possible; second, it helps make McNeil chemists more aware of areas of interest to their colleagues. The structure of this database is similar to that of the proprietary database; however, the group has deleted many datatypes to simplify data input. Figure 2 gives a complete hierarchy of datatypes used for McNeil's in-house literature database. Use of the literature database for storage of reactions forces each chemist to evaluate their reading and formulate concrete opinions on why specific articles are of interest to them.

Another database the group is constructing contains information on the overall conversion of commercially available starting materials to compounds in development. Unlike the first database, this database contains no experimental details on individual reactions. Its purpose is to make information on overall yields, equipment, and manpower requirements readily available. In addition, commercial suppliers and costs of starting materials will be stored in this database. It is anticipated that this database will simplify scheduling in the chemical pilot plant and make procurement of starting materials more efficient.

Metabolism can be viewed as a chemical reaction occurring under a given set of experimental conditions where important parameters are not temperature, pressure, etc. but rather animal species. Furthermore, metabolites can be classified by their distribution. By constructing a metabolite database, chemists at McNeil may be able to predict the metabolism of new drug entities and the distribution of the metabolites in test animals and man.

SUMMARY

The combination of software and data is what determines the usefulness of any database management system for a specific purpose. Ideally, the database management software should be designed to be "user friendly" and flexible. Unfortunately, these two needs are frequently only partially compatible. In order to make a system user friendly, designers of the system must often sacrifice software flexibility. Alternatively, software written for general use frequently requires specific codes for generating file structures and for searching those files. In order to utilize the database management system, one must first learn how to generate the file structure.

REACCS is no exception to the generalization given above. The flexibility of this database management program requires that some individual or individuals must learn the syntax necessary to generate the desired proprietary database structure. However, the effort to learn this syntax is not nearly as great as that required to define the particular database needs and identify the datatypes that will most effectively fulfill those needs. Without a well-planned database, users will be frustrated by their inability to find information that they are seeking. The need for a considerable expenditure of effort to define the purpose of any database prior to its creation cannot be overemphasized. Designers of the system must spend additional time to clearly identify those datatypes necessary to achieve that purpose. Finally, it is necessary to arrange those datatypes in a hierarchy and to add additional datatypes to organize the database.

At McNeil Pharmaceutical, the combination of well-designed databases with the flexible search strategies allowed by REACCS makes this system highly desirable for use by our chemists.

REFERENCES

(1) Heller, S. R. "Reaction Indexing". *Ind. Chem.* **1987**(Feb), 68.
(2) von Kiedrowski, G.; Eifert, A. "Handling Chemical Structures with a Personal Computer". *Intelligent Instrum. Comput. Appl. Lab.* **1986** (March/April), 110.
(3) Seiter, C. H. "Your PC May Solve Your Chem Lab Problems". *Res. Dev.* **1987**(March), 94.
(4) French, S. E. "Our Reaction Access System". *CHEMTECH* **1987** (Feb), 106.
(5) Hrib, N. J. "Recent Developments in Computer-Assisted Organic Synthesis". *Annu. Rep. Med. Chem.* **1986**, *21*, 303-311.
(6) Mills, J. E.; Maryanoff, C. A.; Sorgi, K.; Scott, L.; Stanzione, R. *J. Chem. Inf. Comput. Sci.* (preceding paper in this issue).

Using Analytical Data To Build Expert Systems

W. A. SCHLIEPER* and T. L. ISENHOUR
Department of Chemistry and Biochemistry, Utah State University, Logan, Utah 84321-0300

J. C. MARSHALL
Department of Chemistry, Saint Olaf College, Northfield, Minnesota 55057

Received September 11, 1987

Generating expert systems can be a long, arduous process. Human experts may be able to use some of their hard-won knowledge to help develop expert systems, providing the domain of the system is modest. As the domain expands, the problem of rule formalization may even baffle a human expert. Inductive methods, implemented by computers, can help generate production rules suitable for expert systems. The ID3 algorithm has been applied to two sets of chemical data, resulting in decision trees useful to classify the data. The resulting decision trees were then transformed directly to a set of production rules for an expert system. This procedure allows the attributes of the data set to be represented directly as objects that are descriptive of the actual data and does not require data transformations. This algorithm is also capable of distinguishing when attributes and associated values do not sufficiently span a given domain.

