RUTGERS UNIVERSITY
Center for Expert Systems Research

Technical Report:
Maximizing the Predictive Value of Production Rules

Contract Number
N00014-87-K-0398
Office of Naval Research

August 31, 1988

Principal Investigators:
Sholom M. Weiss
Casimir A. Kulikowski

Table of Contents

Maximizing the Predictive Value of Production Rules[1]

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

A new approach to finding a solution for an important empirical learning problem is described. The problem is to find the single best production rule of a fixed length for classification. Predictive Value Maximization (PVM), a heuristic search procedure through the space of conjunctions and disjunctions of variables and their cutoff values, is outlined. Examples are taken from laboratory medicine, where the goal is t find the best combination of tests for making a diagnosis. Resampling techniques for estimating error rates are integrated into the PVM procedure for rule induction. Excellent results for PVM are reported on data sets previously analyzed in the AI literature using alternative classification techniques.

1. Introduction

Many decision-making problems fall into the general category of classification [Clancey, 1985, Weiss and Kulikowski, 1984, James, 1985]. Diagnostic decision making is a typical example. Empirical learning techniques for classification span roughly two categories: statistical pattern recognition [Duda and Hart, 1973, Fukunaga, 1972] (including neural nets [McClelland and Rumelhart, 1988]) and machine learning techniques for induction of decision trees or production rules. While a method from either category is usually applicable to the same problem, the two categories of procedures can differ radically in their underlying models and the final format of their solution. Both approaches can be used to classify a sample pattern into a specific class. However, a rule-based or decision tree approach offers a modularized, clearly explained format for a decision, and is compatible with a human's reasoning procedures and expert system knowledge bases.

Methods of induction of decision trees from empirical data have been studied by researchers in both artificial intelligence and statistics. Quinlan's ID3 [Quinlan, 1986] and C4 [Quinlan, 1987a] procedures for induction of decision trees are well known in the machine learning community. The Classification and Regression Trees (CART) [Breiman, Friedman, Olshen, and

[1] **This research was supported in part by ONR Contract N00014-87-K-0398 and NIH Grant P41-RR02230.**

Stone, 1984] procedure is a major nonparametric classification technique that was developed by statisticians during the same period as ID3. These procedures developed for decision tree induction are quite similar. The major distinction between CART and ID3/C4 is that the CART procedure uses resampling techniques for both accurate error estimation and tree pruning [Stone, 1974]. Empirical comparisons of CART-derived decision trees with traditional statistical discriminant analysis has show that the decision trees are very competitive in finding a minimum error solution. In almost all instances studied, the induced decision trees were equal or better than traditional statistical methods [Breiman, Friedman, Olshen, and Stone, 1984].

Production rules are related to decision trees; each path in a decision tree can be considered a distinct production rule. Unlike decision trees, a disjunctive set of production rules need not be mutually exclusive. The principal techniques of induction of production rules from empirical data are Michalski's AQ15 system [Michalski, Mozetic, Hong, and Lavrac, 1986] and recent work by Quinlan in deriving production rules from a collection of decision trees [Quinlan, 1987b].

Machine learning techniques for induction of decision rules have evolved from procedures that cover all cases in a data base to more accurate procedures for estimating error by train and test sampling. Procedures that prune a set of decision rules and the components of these rules have been successful in increasing the performance of an induced rule set on new unseen test cases [Michalski, Mozetic, Hong, and Lavrac, 1986, Quinlan, 1987a]. Empirical results reported in the literature indicate that often a relatively short rule may provide a better solution than a more complex set of induced rules [Michalski, Mozetic, Hong, and Lavrac, 1986].

In this paper, we describe Predictive Value Maximization (PVM), a heuristic procedure for learning the *single* best decision rule of a fixed length. In contrast to the decision tree induction techniques, a commitment is not made to split a single test node at a time. Instead, this method is a heuristic approximation to exhaustive generation of all possible rules of a fixed length. While a exhaustive search is not feasible in most applications, a small number of heuristics reduce the search space to manageable proportions.

In Section 2, a detailed description of the underlying model is given. The complexity of exhaustive search is presented in Section 3. The PVM procedure is described in Section 4. In section 5, the concept of resampling and honest error estimation is introduced. In a fashion similar to CART procedure for deciding appropriately sized trees, the PVM procedure is modified to use resampling for finding the appropriate length rule. In Section 6.1 two data sets are analyzed, and the results of PVM are compared with the optimal production rule solution and with several statistical pattern recognition solutions. A comparison of results for two other data sets reported in the machine learning literature is given in Section 6.2.

2. The Model of Induction

In our 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 accuracy* [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 sample.[2] This is illustrated in Figure 2-1.

	Rule Positive (R+)	**Rule Negative** (R-)
Hypothesis Positive (H+)	True Positives (TP)	False Negatives (FN)
Hypothesis Negative (H-)	False Positives (FP)	True Negatives (TN)

Sensitivity	TP / H+
Specificity	TN / H-
Predictive value (+)	TP / R+
Predictive value (-)	TN / R-
Accuracy	(TP+TN) / ((H+) + (H-))

Figure 2-1: Formal Measures of Classification Performance

While all of these measures have their purpose, the one that is implicitly used in large-scale rule-based systems is 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 highly positive predictive values. The two types of errors, false positives and false negatives need not be weighted equally. For example, in medical applications it is often required that the sensitivity be high, i.e. few false negatives with perhaps more false positives.

