Conf-920538--16

Full paper to be presented at the 8th Power Plant Dynamics, Control and Testing Symposium, May 27-29, 1992, Knoxville, Tennessee

ANL/CP--74367

DE92 011843

Comparison of Two Inductive Learning Methods: A Case Study in Failed Fuel Identification*

J. Reifman[1] and J. C. Lee[2]

[1]Reactor Analysis Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, Illinois 60439

[2]Department of Nuclear Engineering
University of Michigan
Ann Arbor, Michigan 48109

*Work supported by the U. S. Department of Energy, Nuclear Energy Programs under Contract W-31-109-ENG-38.

COMPARISON OF TWO INDUCTIVE LEARNING METHODS: A CASE STUDY IN FAILED FUEL IDENTIFICATION

Jaques Reifman
Reactor Analysis Division
Argonne National Laboratory
Argonne, Illinois 60439, U.S.A.

John C. Lee
Department of Nuclear Engineering
University of Michigan
Ann Arbor, Michigan 48109, U.S.A.

ABSTRACT

Two inductive learning methods, the ID3 and Rg algorithms, are studied as a means for systematically and automatically constructing the knowledge base of expert systems. Both inductive learning methods are general-purpose and use information entropy as a discriminatory measure in order to group objects of a common class. ID3 constructs a knowledge base by building decision trees that discriminate objects of a data set as a function of their class. Rg constructs a knowledge base by grouping objects of the same class into patterns or clusters. The two inductive learning methods are applied to the construction of a knowledge base for failed fuel identification in the Experimental Breeder Reactor II. Through analysis of the knowledge bases generated, the ID3 and Rg algorithms are compared for their knowledge representation, data overfitting, feature space partition, feature selection, and search procedure.

INTRODUCTION

Learning by induction or from examples has been shown[1-9] to be an effective method of knowledge acquisition for expert systems. Induction is the process by which structures or regularities underlying a finite number of examples are discovered through analysis of the examples themselves. Given a finite set of examples representative of a problem domain, inductive learning programs automatically extract information from the examples and derive "general rules" that describe the given examples. Inductive learning programs provide an alternative method for the tedious and time-consuming process of acquiring and encoding expert's knowledge.

Two inductive learning methods, the Iterative Dichotomizer 3 (ID3) algorithm,[10] and the entropy minimax algorithm,[11] have been used to automatically construct the knowledge base of expert systems for nuclear engineering applications. The first application of the ID3 algorithm in nuclear engineering was made to construct the knowledge base of an expert system for diagnosing clad rupture in the reactor core of the Fast Flux Test Facility.[2] Other applications of the ID3 algorithm include the construction of knowledge bases for expert systems dealing with contaminated waste sites,[3] failed fuel identification,[5] and solidification of radioactive liquid waste.[6] These applications used commercialized versions of the ID3 algorithm which are marketed as a means for automatic construction of the knowledge base of the accompanying expert system

shell.[8,9] The entropy minimax algorithm was first applied in nuclear engineering to construct a knowledge base for fuel clad failure diagnosis in light water reactors.[12] Also based on the entropy minimax algorithm, the Rule Generation (Rg) program[4] was developed and used to automatically construct a knowledge base for diagnosing transient events in pressurized water reactors. Since inductive learning methods extract knowledge from a data set of examples, all these applications require that representative examples, from actual plant data or obtained through simulation models, be compiled prior to the construction of the knowledge base.

In this paper we compare the ID3 and Rg inductive methods. The two methods are applied to the construction of a knowledge base for failed fuel identification in the Experimental Breeder Reactor II (EBR-II). Through analysis of the knowledge bases generated, the ID3 and Rg algorithms are compared for their knowledge representation, data overfitting, feature space partition, feature selection, and search procedure.

INDUCTIVE LEARNING AND CLASSIFICATION

The discovery of classification patterns in a collection of objects or examples can be considered as an inductive learning process. Given a collection of objects described by their class and corresponding set of characteristics, classification patterns representing the relationships between each class of objects and their characteristics can be inductively extracted by finding the characteristics that group objects of a common class. Therefore, the discovery of classification patterns from a finite number of particular objects is equivalent to the derivation of general rules for classifying objects based on their characteristics. Since the objective of inductive systems is to classify objects that are not present in the collection of objects used to construct the classification rules, such systems should construct rules that are not too much geared to the initial collection of objects. In other words, the inductive system should not "overfit" the data and should avoid constructing contrived rules that characterize the existing data extremely well but may not be realistic.

The process of inductive learning and classification can then be used to model an object's class as a function of its characteristics. Such approach is desirable for representing the relationships between dependent and independent variables in areas where there is a lack of well-understood models. This is certainly the case of failed fuel identification where the relationships between the classes or failure types E_k (k=1,2,...,K), treated as dependent variables, and its characteristics or features F_j (j=1,2,...,J), treated as independent variables, are not reliably predicted by mechanistic fuel performance and failure models.[12] Furthermore, these models represent information from the failure type E_k to the feature F_j, i.e., in the causal direction, while in order to identify a failed fuel one needs to represent information from the observable features F_j to the unobservable failure types E_k, i.e., in the usage direction.

