9 Rule Induction using Information Theory

Padhraic Smyth
Communication Systems Research
Jet Propulsion Laboratory

Rodney M. Goodman
Department of Electrical Engineering
California Institute of Technology

Abstract

Across a variety of scientific, engineering, and business applications it has become commonplace to collect and store large volumes of data. For example, NASA has warehouses of data collected from inter-planetary scientific missions, most of which cannot be processed or examined at present because there simply are not enough scientists and statisticians to sift through it all. On the other hand, we have at our disposal previously unimaginable amounts of computational power due to advances in VLSI technology. Hence, it seems obvious that the development of computation-intensive techniques which explore large databases is a major research challenge as both data volume and computing power continue to increase. In this paper we consider the problem of *generalized* rule induction from databases and provide an overview of our recent work on this topic using information-theoretic models. In generalized rule induction we seek the best predictive rules relating all, or any subset of, the domain variables. This approach is particularly useful for initial analysis of large sets of discrete and/or categorical data, allowing, for example, important causal dependencies to become apparent. We describe the necessary information theoretic and probabilistic foundations for this approach, defining the information content of a probabilistic rule. Given these basic tools we then show how they can be incorporated into a powerful learning algorithm called ITRULE, and we discuss the practical applications of this algorithm to problems such as exploratory data analysis, identification of causal models, and knowledge acquisition for expert systems. In our conclusion we discuss some general research issues which remain to be addressed in this field.

9.1 Introduction

The emergence of electronic and magnetic storage media as convenient and affordable methods to store large amounts of data has led to the coining of phrases such as "the information revolution." The popular notion appears to be that the widespread availability of information will considerably accelerate man's technical progress, this new age being a modern-day equivalent of the industrial revolution of the last century. With the continued progress in increasing the information capacity of both *storage* media (e.g., high density VLSI memory chips, optical discs) and *transmission* media (e.g., optical fibers), one can only predict that the volume of electronic data will continue to grow at a phenomenal rate.

Yet, despite the progress in *handling* this information from a hardware standpoint, progress in *using* the information continues to lag far behind. One of the primary reasons is the sheer quantity and volume of the data. Simply put, there is not enough manpower to analyse and examine the typical large corporate database. For example, in the telecommunications industry at present, there exist very sophisticated networks which automatically report a vast array of traffic information, data on module failures, system performance analyses, etc. These reports are automatically "logged," in turn, on a database system, as a historical record of network operations. However, although the databases contain a wealth of information in terms of system performance and fault diagnosis, they are often too complex to search manually. Another familiar example is the automatic scanners used at checkout counters in modern-day supermarkets. This "scan data" is automatically recorded and used for market research purposes. The volume of data available overwhelms what was previously a manual market-analysis task. This general trend is extremely common across a variety of disciplines.

The premise of this volume is that the *automated* analysis of such large databases is obviously necessary and worthwhile. While the lofty goals of knowledge discovery are worthy indeed, we must nonetheless begin our research at a more concrete level. In this chapter we are going to look at what seems like an innocously simple problem:

> *Given a database described in terms of discrete and/or categorical attributes what are the best rules which characterize the data ?*

We begin by defining the problem in more formal terms, defining and justifying the necessary probabilistic pre-requisites underlying our approach. We then examine the idea of quantifying the quality of a probabilistic rule, and demonstrate the notion of rule information content. In addition we devote some attention to the

problem of robustly estimating probabilities directly from data. Armed with these basic "tools of the trade" we can begin to address the problem described above, that of finding the most informative rules from a data set. In particular we describe the ITRULE algorithm which uses computationally intensive search techniques to search the data for the rules of greatest information content. The workings of the algorithm have been reported in detail elsewhere (Smyth and Goodman, 1990b), hence, the focus here will be more on the applications of the algorithm and the types of problems and data to which it is best suited. We conclude by discussing open problems and research issues.

9.2 The probabilistic rule representation

We define a probabilistic rule as an if-then statement to the effect that if proposition y occurs then there is a probability p that proposition x is true and a probability $1-p$ that proposition $\bar{x}$ is true. It is convenient to define the probability p as the conditional probability $p(x|y)$. Hence our probabilistic rule corresponds to a simple statement regarding the conditional probability of one event given another. While other methods of representing uncertainty have been proposed and are in common use (such as fuzzy logic (Zadeh, 1965), and certainty factors (Adams, 1976)), standard probability theory remains the established and preferred uncertainty model due to its theoretical foundations and proven utility.

Letting $\mathbf{X}$ and $\mathbf{Y}$ be discrete random variables, then x and y are letters from their respective discrete alphabets (as a notational convenience we adopt the convention that $p(y)$ stands for $p(\mathbf{Y}=y)$, etc., as is customary in discussions of this nature). A common situation is where $\mathbf{X}$ is the *class* variable, and $\mathbf{Y}$ is a composite variable of several discrete or categorical *attribute* variables. In this manner our probabilistic rules would be classification rules of the form:

$$\text{If } (\mathbf{Y_1}=y_1 \text{ and } \mathbf{Y_2}=y_2) \text{ then } \mathbf{X}=x \text{ with probability } p$$

Why should we look for rules at all, and probabilistic ones at that ? The rule-based representation plays a central role in most theories of knowledge representation, going back to the early work of Chomsky (1957), the cognitive production rule models of Newell and Simon (1972), and the more recent work of Holland et al. (1986). While the debate continues regarding the virtues of competing cognitive models (such as connectionism) there can be no denying the utility of the rule-based representation, i.e., whether or not we believe that rules are truly part of human reasoning processes they provide a practical and convenient mechanism

by which to explicitly represent knowledge. For example, witness the proliferation of rule-based expert systems as a practical software engineering paradigm.

