MANAGEMENT SCIENCE
Vol. 38, No. 1, January 1992
Printed in U.S.A.

A COMPOSITE APPROACH TO INDUCING KNOWLEDGE FOR EXPERT SYSTEMS DESIGN*

TING-PENG LIANG
Krannert Graduate School of Management, Purdue University, Krannert Building, West Lafayette, Indiana 47907

Knowledge acquisition is a bottleneck for expert system design. One way to overcome this bottleneck is to induce expert system rules from sample data. This paper presents a new induction approach called CRIS. The key notion employed in CRIS is that nominal and nonnominal attributes have different characteristics and hence should be analyzed differently. In the beginning of the paper, the benefits of this approach are described. Next, the basic elements of the CRIS approach are discussed and illustrated. This is followed by a series of empirical comparisons of the predictive validity of CRIS versus two entropy-based induction methods (ACLS and PLS1), statistical discriminant analysis, and the backpropagation method in neural networks. These comparisons all indicate that CRIS has higher predictive validity. The implications of the findings for expert systems design are discussed in the conclusion of the paper.

(KNOWLEDGE ACQUISITION; RULE INDUCTION; EXPERT SYSTEMS; EMPIRICAL LEARNING)

1. Introduction

Expert systems (ES) designed to support or replace human experts have drawn considerable attention in the past several years. Business applications have been reported in areas such as accounting, finance, manufacturing, marketing, taxation, and others (see, e.g., Chandler and Liang 1990). In general, evidence indicates that under certain circumstances expert systems outperform human experts (e.g., Yu et al. 1979) and can be used as valuable decision aids (Liang 1988; Turban and Watkins 1986).

The process of developing an ES includes acquiring knowledge from human experts, representing and organizing the knowledge, storing the knowledge in a knowledge base, and then applying a deductive inference mechanism (usually called the *inference engine*) to the knowledge base for decision making. For most systems, the knowledge acquisition stage plays a key role in determining the quality of the resulting system. Knowledge acquisition usually involves eliciting, analyzing, and interpreting the knowledge human experts use in solving a particular problem, and then transforming this knowledge into a proper representation. Traditionally, knowledge engineers have played an important role in this process. They use techniques such as structured interviews and protocol analyses to elicit knowledge from human experts (called *domain experts*). The domain experts formulate their knowledge and the knowledge engineers encode this knowledge for use by the system (see Kidd 1987 for an introduction to these techniques).

A major problem with this approach is that human experts frequently have difficulty in articulating their knowledge accurately. Acquisition can be a time-consuming process and frequently results in inconsistent knowledge bases. To overcome these problems, a number of researchers have suggested an alternative approach that takes advantage of inductive inference mechanisms to induce decision rules from data (e.g., Carter and Catlett 1987; Greene 1987; Quinlan 1986). Knowledge engineers collect data from previous decisions, identify key attributes (variables) with the help of domain experts, and then use an induction program to construct a set of rules for decision making. The core

* Accepted by George P. Huber, former Departmental Editor; received January 1989. This paper has been with the author 4 months for 3 revisions.

0025-1909/92/3801/0001$01.25

of this approach is an inductive algorithm that accepts a set of data as inputs and produces "If-Then" rules capable of interpreting the data set. Compared to the traditional approach, inductive knowledge acquisition (also called *rule induction*, *inductive learning* or *learning from examples*) generates more consistent rules, and the knowledge engineering process depends less heavily on domain experts.

The key to a successful inductive knowledge acquisition is the reliability of the rule induction algorithm. Induction is an inferential process that develops a structure from instances. It has been a standard methodology in business research for a long time. For example, most statistical classification methods such as regression analysis, discriminant analysis, Probit and Logit are inductive in nature. Rule induction mechanisms are different from statistical methods in two ways. First, the resulting structure is a set of "If-Then" rules rather than mathematical equations. Second, the rule induction algorithm may be based on criteria different from sample mean and variance.

Quinlan's ID3, a popular rule induction algorithm, for instance, uses entropy to measure the information content of each attribute and then derives rules through a repetitive decomposition process that minimizes the overall entropy (Quinlan 1979). Although recent research findings indicate that rules generated by this approach outperform both expert judgments and models derived from statistical discriminant analysis in stock market prediction (Braun and Chandler 1987), loan default (Shaw and Gentry 1988), and bankruptcy analysis (Messier and Hansen 1988), the algorithm has several limitations. First, since it uses a repetitive decomposition process, real numbers must be converted to integers. This may reduce the accuracy of the results. Second, the repetitive decomposition process is inefficient when the sample size is large. Third, the entropy does not consider the distribution of data and hence it is difficult to assess the probabilities associated with rules. Finally, a single algorithm is used to process both *nominal* (also called *categorical*, e.g., male and female) and *nonnominal* (e.g., financial ratios) attributes with completely different properties.

These limitations have reduced the performance and applicability of the algorithm. For example, Liang, Chandler, Han, and Roan (1992) found that ID3 performed rather poorly if the domain was dominated by nonnominal variables. In order to alleviate these shortcomings, researchers have extended the original mechanism to incorporate probability assessment (e.g., Cleary 1987; Quinlan 1987b), to prune or balance the generated decision trees (e.g., Quinlan 1987a), or to replace entropy with another measure (e.g., Goodman and Smyth 1988, 1989; Mingers 1989). Although these modified mechanisms avoid one problem or another, they still have two major drawbacks. First, they process nominal and nonnominal variables in the same way without taking into consideration their different characteristics. Second, the probability assessments are typically based on the frequency of occurrence in the training data set. Although this measure is fine when nominal attributes are involved (there is no other choice in this case), a more accurate method should be used for nonnominal attributes. This is supported by the finding that Probit (a method based on the normality assumption) significantly outperformed ACLS (an improved version of ID3) in predictive accuracy when the domain was nonnominal in nature (Liang, Chandler, Han, and Roan 1992). Mingers (1989) also reported that, after comparing seven different measures for replacing entropy, no significant improvement in predictive accuracy was found.

The goal of this paper is to present a new approach, called a Composite Rule Induction System (CRIS), to overcome these problems. The approach assesses probabilities for rules and applies different methods to handle nominal and nonnominal attributes. Instead of using a single measure such as entropy to handle both nominal and nonnominal attributes, it uses a cross-tabular approach to process nominal attributes and a statistical inference approach to handle nonnominal attributes. A rule scheduling mechanism is then applied to determine the relative importance of the candidate rules and to select the rule set accordingly. Furthermore, it uses sample distributions to infer the population for nonnominal attributes and then estimates the probabilities associated with those rules accordingly.