INTRODUCTION

The classification of complex data objects is frequently of central importance in analyzing chemical data and efficiently retrieving information from data files. Numerical pattern recognition techniques have been widely used for this task.[1-4] These numerical techniques frequently seek to cluster the data by transforming the inherent data attributes to new and possibly composite attributes. This has the conceptual disadvantage that the new attributes may be very difficult to relate to the original data from which they were derived.

Object oriented programming strategies do not require data transformations as all the attributes of the data, both quantitative and qualitative, may be represented as objects that are manipulated directly. This strategy casts the problem into a

0095-2338/88/1628-0159$01.50/0

Table I. Data for Benzene Substitution Used To Generate Decision Tree[a]

compound	degree of substitution	IR ranges, cm^{-1}				
		650–699	700–749	750–799	800–849	850–899
toluene	mono	S	S	W	W	W
m-xylene	meta	S	W	S	W	W
o-xylene	ortho	W	S	S	W	W
p-xylene	para	W	W	S	W	W
1,2,3-trimethylbenzene	1,2,3-tri	W	S	S	W	W
1,2,4-trimethylbenzene	1,2,4-tri	W	W	W	S	M
1,3,5-trimethylbenzene	1,3,5-tri	S	W	W	S	W
1,2,3,4-tetramethylbenzene	1,2,3,4-tetra	W	W	S	S	W
1,2,3,5-tetramethylbenzene	1,2,3,5-tetra	W	S	W	W	S
1,2,4,5-tetramethylbenzene	1,2,4,5-tetra	W	W	W	W	M
pentamethylbenzene	penta	W	W	W	W	S

[a] W, weak or no absorption; M, medium absorption; S, strong absorption.

logic domain that may be conceptualized as a logic tree. Expert system methods are, in theory, capable of classifying data objects on the basis of randomly organized rules characterizing them. However, in complex problems the rules must be efficiently organized or the rule structure will rapidly become unmanageable in both size and logical complexity.

This paper demonstrates the use of the ID3[5-7] (Iterative Dichotomiser 3) algorithm as a method of inductive inference that retains the original meaning of the attributes of the data. This algorithm, operating directly on a file of data objects, is capable of organizing data according to attribute–value pairs so that the most efficient set of rules spanning the data can be derived. The algorithm will also determine when the attributes available do not fully classify all the data. As implemented, the system described will produce a set of optimum rules that can be used directly by an expert system written in PROLOG.

EXPERIMENTAL SECTION

The software used to apply the ID3 algorithm was written in Turbo Prolog (Borland International) and runs on a Leading Edge microcomputer. The microcomputer is equipped with one 360K floppy disk drive, one 10MB Winchester hard disk, and 640K of RAM (random access memory) and runs under the MS-DOS operating system.

THEORY

The knowledge base used by an expert system for classification decisions can be most efficiently represented as a set of rules based on the minimal decision tree spanning the data. The root node of this tree is the attribute that minimizes the number of branches from the root. Each branch from the root node contains a different value of the root attribute. Each branch required from these second-level nodes may be branched further by using attributes different from the previous attributes used to split the data. The class attributes and values will occupy terminal nodes in the decision tree. If more than one attribute is used to describe the data, the decision tree will not be unique. As the number of attributes used to describe the data increases, the number of possible decision trees also increases. In complex data sets, this proliferation makes the most efficient structuring of the decision tree critically important.

The ID3 algorithm is a simple method for determining classification trees that minimize the number of tests needed to classify objects. The algorithm is based on information theory and treats the entropy of classification. The entropy of classification is a measure of the information content of classifying an object in a particular class.

The entropy of classification, $H(C|A)$, can be calculated from the equation[5]

$$H(C|A) = \sum_{j=1}^{M} p(aj)H(A|aj)$$

where M is the number of values for attribute A, $p(aj)$ is the probability that A equals value aj, and $H(A|aj)$ is the entropy for the subtree whose entries all have the same value for the attribute used to make the split in the decision tree. The entropy for the subtree, $H(A|aj)$, is calculated by using Shannon's formula[8]

$$H(A|aj) = -\sum_{i=1}^{N} [p(ci|aj) \log 2[p(ci|aj)]]$$

where $p(ci|aj)$ is the probability that class C will have the value ci when attribute A equals aj. The probabilities in each of these equations is replaced with frequency distributions when used to analyze sets of data.