So why add probabilities to our rules ? There are two ways to answer this. The first answer is that although production rule systems have their roots in logic, our perception of the real world tends to be couched in uncertainty. For example, most successful rule-based expert systems tend to add uncertainty measures to their rules, albeit often in an *ad hoc* manner. So the first answer says that by necessity we need to use probability to deal with real-world problems. The second answer to the question, that of an information theorist, is more dogmatic. Simply put, probabilistic models are a generalization of deterministic models, and, as such, provide a much more expressive and powerful mathematical language to work with. A layman's interpretation of this statement might be that any system which uses probability correctly can always do better than a similar system which has no concept of probability.

Hence, probabilistic rules are a simple and useful technique for knowledge representation. While there exist far more sophisticated knowledge representation schemes, it seems more appropriate that we should begin work on automated knowledge discovery at a fairly simple level. As we shall see later, even the discovery of simple probabilistic rules in data can reveal a wealth of hidden information.

9.3 The information content of a rule

Given a set of probabilistic rules we will need to be able to compare and rank the rules in a quantitative manner, using some measure of "goodness" or utility. The approach we propose is to define the *information content* of a rule, using ideas from information theory. Information theory can be considered a layer above pure probability theory — typically, *given* a set of defined probabilities we wish to calculate various information-based quantities. Traditionally, information theory has a distinguished history of providing elegant solutions to communications problems, originating with Claude Shannon's pioneering work (Shannon, 1948). The relation between communication theory and inductive inference is quite appealing. With communication systems we are involved in the efficient transmission and reception of information from point A to point B. In inductive inference, we are effectively at point B, receiving a message (the data) via some sensory channel, from the environment (point A). In particular, unlike communications applications, we do not know what "code" is being used or what the noise characteristics of the channel (measurement process) are. For example, in classification, we may be trying to

infer the value of the class variable, given related attribute information. In effect, the attributes form a code for the class, which is then corrupted by measurement noise. Even in the presence of perfect information (no measurement noise), the class may be coded ambigously by the available attributes, i.e., there may only exist a probabilistic (rather than deterministic) mapping between the attributes and the class, due to the presence of unmeasured causal variables. For the classifier design problem we have used this analogy to improve our understanding of decision tree design techniques (Goodman and Smyth, 1988a, 1990) and, in a more general sense, the powerful technique of inductive inference via Minimum Description Length Encoding (Rissanen (1989), Quinlan and Rivest (1989)) is also motivated by this communications problem analogy.

Hence, it seems clear that information theory should provide a theoretically sound and intuitively practical basis for our problem of finding the best rules from given data. The first task is to define the information content of a probabilistic rule, where we remind ourselves that a probabilistic rule is defined as

$$\text{If } \mathbf{Y} = y \text{ then } \mathbf{X} = x \text{ with probability p}$$

We have recently introduced a measure called the J-measure for precisely this purpose (Goodman and Smyth, 1988, Smyth and Goodman, 1990a), defined as

$$J(\mathbf{X};\mathbf{Y}=y) = p(y)\left(p(x|y).\log\left(\frac{p(x|y)}{p(x)}\right) + (1-p(x|y)).\log\left(\frac{(1-p(x|y))}{(1-p(x))}\right)\right)$$

This measure possesses a variety of desirable properties as a rule information measure not least of which is the fact that it is unique as a non-negative measure which satisfies the requirement that

$$\sum_y J(\mathbf{X};\mathbf{Y}=y) = I(\mathbf{X};\mathbf{Y})$$

where $I(\mathbf{X};\mathbf{Y})$ is the average mutual information between the variables $\mathbf{X}$ and $\mathbf{Y}$ as originally defined by Shannon (1948). This states that the sum of the information contents (of a set of rules with mutually exclusive and exhaustive left-hand sides) must be equal to the well known average mutual information between two variables. The interested reader is referred to Smyth and Goodman (1990a) for a detailed treatment of the various mathematical properties of the measure. We note in passing that other measures of rule goodness have been proposed, such as that of Piatetsky-Shapiro (1990). who proposes the use of $p(y)(p(x|y) - p(x))$. Measures such as this, based directly on probabilities, will tend to assign less weight to

rarer events compared to measures such as the J-measure which use a log scale (information-based). To a large extent, such "information-based" and "correlation-based" measures in practice often rank rules in a similar order — however, the J-measure's relation to Shannon's average mutual information makes it more desirable from a theoretical point of view.