In the remainder of the paper, CRIS will be discussed in detail. Empirical results comparing the predictive accuracy of CRIS with those of the entropy-based approaches (ACLS and PLS1), the statistical discriminant analysis, and the backpropagation method in neural networks will also be presented.

2. The Rule Induction Problem

The goal of a rule induction algorithm is to construct a set of rules from data to interpret the data and facilitate decision making when a new case is encountered. The input data set includes a number of cases, each of which has values for a dependent attribute and several independent attributes. The dependent attribute is usually nominal, such as bankrupt or not. The independent attributes can be nominal or nonnominal. The resulting structure is composed of rules in the following format:

If ($X \alpha V$) Then ($Y \beta C$) With (Probability γ P) where

X = a certain independent attribute,

V = a hurdle value of the attribute,

Y = the dependent attribute,

C = a value of the dependent attribute,

P = a probability value,

α, β, and γ = relational operators,

$\alpha, \beta, \gamma \in \{=, \geq, \leq, >, <\}$.

EXAMPLE. If humidity ≥ 0.95 Then weather = raining With prob ≥ 0.8.

In order to generate rules, a rule induction mechanism must determine (1) the independent attribute to be considered, (2) the hurdle value of the independent attribute, (3) the corresponding value of the dependent attribute, (4) the probability associated with the rule (i.e., the likelihood that the rule is true), and (5) three relational operators.

There are a number of potential errors in rule induction. In general, these errors fall into two categories: data errors and method errors. Data errors include random errors, sampling errors, measurement errors, and factor errors. *Random errors* occur and are considered natural in most domains. *Sampling errors* occur when the sample is not a true representation of the domain. This often happens in situations where the sample size is small or the sampling method is significantly biased. *Measurement errors* occur when data are wrongly recorded. *Factor errors* occur when the key attributes are not included in the dataset.

Errors due to rule induction methods include hurdle value errors, sequencing errors, and post-treatment errors. *Hurdle value errors* mean that the hurdle values of the rules are determined improperly. For example, a rule induction method may induce a rule "if body temperature is higher than 103 degree, then call your doctor," while the proper rule should be "if the body temperature is higher than 101, then call your doctor." *Sequencing errors* mean that rules are organized improperly. Some existing methods use post-treatments such as pruning to refine a rule structure. Errors due to post-treatments are called *post-treatment errors*. For instance, pruning errors are introduced if good rules are mistakenly cut by pruning heuristics.

Given these potential errors, a good rule induction algorithm should minimize the overall errors through delicate tradeoffs. For example, a pruning heuristic works well if the sampling and sequencing errors it removes are greater than the pruning errors it introduces. Ideally, a good method should tolerate random errors, alleviate sampling errors, and contain no hurdle value and sequencing errors. Because random and sampling errors in the training data are difficult to detect, most research in rule induction focuses on reducing errors due to the method. For example, research on using measurement functions other than the entropy function in ID3 focuses on reducing hurdle value errors. Look-ahead heuristics that take rule dependency into consideration (Tu 1989) try to reduce sequencing errors by considering the combined performance of a rule and its possible successors in the rule selection process. Furthermore, using post-treatments such as tree pruning to avoid overfitting the training data (Breiman, Friedman, and Stone 1984; Quinlan 1987a) involves tradeoffs between sampling, sequencing, and pruning errors.

One problem with the existing rule induction algorithms is that they use a single measurement function to determine the hurdle values for both nominal and nonnominal attributes with substantially different characteristics. For example, the mean and variance of a nonnominal attribute provide valuable information of random errors, whereas those of a nominal attribute may not be useful for rule induction. Therefore, failure to take advantage of distributive information for nonnominal attributes usually results in higher hurdle value errors. An algorithm that treats different types of attributes differently is likely to make improvements. Along this line of reasoning, the CRIS algorithm that processes nominal and nonnominal attributes separately to reduce hurdle value errors is developed.

3. CRIS: A Composite Rule Induction System

In CRIS, the five functions of a rule induction mechanism are performed by three major components:

(1) A *hypothesis generator* that determines hurdle values and the proper relationship between independent and dependent attributes;

(2) A *probability calculator* that determines the probability associated with each rule; and

(3) A *rule scheduler* that determines how candidate rules should be organized to form a structure.

The interaction of the first two components generates candidate rules. These candidate rules are intermediate results that form a rule space. The rules in the rule space may be redundant or conflicting with each other. The third component selects a set of salient rules from the rule space and organizes them into an optimum structure. They are discussed below.

3.1. *Hypothesis Generation*

The first step for CRIS to induce decision rules is to generate hypotheses concerning possible causal relationships in the input data. A *hypothesis* is a preliminary "If-Then" rule whose probability is to be determined by the probability calculator and whose interpretative power is to be determined by the rule scheduler. The purpose of hypothesis generation is to determine hurdle values and identify causal relationships between dependent and independent attributes. CRIS uses two different methods to generate hypotheses for nominal and nonnominal attributes.

3.1.1. *Nominal Attributes*. For nominal attributes, the values are simply arbitrary identifications of different properties. Their mean and variance provide little useful information for rule induction. The attribute "bankruptcy," for example, may have values 1 (yes) and 0 (no). An average value of 0.5 has little use in this case. Therefore, CRIS adopts a cross-tabular approach to determine the relationship between nominal attributes and the dependent attribute:

1. For each nominal attribute, classify all cases in the input data set by their attribute values v_i ($i = 1, \ldots, m$) and dependent attribute values c_j ($j = 1, \ldots, n$), and then count the number of cases (f_{ij}) in each combination. The result of this step is an occurrence frequency table:

		Y		
		c_1	c_j	c_n
	v_1	f_{11}	$\cdots$	f_{1n}
X	v_i	$\cdots$	f_{ij}	$\cdots$
	v_m	f_{m1}	$\cdots$	f_{mn}

2. For each $X = v_k$ ($k = 1, \ldots, m$), select a $Y = c_s$, where $f_{ks} = \max\{f_{kj} \mid j = 1, \ldots, n\}$, to formulate the hypothesis, "If $X = v_k$ Then $Y = c_s$." If there is a tie, all possible hypotheses are generated. Since attribute X has m levels ($k = 1, \ldots, m$), the total number of hypotheses to be generated for the attribute is m (plus the number of ties).

3. Repeat steps 1 and 2 until hypotheses are generated for all nominal attributes.

[EXAMPLE] A set of bankruptcy data shown in Table 1 is used to illustrate the process. CRIS generates the following occurrence frequency table for attribute $V2$ (step 1).

		$V1$	
		0	1
$V2$	0	10	6
	1	0	4

Two hypotheses can be generated for $V2$ (step 2):

H1: If $V2 = 0$ Then $V1 = 0$.

H2: If $V2 = 1$ Then $V1 = 1$.