The decision tree is formed by first choosing an attribute that will minimize the entropy calculation. The tree is split into smaller trees that contain objects that have the same value for the attribute used to make the split. The attribute used to split the tree is then removed from further calculations. Each subtree is then subsequently split according to which of the remaining attributes has the smallest value for the entropy calculation. The process terminates when all objects belonging to the same branch have the same class or when all attributes have been exhausted. The attributes do not sufficiently describe the data if two different class values occupy the same terminal node. The process of splitting the tree on the basis of the attribute with the smallest entropy of classification results in a final decision tree with no redundant information. In the context of an expert system that uses production rules as the basic form of knowledge representation, each path from the root node to a terminal node is an efficient production rule.

RESULTS AND DISCUSSION

Two examples will be given to demonstrate the ability of the ID3 algorithm to assist in the development of expert systems for chemical applications. The first example generates a set of production rules to determine the substitution of a benzene ring by analyzing absorption bands in the infrared. The second example produces a possible expert system data base to classify compounds according to functionality on the basis of melting points, boiling points, and solubilities. While these are small data sets, they demonstrate the generality of this approach.

The infrared (IR) region from 650 to 900 cm^{-1} has been used to determine the substitution of benzene rings.[9-11] A set of data was collected from the Aldrich Library of FT-IR Spectra and is represented in Table I. The absorption bands listed in Table I are evenly spaced from 650 to 899 cm^{-1}. The absorption bands are further quantified according to the intensity of the bands (W, weak or no absorbance; M, medium; S, strong). The band was specified as a weak absorber if the transmittance was greater than 60%, a medium absorber if the transmittance was between 20 and 60%, and a strong absorber if the transmittance was less than 20%. Any ab-

Table II. Entropy Values from ID3 Calculations on Infrared Absorbance Data in Table I

attribute	850–899	800–849	750–799	700–749	650–699
level 0	2.2	2.6	2.5	2.5	2.6
level 1					
group 1		0.0	1.0	1.0	1.0
group 2		1.9	1.9	1.8	1.8
group 3		1.0	1.0	0.0	1.0
level 2					
group 1		0.7	0.7	0.7	
group 2		1.2	2.0	1.0	
level 3					
group 1			0.0		
group 2			0.0		

sorption bands that overlapped the range boundaries were specified in both ranges.

Figure 1 shows the decision tree generated from the set of data in Table I. The entropy of classification for each of the attributes used to describe the data entries is calculated with the ID3 algorithm. The results are presented in Table II by level and group. A new level starts at each row of nodes, while groups consist of the attribute–value pairs associated with each node. The attributes used at the nodes have the associated entropy values underlined in the table. The initial split in this decision tree starts with the absorption band of "850–899" since it gave the lowest entropy value. The three branches emanating from the root node separate the data into three distinct groups that can be further analyzed.

The first group consists of all data entries that have the value of "M" for the root attribute. This data set has only two members, 1,2,4-trimethylbenzene and 1,2,4,5-tetramethylbenzene. After the initial split, these data entries are described by only four attributes, 800–849, 750–799, 700–749, and 650–699, as are each of the three groups on level 1 of the decision tree since the root attribute is removed from future calculations. Of these four attributes, the attribute 800–849 gives the lowest entropy value for the two entries and is the best attribute for the next split. The data in this branch need not be split further since each of the two entries occupies a terminal node with different class values.

The second group on level 1 contains 7 of the 10 entries in this data set. Of the four attributes, 800–849, 750–799, 700–749, and 650–699, the last two attributes have the same entropy value; therefore, either attribute would work equally well. In the case of ties, the last attribute was chosen for consistency. The choice of the attribute 650–699 results in a split of the data into two groups on level 2. The calculations for each of the two branches on level 2 only consider three remaining attributes, 800–849, 750–799, and 700–749.

The third group on level 1 contains only two members, 1,2,3,5-tetramethylbenzene and pentamethylbenzene. The entropy calculations for this group are similar to that of group one on this level. A split based on the attribute 700–749 results in two terminal nodes with distinct class values.

