N94-34044

Finding Accurate Frontiers: A Knowledge-Intensive Approach to Relational Learning

Michael Pazzani and Clifford Brunk

Department of Information and Computer Science
University of California
Irvine, CA 92717
pazzani@ics.uci.edu, brunk@ics.uci.edu

Abstract

An approach to analytic learning is described that searches for accurate entailments of a Horn Clause domain theory. A hill-climbing search, guided by an information based evaluation function, is performed by applying a set of operators that derive frontiers from domain theories. The analytic learning system is one component of a multi-strategy relational learning system. We compare the accuracy of concepts learned with this analytic strategy to concepts learned with an analytic strategy that operationalizes the domain theory.

Introduction

There are two general approaches to learning classification rules. Empirical learning programs operate by finding regularities among a group of training examples. Analytic learning systems use a domain theory[1] to explain the classification of examples, and form a general description of the class of examples with the same explanation. In this paper, we discuss an approach to learning classification rules that integrates empirical and analytic learning methods. The goal of this integration is to create concept descriptions that are more accurate classifiers than both the original domain theory (which serves as input to the analytic learning component) and the rules that would arise if only the empirical learning component were used. We describe a new analytic learning method that returns a frontier (i.e., conjunctions and disjunctions of operational[2] and non-operational literals) instead of an operationalization (i.e., a conjunction of operational literals) and we demonstrate there is an accuracy advantage in allowing an analytic learner to dynamically select the level of generality of the learned concept, as a function of the training data.

In previous work (Pazzani, et al., 1991; Pazzani & Kibler, 1992), we have described FOCL, a system that extends Quinlan's (1990) FOIL program in a number of ways, most significantly by adding a compatible explanation-based learning (EBL) component. In this paper we provide a brief review of FOIL and FOCL, then discuss how operationalizing a domain theory can adversely affect the accuracy of a learned concept. We argue that instead of operationalizing a domain theory, an analytic learner should return the most general implication of the domain theory, provided this implication is not less accurate than any more specialized implication. We discuss the computational complexity of an algorithm that enumerates all such descriptions and then describe a greedy algorithm that efficiently addresses the problem. Finally, we present a variety of experiments that indicate replacing the operationalization algorithm of FOCL with the new analytic learning method results in more accurate learned concept descriptions.

FOIL

FOIL learns classification rules by constructing a set of Horn Clauses in terms of known operational predicates. Each clause body consists of a conjunction of literals that cover some positive and no negative examples. FOIL starts to learn a clause body by finding the literal with the maximum information gain, and continues to add literals to the clause body until the clause does not cover any negative examples. After learning each clause, FOIL removes from further consideration the positive examples covered by that clause. The learning process ends when all positive examples have been covered by some clause.

FOCL

FOCL extends FOIL by incorporating a compatible EBL component. This allows FOCL to take advantage of an initial domain theory. When constructing a clause body, there are two ways that FOCL can add literals. First, it can create literals via the same empirical method used by FOIL. Second, it can create literals by operationalizing a target concept, i.e., a non-operational definition of the concept to be learned (Mitchell, et al., 1986). FOCL uses FOIL's information-based evaluation function to determine whether to add a literal learned empirically or a conjunction of literals learned analytically. In general FOCL learns clauses of the form $r \leftarrow O_i \wedge O_d \wedge O_f$ where O_i is an initial conjunction of operational literals learned empirically, O_d is a conjunction of literals found by operationalizing the domain theory, and O_f is a final conjunction of literals learned empirically[3]. Pazzani, et al. (1991) demonstrate

1. We use *domain theory* to refer to a set of Horn-Clause rules given to a learner as an approximate definition of a concept and *learned concept* to refer to the result of learning.

2. We use the term *operational* to refer to predicates that are defined *extensionally* (i.e., defined by a collection of facts). However, the results apply to any satirically determined definition of operationality.

3. Note the target concept is operationalized at most once per clause and that either O_i, O_d, or O_f may be empty.

that FOCL can utilize incomplete and incorrect domain theories. We attribute this capability to its uniform use of an evaluation function to decide whether to include literals learned empirically or analytically.

Operationalization in FOCL differs from that of most EBL programs in that it uses a set of positive and negative examples, rather than a single positive example. A non-operational literal is operationalized by producing a specialization of a domain theory that is a conjunction of operational literals. When there are several ways of operationalizing a literal (i.e., there are multiple, disjunctive clauses), the information gain metric is used to determine which clause should be used by computing the number of examples covered by each clause. Figure 1 displays a typical domain theory with an operationalization $(f \wedge g \wedge h \wedge k \wedge l \wedge p \wedge q)$ represented as bold nodes.

