Optimizing the Predictive Value of Diagnostic Decision Rules

Sholom M. Weiss*, Robert S. Galen**, and Prasad V. Tadepalli*

*Department of Computer Science, Rutgers University, New Brunswick, NJ 08904
**Department of Biochemistry, Cleveland Clinic Foundation, Cleveland, Ohio

Abstract

An approach to finding an optimal solution for an important diagnostic problem is described. Examples are taken from laboratory medicine, where the problem can be stated as finding the best combination of tests for making a diagnosis. These tests are typically numerical with unknown decision thresholds. Because of uncertainty in classification, the solution is described in terms of maximizing measures of decision rule performance on a data base of cases, for example maximizing positive predictive value, subject to a constraint of a minimum sensitivity. The resultant rules are quite similar to classification production rules, and the procedures described should be valid for many knowledge acquisition and refinement tasks. The solution is found by a heuristic search procedure, and empirical results for several data bases and published studies are described.[1]

I. Introduction

Rule based systems have found increasing success in capturing experiential knowledge. While relatively simple in structure, these systems have proved useful because they capture knowledge in forms that are familiar and easy to explain. Many rule-based expert systems solve problems that fall into the general category of classification [Clancey, 1985, Weiss and Kulikowski, 1984] problems. Diagnostic decision making is a typical example. This type of problem has many characterizations, and formal solutions have been developed under various assumptions through statistical hypothesis testing, discriminant analysis, and pattern recognition [Duda and Hart, 1973]. Rule based systems can incorporate formal decision models, if the statistical information is available to build them, together with the pragmatic knowledge of when to invoke them. But in general this is not the case, and an expert system is resorted to precisely because one needs to start with the human expert's best guess at what constitute good decision rules. The decision rules chosen by experts have to be easy to compute and explain. They therefore tend to involve relatively small *chunks* of information in their antecedent conditions, and tend to use easy-to-understand logical connectives (conjunction and disjunction), rather than the more difficult to interpret mathematical combining functions (such as linear combinations of finding values in linear discriminants). While many expert systems have been built over the past decade, there has been little progress in relating their performance to more traditional decision-making approaches. This is tied-in with the often cited weaknesses of the mathematics underlying many expert systems' inference schemes for handling uncertainty, and their related difficulties in automatic learning.

The AI literature has recently had numerous discussions of the various approaches for combining rule scores in a valid probabilistic manner, e.g. Dempster-Shafer [Gordon and Shortliffe, 1985] or Bayesian approaches. The optimizing approach that we describe in this section has a different emphasis. Rather than worry about combining different scores, we pose the simpler question of finding the left hand side of a rule (with certain simplifying structural constraints) that has the best potential for yielding correct classification. Because the rules are simple forms of standard production rules for classification problems; they can be analyzed quite exactly. While the knowledge engineering approach to building rule-based systems has been predominant in recent years, there continues to be a strong interest in search strategies that can potentially yield optimal solutions [Kumar and Kanal, 1983, Kumar, 1984, Pearl, 1984]. In limited contexts such solutions can be of use for certain aspects of knowledge acquisition, e.g. optimal decision trees [Martelli and Montanari, 1978].

In this paper, we show how a classical medical diagnostic problem, subject to some simplifying representational choices, can be solved in an optimal or near-optimal fashion. This should prove a particularly powerful tool for laboratory-based medicine, since it can help indicate what is the *best* set of tests to perform. Of general interest to the AI community is that the solution to this problem is an optimal decision rule that is posed as a logical set of clauses. While an optimal solution is stated in terms of statistical constraints, the identification of a solution to the problem is described as a heuristic search procedure.

II. Statement of the Problem

In this discussion, examples from laboratory medicine will be used. However, the solution is general and should be applicable to many areas outside medicine. Let us assume that we are developing a new diagnostic test whose measurement yields a numerical result in a continuous range. For a single test, the problem is to select a cutoff point, known formally as a *referent value*, that will lead to satisfactory decisions. For example, a physician may conclude that all patients having a result greater than a specific cutoff have the disease, while others do not. There are well-known measures to describe the performance of a test at a specific cutoff for a sample population. These measures are *sensitivity, specificity, positive predictive value, negative predictive value, and efficiency* [Galen and Gambino, 1975]. Thus, results at each cutoff can be described in terms of these measures. Using a specific cutoff, there are four possible outcomes for each test case in the

[1]This research was supported in part by the Biomedical Research Technology Program, National Institutes of Health, Grant P41 RR02230.

sample.[2] This is illustrated in Figure 1.