In the case of failed fuel identification, the failure types E_k (k=1,2,...,K) are discrete events such as E_1 = failure in the column region of metal fuel, E_2 = failure in the dimple region of metal fuel, and E_3 = failure in the welding of metal fuel. The features F_j (j=1,2,...,J) used to characterize failures E_k are either qualitative or quantitative and have discrete values F_{ji} (i=1,2,...,I). For instance, the qualitative feature F_1 = slope of the fraction of fission gas (FG) released out of the fuel, has four discrete values: F_{11} = erratic, F_{12} = sharp, F_{13} = gradual, and F_{14} = burp. In contrast, the quantitative feature F_2 = magnitude of the longest spike in ^{135m}Xe activity, could have two "discrete" intervals: F_{21} = [0, 10) and F_{22} = (10, 400]. Quantitative

features should be partitioned into discrete feature intervals where each interval is expected to represent the same qualitative value.

Inductive learning approaches have a number of important properties. Because information is extracted from a data set of examples, the data set needs to be as complete and validated as possible. Although the "learned" rules are supposed to be quite general they cannot extrapolate much beyond the data set used in their generation. For instance, if a given fuel failure class E_k is not present in the data set, the inductively generated knowledge base will not be able to identify E_k. Furthermore, the features F_j need to be defined as input to the inductive program. If a good set of features is defined, the induction problem is simplified and the obtained knowledge base is simple and compact. Inductively constructed knowledge bases are also logically consistent and complete,[13] and represent information in the readily useful direction, i.e., from observable features F_j to unobservable classes E_k. Finally, there is not a unique way to group objects of a common class, except for trivial cases, and different inductive learning systems will produce, in general, different classification rules. The grouping of objects, among other factors, is a function of the discriminatory measure used by the inductive system.

INFORMATION ENTROPY

Shannon and Weaver's[14] information-theoretic entropy S(E|X) is used in both the ID3 and Rg algorithms as a discriminatory measure to partition a data set of objects such that objects of a common class tend to be grouped together. The entropy S(E|X) can be interpreted as the expected value of the *excess* amount of information we would gain from learning the class E_k of an object *above* the amount of information gained by knowing its properties. Hence, by partitioning the data set as a function of the object's properties such that entropy

$$S(E|X) = -\sum_{i=1}^{I} p(X_i) \sum_{k=1}^{K} p(E_k|X_i) \ln p(E_k|X_i), \qquad (1)$$

is minimized, we would be extracting maximum information from the data set. Here, $p(X_i)$ is the marginal probability of objects of any class having property X_i and $p(E_k|X_i)$ is the conditional probability that objects with property X_i will belong to class E_k. These probabilities are calculated based on the number of objects in the data set.

In the ID3 algorithm, the set X in Eq. (1) corresponds to each one of the features F_j (j=1,2,...,J) which has discrete values F_{ji} (i=1,2,...,I), while in the Rg algorithm X corresponds to a set C of clusters C_i (i=1,2,...,I) which represents a combination of selected values F_{ji} of the features. Hence, in the ID3 algorithm Eq. (1) takes the form

$$S(E|F_j) = -\sum_{i=1}^{I} p(F_{ji}) \sum_{k=1}^{K} p(E_k|F_{ji}) \ln p(E_k|F_{ji}), \qquad (2)$$

by taking $X = F_j$ and $X_i = F_{ji}$. Similarly, by taking X = C and $X_i = C_i$ in Eq. (1) we obtain the entropy S(E|C) for the Rg algorithm

$$S(E|C) = -\sum_{i=1}^{I} p(C_i) \sum_{k=1}^{K} p(E_k|C_i) \ln p(E_k|C_i). \qquad (3)$$

Comparison of Eqs. (2) and (3) allows us to clarify the main differences between the two algorithms. Entropy S(E|C) of Eq. (3) simultaneously selects the most discriminatory set of feature values, i.e., cluster C_i in Rg, while entropy $S(E|F_j)$ of Eq. (2) uses a stepwise approach selecting the most discriminatory feature F_j at each step of the ID3 algorithm.

THE ID3 ALGORITHM

The ID3 algorithm,[10] which is a descendent of Hunt's Concept Learning System,[15] models the relationships between an object's class and its features, i.e., characteristics, by building decision trees that discriminate the objects of a data set as a function of their class. ID3 builds a decision tree by first scanning the data set and choosing the most informative feature F_j, i.e., the feature that can best classify the objects according to their class, as the root node of the decision tree. This is done by obtaining feature F_j that minimizes entropy $S(E|F_j)$ in Eq. (2) in the formation of the decision tree. The root node branches out into I branches corresponding to the discrete values F_{ji} (i=1,2,...,I) of F_j. The objects of the data set are then sorted through the I branches dividing the data set into I subsets, according to their F_j value. Figure 1 illustrates this procedure, for the case of the EBR-II failed fuel identification problem, where feature F_1 = slope of the fraction of FG released out of the fuel, was selected as the root node of the decision tree from a data set of 43 objects each having one of 8 possible classes E_k. The class distribution of the 43 objects is indicated by the 8 numbers inside the brackets. The four branches of the first node (erratic, sharp, gradual, and burp) are also shown together with their corresponding subsets of the original data set. For instance, there are 10 objects of class E_1 in the data set and all 10 objects have feature value F_{11}= erratic for feature F_1.