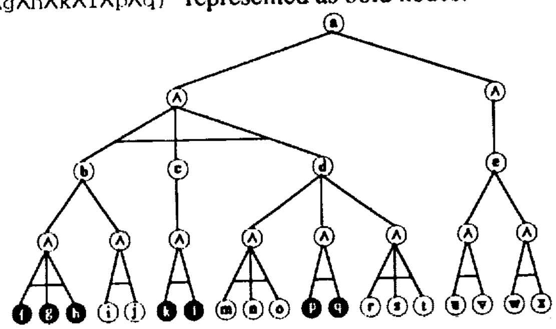

Figure 1. The bold nodes represent one operationalization $(f \wedge g \wedge h \wedge k \wedge l \wedge p \wedge q)$ of the domain theory. In standard EBL, this path would be chosen if it were a proof of a single positive example. In FOCL, this path would be taken if the choice made at a disjunctive node had greater information gain (with respect to a set of positive and negative examples) than alternative choices.

Operationalization

The operationalization process yields a specialization of the target concept. Indeed, several systems designed to deal with overly general theories rely on the operationalization process to specialize domain theories (Flann & Dietterich, 1990; Cohen, 1992). However, fully operationalizing a domain theory can result in several problems:

1. Overspecialization of correct non-operational concepts. For example, if the domain theory in Figure 1 is completely correct, then a correct operational definition will consist of eight clauses. However, if there are few examples, or some combinations of operationalizations are rare, then there may not be a positive example corresponding to all combinations of all operationalizations of non-operational predicates. As a consequence, the learned concept may not include some combinations of operational predicates (e.g., $i \wedge j \wedge k \wedge l \wedge r \wedge s \wedge t$), although there is no evidence that these specializations are incorrect.
2. Replication of empirical learning. If there is a literal omitted from a clause of a non-operational predicate, then this literal will be omitted from each operationalization involving this predicate. For example, if the domain theory in Figure 1 erroneously contained the rule $b \leftarrow f \wedge h$ instead of $b \leftarrow f \wedge g \wedge h$, then each operationalization of the target concept using this predicate (i.e., $f \wedge h \wedge k \wedge l \wedge m \wedge n \wedge o$, $f \wedge h \wedge k \wedge l \wedge p \wedge q$, and $f \wedge h \wedge k \wedge l \wedge r \wedge s \wedge t$) will contain the same omission. FOCL can recover from this error if its empirical component can find the omitted literal, g. However, to obtain a correct learned concept description, FOCL would have to find the same condition independently three times on three different sets of examples. This replication of empirical learning is analogous to the replicated subtree problem in decision trees (Pagallo & Haussler, 1990). This problem should be most noticeable when there are few training examples. Under this circumstance, it is unlikely that empirical learning on several arbitrary partitions of a data set will be as accurate as learning from the larger data set.
3. Proofs involving incorrect non-operational predicates may be ignored. If the definition of a non-operational predicate (e.g., c in Figure 1) is not true of any positive example, then the analytic learner will not return any operationalization using this predicate. This reduces the usefulness of the domain theory for an analytic learner. For example, if c is not true of any positive example, then FOCL as previously described can find only two operationalizations: $u \wedge v$ and $w \wedge x$. Again, we anticipate that this problem will be most severe when there are few training examples. With many examples, the empirical learner can produce accurate clauses that mitigate this problem.

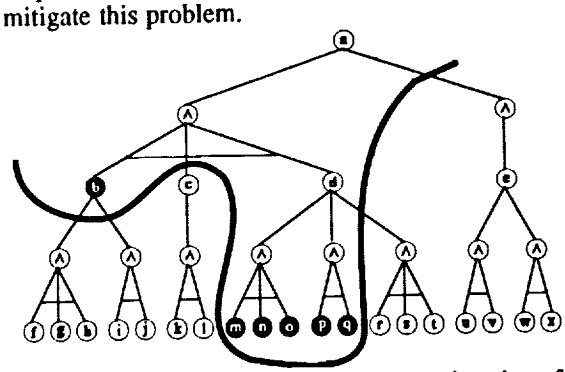

Figure 2. The bold nodes represent one frontier of the domain theory, $b \wedge ((m \wedge n \wedge o) \vee (p \wedge q))$.

Frontiers of a Domain Theory

To address the problems raised in the previous section, we propose an analytic learner that does not necessarily fully operationalize target concepts. Instead, the learner returns a *frontier* of the domain theory. A frontier differs from an operationalization of a domain theory in three ways. The frontier represented by those nodes immediately above the line in Figure 2, $b \wedge ((m \wedge n \wedge o) \vee (p \wedge q))$, illustrates these differences:

1. Non-operational predicates (e.g., b) can appear in the frontier.

2. A disjunction of two or more clauses that define a non-operational predicate (e.g., $(m \wedge n \wedge o) \vee (p \wedge q)$) can appear in the frontier.
3. A frontier does not necessarily include all literals in a conjunction (e.g., neither c, nor any specialization of c, appears in the frontier).