Intuitively we can interpret the J-measure as follows. Let us decompose the J-measure into two terms, namely $p(y)$ and $j(\mathbf{X};\mathbf{Y}=y)$ where

$$j(\mathbf{X};\mathbf{Y}=y) = p(x|y).\log\left(\frac{p(x|y)}{p(x)}\right) + (1-p(x|y)).\log\left(\frac{(1-p(x|y))}{(1-p(x))}\right)$$

The probability term $p(y)$ can be viewed as a preference for generality or simplicity in our rules, i.e., the left-hand side must occur relatively often in order for a rule to be deemed useful. The other term, $j(\mathbf{X};\mathbf{Y}=y)$, is familiar to information theorists as a distance measure (namely, the cross entropy) between our *a posteriori* belief about $\mathbf{X}$ and our *a priori* belief. Cross entropy is a well-founded measure of the goodness-of-fit of two distributions (Shore and Johnson, 1980). Hence, maximizing the product of the two terms, $J(\mathbf{X};\mathbf{Y}=y)$, is equivalent to simultaneously maximizing both the simplicity of the hypothesis y, and goodness-of-fit between y and a perfect predictor of $\mathbf{X}$. There is a natural trade-off involved here, since typically one can easily find rare conditions (less probable y's) which are accurate predictors, but one has a preference for more general, useful conditions (more probable y's). This basic trade-off between accuracy and generality (or goodness-of-fit and simplicity) is a fundamental principle underlying various general theories of inductive inference (Angluin and Smith (1984) , Rissanen (1989)).

Symptom A	Symptom B	Disease x	Joint Probability
no fever	no sore throat	absent	0.20
no fever	no sore throat	present	0.00
no fever	sore throat	absent	0.30
no fever	sore throat	present	0.10
fever	no sore throat	absent	0.02
fever	no sore throat	present	0.08
fever	sore throat	absent	0.03
fever	sore throat	present	0.27

Table 1: Joint probability distribution for medical diagnosis example

An example of the J-measure in action will serve to illustrate its immediate applicability. Consider the three attributes shown in Table 1, along with their associated

joint probability distribution. The data is supposed to represent the hypothetical distribution of patients arriving into a doctor's office. In practice we might have a large sample of patient data available, in which case the joint distribution shown in Table 1 might be an *estimate* of the true distribution. The attributes "fever" and "sore throat" represent whether or not a patient exhibits these symptoms at present, while "disease x" is some mysterious illness which each patient actually will or will not develop at some point in the future.

We are of course interested in predictive "symptom-disease" rules, of the variety a medical practioner might use in the course of a cursory diagnosis. Note that from Table 1 alone, or indeed from the original sample data, it would be very difficult to manually detect the most informative rules. In Table 2 we list the rule conditional probability $p(x|y)$, the left-hand side probability ($p(y)$), the cross entropy $j(\mathbf{X};\mathbf{Y}=y)$, and their product $J(\mathbf{X};\mathbf{Y}=y)$ for each of 6 possible rules.

Rule	Rule Description	$p(x\|y)$	$p(y)$	$j(\mathbf{X};y)$	$J(\mathbf{X};y)$
1	if *fever* then *disease x*	0.875	0.4	0.572	0.229
2	if *sore throat* then *disease x*	0.5285	0.7	0.018	0.012
3	if *sore throat and fever* then *disease x*	0.9	0.3	0.654	0.196
4	if *sore throat and no fever* then *not disease x*	0.75	0.4	0.124	0.049
5	if *no sore throat and no fever* then *not disease x*	1.0	0.2	0.863	0.173
6	if *sore throat or fever* then *disease x*	0.5625	0.8	0.037	0.029

Table 2: Hypothetical predictive 'symptom-disease' rules and their information content

The three best rules, as ranked by information content, are 1, 3 and 5, in that order. Rule 5 is a perfect predictor of a patient not having the disease, however it only occurs 20% of the time, limiting its utility. Rules 2, 4 and 6 are of limited predictive value because for each rule the conditional probability is relatively close the prior probability of the right-hand side. Hence, the information content for each is low. If we used cross entropy ($j(\mathbf{X};y)$ as the ranking criterion (and ignored the probability $p(y)$) rule 1 would only rank third. When $p(y)$ is taken into account, rule 1 provides the best generality/accuracy trade-off with the highest J-measure of 0.229 bits of information.

In practice, since rule 3 is a more specialized form of rule 1, with lower information content, it serves no practical purpose and would be eliminated from a simple model. Hence, a practioner might choose to remember only rules 1 and 5 from this set, and seek information regarding other symptoms if neither of these rules' conditions

are met (prior to making a diagnosis). This simple hypothetical example serves to illustrate the utility of the J-measure. The next step is to automate the rule finding procedure, i.e., to define an algorithm which automatically finds the most informative rules.

9.4 The ITRULE rule-induction algorithm

Let us formally define the generalized rule induction problem once again, in the context of information content:

> Given a set of K discrete (and/or categorical) random variables (called features or attributes), and a set of N sample vectors (i.e., instances or samples of the attributes, perhaps a database), find the set of R most informative probabilistic rules from the data, where probabilistic rules consist of conjunctions of attribute values on the left-hand side, a single attribute-value assignment on the right-hand side, and an associated conditional probability value. Calling one of the attributes the "class" and simply deriving classification rules is a special case.