3.1.2. *Nonnominal Attributes*. For nonnominal attributes, sample mean and variance provide valuable information about the population and hence are useful for hypothesis formulation.

In a two-class classification problem, assuming distributions of attribute X for classes i (i.e., $Y = c_i$, $i = 1$ or 2) are $[\mu_i, \sigma_i^2]$,[1] then we use $\bar{X}_i$ and S_i^2 to estimate μ_i and σ_i^2. In order to differentiate these two classes, we first find a value X_c where cases are equally likely to be classified as either class (see Figure 1). This value is called the *cut* between these two classes, which means that if the attribute value of a case is higher (lower) than the cut, then the case is likely to fall in the class with the higher (lower) mean. By assuming that attribute values in both classes are normally distributed, the cut can be calculated by the following equation:[2]

$$X_c = \frac{S_1 \bar{X}_2 + S_2 \bar{X}_1}{S_1 + S_2}. \tag{1}$$

The cut provides a basic hurdle value for hypothesis formulation. If $\bar{X}_2 \geq \bar{X}_1$, for instance, then two hypotheses can be formulated:

[1] μ and σ^2 stand for population mean and variance, whereas $\bar{X}$ and S^2 stand for sample mean and variance.

[2] The normality assumption is chosen for implementation because of its popularity. In fact, other distributions can also be used. For example, if evidence indicates that the data fit a logistic normal distribution or an exponential distribution, then cut values and probabilities can be calculated based on the chosen distribution. The CRIS method itself is independent of the particular data distribution chosen for implementation. A possible extension of the CRIS approach, therefore, is to incorporate a pre-processor for determining the most likely data distribution and then generate cut values and probabilities accordingly.

TABLE 1

A Set of Bankruptcy Data

ID	*V*1	*V*2	*V*3	*V*4	*V*5
1	0	0	0.1113	0.3880	1.9862
2	0	0	0.0537	0.2087	1.6827
3	0	0	0.0178	0.4831	1.3325
4	0	0	0.0136	0.2014	0.7537
5	0	0	0.0975	0.4730	2.7911
6	0	0	0.1237	0.2982	2.8921
7	0	0	0.0539	0.5189	2.5375
8	0	0	0.1921	0.4395	2.9946
9	0	0	0.0777	0.3689	2.5478
10	0	0	−0.0621	0.7563	2.1047
11	1	1	−0.0656	1.5557	2.9152
12	1	0	0.0189	0.2409	1.2443
13	1	1	−0.1953	0.0113	0.0015
14	1	0	−0.1356	0.4794	2.4443
15	1	1	−0.0038	0.6956	1.9334
16	1	0	0.0118	0.9479	0.1530
17	1	0	0.0029	0.3398	1.8195
18	1	1	0.0448	0.8165	1.4482
19	1	0	−0.1046	0.7100	1.1111
20	1	0	−0.0569	0.3652	2.2768

Where:

*V*1: bankruptcy; 0 = no; 1 = yes;

*V*2: auditor's opinion; 0 = unqualified, 1 = qualified opinion;

*V*3: the ratio of net income/total assets;

*V*4: the ratio of current assets/total assets;

*V*5: the ratio of current assets/current liabilities.

(1) If $X \geq X_c$ Then $Y = c_2$.

(2) If $X < X_c$ Then $Y = c_1$.

Although these hypotheses can be used directly for classification, their accuracy is frequently below the desirable level. In order to ensure the quality of the resulting model, therefore, hypotheses with higher classification accuracy must be developed. In other words, the hurdle value needs to be higher than X_c for hypothesis (1) and lower than X_c for hypothesis (2).

One way to find hurdle values with higher accuracy is to control the probability that a case falls in a particular class. The rationale for this approach is that the lower the probability that a case belongs to a certain class, the greater the probability that it belongs to other classes. In Figure 1, for example, $X_1(0.90)$ is the 90th percentile of X_1, which

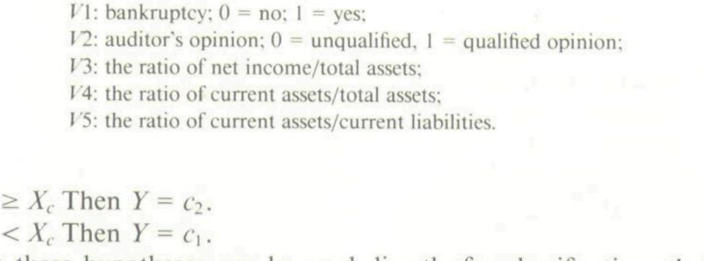
Class 1 ~ N($\bar{X}_1$, S_1^2)
Class 2 ~ N($\bar{X}_2$, S_2^2)
X_c
$\bar{X}_1$
$\bar{X}_2(0.10)$
$\bar{X}_1(0.90)$
$\bar{X}_2$

FIGURE 1. Basic Concept of Classification.

indicates that if the attribute value of a case is greater than $X_1(0.90)$, then the chance that the case falls into class 1 is less than 10%. Therefore, replacing X_c in hypothesis (1) with $X_1(0.90)$ will increase the accuracy of the hypothesis. Similarly, replacing X_c in hypothesis (2) with the 10th percentile of X_2, $X_2(0.10)$, will increase the accuracy of hypothesis (2).

Hurdle values identified by the above approach usually increase the classification accuracy for one class at the price of the other. Therefore, only one of the two potential hypotheses is useful. For example, $X_1(0.90)$ can be used to replace X_c in hypothesis (1), but not in hypothesis (2), to improve accuracy. Equations (2) and (3) show how these hurdle values can be calculated from sample mean, variance, and a probability, P. The $z(P)$ in the equation is the z-value at probability P of a standard normal distribution.[3] Equation (2) applies to the class with the lower mean, whereas equation (3) applies to the one with the higher mean.

$$X_i(P) = \bar{X}_i + z(P)*S_i, \tag{2}$$

$$X_i(1-P) = \bar{X}_i + z(1-P)*S_i. \tag{3}$$

Procedures for hypothesis formulation for nonnominal attributes can be summarized as follows:

1. Calculate the mean and variance of the attribute to be analyzed for each class.
2. Calculate the cut, X_c, to generate two basic hypotheses.
3. Specify the desired probabilities, and then generate more hypotheses based on the hurdle values calculated by equations (2) and (3).
4. Repeat steps 1 to 3 until hypotheses are generated for all nonnominal variables.

[EXAMPLE] The above procedures allows $V3$, $V4$, and $V5$ in the bankruptcy example to be analyzed. First, for each attribute, sample means and variances of bankrupt firms ($V1 = 1$) and nonbankrupt firms ($V1 = 0$) are calculated separately. Then the cut values are calculated from sample means and variances. Finally, by specifying the desired probabilities for hypothesis formulation, say 90% and 85%, hurdle values, $X_1(0.90)$, $X_2(0.10)$, $X_1(0.85)$ and $X_2(0.15)$, can be calculated. Table 2 shows the results of these three steps.

