ELSEVIER

Food Control 10 (1999) 107–116

FOOD CONTROL

Computer-aided microbial identification using decision trees

J. Rattray, J.D. Floros *, R.H. Linton

Department of Food Science, Purdue University, West Lafayette, IN 47907-1160, USA

Abstract

This paper will demonstrate the utility of using machine learning methods to develop more efficient microbial identification (ID) techniques. The use of computer algorithms to create new decision trees can improve efficiency and increase systematization in the field of microbiology. Preliminary results indicate that decision tree algorithms can create new structures that require fewer tests on average to reach a positive identification of an unknown organism. Including test time and cost factors can make further improvements, resulting in systems that are more time-efficient and/or cost-effective. Machine learning techniques can also create customized ID systems for specific applications. This paper will explain the induction of decision trees and show examples of their use in microbial ID. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Microbial identification; Artificial intelligence; Decision trees

1. Introduction

Microbial identification (ID) is a common task in food science disciplines, both in research and in industry. The current system for identifying an unknown bacterium has been empirically developed as the field has grown over the years. It is a series of tests such as those outlined in Bergey's Manual of Bacteriology (Tansil, 1984). ID usually begins with a Gram stain and continues through a series of branches based on the outcome of the previous tests. Tests include morphological/physical features, metabolic characteristics and chemical reactions. Structurally, the test series resembles a tree rooted at the Gram stain and branching out to many tests. The major branches of the tree define the families of microbes, the minor branches are the genuses, and the leaves (ultimate ends) are the species and types.

Identifying unknown microbes found in foods or food processing equipment involves significant amounts of time and resources spent in the laboratory. Many efforts have been made to reduce the work required for this task, resulting in the development of specialized test kits for certain types of microbial ID. However, despite these aids, no overall standardized system exists for determining test sequences for microbial ID or for designing new tools to identify other microorganisms efficiently. Many microbial test protocols have been standardized, but test selection and test order have not. Sometimes one can use background information or prior experience with a given process or piece of process equipment to predict the identity of an unknown and then confirm it with a few simple tests. Usually, however, it is left up to individual microbiologists and lab technicians to design test sequences and successfully identify the unknown microbes they encounter. There is a need for more efficient and customizable methods to identify foodborne microorganisms, and for methods to select the most efficient sequences of tests for identification.

Machine learning is a growing field of artificial intelligence research. A number of useful techniques have been developed to extract knowledge from datasets and create generalized classification schemes. One of these techniques is decision tree induction, which is used to create classification and prediction systems very similar to the trees used for microbial ID. Machine learning algorithms analyze the potential gain in information at each fork in the tree and determine which test is the most advantageous to be used at that point.

In the machine learning field, microbial ID is a very large domain. There are thousands of species (classes)

* Corresponding author. Tel.: +1-765-494-9111; fax: +1-765-494-7953; e-mail: floros@foodsci.purdue.edu

0956-7135/99/$ – see front matter
PII: S0956-7135(99)00004-3

and hundreds of features. The current standard tree has been empirically constructed, with new branches being grafted on an as needed basis when new species are discovered. The tree has grown with the field of microbiology, incorporating new tests as they are created and new microbes as they are found. The commonly mentioned groupings of microbes, such as "gram-negative rod-shaped bacteria", are based on the tree branches and features. Some branches are more thorough than others, as certain features tend to indirectly divide microorganisms based on their habitat, and only the relevant and/or important branches are followed extensively.

The objective of this project was to create a computer-based tool that generates classification schemes for microbial ID on demand, reduces the number of tests required to classify an organism, and can create customized ID systems. This tool, in its present form, is not expected to recognize unknown new species as such.

2. Theory

2.1. Decision trees – what they are and how they work

A decision tree is a structured series of tests starting from a single root and branching out repeatedly to a set of leaves or classifications (Fig. 1). They break a mixed group of information or identifiers down to pure classes. Decision trees can be used for both prediction and classification tasks.

There is some specific terminology from the machine learning community associated with decision trees. A class is a grouping or label (e.g. *Salmonella* spp). A feature is an attribute that can be tested, such as Gram stain reaction. An instance is a set of features with a class label; a description of one sample out of a class. A node is a test that splits a group of instances into two or more groups based on the value of a feature, and a leaf is the utmost extremity of a tree. Instances reaching a leaf are labeled as belonging to a certain class.