A cursory glance at the literature on machine learning will confirm that there are many flavors and varieties of rule-induction algorithms. A significant number of these algorithms are based on symbolic, non-statistical techniques, for example the AQ15 algorithm of Michalski et al.(1986). While such learning algorithms provide useful *qualitative* insights into the basic nature of the learning problem, we believe that a statistical framework is necessary for any robust, practical learning procedure, in particular for real-world problems. Many rule-induction algorithms which use a statistical basis fall into the tree-based classifier category, for example the well known ID3 algorithm (Quinlan, 1986) and its variants. These algorithms derive classification rules in the form of a tree structure. The restriction to a tree structure makes the search problem much easier than the problem of looking for general rules. Quinlan has more recently proposed the C4 algorithm (Quinlan, 1987) which prunes back an original ID3-like tree structure to a set of modular rules. Clark and Niblett (1988) described the CN2 algorithm which produces a decision-list classifier structure, allowing arbitrary subsets of categorical events to be used as tests at intermediate nodes in the list. Both of these techniques, and almost all related algorithms, are strictly *classifiers*, and all use some form of restricted rule structure (tree, decision list) allowing the search algorithm to use a divide-and-conquer strategy in searching the data. The only vaguely similar approaches to the problem of *generalized* rule induction of which we are aware is a Bayesian

approach presented by Cheeseman (1984) and the ENTAIL algorithm of and Gaines and Shaw (1986) which is based on fuzzy logic measures rather than probability theory. In addition, in this volume, Piatetsky-Shapiro (1990) describes an approach which looks at generalized rule induction for strong rules, where "strong" is defined in the sense of having rule transition probabilities near 1.

The problem of generalized rule-induction is quite difficult. One cannot partition the data in a simple divide-and-conquer manner, making the search for rules considerably more computationally demanding than tree induction. We have developed an efficient algorithm for the problem, namely the ITRULE algorithm (Goodman and Smyth, 1988b,c, 1989, Smyth and Goodman, 1990b).

The input to the algorithm consists of the data (a set of N discrete and/or categoric-valued "attribute vectors"), R (the number of rules required), and s, the maximum size of the conjunctions allowed in the rules where $1 \leq s \leq K - 1$. The algorithm returns as output the R most informative rules, up to order s, in rank order of information content. The "order" of a rule is defined as the number of conjunctions on the left-hand side of the rule. In addition, the user can supply a constraint matrix (size $K \times K$) of left-hand side/right-hand side attribute combinations, where an entry of "1" indicates that that combination is not to be considered among the candidate rules, and a "0" entry the opposite. The default value for the matrix is the identity matrix. This constraint matrix is a simple technique to restrict the focus of attention of the algorithm to rules of interest to the user. For example, this allows one to enforce causal constraints or to implement the special case of classification rules for a specific attribute.

The algorithm operates by keeping a list of the R best rules found so far as it searches the rule space. It considers in turn each of the possible first-order rules for each possible right-hand side, calculates their J-measure and includes them in the rule list if their information content is greater than that of the Rth best rule found so far. The J-measure calculations are made based on *estimates* from the data of the various probabilities involved. This estimation step is a critical element of the algorithm and is described in more detail in the Appendix. A decision is then made whether or not it is worth specializing the rule further. Specializing the rule consists of adding extra conditions to the left-hand side. The key efficiency of the algorithm lies in the fact that it uses information-theoretic bounds to determine how much information can be gained by further specialization (Smyth and Goodman 1990a,b). If the upper bound on attainable information content is *less* than the information of the Rth rule on the list, the algorithm can safely ignore all specializations of that rule and backs up from that point. In this manner it continues to search and bound until it has covered the entire space of possible rules.

The worst-case complexity of the algorithm is exponential in number of attributes. More precisely, for K m-ary attributes (i.e., attributes which can take on m values), the number of possible rules to be examined by the algorithm is

$$R = Km\Big((2m+1)^{K-1} - 1\Big)$$

where $m = 1$ for the special case of binary attributes. However this worst-case scenario can only occur if the attributes are all entirely independent of each other (so that none of the bounds take effect) *and* the size of the training data set is significantly greater than 2^K (so that one is not limited by small-sample estimation effects). In practical situations, the combination of bounds and small sample bias ensure that the algorithm rarely searches any rules of order much greater than 3 or 4 — in Smyth and Goodman (1990b) we have shown empirical results validating this effect on well-known data sets. The size of the data set N is only a linear factor in the complexity, i.e., doubling the size of the data set will cause the algorithm to take roughly twice as long to run. A more significant practical limitation is the alphabet size m. In speech and computer vision problems, m can be quite large, for example, for the text-to-phoneme mapping problem (Sejnowski and Rosenberg, 1987) $m = 26$. A practical approach to this problem is to limit allowable order s of the rules to say 2 or 3, a sub-optimal but necessary fix.

