An Information Theoretic Approach to Rule Induction from Databases

Padhraic Smyth, *Member, IEEE*, and Rodney M. Goodman, *Member, IEEE*

Abstract— **The knowledge acquisition bottleneck in obtaining rules directly from an expert is well known. Hence, the problem of *automated* rule acquisition from data is a well-motivated one, particularly for domains where a database of sample data exists. In this paper we introduce a novel algorithm for the induction of rules from examples. The algorithm is novel in the sense that it not only learns rules for a given concept (classification), but it simultaneously learns rules relating multiple concepts. This type of learning, known as generalized rule induction is considerably more general than existing algorithms which tend to be classification oriented. Initially we focus on the problem of determining a quantitative, well-defined rule preference measure. In particular, we propose a quantity called the *J-measure* as an information theoretic alternative to existing approaches. The J-measure quantifies the information content of a rule or a hypothesis. We will outline the information theoretic origins of this measure and examine its plausibility as a hypothesis preference measure. We then define the ITRULE algorithm which uses the newly proposed measure to learn a set of optimal rules from a set of data samples, and we conclude the paper with an analysis of experimental results on real-world data.**

Index Terms—**Cross entropy, expert systems, information theory, machine learning, knowledge acquisition, knowledge discovery, rule-based systems, rule induction.**

I. A Statement of the Problem

CONSIDER a company which has a large database of information, which is, perhaps, lying idle. For example, a telecommunications company might have logged hundreds of thousands of trouble reports, or a financial services company might have a database of past loan applications and credit histories of their customers. With the advent in recent years of inexpensive electronic and magnetic storage media and the increased use of office automation, such databases are quite commonplace. The company wishes to develop a rule-based expert system for the domain to which the data applies. The application of this expert system could be for prediction, diagnosis, simulation, training purposes, etc. Can one use the existing database to automatically derive rules for the expert system? The purpose of this paper is to set forth a basic theory for automated rule induction using information theory and describe the ITRULE algorithm which precisely addresses this task. The motivation and rationale for using rule-based expert systems is well documented and will not be repeated here. The problem or "bottleneck" of manual knowledge acquisition for such systems is perhaps their major drawback. It is notoriously difficult to obtain rules directly from human experts [1]–[3]. Furthermore, if the domain necessitates reasoning under uncertainty (probabilistic reasoning), humans are well known to be inconsistent in their description of subjective probabilities (Kahneman *et al.*, [4]). Hence, it is quite clear that if our hypothetical company has an existing database of sample data available, a rule induction algorithm would be very useful. As we shall see, the problem can be rendered more general than simply deriving rules for an expert system — in a sense we are involved in a data reduction process, where we want to reduce a large database of information to a small number of rules describing the domain.

Consider that we have M observations or samples available, e.g., the number of items in a database. Each sample datum is described in terms of N attributes or features, which can assume values in a corresponding set of N discrete alphabets. For example, our data might be described in the form of 10-component binary vectors. We note that this representation can be transformed into an N-fold discrete contingency table as is commonly referred to in multivariate statistical analysis. However, for $N > 2$, the contingency table representation is awkward and consequently we will prefer to think of the input data as simply a list of M attribute vectors. We will not dwell on statistical aspects of the problem (statistical analyses of contigency tables are well documented elsewhere, e.g., Bishop *et al.* [5]) except to note that we implicitly assume throughout that the sample data is a true random sample of the population at large. The requirement for *discrete* rather than *continuous*-valued attributes is dictated by the very nature of the rule-based representation. It is worth noting, however, that techniques for converting both continuous and mixed mode data are available but will not be described here [6].

In addition it is important to note that we do *not* assume that the sample data are somehow exhaustive and "correct." In the field of machine learning and/or artificial intelligence it is often assumed, for a classification problem say, that any given attribute can be perfectly described in terms of the other $M - 1$ attributes. In this case, the learning problem reduces to a simple search of the $M - 1$ dimensional "hypothesis space," i.e., the space of possible classifiers based on functions of the predictor attributes. While this assumption may hold true in

Manuscript received October 25, 1989; revised April 16, 1990. This work was supported in part by Pacific Bell, in part by the U. S.Army Research Office under Contract DAAL03-89-K-0126 and by the California Institute of Technology's program in Advanced Technologies sponsored by Aerojet General, General Motors, and TRW. Part of this work was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

P. Smyth is with the Communications Systems Research Section, Jet Propulsion Laboratory 238–420, California Institute of Technology, Pasadena, CA 91109.

R. M. Goodman is with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125.

IEEE Log Number 9200960.

certain domains such as game playing, it is rarely if ever true in real-world problems. Typically, the chosen attributes can only incompletely specify each other, *at best*. Hence, our viewpoint is very much in line with the statistical pattern recognition philosophy as opposed to what might be termed the artificial intelligence or symbolic learning approach. We will return to this point later.

Our approach is inherently probabilistic, i.e., we adhere to the basic axioms of probability theory rather than adopting any of the more recent uncertainty paradigms such as the Dempster–Shafer or fuzzy logic theories. The rationale for statistical models as a *necessary* (though not necessarily *sufficient*) component of a general model of learning and reasoning under uncertainty has been clearly stated elsewhere (Lindley [7], Cheeseman [8]) and will not be repeated here.

A *rule* is a statement to the effect that "if event i occurs, then event j will probably occur," where the events are propositions of the form of attribute A_i taking on some particular value from its alphabet. In general, the rule has an associated belief parameter such as a conditional probability or a "certainty factor." For our purposes we will use the conditional probability p(event j|event i). Given sample data as described earlier we pose the problem as follows: can we find the "best" rules from a given data set, say the K best rules ? We will refer to this problem as that of *generalized rule induction*, in order to distinguish it from the special case of deriving *classification* rules. Classification only derives rules relating to a single "class" attribute, whereas generalized rule induction derives rules relating any or all of the attributes. Clearly, we require a preference measure to rank the rules and a learning algorithm which uses the preference measure to find the K best rules. This paper reviews our recently introduced rule preference measure known as the J-measure [9], but is primarily focused on the *learning* aspect of the problem and, in particular, the ITRULE algorithm.

Beginning with a review of related work on rule induction algorithms, we will see that existing approaches lack robustness and generality for the problem we have described. We then define in Section III the basic rule preference measure and outline its information theoretic properties. Section IV analyzes the measure from a general theory of learning viewpoint. It is established that the measure is consistent in the sense that it trades-off hypothesis simplicity with goodness-of-fit. In Section V we explore in more detail the nature of this trade-off and in particular establish some information theoretic bounds. These bounds are used in Section VI where we define the ITRULE algorithm itself. Section VII contains experimental results and analysis on real-world data sets.

II. Background on Rule Induction Algorithms

Comparison of learning algorithms is quite difficult since many algorithms address different goals and are based on different implicit assumptions. However, there are a few broad dimensions along which we can classify these approaches. Induction, or learning from examples, can be viewed as a search for hypotheses (restricted to some *hypothesis space*) to account for a set of instances or examples which are often assumed to be restricted to some *instance space*. For the purposes of this paper, the hypothesis space will be restricted to the conjunctive propositions in the discrete space defined by the Cartesian product of the sample spaces of the individual attributes—the extension to more general hypothesis spaces remains a topic for further investigation. For a *given concept* (in our terminology, a particular attribute value pair) the *hypothesis* space is defined as the Cartesian product of the sample spaces of the other $N-1$ individual attributes, whereas the *instance* space is defined over the entire N-dimensional product space.

In general, the learning problem consists of being given positive and negative instances of some *concept* and trying to find a hypothesis in the hypothesis space which best describes this concept. Let v be any positive instance in the instance space for some concept. Symbolic algorithms try to find a deterministic mapping, or a Boolean function F, from the instance space to the hypothesis space, to describe the concept, i.e., seek an F such that $F(v) = 1$ for all v, where F is in the hypothesis space. The statistical approach, however, tries to find a probabilistic mapping, or a probability distribution, between the two spaces, i.e., $prob(F(v) = 1) \geq 1 - \delta$, where δ is as close to 0 as possible but may be lower bounded by a fundamental parameter of the hypothesis space, such as the Bayes' misclassification rate [10]. This distinction, between approaches which implicitly assume that $\delta = 0$ and those that do not, is important since a variety of results obtained in the area of theoretical inductive learning (e.g., Gold [11], Valiant [12], Haussler *et al.* [13]) cannot be readily extended to the case where the Bayes' risk for the problem is nonzero.

Learning algorithms can be viewed as searching the hypothesis space in some manner. A "bottom-up" approach (e.g., symbolic learning) involves incremental generalization of specialized hypotheses, while a "top-down" approach (e.g., statistical algorithms) is based on the specialization of more general hypotheses, i.e., an initially simple and general model is refined and specialized to improve the goodness-of-fit. It is interesting to note that connectionist learning such as the backpropagation algorithm [14] is inherently "bottom-up" in this sense. The approach followed by our ITRULE algorithm will be "top-down." One might speculate as to the statistical robustness and convergence rates of the respective approaches, e.g., the bottom-up approach is less robust in the sense that it may be order sensitive. We will not pursue this topic further in this paper.

A good taxonomy of automatic induction algorithms is given in Cohen and Feigenbaum [15]. These algorithms can loosely be categorized into two main areas, those which use symbolic manipulation techniques and those which use statistically oriented techniques. Mitchell's "version spaces" algorithm [16] is perhaps the best known symbolic concept learning algorithm. Another example is the AQ11 algorithm of Michalski and Larson [17] which achieved success in the domain of plant disease diagnosis [18]. Typically, these algorithms examine the examples *sequentially* and refine what is known as the rule space until a set of general classification rules covering the examples are arrived at. However, noise is not easily handled by the symbolic approaches, since they

involve an implicit assumption that the Bayes error rate for the problem is zero, i.e., "perfect" classification of each attribute is possible in terms of the other attributes. In addition, the algorithms are computationally unattractive. Consequently, their use has been limited to research-oriented endeavors rather than practical applications such as knowledge acquisition for expert systems.

Methods which can be termed as statistical, exploit *average* properties of the example set. However, *existing* statistical learning algorithms generally lack the flexibility we require, by either imposing a particular parametric statistical model on the environment, or, as with tree-induction algorithms, imposing a particular *structure* on the nature of the solution. Algorithms such as ID3 (which derives classification decision trees from sample data [19], [20]) have been widely used for rule induction. However, such trees are essentially sequential decision algorithms which are quite different in nature to the data driven nature of expert systems. Rule bases are data driven in the sense that any set of input data can potentially be used to begin the inference. Trees must always begin with the attribute associated with the root node. In addition, rule bases can accomodate missing attribute information, whereas trees are not designed to do so. Trees can also be difficult to understand for the user [21], a problem which should not be underestimated in light of the overall advantages of explicit knowledge representation inherent to production rules. We were originally motivated to look at this problem of generalized rule induction as the limitations of tree structures became apparent in relation to expert systems. In short, rules provide a much more flexible representation than tree structures. This is not to say that trees are not useful in problem areas, such as classification where a predetermined "hard-wired" solution is sufficient [22], [23]. However, by their very definition, expert systems tend to be used for problems where variable inputs can be handled (missing, uncertain, or changing data), variable outputs (different goals) may be specified, and there is a need for an explicit representation of the system's knowledge for user interaction.