The first group on level 2 consists of the three members toluene, *m*-xylene, and 1,3,5-trimethylbenzene. The entropy values for each of the attributes was 0.7, which means that any of the three attributes will give the same degree of splitting

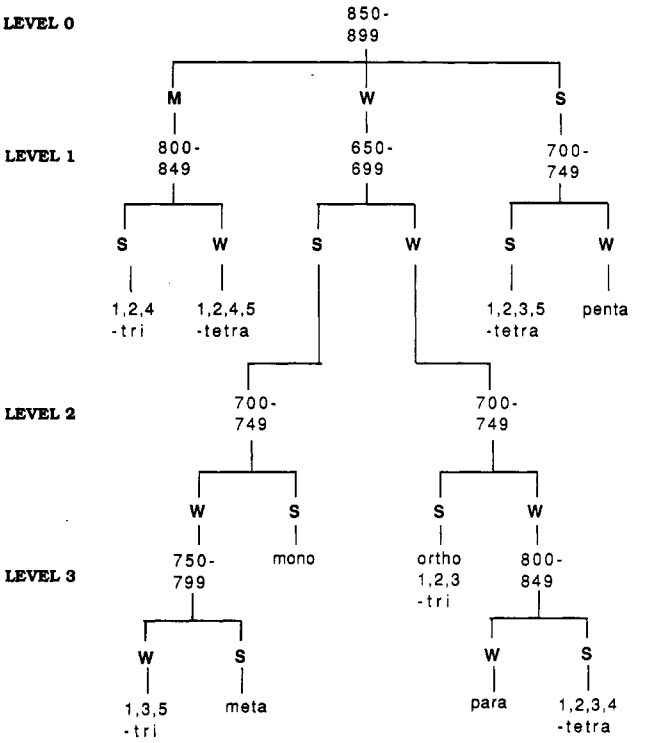

Figure 1. Decision tree generated from infrared absorbance data in Table I. The node attributes describe absorbance ranges, while the branches define the intensity of the absorbance.

for the subtree below this node. Choosing 700–749 as the split attribute results in one terminal node and an additional branch that can reach a terminal node for the final two entries by using the attribute 750–799.

The second group on level 2 consists of *o*-xylene, *p*-xylene, 1,2,3-trimethylbenzene, and 1,2,3,4-tetramethylbenzene. On the basis of the entropy value for the attribute, 700–749 generates the next level in the decision tree which consists of two groups. The members of the first group cannot be differentiated by the remaining attribute–value pairs; therefore, two different class values occupy the same terminal node. The second group can be further separated on the attribute 800–849.

This set of data examples represents a case in which the attributes, even though all are used in calculation of the decision tree, are not sufficient to completely classify the data. This inadequacy is apparent since two substitution values occupy the same terminal node. Additional wavelength ranges could be added to attempt to fully describe the system.

Inductively generated rules can never be proven, although they can be disproved. A data entry, not used to generate the rules, may not be properly classified. One way to increase the reliability of the decision tree is to use the decision tree thus created to classify data that was not used in its generation. Table III gives a set of compounds that are similar to some of the compounds used to generate the tree. These data were taken from the same source and in the same manner as the original set of compounds.

The testing of compounds begins at the root node of the

Table III. Set of Compounds Used To Test Accuracy of Decision Tree in Figure 1[a]

compound	degree of substitution	IR ranges, cm^{-1} 650–699	700–749	750–799	800–849	850–899
cumene	mono	S	S	S	W	W
1,2-diethylbenzene[b]	meta	W	S	S	W	W
1,3-diethylbenzene[b]	ortho	S	S	M	M	M
1,4-diethylbenzene[b]	para	W	W	W	S	W
1,3,5-triethylbenzene[b]	1,3,5-tri	W	M	W	W	S

[a] W, weak or no absorption; M, medium absorption; S, strong absorption. [b] Misclassified.