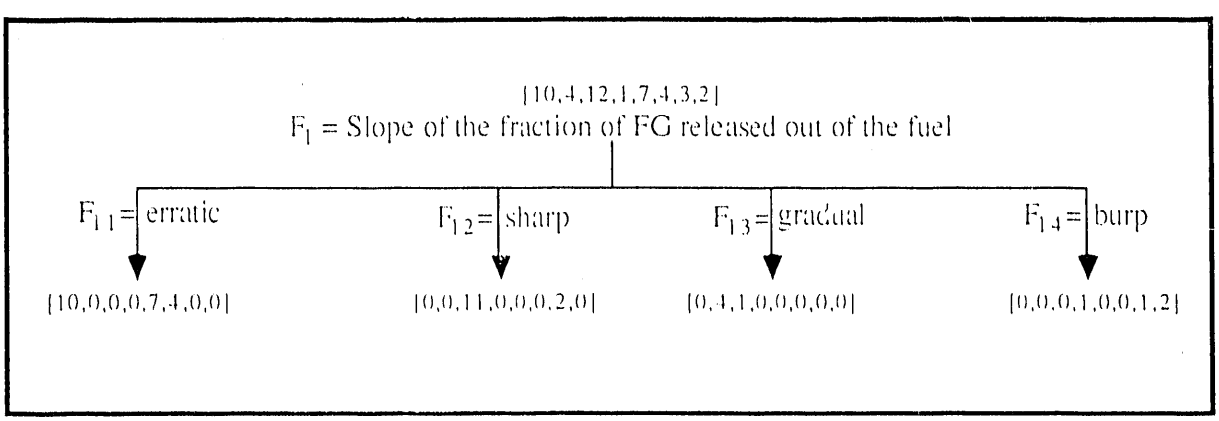

Fig. 1. Formation of a Decision Tree Using the ID3 Algorithm.

The remaining nodes of the tree are obtained by repeating the process, for each branch, with the corresponding subsets of the data set, such that the subset at each node of the tree will be "purer" than the parent subset. By choosing F_j at each node that minimizes $S(E|F_j)$ in Eq. (2), based on each node's subset of the original data set, we would be forming the most discriminatory and hence the minimal size tree. This process continues until each branch has no objects or all its objects have the same class. The objects corresponding to the last level or terminal node of each branch, i.e., the subset where all objects have a common class, form the leaves of the decision

tree. A key step in building a decision tree is the selection of discriminatory tree nodes, i.e., minimum entropy features, since a series of bad choices will result in a tree with a leaf for each object of the training set.

Once the decision tree is built, a new object with known feature values and unknown class is classified by starting at the root node of the decision tree, finding the value of the root node feature in the given object, taking the branch appropriate to that value, and continuing in the same fashion until a leaf is reached. Using a decision tree to classify an object is equivalent to using a set of "if (condition) then (consequence)" production rules where the condition part of the rule is formed by the conjunction, i.e., a logical AND, of all discrete feature values of the tree defining the path from the root node to each one of its leaves, and the consequence part is the class of the objects at each leaf. Hence, a decision tree can be represented by a set of production rules.

THE Rg ALGORITHM

The Rg algorithm,[4] which is a descendent of Christensen's Entropy Minimax approach for pattern recognition,[11] models the relationships between an object's class and its features by forming an N-dimensional feature-space populated with the objects of the data set and partitioning the entire feature space into I "optimal" patterns or clusters C_i (i=1,2,...,I) such that objects of a common class are located in the same cluster. The N-dimensional feature space is formed by selecting the N "best" features of the data set and using each of the N features $F_1,F_2,...,F_N$ as an axis. Optimal patterns maximize the classification information extracted from the data set such that each subspace or cluster C_i in feature space is closely associated with only one class E_k. In order to handle the "curse of dimensionality," typical of searching procedures in multidimensional spaces, the Rg program makes approximations in the three major steps of the entropy minimax algorithm, the partition of the feature space, the selection of the N best features, and the discovery of patterns, which are described in the following paragraphs and is illustrated in Fig. 2.

Partition of the Feature Space

In step 1, the simultaneous partition of the N'-dimensional feature space, where N' is the total number of quantitative features, is approximated by N' independent one-dimensional partitions. Partitioning of each of the N' features F_j into a maximum of four intervals F_{ji} (i=1,...,≤4) is obtained by projecting all data points of the data set onto each feature axis F_j and finding three or less cuts in F_j that minimize entropy $S(E|F_j)$ in Eq. (2). Through this approach, the initial N' quantitative measurements or feature variables are discretized into at most 4N' feature intervals which are then added to the already discretized qualitative features to form a total set of N'' feature intervals. Thus, in step 1, by decomposing the partition operation of the quantitative features into N' independent operations, the number of possibilities is greatly reduced and yet good partitions of the feature space can be obtained as long as the objects are reasonably well separated.

Feature Selection

The goal of automatic feature selection in step 2 is to select and retain a subset of N salient features, $F_1,F_2,...,F_N$, from the initial N'' features such that the process of pattern discovery is implemented in a vastly reduced feature space without degrading its performance. The underlying philosophy in the selection of N key salient features is two-fold: elimination of features that are interrelated and hence do not contribute additional information, and retention of features that can clearly characterize or discriminate each class from the remaining classes.