There are several popular systems for creating decision trees. The basic process is straightforward and common to all of them. Creation begins with all of the instances and classes in one group. Selecting a feature creates the root node and splits the group into two or more subgroups based on its response to that feature. The process then recurses down each branch, selecting another test for each subgroup (node) and subdividing again until pure classes are reached or there are no more useful features remaining. At this point, the subgroup is a leaf and is labeled with its majority class.

The key to successful decision tree creation is in selecting the "best" feature at each node. The best feature is defined as the feature that provides the most information. Decision tree algorithms use measurements of information based on information transmission theory. These measurements evaluate how much more information is needed, on average, to classify the instances at a given node, and are based on the distribution of classes at the node. For a given node N, the information required to classify the instances at N is a function of the number and distribution of the classes at N. For a sys-

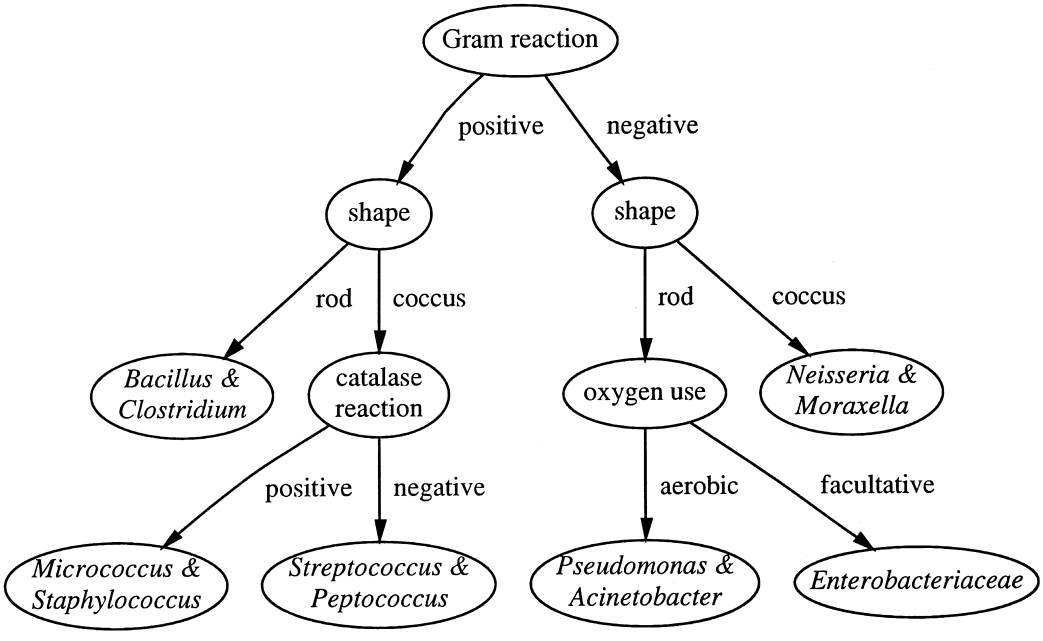

Fig. 1. An example of a decision tree (adapted from Villarreal & Sun, 1995).

tem with two classes, C_1 and C_2, the information requirement is

$$\text{info}(N) = -\frac{c_1}{c_1+c_2}\log_2\left(\frac{c_1}{c_1+c_2}\right) - \frac{c_2}{c_1+c_2}\log_2\left(\frac{c_2}{c_1+c_2}\right), \quad (1)$$

where c_1 is the number of instances in class C_1 and C_2 is the number of instances in class C_2 at the node of interest. This method can easily be extended to systems with more than two classes. A system with k classes $C_1 \ldots C_k$ would have its information gain calculated by

$$\text{info}(N) = -\frac{c_1}{\sum c}\log_2\left(\frac{c_1}{\sum c}\right) - \frac{c_2}{\sum c}\log_2\left(\frac{c_2}{\sum c}\right) - \cdots - \frac{c_k}{\sum c}\log_2\left(\frac{c_k}{\sum c}\right). \quad (2)$$

Determining which feature to split on requires evaluation of the information obtained by splitting on each of the available features. Splitting a node with a feature F with f values F_1 through F_f (see Fig. 2) creates the new nodes N_1 through N_f, with information requirements info(N_1) through info(N_f). The information gained by splitting on feature F is given by

$$\text{gain}(F) = \text{info}(N) - \text{info}_F(N), \quad (3)$$

where $\text{info}_F(N) = \sum_{i=1}^{f} \text{info}(N_i)$ and N_i are the nodes N_1 through N_f created by splitting N on F.