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

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

Galen, 1983]. Following surgery, only 85 of 106 patients were confirmed by biopsy to have had appendicitis. Thus, the ability to discriminate the true appendicitis patients by labs tests prior to surgery would prove extremely valuable. In the example of Figure 2-2, the white blood cell count (WBC) is used as a test to determine the true appendicitis patients.

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

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

Figure 2-2: 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,[3] 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.[4] Constraints are the minimum required values for sensitivity, specificity, predictive values, and accuracy.[5] Finding the optimum cutoff for WBC can be posed in the form illustrated in Figure 2-3.

MAXIMIZING Predictive value (+) of WBC

The constraints are given below:

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

Figure 2-3: Example of Problem Constraints for a Single Test

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

[4] 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 greatly.

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

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.[6] When more than one test is specified, combinations are formed by using logical AND or OR operators. We formulate the problem as finding the *best* 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.[7] In Figure 2-4 using the appendicitis data base, the problem is to find the best solution in the form of a logical expression whose length is no greater than 3 tests.[8]

MAXIMIZING Predictive value (+)

The constraints are given below:

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

Figure 2-4: 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.[9] Thus, they have the intuitive appeal of explaining decisions in a format consistent with human reasoning, while being supported empirically by their performance over the data base. 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.

[6] 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 maximizing accuracy, either sensitivity or specificity can be chosen as the next decisive function.

[7] 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.

[8] As noted in Section 6.1.1, the optimal solution is a disjunction of 2 tests.

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

3. Complexity of Exhaustive Generation of Expressions

In Section 2, 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,[10] 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.[11] 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 *i*th 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 *i*th Bell number.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

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

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.[13] Even taking this into account, relatively small values of *c* will make the computation prohibitive.

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

[11] It is assumed that the less than or greater than operators are selected simply on the basis of the means for each class.

[12] This normal form corresponds to that used by the heuristic procedure described in Section 4.

[13] 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.

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.[14] For the appendicitis data base having a sample of 106 cases, we computed an average of 65 expressions/second on a VAX/785.[15]

Very effective branch and bound methods are known for finding the optimal feature subset of a given size [Narenda and Fukunaga, 1977, Roberts, 1984, Foroutan and Sklansky, 1985]. These procedures start with all the features, eliminate one test at a time, and evaluate classification performance for a test subset. If a test subset has poorer performance than the current best, for many types of classifiers (or distance measures) there is no need to consider further subsets of this test subset. These procedures work well for some statistical classifiers because it is possible to readily compute a unique classifier or distance measure using n features. The complexity of production rule enumeration is largely due to the number of cutoffs that must be evaluated. Even before feature subset evaluation, it is not feasible to evaluate rules with all n features over all possible cutoff instantiations.

4. A Heuristic Procedure for Maximizing Predictive Values

Because of the computational complexity of an exhaustive search, we have developed a heuristic search procedure for finding the best 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 quite good. In Section 6, empirical evidence is provided to demonstrate that in several situations the optimal production rule is found. In almost every real experimental situation, the logical expression found by the computer should be better than what a human experimenter could compose. These are situations where the experimenter is analyzing new data and does not know a priori the best rule.

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 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.

The Predictive Value Maximization (PVM) procedure was originally developed for finding the

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

[15] This is the average for length less than 4.

best logical combination of laboratory tests for making a diagnosis. In this section, we give a brief overview of the procedure.

The goal is to find the single best rule of length less than or equal to *n*. A rule for a hypothesis or class consists of variables, constants, arithmetic operators and logical operators. The arithmetic operators are *less than, greater than,* or *equals*. The logical operators are AND or OR. For example, *X>30 OR Y<100* is a valid rule format. In terms of overall accuracy of classification, the best rule is the one that has the fewest number of errors in classification where the number of variables in the expression is no more than the stated length. The method is an approximation to exhaustive generation of all possible rules of a fixed length or less.

For each variable, *interesting* constants are determined. These cutoff points are local maximums of the predictive values. Logical expressions with variables are generated (in disjunctive normal form) and instantiated with constants. A relatively small table of the most promising expressions is kept. Combinations of the stored expressions are used to generate longer expressions. The most promising longer expressions in turn are stored in the table and are used to generate even longer expressions.

Figure 4-1 illustrates the key steps of the heuristic procedure. In Section 4.1, the approach taken to greatly reduce the number of (interesting) cutoffs is discussed.

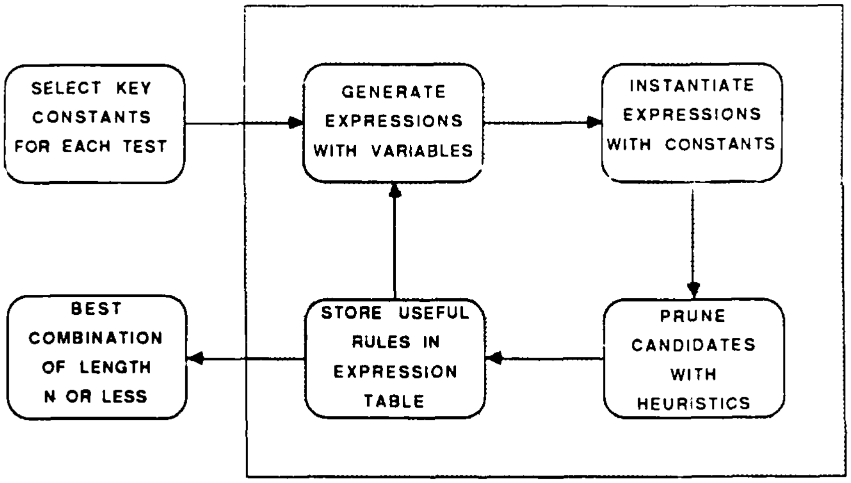
SELECT KEY CONSTANTS FOR EACH TEST
GENERATE EXPRESSIONS WITH VARIABLES
INSTANTIATE EXPRESSIONS WITH CONSTANTS
PRUNE CANDIDATES WITH HEURISTICS
STORE USEFUL RULES IN EXPRESSION TABLE
BEST COMBINATION OF LENGTH N OR LESS