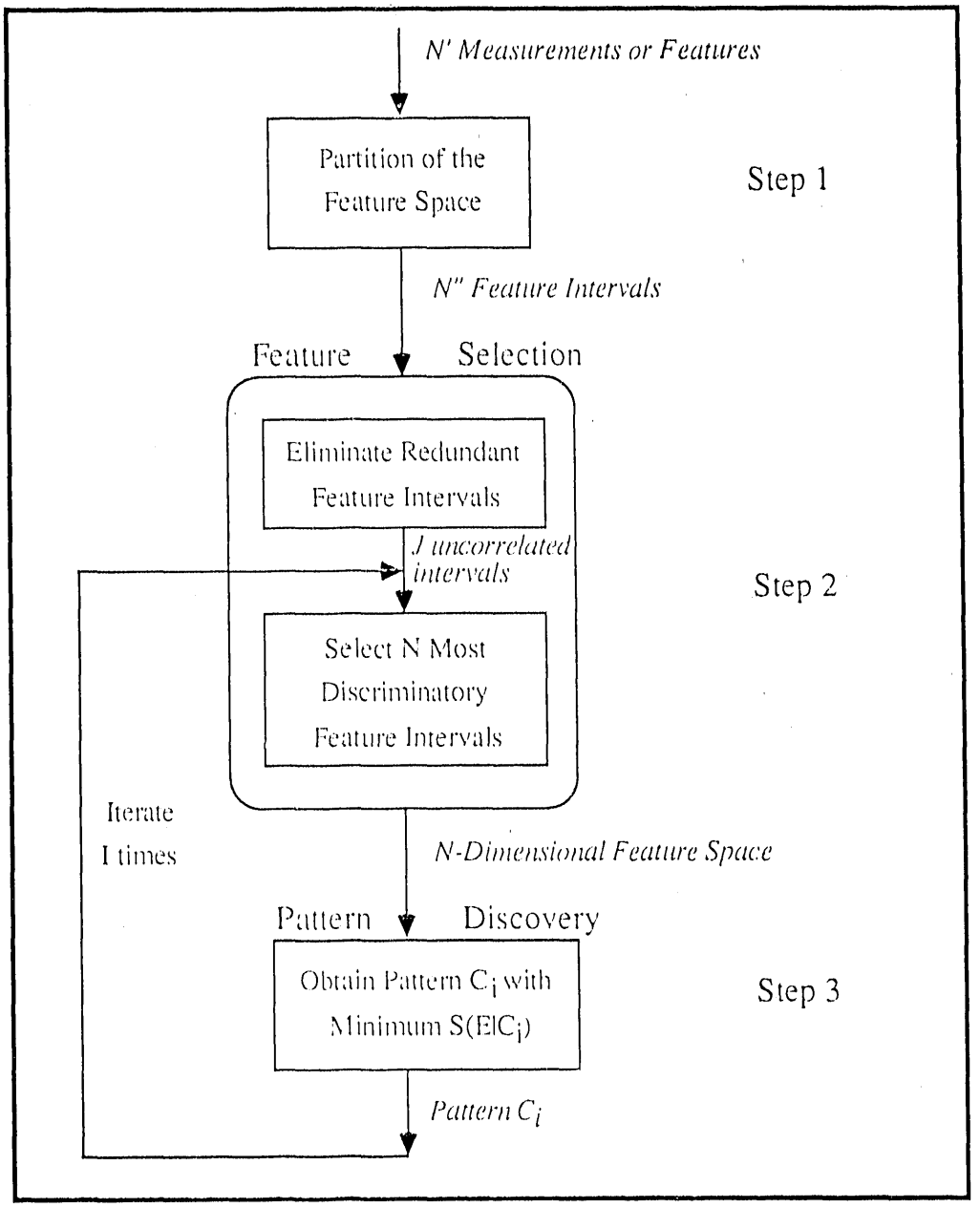

Fig. 2. Flow Chart of the Three Major Steps of the Rg Program.

Feature intervals that share common information with another feature interval are redundant and should not be selected. Including redundant feature intervals would only increase the dimensionality of the feature space, where patterns are to be found, with no extra contribution. This redundancy check is made by calculating the linear correlation coefficient between each pair of feature intervals and keeping only one of two or more feature intervals when their linear correlation coefficient is beyond a specified threshold. After the correlated feature intervals have been removed from the initial list of possible feature intervals, the remaining J uncorrelated intervals are ranked based on their discriminatory power. The discriminatory power for each feature interval F_{ji} of the set of J intervals is measured through the inverse of entropy $S(E|F_{ji})$

$$S(E|F_{ji}) = -\sum_{k=1}^{K} p(E_k|F_{ji}) \ln p(E_k|F_{ji}), \qquad (4)$$

where $p(E_k|F_{ji})$ is the conditional probability that objects with value F_{ji} of feature F_j will belong to class E_k. In order to account for the discriminatory power of each feature interval in conjunction with the other features, $S(E|F_{ji})$ is calculated with F_{ji} represented by the intersection of feature interval F_{ji} with all J feature intervals. The N features corresponding to the N most discriminatory feature intervals are used to form the N-dimensional feature space where one M-dimensional ($M \leq N$) cluster C_i is to be found in step 3. To find the entire set of clusters C_i (i=1,2,...,I), step 3 is iteratively used I times.

Pattern Discovery

A simultaneous discovery of I clusters C_i (i=1,2,...,I) that minimizes the global entropy S(E|C) of Eq. (3) requires a complete search over all possible configuration of clusters in the partitioned feature space. Since the number of possible configurations grows exponentially with N, an exhaustive search becomes infeasible for any realistic value of N. In the Rg program, the simultaneous minimization of S(E|C) is approximated by a sequential discovery of clusters C_i, one at a time, which greatly reduces the computation requirements. At the *i-th* step of such stepwise approximation, a cluster C_i is discovered with minimum local entropy $S(E|C_i)$

$$S(E|C_i) = -\sum_{k=1}^{K} p(E_k|C_i) \ln p(E_k|C_i). \qquad (5)$$

The stepwise approximation is necessary but still not sufficient for eliminating the "curse of dimensionality." The number of possible clusters is still very high for any reasonable value of N. Therefore, the Rg program restricts the type of logical propositions by which a pattern can be formed. Patterns are restricted to univariate or multivariate intersections of the selected feature intervals forming the N-dimensional feature space.