	Test Positive	Test Negative
Hypothesis Positive	True Positives	False Negatives
Hypothesis Negative	False Positives	True Negatives

Figure 1: Possible Outcomes of Tests for Hypotheses

Figure 2 formally describes these measures of performance.

	T+	T-
H+	TP	FN
H-	FP	TN

Sensitivity	TP / H+
Specificity	TN / H-
Predictive value (+)	TP / T+
Predictive value (-)	TN / T-
Efficiency	(TP+TN) / TOTAL

Figure 2: Formal Measures of Diagnostic Performance

While all of these measures have their purpose, the one that is typically most valuable for rule-based systems is the positive predictive value. Positive predictive value measures how often a decision is correct when a test result is positive. Thus one may use a positive test that has high predictive value in rules that confirm a diagnosis, and apply different tests when the result is negative. Many rule based systems may be thought of as collections of rules with very high positive predictive values.[3] We illustrate these points by describing data taken from a published study on the assessment of 8 laboratory tests to confirm the diagnosis of acute appendicitis for patients admitted to an emergency room with a tentative diagnosis of acute appendicitis [Marchand, Van Lente, and Galen, 1983]. In the example of Figure 3, the white blood cell count (WBC) is used as the test.

	T+	T-
H+	71	14
H-	6	15

Sensitivity	83.5%
Specificity	71.4%
Predictive value (+)	92.2%
Predictive value (-)	51.7%
Efficiency	81.1%

Figure 3: Example of the 5 Measures of Performance for WBC>10000

In summary, for a single test with a given cutoff and the application of an arithmetic operator,[4] these five measures can be determined for a population. The problem of determining an optimal cutoff can be described as maximizing one of these measures subject to specific constraints on the other measures.[5] Constraints are the minimum required values for sensitivity, specificity, predictive values, and efficiency.[6] Finding the optimum cutoff for WBC can be posed in the form illustrated in Figure 4.

MAXIMIZING Predictive value (+) of WBC

The constraints are given below:

Sensitivity	≥	100.00%
Specificity	≥	0.00%
Predictive value (-)	≥	0.00%
Efficiency	≥	0.00%

Figure 4: Example of Problem Constraints for a Single Test

We note that this problem can be seen as a special restriction on a statistical decision-making or pattern recognition problem. Here the cutoff threshold is not given and instead must be determined. For *known* population statistics, the threshold for each of these measures that corresponds to the optimal likelihood ratio choice might be determined. However, in diagnostic testing, it is rare that population statistics for large numbers of combinations of tests can be established. Our form of analysis then answers questions of optimal diagnostic performance on an empirical basis, for a particular sample of cases.

Referent value analysis, or cutoff selection, is commonly done for single tests. We have developed procedures that allow for the possibility of choosing the set of constraints and maximizing the remaining measure not only for one or two, but for a larger number of tests.[7] When more than one test is specified, combinations are formed by using logical AND or OR operators. We formulate the problem as finding the *optimal* combination of tests that will satisfy the given constraints for the data base. An additional constraint is added to the problem, in that the length of the expression is limited by a chosen threshold.[8] In Figure 5 using the appendicitis data base, the problem is to find an optimal solution in the form of a logical expression whose length is

[2]For purposes of this discussion, we are eliminating the possibility of unknowns.

[3]This minimizes the effect of an unknown prevalence.

[4]These operators are less than or greater than.

[5]Sensitivity and specificity move continuously in opposite directions. For example, a 100% sensitivity cutoff with 0% specificity can always be found by classifying every sample as having the hypothesis. Predictive values have no such relationship and vary all over the place.

[6]The interrelations among these performance parameters, limit the possible patterns of constraints for any given set of data.

[7]If two tests have the same value for the optimized measure, then its conjugate measure is used to decide which test is better. Sensitivity and specificity are treated as conjugates to one another and so are positive and negative predictive values. When optimizing efficiency, positive predictive value is chosen to be the next decisive function.