Figure 4-1: Overview of Heuristic Procedure for Best Test Combination

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

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.[16] 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.[17]

4.2. Expression Generation

Logical expressions of all test variables in all combinations are generated in disjunctive normal form.[18] 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 4-2 is a simple illustration of this process for 3 tests, {a, b, c} and expressions of length 2 or less.

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 4-3.

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 variables, some heuristics can be applied to prune the possibilities. These are discussed in Section 4.3.

[16] 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.

[17] 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%.

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

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

Figure 4-2: Example of Expressions with Variables (tests)

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 4-3: Example of Instantiated Expression

4.3. 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.[19]

After all interesting expressions have been generated, the best expression in the expression table is offere.d as the answer.[20] 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 is immediately available.

4.4. Variations on the Standard PVM Application

The standard application of PVM was described in the previous section. The basic model is for two-class discrimination. Modifications can be made to the procedure to handle multiclass problems and step-wise refinement.

For multi-class problems, PVM is applied to a each class vs. not that class, and the single best rule for each class is found. Thus the n-class problem is solved as n 2-class problems. For n classes, n rules are found, and the best n-1 are used. The remaining class is selected when no rule is satisfied. In some difficult instances, it may be necessary to generate the n best rules, select the best one, remove the cases satisfying the rule, and then recursively re-apply the procedure. For the examples given in Section 6, it was not necessary to recursively apply PVM.

In some situations, another form of step-wise refinement is valuable. Because PVM may initially screen out some variables, the standard application of PVM may not work well for large numbers of variables or a low prevalence situation (i.e. many more cases of one hypothesis than another). PVM currently works with 18 variables at a time, and it uses only tests that have a minimum of 10% sensitivity or specificity. It does not apply both arithmetic operators (greater than and less than) simultaneously to the same variable. With step-wise refinement some o. these restrictions can be overcome. Assuming a 2-class model, two strategies are worthwhile mentioning:

1. Find a highly predictive rule for a class; remove the cases satisfying the rule, and re-apply the procedure to the remaining cases that did not satisfy the rule.

[19] 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.

[20] 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.

2. Find a highly sensitive rule for a class, i.e. a rule that covers most or all of the cases in the class; remove the cases not satisfying the rule, and re-apply the procedure to the remaining cases that did satisfy the rule.

Situation (1) is equivalent to an OR condition, i.e. finding multiple rules to covering a class. An example of this variation is given in Section 6.1.2. Situation (2) is equivalent to an AND condition, i.e. extending a rule in step-wise fashion to create a longer rule. An example of this variation is given in Section 6.2.2.

5. Estimating Error Rates

5.1. Basic Principles of Error Estimation

A procedure has been described that finds a single rule that best covers the cases. It is well known that the *apparent* error rate[21] of a classifier learned from all cases can lead to highly misleading estimates of performance [Duda and Hart, 1973]. This is due to overspecialization of the classifier to the data.[22]

Techniques for estimating error rates have been widely studied in the statistics [Efron, 1982] and pattern recognition [Duda and Hart, 1973, Fukunaga, 1972] literature. The simplest technique for "honestly" estimating error rates, the holdout or H method, is a single train and test experiment. The sample cases are broken into two groups of cases: a training group and a test group. The classifier is independently derived from the training cases, and the error estimate is the performance of the classifier on the test cases. A single random partition of train and test cases can be somewhat misleading. The estimated size of the test sample needed for a 95% confidence interval is described in [Highleyman, 1962]. The following interpretation of these results is offered in [Duda and Hart, 1973]: "If no errors are made on 50 test samples, with a probability 0.95 the true error rate is between zero and eight percent. The classifier would have to make no errors on more than 250 test samples to be reasonably sure that the true error rate is below two percent."

Instead of relying on a single train and test experiment, multiple random test and train experiments can be performed. For each random train and test partition, a new classifier is derived. The estimated error rate is the average of the error rates for classifiers derived for the *independently* and randomly generated partitions. Random resampling can produce better error estimates than a single train and test partition.

[21] sometimes referred to as the *resubstitution error rate*

[22] In the extreme, a classifier can be constructed that simply consists of all patterns in the given sample. Assuming identical patterns do not belong to different classes, this yields perfect classification on the sample cases.

A special case of resampling is known as leaving-one-out [Fukunaga, 1972, Efron, 1982]. Leaving-One-Out is an elegant and straightforward technique for estimating classifier error rates. Because it is computationally expensive, it is often reserved for relatively small samples. For a given method and sample size n, a classifier is generated using n-1 cases and tested on the remaining case. This is repeated n times, each time designing a classifier by *leaving-one-out*. Each case is used as a test case and, each time nearly all the cases are used to design a classifier. The error rate is the number of errors on the single test cases divided by n.