One of the few contributions to the problem of generalized rule induction is an approach based on fuzzy logic which was independently proposed recently by Gaines and Shaw [24]. They define the ENTAIL algorithm which derives rules from a reportory grid (Boose [25]). Their approach is interesting in that they transform the subjective reportory grid numbers (as input by a human expert) into fuzzy logic parameters which, in turn, are used to obtain a measure of the information content of the associated rules. The algorithm outputs the set of most informative rules. This approach is one of the few examples of automated knowledge acquisition tools currently avaliable. However, since it is designed to elicit *subjective* data rather than deal with random sample data, it is not directly applicable to our problem. In addition, our approach is differentiated by the underlying philosophy for dealing with uncertainty, namely standard probability theory rather than fuzzy logic.

Ganascia [26] has also proposed an algorithm for rule induction. His approach is more heuristic in nature than the algorithm to be presented here and is not based on any fundamental measure of rule "goodness." Quinlan described a scheme [27] whereby ID3-induced trees are transformed into production rules. In addition to the drawback that this particular scheme is classification based, we feel that tree transformation techniques in general may not be optimal for performing rule induction. Like Quinlan's approach, Cendrowska's PRISM algorithm [28] is classification based and has an information theoretic basis. The PRISM algorithm is intrinsically a symbolic learning technique since Cendrowska assumes that a "complete" training set will lead to the existence of a perfect classifier (zero error probability) for a given set of attributes. As mentioned earlier, one rarely encounters such a situation in practice, i.e., there is almost always a lower (nonzero) bound, the Bayes risk for uniform losses, on the minimum classification error achievable.

More recently Clark and Niblett have described CN2 [29], a rule induction algorithm which, like PRISM, searches for classification rules directly using a measure of rule goodness. While CN2 incorporates a larger hypothesis space than simple conjunctive terms (by allowing internal disjunction) it constrains its search through the allowable hypothesis space using the notion of a "beam size" which is an *ad hoc* technique to restrict the algorithms' potentially combinatorially large search for rules. We will avoid this problem in our specification of the ITRULE algorithm by using information theoretic bounds (based on existing rules found by the algorithm) to constrain its search through the hypothesis space without loss of optimality. CN2 also produces a set of rules in the form of a decision list [30]—since a decision list is a form of decision tree this form of derived rule representation suffers from the limitations mentioned earlier with respect to trees.

In fact, neither the CN2 and PRISM rule measures include an *a priori* probability term. Incorporation of *a priori* belief is a necessary component of any scheme which performs *generalized* rule induction since it allows one to compare not only competing hypotheses for the same concept, but also hypotheses for different concepts. From an information theoretic point of view, the rarer the occurence of an event, the more valuable is the information confirming its occurence. This ability to rank competing hypotheses for multiple concepts is fundamental for a learning agent in a resource constrained environment and is a central theme of our paper. The problem of generalized rule induction has not previously been addressed directly, although it is implicit in both the Bayesian approach of Cheesman [31] and the ENTAIL algorithm of Gaines and Shaw [24].

III. The Information Content of a Rule

We propose to use the following simple model of a rule, i.e.:

$$\text{If } \boldsymbol{Y} = y, \text{ then } \boldsymbol{X} = x \text{ with probability } p \qquad (1)$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are two attributes (dimensions in the instance space) with "x" and "y" being values in their respective discrete alphabets. For our purposes we may treat $\boldsymbol{X}$ and $\boldsymbol{Y}$ as discrete random variables. We restrict the right-hand expression to being a single value assignment expression while the left-hand side may be a conjunction of such expressions.

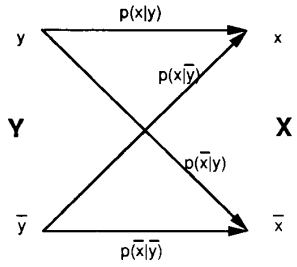

Fig. 1. Two variables connected by a discrete memoryless channel.

Intuitively we can view the two random variables as being connected by a discrete memoryless channel, as in Fig. 1. The channel transition probabilities are simply the conditional probabilities between the two variables.

A *rule* corresponds to a particular input event $\boldsymbol{y} = y$, rather than the average over all input events as is normally defined for communication channels, and p, the rule probability, is the transition probability $p(\boldsymbol{X} = x|\boldsymbol{Y} = y)$. Let us define $f(\boldsymbol{X}, \boldsymbol{Y} = y)$ as the *instantaneous* information that the event $\boldsymbol{Y} = y$ provides about $\boldsymbol{X}$, i.e., the information that we receive about $\boldsymbol{X}$ given that $\boldsymbol{Y} = y$ has occurred. The instantaneous information is the information content of the rule *given* that the left-hand side is true. A reasonable requirement to make (the interested reader can refer to Shannon's original paper [32] for a complete discussion), is that

$$E_y[f(\boldsymbol{X}:\boldsymbol{Y} = y] = I(\boldsymbol{X}:\boldsymbol{Y}) \tag{1}$$

where E_y denotes the expectation with respect to the random variable $\boldsymbol{Y}$. The equation requires that the average information from all rules should be consistent with the standard definition for average mutual information beteen two random variables. Blachman has shown [33] that $f(\boldsymbol{X}:\boldsymbol{Y})$ as defined above is not unique. In his paper he proposes 2 candidates which satisfy this equation. We shall refer to these 2 functions as the *i-measure*, $I(\boldsymbol{X}:\boldsymbol{Y})$, and the *j-measure*, where$j(\boldsymbol{X}:\boldsymbol{Y} = y)$,

$$\begin{aligned} i(\boldsymbol{X}:\boldsymbol{Y} = y) &= H(\boldsymbol{X}) - H(\boldsymbol{X}|\boldsymbol{Y} = y) \\ &= \sum_x p(x) \log\left(\frac{1}{p(x)}\right) \\ &\quad - \sum_x p(x|y) \log\left(\frac{1}{p(x|y)}\right) \end{aligned} \tag{2}$$

and

$$j(\boldsymbol{X}:\boldsymbol{Y} = y) = \sum_x p(x|y). \log\left(\frac{p(x|y)}{p(x)}\right). \tag{3}$$

These two measures have quite different interpretations. In words, the i-measure is the difference in the *a priori* and *a posteriori* entropies of $\boldsymbol{X}$, while the j-measure is the average mutual information between the *events* x_i and y with the expectation taken with respect to the *a posteriori* probability distribution of $\boldsymbol{X}$. The difference is subtle, yet significant enough that the j-measure is always non-negative, while the i-measure may be either negative or positive. In fact, Blachman has proven that the j-measure is unique as a non-negative information measure which satisfies (1), i.e., it is the *only* non-negative measure. We note in passing that CN2 minimizes $H(\boldsymbol{X}|\boldsymbol{Y} = y)$ (in (2)) in order to search for good rules—as mentioned earlier this ignores any *a priori* belief pertaining to $\boldsymbol{X}$, thus precluding the use of these algorithms for *generalized* rule induction. In addition, the term $H(\boldsymbol{X}|\boldsymbol{Y} = y)$ only measures the *a posteriori* entropy of $\boldsymbol{X}$—as we shall shortly see, this is not sufficient for defining a general rule goodness measure.

We have demonstrated elsewhere [9] that there is a fundamental problem with using measures which are negative. We see that $I(\boldsymbol{X}:\boldsymbol{Y} = y)$ can be equal to zero even if $p(x|y) \neq p(x)$, e.g., $p(x|y) = p(\bar{x})$, where $\boldsymbol{X}$ is a binary variable. In other words, the i-measure is zero if the transition probabilities in the channel, for a given input, form a permutation of the output probabilities. An appropriate title for this phenomenon is the *information paradox*, i.e., there is no change in the entropy but we *have* recieved information about $\boldsymbol{X}$. This is an example of a fundamental difference between using channel models for cognitive modeling, and using them for standard communication purposes. In the case of the latter, we do not distinguish between individual random events, except in terms of their attached probabilities of occurrence. The entropy of a discrete random variable is the same, independent of which probabilities are assigned to which events in the event space of the variable. Consider the case of an n-valued variable where a particular value of $\boldsymbol{X} = x_1$ is particularly likely *a priori* ($p(x_1) = 1 - \epsilon$), while all other values in $\boldsymbol{X}$'s alphabet are equally unlikely with probability $\epsilon/n - 1$. In this case a conditional permutation of these probabilities (the conditional $p(\boldsymbol{X}|y)$) would be significant, i.e., a rule which predicts the relatively rare event $\boldsymbol{X} = x_k$ for some k. However, the i-measure, because it cannot distinguish between particular events, would yield zero information for such events. Hence, we argue that the i-measure is inappropriate as a basic measure of rule information content.

Consider the alternative, the j-measure. It can be shown that the j-measure satisfies a variety of desirable mathematical properties which render it acceptable [34], including appropriate limiting properties. For example, as the transition probability approaches 1, the information content of the rule approaches the self-information of the right-hand event, $\log(1/p(x))$. For our purposes, i.e., with a rule rather than a channel, $j(\boldsymbol{X}:\boldsymbol{Y} = y)$ has the special form,

$$\begin{aligned} j(\boldsymbol{X}:\boldsymbol{Y} = y) =& p(x|y). \log\left(\frac{p(x|y}{p(x)}\right) \\ &+ (1 - p(x|y). \log\left(\frac{(1 - p(x|y))}{(1 - p(x))}\right) \end{aligned} \tag{4}$$

since a rule only gives us information about the event $\boldsymbol{X} = x$ and its complement $\bar{x}$. Because of this form we can plot some typical curves for $j(\boldsymbol{X}:\boldsymbol{Y} = y)$, as shown in Fig. 2. A further point worth making about the j-measure at this juncture is that it appears in the information theoretic literature under various guises. For instance, it can be viewed as a special case of the *cross-entropy* (Shore and Johnson [35]) or the *discrimination* (Kullback [36], Blahut [37]), a measure which defines the information theoretic similarity between two

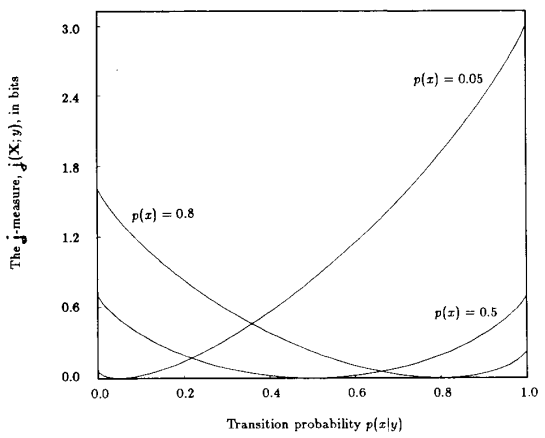

Fig. 2. Typical plots of the j-measure for various values of $p(x)$.

probability distributions. In this sense the j-measure is a well-defined measure of how dissimilar our *a priori* and *a posteriori* beliefs are about $\boldsymbol{X}$—useful rules imply a high degree of dissimilarity.