Each one of the patterns C_i discovered by the Rg program can again be represented by a "if (condition), then (consequence) $p(E_K|C_i)$" rule R_i. The condition part of rule R_i corresponds to the location in feature space of cluster C_i, i.e., univariate or multivariate interactions of feature intervals F_{ji}, and the consequence part of the rule corresponds to the probability distribution $p(E_K|C_i)$ of the classes of the objects located inside cluster C_i. Good discriminatory rules are formed by a sharp distribution of $p(E_K|C_i)$ over only one class E_K.

CLASSIFICATION OF FAILED FUEL

The classification of fuel failure events at EBR-II was selected to compare the ID3 and the Rg inductive approaches for automatic knowledge base construction. Fuel failure events that occurred at EBR-II between May 1986 and December 1990 were used to form a data set of 43 objects[16] where each object, i.e., fuel failure event, has one of 8 possible classes (E_1,...,E_8) and a set of 7 features (F_1,...,F_7). Each fuel failure class E_k represents the type of breached fuel element, e.g., metal or mixed oxide, and the location of breach along the fuel pin, e.g., column region, dimple region, plenum region. Table I describes the 8 classes of fuel failure along with the distribution of the 43 failures in each class. The 7 features chosen to characterize the 8 fuel failure types represent directly monitored and calculated characteristics of fission product activities, FG released, and

delayed-neutron (DN) signals. As illustrated in Table II, three features, F_1, F_5, and F_7, are qualitative while the other four are quantitative. Using predefined criteria, the two inductive approaches automatically partition the total range of the quantitative feature values shown in Table II into feature intervals representing "discrete" feature values.

Table I. Classes of Failed Fuel and Number of Occurrences at EBR-II.

Number of Occurrences	Class of Failed Fuel
10	E_1 = Failure in the column region of metal fuel
4	E_2 = Failure in the dimple region of metal fuel
12	E_3 = Failure in the welding of metal fuel
1	E_4 = Multiple failures. Metal fuel failure preceded by mixed oxide failure
7	E_5 = Fresh failure in the column region of mixed oxide fuel
4	E_6 = Multiple failures. Failure in the column region of mixed oxide fuel combined with another mixed oxide fuel failure
3	E_7 = Failure in the plenum region of mixed oxide fuel
2	E_8 = Previous failure in the column region of mixed oxide fuel

The decision tree[5,16] obtained by applying the ID3 algorithm to the failed fuel data set is duplicated in Fig. 3. The decision tree uses all 7 features and discriminates the 8 failure classes such that each one of the 12 leaves characterizes only one class, even in the case where there is only one object per leaf. The 8 numbers inside the brackets of each node indicate the number of objects in each of the 8 classes E_k. For instance, the 8 numbers inside the bracket of the fifth leaf from the left, [0,0,11,0,0,0,0,0], indicate that there are 11 objects of class E_3 = failure in the welding of metal fuel and no objects of other classes in this leaf. Also note that the paths from the root node to the 12 leaves can be represented by 12 classification rules, $L_1, L_2, \ldots, L_{12}$. For example, the path from the root node to the fifth leaf can be represented by L_5

if (F_1 = sharp and $F_2 < 10$)

then (E_3 has occurred).

Table II. Features and Feature Values Used for Classifying Failed Fuel at EBR-II.

Features	Values
F_1 = Slope of the fraction of fission gas released out of the fuel	F_{11} = erratic F_{12} = sharp F_{13} = gradual F_{14} = burp
F_2 = Magnitude of largest spike in ^{135m}Xe activity	[0, 400]
F_3 = Increase in the level of delayed neutron signal	[0, 2000]
F_4 = Magnitude of largest spike in ^{87}Kr activity	[0, 1400]
F_5 = Behavior of delayed neutron signal after breach	F_{51} = normal F_{52} = step increase F_{53} = spikes F_{54} = sharp spike
F_6 = Maximum ^{133}Xe activity level	[0, 35000]
F_7 = Existence of delayed neutron signal at background level prior to fuel breach	F_{71} = no F_{72} = yes

The 7 classification rules generated by applying the Rg algorithm to the failed fuel data set are illustrated in Table III. For each rule R_i a cluster C_i is formed by an intersection of the feature intervals in the second column of the table which corresponds to the condition part of an if (condition) then (consequence) rule. The third column describing the number of objects and the probabilities $p(E_k|C_i)$ for each of the 8 classes E_k represents the consequence part of the rule. The symbol ε is used to denote a probability $p(E_k|C_i)$ value smaller or equal than 0.10. The probabilities $p(E_k|C_i)$ are estimated based on both the observational data inside the clusters and prior experiences not explicitly included in the current data set[4,11]

$$p(E_k|C_i) = \frac{M_k + W_k}{M + W}, \qquad (6)$$

where

M_k = number of objects E_k in cluster C_i,

$M = \sum_{k=1}^{K} M_k$ = total number of objects in cluster C_i,

W_k = prior weight associated with E_k, and

$W = \sum_{k=1}^{K} W_k$ = total prior weight associated with the entire M events.

Hence, the probabilities $p(E_k|C_i)$ for rules or clusters where all objects are of the same class are not equal to 1.0 which would be the case with null prior weights. Furthermore, the probabilities in Table III, calculated with W_k =1, are not sharply distributed because of the small number of objects M_k in each cluster and the selection of a relatively large total prior weight W = K = 8.