Evidence for the superiority of the leaving-one-out approach is well-documented [Lachenbruch and Mickey, 1968, Efron, 1982]. While leaving-one-out is a preferred technique, with large samples it may be computationally expensive. However as the sample size grows, traditional train and test methods improve their accuracy in estimating error [Kanal and Chandrasekaran, 1971].

The leaving-one-out error technique is a special case of the general class of *cross validation* error estimation methods [Stone, 1974]. In k-fold cross validation, the cases are randomly divided into k mutually exclusive test partitions of approximately equal size. The cases not found in each test partition are independently used for training, and the resulting classifier is tested on the corresponding test partition. The average error rates over all k partitions is the cross-validated error rate. The CART procedure was extensively tested with varying numbers of partitions and 10-fold cross validation seemed to be adequate and accurate, particularly for large samples where leaving-one-out is computationally expensive [Breiman, Friedman, Olshen, and Stone, 1984][23] For small samples, bootstrapping, a method for resampling with replacement, has shown some promise as a low variance estimator for classifiers [Efron, 1983, Jain, Dubes, and Chen, 1987, Crawford, 1988]. This is an area of active research in applied statistics.

Figure 5-1 compares the techniques of error estimation for a sample of n cases. The estimated error rate is the average of the error rates over the number of iterations. While these error estimation techniques were known and published in the 1960s and early 1970s, the increase in computational speeds of computers, makes them much more viable today for larger samples and more complex classification techniques [Steen, 1988].

	Holdout	Random Resampling	Leaving-One-Out	10-fold CV
Training cases	j	j	n-1	10%
Testing cases	n-j	n-j	1	90%
Iterations	1	B<<n	n	10

Figure 5-1: Comparison of Techniques for Estimating Error Rates

[23] Empirical results also support the stratification of cases in the train and test sets to approximate the percentage (prevalence) of each class in the overall sample.

Besides improved error estimates, there are a number of significant advantages to resampling. The goal of separating a sample of cases into a training set and testing set is to help design a classifier with a minimum error rate. With a single train and test partition, too few cases in the training group can lead to the design of a poor classifier, while too few test cases can lead to erroneous error estimates. Leaving-One-Out, and to a lesser extent random resampling, allow for accurate estimates of error rates while training on most cases. For purposes of comparison of classifiers and methods, resampling provides an added advantage. Using the same data, researchers can readily duplicate analysis conditions and compare published error estimates with new results. Using only a single random train and test partition introduces the possibility of variability of partitions to explain the divergence from a published result.

While error rates on test cases should be used to estimate the overall error rate for competing classifiers and methods, the best classifier design uses *all* cases in the sample set [Kanal and Chandrasekaran, 1971]. Resampling techniques provide better estimates of the error rates than a single train and test partition of the sample set [Efron, 1982].

5.2. Resampling with PVM

Because PVM searches for a single rule of a fixed length, the procedure is particularly amenable to resampling techniques. Resampling is not limited to error estimation and can be used to estimate any population parameter [Efron, 1982]. PVM can be used in conjunction with resampling to estimate the expression length having the minimum expected error rate. The PVM induction procedure described in Section 4 does not directly indicate the specific rule length that yields the best performance. While increasing the length will never decrease performance on the training cases, performance on test cases may decrease. Thus after a certain length, estimated error rates may increase, due to overspecialization of the rule. Leaving-One-Out and random resampling techniques can be used to provide estimates of the error rates for a specific expression length. In addition, these techniques can help perform a sensitivity analysis on competing expressions. Two estimating techniques are described: leaving-one-out and random resampling.

5.2.1. Leaving-One-Out

PVM uses leaving-one-out in the following manner:

- For each expression length i, let J_i be the estimated error rate by leaving-one-out. Choose length k, such that J_k is minimum, i.e. choose the length that has the minimum expected error rate. Choose the best expression of length k for all n cases in the sample set.

- Alternatively, let k be the length of the rule with the minimum error rate. The leave-one-out procedure will generate n classifiers, where n is the sample size (total number of cases). Choose the rule that repeats the most times, i.e. the modal rule. This corresponds to a form of sensitivity analysis. Since only a single case is left out in each cycle, the pattern that is most stable and consistent with the estimated error rate is selected.

5.2.2. Random Resampling

When the data set is large, or the length of the expression is relatively long, leaving-one-out may be computationally too expensive. PVM uses random resampling or 10-fold cross validation in the following manner:

- For each expression length i, let RS_i be the estimated error rate by random resampling. Choose length k, such that RS_k is minimum, i.e. choose the length that has the minimum expected error rate. Select the best expression of length k for all n cases in the sample set.

- Alternatively, let k be the length of the rule with the minimum error rate. For each of the B test samples, generate the best rule of length k or less. If a rule frequently repeats, i.e. the mode is relatively large, choose the modal rule. If a pattern of variables and operators frequently repeats, but the constants vary (e.g. X> ? & Y< ?), apply the induction method to *all* n cases. However, limit the process to the same variables and logical operators, adjusting only the constants.

6. Empirical Results

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.