[8]This sets a limit on the number of tests that may be used in the decision rule. Some tests may be also deliberately excluded from consideration and some tests may be designated as mandatory. This allows for further pruning of the search space.

no greater than 3 tests.[9]

MAXIMIZING Predictive value (+)

The constraints are given below:

Sensitivity	≥	100.00%
Specificity	≥	0.00%
Predictive value (-)	≥	0.00%
Efficiency	≥	0.00%
Number of terms	≤	3

Figure 5: Example of Problem Constraints for 3 or Fewer Tests

At this point we note that the rules are just like many found in typical classification expert systems, since, like productions, they are described as logical combinations of findings that are not mutually exclusive.[10] Thus, they have the intuitive appeal in explaining decisions because of their modular nature, while being supported empirically by their performance over the data base. In contrast to traditional machine learning [Mitchell, 1982, Quinlan, 1986, Michalski, Mozetic, Hong, and Lavrac, 1986], the objective here is to find the *single best* conjunctive or disjunctive rule of a fixed size. Starting with undetermined cutoffs for continuous variables, these rules classify under conditions of uncertainty, where two types of classification errors, false positives and false negatives, need not be considered of equal importance.

III. Complexity

In Section II, we described the problem as finding the best logical expression of a fixed length or less that covers a sample population. In this section, we consider the complexity of exhaustively generating and testing all possibilities. Except for relatively small populations or numbers of tests,[11] the exhaustive approach is not computationally feasible.

Equation 1 is the number of expressions having only ANDs; Equation 2 is for expressions having either ANDs or ORs. In these equations, n is the number of tests, k is is the maximum number of tests in the expression, c is the number of constants (cutoff values) to be examined for each test, and c^i is c raised to the ith power. While the *number* of distinct values that must be examined for each test may vary, we have have used a fixed number, c, to simplify the notation and analysis. In Equation 2, expressions are generated in disjunctive normal form.[12]

$$\sum_{i=1}^{k} \binom{n}{i} c^i \qquad (1)$$

$$\sum_{i=1}^{k} \binom{n}{i} c^i B_i \qquad (2)$$

where B_i is the ith Bell number.[13]

The most computationally expensive (exponential) component of Equation 2 component is c^i. It is possible to devise exhaustive procedures that do not require the examination of every value of a test found in the data base. For each test, one may examine only those points that overlap in the H+ and H- populations. Moreover, only the smaller set of the two sets of points in the overlapping zone need be candidates for cutoffs.[14] Even taking this into account, relatively small values of c will make the computation prohibitive.

Because one may allow for the repetition of a test in an expression, the number of generated expressions may be substantially greater than Equation 2.[15] For the appendicitis data base having a sample of 106 cases, we computed an average of 65 expressions/second on a VAX/785.[16]

IV. A Heuristic Procedure for Optimizing Predictive Values

Because of the computational complexity of an exhaustive search, we have developed a heuristic search procedure for finding the optimal combination. In this section, we describe the procedure. While this procedure is not guaranteed to find an optimal solution, the expression found should almost always be near-optimal. In Section V, empirical evidence is provided to demonstrate that in numerous situations the optimal solution is found. In almost every real experimental situation,[17] the logical expression found by the computer should be better than what a human experimenter could compose.

Before specifying the heuristic procedure, a few general comments can be made. In an exhaustive search approach, it is possible to specify a procedure that needs no additional memory. Logical expressions are generated and they are compared with the current best. The heuristic procedure is based on an alternative strategy. A relatively small table of the most promising expressions is kept. Combinations of expressions are used to generate longer expressions. The most promising longer expressions in turn are stored in the table

[9] As noted in Section V, the optimal solution is a disjunction of 2 tests.

[10] An OR condition may encompass several conditions that are not mutually exclusive. The classification may have less than 100% diagnostic accuracy.

[11] These are tests with relatively few potential cutoffs.

[12] This normal form corresponds to that used by the heuristic procedure described in Section IV.

[13] The Bell number is the number of ways a set of i elements can be split into a set of disjoint subsets. For i=0,1,2,3, B_i=1,1,2,5 respectively [Andrews, 1976]. The Bell number is defined recursively as

$$B_{i+1} = \sum_{k=0}^{i} \binom{i}{k} B_k.$$

[14] Each test would have a a distinct number of cutoffs that must be examined, c_t. In the equations, instead of c^i, the products of c_t for each generated expression must be summed.

[15] For example, a>50 OR (a >30 AND b <20).

[16] This is the average for length less than 4. Another data base mentioned in Section V has approximately 3000 cases, which increases the computations correspondingly.