Combined, the first two distinguishing features of a frontier address the first two problems associated with operationalization. Overspecialization of correct non-operational concepts can be avoided if the analytic component returns a more general concept description. Similarly, replication of empirical learning can be avoided if the analytic component returns a frontier more general than an operationalization. For example, if the domain theory in Figure 2 erroneously contained the rule $b \leftarrow f \wedge h$ instead of $b \leftarrow f \wedge g \wedge h$ and frontier $f \wedge h \wedge k \wedge l \wedge d$ was returned, then an empirical learner would only need to be invoked once to specialize this conjunction by adding g. Of course, if one of the clauses defining d were incorrect, it would make sense to specialize d. However, operationalization is not the only means of specialization. For example, if the analytic learner returned $f \wedge h \wedge k \wedge l \wedge ((m \wedge n \wedge o) \vee (p \wedge q))$, then replication of induction problem could also be avoided. This would be desirable if the clause $d \leftarrow r \wedge s \wedge t$ were incorrect.

The third problem with operationalization can be addressed by removing some literals from a conjunction. For example, if no positive examples use $a \leftarrow b \wedge c \wedge d$ because c is not true of any positive example, then the analytic learner might want to consider ignoring c and trying $a \leftarrow b \wedge d$. This would allow potentially useful parts of the domain theory (e.g. b and d) to be used by the analytic learner, even though they may be conjoined with incorrect parts.

The notion of a frontier has been used before in analytic learning. However, the previous work has assumed that the domain theory is correct and has focused on increasing the utility of learned concepts (Hirsh, 1988; Keller, 1988; Segre, 1987) or learning from intractable domain theories (Braverman & Russell, 1988). Here, we do not assume that the domain theory is correct.

We argue that to increase the accuracy of learned concepts, an analytic learner should have the ability to select the generality of a frontier derived from a domain theory. To validate our hypothesis, we will replace the operationalization procedure in FOCL with an analytic learner that returns a frontier. In order to avoid confusion with FOCL, we use the name FOCL-FRONTIER to refer to the system that combines this new analytic learner with an empirical learning component based on FOIL. In general, FOCL-FRONTIER learns clauses of the form $r \leftarrow O_i \wedge F_d \wedge O_f$ where O_i is an initial conjunction of operational literals learned empirically, F_d is a frontier of the domain theory, and O_f is a final conjunction of literals learned empirically. We anticipate that due to its use of a frontier rather than an operationalization, FOCL-FRONTIER will be more accurate than FOCL, particularly when there are few training examples or the domain theory is very accurate.

Enumerating Frontiers of a Domain Theory

Formally, a frontier can be defined as follows. Let b represent a conjunction of literals and p represent a single literal.

1. The target concept is a frontier.
2. A new frontier can be formed from an existing frontier by replacing a literal p with $b_1 \vee ... \vee b_i \vee ... \vee b_n$ provided there are rules $p \leftarrow b_1, ..., p \leftarrow b_i, ..., p \leftarrow b_n$.
3. A new frontier can be formed from an existing frontier by replacing a disjunction $b_1 \vee ... \vee b_{i-1} \vee b_i \vee b_{i+1} \vee ... \vee b_n$ with $b_1 \vee ... \vee b_{i-1} \vee b_{i+1} \vee ... \vee b_n$ for any i. This deletes b_i.
4. A new frontier can be formed from an existing frontier by replacing a conjunction $p_1 \wedge ... \wedge p_{i-1} \wedge p_i \wedge p_{i+1} \wedge ... \wedge p_n$ with $p_1 \wedge ... \wedge p_{i-1} \wedge p_{i+1} \wedge ... \wedge p_n$ for any i. This deletes p_i.

One approach to analytic learning would be to enumerate all possible frontiers. The information gain of each frontier could be computed, and if the frontier with the maximum information gain has greater information gain than any literal found empirically, then this frontier would be added to the clause under construction. Such an approach would be impractical for all but the most trivial, non-recursive domain theories. Since each frontier specifies a unique combination of leaf nodes of an and-or tree (i.e., selecting all leaves of a subtree is equivalent to selecting the root of the subtree and selecting no leaves of a subtree is equivalent to deleting the root of a subtree), there are 2^k frontiers of a domain theory that has k nodes in the and/or tree. For example, if every non-operational predicate has n clauses, each clause is a conjunction of m literals, and inference chains have a depth of d *and*-nodes, then the number of frontiers is $2^{m^d n^d}$.