Table IV. Set of Compounds Used To Create the Decision Tree in Figure 3[a]

compound	boiling point	melting point	solubilities		
			water	alcohol	ethane
acetic acid	100–125 (118)	0–25 (17)	INF	INF	INF
formic acid	100–125 (101)	0–25 (8)	INF	INF	INF
propanoic acid	126–150 (141)	−0 to −25 (−21)	INF	INF	S
butanoic acid	151–175 (164)	−0 to −25 (−4)	INF	INF	INF
pentanoic acid	176–200 (186)	−26 to −50 (−34)	S	S	S
hexanoic acid	201–225 (205)	−0 to −25 (−2)	I	S	S
heptanoic acid	201–225 (223)	−0 to −25 (−8)	SS	S	S
formaldehyde	−0 to −25 (−21)	−76 to −100 (−92)	S	S	INF
acetylaldehyde	0–25 (21)	−101 to −125 (−121)	INF	INF	INF
propanal	26–50 (49)	−76 to −100 (−81)	S	INF	INF
butanal	76–100 (76)	−76 to −100 (−99)	S	INF	INF
pentanal	101–125 (103)	−76 to −100 (−92)	SS	S	S
hexanal	126–150 (128)	−51 to −75 (−56)	SS	V	V
2-propanone	51–75 (56)	−76 to −100 (−95)	INF	INF	INF
2-pentanone	101–125 (102)	−76 to −100 (−78)	SS	INF	INF
3-pentanone	101–125 (102)	−26 to −50 (−40)	V	INF	INF
2-hexanone	126–150 (128)	−51 to −75 (−57)	SS	INF	INF
3-heptanone	126–150 (147)	−26 to −50 (−39)	I	INF	INF
4-heptanone	126–150 (144)	−26 to −50 (−33)	I	INF	INF

[a] I, insoluble; SS, slightly soluble; S, soluble; V, very soluble; INF, miscible.

decision tree. The first comparison is based on the intensity of the absorbance band 850–899. Depending on the intensity of the test compound over this range, the comparison continues to the next node. This process continues until either a terminal node is reached or a branch does not exist in the tree for an intensity value of the test compound. If the test continued to a terminal node and the substitution was properly determined, the test was successful. Otherwise, the test failed.

One of the five test compounds, cumene, was properly classified by these decision rules while the remaining compounds were improperly classified due to slight differences in intensity and/or locations of absorbance bands as compared to similar compounds in the teaching set. To correct the decision tree, the misclassified compounds can be added to the original set of data and used to regenerate the decision tree. The decision tree resulting from the addition of the test compounds is shown in Figure 2. The process of testing the decision tree continues until some acceptable ratio of successes to failures is realized.

The second example demonstrates the separation of compounds in functional classes according to melting points, boiling points, and solubilities in water, ethyl alcohol, and ethane. The data for this example were collected from the 58th edition of the *Handbook of Chemistry and Physics*. Melting point and boiling point ranges were used instead of the exact melting and boiling points since classification of functionalities, instead of specific compounds, was planned. The actual boiling points and melting points are in parentheses following the range specifications in Table IV.

The attributes used to describe these data could have been set to any range that was appropriate. The range of 25 °C was used just to show the principle of data classification in the manner of a human expert. The temperatures were rounded to the nearest integer and were reported in Celsius. The range −0 to −25 was used to represent any values less than zero yet greater than or equal to 25 °C. The values for the solubility can take on the following values: insoluble (I); slightly soluble (SS); soluble (S); very soluble (V); and miscible (INF).

The decision tree built from the data in Table IV is shown in Figure 3. The calculations made for this decision tree will not be covered in this text but do appear in Table V by level and group. The levels in Figure 3 are numbered from left to right at each node starting at zero. The groups are numbered from top to bottom on each level. As previously mentioned, the last attribute was used for the split if two or more entropy numbers on the same level and in the same group are equal.