[17] These are situations where the experimenter is analyzing new data and does not know a priori the best rule.

and are used to generate even longer expressions. Thus memory is needed to store the most promising or useful expressions. In Equation 2, the exponential component is the c^i. Thus, if one can reduce the number of points in c, i.e. the number of cutoffs for a test, the possible combinations are greatly reduced. Figure 6 illustrates the key steps of the heuristic procedure. In Section IV.A, the approach taken to greatly reduce the number of cutoffs is discussed.

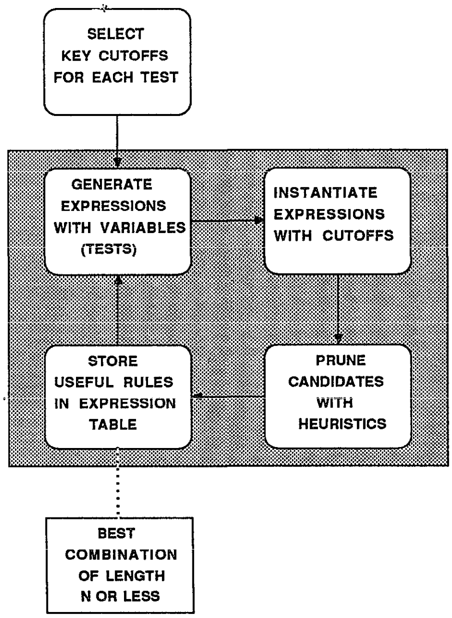

Figure 6: Overview of Heuristic Procedure for Best Test Combination

A. Selection of Cutoffs

For each test in the data base, the mean is found for the cases satisfying the hypothesis (H+) and the cases not satisfying the hypothesis (H-). If the H+ has the greater mean, the ">" operator is used. If H+ has the smaller mean, the "<" operator is used.[18]

The next task is to select the test cutoffs. For a test, cutoffs that fall at *interesting boundaries* are selected. Interesting boundaries are those where the predictive values (positive or negative) are locally maximum. For example, if WBC>10000 has a positive predictive value of 97% and WBC>9900 and WBC>10100 each has a positive predictive value less than 97%, then 10000 is an interesting boundary for WBC. The procedure first determines the interesting boundaries on a coarse scale. Then it zooms in on these boundaries and collects all the interesting boundaries on a finer scale.[19] Finally, the boundaries are smoothened without changing the predictive statistics of the rule. Test cutoffs that have very low sensitivity or specificity are immediately pruned.[20]

B. Expression Generation

Logical expressions of all test variables in all combinations are generated in disjunctive normal form.[21] This method avoids duplication of equivalent expressions since AND and also OR are symmetric. These expressions are stored in an expression table and longer expressions are generated combining shorter expressions. As each new expression is generated, the test variables are instantiated in all combinations of cutoff values. The test cutoffs were selected prior to expression generation. Figure 7 is a simple illustration of this process for 3 tests, {a, b, c} and expressions of length 2 or less.

```
a
b
c
a AND b
a AND c
b AND c
a OR b
a OR c
b OR c
```

Figure 7: Example of Expressions with Variables (tests)

If b has interesting cutoffs at b>10, b>20 and c has interesting cutoffs at c<30, c<40, c<50, then the expression b AND c would lead to the possibilities of Figure 8.

```
b>10 AND c<30
b>10 AND c<40
b>10 AND c<50
b>20 AND c<30
b>20 AND c<40
b>20 AND c<50
```

Figure 8: Example of Instantiated Expression

Because new longer expressions are generated from shorter expressions that have been stored in a table, those expressions that have been pruned will no longer appear in any longer expression. During the course of instantiation of the

[18]The equality operator "=" may also be used for discrete tests corresponding to simple encodings such as multiple choice questions. A discrete test is considered to be a test whose values are always between 0 and 10.

[19]A local maximum corresponds approximately to the following conditions for the cutoff and its two neighbors: One neighbor of the cutoff has the same number of correct classifications but more errors. The other neighbor has fewer correct classifications but the same number of errors.

[20]In the current version of the program, 10 equally spaced intervals are used for the region where the two populations overlap. For zooming in on an interval, 20 finer intervals are used between its 2 neighbors on the coarse scale. The minimum acceptable sensitivity or specificity for a test is currently set to be 10%.

[21]For example, a AND (b OR c) must be written as (a AND b) OR (a AND c).

variables, some heuristics can be applied to prune the possibilities. These are discussed in Section IV.C.