Deriving Frontiers from the Target Concept

Due to the intractability of enumerating all possible frontiers, we propose a heuristic approach based upon hill-climbing search. The frontier is initialized to the target concept. A set of transformation operators is applied to the current frontier to create a set of possible frontiers. If none of the possible frontiers has information gain greater than that of the current frontier[4], then the current frontier is returned. Otherwise, the potential frontier with the maximum information gain becomes the current frontier and the process of applying transformation operators is repeated. The following transformation operators are used[5]:

- *Clause specialization*:
  If there is a frontier containing a literal p, and there are exactly n rules of the form $p \leftarrow b_1, ..., p \leftarrow b_i, ..., p \leftarrow b_n$, then n frontiers formed by replacing p with b_i are evaluated.

4. The information gain of a frontier is calculated in the same manner than Quinlan (1990) calculates the information gain of a literal: by counting the number of positive and negative examples that meet the conditions represented by the frontier.

5. The numeric restrictions placed upon the applicability of each operator are for efficiency reasons (i.e., to ensure that each unique frontier is evaluated only once).

- *Specialization by removing disjunctions*:
  a. If there is a frontier containing a literal p, and there are n rules of the form $p \leftarrow b_1, \ldots, p \leftarrow b_i, \ldots, p \leftarrow b_n$, then n frontiers formed by replacing p with $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_n$ are evaluated (provided n>2).
  b. If there is a frontier containing a disjunction $b_1 \vee \ldots \vee b_{i-1} \vee b_i \vee b_{i+1} \vee \ldots \vee b_m$, then m frontiers replacing this disjunction with $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_m$ are evaluated (provided m>2).
- *Generalization by adding disjunctions*:
  If there is a frontier containing a (possibly trivial) disjunction of conjunction of literals $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_m$ and there are rules of the form $p \leftarrow b_1, \ldots, p \leftarrow b_{i-1}, p \leftarrow b_i, p \leftarrow b_{i+1}, \ldots, p \leftarrow b_n$ and m<n-1, then n−m frontiers replacing the disjunction $b_1 \vee \ldots \vee b_{i-1} \vee b_{i+1} \vee \ldots \vee b_m$ with $b_1 \vee \ldots \vee b_{i-1} \vee b_i \vee b_{i+1} \vee \ldots \vee b_m$ are evaluated. This is implemented efficiently by keeping a derivation of each frontier, rather than by searching for frontiers matching this pattern.
- *Generalization by literal deletion*:
  If there is a frontier containing a conjunction of literals $p_1 \wedge \ldots \wedge p_{i-1} \wedge p_i \wedge p_{i+1} \wedge \ldots \wedge p_n$, then n frontiers replacing this conjunction with $p_1 \wedge \ldots \wedge p_{i-1} \wedge p_{i+1} \wedge \ldots \wedge p_n$ are evaluated.

There is a close correspondence between the recursive definition of a frontier and these transformation operators. However, there is not a one-to-one correspondence because we have found empirically that in some situations it is advantageous to build a disjunction by adding disjuncts and in other cases it is advantageous to build a disjunction by removing disjuncts. The former tends to occur when few clauses of a predicate are correct while the latter tends to occur when few clauses are incorrect.

Note that the first three frontier operators derive logical entailments from the domain theory while the last does not. Deleting literals from a conjunction is a means of finding an abductive hypothesis. For example, in EITHER (Ourston & Mooney, 1990), a literal can be assumed to be true during the proof process of a single example. One difference between FOCL-FRONTIER and the abduction process of EITHER is that EITHER considers all likely assumptions for each unexplained positive example, and FOCL-FRONTIER uses a greedy approach to deletion based on an evaluation of the effect on a set of examples.

Evaluation

In this section, we report on a series of experiments in which we compare FOCL using empirical learning alone (EMPIRICAL), FOCL using a combination of empirical learning and operationalization, and FOCL-FRONTIER. We evaluate the performance of each algorithm in several domains. The goal of these experiments is to substantiate the claim that analytic learning via frontier transformations results in more accurate learned concept descriptions than analytic learning via operationalization. Throughout this paper, we use an analysis of variance to determine if the difference in accuracy between algorithms is significant.