Based on the data in Table 2, the following hypotheses are formulated:

V3 (Net income/total assets).
H3: If $V3 \geq 0.0128$ Then $V1 = 0$.
H4: If $V3 < 0.0128$ Then $V1 = 1$.
H5: If $V3 \geq 0.0327$ Then $V1 = 0$.
H6: If $V3 \geq 0.0535$ Then $V1 = 0$.
H7: If $V3 < -0.0051$ Then $V1 = 1$.
H8: If $V3 < -0.0239$ Then $V1 = 1$.

V4 (Current assets/total assets).
H9: If $V4 \geq 0.4688$ Then $V1 = 1$.
H10: If $V4 < 0.4688$ Then $V1 = 0$.
H11: If $V4 \geq 0.5842$ Then $V1 = 1$.
H12: If $V4 \geq 0.6281$ Then $V1 = 1$.
H13: If $V4 < 0.1609$ Then $V1 = 0$.
H14: If $V4 < 0.0438$ Then $V1 = 0$.

V5 (Current assets/current liabilities).
H15: If $V5 \geq 1.8881$ Then $V1 = 0$.
H16: If $V5 < 1.8881$ Then $V1 = 1$.

[3] When the sample size is small, the z-value, $z(P)$, can be replaced by a t-value, $t(df, P)$, where df = degree of freedom.

TABLE 2

Analysis of Three Nonnominal Attributes

Attribute	Class	Mean	St. Dev.	X_c	$X_i(P1)^{1,3}$	$X_i(P2)^{2,3}$
$V3$	$V1 = 0$	0.0679	0.0664	0.0128	0.0535	0.0327
	$V1 = 1$	−0.0483	0.0736	0.0128	−0.0239	−0.0051
$V4$	$V1 = 0$	0.4136	0.1551	0.4688	0.0438	0.1609
	$V1 = 1$	0.6162	0.4139	0.4688	0.6281	0.5842
$V5$	$V1 = 0$	2.1622	0.6962	1.8881	2.7759	2.5220
	$V1 = 1$	1.5347	0.8975	1.8881	1.1994	1.3964

Notes: [1] The values are $X_i(0.90)$ for the class with higher mean (e.g., the first row in $V3$ is $X_{V3=1}(0.90)$) and $X_i(0.10)$ for the class with lower mean (e.g., the second row in $V3$ is $X_{V3=0}(0.10)$).

[2] The values are $X_i(0.85)$ for the class with higher mean (e.g., the first row in $V3$ is $X_{V3=1}(0.85)$) and $X_i(0.15)$ for the class with lower mean (e.g., the second row in $V3$ is $X_{V3=0}(0.15)$).

[3] Since the sample size was 10 for each class, t-values were used in calculating these hurdle values.

H17: If $V5 \geq 2.5220$ Then $V1 = 0$.
H18: If $V5 \geq 2.7759$ Then $V1 = 0$.
H19: If $V5 < 1.3964$ Then $V1 = 1$.
H20: If $V5 < 1.1994$ Then $V1 = 1$.

3.2. *Probability Assessment*

After a hypothesis is generated, the probability calculator determines its probability. This probability is conditional. It indicates the likelihood that the conclusion is true if the condition of the hypothesis is met.

For a problem with n classes, $c_1, \ldots, c_n$, the probability of a "greater-than" hypothesis, "If $X \geq v$ Then $Y = c_k$," is the conditional probability, $P(Y = c_k | X \geq v)$, which can be calculated from the prior probability of the class and other conditional probabilities. Two kinds of information are usually available from the input data: (1) the prior probability of class i, $P(Y = c_i)$, where $i = 1, \ldots, n$, and (2) the conditional probability that, given the class i, the probability that the attribute value of a case falls into a certain range, $P(X \geq v | Y = c_i)$. These two kinds of probabilities allow the desired posterior probability to be calculated by the following equation derived from the Bayesian Theorem:

$$P(Y = c_k | X \geq v) = \frac{P(Y = c_k) * P(X \geq v | Y = c_k)}{\sum_{i=1}^{n} P(Y = c_i) * P(X \geq v | Y = c_i)}. \qquad (4)$$

3.2.1. *Nominal Attributes*. For nominal attributes, information about data distribution is unavailable. Hence, the conditional probability is assessed by its relative frequency of occurrence in the training data. Because both the numerator and denominator are divided by the same constant (i.e., total number of occurrences), equation (4) can be simplified as follows (f_{vk} stands for the frequency in the situation where $X = v$ and $Y = c_k$):

$$P = \frac{f_{vk} * P(Y = c_k)}{\sum_{i=1}^{n} f_{vi} * P(Y = c_i)}. \qquad (5)$$

[EXAMPLE] Assuming that the prior probability is 0.5 for either class in the bankruptcy example, then the probabilities associated with H1 and H2 can be assessed as 0.625 (10/16) and 1.0 (4/4), respectively.

3.2.2. *Nonnominal Attributes*. For nonnominal attributes, the conditional probability $P(X \geq v | Y = c_i)$ is determined by the distribution of X for class i ($i = 1, \ldots, n$).

Assuming that the mean and standard deviation of the distribution are $\bar{X}_i$ and S_i, then the probability $P(X \geq v \mid Y = c_i) = 1 - P(z = (v - \bar{X}_i)/S_i)$. Hence, equation (4) can be transformed to:

$$P(Y = c_k \mid X \geq v) = \frac{P(Y = c_k) * \left(1 - P\left(z = \dfrac{v - \bar{X}_k}{S_k}\right)\right)}{\sum_{i=1}^{n} P(Y = c_i) * \left(1 - P\left(z = \dfrac{v - \bar{X}_i}{S_i}\right)\right)}. \tag{6}$$

Similarly, the equation for calculating the probability associated with a less-than hypothesis, "If $X \leq v$ Then $Y = c_k$," is:

$$P(Y = c_k \mid X \leq v) = \frac{P(Y = c_k) * P\left(z = \dfrac{v - \bar{X}_k}{S_k}\right)}{\sum_{i=1}^{n} P(Y = c_i) * P\left(z = \dfrac{v - \bar{X}_i}{S_i}\right)}. \tag{7}$$