C. Heuristics for Pruning Expressions

Although the heuristic cutoff analysis limits the search space to the most *interesting* cutoffs, the search space may still remain relatively large. Several heuristics and some provably correct pruning rules are employed by the procedure. The first 3 pruning rules are always correct, the others are heuristics that attempt to consider the most promising candidates for combination into new longer rules.

1. If the sensitivity and specificity values of an expression are both less than the constraints then that expression does not contribute to any useful rules.

2. If an expression has less specificity than required, then any expression formed by ORing that expression with another will also have less specificity than required.

3. If an expression cannot be extended to one that contains all the mandatory tests, while satisfying the length constraint, it is immediately pruned.

4. If an expression has better positive and negative predictive values than another expression that differs from the first only by the constants in the expression, then the expression with lower predictive values is ignored.

5. If there are rules shorter and better than a new candidate rule, compute the sum of their lengths. If this sum, including the length of the current rule, exceeds the maximum length possible for any rule, then ignore the new rule.[22]

After all interesting expressions have been generated, the best expression in the expression table is offered as the answer.[23] Because all promising expressions are stored, a program that implements this procedure can readily determine its next best expression. If the constraints are made stricter, the expression table remains valid and the procedure's new best expression should be immediately available.

[22]In the current implementation, the maximum rule length is fixed as 6. As the expression length increases, the number of potential combinations greatly increases. The objective of this heuristic is to emphasize the most promising shorter rules that will be combined into lengthier rules.

[23]During expression generation, whenever a superior expression is found, it is displayed. If no expression is found meeting the constraints, this is indicated when the search terminates. Depending on the allocated table space for storing intermediate expressions, the program may terminate from an overflow of the table. This is unlikely to occur with relatively small expressions.

V. Results and Discussion

The heuristic procedure has been implemented in a computer program. Because of the underlying empirical nature of the problem, by examining hundreds of possibilities, the program should be able to find better logical expressions than the human experts when the samples are representative. This is particularly true when the human experimenter is examining new tests or performing an original experiment. Because of the heuristic nature of the search, one wonders about the optimality of the solutions.

Several experiments were performed to test the capability of the program to find optimal or near optimal solutions. Several years after the appendicitis data used in our examples were reported in the medical literature, we re-analyzed the data. The samples consisted of 106 patients and 8 diagnostic tests. Because only 21 patients were normal, it is possible to construct an exhaustive procedure.[24] In original study, the experimenters were interested in maximizing positive predictive value, subject to the constraint of 100% sensitivity. They cited a logical expression consisting of the disjunction of 3 diagnostic tests with positive predictive value of 89%. Using the heuristic procedure, the following results can be reported:

- A superior logical expression composed of only 2 tests can be cited. This test has positive predictive value of 91%. The analysis takes 3 minutes of cpu time on a VAX 785.

- Using exhaustive search, the optimal expression of length 3 or less is identical to the one found by the heuristic procedure. The exhaustive search took 10 hours of cpu time on a VAX 785.[25]

Using a large data base of approximately 3000 cases, additional experiments were performed. These cases belong to a knowledge-based system that is being developed for a laboratory medicine application involving routine health screening.[26] The data base consists almost exclusively of diagnostic laboratory tests. In several instances, there are relatively short, length 3 or less, rules in the knowledge base that reach specific conclusions. When a rule is the sole rule for a conclusion, we have a rule that has 100% sensitivity and 100% positive predictive value. Because this is an expert's rule, we know that the rule has a strong scientific and experiential support. For experimental purposes we limit our task to finding the expert's rule by analyzing the case data base. We have a situation where the optimal solution is known to us before an empirical analysis. For the five rules that we selected that had 100% sensitivity and 100% predictive value, we were able to match the expert's rule.[27]

The results of these experiments are encouraging. While

[24]In this case, c=21 for Equation 2.

[25]The result reported in the literature was WBC>10500 OR MBAP>11% OR CRP>1.2. The optimal solution is WBC>8700 OR CRP>1.8.

[26]Unlike the appendicitis population, in this population the overwhelming number of samples are normal patients.

[27]In some instances, a shorter rule was found that was a subset of the expert's rule. This is due to a relatively small number of cases in the H+ population.

the optimal solution to the problem is clearly the goal, near-optimal solutions in reasonable times are also extremely valuable. In many practical situations, humans cannot solve this problem. For example, while combination tests are often cited in the diagnostic medical literature, in almost all instances the logical expression is found on the basis of previous experience, intuition, and trial and error analysis. We believe that the approach cited in this paper offers an opportunity to analyze data and present results in an optimal or near-optimal fashion.

