional definition of *date-block* is used. That is, the recursive call to *date-block* is satisfied if the value bound to variable Y is present in the set of positive examples (the extensional definition) for *date-block*. During classification of test examples, to determine if the recursive call is satisfied, we use the set of rules (the intensional description) learned for *date-block*. One issue that arises when recursive descriptions are mixed with multiple models is whether in order to check if the recursive call succeeds one should check all models of *date-block* or just the current model. That is, if the rule being matched against a test example comes from the i-th model, then to determine if a recursive call succeeds, should one check just rules of the i-th model or those of all models? We chose the former option in order to maintain the independence of the models. Table 3 shows that using multiple models in this way leads to a significant increase in accuracy. We also have to ensure that the matching process of a ruleset to an example terminates. Currently, we employ an absolute depth-limit in the SLD resolution tree to ensure termination.

Multiple rulesets versus rules- To compare multiple rule sets to multiple rules, we adapt HYDRA to produce HYDRA-R which stochastically learns multiple concept descriptions, each consisting of one rule. HYDRA-R was only able to achieve a 87.9% accuracy on the KRK domain with 320 examples and 20% class noise, whereas HYDRA-MM achieves 94.7%. This is because it is known that that a single purely conjunctive rule is insufficient to describe the KRK concept (from the standard set of relations used for this domain). HYDRA and HYDRA-MM are successful on the KRK domain because they learn many rules to describe each class in this domain. Because they remove training exmaples belonging to

the major disjunct they are better able to learn rules for the other disjuncts. Even though HYDRA-R learns many rules, it does not separate examples like HYDRA does and so is unable to model the minor disjuncts well. This illustrates that for some domains, it is necessary to learn descriptions consisting of rule sets. Learning multiple descriptions, each consisting of a single rule, is no substitute for the separate and conquer approach necessary for such problems.

5 Conclusion

We have characterized an experimental property (hypothesis space flatness) which indicates when learning of multiple models is useful. We have some empirical evidence that learning multiple concept descriptions consisting of rule sets yields more accurate classifications than learning multiple descriptions consisting of rules. Another outcome of this work was learning of multiple models containing recursive rules which also were more accurate than a single recursive description. Finally, we have presented a system that makes classifications in a theoretically sound manner and uses a tractable approximation of the Bayes theory to learn multiple relational models consisting of rule sets.

References

Ali K. and Pazzani M. (1993). HYDRA: A Noise-tolerant Relational Concept Learning Algorithm. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence.* Chambery, France: Morgan Kaufmann.

Pazzani M. and Brunk C. (1991). Detecting and correcting errors in rule-based expert systems: an integration of empirical and explanation-based learning. *Knowledge Acquisition, 3*, 157-173.

Buntine W. (1990). *A Theory of Learning Classification Rules.* Doctoral dissertation. School of Computing Science, University of Technology, Sydney, Australia.

De Raedt L. (1992). *Interactive Theory Revision: an Inductive Logic Programming Approach.* Academic Press.

Esposito F., Malerba D. and Semeraro G. (1992). Classification in Noisy Environments Using a Distance Measure Between Structural Symbolic Descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 3.

Kononenko I. and Kovacic M. (1992). Learning as Optimization: Stochastic Generation of Multiple Knowledge. In *Machine Learning: Proceedings of the Ninth International Workshop.* Aberdeen, Scotland. Morgan Kaufmann.

Kwok S. and Carter C. (1990). Multiple decision trees. *Uncertainty in Artificial Intelligence, 4*, 327-335.

Muggleton S., Bain M., Hayes-Michie J. and Michie D. (1989). An experimental comparison of human and machine-learning formalisms. In *Proceedings of the Sixth International Workshop on Machine Learning.* Ithaca, NY. Morgan Kaufmann.

Pazzani M. and Kibler D. (1991). The utility of knowledge in inductive learning. *Machine Learning, 9, 1*, 57-94.

Quinlan R. (1990). Learning logical definitions from relations. *Machine Learning, 5, 3.*

Smyth P. and Goodman R. (1992). Rule Induction Using Information Theory. In G. Piatetsky-Shapiro (ed.) *Knowledge Discovery in Databases*, Menlo Park, CA: AAAI Press, MIT Press.