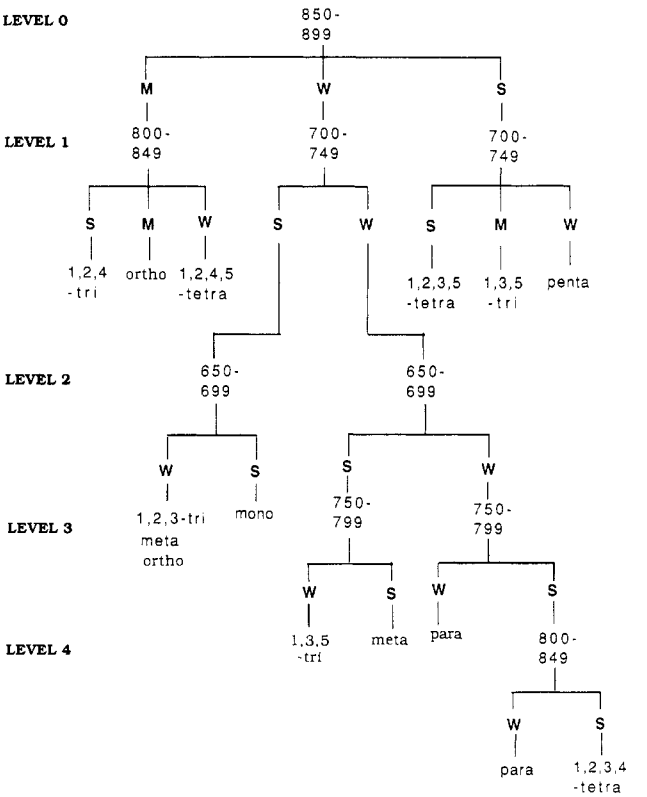

Figure 2. Decision tree generated from infrared absorbance data in Tables I and II. The node attributes describe absorbance ranges, while the branches define the intensity of the absorbance.

Table V. Entropy Values from ID3 Calculations on Melting Point, Boiling Point, and Solubility Data from Table IV

attribute	water	ethane	alcohol	boiling	melting
level 0	1.1	1.2	1.3	0.8	0.6
level 1					
group 1	0.3	0.8	0.7	0.3	
group 2	1.0	0.0	0.0	1.0	
group 3	0.0	0.0	0.0	0.0	
level 2		1.0	0.0	0.0	

The attribute–value pairs used to generate this tree were sufficient to fully describe the data since each terminal node consists of only one classification.

The attribute–value pairs used to generate this tree were sufficient to fully describe the data since each terminal node

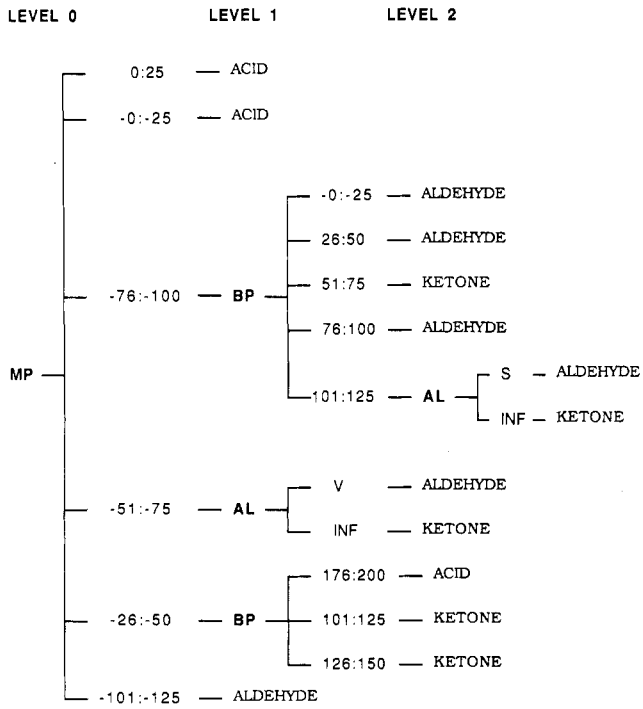

Figure 3. Decision tree for functionality classification based on melting points, boiling points, and solubility.

consists of only one classification. Some of the nodes can be collapsed into single nodes. For instance, the two nodes for the melting point ranges 0–25 and −0 to −25 both contain the classification of acid and could be reclassified by using the range 25 to −25. Likewise, the terminal nodes for the boiling point ranges of 101–125 and 126–150 could be replaced with a single node with the range 101–150 since the only classification on the two nodes is ketone.

Another interesting point about the attributes used in this test set is that they may not all be used in the final decision tree. The attributes describing the solubilities in ethane and water do not include any additional information about the system and do not occur in the decision tree. The absence of an attribute in the final decision tree does not mean the attribute is not useful in describing the data. It could mean that the attribute is highly correlated with another attribute used in the decision tree. The reliability of this decision tree can be tested by classifying data not used in the generation of the tree.