From our original definition in (1), the *average* information content of a rule can be defined as

$$J(\boldsymbol{X}:\boldsymbol{Y}=y) \;=\; p(y)\;.\; j(\boldsymbol{X}:\boldsymbol{Y}=y). \tag{5}$$

Note that this measure is an average in the sense there is an implicit assumption that the instantaneous information from the other "Y-terms" is zero. This is consistent with the cognitive science approach to production rules where essentially we can only draw inferences based on the occurence of a particular event but not its complement. More generally, in the context of learning in a resource constrained environment, each rule must be significant in its own right. In particular, rules which have left-hand sides that are the complements of existing rules must be evaluated separately. In the next section we will demonstrate the appropriateness of the previous definition for average information content. In an intuitive sense the average measure relates to the average value of the rule information content (useful for learning), while the instantaneous measure can be used to rank rules after the event $\boldsymbol{Y} = y$ has occured (useful for forward chaining in rule-based inference).

We shall see later that bounding the information content of a rule can help considerably when we are trying to learn rules from data. At this point it is sufficient to point out that the J-measure must obey the following basic inequality [34]:

$$J(\boldsymbol{X};\boldsymbol{Y}=y) \;\leq\; p(y)\left(\min\{\max_i\{\log(\frac{1}{p(x_i)})\},\log(\frac{1}{p(y)})\}\right). \tag{6}$$

In particular we note that

$$J(\boldsymbol{X}:\boldsymbol{Y}=y) \;\leq\; p(y)\log(\frac{1}{p(y)}) \tag{7}$$

$$\leq 0.53 \text{ bits}. \tag{8}$$

We will later investigate more detailed bounds on the measure for use in the ITRULE algorithm.

IV. Properties of the J-Measure as a Hypothesis Preference Criterion

The next step is to understand the nature of the J-measure as a basic preference measure among competing hypotheses, i.e., rather than considering its mathematical properties, we will consider its interpretation in terms of classical induction theory. Consider the problem of finding a hypothesis to fit some given data, i.e., a general learning problem. There appears to be a general consensus that the two primary criteria for evaluating a hypothesis are the *simplicity* of the hypothesis and the *goodness-of-fit* between the hypothesis and the data (Angluin and Smith [38], Gaines [39], and Michalski [40]). The problem is to combine these two criteria into a single measure such that the hypotheses can be ordered. In terms of the probabilistic rules defined earlier, let us interpret the event $\boldsymbol{X} = x$ as the concept to be learned and the event (possibly conjunctive) $\boldsymbol{Y} = y$ as the hypothesis describing this concept.

The J-measure is the product of two terms. The first, $p(\boldsymbol{Y} = y)$, is the probability that the hypothesis will occur and, as such, can be interpreted as a measure of hypothesis *simplicity*. Symbolic algorithms use more *ad hoc* techniques to determine the simplicity of a hypothesis, such as enumerating the number of basic propositions which make up a conjunctive hypothesis (Angluin and Smith [38]). Such techniques may work in given domains but lack generality. In contrast, the probabilistic criterion for simplicity is perfectly general.

The second term making up $J(\mathbf{X}:y)$ is $j(\boldsymbol{X}:y)$. As we have seen in the last section, $j(\boldsymbol{X}:y)$ can be interpreted as the cross entropy of $\boldsymbol{X}$ with the variable "$\boldsymbol{X}$ conditioned on the event $\boldsymbol{Y} = y$". Cross entropy is well known as a goodness of fit measure between two distributions (Shore and Johnson [35]). It can be interpreted as a distance measure where "distance" corresponds to the amount of information required to specify a random variable. It is frequently used to find the conditional distribution which most closely agrees with the original distribution. For our purposes the goodness-of-fit should be maximized when the transition probability equals 1 (or 0), and it should be minimized when the transition probability equals the *a priori* probability $p(x)$. Clearly $j(\boldsymbol{X}:\boldsymbol{Y} = y)$ is a monotonic distance measure in this sense as can be seen from Fig. 2. Consequently, the product term, $J(\boldsymbol{X}:\boldsymbol{Y} = y) \;=\; p(y)j(\boldsymbol{X}:\boldsymbol{Y} = y)$, possesses a direct interpretation as a multiplicative measure of the simplicity and goodness-of-fit of a given rule.

As an example of this trade-off consider the following hypothetical reptile domain which is described in terms of 3 binary attributes, namely, *legs* (true or false), *habitat* (desert or not), and *snake* (true or false). A joint distribution over these attributes is specified in Table I. Let us say that we are interested in rules which confirm the attribute-value pair *snake* = *true* as a right hand side. The rule

If *habitat* = *desert* then *snake* = *true*
with probability = 0.625,
$j = 0.225$. $J = 0.09$

is a reasonable rule. The *a priori* probability of a reptile being a snake is 0.35, while the *a posteriori* probability is 0.625,

TABLE I

desert	legs	snake	joint probability
0	0	0	0.0
0	0	1	0.1
0	1	0	0.5
0	1	1	0.0
1	0	0	0.0
1	0	1	0.25
1	1	0	0.15
1	1	1	0.0

given that the rule fires, an event which has a probability of 0.4. The J-measure for this rule is calculated to be 0.09 bits of information, i.e., this is the information we will acquire on average by using this rule. If we *specialize* this rule by adding another term to the left-hand side, we would obtain

If *habitat* = *desert* and *legs=false* then
snake = *true* with probability = 1.0.
$j = 1.51, \quad J = 0.379$

which has a much greater information content of 0.379 bits. The decrease in simplicity (by a factor of 0.25/0.4) is more than offset by the approximately 8-fold increase in goodness-of-fit, as measured in bits. If we now *generalize* once more to

If *legs=false* then *snake* = *true* with probability = 1.0.
$j = 1.51, \quad J = 0.53$

we obtain a rule with 0.53 bits of information, which is, in fact, the best rule. The key point to note is the advantage of using a *quantitative* rule preference measure to easily compare the more general and specialized versions of the same basic rule.

It is worth pointing out in passing that cognitive scientists consider generalization and specialization to be two of the most basic techniques used by the brain to generate new rules [41, pp. 84–88]—while we are not interested in cognitive modeling *per se* it is interesting to note that our measure supports these rule generation principles in a robust and quantitative manner.

V. The Basis for ITRULE: Specializing Rules to Increase the J-Measure

Before describing the ITRULE algorithm we must first develop some quantitative bounds on the nature of specialization. The basic premise of the algorithm will revolve around instance-based generalization from examples to generate an initial set of rules, followed by specialization of these rules to optimize the rule set. The exact nature of the specialization is critical to the performance of the algorithm.

Specialization is the process by which we hope to increase a rule's "goodness" by adding an extra condition to, or specializing, the rule's left-hand side. The consequent necessary decrease in simplicity of the hypothesis should be offset by an increase in the goodness-of-fit to the extent that the overall goodness measure is increased. We will examine specialization, using the J-measure as our definition of rule goodness, with $p(y)$ corresponding to simplicity and $j(\boldsymbol{X};\boldsymbol{Y}=y)$ corresponding to goodness-of-fit.

The question we pose is as follows: given a particular general rule, what quantitative statements can we make about specializing this rule? In particular, if we define J_s and J_g as the J-measures of the specialized and general rules, respectively, can we find a bound of the form

$$J_s \leq f(J_g) \tag{9}$$

for some $f(.)$ defined on J_g or its component terms? The motivation for bounding J_s in this manner is two-fold. Firstly, it produces some theoretical insight into specialization, while secondly, the bound can be used by rule induction algorithms to search the rule space (hypothesis space) efficiently. This section will be devoted to stating and analyzing a very useful bound of this form.

Consider that we are given a general rule whose J-measure, J_g, is defined as

$$J_g = J(\boldsymbol{X};\boldsymbol{Y}=y) \tag{10}$$

$$= p(y)\left(p_g.\log\frac{p_g}{p_x} + (1-p_g).\log\left(\frac{1-p_g}{1-p_x}\right)\right) \tag{11}$$

$$= p(y)j(\boldsymbol{X};\boldsymbol{Y}=y) \tag{12}$$

where $p_g = p(x|y)$ and $p_x = p(x)$. The probability p_g is the transition probability of the general rule. We wish to bound

$$J_s = J(\boldsymbol{X};\boldsymbol{Y}=y,\boldsymbol{Z}=z) \tag{13}$$

$$= p(y.z)\left(p_s.\log\frac{p_s}{p_x} + (1-p_s).\log\left(\frac{1-p_s}{1-p_x}\right)\right) \tag{14}$$

$$= p(z|y)\ p(y)\ j_s \tag{15}$$

where j_s is the specialized j-measure, and $p_s = p(x|y.z)$ which is the transition probability of the specialized rule. Without loss of generality we assume that $p_g > p_x$, since if $p_g < p_x$ we can simply reverse the labeling on x and $\bar{x}$, while if $p_g = p_x$ then $J_g = 0$ and the case is not of interest since any condition $z \neq y$ will lead to J_s being greater than J_g. Given no information about $\boldsymbol{Z}$ whatsoever, we can state the following result.

Theorem:

$$J_s \leq \max\left\{p(y)p_g\log\frac{1}{p_x},\ p(y)(1-p_g)\log\frac{1}{1-p_x}\right\} \tag{16}$$

$$= \max\left\{p(x,y)\log\frac{1}{p_x},\ p(\bar{x},y)\log\frac{1}{p_{\bar{x}}}\right\}. \tag{17}$$

The proof is given in the Appendix. If we recall the original bound we stated in (6), and we make the assumption that $p(y) \leq p(x)$ and $p(y) \leq p(\bar{x})$, then the equivalent original bound can be stated as

$$J_s \leq p(y).\max\{\log(\frac{1}{p(x)}),\ \log(\frac{1}{p(\bar{x})})\}. \tag{18}$$

Comparing the two inequalities we see that the new result gives an improvement of a factor of $p(x|y)$ (or p_g). It is interesting to note that the transition probability of the *general* rule plays such a limiting multiplicative role in the bound. In

essence, it tells us the limits imposed by the continued presence of the y term in any more specialized rule.