The examples cited here were from realistic and important diagnostic medical applications. In future years, we can expect that laboratory medicine and diagnostic tests will assume an ever more important role in diagnostic decision-making. While this form of diagnostic performance analysis, i.e. the five measures of performance, is the standard in the medical diagnostic literature, there is nothing that is specific to medical data.

Because medical tests have a clear physiological basis, the expectation of continued performance on new populations is great. We have presented our work as the optimal fitting of a logical expression to *existing* data. Thus we have not addressed the question of experimental design or validation of results for a specific application. Unless one derives very highly predictive rules, this form of data analysis is subject to inaccuracies based on unrepresentative samples or prevalences.[28] As is done in pattern recognition applications, estimates of future performance can be made by train and test experiments or jackknifing [Efron, 1982].

In terms of knowledge base acquisition, this approach can prove valuable in both acquiring new knowledge, refining existing knowledge [Wilkins and Buchanan, 1986, Ginsberg, Weiss, and Politakis, 1985], and verifying correctness of old knowledge. Because a knowledge base of rules summarizes much more experiential knowledge than is usually covered by a data base of cases, in many instances this approach can be thought of as supplementary to the knowledge engineering approach to knowledge acquisition in rule-based systems.

Acknowledgments

We thank Casimir Kulikowski for his critique of an early draft of this paper, and his clarifications of the relationship of our work to statistical pattern recognition. We acknowledge the programming support of Kevin Kern, who programmed and tested many of the procedures described in this paper.

[28] This point is also valid for any knowledge-based system reasoning with uncertainty.

References

Andrews, G. (1976). *Encyclopedia of Mathematics and its Applications II - The Theory of Partitions.* Reading, Mass.: Addison-Wesley.

Clancey, W. (1985). Heuristic Classification. *Journal of Artificial Intelligence, 27*, 289-350.

Duda, R., and Hart, P. (1973). *Pattern Classification and Scene Analysis.* New York: Wiley.

Efron, B. (1982). The Jackknife, the Bootstrap and Other Resampling Plans. *SIAM*. Philadelphia, Pa.

Galen, R. and Gambino, S. (1975). *Beyond Normality.* New York: John Wiley and Sons.

Ginsberg A., Weiss, S. and Politakis, P. (1985). SEEK2: A Generalized Approach to Automatic Knowledge Base Refinement. *Proceedings IJCAI-85.* pp. 367-374. Los Angeles, Ca.

Gordon, J. and Shortliffe, E. (1985). A Method for Managing Evidential Reasoning in a Hierarchical Hypothesis Space. *Journal of Artificial Intelligence*, 26, 323-327.

Kumar, V. (1984). A General Bottom-up Procedure for Searching And/Or Graphs. *Proceedings AAAI-84.* pp. 182-187. Austin, Texas.

Kumar, V. and Kanal, L. (1983). The Composite Decision Process: A Unifying Formulation for Heuristic Search, Dynamic Programming and Branch & Bound Procedures. *Proceedings AAAI-83.* pp. 220-224. Washington D.C.

Marchand, A., Van Lente, F., and Galen, R. (1983). The Assessment of Laboratory Tests in the Diagnosis of Acute Appendicitis. *American Journal of Clinical Pathology, 80(3)*, 369-374.

Martelli, A. and Montanari, U. (1978). Optimizing Decision Trees Through Heuristically Guided Search. *Comm. of the ACM, 21*, 1025-1039.

Michalski, R., Mozetic, I., Hong, J., and Lavrac, N. (1986). The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains. *Proceedings AAAI-86.* pp. 1041-1045. Philadelphia, Pa.

Mitchell, T. (1982). Generalization as Search. *Artificial Intelligence, 18*, 203-226.

Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving.* Addison-Wesley.

Quinlan, J. (1986). The Effect of Noise on Concept Learning. In Michalski, R., Carbonell, J., and Mitchell, T. (Eds.), *Machine Learning.* Morgan Kaufmann.

Weiss, S. and Kulikowski, C. (1984). *A Practical Guide to Designing Expert Systems.* Totowa, New Jersey: Rowman and Allanheld.

Wilkins, D. and Buchanan, B. (1986). On Debugging Rule Sets When Reasoning Under Uncertainty. *Proceedings AAAI-86.* pp. 448-454. Philadelphia, Pa.