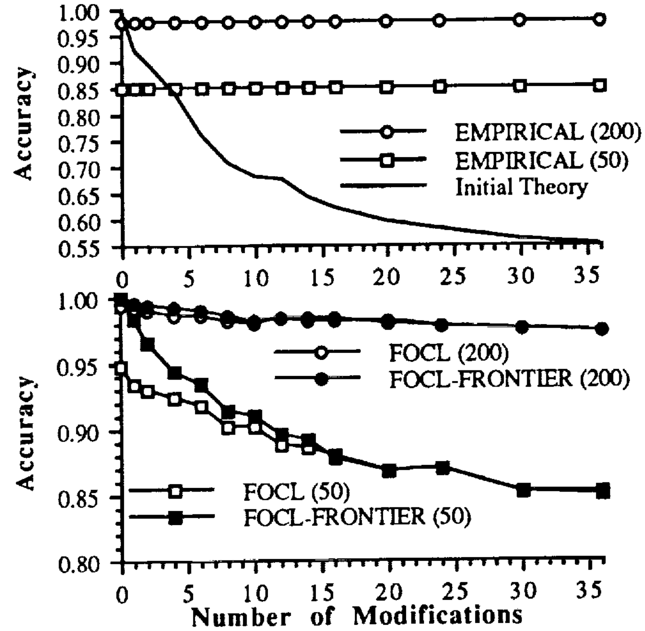

Figure 3. A comparison of FOCL's empirical component (EMPIRICAL), FOCL using both empirical learning and operationalization, and FOCL-FRONTIER in the chess end gain domain. **upper**: The accuracy of EMPIRICAL (given training sets of size 50 and 200) and the average accuracy of the initial theory as a function of the number of changes to the domain theory. **lower**: The accuracy of FOCL and FOCL-FRONTIER on the same data.

Chess End Games

The first problem we investigate is learning rules that determine if a chess board containing a white king, white rook, and black king is in an illegal configuration. This problem has been studied using empirical learning systems by Muggleton, et al. (1989) and Quinlan (1990). Here, we compare the accuracy of FOCL-FRONTIER and FOCL using a methodology identical to that used by Pazzani and Kibler (1992) to compare FOCL and FOIL.

In these experiments the initial theory given to FOCL and FOCL-FRONTIER was created by introducing either 0, 1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 30 or 36 random modifications to a correct domain theory that encodes the relevant rules of chess. Four types of modifications were made: deleting a literal from a clause, deleting a clause, adding a literal to a clause, and adding a clause. Added clauses are constructed with random literals. Each clause contains at least one literal, there is a 0.5 probability that a clause will have at least two literals, a 0.25 probability of containing at least three, and so on.

We ran experiments using 25, 50, 75, 150, and 200 training examples. On each trial the training and test examples were drawn randomly from the set of 8^6 possible board configurations. We ran 32 trials of each algorithm and measured the accuracy of the learned concept description on 1000 examples. For each algorithm the curves for 50 and 200 training examples are presented. Figure 3 (upper) graphs the accuracy of the initial theory and the concept description learned by FOCL's empirical component as functions of the number of modifications to the correct domain theory. Figure 3 (lower) graphs the accuracy of FOCL and FOCL-FRONTIER.

The following conclusions may be drawn from these experiment. First, FOCL-FRONTIER is more accurate than FOCL when there are few training examples. An analysis of variance indicates that the analytic learning algorithm has a significant effect on the accuracy ($p<.0001$) when there are 25, 50 and 75 training examples. However, where there are 150 or 200 training examples, there is no significant difference in accuracy between the analytic learning algorithms because both analytic learning algorithms (as well as the empirical algorithm) are very accurate on this problem with larger numbers of training examples. Second, the difference in accuracy between FOCL and FOCL-FRONTIER is greatest when the domain theory has few errors. With 25 and 50 examples, there is a significant interaction between the number of modifications to the domain theory and the algorithm ($p<.0001$ and $p<.005$, respectively).

During these experiments, we also recorded the amount of work EMPIRICAL, FOCL and FOCL-FRONTIER performed while learning a concept description. Pazzani and Kibler (1990) argue that the number of times information gain is computed is a good metric for describing the size of the search space explored by FOCL. Figure 4 graphs these data as a function of the number of modifications to the domain theory for learning with 50 training examples. FOCL-FRONTIER tests only a small percentage of the 225 frontiers of this domain theory with 25 leaf nodes. The frontier approach requires less work than operationalization until the domain theory is fairly inaccurate. This occurs, in spite of the larger branching factor because the frontier approach generates more general concepts with fewer clauses than those created by operationalization (see Table 1). When the domain theory is very inaccurate, FOCL and FOCL-FRONTIER perform slightly more work than EMPIRICAL because there is a small overhead in determining that the domain theory has no information gain.