Consider the reptile domain rules discussed in the previous section. Had we applied the above bound to the general rule with *habitat=desert* as its left-hand side we could have determined that the most information we could get from specializing that rule further would be 0.3768 bits. In fact, it turned out that the specialized rule we considered achieved this bound as does the third example rule, with *snake=true* as its left-hand side. Both cannot be improved upon since the transition probabilities are 1.

For the case when $p(y) > p(x)$, or $p(y) > p(\bar{x})$, we note that this introduces an extra constraint into the problem by effectively limiting the achievable value of p_g, and consequently p_s. Clearly, the bounds still hold but are no longer achievable. Equivalent achievable bounds can be derived, but are omitted in this paper, since such pathological cases are not of general interest.

As a consequence of this theorem we note that since the bound is achievable, then without further information about $\boldsymbol{Z}$, it cannot be improved upon. In fact, if we set $y = z$ then we find that J_g itself also obeys this bound. The logical consequence of this statement is that it precludes using the bound to discontinue specializing based on the value of J_g alone, since unless $p_g = 0$ or $p_g = 1$, the result holds as a strict inequality for J_g. Conversely, if p_g is not equal to either 1 or 0, then with no information at all available about the other variables, there may always exist a more specialized rule whose information content is strictly greater than that of the the general rule. However, as we shall see, we could certainly compare the *bound* with any rules we might already have. In particular, if the bound is less (in bits) than the information content of the worst rule, then specialization cannot possibly find any better rule. This principle will be the basis for restricting the search space of the ITRULE algorithm.

VI. The ITRULE Algorithm

We will now define the ITRULE algorithm and discuss the basic ideas which motivated this particular definition. The ITRULE algorithm takes sample data in the form of discrete attribute vectors and generates a set of K rules, where K is a user-defined parameter. The set of generated rules are the K most informative rules from the data as defined by the J-measure. In this sense the algorithm can be described as optimal. The probabilities required for calculating the J-measures are estimated directly from the data using standard statistical point estimation techniques [42].

The algorithm proceeds by first finding K rules, calculating their J-measures, and then placing these K rules in an ordered list. The smallest J-measure, that of the Kth element of the list, is then defined as the running minimum J_{min}. From that point onwards, new rules which are candidates for inclusion in the rule set have their J-measure compared with J_{min}. If greater than J_{min} they are inserted in the list, the Kth rule is deleted, and J_{min} is updated with the value of the J-measure of whatever rule is now Kth on the list. The critical part of the algorithm is the specialization criterion since it determines how much of the exponentially large hypothesis space actually needs to be explored by the algorithm.

For each of the $n.m$ possible right-hand sides, the algorithm employs *depth-first* search over possible left-hand sides, starting with the first-order conditions and specializing from there. Specialization ceases on a general rule if the bound above is *less* than J_{min}. In addition, if the transition probability of a given general rule is equal to 1 or 0, then as we have seen earlier, we can also cease specializing. The algorithm systematically tries to specialize all $nm.(n-1).2m$ first-order rules and terminates when it has determined that no more first-order rules exist which can be specialized to achieve a higher J-measure than J_{min}.

The general situation occurs when we have a right-hand side $\boldsymbol{X} = x$ and a left-hand side $y_1, \ldots, y_k$, where we have just evaluated J_g and inserted the rule in the list if $J_g > J_{min}$. In practical terms, in order to calculate J_g, we have sorted the original data into a subtable conditioned on $y_1, \ldots, y_k$. We now wish to decide (using the bounds) whether further specialization, and consequent sorting, is worthwhile. The decision whether to continue specializing or to back-up on the depth-first search is determined by the following sequence:

i) if $p_g = 1$ or $p_g = 0$ then back-up the search, else;

ii) if $J(\boldsymbol{X}; y_1, \ldots, y_k) \leq J_{min}$ then check if for *any* z we can hope to find $J_s > J_{min}$, i.e., calculate

$$J_s = \max\left\{ p(y)p_g \log \frac{1}{p_x}, p(y)(1-p_g) \log \frac{1}{1-p_x} \right\}$$

and (by Theorem 1) if $J_s \leq J_{min}$, then back-up the search;

iii) else continue to specialize.

The general description of ITRULE given earlier is not intended as a definitive statement of how the algorithm should be implemented. Particular implementations may depend heavily on the nature of the particular problem. For instance, in data analysis we may only want to look at rules which conclude certain propositions of interest. The algorithm simply restricts the right-hand side propositions to that subset of interest (the limiting form of this approach is, of course, a classifier where we are only interested in propositions in the event space of a single variable, the "class").

VII. On the Complexity of ITRULE

With n m-ary attributes the number of possible rules in the data is R where,

$$R = nm\Big((2m+1)^{n-1} - 1\Big) \tag{19}$$

since for each of the nm possible right-hand sides, the other $n-1$ attributes have $2m+1$ possible states, namely, a truth statement and its negation for each of the m propositions and a "don't care" state for the attribute as a whole (for the case of *binary* attributes $m = 1$ because the negation of a proposition is also a basic proposition). As an example, if we have 10 binary attributes, there are $N = 10.3^9 - 10 = 196820$ possible rules. From a practical point of view, of course, we are likely to have neither the data to support so many inductive assertions nor the computational resources to manage them. Hence, in

order to define a tractable algorithm we will need to "prune" the set of possible rule candidates considerably. Let us define a kth-order rule as a rule with k basic propositions in its left-hand side. Let r_k be the number of possible kth-order rules, so that we have

$$r_k = (2m)^k.\binom{n-1}{k}.nm, \qquad 1 \leq k \leq n-1 \tag{20}$$

since there are $\binom{n-1}{k}$ sets of propositions of size k and $(2m)^k$ rules for each set. (That $\sum_k r_k = R$ holds can be verified by using binomial identities such as those given in in Feller [43, p. 63]). The ratio

$$\alpha = \frac{r_k}{r_{k-1}} = \frac{2m.(n-k)}{k} \tag{21}$$

gives the ratio of the number of rules of order k to the number of order $k-1$. When α becomes less than 1 we have the condition that the number of rules is falling off rather than increasing, i.e., when

$$k > \left(\frac{2m}{2m+1}\right) n \ . \tag{22}$$

The significance of this result is that for all practical purposes, as we increase the order of the rules from $k = 1$ upwards, the size of the search space *increases*, and for k, which is relatively small compared to n, it increases geometrically (by (21)). We can write R as

$$R = nm \sum_{k=1}^{n-1} \alpha_k.r_{k-1} \tag{23}$$

$$= nm \sum_{k=1}^{n-1} \left(\prod_{i=1}^{k} \alpha_i\right), \tag{24}$$

where

$$\alpha_i = 2m.\left(\frac{n-i}{i}\right) \quad \text{and} \quad r_0 = 1 \ . \tag{25}$$

If we imagine implementing an algorithm which begins with first-order rules and specializes to higher orders (in order to find rules with higher J-measures) then an algorithm using blind search would have complexity $O(R) = n(2m)^n$, as defined previously. On the other hand, an algorithm which "prunes" the search space will have complexity

$$O(R') = nm \sum_{k=1}^{n-1} \left(\prod_{i=1}^{k} \beta_i\right) \tag{26}$$

where the $\beta_k < \alpha_k$. A tractable algorithm will have $\beta_k < 1$ for (at least) k greater than some small fraction of n.

The complexity of the ITRULE algorithm cannot be determined exactly since it is highly dependent on the nature of the input data (in referring to "the algorithm," we mean the general version, where, the exact nature of the specialization may vary). Probabilistic analysis, based on average performance over all possible input data sets, is too difficult to carry out directly without invoking unrealistic assumptions concerning the nature of the inputs. The best we can do is to invoke the argument that as specialization (or rule order) increases, the simplicity of the hypotheses decreases to the extent that their probability of occurrence is very small. Hence, our bounds should eliminate the majority of the higher order rules from consideration. In effect, the β_k should become negligible as k increases. We will see later how β behaves for real data sets. A worst-case upper bound occurs for the pathological case of a set of N binary attributes whose Nth-order joint distribution is entirely uniform, i.e., all transition probabilities are equal to 0.5. In this case *all* rules yield zero information, and hence, J_{min} would always be zero. However, the bounds would be nonzero in general, in which case the algorithm would specialize to all possible $R = nm((2m+1)^{n-1} - 1)$ rules. Let us note in conclusion that the lack of quantitative results on the complexity of the ITRULE algorithm reflects the well-known inherent difficulty in quantifying the complexity of "open-ended" induction problems.

The choice of K, the number of rules which the algorithm keeps in the list, obviously affects the computational complexity, since the value of the J-measure of the Kth rule has a considerable impact on the effectiveness of the bounding. For example, K may be chosen so large that J_{min} is zero or near zero at all times. However, there is normally no reason to choose such large K. If we are just interested in data analysis, then very often some value of K between say 20 and 100 is sufficient for our purposes. However, if we wish to use the rules for probabilistic *inference* then we generally require more rules. In particular for each proposition in the system, we would like to have at least r rules with that proposition in their conclusions, or in terms of a graph where each proposition is a node, r is the number of rules entering a node or the "fan in" of the node. In order for the system to perform useful inference (for example, multiple pieces of evidence supporting the same hypothesis) we require that r be some integer greater than 1. Yet r should not be too large in order that the inference itself is computationally feasible. Hence, we can say that for inference purposes, $O(K) = nm$.

VIII. Experimental Results on Three Data Sets

We consider the results of applying the ITRULE algorithm to three "real-world" data sets—the first, a financial domain, in some detail, followed by a brief overview of the results obtained on congressional voting records and chess end-games. The first data set comes from published financial information on no-load mutual funds [44]. Fig. 3 shows a set of typical sample data. Each line is an instance of a fund (with name omitted), and each column represents an attribute of the fund. A typical categorical attribute is "fund type" which reflects the investment objectives of the fund (growth, growth and income, and agressive growth). Among the noncategorical attributes are "five year return on investment" expressed as a percentage, "yield" (the dividend payments as a proportion of net asset value), "turnover rate" (a measure of the trading activity of the fund), and "expense ratio" (the amount of administrative fees).