[EXAMPLE] Assuming that the prior probability of bankruptcy or nonbankruptcy is 0.5, the probability associated with hypotheses H3 to H20 can be assessed. For example, the probability of hypothesis H6, "If $V3 \geq 0.0535$ Then $V1 = 0$," is calculated as follows (since the sample size is small, t-values are used to replace the z-values in equation (6)):

$$P(V1 = 0) = 0.5; \qquad P(V1 = 1) = 0.5;$$

$$P(V3 \geq 0.0535 \mid V1 = 0) = 1 - P\left(t\left(9, \frac{0.0535 - 0.0679}{0.0664}\right)\right) = 0.58;$$

$$P(V3 \geq 0.0535 \mid V1 = 1) = 1 - P\left(t\left(9, \frac{0.0535 - (-0.0483)}{0.0736}\right)\right) = 0.10.$$

Therefore,

$$P(V1 = 0 \mid V3 \geq 0.0535) = \frac{0.58}{0.10 + 0.58} = 0.85.$$

Because the sample means and variances of different classes may differ significantly, it is possible that the assessed probability for a certain hypothesis is lower than that of the cut hypothesis. In this case, the hypothesis needs to be modified. For example, the probabilities associated with hypotheses H13 and H14 are 0.32 and 0.17, respectively. These numbers indicate that it is more appropriate to hypothesize that $V1 = 1$ when $V4$ is less than 0.1609 or 0.0438. The probabilities of the new hypotheses are 0.68 and 0.83, respectively. If a hypothesis is dominated by its corresponding cut rule, then the hypothesis is removed from the rule space.

3.3. *Structure Construction*

A hypothesis, along with its associated probability, is called a *candidate rule*. General guidelines for determining the relational operators α, β, and γ for a candidate rule are: (1) γ is "=" when α is "="; (2) γ is "≥" when α is otherwise;[4] and (3) β usually is "=" if the dependent attribute is nominal.

Candidate rules are the basic elements of the knowledge base of an expert system. Because more than one candidate rule is generated for each attribute in the previous

[4] In practice, γ usually is "=", which means that the probability of the rule is at least equal to the specified value. This simplifies the representation of rules.

process, these rules may be redundant or inconsistent. Additionally, these rules are generated based on information concerning a single attribute. Therefore, a mechanism is necessary to evaluate the relative importance of these candidate rules and form a structure to classify correctly a maximum number of cases.

Unlike the ID3 algorithm that selects attributes based on their entropy values, the rule scheduler of CRIS examines the extent to which these rules cover the cases in the input file and then organizes them based on their saliency. The *saliency* of a candidate rule is defined as the difference between the number of cases correctly covered and those incorrectly interpreted by the rule. These numbers are called the *hit value* and *miss value* of the rule, respectively. The cases used for determining the saliency of a rule are called *training cases*. The resulting structure is a decision tree with rules as its nodes. The construction process includes:

1. *Determination of Rule Saliency*. Apply all rules to the training cases to determine their hit and miss values.

2. *Selection of a Rule*. The rules generated from cut values (called *cut rules*) and high accuracy rules (called *regular rules*) have different properties. The former provides an equal-likelihood split between classes, whereas the latter specifies hurdle values for higher accuracy in classifying a certain class. Therefore, the heuristic for rule scheduling includes two steps. First, the regular rules are selected to interpret as many training cases as possible. Then the cut rules are applied to cover the remainder in order to guarantee the completeness of the resulting structure. Guidelines for rule selection are:

2.1. If there are rules whose miss values are zero and whose hit values are positive, then select the one with the highest hit value.

2.2. If all rules have positive miss values, then calculate the saliency for each rule by deducting its miss value from its hit value and selecting the one with the highest positive saliency value.

2.3. If more than one rule has the same saliency value, then choose the one with the highest probability.

2.4. If more than one rule has the same saliency value and probability, then choose the one associated with the most significant attribute. The significance of an attribute is measured by the following formula. The higher the value is, the more significant the attribute is.

$$\text{Significance} = \frac{\sum_{i=1}^{n} (\bar{X}_i - \bar{X})}{\sum_{i=1}^{n} \sqrt{S_i^2 / n_i}}, \qquad \text{where} \tag{8}$$

$\bar{X}_i$ = mean of attribute X for class i;
$\bar{X}$ = overall mean of attribute X;
S_i^2 = variance of attribute X for class i;
n_i = number of cases for class i; and
i = number of classes in the data set.

3. *Redefinition of the Training Cases*. The selected rule splits the original set of training cases into two subsets: cases covered by the rule (both correctly and incorrectly) and the remainder.

3.1. *The covered set*. If all cases covered by the rule are correctly interpreted, then add the rule to the final structure and stop processing this subset. Otherwise, add the rule to the structure, assign the cases covered by the rule to be the new training set, and then go to step 1 for further analysis.

3.2. *The remainder*. If no case is left after applying a rule, then keep the existing training set and go to step 5 to find a pair of cut rules. Otherwise, assign the remainder to be the new training set and go to step 1.

4. *Iteration of the Process*. Repeat steps 1 to 3 for the regular rules until the termination conditions stated in 3.1 and 3.2 are met or no regular rules that have positive saliency

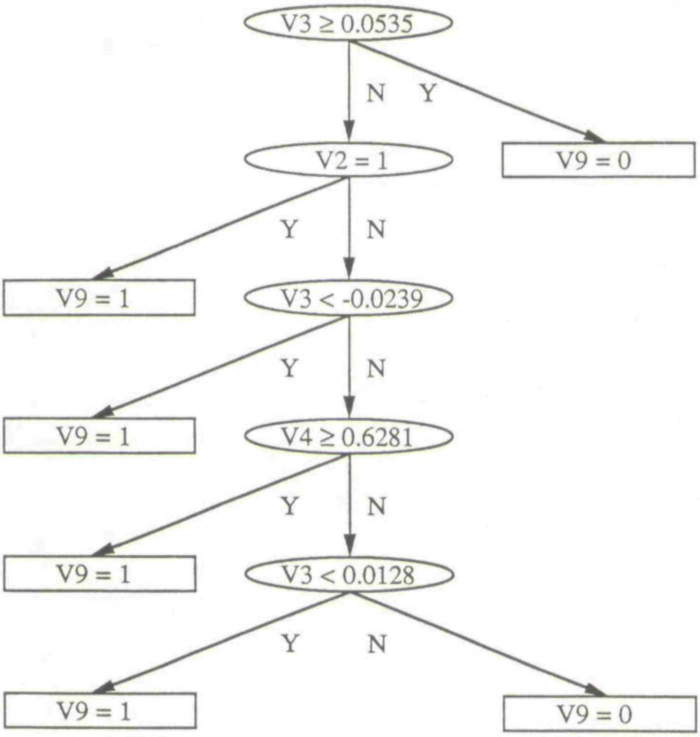

FIGURE 2. Resulting Decision Tree from the Example.

values exist. By the iteration process, the scheduler moves down the decision tree to construct the structure.