After calculating the information that can be gained by splitting with each feature available at node N, the instances at the node are split using the feature with the best (highest) information gain, and the process is repeated at the new nodes created by the split.

The information gain criterion as described above has been shown to be biased towards multi-way splits, features that divide the instances into many groups (Kononenko, Bratko & Roskar, 1984; Quinlan, 1986). An improved criterion, known as the gain ratio, has been formulated to eliminate this bias. It is defined as (Quinlan, 1986)

$$\text{Gain ratio}(F) = \frac{\text{gain}(F)}{\text{split info}(F)}, \quad (4)$$

where split info$(F) = -\sum_{j=1}^{f}(c_{1_j}+c_{2_j})/(c_1+c_2)\log_2(c_{1_j}+c_{2_j})/(c_1+c_2)$ for a system with two classes. The notation c_{1_j} indicates the number of elements in class c_1 at node N whose value for feature F is F_j. Similarly, c_{2_j} indicates the number of elements in class c_2 at node N whose value for feature F is F_j. The function split info(F) is known as the information value of F. It accounts for the information contained in answering the question "what is the value of the feature F?" (Quinlan, 1986). This notation can also be extended to systems with more than two classes.

A number of decision tree software packages are available in the machine learning community. Each has its own technical refinements and slightly different approach to the task. Work in this study was performed with a modified version of c4.5, revision 8 (Quinlan, 1991) due to its availability in the public domain. This package was modified for improved output quality and graphics capability. Figures and graphs were produced using *dot*, part of the *graphviz* package from AT&T Bell Labs.

3. Materials and methods

3.1. Part 1 – pilot study

A proof-of-concept pilot study was performed with a small dataset containing 26 microbial species representing 16 families (Table 1). A database containing 14 commonly used features (Table 2), along with an empirically created decision tree (Fig. 3), was used to differentiate between them (Villarreal & Sun, 1994). Some

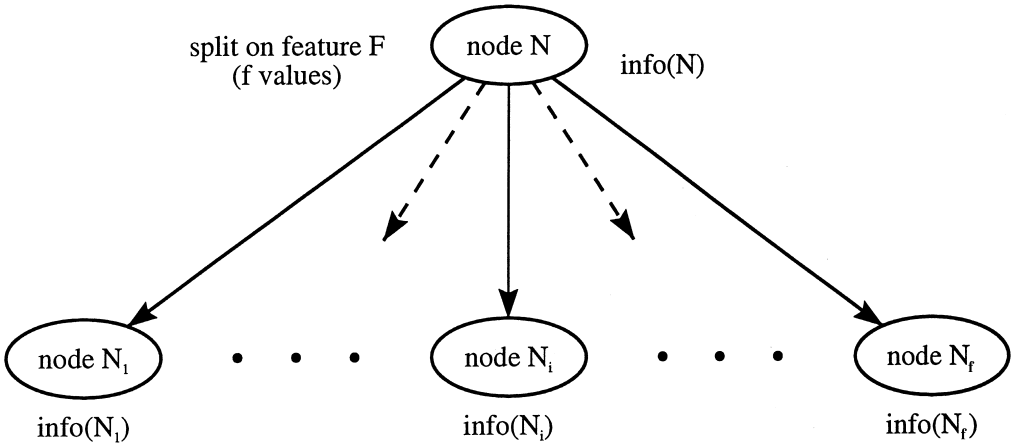

Fig. 2. Splitting a node N on feature F.