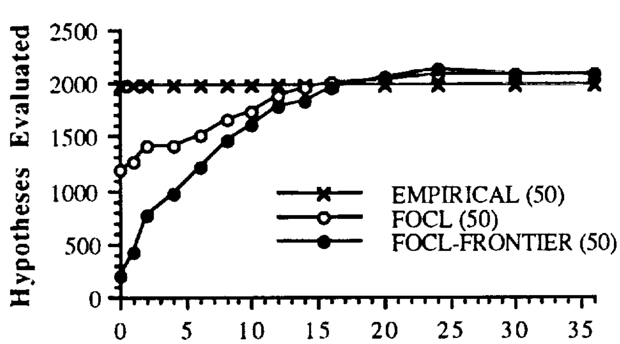

Figure 4: The number of times the information gain metric is computed for each algorithm.

FOCL (92.6% accurate)

```
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKf,WRf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKr,WRr).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←near(WKr,BKr)  ∧
                                 near(WKf,BKf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKr,WKf)  ∧
                                 equal(WKr,BKr)  ∧
                                 near(WKf,BKf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(WKr,WRr)  ∧
                                 equal(WKf,WRf).
```

FOCL-FRONTIER (98.3% accurate)

```
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←k_attack(WKr,WKf,BKr,BKf)  v
                                 r_attack(WRr,WRf,BKr,BKf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←equal(BKf,WRf).
illegal(WKr,WKf,WRr,WRf,BKr,BKf)←same_pos(WKr,WKf,WRr,WRf).
```

Table 1. Typical definitions of `illegal`. The variables refer to the rank and file of the white king, white rook, and the black king. The domain theory was 91.0% accurate and 50 training examples were used.

Educational Loans

The second problem studied involves determining if a student is required to pay back a loan based on enrollment and employment information. This theory was constructed by an honors student who had experience processing loans. This problem, available from the UC Irvine repository, was previously used by an extension to FOCL that revises domain theories (Pazzani & Brunk, 1991). The domain theory is 76.8% accurate on a set of 1000 examples.

We ran 16 trials of FOCL and FOCL-FRONTIER with this domain theory on randomly selected training sets ranging from 10 to 100 examples and measured the accuracy of the learned concept by testing on 200 distinct test examples. The results indicate that the learning algorithm has a significant effect on the accuracy of the learned concept ($p<.0001$). Figure 5 plots the mean accuracy of the three algorithms as a function of the number of training examples.

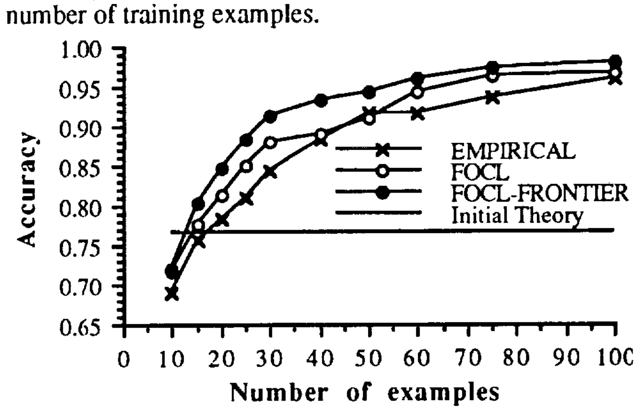

Figure 5. The accuracy of FOCL's empirical component alone, FOCL with operationalization and FOCL-FRONTIER on the student loan data.

Nynex Max

Nynex Max (Rabinowitz, et al., 1991) is an expert system that is used by NYNEX (the parent company of New York Telephone and New England Telephone) at several sites to determine the location of a malfunction for customer-reported telephone troubles. It can be viewed as solving a classification problem where the input is data such as the type of switching equipment, various voltages and resistances and the output is the location to which a repairman should be dispatched (e.g., the problem is in the customer's equipment, the customer's wiring, the cable facilities, or the central office). Nynex Max requires some customization at each site in which it is installed.

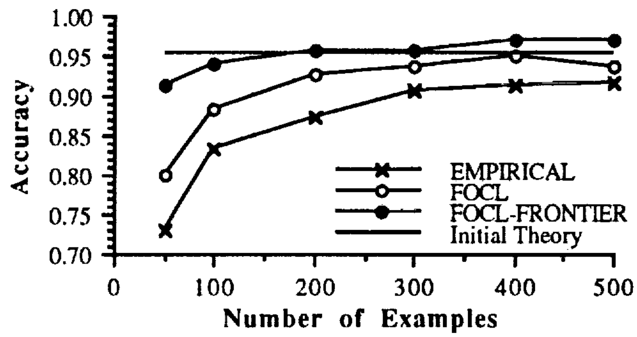

Figure 6. The accuracy of the learning algorithms at customizing the Max knowledge-base.