The vocabularies available to Turbo Prolog allow the programmer the option of creating elaborate user interfaces if needed. For simplicity, this implementation of the ID3 algorithm interacted with a preexisting data-base file. Each data-base entry in this file contains a list of attribute–value pairs. After the data are analyzed, the resulting Prolog rules are placed in another file.

A list in Prolog is represented by a series of entries separated by commas and enclosed in a set of brackets. For the purposes of generating a decision tree, the list must contain a minimum of two attribute–value pairs, one used to split the tree and one to classify the object. The format used is

```
exper([att(Attribute1,Value1),att(Attribute2,Value2),...])
```

where "exper" is the predicate used to describe an individual record and "att" describes one attribute–value pair, each of which contains an attribute and a value. Any symbol can be used to represent the attributes, for instance, melting point or boiling point. The value of an attribute–value pair can in fact be a list of symbols used to represent the attribute. An example of the value list is [S], where the S, the only member in the list, could represent a strong absorbance. Symbols in the Prolog sense can be any alphanumeric character(s).

The output file contains the set of optimum rules generated from the branches of the decision tree. Each record in the output file occupies one line and takes on the format

```
rules([att(Attribute1,Value1),...],att(Class,ClassValue))
```

where "rules" is the predicate used to describe a record, [att(Attribute1,Value1),...] is the list of attribute–value pairs used to split the decision tree, and att(Class,ClassValue) is the terminal attribute–value pair used to classify the data. This representation is synonymous with the more familiar representation of

```
IF      Attribute1 = Value1
AND     Attribute2 = Value2
AND     ...
THEN    Class = ClassValue.
```

The number of attribute–value pairs in each record depends upon the number of splits in the decision tree from the root node to the terminal node. These rules are in a format that can be directly used in an expert system written in prolog.

CONCLUSION

The ID3 algorithm has been applied to two sets of chemical data, resulting in decision trees useful to classify the data. The resulting decision trees were then transformed directly to a set of production rules for an expert system. The procedure described, based on information theoretic considerations, will automatically produce the most efficient possible set of production rules for expert system applications while giving feedback on the ability of the attributes to span the domain.

The applicability of an expert system built in this way is limited only by the domain of the input data. This procedure allows the attributes of the data set to be represented directly as objects that are descriptive of the actual data and does not require data transformations. The execution time of the algorithm does increase with the number of attribute–value pairs and the number of lists. The exact relationship execution time cannot be exactly quoted.

REFERENCES AND NOTES

(1) Jurs, P. C.; Isenhour, T. L. *Chemical Applications Of Pattern Recognition*; Wiley: New York, 1975.
(2) Massart, D. L.; Kaufman, L. *The Interpretation Of Analytical Chemical Data By The Use Of Chemical Analysis*; Wiley: New York, 1983.
(3) Malinowski, E. R.; Howery, D. G. *Factor Analysis In Chemistry*; Wiley-Interscience, New York, 1980.
(4) Hangac, G.; Wieboldt, R. C.; Lam, R. B.; Isenhour, T. L. *Appl. Spectrosc.* **1982**, *36*, 40.
(5) Quinlan, J. R. "Learning Efficient Classification Procedures And Their Application To Chess End Games". In *Machine Learning—An Artificial Intelligence Approach*; Michalski, R. S., Carbonell, J. G., Mitchell, T. M., Eds.; Tioga: Palo Alto, CA, 1983.
(6) Thompson, B.; Thompson, W. *Byte* **1986**, *11*(13), 149.
(7) Derde, Marie-Paule, et al. *Anal. Chem.* **1987**, *59*, 1868.
(8) Eckschlager, Karel; Stepanek, Vladimir. *Information Theory As Applied To Chemical Analysis*; Wiley: 1979; pp 70–74.
(9) Bellamy, L. J. *The Infra-red Spectra Of Complex Molecules*; Wiley: New York, 1975; pp 84–90.
(10) Socrates, G. *Infrared Characteristic Group Frequencies*; Wiley: New York, 1980; p 83.
(11) Baker, A. J.; Cairns, T. *Spectroscopic Techniques In Organic Chemistry*; Heyden: London, 1966.