Table 1
Microbial species used in pilot study

Actinobacter amitratus	*Clostridium difficile*	*Neisseria perflava*	*Staphylococcus aureus*
Actinobacter woffii	*Enterobacter aerogenes*	*Peptococcus activius*	*Staphylococcus caprae*
Bacillus alvei	*Escherichia coli*	*Proteus vulgaris*	*Staphylococcus epidermis*
Bacillus pumilis	*Escherechia hermannii*	*Pseudomonas aeruginosa*	*Staphylococcus warneri*
Bacillus subtilus	*Klebsiella pneumonia*	*Pseudomonas putrefaciens*	*Streptococcus alcaligenes*
Citrobacter diversius	*Micrococcus luteus*	*Salmonella arizonae*	*Streptococcus faecalis*
Citrobacter freundii	*Moraxella phenylpyruvia*		

Table 2
Features used in pilot study

Catalase reaction	H_2S production	Methyl Red	Cell shape
Citrate utilization	Indole reaction	Oxidase reaction	Sucrose fermentation
Glucose fermentation	Lactose fermentation	Oxygen use	Urease reaction
Gram reaction	Mannitol fermentation		

of the feature values needed could be deduced from the existing tree structure, and the remainder were assumed to be available from the common literature. It should be noted that the real-life ID of the species used here requires a much more complex tree to differentiate them from the many species not included in this experiment.

Unfortunately, not all needed data were available. Obtaining 14 features for 26 separate microbes required a total of 364 pieces of information. The existing tree structure and a search of standard microbiology references yielded a total of 271 pieces of information (75%). The reason behind the unavailability of certain data appears to be that researchers have felt constrained by the existing tree. Some tests simply have not been performed on certain microbes because researchers have never had any reason to do so. Alternately, if the tests had been done, the results were not deemed relevant enough to be included in the standard reference texts. The missing data could have been collected by experimentation in a microbiology lab, but because this project was intended simply to determine the viability of the approach, work proceeded with the available data. C4.5, like most decision tree induction systems, has mechanisms to compensate for missing data. A new decision tree for the pilot data set was generated using an Ultra Sparc 1 Model 140 computer with 64 MB of memory running the Solaris 5.5 operating system.

3.2. Part 2 – Clostridium classification

Based on the knowledge gained in the initial pilot study, it was desirable to work with a larger data set. The *Clostridium* species were selected for the next phase due to the large volume of published information available, and the existence of an empirically created decision tree that could be used for benchmarking (Fig. 4). Note that the individual labels in this tree have been omitted for clarity, while the form remains for comparison.

This data set contained all the *Clostridium* species listed in Bergey's Manual of Systematic Bacteriology (Tansil, 1984) and all the features listed in the presumptive ID tree found there. A number of other features were also included in the data set. These were selected due to the availability of information and the utility of these features in general microbial ID tasks. A complete listing of features is shown in Table 3.

Some of the features included in the database, particularly some coming from the empirical tree, were 'sparse', meaning that the value of the feature could only be determined for a small percentage (less than 50%) of the species involved. Although unlikely to be useful, these features were included for completeness. Including the sparse features, the overall data density was 79.18%. Without them, data density increased to 94.72%.

The *Clostridium* study introduced a new issue that was not addressed in the pilot work. This was the area of multiple-valued features. Some microbes can have more than one legitimate outcome to certain tests, depending on the strain involved and the test conditions. Each such response required a split in the final dataset, with one instance containing the first response, and another instance the second. Two such features for the same microbe required a total of four instances to cover all possible combinations, and a microbe with s split features required $2s$ instances, assuming that all splits were only two-way. Several of the microbes involved had high numbers of split features. *Clostridium clostridoforme* was the most difficult, with 26 split features calling for $226 = 67{,}108{,}864$ instances.

Data were stored in a format recording the split feature values, with a preprocessing program to "unpack" the dataset into a full set of instances to be used with the decision tree software. Due to the extreme number of total instances in the data set (107,856,870), it was not possible to unpack the data on any available