In the previous sections, the PVM procedure for rule induction was described. In the following sections, we will explore a number of remaining issues related to the performance of this procedure. Several data sets for which published studies are available were analyzed. The analysis of these data sets should help address the following questions:

- How close is the PVM solution to the optimal solution for the underlying model of a production rule formed by conjunction or disjunction of variables with constant cutoffs?

- How competitive is the rule-based model to other models, such as traditional statistical models?

- How competitive is PVM with other machine learning procedures?

6.1. Optimality and Model Adequacy

6.1.1. Production Rule Optimality

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. In original study, the experimenters were interested in maximizing accuracy, subject to the constraint of 100% sensitivity. Failure to treat was much less desirable than treating too many patients. In their paper, 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. 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. Figure 6-1 compares the results for these two rules.[24]

	iginal Rule	New Rule
Number of tests	3	2
Sensitivity	1.000	1.000
Specificity	0.474	0.579
Predictive value (+)	0.895	0.914
Predictive value (-)	1.000	1.000
Accuracy	0.904	0.923

Figure 6-1: Comparison of Performance of Rules

Figure 6-2 compares the apparent error rate for this rule, Err_{App}, and the leaving-one-out error rate estimate, Err_{CV}.

[24] Both rules do not classify 2 cases because of missing data.

	Err_{App}	Err_{Cv}
Sensitivity	1.000	1.000
Specificity	0.579	0.526
Predictive value (+)	0.914	0.904
Predictive value (-)	1.000	1.000
Accuracy	0.923	0.913

Figure 6-2: Apparent and Leaving-One-Out Error Estimates for the Appendicitis Study

6.1.2. Comparative Analysis for Normally Distributed Data

The iris data was used by Fisher in his derivation of the linear discriminant function [Fisher, 1936], and it still is the standard discriminant analysis example used in most current statistical routines such as SAS or IMSL. Linear or quadratic discriminants under assumptions of normality perform extremely well on this data set. Three classes of iris are discriminated using 4 continuous features. The data sets consists of 150 cases, 50 for each class. Figure 6-3 summarizes the results for the ruled-based solution and several statistical methods. The optimal rules of size two were found by exhaustive search. These rules are quite simple and fully competitive with the other classifiers. *Petal length* < *3* perfectly separates Iris Setosa from the other classes and *Petal length* > *5 OR Petal Width* > *1.7* separates Iris Virginica from the other classes with 3 errors. The PVM procedure directly finds two rules for Iris Virginica that have one more error than the optimal solution. By re-applying PVM to cases that did not satisfy one of the initially derived rules, the resultant ORed rule is equivalent to the optimal rule.[25]

Method	Err_{App}	Err_{Cv}
Linear	.02	.02
Quadratic	.02	.027
Nearest Neighbor	.04	.04
Optimal Rule	.02	.02
PVM direct	.027	.04
PVM indirect	.02	.02

Figure 6-3: Comparative Performance on Fisher's Iris Data

We see that even in the classic normal case, the rule based approach does well, and PVM finds an excellent expression. The CART work showed that decision trees perform extremely well relatively to competitive statistical classifiers [Breiman, Friedman, Olshen, and Stone, 1984].

[25] The rule is in the form of F3>5.1 OR F4>1.8 OR F3>4.9 OR F4>1.6.

Because production rules are related to decision trees, we can expect that rule-based solutions should do well. In the next sections, we turn our attention to comparisons with alternative machine learning methods.

6.2. Comparison with Alternative Machine Learning Methods

6.2.1. Alternative Rule Induction Methods

A data set for evaluating the prognosis of breast cancer recurrence was analyzed by Michalski's AQ15 rule induction program and reported in [Michalski, Mozetic, Hong, and Lavrac, 1986]. There are 286 samples, 2 decision classes (recurrence of cancer or nonrecurrence) and 9 tests. They reported a 64% accuracy rate for expert physicians, and a 68% rate for AQ15, and a 72% rate for the pruned tree procedure of ASSISTANT [Kononenko, Bratko, and Roskar, 1986], a descendant of ID3.[26] The authors derived the accuracy rates by randomly resampling 4 times using a 70% train and a 30% test partition.

Because the authors randomly resampled, the experimental conditionscan be replicated. Figure 6-4 is a summary of performance results (on the test cases). For length 2, the same expression,

Involved Nodes>0 & Degree=3

was selected by PVM on each of four 70% training samples, with an average accuracy of 77% on the test samples.[27] For these data, it is feasible to attempt to derive more accurate error estimates than can be found by randomly resampling four times on a 70% train, 30% test partition of the data set. By leaving-one-out the complete data set for rule length 2 and 3, one can see that the accuracy peaks at length 2 (.773 vs. .769 for length 3), and the same expression repeats itself each of the 286 times. Thus the modal rule is the only expression that is generated.

Method	Variables	Rules	Error Rate
AQ15	7	2	32%
PVM	2	1	23%

Figure 6-4: Comparative Summary for AQ15 and PVM on Breast Cancer Data

[26] The prevalence of the larger class is 70%.

[27] Using the same size partition, 20 additional trials were performed. The resultant error estimate was 76% on the test cases, and this rule appeared 16 times.