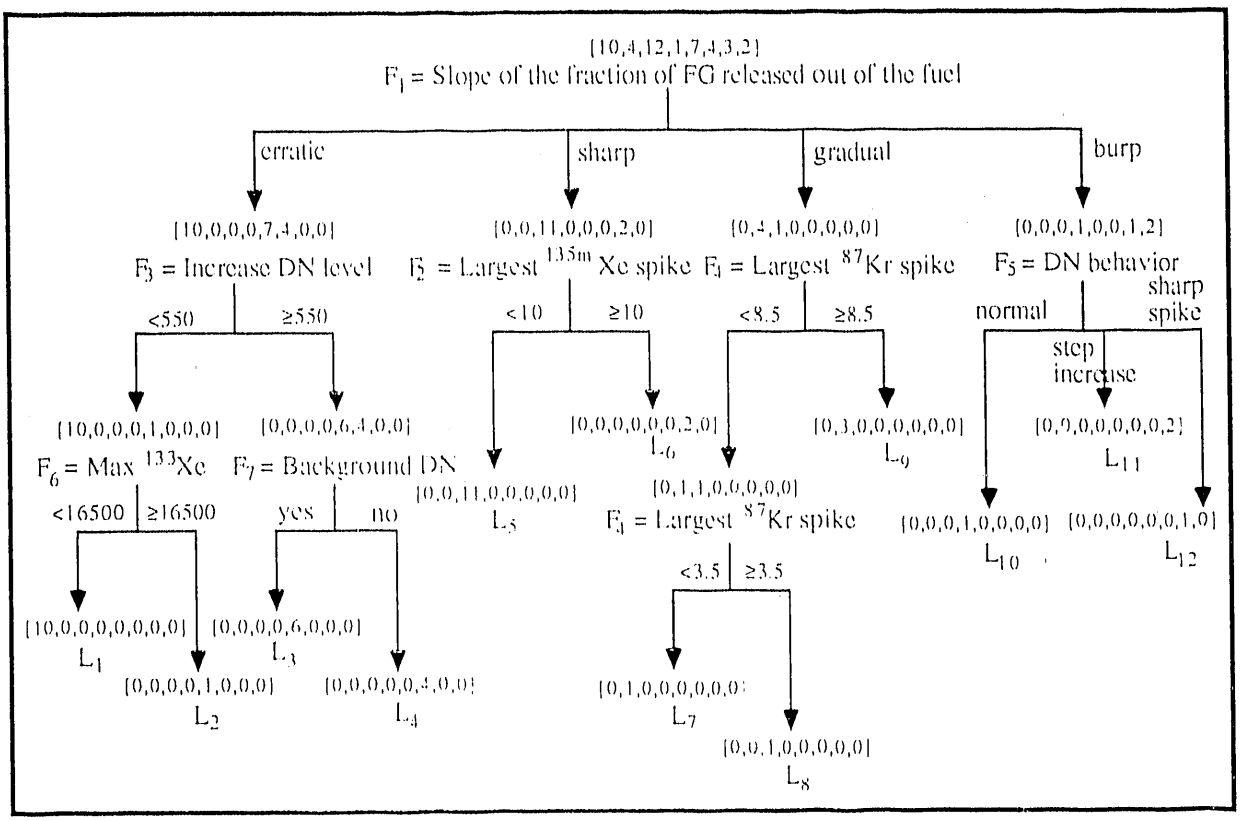

Fig. 3 Decision Tree for Failed Fuel at EBR-II Using the ID3 Algorithm.

The rules in Table III identify 7 of the 8 failed fuel classes, failure class E_4 is not characterized by the program since there is only one object with that class in the data set. All rules are characterized by either 2 or 3 features and unlike the ID3 results feature F_1 is not used to characterize every single rule, and features F_4 and F_5 are not sufficiently discriminating and hence not used in any of the 7 rules. Also in contrast with the ID3 results, the rules generated by Rg are not always pure; the rules represent more than one class, as indicated by rules R_5 and R_6. Furthermore, not all objects of the data set are used to construct the rules. One object of class E_5 is not found similar to the remaining 6 objects of class E_5 and is not used in rule R_3. Two rules, R_1 and R_2, corresponding to failure classes E_3 and E_1, respectively, are identical to the ID3 rules L_5 and L_1, respectively. Most of the other rules are different from the ID3 results by only one feature. A comparison of the rules obtained by the ID3 and Rg algorithms is illustrated in Table IV. The table shows the equivalence between the 12 ID3 rules, $L_1,...,L_{12}$, and the 7 Rg rules, $R_1,...,R_7$, based on the 8 classes they represent.

Table III. Identification Rules for Failed Fuel at EBR-II Using the Rg Algorithm.