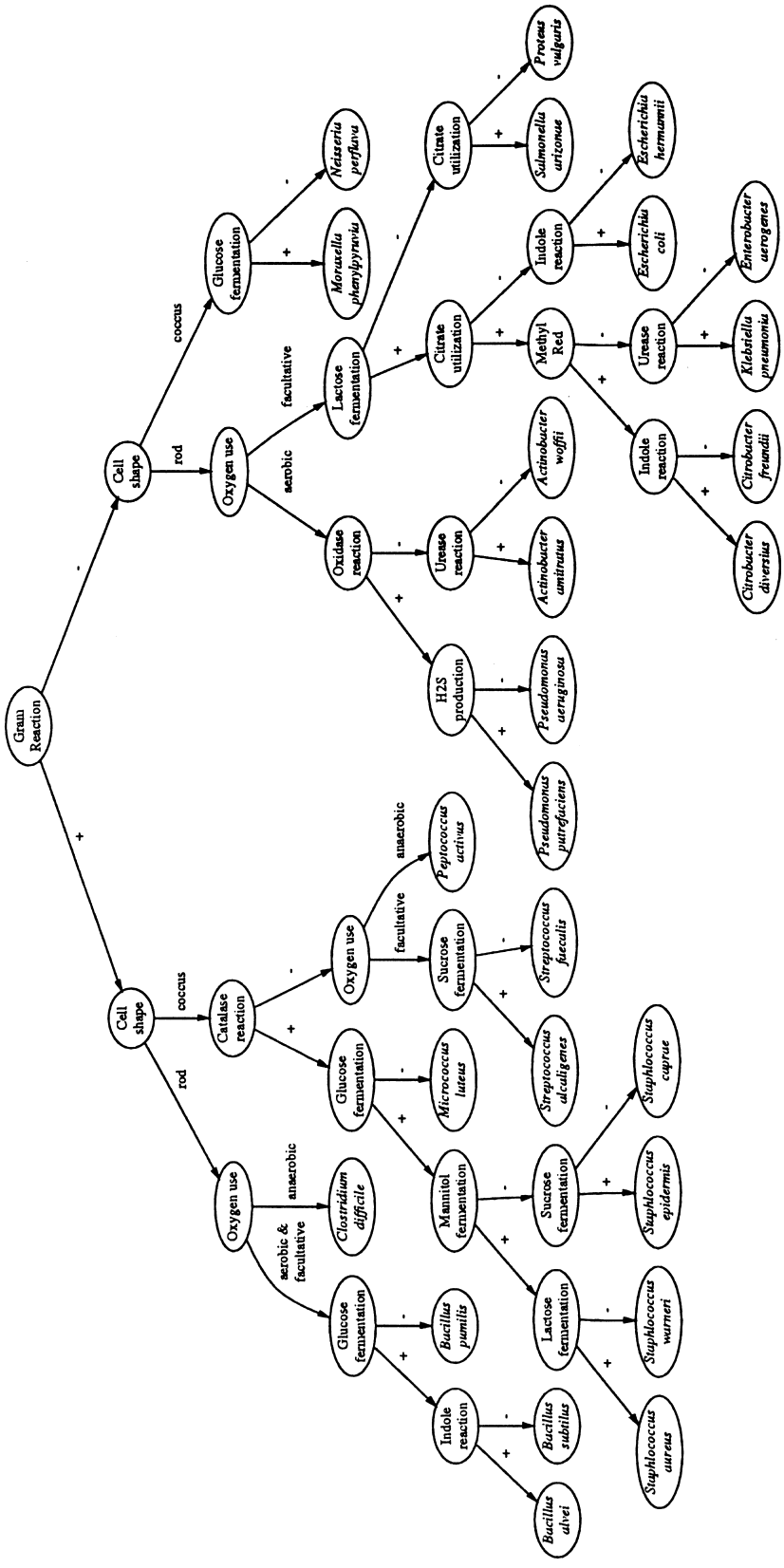

Fig. 3. Empirical tree from pilot study (produced from data and information in Villarreal & Sun, 1994).