9.5 Applications of the ITRULE algorithm

The ITRULE algorithm is ideally suited for problems with a large number of discrete-valued or categorical variables whose interaction is poorly understood, i.e., where there is little prior domain knowledge. In particular, domains characterized by non-linear relationships are particularly well-matched by the probabilistic rule representation. Applications of the algorithm can be characterized into four basic categories:

1. **Exploratory data analysis:** The algorithm is perhaps most useful for generating an initial understanding of dependencies among variables, causal relationships, etc. In practice this tends to be very useful to get a "feel" for the data. One of the early succesful applications of ITRULE was to a financial database describing the characteristics and performance of a variety of mutual fund investment companies averaged over a five-year time period (Goodman and Smyth, 1988c). The algorithm extracted a number of interesting (and previously unknown) general domain rules.

2. **Knowledge acquisition for expert systems:** The probabilistic rule output can be used directly as the knowledge-base for an expert system. Hence, one can use ITRULE to automate the rule elicitation process, circumventing the often inefficient manual knowledge acquisition methodologies. Indeed, even when no database is available, one can in principle use expert-supplied case studies as a synthetic data set. We have routinely used the algorithm to produce rules from data for various commercial rule-based shells — the ability to go directly from data to a working expert system is particularly powerful, allowing for rapid prototyping of a system and iterative improvement by adding new attributes and rerunning the induction. Goodman et al.(1989) report an application of this technique to the development of expert systems for telecommunications network management and control.

3. **Rule-based classifiers:** By running the algorithm to find only classification rules, the resulting rule set forms a hybrid rule-based/probabilistic classifier. This classifier, which has achieved excellent classification performance in empirical tests (Smyth et al., 1990a), uses appropriate conditional independence assumptions to combine rule probabilities into an estimate of the class probability. In addition, the equivalent log-likelihoods or "weights of evidence" (for each rule which contributes to the estimate) can be used to construct an explanation of how the classification decision was arrived at, providing the basis for a decision support system.

4. **Identification of Markov chains:** By interpreting state transitions in a Markov chain as probabilistic rules, the algorithm can be used to estimate Markov chain structure from data. For example one can infer general prediction and performance rules for complex engineering systems directly from a system simulation (Smyth et al. 1990b). In addition, for speech and computer vision problems the technique shows considerable potential for detecting high-order components of Hidden Markov Models, Markov Random Fields, etc.

As an example of the output of the algorithm we show in Table 3 the 8 best rules obtained for the congressional voting records database as described by Schlimmer (1987) (and available publicly from the U.C. Irvine Machine Learning database).

Rule	Rule Description	$p(x\|y)$	$p(y)$	$J(\mathbf{X};y)$
1	if *politics:republican* then *phys-freeze:yes*	0.980	0.387	0.428
2	if *phys-freeze:yes and syn-fuels:no* then *politics:republican*	0.967	0.318	0.363
3	if *phys-freeze:yes* then *politics:republican*	0.913	0.407	0.361
4	if *contra-aid:no and crime:yes* then *el-salv-aid:yes*	0.994	0.380	0.355
5	if *contra-aid:no* then *el-salv-aid:yes*	0.983	0.410	0.353
6	if *phys-freeze:no* then *politics:democrat*	0.988	0.568	0.352
7	if *el-salv-aid:no* then *contra-aid:yes*	0.986	0.478	0.332
8	if *phys-freeze: yes and mx-missile:no* then *el-salv-aid:yes*	0.994	0.355	0.330

Table 3: The 8 best rules from the congressional voting database

The algorithm was run with $s = 2$ (maximum rule-order of 2) in order to keep the output simple. The database consists of voting records from a 1984 session of the U.S. Congress. Each datum corresponds to a particular politician and the attributes correspond to the party affiliation of the voter plus 16 other attributes describing how they voted on particular budget issues such as aid to the Nicaraguan contra's, freezing physician's fees, aid to El Salvador, synthetic fuel funding, etc. Because of the probable imposition of party-line voting on many of the issues, this domain is characterized by very strong rules, i.e., predictive accuracies in the high 90% region. We can see from the table that there are redundancies, rules of near equal information content which have similar left-hand sides for the same right-hand side, differing perhaps by an extra term. An obvious extension of the algorithm is to refine this original rule-set by removing such redundancies — an initial such "rule-pruning" algorithm is described in Smyth et al. (1990a).

The ITRULE algorithm has been implemented in the C programming language on both Sun and Macintosh computers. We have run the algorithm on many of the other data sets which are publicly available in the U. C. Irvine database, Quinlan's chess end-game database (Quinlan, 1979), Sejnowski's text-to-phonemes database (Sejnowski and Rosenberg, 1987), and a variety of various character recognition problems. Various other projects, for both engineering and business applications, are currently underway. In general there is not much to be gleaned from asking how the algorithm performed in terms of rules produced on a particular data-set, since, by definition, the rules produced are the R most informative up to order s. More important is the question of how practically large can s be ? This is of importance if the structure of the dependencies is high-order, e.g., for certain types of Boolean functions such as parity. In practice when running the algorithm to look for discrete-time Markov chains the data vectors are created by successive

windowing of the time sequence system states. The size of this window effectively defines the maximum amount of memory we are able to model with the rules. If, as in the text-to-phoneme example mentioned earlier, the number of possible states at each time step is large, then practical considerations limit the amount of memory (maximum rule-order) we can look at. In general, however, for most applications there are no such constraints.