5. *Application of Cut Rules*. The cut rules are used when no regular rule is available for further classifying the training set. The procedures are the same as applying the regular rules, except that the cut rules must be applied in pairs and hence their saliency value is the sum of their individual values. It is possible to apply more than one set of cut rules to interpret a training set, as long as the number of cases correctly interpreted increases. The whole process stops when further improvement is impossible. Following these procedures, we can generate a rule structure as shown in Figure 2 to interpret the bankruptcy data.

In summary, the CRIS mechanism works in the following manner. First, a set of data containing a nominal dependent attribute and several independent attributes is entered. Then hypotheses are generated by the system's hypothesis generator. Based on different properties of nominal and nonnominal attributes, different algorithms are used for hypothesis generation. Third, the hypotheses are converted to candidate rules by assessing their probabilities and making necessary modification. Finally, the resulting candidate rules are evaluated and selected to form a decision structure that can interpret the existing cases and facilitate future decision making.

4. Empirical Evaluation of CRIS

Three experiments were conducted to evaluate the performance of CRIS. In the first experiment, data for bankruptcy prediction were used to compare CRIS with the existing entropy-based approaches, discriminant analysis, and the backpropagation mechanism of neural networks. Theoretically, these approaches make different assumptions on the distribution of data, use different criteria to evaluate the relative importance of attributes,

TABLE 3

Comparison of CRIS, ACLS, PLS1, BP and Discriminant Analysis

(1) Major assumptions

Discriminant analysis (DA)

—Data population is multivariate normal distribution

—No perfect correlation among independent attributes

—Equal covariance matrices for classes

ACLS algorithm (entropy-based)

—No conflict in the training set and numerical values must be integer

*PLS*1 *algorithm* (entropy-based)

—Numerical values must be integer

Backpropagation (BP)

—None

CRIS algorithm

—Nonnominal data follow a certain data distribution for each class

(2) Selection criteria, processes, and resulting models

Methods	MDA	Entropy-based	BP	CRIS
Selection criteria	Covariance matrix	Entropy	Delta rule	Rule saliency
Selection processes	Matrix operations	Repetitive decomposition	Simulation	Rule scheduling
Resulting models	Linear equations	Rule structure	Network structure	Rule structure

and generate different models from data. These differences are summarized in Table 3. It is reasonable to assume that they perform differently with different types of problems.

The results from the bankruptcy data indicate that CRIS outperforms the other four methods. To verify these results, a second experiment was conducted on sets of LIFO/FIFO choice data. Again, CRIS is proved to be more accurate.

In order to understand when and why CRIS or other rule induction methods work better, computer-generated data were used in a third experiment to examine how different data characteristics affect the performance of selected rule induction methods. This experiment demonstrates that data distribution and attribute correlation affect the relative performance of different methods.

4.1. *Bankruptcy Prediction*

The bankruptcy data set used in the experiment contains 50 cases. Each case includes four nominal and five nonnominal attributes. Twelve experiments were conducted on the bankruptcy data. In each experiment, the data set was randomly divided into a training set and a testing set. The training set contained cases used for inducing the model; the testing set contained hold-out cases for evaluating the predictive validity of the resulting model. All five methods were applied to each training set. The induced models were then used to predict the cases in the corresponding testing set. The accuracy of a model was measured by the number of cases correctly predicted by the model divided by the total number of cases in the testing set.

Twelve observations were obtained for each method. The tools used for running the entropy-based algorithms were ACLS (Analog Concept Learning System) and PLS1 (Probabilistic Learning System 1). ACLS is a modified version of the original ID-3 algorithm (see Braun and Chandler 1987 and Paterson and Niblett 1982 for details). PLS1 further allows probabilities to be estimated from the frequency of occurrence, called the utility of a rule (Rendell 1983, 1986). The discriminant analysis program used was the DISCRIM procedure in the SAS package.

The backpropagation algorithm is the most popular paradigm in neural networks. It allows cases to be classified by assigning proper weights to paths connecting processing elements (called *connection weights*). The weights are determined by a trial-and-error process that minimizes the difference between actual outputs and desired outputs. The initial weights are determined randomly. The outputs generated from these weights are compared with the desired outputs. The errors are then propagated backward to adjust the connection weights. This process continues until an acceptable level of errors or a certain number of iterations is reached. The software used in the experiment was adapted from the source code in Pao (1989). The network configuration was determined arbitrarily. It includes one hidden layer with three processing elements. For each data set, 1000 iterations were performed.

Two different sample sizes were used in the training set to determine whether sample sizes may have an effect on predictive accuracy. Six of these sets had 20 cases while the other six of them had 30 cases. All testing sets included 20 cases.

The results of the experiment indicate that among five methods CRIS had the highest average accuracy in predicting hold-out samples (0.771 for ACLS, 0.754 for PLS1, 0.758 for discriminant analysis, 0.783 for backpropagation, and 0.808 for CRIS). The results of the Wilcoxon paired rank tests indicate that CRIS outperforms PLS1 and MDA significantly in pairwise tests ($p = 0.0367$ and 0.0382, respectively). The difference between ACLS and CRIS is close to a 10% significance level. The difference between backpropagation (BP) and CRIS is not significant.[5] The effect of sample size is not significant.

4.2. *LIFO/FIFO Choice*

The LIFO/FIFO data included 58 pairs of training and testing data sets divided into two categories based on the relative effect of nominal attributes. Twenty-eight of them contained cases whose LIFO/FIFO choices were primarily affected by the industry type (a nominal attribute). They are called industry-dominated data sets (i.e., nominal attributes have dominant effects). The other 30 sets included cases whose LIFO/FIFO decisions were not strongly affected by the nominal variable. This allows the effect of nominal attributes on predictive accuracy to be examined.

In each category, three different sample sizes were examined. The training and testing data were paired in three different ways: (1) using a large-size training sample to predict a small-size testing sample (L/S), (2) using a medium-size training sample to predict a medium-size testing sample (M/M), and (3) using a small-size training sample to predict a large-size testing sample (S/L). Therefore, there were a total of six different settings. The sample sizes in the industry-dominated category were 98 (large), 73 (medium), and 49 (small). The sample sizes in the nondominated category were 78 (large), 58 (medium), and 39 (small). Each case was represented as one nominal variable and eight nonnominal variables. The dependent variable was the adopted inventory method, either LIFO or FIFO. A detailed description of variable selection and data collection can be found in Liang, Chandler, Han, and Roan (1992).