Real-valued attributes (or indeed attributes whose alphabet size is large, but finite) are quantized *a priori*. While this is not necessarily an optimal procedure (quantization based

Mutual Funds Database (American Association of Investors, 1988)

FundType	5yrRtrn	Divrsty	Beta	Stocks	Yield	XpnsRat%	Turnover	Assets	CapGain
Growth	belowS&P	low	under1	over75%	under3%	low	low	large	10to20%
Growth	belowS&P	low	over1	over75%	under3%	low	high	small	under10%
Gth&Inc	belowS&P	high	under1	over75%	under3%	low	high	small	10to20%
AggrGth	belowS&P	high	over1	over75%	under3%	low	high	small	under10%
Gth&Inc	aboveS&P	low	under1	under75%	over3%	low	low	large	over20%
Gth&Inc	belowS&P	high	under1	over75%	over3%	low	low	small	under10%
Growth	belowS&P	high	under1	over75%	under3%	high	low	small	10to20%
Gth&Inc	aboveS&P	low	under1	under75%	over3%	low	high	large	over20%
AggrGth	belowS&P	high	over1	over75%	under3%	low	high	small	10to20%
Growth	belowS&P	high	under1	over75%	over3%	low	low	large	over20%
Growth	belowS&P	low	under1	over75%	under3%	high	low	small	over20%
Growth	aboveS&P	high	under1	over75%	under3%	low	low	large	10to20%
AggrGth	belowS&P	high	over1	over75%	under3%	high	low	small	under10%
Gth&Inc	belowS&P	low	under1	under75%	over3%	high	low	small	10to20%
Growth	aboveS&P	low	over1	over75%	under3%	low	low	large	under10%
Growth	belowS&P	high	over1	over75%	under3%	low	high	large	under10%
Growth	aboveS&P	low	under1	over75%	under3%	low	low	small	10to20%
Growth	belowS&P	low	under1	under75%	under3%	high	high	small	under10%
Growth	belowS&P	low	over1	over75%	under3%	low	low	small	over20%
Gth&Inc	belowS&P	low	under1	over75%	over3%	high	low	small	under10%
Gth&Inc	aboveS&P	high	under1	under75%	over3%	low	low	small	10to20%
Gth&Inc	aboveS&P	low	over1	over75%	under3%	low	low	small	10to20%
Gth&Inc	belowS&P	low	under1	under75%	over3%	low	low	large	10to20%
Growth	belowS&P	low	under1	over75%	under3%	low	low	large	over20%
Gth&Inc	belowS&P	low	under1	under75%	under3%	low	low	large	10to20%
Gth&Inc	belowS&P	low	under1	over75%	over3%	low	low	large	10to20%
AggrGth	belowS&P	high	under1	over75%	under3%	low	low	large	over20%
Gth&Inc	aboveS&P	low	under1	under75%	over3%	low	low	large	over20%
AggrGth	aboveS&P	low	under1	over75%	under3%	low	low	small	over20%
Growth	belowS&P	high	over1	over75%	under3%	low	low	small	under10%
Growth	belowS&P	high	over1	over75%	under3%	low	high	small	10to20%
Gth&Inc	aboveS&P	high	under1	under75%	over3%	low	high	large	10to20%

Fig. 3. Subsample of mutual funds data set.

on *conditional* distributions may be much more predictive), we will not dwell on this topic here since the purpose of the paper is to focus on the ITRULE algorithm which is primarily intended to deal with categorical data. Quantization techniques, based both on domain knowledge and information theoretic criteria, are easy to derive. An example of using domain knowledge for quantization occurs with the attribute "Beta" or risk (volatility relative to the market), which has a natural cut-point of 1 since the market Beta is always defined to be 1. Domain knowledge also indicates that funds with expense ratios above 1.5% are high, and should be viewed critically. In the abscence of domain knowledge we use statistical and maximum entropy techniques for clustering the data into statistically significant categories. For example, the automatic technique splits the Stocks attribute (the percentage of fund assets in common stocks) at 75%. A domain expert may accept this advice or modify the value to make the categorization more meaningful.

Fig. 4 shows the results of asking ITRULE for the 10 best rules, where we restricted the maximum rule order to 2 for the purposes of making the output easier to interpret. A point to note, in the figures of rule sets to follow, is that we have implemented a "subsumption" function on the displayed rule output, i.e., we remove any rules for which there is a more *general* rule ranked higher on the list. The more general rule is considered to subsume the more specialized rule. The columns are relatively self-explanatory, and the probabilities correspond to sample estimates from the data as mentioned earlier. However, there is a potentially confusing notation used with respect to the labeling of the event x—x in general is a label for the rule right-hand side or its negation, *except* for the first column "$p(x|y)$", which is always written as the *greater* of the two transition probabilities, i.e., it is really $\max\{p(x|y).p(\bar{x}|y)\}$. The final two columns "y" and "xy" are the actual of number of occurences of the events y and xy, respectively.

From the figure we note that obvious rules emerge, confirming that the algorithm is on the right track. For example, among the most informative rules are rules relating fund type of "Growth and Income" to high yield funds (rules 3 and 4). This is obvious because income funds aim to do just that—pay dividends; they thus usually have nonzero yield. This ability to spot obvious rules is a powerful feature of the algorithm. It is usually the obvious domain rules that pose the biggest problem early in the knowledge acquisition process. The expert has difficulty in going back to basics, and explicitly identifying the vast number of fundamental rules applicable to the particular domain. Also, by looking at the trade-off between the instantaneous information or goodness-of-fit $j(\boldsymbol{X}:\boldsymbol{Y}=y)$ and the simplicity of a rule $p(y)$, we see that rule 4 is ranked lower than a rule which has much less instantaneous information (rule 3), but which fires more often.

Fig. 5 shows the 10 best rules (still limited to second order) obtained when we run ITRULE as a classifier, i.e., restricting the right-hand side to a single variable of interest, namely, "5 year return" which is either above or below the Standard and Poor index over the same 5 years. This is obviously a variable of considerable interest to prospective investors. However, we see that while rule 5 gives a reasonably accurate condition for determining *below* average funds, there is no single rule for

Automated Rule Induction - ITRULE

Data Set : Mutual Funds Data set (American Association of Investors, 1988)

No. Attributes = 10
No. Examples = 88
Maximum Rule Order = 2

			p(x|y)	p(y)	p(x)	j(X;y)	J(X;y)	y	xy
1	if[FundType = Gth&Inc Beta = under1]	then [Yield = over3%]	0.803	0.344	0.356	0.608	0.210	30	25
2	if[FundType = Gth&Inc Stocks = under75%]	then [Yield = over3%]	0.891	0.222	0.356	0.901	0.200	19	18
3	if[Beta = under1 Yield = over3%]	then [FundType = Gth&Inc]	0.804	0.344	0.367	0.580	0.200	30	25
4	if[Stocks = under75% Yield = over3%]	then [FundType = Gth&Inc]	0.892	0.222	0.367	0.869	0.193	19	18
5	if[FundType = not Growth Stocks = under75%]	then [Yield = over3%]	0.857	0.244	0.356	0.777	0.190	21	19
6	if[Yield = over3%]	then [FundType = Gth&Inc]	0.780	0.356	0.367	0.513	0.183	31	25
7	if[FundType = Gth&Inc]	then [Yield = over3%]	0.756	0.367	0.356	0.481	0.177	32	25
8	if[5yrReturn = aboveS&P Yield = over3%]	then [Stocks = under75%]	0.847	0.167	0.278	1.021	0.170	14	13
9	if[CapGain = not under 10% Yield = over3%]	then [Stocks = under75%]	0.698	0.300	0.278	0.549	0.165	26	19
10	if[Yield = over3% FundType = not Growth]	then [Stocks = under75%]	0.698	0.300	0.278	0.549	0.165	26	19

Fig. 4. The 10 best rules from the mutual funds data set (up to order 2).

Automated Rule Induction - ITRULE

Data Set : Mutual Funds Data set (American Association of Investors, 1988)

No. Attributes = 10
No. Examples = 88
Maximum Rule Order = 2

			p(x|y)	p(y)	p(x)	j(X;y)	J(X;y)	y	xy
1	if[Stocks = under75% XpnsRat% = low]	then [5yrRtrn = aboveS&P]	0.744	0.222	0.311	0.570	0.127	19	4
2	if[Stocks = under75% Assets = large]	then [5yrRtrn = aboveS&P]	0.757	0.189	0.311	0.605	0.114	16	3
3	if[FundType = Gth&Inc Assets = large]	then [5yrRtrn = aboveS&P]	0.710	0.233	0.311	0.483	0.113	20	5
4	if[Stocks = under75% CapGain = not under10%]	then [5yrRtrn = aboveS&P]	0.665	0.267	0.311	0.380	0.101	23	7
5	if[Stocks = over75% Assets = small]	then [5yrRtrn = belowS&P]	0.914	0.456	0.689	0.213	0.097	40	37
6	if[Stocks = under75%]	then [5yrRtrn = aboveS&P]	0.639	0.278	0.311	0.328	0.091	24	8
7	if[FundType = not Growth Assets = large]	then [5yrRtrn = aboveS&P]	0.651	0.256	0.311	0.351	0.090	22	7
8	if[FundType = Gth&Inc CapGain = not under10%]	then [5yrRtrn = aboveS&P]	0.608	0.311	0.311	0.268	0.083	27	10
9	if[FundType = Gth&Inc XpnsRat% = low]	then [5yrRtrn = aboveS&P]	0.587	0.322	0.311	0.233	0.075	28	11
10	if[FundType = not AggrGth Assets = large]	then [5yrRtrn = aboveS&P]	0.541	0.433	0.311	0.162	0.070	38	17

Fig. 5. The 10 best rules for classifying "5-year return" (up to order 2).

predicting *above* average funds with an accuracy greater than about 75%.

For completeness, we show in Fig. 6 the 15 best rules obtained when ITRULE is allowed to search up to order 5. Note that more general rules tend to dominate the list, as one might expect given the small sample size. However there are some third-order rules present, characterized in general by relatively high transition probabilities.

It would be naive to assume that the rules derived by ITRULE are necessarily an accurate reflection of the domain. For example, in this data set, there may be temporal variations masked out by the 5-year averaging on some attributes. Nonetheless the algorithm gives an immediate feel for the data and is particularly useful as an exploratory data analysis tool—essentially the produced rules are as good as the data is. The algorithm may be particularly effective when used in an iterative manner in conjunction with a domain expert—a given set of rules may suggest the inclusion of new attributes and the exclusion of others.