In this experiment, we compare the effectiveness of FOCL-FRONTIER and FOCL at customizing the Nynex Max knowledge-base. The initial domain theory is taken from one site, and the training data is the desired output of Nynex Max at a different site. Figure 6 shows the accuracy of the learning algorithms (as measured on 200 independent test examples), averaged over 10 runs as a function of the number of training examples. FOCL-FRONTIER is more accurate than FOCL ($p<.0001$). This occurs because the initial domain theory is fairly large (about 75 rules), very disjunctive, and fairly accurate (about 95.4%). Under these circumstances, FOCL requires many examples to form many operational rules, while FOCL-FRONTIER learns fewer, more general rules. FOCL-FRONTIER is the only algorithm to achieve an accuracy significantly higher than the initial domain theory.

Related Work

Cohen (1990; 1991a) describes the ELGIN systems that makes use of background knowledge in a way similar to FOCL-FRONTIER. In particular, one variant of ELGIN called ANA-EBL, finds concepts in which all but k nodes of a proof tree are operational. The algorithm, which is exponential in k, learns more accurate rules from overly general domain theories than an algorithm that uses only operational predicates. A different variant of ELGIN, called K-TIPS, selects k nodes of a proof tree and returns the most general nodes in the proof tree that are not ancestors of the selected nodes. This enables the system to learn a set of clauses containing at most k literals from the proof tree. Some of the literals may be non-operational and some subtrees may be deleted from the proof tree. In some ways, ELGIN is like the optimal algorithm we described above that enumerates all possible frontiers. A major difference is that ELGIN does not allow disjunction in proofs, and for efficiency reasons is restricted to using small values of k. FOCL-FRONTIER is not restricted in such a fashion, since it relies on hill-climbing search to avoid enumerating all possible hypotheses. In addition, the empirical learning component of FOCL-FRONTIER allows it to learn from overly specific domain theories in addition to overly general domain theories.

In the GRENDEL system, Cohen (1991b) uses a grammar rather than a domain theory to generate hypotheses. Cohen shows that this grammar provides an elegant way to describe the hypothesis space searched by FOCL. It is possible to encode the domain theory in such a grammar. In addition, it is possible to encode the hypothesis space searched by FOIL in the grammar. GRENDEL uses a hill-climbing search method similar to the operationalization process in FOCL to determine which hypothesis to derive from the grammar. Cohen (1991b) shows that augmenting GRENDEL with advice to prefer grammar rules corresponding to the domain theory results in concepts that are as accurate as those of FOCL (with operationalization) on the chess end game problem. The primary difference between GRENDEL and FOCL-FRONTIER is that FOCL-FRONTIER contains operators for deleting literals from and-nodes and for incorporating several disjunctions from or-nodes. However, due to the generality of GRENDEL's grammatical approach, it should be possible to extend GRENDEL by writing a preprocessor that converts a domain theory into a grammar that simulate these operators. Here, we have shown that these operators result in increased accuracy, so it is likely that a grammar based on the operators proposed here would increase GRENDEL's accuracy.

FOCL-FRONTIER is in some ways similar to theory revision systems, like EITHER (Ourston & Mooney, 1990). However, theory revision systems have an additional goal of making minimal revisions to a theory, while FOCL-FRONTIER uses a set of frontiers from the domain theory (and/or empirical learning) to discriminate positive from negative examples. EITHER deals with propositional theories and would not be able to revise any of the relational theories used in the experiments here. A more recent theory revision system, FORTE (Richards & Mooney, 1991), is capable of revising relational theories. It has been tested on one problem on which we have run FOCL, the illegal chess problem from Pazzani & Kibler (1992). Richards (1992) reports that with 100 training examples FOCL is significantly more accurate than FORTE (97.9% and 95.6% respectively). For this problem, FOCL-FRONTIER is 98.5% accurate (averaged over 20 trials). FORTE has a problem with this domain, since it contains two overly-general clauses for the same relation and its revision operators assume that at most one clause is overly general. Although it is not possible to draw a general conclusion form this single example, it does indicate that there are techniques for taking advantage of information contained in a theory that FOCL utilizes that are not incorporated into FORTE.

Future Work

Here, we have described one set of general purpose operators that derive frontiers. We are currently experimenting with more special purpose operators designed to handle commonly occurring problems in knowledge-based systems. For example, one might wish to consider operators that negate a literal in a frontier (since we occasionally omit a not from rules) or that change the order of arguments to a predicate. Initial experiments (Pazzani, 1992) with one such operator in FOCL (replacing one predicate with a related predicate) yielded promising results.