6.2.2. Alternative Decision Tree Induction Methods

Quinlan briefly reported on results of his analysis of hypothyroid data in [Quinlan, 1987b], and in greater detail in [Quinlan, 1987a]. The data consists of 3772 thyroid cases, representing almost all thyroid tests done at the Garvan Institute during 1985. Four hypotheses are considered, 3 types of hypothyroid disease (7.6% of the samples) and nonhypothyroidism. One of the classes is represented by only one case. Over 10% of the lab tests were unavailable, but all cases were classified. In the original study, a single random train and test partition was used: 3143 cases for training and 629 cases for testing. In some instances, only 2514 cases were used for training.[28] Quinlan's C4 program produced decision trees, and he used pruning routines to produce a small set of production rules that performed better (than the original tree) on the test cases [Quinlan, 1987a]. In the published study we are given a set of two induced rules.[29]

The question we address is whether there are better rules that can be induced from the 3772 cases. A number of factors, which taken together, make a comparative analysis between the published results and PVM's results seem difficult. These include the use of a single random partition of test cases, the low prevalence of 7.6% for hypothyroidism, and the excellent very low error rates achieved by Quinlan's C4 program. However, a new analysis is quite feasible because 3428 new cases for the year 1986 are also available. Without training on them, the 3428 new cases can provide objective verification as to whether improved results have been achieved.

Figure 6-5 summarizes C4's published results and PVM's on all 3772 cases from the year 1985 and on the 3428 new cases from 1986. Only the cases from 1985 were used for rule induction. The cases from 1986 are used solely for verification of the results. Because of the large number of cases and high accuracy levels, the number of errors is cited instead of error rates.

Method	Variables	Rules	Errors (1985)	Errors (1986)
C4 pruned rules single holdout	8	2	31	43
PVM random resampling	8	2	17	30

Figure 6-5: Comparative Summary for C4 and PVM on Hypothyroid Data

PVM's performance was achieved using the random resampling procedure described in Section 5.2.2. The leaving-one-out procedure is computationally too expensive for this size data set. While standard procedure would involve using 3143 cases training cases, we used only 2514 training

[28] Quinlan performed experiments to examine whether it is advantageous to have a separate set of cases that are used during training to guide the induction procedure. A second set of 629 cases were drawn from the 3143 training cases for this purpose, leaving 2514 training cases.

[29] A third rule cited for nonhypothyroid is equivalent to the absence of either of the two rules for the specified diseases.

cases and 629 test cases for consistency with all of Quinlan's pruning experiments. Ten randomly drawn samples of train and test cases were drawn for each of the two diagnoses and the average number of errors (on the 629 test cases) for each length is given in Figure 6-6. Lengths beyond 6 were not considered. A length of zero represents the number of errors for no rule, i.e. the prevalence.

For the primary hypothyroid diagnosis the minimum error length is 2 and the modal rule is *TSH>6.1 & FTI <65*. This is also the rule that PVM induces for all 3772 cases. The characteristics of the random resampling analysis for the primary diagnosis are listed in Figure 6-7.

Class	length 0	1	2	3	4	5	6
Primary	16	8.0	2.9	3.0	3.4	3.3	3.4
Compensated	32	28.1	17.3	17.0	2.8	2.2	1.9

Figure 6-6: Estimated (average) Errors for Hypothyroid Diagnoses

For the compensated hypothyroid diagnosis, the minimum error length is 6, and the modal rule is

TSH>6 & TT4<149 & On Thyroxin=false & (FTI>64 or unknown) & TT4>50 & Surgery=false.

This is also the rule that PVM induces for all 3772 cases. A shorter rule also yields good results. If the rules are restricted to length 4 or less, the results are 24 errors for the year 1985 cases and 36 errors on the year 1986 cases.[30]

PVM was originally designed to find the best combinations of medical lab tests. A typical application of this type would have a few hundred cases and relatively few unknown test results. The PVM procedure eliminates from consideration tests not having at least 10% sensitivity or specificity, because these are not considered good tests for a diagnostic class. We also prefer not to classify cases when the induced rules cannot make a decision because of missing data.

As presented, the original hypothyroid data analysis is somewhat atypical of an expected PVM application. The sample sizes are quite large, and most classes have a low prevalence. While the PVM procedure was not modified for this application, PVM was applied in two stages. This was also necessary for computational reasons. Lengths beyond 3, were calculated in two parts: (a) the best rule of length 3 with 90% sensitivity, and the continuation, (b) the best rule up to an additional length 3 for cases satisfying rule (a). In a low prevalence environment, the two part application is helpful in the selection and filtering of useful tests and in the classification of unknowns. Tests that

[30] A more direct comparison with the with the original C4 experiments can be made when each trial is considered a single holdout trial, and the minimum error rule on 629 test cases is selected. None of the 10 PVM runs had more than 26 errors on the cases from 1985 or 39 errors on the cases from 1986, and the average was 21.5 errors for year 1985 and 33.8 for 1986.

Attribute	Value
Number of runs	10
Training sample size	2514
Test sample size	629
Minimum error length	2
Modal rule	TSH>6.1 & FTI<65
Rule mode j=2514	5
Modal variables	TSH>? & FTI<?
Variables only mode j=2514	10

Figure 6-7: Summary of Analysis of Primary Hypothyroid Diagnosis

have less than 10% sensitivity for all cases are not used in finding rule (a). These same unused tests may have greater than 10% sensitivity for cases satisfying rule (a) and may be used in finding rule (b) While some class prevalences may be low over all cases, the prevalence for classes satisfying rule (a) may be high. This may change the classification of cases that satisfy rule (a) but are unknown for the continuation, i.e. rule (b). PVM does not induce rules that explicitly state that a test must be unknown to reach a conclusion. However, for the rule induced for compensated hypothyroidism, FTI is the only test that has unknown values in the data set. The FTI component of the rule is induced in the second stage, when the odds have already shifted to compensated hypothyroidism.