The algorithm is not directly suitable for domains characterized by continuous variables with regular functional relationships, e.g., polymomial relations between real-valued variables. However, this restriction results from the choice of hypothesis space (conjunctive rules) — the underlying information theoretic ideas should in principle be applicable to more general representations. Bridging the gap between continuous variables and symbolic representation techniques remains an open research issue, although, in practice, direct quantization of continuous variables (in an appropriate manner) rarely causes major problems. In addition, since the probability estimation procedure underlying the algorithm effectively assumes that the data is a true random sample, data sets which do not obey this assumption are not directly suitable for this technique, e.g., time-series data.

9.6 Future directions in learning from databases

It is worth making the general point that more cross-disciplinary research between computer scientists, information theorists and statisticians is needed. Statistics in particular must play a basic role in any endeavour which purports to infer knowledge from data. One might say that statistical models are a necessary but non-sufficient component of knowledge discovery. Historically, statistical theory has developed as a means for testing hypotheses in a controlled experiment scenario. The founding fathers of the field typically worked with pencil and paper with relatively small data sets where each datum was painstakingly collected in a well-characterized sampling methodology. Data was expensive and analysing it was a purely manual operation. In contrast, many domains at present are characterised by vast amounts of data which has been collected in a manner far removed from ideal random sampling techniques, and which can be analyzed in any number of ways in an automated manner. In essence the rules of the game have changed, and when applying statistical theories it is worth keeping in mind the original context in which they were developed.

It is interesting to note that early applications of computer algorithms in the 1960's in the statistical field led to controversy over whether or not such techniques

were in fact "fishing" for theories where none really existed (Selvin and Stuart, 1966). This is an especially important point where the number of attributes and the number of data samples are of the same order. Essentially, if one keeps applying different hypothesis tests to the same data set, it becomes more likely that one will accept a false hypothesis, i.e., confuse a random correlation with a true dependency. The solution is to make one's "hypothesis acceptance criteria" dependent on the number of hypotheses tested so far — however, this is extremely difficult to model in all but simple problems. This type of problem can be circumvented by having very large datasets but, nonetheless, its relevance to any knowledge discovery algorithm is apparent.

A problem which we have not discussed is that of incremental or "on-line" learning as opposed to "batch" learning, i.e., the ability to incorporate new data into the model without the need for re-running the entire induction algorithm or the need to store all the previous data. Various *ad hoc* schemes have been proposed in the machine learning or (more recently) the neural network literature. Typically these schemes fail on two accounts. Firstly they confuse parameter adaptation with model adaptation, i.e., they fine-tune the parameters of a particular model *without* considering the possibility of other models. Secondly, they fail in any even rudimentary manner to take into account what basic statistical theory has to say about estimation over time, e.g., the notion of stationarity. It is worth emphasizing that seeking *universal* incremental learning algorithms is probably ill-advised — the engineering approach of domain-specific solutions to particular problems seems more promising (see Buntine (1990) for a similar viewpoint). The implications for database discovery algorithms may be that taking into account the nature of the data and the manner in which it was collected will prove to be the most profitable avenue for exploration, rather than seeking generic, domain-independent algorithms.

Another major issue is that of prior knowledge. One of the paradigm shifts in machine learning in recent years has been away from the idea that a machine can acquire all knowledge starting from nothing, to a gradual realization that the machine can do much better in learning tasks with only a little (appropriate) prior knowledge. So far most of this theory-based learning work has been largely isolated from the type of quantitative probability-based methodologies we have presented here. The incorporation of prior knowledge is a non-trivial problem if we consider the statistical ramifications — an *a priori* domain theory corresponds to *a priori* assumptions or a statistical bias towards certain models. Despite what proponents of Bayesian inference may claim, getting accurate and consistent subjective prior estimates for *complex* hypothesis spaces (such as one has in a typical database) is

quite difficult and there is a dearth of practical literature and experience in this area.

As a final issue, while synthetic domains are useful for initial experimentation and comparison purposes, more work needs to be done with real databases. Typically, real databases will not consist of random samples and may contain missing and mixed-mode data. The treatment of missing data, for example, is again subject to various assumptions and may be domain dependent to a large extent. Prior work in statistical pattern recognition has addressed some of these topics (Dixon, 1979). Techniques such as these need to become established tools for learning and discovery algorithms.

9.7 Conclusion

We view the ITRULE algorithm's primary practical use to be that of an exploratory data analysis tool for discrete/categorical data, rather than a general purpose "wonder algorithm." Of more fundamental significance, than the algorithm itself, is the basic underlying *idea* of intensive hypothesis search guided by information theoretic principles as a paradigm for managing large volumes of data where we have limited prior knowledge. In this context, the work presented in this chapter, and indeed in this volume as a whole, will hopefully be viewed in retrospect as a small but important early step in the field of automated knowledge discovery.