Conclusion

In this paper, we have presented an approach to integrating empirical and analytic learning that differs from previous approaches in that it uses an information theoretic metric on a set of training examples to determine the generality of the concepts derived from the domain theory. Although it is possible that the hill-climbing search algorithm will find a local maximum, experimentally we have demonstrated that in situations where there are few training examples, the domain theory is very accurate, or the domain theory is highly disjunctive this approach learns more accurate concept descriptions than either empirical learning alone or a similar approach that integrates empirical learning and operationalization. From this we conclude that there is an advantage in allowing the analytic learner to select the generality of a frontier derived from a domain theory both in terms of accuracy and in terms of the amount of work required to learn a concept description.

Acknowledgments

This research is supported by an Air Force Office of Scientific Research Grant, F49620-92-J-0430, and by the University of California, Irvine through an allocation of computer time. We thank Kamal Ali, William Cohen, Andrea Danyluk, Caroline Ehrlich, Dennis Kibler, Ray Mooney and Jim Wogulis for comments on an earlier draft of this paper.

References

Braverman, M. & Russell, S. (1988). Boundaries of operationality. Proceedings of the Fifth International Conference on Machine Learning (pp. 221–233). Ann Arbor, MI: Morgan Kaufmann.

Cohen, W. (1990). Explanation-based generalization as an abstraction mechanism in concept learning. Technical Report DCS-TR-271 (Ph.D. dissertation). Rutgers University.

Cohen, W. (1991a). The generality of overgenerality. Proceedings of the Eighth International Workshop on Machine Learning (pp. 490–494). Evanston, IL: Morgan Kaufmann.

Cohen, W. (1991b). Grammatically biased learning: Learning Horn Theories using an explicit antecedent description language. AT&T Bell Laboratories Technical Report 11262-910708-16TM (available from the author).

Cohen, W. (1992). Abductive explanation-based learning: A solution to the multiple inconsistent explanation problem. Machine Learning.

Flann, N., & Dietterich, T. (1990). A study of explanation-based methods for inductive learning. Machine Learning, 4, 187–226.

Hirsh, H. (1988). Reasoning about operationality for explanation-based learning. Proceedings of the Fifth International Conference on Machine Learning (pp. 214–220). Ann Arbor, MI: Morgan Kaufmann.

Keller, R. (1988). Operationality and generality in explanation-based learning: Separate dimensions or opposite end-points. AAAI Spring Symposium on Explanation-Based Learning. Stanford University.

Mitchell, T., Keller, R., & Kedar-Cabelli, S. (1986). Explanation-based learning: A unifying view. Machine Learning, 1, 47–80.

Muggleton, S., Bain, M., Hayes-Michie, J., & Michie, D. (1989). An experimental comparison of human and machine learning formalisms. Proceedings of the Sixth International Workshop on Machine Learning (pp. 113–118). Ithaca, NY: Morgan Kaufmann.

Ourston, D., & Mooney, R. (1990). Changing the rules: A comprehensive approach to theory refinement. Proceedings of the Eighth National Conference on Artificial Intelligence (pp. 815–820). Boston, MA: Morgan Kaufmann.

Pagallo, G., & Haussler, D. (1990). Boolean feature discovery in empirical learning. Machine Learning, 5, 71–100.

Pazzani, M., & 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.

Pazzani, M., Brunk, C., & Silverstein, G. (1991). A knowledge-intensive approach to learning relational concepts. Proceedings of the Eighth International Workshop on Machine Learning (pp. 432–436). Evanston, IL: Morgan Kaufmann.

Pazzani, M., & Kibler, D. (1992). The role of prior knowledge in inductive learning. Machine Learning.

Pazzani. M., (1992). When Prior Knowledge Hinders Learning. AAAI workshop on constraining learning with prior knowledge. San Jose.

Quinlan, J.R., (1990). Learning logical definitions from relations. Machine Learning, 5, 239–266.

Rabinowitz, H., Flamholz, J., Wolin, E.,& Euchner, J. (1991). Nynex Max: A telephone trouble screening expert system. In R. Smith & C. Scott (Eds.) Innovative applications of artificial intelligence, 3, 213–230.

Richards, B. (1992). An Operator-Based Approach to First-Order Theory Revision. Ph.D. Thesis. University of Texas, Austin.

Richards, B. & Mooney, R. (1991). First-Order Theory Revision. Proceedings of the Eight International Workshop on Machine Learning (pp. 447–451). Evanston, IL: Morgan Kaufmann.

Segre, A. (1987). On the operationality/generality trade-off in explanation-based learning. Proceedings of the Tenth International Joint Conference on Artificial Intelligence (pp. 242–248). Milan, Italy: Morgan Kaufmann.