The same 3772 cases from 1985 were used in a separate study of rule induction for hyperthyroidism [Quinlan, 1987c]: 2800 cases for training and 972 cases for testing. There are sufficient cases to attempt to diagnose 3 hyperthyroid conditions. Again we ask the question whether better rules can be induced from the 3772 cases than those cited in [Quinlan, 1987c]. Using a 2100 case training set and 700 case test set, the error rates (on the 700 test cases) for each length is summarized in Figure 6-8.

Class	length 0	1	2	3	4	5	6
Hyperthyroid	14	13.2	7.9	4.6	2.8	3.1	3.8
T3 toxic	2	13.7	3.2	2.6	2.2	2.2	2.2
Toxic goiter	2	7.4	2.2	1.2	1.4	1.3	1.3

Figure 6-8: Estimated (average) Errors for Hyperthyroid Diagnoses

Two sets of rules are cited in [Quinlan, 1987c]: the rules implied by a single decision tree and a

Method	Variables	Rules	Errors (1985)	Errors (1986)
C4 pruned tree single holdout	13	3	41	54
C4 pruned rules single holdout	38	7	28	48
PVM random resampling	7	2	31	46

Figure 6-9: Comparative Summary for C4 and PVM on Hyperthyroid Data

collection of decision trees. Their performance and that of PVM using 10 randomly drawn samples for each diagnosis is given in Figure 6-9.[31] Interestingly, there are insufficient data for inducing a rule for T3 toxic, because the expected error rate is greater than the prevalence. The rule found by PVM for hyperthyroidism is

FTI>155 & TT4>149 & On Thyroxin=false & (TSH<0.3 or unknown)

and the rule found for toxic goiter[32] is *Goiter & T3>2.8 & FTI<153.*

7. Discussion

The PVM procedure was originally developed for laboratory medicine applications [Weiss, Galen, and Tadepalli, 1987]. It was intended to help researchers find combinations of numerical tests that have greater predictive value than single tests. PVM assumes that a *single* short rule exists to classify a hypothesis. It does not expect perfect classification, and it can tradeoff false positive vs. false negative error rates.

Because relatively few tests are expected to be analyzed, an approximation to exhaustive enumeration was considered. For several hundred (varying) cases, exhaustive enumeration is not feasible, but experimental results support the contention that the PVM procedure will yield excellent, sometimes optimal results. In two studies where the optimal results for rules of a fixed length can be determined, PVM was able to find an optimal or near-optimal solution. Rule-based solutions appear to be quite competitive with alternative statistical procedures, with the advantage of simplicity and clarity of presentation. In its current implementation, PVM handles up to 18 tests at a time; filtering procedures and multi-stage analysis can be employed to reduce the number of tests to 18 at each stage.

In this paper, we re-analyzed data that had been analyzed using prominent machine learning

[31] If each trial is considered a single holdout trial, and the minimum error rule on 700 test cases is selected, then none of the 10 runs had more than 38 errors on the cases from 1985 or 53 errors on the cases from 1986, and the average was 33.4 errors for year 1985 and 46.1 for 1986. Only one of the ten trials had more than 46 errors on the cases from 1986.

[32] This is the result for length 3 with 90% sensitivity.

techniques. We showed that superior rules could be induced from these data sets. In the case of Michalski's cancer data, a simple two variable rule produces better results than the more complex rules cited in the literature. While Quinlan's original data analysis produced excellent results, we showed that somewhat better rules can be induced than those cited in the original studies.

As is used in the CART procedure, resampling techniques are employed by PVM to estimate error rates for induced production rules. These techniques can be time-consuming, but can lead to better induction results. Because PVM induces rules for a fixed, relatively short length, resampling procedures are a natural extension of the basic method. The major advantage is that error estimates can be derived, while essentially the complete data sample may be used for classifier design.

PVM is not always superior to other empirical rule or tree induction procedures. Unlike the alternative methods, PVM in practice is limited to the induction of single short rules. However, if a good solution exists in the form of a single short rule, PVM should have an advantage. Unlike incremental empirical induction procedures that select one test at a time, PVM examines combinations of tests with varying constants. There are many applications, such as when testing is expensive, where a short rule is highly desirable.

Researchers in machine learning have noted that relatively small pruned rules often yield better results than more complex sets of induced rules [Quinlan, 1987b, Michalski, Mozetic, Hong, and Lavrac, 1986]. The number and size of rules that can be effectively inferred from even large data sets is often surprisingly small. The number of rules in many rule-based expert systems far exceeds those found in these machine learning applications. However, the rules in an expert system knowledge base are based on current known expertise. Induction procedures offer the potential to learn rules that are are currently unknown. Clearly, humans are not competitive in this form of analysis. Using strictly empirical data, it is unlikely that a human can find a better rule than the computer. While the same argument could be made for a purely statistical analysis, decision rules are more consistent with human decision-making. With improved techniques and faster computers, we can expect to see greater use of induction techniques to help discover new decision rules and to verify and refine the quality of current rules acquired from experts.