Rule	Features	Classes and Probabilities
R_1	F_1 = sharp and $0 < F_2 < 0.01$	[0,0,11,0,0,0,0,0] [ε,ε,0.63,ε,ε,ε,ε,ε]
R_2	F_1 = erratic and $0 < F_3 < 500$ and $0 < F_6 < 14000$	[10,0,0,0,0,0,0,0] [0.61,ε,ε,ε,ε,ε,ε,ε]
R_3	$5 < F_2 < 400$ and $600 < F_3 < 2500$ and F_7 = yes	[0,0,0,0,6,0,0,0] [ε,ε,ε,ε,0.5,ε,ε,ε]
R_4	$5 < F_2 < 400$ and $600 < F_3 < 2500$ and F_7 = no	[0,0,0,0,0,4,0,0] [ε,ε,ε,ε,ε,0.42,ε,ε]
R_5	F_1 = gradual and $0 < F_3 < 500$	[0,4,1,0,0,0,0,0] [ε,0.38,0.15,ε,ε,ε,ε,ε]
R_6	F_1 = burp and $0 < F_2 < 0.01$ and F_7 = yes	[0,0,0,0,0,0,1,2] [ε,ε,ε,ε,ε,ε,0.18,0.27]
R_7	F_1 = sharp and $5 < F_2 < 400$ and $0 < F_3 < 500$	[0,0,0,0,0,0,2,0] [ε,ε,ε,ε,ε,ε,0.3,ε]

Table IV. Comparison of the ID3 and Rg Production Rules.

Inductive Method	Equivalent Rules										
ID3	L_1	L_2	L_3	L_4	L_5	L_6	L_7	L_8	L_9	L_{10}	L_{11} L_{12}
Rg	R_2	-	R_3	R_4	R_1	R_7	R_5			-	R_6

COMPARISON OF THE TWO METHODS

Both the ID3 and Rg inductive algorithms generated concise classification rules that are described by only two or three features. The conciseness of the rules is a consequence of well defined feature variables and the high classification power of information entropy used as a discriminatory measure in both approaches. Although at first glance the rules generated by the ID3 and the Rg approaches seem very similar, there are a number of underlying differences between these two inductive learning approaches. In this section, the advantages and disadvantages of the methods are compared through analysis of their knowledge representation, data overfitting, feature space partition, feature selection, and search procedure.

Knowledge Representation

The ID3 algorithm acquires knowledge from a data set of examples by constructing decision trees such that each leaf represents objects of only one class, even if the leaf has only one object. For instance, the second leaf in Fig. 3 has only one object of class E_5. The information of the decision tree can be represented by non-overlapping if (condition) then (consequence) production rules where the condition part of the rules is represented by the features used to form the paths between the root node and the leaves of the tree, while the consequence part of each rule is represented by one of the possible classes E_k (k=1,2,...,K) of the leaf. The representation of the consequence part of the rule is equivalent to setting $p(E_k|C_i) = \delta_{km}$, where δ_{km} is the Kronecker delta. The Rg algorithm acquires knowledge from a data set by discovering non-overlapping patterns in a feature space populated with objects of the data set such that events of a common class tend to be located in the same pattern. The information embedded in the patterns are also represented by non-overlapping production rules. The condition part of the rule is represented by the features used to form the boundaries of the pattern and the consequence part of each rule is represented by the probability distribution $p(E_k|C_i)$ over the K possible classes based on the objects that fall inside each pattern and prior experiences. Hence, the Rg inference mechanism offers the probability $p(E_k|C_i)$ as a measure of confidence of its inference based on the objects included in each pattern. ID3 does not provide such useful information.

Data Overfitting

An important property of any inductive learning approach is the capability of the algorithm to construct general rules that are able to predict the class of objects not used in its construction. To achieve such property, the construction of the rules should not be too much geared to the data set, i.e., the data should not be overfit. This property is accomplished in the Rg program by restricting the type of logical propositions by which a pattern can be found to univariate and multivariate intersections of selected features. The rules constructed by the Rg program are general and tend to represent the global behavior of the data. On the other hand, the ID3 algorithm does not have, in general, a provision to restrict the formation of contrived rules and can form rules too much geared to the data set. For instance, in Fig. 3 the seventh leaf has one object of class E_2, the eighth leaf has one object of class E_3, and the ninth leaf has three objects of E_2. In contrast, Rg constructs only one global rule, R_5 in Tables III and IV, to represent these five events. A similar situation occurs for the eleventh and twelfth leaves in Fig. 3, which correspond to rule R_6 in Tables III and IV. Provisions could be added to the ID3 algorithm by defining stopping rules or pruning the decision tree[17] to achieve general inference rules that are not contrived.

Partition of the Feature Space

The feature space must be partitioned before a feature is selected, for the ID3 algorithm, or a set of features is selected, for the Rg algorithm. Qualitative features are already partitioned and require no further manipulation. In contrast, quantitative features representing continuous variables need to be properly partitioned into ranges or feature intervals. Since patterns in the Rg algorithm are discovered by searching a multidimensional feature space, ideally the N' quantitative features should be partitioned simultaneously. As described in a previous section, the simultaneous partition is computationally infeasible and instead the partitioned is approximated by N' one-dimensional partitions using the entire data set. In the commercial implementations of the ID3 algorithm, quantitative features are also partitioned, one at a time, as features are selected to form a node.[8,9] However, unlike the Rg algorithm, here a feature is partitioned based only on the subset of the initial data set of the corresponding node. This partitioning approach causes the partitions to be very specialized resulting in overfitted or contrived rules. For instance, the three rules, L_7, L_8, and L_9, formed by the paths from the root node to the seventh, eighth, and ninth

terminal nodes in Fig. 3, respectively, partition the same feature F_4 twice making it more and more specific as subsequent partitions are performed. One approach to eliminate this overfitting would be to partition each feature based on the entire data set before any feature is selected as in Rg. However, such a procedure could cause two objects belonging to different classes to have identical feature values which cannot be handled by the ID3 algorithm.