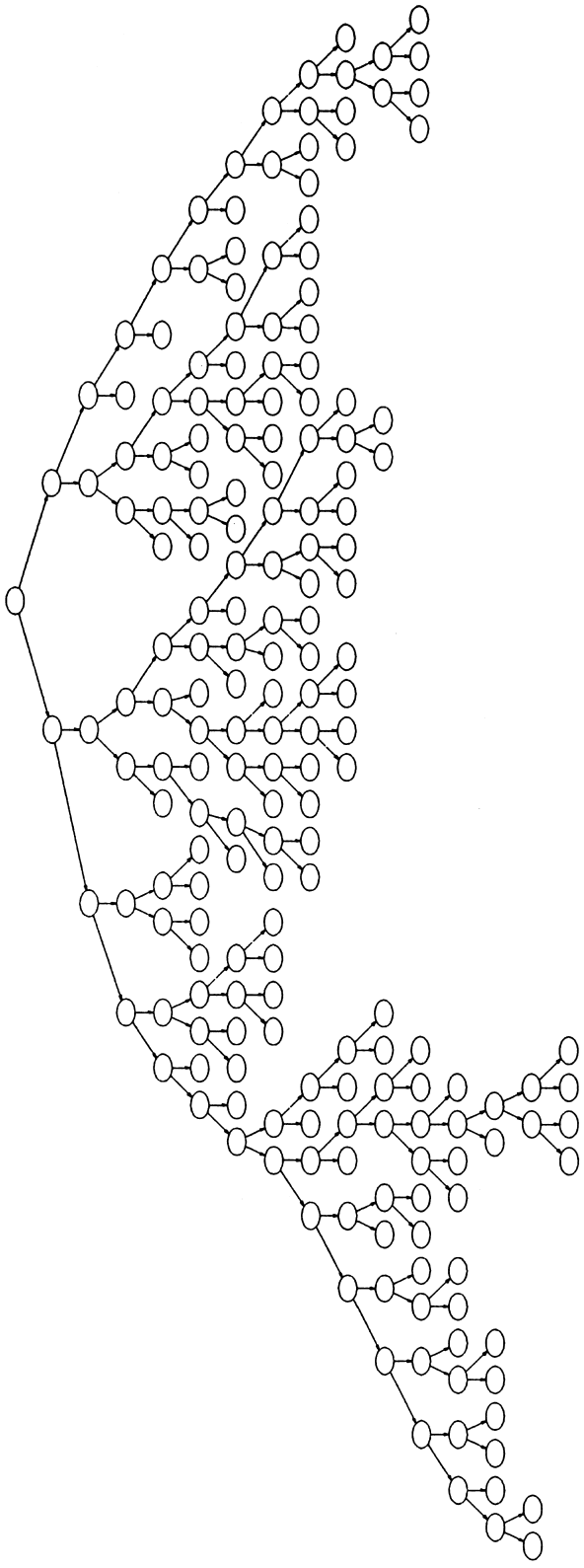

Fig. 4. Presumptive identification of *Clostridium* species (produced from data and information in (Tansil, 1984).

computer. The estimated size of the unpacked dataset was 18,203 MB (18 GB).

It was possible to reduce the size of the dataset to manageable proportions by either removing microbes or removing features. Removing microbes would have reduced the utility of the resulting classification tree, especially because some of the high-split features were among the more important in terms of food safety (e.g. *C. perfringens*). Removing features would have reduced the number of splits, and thus total instances, but it would also have reduced the number of meaningful features the decision tree induction system could choose from. Therefore, a compromise was used. The dataset was first reduced by removing the microbes with the most multiple-valued features until the projected data set size became small enough to be processed by a computer. This involved removing a total of 11 microbes (12.5% of the microbial species in the dataset), which reduced the number of total instances to 639,974 (a 99.4% reduction). A decision tree was then induced on this smaller (113 MB) data set. The features actually used in this tree were extracted (Table 4), and the original oversized dataset was again reduced by removing all features not selected from the smaller dataset. Data density on this final data set was 97.19%. A final decision tree for all *Clostridium* species was induced on this reduced data set. The hardware used was a 174 MHz R10000 IP32-based Silicon Graphics workstation model O2 with 128 megabytes of RAM running IRIX version 6.3.

4. Results and discussion

4.1. Part 1 – pilot study

The tree formed in the initial study is shown in Fig. 5. It is noticeably different from the standard decision tree shown in Fig. 3, but differentiates all the microbes correctly.

Most importantly, the new tree makes logical sense. The tests near the top of the tree are those usually used to separate large numbers of microbes, while the tests near the bottom of the tree are used to differentiate among relatively small groups of species within families. The tree starts off with the shape of the microbes, which is a fundamental feature. The two branches go to glucose fermentation, a basic metabolic reaction, and the catalase test, one of the simplest chemical tests. The Gram reaction, normally considered to be the primary dividing test, also figures highly in the new tree.