ACLS and CRIS were applied to all 58 pairs of training sets to derive rule structures for predicting the cases in their corresponding testing sets. The resulting average predictive accuracy in each setting is summarized in Table 4. Again, CRIS predicts more accurately than the entropy-based ACLS in both industry-dominated and nondominated situations. A three-way ANOVA test indicates that two factors are significant at the 1% level in this experiment: the nature of data and the method. Both methods perform better when

[5] Because the network configuration was determined arbitrarily, the results may not indicate the performance of neural networks in general. Unfortunately, there are no clear guidelines for selecting the most appropriate configuration. This is a research issue outside the scope of this paper.

TABLE 4

Empirical Results of the Second Experiment

	Industry-Dominated Sets				Nondominated Sets				Global Average
	L/S	M/M	S/M	Mean	L/S	M/M	S/L	Mean	
ACLS	90.0	89.0	88.3	89.1	62.3	62.6	60.9	61.9	75.5%
CRIS	92.2	89.8	91.2	91.1	69.5	70.1	67.8	69.1	80.1%

Note: The numbers are average predictive accuracy.

nominal attributes play a major role in the data sets. The effect of sample size is again insignificant. This indicates that both methods are insensitive to sample size.[6]

In order to understand further how these two methods performed, two ANOVA analyses were conducted on dominated and nondominated settings separately. In the nondominated settings, the method effect is significant at 1% level ($F = 22.8$). In the dominated settings, however, the method effect is not significant at 5% level ($F = 2.9$). This shows that the superiority of CRIS is primarily due to its ability to handle nonnominal attributes better and confirms our original assumption—different types of attributes should be processed in different ways.

In summary, at least three interesting findings have been observed in the first two experiments. First, CRIS outperforms other entropy-based methods when nonnominal attributes are important in the domain. Second, CRIS outperforms other entropy-based methods when attribute correlations exist. Finally, because the current implementation of CRIS assumes normal distribution of data, CRIS should perform better when this assumption is fulfilled. In the third experiment, the three observations stated above are evaluated on computer-generated data.

4.3. *Simulation on Computer-Generated Data*

Three factors were controlled in the third experiment: nature of domains, data distribution, and attribute correlation. Domains were divided into three types: purely nominal attributes, purely nonnominal attributes, and a mixture of the two. Each case included four attributes. In a purely nominal domain, all four attributes were nominal. In a purely nonnominal domain, all four attributes were nonnominal. In a mixed domain, two attributes were nominal and two were nonnominal.

Data distributions included two levels: normal and nonnormal. In the normal distribution situation, data were randomly generated from a multivariate normal distribution function. In the nonnormal situation, a bimodal distribution was chosen as a representative because its shape is clearly different from a normal distribution.

For both normal and nonnormal situations, attribute correlations were controlled in the data generation process. In the high correlation situation, the covariance matrices for classes 1 and 2 were $V1$ and $V2$, as follows. In the low correlation situation, the nondiagonal numbers in the matrices were zero.

$$V1 = \begin{bmatrix} 100 & 50 & 40 & 30 \\ & 64 & 30 & 20 \\ & & 25 & 10 \\ & & & 25 \end{bmatrix}, \qquad V2 = \begin{bmatrix} 100 & 50 & 40 & 30 \\ & 64 & 30 & 30 \\ & & 36 & 20 \\ & & & 36 \end{bmatrix}.$$

[6] Statistical methods are usually more sensitive to sample size. For example, the performance of Probit dropped significantly in the S/L setting (Liang, Chandler, Han, and Roan 1992). Both ACLS and CRIS are better than statistical methods in this aspect.

TABLE 5

Average Predictive Accuracy in Various Settings

Distribution		Normal		Bimodal	
Correlation		High	Low	High	Low
Nonnominal	ACLS	0.875	0.94	0.825	0.865
	CRIS	0.925	0.95	0.835	0.845
	PLS1	0.84	0.81	0.722	0.715
Mixed	ACLS	0.855	0.915	0.79	0.835
	CRIS	0.91	0.96	0.83	0.87
	PLS1	0.849	0.881	0.76	0.844
Nominal	ACLS	0.815	0.945	0.725	0.78
	CRIS	0.84	0.96	0.82	0.85
	PLS1	0.85	0.96	0.833	0.86

Combining these three factors resulted in twelve (3 × 2 × 2) different settings. For each setting, ten sets of data with 40 cases each were generated. Five of them were used for inducing rules while the other five were used for testing the rules. ACLS, CRIS, and PLS1 methods were applied to all data sets (totally 60 pairs of training and testing data sets). The predictive accuracy was then calculated for each method. Table 5 shows the average predictive accuracy of different methods in different settings.

The results shown in Table 5 indicate that ACLS performs well if the domain includes nonnominal attributes and attribute correlation is low. CRIS performs well if the domain includes nonnominal attributes and attribute correlation is high. PLS1 performs well if the domain includes nominal attributes and data distribution is bimodal.[7] A four-way ANOVA test results in the following significant effects: (1) data distribution ($F = 95.3$, $p < 0.01$), (2) attribute correlation ($F = 33.5$, $p < 0.01$), (3) induction method ($F = 14.5$, $p < 0.01$), (4) interaction of attribute type and attribute correlation ($F = 4.3$, $p < 0.05$), and (5) interaction of attribute type and induction method ($F = 11.4$, $p < 0.01$).

The significant effects of data distribution, attribute correlation, and method conclude that the predictive accuracy of the induced rules is affected by different data distribution, the degree of attribute correlation, and the rule induction method. The significance of the interaction effect of attribute type (i.e., nominal or nonnominal) and attribute correlation indicates that the predictive accuracy of a method is affected by their interaction. The significant interaction effect of attribute type and induction method indicates that method selection must take into consideration the types of attributes the domain has. In fact, patterns can be found easily if the data in Table 5 are read column-by-column. The predictive accuracy of CRIS and ACLS decreases, whereas the predictive accuracy of PLS1 increases from nonnominal to nominal.

Another fact worth discussion is that the predictive accuracy of CRIS is more stable than those of ACLS and PLS1. In the worst case (bimodal, high correlation, and nominal), the average predictive accuracy of CRIS is 0.82 (see Table 5), which is not significantly lower than that of the best method (0.833 for PLS1). The worst-case figures for ACLS and PLS1, however, are 0.725 and 0.715, respectively. These are significantly lower than the performance of the best method in their respective settings. This indicates the robustness of the CRIS method.

[7] A problem I found in PLS1 is that it completely depends on the occurrence frequency to define proper hyperplanes. As a result, the induced rules are frequently unable to classify certain testing cases if their attribute values fall out of the range revealed in the training data. This is a major reason that it performs poorly when nonnominal attributes are present.

Finally, a major question to be answered is whether the average predictive accuracy of CRIS is better than existing entropy-based rule induction methods such as ACLS and PLS1. Two univariate F-tests were performed for comparing the average predictive accuracy of these three methods. The results conclude that CRIS is significantly better than ACLS ($F = 11.257, p < 0.001$) and PLS1 ($F = 28.299, p < 0.0001$). The overall average predictive accuracy of the three methods is 0.847 for ACLS, 0.883 for CRIS, and 0.827 for PLS1.