Feature Selection

Feature selection is performed in the ID3 algorithm by choosing the most informative feature, at each node, that best discriminates the objects of the node according to their class. As the nodes of the tree are built, feature F_j that minimizes entropy $S(E|F_j)$ is chosen, one at a time, and used in the formation of the tree. Hence, ID3 approximates the formation of the most discriminatory rule with M features by choosing the M most discriminatory features one at a time, based on a different subset of the original data set at each time. Feature selection is performed in the Rg algorithm through a two-step approach. First, the entire data set is used to eliminate redundant features and then a set of features is selected in the reverse order of their entropy $S(E|F_{ji})$, calculated either as an individual feature interval or as a pair of feature intervals. Unlike ID3, all features are selected based on the same set of objects. Thus, the rules generated by the Rg algorithm represent a more global view of the data set.

Search Procedure

Finally, let us now compare the two approaches based on their search procedures. ID3 avoids the combinatorial explosions of possibilities for searching N features simultaneously by performing N one-dimensional searches at each node of the tree. While such an approach is easy to implement and greatly simplify the search in multidimensional spaces, it also causes ID3 to "lose" the global picture of finding a set of features that collectively group objects of a common class. On the other hand, Rg performs global N-dimensional searches in feature space although patterns are discovered one at a time. Each M-dimensional pattern, where $M \leq N$, is discovered by an N-dimensional search allowing for a collective analysis of the data set and the formation of more general rules that capture the essence of the data. The Rg algorithm pays a price for this more global analysis of the data with a search procedure that is more complex and difficult to implement.

CONCLUSIONS

Both the ID3 and Rg inductive algorithms are effective alternatives to the painstaking process of knowledge acquisition since they require minimum human intervention. The use of these two inductive methods save time and effort in the development of knowledge bases for expert systems. The knowledge acquired by these methods can be encoded as *if...then* production rules that are logically complete and consistent. Information entropy is used to generate discriminatory rules that are described by only a small number of features. The Rg algorithm construct rules that are more general and less contrived than the ID3 rules and should be better able to classify new objects. This advantage of the Rg algorithm is contrasted with an easier to implement and faster computation of the ID3 algorithm. As more EBR-II failed fuel data becomes available, a more in-depth comparison between the two approaches should be performed by evaluating the two algorithms as a function of their misclassification rate. A cross-validation technique[17] could be used to divide the data set into training and test data and the misclassification rate for the test data would be analyzed based on rules constructed using the training data.

REFERENCES

1. B. G. Buchanan and E. A. Feigenbaum, "Dendral and Meta-Dendral: Their Applications Dimension," *Artificial Intelligence*, **11**, 5-24, 1978.

2. B. D. Zimmerman and J. A. Rawlins, "CRAW: An Expert System for Nuclear Reactor Cover Gas Alarm Analysis," HEDL-SA-3504FP, Hanford Engineering Development Laboratory, 1985.

3. B. D. Zimmerman, G. Jansen, Jr., and M. J. Moen, "Rule Induction Analysis and Expert System Construction," *Transaction of the American Nuclear Society*, **62**, pp. 123-124, 1990.

4. J. Reifman and J. C. Lee, "Reactor Diagnostics Rule Generation Through Statistical Pattern Recognition," *Nuclear Science and Engineering*, **107**, 291-314, 1991.

5. R. Mikaili and J. D. B. Lambert, "An Expert System for Fuel Failure Diagnosis in EBR-II," *Transactions of the American Nuclear Society*, **63**, pp. 116-118, 1991.

6. B. H. O'Brien and B. J. Newby, "Inductive Classification of Operating Data From Fluidized Bed Calciner," *Proceedings of the American Nuclear Society Topical Meeting on Frontiers in Innovative Computing for Nuclear Industry*, pp. 876-882, Jackson, Wyoming, September 15-18, 1991.

7. W. J. Leech, "An Overview of Artificial Intelligence and Neural Network Applications in the Commercial Nuclear Fuel Division," *Proceedings of the American Nuclear Society Topical Meeting on Frontiers in Innovative Computing for Nuclear Industry*, pp. 421-427, Jackson, Wyoming, September 15-18, 1991.

8. EXPERT-EASE, distributed by Jeffrey Perrone & Associates, Inc., San Francisco, California, 1983.

9. 1st CLASS, distributed by 1st Class Expert Systems, Inc., Wayland, Massachusetts, 1989.

10. J. R. Quinlam, "Learning Efficient Classification Procedures and their Applications to Chess End Games," *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., Tioga Publishing Company, pp. 463-483, 1983.

11. R. Christensen, "Entropy Minimax Multivariate Statistical Modeling I: Theory," *International Journal of General System*, **11**, 231-277, 1985.

12. G. S. Was, R. Christensen, C. Park, and R. W. Smith, "Statistical Patterns of Fuel Failure in Stainless Steel Clad Light Water Reactor Fuel Rods," *Nuclear Technology*, **71**, 445-457, 1985.

13. M. Suwa, A. C. Scott, and E. H. Shortliffe, "An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System," *Artificial Intelligence Magazine*, **3**, 4, 16-21, 1982.

14. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1949.

15. E. B. Hunt, J. Marin, and P. T. Stone, *Experiments in Induction*, Academic Press, 1966.

16. R. Mikaili, *Design of an Expert System for Failed Fuel Identification and surveillance in EBR-II*, PhD Dissertation, Iowa State University, 1990.

17. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, 1984.

ACKNOWLEDGMENTS

The authors would like to thank R. Mikaili for providing the EBR-II failed fuel data. The first author was supported by the U. S. Department of Energy, Nuclear Energy Programs, under contract number W-31-109-ENG-38.

END

DATE
FILMED
6/03/92