As was expected, the new tree is slightly smaller than the original. The average depth of the traditional tree is 5.23 tests, while the new tree averages 5.12 tests deep. Unfortunately, a *t*-test indicates that this difference is not statistically significant. This small and not significant improvement is probably due to the relatively small

Table 3
Features used in *Clostridium* study

AbundantgasinPYGdeepagar	AcidfromSorbitol[a]	Growthat22°C	Motility[a]
AbundantH_2fromPYG[a]	AcidfromStarch[a]	Growthat25°C	Neutralredreduced
AceticAcidfromPYG	AcidfromSucrose[a]	Growthat30°C	Nitrateredreaction
AcidfromAmygdalin	AcidfromTrehalose	Growthat37°C[a]	Nogrowth@25or37°C[a]
AcidfromArabinose	AcidfromXylose[a]	Growthat45°C	Optimumtemp22°C[a]
AcidfromCellobiose[a]	Ammoniaproduced	Growthat60°C	Optimumtemp25°C[a]
AcidfromFructose[a]	AtmosphericN_2fixed	Growthat70°C[a]	Optimumtemp60°C[a]
AcidfromGalactose[a]	Biotinonlyvitaminrequired[a]	Growthinaerobicbloodagar[a]	Producesacetate[a]
AcidfromGlucose[a]	ButanolfromPYG	H_2produced[a]	PropanolfromPYG
AcidfromGlycogen[a]	ButyricAcidfromPYG[a]	H_2Sproduced	PropionicAcidfromPYG[a]
AcidfromInositol	Canutilizemethanol[a]	Indoleproduced[a]	Resazurinreduced
AcidfromInulin[a]	CaproicAcidfromPYG[a]	Iso-acidsformed[a]	Salacinfermented[a]
AcidfromLactose[a]	Cellscoiled[a]	IsobutanolfromPYG	Specifictoxinantigen[a]
AcidfromMaltose[a]	CO_2fromPYG	IsobutyricAcidfromPYG	Sporeshape
AcidfromMannitol[a]	Colonypigmentsstable[a]	IsocaproicAcidfromPYG	Sporesswellcell
AcidfromMannose[a]	Esculinhydrolyzed[a]	IsopentanolfromPYG	Sporesterminalorsubterminal[a]
AcidfromMelezitose	EthanolfromPYG	IsovalericAcidfromPYG[a]	Starchhydrolyzed[a]
AcidfromMelibiose	Ethanolrequired[a]	LacticAcidfromPYG[a]	SuccinicAcidfromPYG
AcidfromRaffinose	Fermentablecarb.Needed[a]	Lecithinaseproduced[a]	Toxictomice[a]
AcidfromRhamnose	FormicAcidfromPYG[a]	Lipaseproduced[a]	Ureasepositive[a]
AcidfromRibose	Gelatinhydrolyzed[a]	Meatdigested	ValericAcidfromPYG
AcidfromSalicin	Growthat15°C	Milkreaction	

[a] Featuresusedintheempiricaltree.

Table 4
Features used to create final *Clostridium* tree

Abundant gas in PYG deep agar	Acid from Lactose[a]	Formic Acid from PYG[a]	Lactic Acid from PYG[a]
Acetic Acid from PYG	Acid from Maltose[a]	Gelatin hydrolyzed[a]	Lecithinase produced[a]
Acid from Amygdalin	Acid from Melibiose	Growth at 37°C[a]	Lipase produced[a]
Acid from Arabinose	Acid from Raffinose	Growth at 45°C	Milk reaction
Acid from Cellobiose[a]	Acid from Rhamnose	H_2 produced[a]	Motility[a]
Acid from Fructose[a]	Acid from Sucrose[a]	Indole produced[a]	Propionic Acid from PYG[a]
Acid from Galactose[a]	Butyric Acid from PYG[a]	Isobutyric Acid from PYG	Spores swell cell
Acid from Glucose[a]	Esculin hydrolyzed[a]	Isovaleric Acid from PYG[a]	Starch hydrolyzed[a]
Acid from Glycogen[a]			

[a] Features used in the empirical tree.

size of the pilot group, which did not allow the algorithm to select alternate features. Adding more common features would have given the algorithm greater flexibility. In other words, taking a decision tree and trying to reorganize it for greater efficiency using only the features already in the tree is very difficult. Having more features to choose from (or a larger set of classes and features) would make the task considerably easier. The process was further constrained by the missing data. It should be noted that the original tree is close to optimal within its original feature set. With 26 classes, the minimum average depth of tree would be 4.70 tests per class for purely Boolean features. One of the features here is a three-valued discrete attribute, which would make the best possible theoretical tree depth somewhat smaller.