5. Concluding Remarks

This article presents a new approach for inducing rules from data which can be used to acquire knowledge for developing expert systems. The major features that make it different from existing approaches are: (1) it uses different techniques to generate hypotheses for nominal and nonnominal attributes; (2) it uses sample distribution (for nonnominal attributes) and frequency table (for nominal attributes) approaches to estimate the probabilities associated with rules; and (3) it uses a rule scheduling technique to determine the relative importance of different attributes and to construct the optimum rule structure. The results of the empirical study indicate that the new approach outperforms the existing rule induction algorithms, ACLS and PLS1, and the statistical discriminant analysis in predictive accuracy.

Given the increased use of expert systems in various business areas, this work is a step toward improving the knowledge acquisition tool for expert system design. With the improved rule induction system, a knowledge engineer can construct knowledge by collecting previous cases solved by the experts, identifying attributes that may have effects on the decision (experts can provide valuable advice in these two stages), and executing a rule induction program. For those cases where rules are a good representation of the expert's knowledge, the tedious process of interview and protocol analysis can be reduced to a minimum level.

This research also provides a powerful tool for classification. An important implication of this work is that due to different natures of nominal and nonnominal attributes, methods applying a single criterion to process them may not lead to the optimum model. The entropy-based approaches are useful in handling nominal attributes, while the statistical methods are powerful only in handling nonnominal attributes. A proper integration of these methods can produce tools capable of constructing more accurate models (Liang, Chandler, and Han 1990). The CRIS project is only the first step. Further research is needed to investigate the following issues: (1) how the rule induction method can take advantage of theories developed in statistics, (2) how a globally superior method can be developed, (3) how information and misclassification costs can be taken into consideration in designing expert systems, and (4) how to choose the most appropriate method when a globally superior method does not exist. Works on these issues should provide much insight into improving the performance of expert systems.[8]

[8] The author thanks the editors and reviewers for their help in improving the manuscript, John Chandler and Larry Rendell for making available ACLS and PLS1 systems, Peter Silhan and Mark Silver for their comments on earlier versions of the paper. The author also thanks James C. McKeown for providing the bankruptcy data, Ingoo Han for performing some empirical and statistical analyses, the Campus Research Board and the Department of Accountancy of the University of Illinois for partial financial support.

References

BRAUN, H. AND J. S. CHANDLER, "Predicting Stock Market Behavior Through Rule Induction: An Application of the Learning-From-Example Approach," *Decision Sci.*, 18, 3 (1987), 415–429.

BREIMAN, L., J. H. FRIEDMAN AND C. J. STONE, *Classification and Regression Trees*, Wadsworth & Brooks, Monterey, CA, 1984.

CARTER, C. AND J. CATLETT, "Assessing Credit Card Applications Using Machine Learning," *IEEE Expert*, (Fall 1987), 71–79.

CHANDLER, J. C. AND T. P. LIANG, *Developing Expert Systems for Business Applications*, Merrill Publishing Co., Columbus, OH, 1990.

CLEARY, J. G., "Acquisition of Uncertain Rules in a Probabilistic Logic," *Internat. J. Man-Machine Studies*, 27 (1987), 145–154.

GOODMAN, R. M. AND P. SMYTH, "Decision Tree Design from a Communication Theory Standpoint," *IEEE Trans. Information Theory*, 34, 5 (1988), 979–994.

——— AND ———, "The Induction of Probabilistic Rule Sets—The ITRULE Algorithm," *Proc. Sixth Machine Learning Workshop*, 1989, 129–132.

GREENE, D. P., "Automated Knowledge Acquisition Overcoming the Expert System Bottleneck," *Proc. 8th ICIS Conf.*, Pittsburgh, 1987, 107–117.

KIDD, A. L., *Knowledge Acquisition for Expert Systems*, Plenum Press, New York, 1987.

LIANG, T. P., "Expert Systems as Decision Aids: Issues and Strategies," *J. Information Systems*, 2, 2 (1988), 41–50.

———, J. S. CHANDLER AND I. HAN, "Integrating Statistical and Inductive Learning Methods for Knowledge Acquisition," *Expert Systems with Appl.*, 1, 4 (1990), (forthcoming).

———, ———, ——— AND J. ROAN, "An Empirical Investigation of Some Data Effects on the Classification Accuracy of Probit, ID3 and Neural Networks," *Contemporary Accounting Res.*, (1992) (forthcoming).

MESSIER, W. F., JR. AND J. V. HANSEN, "Inducing Rules for Expert System Development: An Example Using Default and Bankruptcy Data," *Management Sci.*, 34, 12 (1988), 1403–1415.

MINGERS, J., "An Empirical Comparison of Selection Measures for Decision-Tree Induction," *Machine Learning*, 3 (1989), 319–342.

PAO, Y., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, MA, 1989.

PATERSON, A. AND T. NIBLETT, *ACLS User Manual*, Intelligent Terminal Ltd., Glasgow, Scotland, 1982.

QUINLAN, J. R., "Discovering Rules from Large Collections of Examples: A Case Study," in D. Michie (Ed.), *Expert Systems in the Micro Electronic Age*, Edinburgh University Press, Edinburgh, Scotland, 1979.

———, "Induction of Decision Trees," *Machine Learning*, 1, 1 (1986), 81–106.

———, "Simplifying Decision Trees," *Internat. J. Man-Machine Studies*, 27 (1987a), 221–234.

———, "Decision Trees as Probabilistic Classifiers," *Proc. Fourth Internat. Workshop on Machine Learning*, Morgan Kauffman, Los Altos, CA, 1987b, 31–37.

RENDELL, L., "New Basis for State-Space Learning Systems and a Successful Implementation," *Artificial Intelligence*, 20 (1983), 369–392.

———, "A General Framework for Induction and a Study of Selective Induction," *Machine Learning*, 1 (1986), 177–226.

SHAW, M. J. AND J. A. GENTRY, "Using an Expert System with Inductive Learning to Evaluate Business Loans," *Financial Management*, 17, 3 (1988), 45–56.

TU, P., "Toward an Intelligent Classification-Tree Approach to Problem-Solving," Unpublished Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1989.

TURBAN, E. AND P. R. WATKINS, "Integrating Expert Systems and Decision Support Systems," *MIS Quart.*, 10, 2 (1986), 121–136.

YU, V. L. ET AL., "Antimicrobial Selection by Computers: A Blinded Evaluation to Infectious Disease Experts," *J. Amer. Medical Assoc.*, 242, 21 (1979), 1279–1282.