The astute reader will also have noted that ITRULE produces the *set of best rules* rather than the *best set of rules*, i.e., no attempt is made to evaluate the collective properties

AUTOMATED RULE INDUCTION - ITRULE

DATA SET : MUTUAL FUNDS DATA SET (AMERICAN ASSOCIATION OF INVESTORS, 1988)

NO. ATTRIBUTES = 10
NO. EXAMPLES = 88
MAXIMUM RULE ORDER = 5

													p(x|y)	p(y)	p(x)	j(X;y)	J(X;y)	y	xy
1	FUNDTYPE	NOT GROWTH	YIELD	OVER3%	CAPGAIN	NOT UNDER10%	THEN	STOCKS	UNDER75%				0.815	0.256	0.278	0.901	0.230	22	19
2	FUNDTYPE	GTH&INCOME	YIELD	OVER3%	CAPGAIN	NOT UNDER10%	THEN	STOCKS	UNDER75%				0.807	0.244	0.278	0.873	0.214	21	18
3	FUNDTYPE	GTH&INCOME	BETA	UNDER1			THEN	YIELD	OVER3%				0.803	0.344	0.356	0.608	0.210	30	25
4	FUNDTYPE	GTH&INCOME	STOCKS	UNDER75%			THEN	YIELD	OVER3%				0.891	0.222	0.356	0.901	0.200	19	18
5	BETA	UNDER1	YIELD	OVER3%			THEN	FUNDTYPE	GTH&INCOME				0.804	0.344	0.367	0.580	0.200	30	25
6	STOCKS	UNDER75%	YIELD	OVER3%			THEN	FUNDTYPE	GTH&INCOME				0.930	0.222	0.367	1.025	0.194	19	18
7	STOCKS	UNDER75%	YIELD	OVER3%	XPNSRAT%	LOW	THEN	FUNDTYPE	GTH&INCOME				0.892	0.222	0.367	0.869	0.193	19	18
8	FUNDTYPE	NOT GROWTH	STOCKS	UNDER75%			THEN	YIELD	OVER3%				0.857	0.244	0.356	0.777	0.190	21	19
9	YIELD	OVER3%					THEN	FUNDTYPE	GTH&INCOME				0.780	0.356	0.367	0.513	0.183	31	25
10	FUNDTYPE	GTH&INCOME	BETA	UNDER1	CAPGAIN	NOT UNDER10%	THEN	STOCKS	UNDER75%				0.724	0.289	0.278	0.618	0.179	25	19
11	BETA	UNDER1	YIELD	OVER3%	CAPGAIN	NOT UNDER10%	THEN	STOCKS	UNDER75%				0.724	0.289	0.278	0.618	0.179	25	19
12	FUNDTYPE	GTH&INCOME					THEN	YIELD	OVER3%				0.756	0.367	0.356	0.481	0.177	32	25
13	5YRRETURN	ABOVES&P	YIELD	OVER3%			THEN	STOCKS	UNDER75%				0.847	0.167	0.278	1.021	0.170	14	13
14	FUNDTYPE	NOT GROWTH	YIELD	OVER3%			THEN	STOCKS	UNDER75%				0.698	0.300	0.278	0.549	0.165	26	19
15	YIELD	OVER3%	CAPGAIN	NOT UNDER10%			THEN	STOCKS	UNDER75%				0.698	0.300	0.278	0.549	0.165	26	19

Fig. 6. The 15 best rules from the mutual funds data set (up to order 5).

of the rules. It may be conjectured that this problem is computationally intractable to solve optimally for arbitrary K (assuming that somehow we can quantify the "goodness" of a rule set). Current research is focused on effective heuristics for generating *pruned* rule sets where, for example, accuracy can be traded off with generality and redundancy. We make a point of *not* describing such extensions to the algorithm in this paper since the purpose here is to focus on the basic algorithm.

Results obtained on two other data sets are summarized in Figs. 7 and 8. For each data set we show the 10 most informative rules up to and including second order. We again purposely restricted the rules to low orders in order to make the output easier to interpret. The "voting" data set (as previously reported by the machine learning community [45], [46]) consists of voting records in a 1984 session of Congress, each piece of data corresponding to a particular politician. The obvious class variable is party affiliation or "politics" (republican or democrat), the other 16 attributes being yes/no votes on particular motions such as Contra-aid and budget cuts. The derived rules highlight the political topics which tend to segegrate politicians best—not surprisingly, there are strong correlations between foriegn policy, defense issues, and social programs, issues which traditionally separate the two parties. Given the probable imposition of party "whips" on many of these issues (i.e., all party members are instructed to vote in a certain manner) we did not expect any significant surprises from this data set. The primary intent was to verify that the algorithm would indeed find the expected relationships.

The second data set is taken from a chess end-game problem described in Quinlan's 1979 paper [47, pp. 177–180]. There are 7 attributes which characterize particular end-game configurations. With the 4 pieces (black knight and king, white rook, and king) there are 647 legal configurations. These 647 examples completely describe this domain. The object of the exercise is to classify whether the end-game is *lost* two-ply in a black-to-move situation—details are given in Quinlan's paper. This rule set is interesting in that, as shown in Fig. 8, ITRULE generates probabilistic rules (namely, the first three) as well as "factual" rules (rules 4–10). Since this domain is deterministic, i.e., perfectly classifiable given the attributes, both PRISM and ID3 tend to produce only perfect rules, i.e., rules with an effective transition probability of 1 or 0 (as reported by Cendrowska [29]). While ITRULE will find these rules, it also generates probabilistic rules or domain heuristics. For example, rules 1 and 2 tell us that if the black knight, king, and white rook are in line and if the rook bears on either the black king or knight, then there is roughly an 80% chance that the game is *safe*. More significantly, the probability that the game is *lost* has risen from an *a priori* value of 0.054 to an *a posteriori* probability of 0.21. In a statistical decision sense this change in probability could be very significant if the risk (associated with losing) significantly outweighs the benefit associated with a safe position. The rules shown in Fig. 8 were produced by the general "attribute–attribute" version of the algorithm rather than running it as a classifier. Hence, *non*classification rules appear in the output, i.e., rules 4–6. These rules are essentially the opposite of predictive class rules—*given* the class value of "lost," it is highly likely that certain piece configurations occurred, giving useful analysis information.

From Fig. 8 we can also discern the limitation imposed by using only a *conjunctive* hypothesis space for learning. Clearly, the first three rules could be replaced by a more concise rule using the function "any 2 of 3." More generally, the extension to arbitrary "X of N" functions in the hypothesis representation language (of which disjunction ("1 of N") and

Automated Rule Induction - ITRULE

Data Set : 1984 Congressional Voting Records (U.C. Irvine Database)

No. Attributes = 17
No. Examples = 435
Maximum Rule Order = 2
Maximum No.Rules = 10

			p(x|y)	p(y)	p(x)	j(X;y)	J(X;y)	y	xy
1	if[politics = rep]	then [phys-freeze = y]	0.980	0.387	0.418	1.106	0.428	168	163
2	if[phys-freeze = y synfuels = n]	then [politics = rep]	0.967	0.318	0.387	1.142	0.363	138	135
3	if[phys-freeze = y]	then [politics = rep]	0.913	0.407	0.387	0.887	0.361	177	163
4	if[contra-aid = n crime = y]	then [el-salv-aid = y]	0.994	0.380	0.505	0.934	0.355	165	163
5	if[contra-aid = n]	then [el-salv-aid = y]	0.983	0.410	0.505	0.863	0.353	178	172
6	if[phys-freeze = n]	then [politics = dem]	0.988	0.568	0.613	0.619	0.352	247	2
7	if[el-salv-aid = n]	then [contra-aid = y]	0.986	0.478	0.576	0.694	0.332	208	2
8	if[phys-freeze = y mx-missile = n]	then [el-salv-aid = y]	0.994	0.355	0.505	0.931	0.330	154	153
9	if[politics = dem contra-aid = y]	then [phys-freeze = n]	0.981	0.501	0.582	0.656	0.329	218	3
10	if[phys-freeze = n contra-aid = y	then [el-salv-aid = n]	0.937	0.485	0.495	0.672	0.326	211	12

Fig. 7. The 10 best rules from the congressional voting data set (up to order 2).

Automated Rule Induction - ITRULE

Data Set : 7 attribute King-Knight King-Rook Chess End Game (Quinlan, 1979)

No. Attributes = 8
No. Examples = 647
Maximum Rule Order = 2
Maximum No.Rules = 10

			p(x|y)	p(y)	p(x)	j(X;y)	J(X;y)	y	xy
1	if[in-line = t rk.brs.bkg = t]	then [game = safe]	0.790	0.250	0.946	0.207	0.052	161	127
2	if[in-line = t rk.brs.kn = t]	then [game = safe]	0.790	0.250	0.946	0.207	0.052	161	127
3	if[rk.brs.bkg = t rk.brs.kn = t]	then [game = safe]	0.790	0.250	0.946	0.207	0.052	161	127
4	if[game = lost]	then [in-line = t]	0.972	0.054	0.499	0.819	0.044	34	0
5	if[game = lost]	then [rk.brs.bkg = t]	0.972	0.054	0.499	0.819	0.044	34	0
6	if[game = lost]	then [rk.brs.kn = t]	0.972	0.054	0.499	0.819	0.044	34	0
7	if[in-line = f]	then [game = safe]	1.000	0.501	0.946	0.078	0.039	324	324
8	if[rk.brs.bkg = f]	then [game = safe]	1.000	0.501	0.946	0.078	0.039	324	324
9	if[rk.brs.kn = f]	then [game = safe]	1.000	0.501	0.946	0.078	0.039	324	324
10	if[bkg-kn = not 3 wkg-kn = not 1]	then [game = safe]	1.000	0.445	0.946	0.078	0.035	288	288

Fig. 8. The 10 best rules from the chess data set (up to order 2).

conjunction ("N of N") are special cases) is a topic under current investigation [48]. As always, richer representation languages imply a larger search space for the induction algorithm—finding *efficient* representation languages for arbitrary domains remains an open problem.

IX. Experimental Evaluation of the Effectiveness of the Bounds

Given that the computational complexity of ITRULE does not admit to direct analysis, we resorted in Section VII to intuitive arguments as to why we expected it to behave well in practice, on average. Recall that the number of possible rules is exponential in the number of attributes and the cardinality of their event spaces. We argued that in practice our bounds may be expected to become more effective as we go to higher and higher order rules—what we could not show was whether the constraints introduced by the bounds could overcome the tendency of the rule space to grow exponentially. In this section we present experimental evidence, based on the data sets described in the last section, which suggest that, in fact, the bounds are quite powerful. Naturally, for finite sample sizes the small sample bias in the point probability estimates also tends to cut down on the number of rules examined.

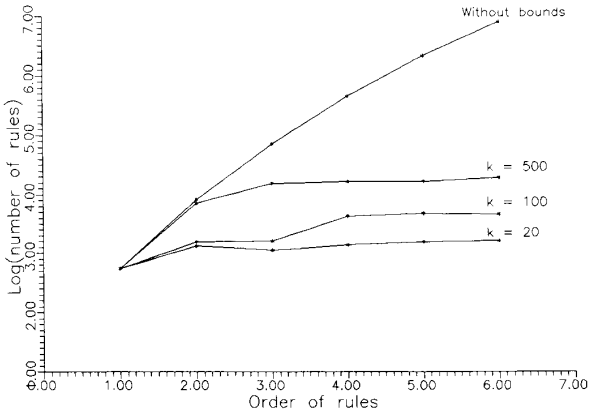

Fig. 9. Cumulative total of rules generated by ITRULE plotted on a log-scale versus rule order, with $k = \{20, 100, 500\}$: the theoretical cumulative total if no bounds were used is also shown.