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, 1988], 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.

References

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

[Breiman, Friedman, Olshen, and Stone, 1984]
Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Tress.* Monterrey, Ca.: Wadsworth, 1984.

[Clancey, 1985]
Clancey, W. "Heuristic Classification." *Artificial Intelligence.* 27 (1985) 289-350.

[Crawford, 1988]
Crawford, S. "Extensions to the CART Algorithm." *International Journal of Man-Machine Studies.* (1988) in press.

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

[Efron, 1982]
Efron, B. "The Jackknife,the Bootstrap and Other Resampling Plans." In *SIAM.* Philadelphia, Pa., 1982.

[Efron, 1983]
Efron, B. "Estimating the Error Rate of a Prediction Rule." *Journal of the American Statistical Association.* 78 (1983) 316-333.

[Fisher, 1936]
Fisher, R. "The Use of Multiple Measurements in Taxonomic Problems." *Ann. Eugenics.* 7 (1936) 179-188.

[Foroutan and Sklansky, 1985]
Foroutan, I. and Sklansky, J. "Feature Selection for Piecewise Linear Classifiers." In *IEEE Proc. on Computer Vision and Pattern Recognition.* San Franscisco, 1985, 149-154.

[Fukunaga, 1972]
Fukunaga, K. *Introduction to Statistical Pattern Recognition.* New York: Academic Press, 1972.

[Galen and Gambino, 1975]
Galen, R. and Gambino, S. *Beyond Normality: The Predictive Value and Efficiency of Medical Diagnoses.* New York: John Wiley and Sons, 1975.

[Ginsberg, Weiss, and Politakis, 1988]
Ginsberg, A., Weiss, S., and Politakis, P. "Automatic Knowledge Base Refinement for Classification Systems." *Artificial Intelligence.* (1988) 197-226.

[Highleyman, 1962]
Highleyman, W. "The Design and Analysis of Pattern Recognition Experiments." *Bell System Technical Journal.* 41 (1962) 723-744.

[Jain, Dubes, and Chen, 1987]
Jain, A., Dubes, R., and Chen, C. "Bootstrap Techniques for Error Estimation." *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 9 (1987) 628-633.

[James, 1985]
James, M. *Classification Algorithms*. New York: John Wiley & Sons, 1985.

[Kanal and Chandrasekaran, 1971]
Kanal, L. and Chandrasekaran "On Dimensionality and Sample Size In Statiistical Pattern Classification." *Pattern Recognition*. (1971) 225-234.

[Kononenko, Bratko, and Roskar, 1986]
Kononenko, I., Bratko, I., Roskar, E. "ASSISTANT: A System for Inductive Learning." *Informatica*. 10 (1986) .

[Lachenbruch and Mickey, 1968]
Lachenbruch, P. and Mickey, M. "Estimation of Error Rates in Discriminant Analysis." *Technometrics*. (1968) 1-111.

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

[McClelland and Rumelhart, 1988]
McClelland, J. and Rumelhart, D. *Explorations in Parallel Distributed Procesing*. Cambridge, Ma.: MIT Press, 1988.

[Michalski, Mozetic, Hong, and Lavrac, 1986]
Michalski, R., Mozetic, I., Hong, J., and Lavrac, N. "The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains." In *Proceedings of the Fifth Annual National Conference on Artificial Intelligence*. Philadelphia, Pa., 1986, 1041-1045.

[Narenda and Fukunaga, 1977]
Narenda, P. and Fukunaga, K. "A Branch and Bound Algorithm for Feature Subset Selection." *IEEE Transactions on Computers*. C-26 (1977) 917-922.

[Quinlan, 1986]
Quinlan, J. "Induction of Decision Trees." *Machine Learning*. 1 (1986) 1.

[Quinlan, 1987a]
Quinlan, J. "Simplifying Decision Trees." *International Journal of Man-Machine Studies*. (1987) in press, also Tech. Report 87.4, New South Wales Intitute of Tecnology, School of Computing Sciences.

[Quinlan, 1987b]
Quinlan, J. "Generating Production Rules from Decision Trees." In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*. Milan, Italy, 1987, 304-307.

[Quinlan, 1987c]
Quinlan, J. "Induction, Knowledge and Expert Systems." In *Australian Joint Conference On AI*. Sydney, Australia, 1987, .

[Roberts, 1984]
Roberts, S. "A Branch and Bound Algorithm for Determining the Optimal Feature Subset of Given Size." *Applied Statistics*. 33 (1984) 236-241.

[Steen, 1988]
Steen, L. "The Science of Patterns." *Science*. 240 (1988) 611-616.

[Stone, 1974]
Stone, M. "Cross-Validatory Choice and Assessment of Statistical Predictions." *Journal of the Royal Statistical Society.* 36 (1974) 111-147.

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

[Weiss, Galen, and Tadepalli, 1987]
Weiss, S., Galen, R., and Tadepalli, P. "Optimizing the Predictive Value of Diagnostic Decision Rules." In *Proceedings of the Sixth Annual National Conference on Artificial Intelligence.* Seattle, Washington, 1987, in press

[Wilkins and Buchanan, 1986]
Wilkins, D. and Buchanan, B. "On Debugging Rule Sets When Reasoning Under Uncertainty." In *Proceedings of the Fifth Annual National Conference on Artificial Intelligence.* Philadelphia, Pa., 1986, 448-454.