9.8 Acknowledgements

This research was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.In addition this work is supported in part by Pacific Bell, and in part by the Army Research Office under Contract No. DAAL03-89-K-0126.

9.9 Appendix: Estimating probabilities from data

A necessary component of any statistical approach to rule induction is the ability to estimate probabilities accurately from data. The approach with which most people are familiar is the simple frequency ratio, i.e., if we count r occurrences of an event in a total sample of size n, we then estimate the probability of this event in general as the simple frequency ratio r/n. In statistical estimation theory this is known as the *maximum likelihood* estimate. For large values of n this estimate is

well-behaved, however, for small values of n it can cause problems.

Consider, for example, the case where a doctor arrives in a foriegn country for a temporary working assignment and, of the first three patients he examines, all have the same particular disease. How should the doctor estimate the probability p of the disease occurring among the general population ? Clearly the maximum likelihood estimate of $p = 3/3 = 1$ is over-pessimistic and highly unlikely to be true. A proponent of Bayesian estimation methods (see Berger (1985) for a comprehensive treatment) might argue that the doctor would have an *a priori* belief about the value of p (perhaps the value of p which he has estimated from experience in his own country), which is then updated to a new *a posteriori* value for p on the basis of the three new observations. A more conservative information theorist might argue that since this is a foreign country, the doctor has really no prior information, and hence a *maximum entropy* (ME) estimate is most appropriate — the technique of maximum entropy estimation was originally proposed by Jaynes (1968) and explicitly espouses the principle of adding no extraneous information to the problem. Hence, for m mutually exclusive and exhaustive events, the ME estimate of the probability of any event is $1/m$, since there is no initial information given to suggest that any one event is more likely than any other.

Naturally, one can view the Bayesian and ME estimation techniques as completely compatible, differing only in the credence given to initial information. In the medical example described above, the Bayesian technique would likely be the most practical and appropriate, given the difficulty in selecting the proper event space to construct an ME estimate. Given that selecting an initial estimate is not a problem in principle, the real issue becomes one of how to update this estimate in the light of new data. In a sense this is a problem in choosing an interpolation formula as a function of n (the sample size), where n ranges from 0 to ∞. At $n = 0$ our formula should give the initial Bayesian/ME estimate, and it should change smoothly as a function of increasing n, approaching the maximum likelihood estimate r/n as $n \rightarrow \infty$.

Such techniques exist in the statistical literature. In our work we have chosen to use the Beta distribution as described by I. J. Good in his 1964 monograph on point estimation (Good, 1964). Without going into the technical details, one effectively parametrises the Beta distribution to encode one's beliefs both about the expectation of the probability p for $n = 0$ (the initial Bayes estimate), and the degree of confidence in our estimate for p. The latter parameter controls the effective rate at which the Beta estimate changes from the prior value of p to the maximum likelihood estimate r/n, as a function of n. In Good's treatment one

chooses the parameters α and β such that

$$p_0 = \frac{\alpha}{\alpha + \beta}$$

where p_0 is one's initial estimate of p having seen no data, and $\alpha > 0, \beta > 0$. One's estimate for p, having seen r successes from n trials is then

$$\hat{p}(r, n) = \frac{\alpha + r}{\alpha + \beta + n}$$

Clearly, specifying p_0 only constrains the ratio of α and β — to solve for their actual values, Good further defines a second equation for an initial estimate of the variance of, or confidence in, p_0. We find the specification of an initial variance term somewhat non-intuitive and difficult to judge in practice. Instead we use the following approach, which is entirely equivalent to Good's approach (in that our estimate implicitly results in a prior variance term) except that it is more intuitive for practical use.

Let us define

$$k = \alpha + \beta$$

to be the "effective" sample size corresponding to our prior belief p_0, i.e., consider this to be number of samples by which we wish to weight our prior belief. Hence we can rewrite our estimator in the form of

$$\hat{p}(r, n) = \frac{r + kp_0}{n + k}$$

We have found this particular small sample estimator to be robust and easy to use in practice. In the ITRULE algorithm described earlier, one supplies the parameter k ($k > 0$) to the algorithm — choosing k large makes the algorithm more conservative, while k small (such as $k = 2$) makes it more liberal in inductive inference. For our purposes p_0 is chosen automatically by the algorithm depending on the context. Prior probabilities of simple events employ the ME technique of using $1/m$, whereas estimation of conditional probabilities use an equivalent ME technique where an initial estimate using the unconditional prior is chosen, i.e.,

$$p_0(x|y) = \hat{p}(x)$$

In general we have found that the use of these relatively simple estimation techniques make a considerable difference to the robustness of our algorithms.

9.10 References

J. B. Adams (1976), 'Probabilistic reasoning and certainty factors,' *Mathematical Biosciences*, **32**, 177–186.

D. Angluin and C. Smith (1984), 'Inductive inference: theory and methods,' *ACM Computing Surveys, 15(3)*, 237–270.

J. O. Berger (1985), *Statistical Decision Theory and Bayesian Analysis*, Springer-Verlag, New York.

W. Buntine (1990), 'Myths and legends: in learning classification rules,' *Proceedings of AAAI-90*, Morgan Kaufmann: San Mateo, CA.