The data we chose for evaluation purposes were the aforementioned "voting" data set [46], the choice being made primarily on the basis that all variables are binary, hence, permitting relatively easy computation of upper bounds, etc. The algorithm was run with 3 different values of k (20, 100, 500), chosen as a representative of the range of extremes which might be used in practice (from data analysis up to inference). Fig. 9 shows a semi-log plot of the number of rules generated by ITRULE during its search as a function of the order of the rules, e.g., the data point at "order 3" represents the cumulative total of rules of order 3 or less generated by the algorithm. It is important to note that each point was generated by a separate run of the algorithm where the maximum order of rules was restricted (from 1 to 6). As a comparison, the number of rules which would be searched if *no* bounds were used (let us call this parameter $R(i)$ where i is the rule order) is also shown in Fig. 9. This shows the exponential growth of $R(i)$, as can be seen from (20) which gives us

$$R(i) = \sum_{j=1}^{j=i} 2^j . \binom{n-1}{j} . n \tag{27}$$

where $n = 17$ for the voting data set. The benefit of bounding is immediately obvious from Fig. 9.

Fig. 10 shows the noncumulative rule totals on a linear scale using the same data as for Fig. 9, i.e., it plots the number of additional (or new) rules processed from one order to the next. What is evident from this plot is the fact that for each value of k, the algorithm peaks at some order (always 3 or 4 in this case) and from that point onwards, the number of new rules begins to drop off. These same data are presented in a different format in Fig. 11, where we plot the *ratio* of the number of new rules generated at order i to the number at order $i-1$. These are the β factors discussed earlier in Section VII—the graph verifies that the β factors indeed drop below 1 as we would like.

Of course, these results only pertain to one data set, a data set which we have no particular reason to believe is "typical" of data sets in general. However, it has been our experience that the bounding is equally effective on the other data sets reported earlier, and on a variety of unreported data sets. Invariably, there will be cases such as the random problem described earlier, where the bounds may not be effective. However, we believe that such cases will be relatively rare in practice.

X. Conclusion

In this paper we have demonstrated the applicability of our proposed J-measure for induction from both a theoretical and practical standpoint. We developed an interpretation of the measure as a hypothesis preference criterion which trades-off simplicity and goodness-of-fit, and thus supports the basic inductive mechanisms of generalization and specialization. The ITRULE algorithm was described and we gave a practical example in the form of extracting rules from mutual fund data. The rules produced by ITRULE can be used either as a human aid to understanding the inherent model embodied in data, or as a tentative input set of rules to an expert system. In this case, ITRULE can ease the knowledge acquisition bottleneck by presenting the expert with a tentative rule set, or, in cases where no human expert exists, it may directly transform data into rule-based systems. Current work is focused on extensions and refinements of the basic ITRULE ideas and practical applications in a number of domains are in progress.

Appendix

Proof of the Specialization Theorem (Section V):

We consider 3 distinct cases; i) $p_s > p_g$, ii) $p_s < p_x$, and iii) $p_g \geq p_s \geq p_x$.

Case i) $p_s > p_g$:

We can write

$$p(x|y) = p(x|y,z)p(z|y) + p(x|y,\bar{z})p(\bar{z}|y) \tag{28}$$

or equivalently:

$$p_g = p_s.p(z|y) + \theta.(1 - p(z|y)) \tag{29}$$

where $\theta = p(x|y,\bar{z})$. The left-hand side, p_g, is fixed and represents a constraint which p_s, θ and $p(z|y)$ must satisfy. We want to find a variable $\boldsymbol{Z}$ which maximizes J_s subject to this constraint. First we note that by (29):

$$\min\{p_s, \theta\} \leq p_g \leq \max\{p_s, \theta\} \tag{30}$$

since $p(z|y)+p(\bar{z}|y) = 1$. Since we have assumed that $p_s > p_g$ initially, we can state that

$$p_s > p_g > \theta \ . \tag{31}$$

From (29), we have that

$$p(z|y) = \frac{p_g - \theta}{p_s - \theta} \ . \tag{32}$$

Hence, our expression for J_s can be written as

$$J_s = p(y).\left(\frac{p_g - \theta}{p_s - \theta}\right).\left(p_s \log\left(\frac{p_s}{p_x}\right) + (1-p_s)\log\left(\frac{1-p_s}{1-p_x}\right)\right). \tag{33}$$

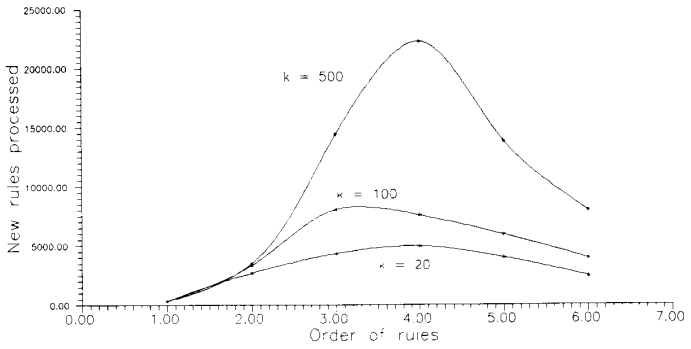

Fig. 10. Noncumulative total of rules generated by ITRULE plotted versus rule order, with $k = \{20.100.500\}$.

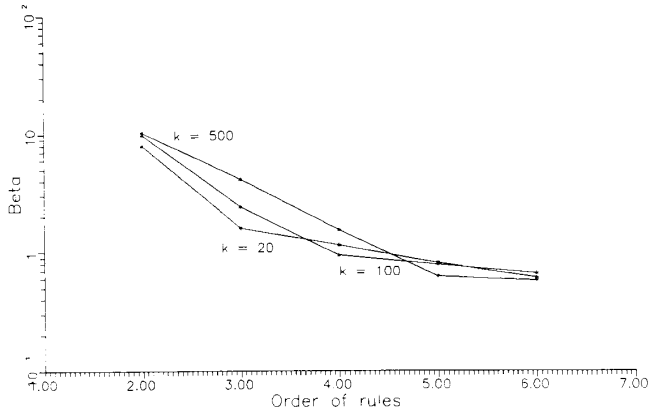

Fig. 11. Beta-factors (β) as generated by ITRULE plotted versus rule order, with $k = \{20.100.500\}$.

The only remaining free parameters are p_s and θ, which we will choose to maximize J_s. The probabilities p_s and θ are jointly constrained by the fact that

$$0 \leq \left(\frac{p_g - \theta}{p_s - \theta}\right) \leq 1 \ . \tag{34}$$

Since this is a multiplicative term in J_s, to maximize J_s we should maximize this term and then check if the value of this maximum constrains p_s in any way. If it does, then we cannot maximize the product terms in (33) to find an achievable bound. From (31) we know that $0 \leq \theta < p_g$. The following lemma is useful.

Lemma A.1:

$$\max_{0 \leq \theta < p_g} \left\{\frac{p_g - \theta}{p_s - \theta}\right\} = \frac{p_g}{p_s} \tag{35}$$

Proof: Let $\theta_1 < \theta_2$. Therefore:

$$\theta_1(p_s - p_g) < \theta_2(p_s - p_g) \tag{36}$$

since $p_s - p_g > 0$, and by adding $p_s p_g + \theta_1\theta_2$ to each side we obtain

$$p_s p_g + \theta_1\theta_2 + \theta_1 p_s - \theta_1 p_g < p_s p_g + \theta_1\theta_2 + \theta_2 p_s - \theta_2 p_g \tag{37}$$

$$\Rightarrow (p_g - \theta_2).(p_s - \theta_1) < (p_s - \theta_2).(p_g - \theta_1) \tag{38}$$

and so

$$\frac{p_g - \theta_2}{p_s - \theta_2} < \frac{p_g - \theta_1}{p_s - \theta_1} \tag{39}$$

which implies that the maximum occurs for $\theta = 0$.

Accordingly, the choice of $\theta = 0$ minimizes the multiplicative term in (33) without introducing any extra constraints on p_s. Hence, we can maximize the two terms separately and still obtain an achievable bound. We will refer to this bound as *the product bound*. From (33) and the result of the lemma we obtain

$$J_s \leq p(y).\frac{p_g}{p_s}.\left(p_s \log\left(\frac{p_s}{p_x}\right) + (1 - p_s)\log\left(\frac{1 - p_s}{1 - p_x}\right)\right) \tag{40}$$

$$= p(y).p_g.\left(\log\left(\frac{p_s}{p_x}\right) + \left(\frac{1}{p_s} - 1\right).\log\left(\frac{1 - p_s}{1 - p_x}\right)\right) \tag{41}$$

$$\leq p(y).p_g.\log\left(\frac{p_s}{p_x}\right) \quad \text{(since the second term is negative)} \tag{42}$$

$$\leq p(y).p_g.\log\left(\frac{1}{p_x}\right) \tag{43}$$

This proves case i) of the Theorem.

Next we consider case ii) where $p_s < p_x < p_g$. Intuitively what happens here is that the new condition "changes the direction of the rule" so that $\bar{x}$ is being confirmed rather than x. In practice this case is far less likely to occur than case i). Nonetheless, we must analyze this case to obtain a general bound. Proceeding as in case i) we get the equivalent condition to (31):

$$p_s < p_x < p_g < \theta \tag{44}$$

and so we have that

$$p(z|y) = \frac{\theta - p_g}{\theta - p_s} . \tag{45}$$

Lemma A.2:

$$\max_{p_g < \theta \leq 1} \left\{ \frac{\theta - p_g}{\theta - p_s} \right\} = \frac{1 - p_g}{1 - p_s} \tag{46}$$

Proof: Let $\theta_1 > \theta_2$. Therefore:

$$\theta_1(p_g - p_s) > \theta_2(p_g - p_s) \tag{47}$$

since $p_g - p_s > 0$, and by adding $p_s p_g + \theta_1\theta_2$ as before to each side we obtain

$$(\theta_2 - p_s).(\theta_1 - p_g) > (\theta_2 - p_g).(\theta_1 - p_s) \tag{48}$$

$$\Rightarrow \frac{\theta_1 - p_g}{\theta_1 - p_s} > \frac{\theta_2 - p_g}{\theta_2 - p_s} \tag{49}$$

and so, unlike case i), the maximum occurs for $\theta = 1$.

As before, maximizing the product term does not constrain p_s in any way since $0 < \frac{(1-p_g)}{(1-p_s)} < 1$ for all allowable values of p_s. Hence, we have that

$$J_s \leq p(y).\left(\frac{1 - p_g}{1 - p_s}\right).\left(p_s \log\left(\frac{p_s}{p_x}\right) + (1 - p_s)\log\left(\frac{1 - p_s}{1 - p_x}\right)\right) \tag{50}$$

$$= p(y).(1 - p_g).\left(\frac{p_s}{1 - p_s}\log\left(\frac{p_s}{p_x}\right) + \log\left(\frac{1 - p_s}{1 - p_x}\right)\right) \tag{51}$$

Since $\log(\frac{p_s}{p_x}) < 0$, given that $p_s < p_x$, then

$$J_s \leq p(y).(1 - p_g).\log\left(\frac{1}{1 - p_x}\right) . \tag{52}$$

This proves the bound for case ii). For case iii) we can apply the following arguments. If $p_s = p(x)$ then $J_s = 0$ and so the bound holds. If $p(x) < p_s < p_g$, then from case ii) we see that the simplicity component

$$p(y).\left(\frac{\theta - p_g}{\theta - p_s}\right) \leq p(y) \tag{53}$$

while the goodness-of-fit component

$$j_s \leq p_g.\log(\frac{p_g}{p_x}) + (1 - p_g).\log(\frac{1 - p_g}{1 - p_x}) \tag{54}$$

$$< p_g.\log(\frac{p_g}{p_x}) \tag{55}$$

due to the fact that the second term is negative since $(1-p_g) < (1 - p_x)$. Hence, we get that

$$J_s < p(y).p_g.\log(\frac{p_g}{p_x}) \tag{56}$$

, which is less than (43), the bound for case i). Finally if $p_s = p_g$, we can apply the same argument for the goodness-of-fit, j_s, and noting that the simplicity component must be less than or equal to $p(y)$, we obtain the same result as (52). By combining the results of cases i)–iii), we obtain the desired result. This proves the theorem in its entirety, which we will now restate:

$$J_s \leq p(y).\max\left\{p_g \log\frac{1}{p_x}.\ (1 - p_g)\log\frac{1}{1 - p_x}\right\} \tag{57}$$

$$= \max\left\{p(x.y)\log\frac{1}{p_x}.\ p(\bar{x}.y)\log\frac{1}{p_{\bar{x}}}\right\}. \tag{58}$$

ACKNOWLEDGMENT

The authors gratefully acknowledge the assistance of David Aha of the University of California-Irvine in providing the voting data set, and also Brain Gaines of the University of Calgary and Ross Quinlan of the University of Sydney for providing the chess data set.

REFERENCES

[1] N. E. Johnson, "Mediating representations in knowledge elicitation," in *Proc First European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Reading, England, 1987.
[2] P. E. Johnson, "What kind of expert system should a system be?," *J. Med. Philosophy*, vol. 8, pp. 77–97, 1983.
[3] A. Hart, *Knowledge Acquisition for Expert Systems.* New York: McGraw Hill, 1986.
[4] D. Kahneman, P. Slovic, and A. Tversky, *Judgement under Uncertainty: Heuristics and Biases.* Cambridge, England: Cambridge University, 1982.
[5] Y. Bishop, S. E. Fienberg, and P. W. Holland, *Discrete Multivariate Analysis: Theory and Practice.* Cambridge, MA: MIT, 1975.
[6] A. K. C. Wong and D. K. Y. Chiu, "Synthesizing statistical knowledge from incomplete mixed-mode data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, pp. 796–805, June 1987.
[7] D. V. Lindley, "Scoring rules and the inevitability of probability," *Int. Statist. Rev.*, vol. 50, pp. 1–26, Jan. 1986.
[8] P. Cheeseman, "In defense of probability," in *Proc. Ninth Int. Joint Conf. on Artificial Intelligence.*, vol. 2, 1985, pp. 1002–1009.
[9] R. M. Goodman and P. Smyth, "An information-theoretic model for rule-based expert systems," presented at the 1988 Int. Symp. on Information Theory, Kobe, Japan, 1988.
[10] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.
[11] E. M. Gold, "Language identification in the limit," *Inform. Control*, 10, pp. 447–474, 1967.
[12] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, Nov. 1984.
[13] D. Haussler, "Bias, version spaces and Valiant's learning framework," *Proc. Fourth Int. Workshop on Machine Learning*, 1987, pp. 324–336.
[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, D. E. Rumelhart and J. J. McClelland, eds. Cambridge, MA: MIT, 1986, pp. 318–363.
[15] P. R. Cohen and E. A. Feigenbaum, *The Handbook of Artificial Intelligence (Vol. 3).* Los Altos, CA: William Kaufmann, 1982.
[16] T. M. Mitchell, "Generalization as search," *Artif. Intell.*, vol. 18, no. 2, pp. 203–226, 1982.
[17] R. S. Michalski and J. B. Larson, "Selection of most representative training examples and incremental generation of VL1 hypotheses," Rep. 867, Computer Science Department, Univ. of Illinois, 1978.

[18] R. S. Michalski and R. L. Chilausky, "Learning by being told and learning from examples", *Int. J. Policy Anal. Inform. Syst.*, vol. 4, pp. 125–161, 1980.
[19] J. R. Quinlan, "Induction of decision trees," *Mach. Learning*, vol. 1, pp. 81–106, 1986.
[20] J. R. Quinlan and R. L. Rivest, "Inferring decision trees using the minimum description length principle," *Inform. Comput.*, vol. 80, pp. 227–248, Jan. 1989.
[21] B. Arbab and D. Michie, "Generating rules from examples," in *Proc. Ninth Int. Joint Conf. on Artificial Intelligence*, 1985, pp. 631–633.
[22] R. M. Goodman and P. Smyth, "Decision tree design from a communication theory standpoint," *IEEE Trans. Infor. Theory*, vol. 34, pp. 979–994, Sept. 1988.
[23] ——, "Decision tree design using information theory," *Knowledge Acquisition*, vol. 2, pp. 1–19, 1990.
[24] B. R. Gaines and M. L. G. Shaw, "Induction of inference rules for expert systems," *Fuzzy Sets Sys.*, vol. 18, no. 3, pp. 315–328, Apr. 1986.
[25] J. Boose, "Personal construct theory and the transfer of expertise," in *Proc. AAAI*, 1984, pp. 27–33.
[26] J. G. Ganascia, "Learning with Hilbert cubes," in *Proc. Second European Workshop on Machine Learning (EWSL)*, Bled, Yugoslavia, 1987.
[27] J. R. Quinlan, "Generating production rules from examples," in *Proc. Tenth Int. Joint Conf. on Artificial Intelligence*, 1987, pp. 304–307.
[28] J. Cendrowska, "PRISM: An algorithm for inducing modular rules," *Int. J. Man-Machine Studies*, vol. 27, pp. 349–370, 1987.
[29] P. Clark and T. Niblett, "The CN2 induction algorithm," *Mach. Learning*, vol. 3, pp. 261–283, 1989.
[30] R. L. Rivest, "Learning decision lists," *Mach. Learning*, vol. 2, pp. 229–246, 1987.
[31] P. Cheeseman, "Learning of expert systems from data," in *Proc. First IEEE Conf. on Applications of Artificial Intelligence*, 1984.
[32] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no.3, pp. 379–423, July 1948.
[33] N. M. Blachman, "The amount of information that y gives about X," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 27–31, Jan. 1968.
[34] P. Smyth and R. M. Goodman, "The information content of a probabilistic rule," to be published.
[35] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 26–37, Jan. 1980.
[36] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.
[37] R. E. Blahut, *Principles and Practice of Information Theory* Reading, MA: Addison-Wesley, 1987.
[38] D. Angluin and C. Smith, "Inductive inference: Theory and methods," *ACM Comput. Surveys*, vol. 15, no. 3, pp. 237–270, 1984.
[39] B. R. Gaines, "Behavior/structure transformations under uncertainty," *Int. J. Man-Mach. Stud.*, vol. 8, pp. 337–365, 1976.
[40] R. S. Michalski, "Pattern recognition as rule-guided inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, pp. 349–361, July 1980.
[41] J. H. Holland, K. J. Holyoak, R. E. Nisbett, and P. R. Thagard, *Induction: Processes of Inference, Learning and Discovery*. Cambridge, MA: MIT, 1986.
[42] I. J. Good, "The estimation of probabilities: An essay on modern Bayesian methods," Res. Monograph 30, MIT, Cambridge: MA, 1965.
[43] W. Feller, *An Introduction to Probability Theory and its Applications* New York: Wiley, vol. 1, 1968.
[44] American Association of Investors, *The Individual Investor's Guide to No-load Mutual Funds*. Chicago, IL: International, 1987.
[45] *Congressional Quarterly Almanac, 98th Congress*, 2nd session 1984, Washington DC, 1985.
[46] J. C. Schlimmer, "Concept acquisition through representational adjustment," Ph. D. dissertation, Dep. Comput. Sci., Univ. of California at Irvine, CA, 1987.
[47] J. R. Quinlan, "Discovering rules by induction from large collections of examples," in *Expert Systems in the Micro-electronic Age*, D. Michie, ed. Edinburgh, Scotland: Edinburgh University, 1979.
[48] R. M. Goodman, J. W. Miller, and P. Smyth, "The information provided by a linear threshold function with binary weights," presented at the 1990 IEEE Int. Symp. Information Theory, San Diego, CA, Jan. 1990.

Padhraic Smyth (S'85–M'88) was born in County Mayo, Ireland, on July 25, 1962. He received the B.E. degree from the University College Galway, National University of Ireland, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1985 and 1988, respectively.

From 1985 to 1988, he worked part-time as a Research Consultant with Pacific Bell in the areas of telecommunications switching systems and automated network management. In 1988, he joined the Communication Systems Research Section, Jet Propulsion Laboratory, Pasadena, CA. His research interests include information theory, telecommunications, pattern recognition, machine learning, and the statistical modeling of problems in artificial intelligence.

Rodney M. Goodman (M'85) was born in London, England, on February 22, 1947. He received the B.Sc. degree in electrical engineering from Leeds University, Yorkshire, England, in 1968, and the Ph.D. degree in electronics from the University of Kent at Canterbury, in 1975.

In 1985, he joined the Faculty of the Department of Electrical Engineering, California Institute of Technology as an Associate Professor. He has consulted for a wide variety of government and commercial organizations and is currently a consultant for the Jet Propulsion Laboratory and Pacific Bell. He is also a founder of two advanced technology research and development companies in England. His research interests include error control coding, cryptography, medical electronics, neural networks, and expert systems.

Dr. Goodman is a Chartered Electrical Engineer.