P. Clark and T. Niblett (1989), 'The CN2 induction algorithm,' *Machine Learning*, vol.3, 261–283.

P. Cheeseman (1984), 'Learning of expert systems from data,' *First IEEE Conference on Applications of Artificial Intelligence*, IEEE Computer Society, Los Alamitos: CA.

A. N. Chomsky (1957), *Syntactic Structures*, Mouton Press, The Hague.

J. K. Dixon (1979), 'Pattern recognition with partly missing data,' *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC–9, no.10, 617–621.

B. R. Gaines and M. L. G. Shaw (1986), 'Induction of inference rules for expert systems,' *Fuzzy Sets and Systems*, **18** (3), 315–328.

I. J. Good (1965), *The estimation of probabilities: an essay on modern Bayesian methods*, Research monograph no.30, The MIT Press, Cambridge: MA.

R. M. Goodman, J. W. Miller, P. Smyth, and H. Latin (1989), 'Real-time autonomous expert systems in network management,' in *Integrated Network Management I*, B. Meandzija and J. Westcott (eds.), North Holland: Amsterdam, 588–624.

R. M. Goodman and P. Smyth (1988a), 'Decision tree design from a communication theory standpoint,' *IEEE Trans. Information Theory*, vol 34, no. 5, 979–994.

R. M. Goodman and P. Smyth (1988b), 'An information-theoretic model for rule-based expert systems,' presented at the 1988 International Symposium on Information Theory, Kobe, Japan.

R. M. Goodman and P. Smyth (1988c), 'Information-theoretic rule induction,' *Proceedings of the 1988 European Conference on Artificial Intelligence*, Pitman Publishing, London.

R. M. Goodman and P. Smyth (1989), 'The induction of probabilistic rule sets — the ITRULE algorithm,' *Proceedings of the Sixth International Workshop on Machine Learning*, Morgan Kaufmann: San Mateo, CA, 129–132.

R. M. Goodman and P. Smyth (1990), "Decision tree design using information theory," *Knowledge Acquisition*, vol.2, no.1, 1–19.

E. T. Jaynes (1957), 'Information Theory and Statistical Mechanics I,' *Phys. Rev.* 106, 620–630.

J. H. Holland, K. J. Holyoak, R. E. Nisbett, P. R. Thagard (1986), *Induction: Processes of Inference, Learning and Discovery*, The MIT Press, Cambridge: MA.

R. S. Michalski, I. Mozetic, and J. R. Hong, 'The multi-purpose incremental learning system AQ15 and its testing application to three medical domains,' *Proceedings of the 1986 AAAI Conference*, Morgan Kaufmann: San Mateo, CA.

A. Newell and H. A. Simon (1972), *Human Problem Solving*, Prentice Hall, Englewood Cliffs: NJ.

G. Piatetsky-Shapiro (1990), 'Discovery of strong rules in databases,' this volume.

J. R. Quinlan (1979), 'Discovering rules by induction from large collections of examples,' in *Expert Systems in the Micro-electronic Age*, ed. D. Michie, Edinburgh University Press, Edinburgh.

J. R. Quinlan (1986), 'Induction of decision trees,' *Machine Learning*, vol. 1, 81–106.

J. R. Quinlan (1987), 'Generating production rules from examples,' *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann: San Mateo, CA, 304–307.

J. R. Quinlan and R. L. Rivest (1989), "Inferring decision trees using the minimum description length principle," *Information and Computation*, vol.80, 227–248.

J. Rissanen (1989), *Stochastic Complexity in Statistical Inquiry*, World Scientific Publishing: Singapore.

J. C. Schlimmer (1987), *Concept Acquisition through Representational Adjustment*, Ph.D. thesis, Department of Computer Science, University of California at Irvine, CA.

T. J. Sejnowski and C. M. Rosenberg (1987), 'Parallel networks that learn to pronounce English text,' *Complex Systems*, 1:145–168.

H. C. Selvin and A. Stuart (1966), 'Data-dredging procedures in survey analysis,' *American Statistician*, 20(3), pp.20–23.

C. Shannon (1948), 'A mathematical theory of communication,' *Bell System Technical Journal*, vol.27, no.3, 379–423.

J. E. Shore and R. W. Johnson (1980), 'Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy,' *IEEE Trans. Inform. Theory*, **IT–26** (1), 26–37.

P. Smyth and R. M. Goodman (1990a), 'The information content of a probabilistic rule,' submitted for publication.

P. Smyth and R. M. Goodman (1990b), 'An information theoretic approach to rule induction from databases,' to appear in *IEEE Transactions on Knowledge and Data Engineering.*

P. Smyth, R. M. Goodman and C. Higgins (1990a), 'A hybrid rule-based/Bayesian classifier,' in *Proceedings of the 1990 European Conference on Artificial Intelligence*, Pitman Publishing, London.

P. Smyth, J. Statman, and G. Oliver (1990b), "Combining knowledge-based techniques and simulation with applications to communications network management," submitted to IEEE Globecom Conference, 1990.

L. A. Zadeh (1965), 'Fuzzy sets,' *Information and Control*, **8**, 338–353.