4.2. Part 2 – Clostridium classification

The new decision tree created by c4.5 for the *Clostridium* study is shown in Fig. 6. Again, the individual labels have been omitted for clarity. Visual inspection shows that the new tree is smaller than the original empirical one. This observation is backed up by statistical analysis. The Bergey's tree takes an average of 8.67 tests to identify an unknown, while the new tree requires only 7.30. A *t*-test indicates that the difference is statistically significant at the 99.9% confidence level. In another important difference, the original tree uses 48 different features, while the new tree uses only 35. This means that a laboratory facility would not need to run so many different tests to complete an ID using the new tree.

Several other experiments relating to custom ID system design were performed with this set of data. Two other classification trees were created with different subsets of the data to test the system's ability to create classification schemes from specific groups of tests. One system was made using only the products produced from PYG as features. The other used a feature set composed only of data on acid production from starches. The PYG-only tree had performance comparable to the first

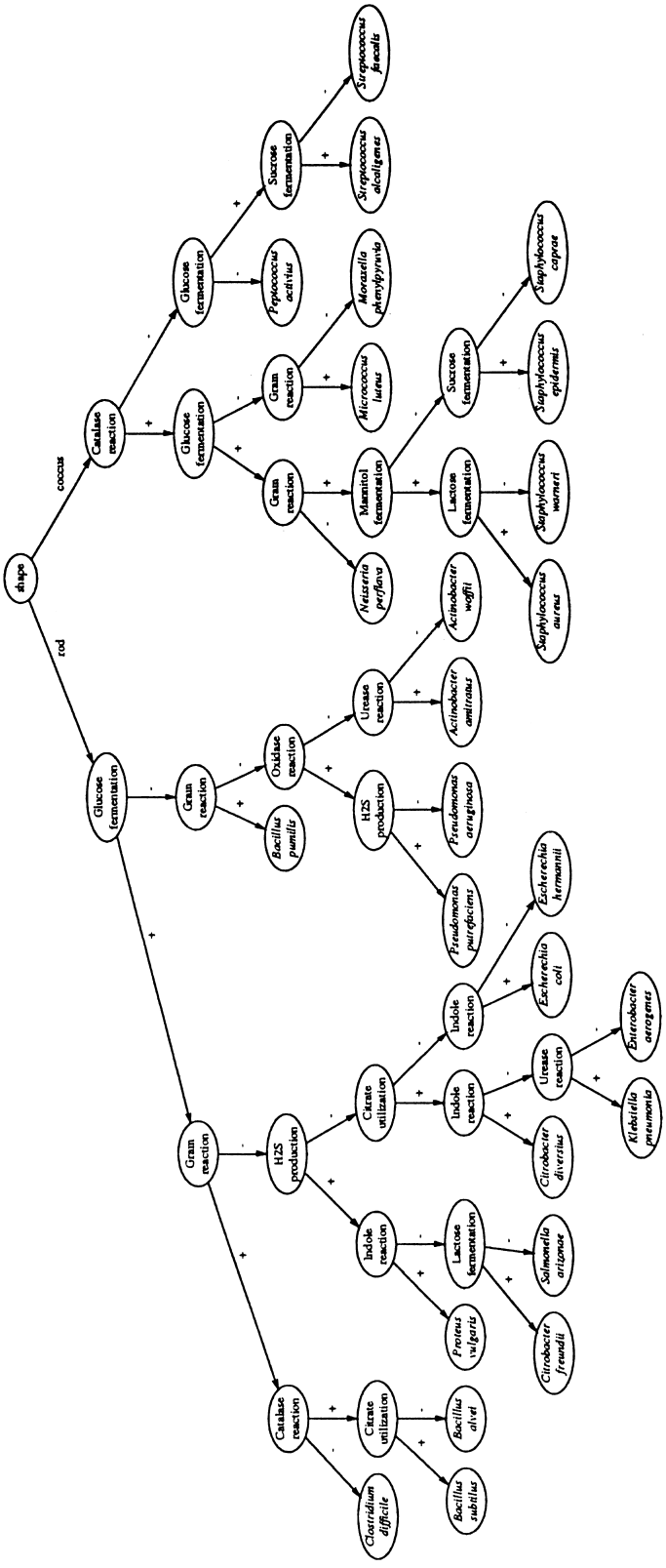

Fig. 5. Pilot study decision tree.

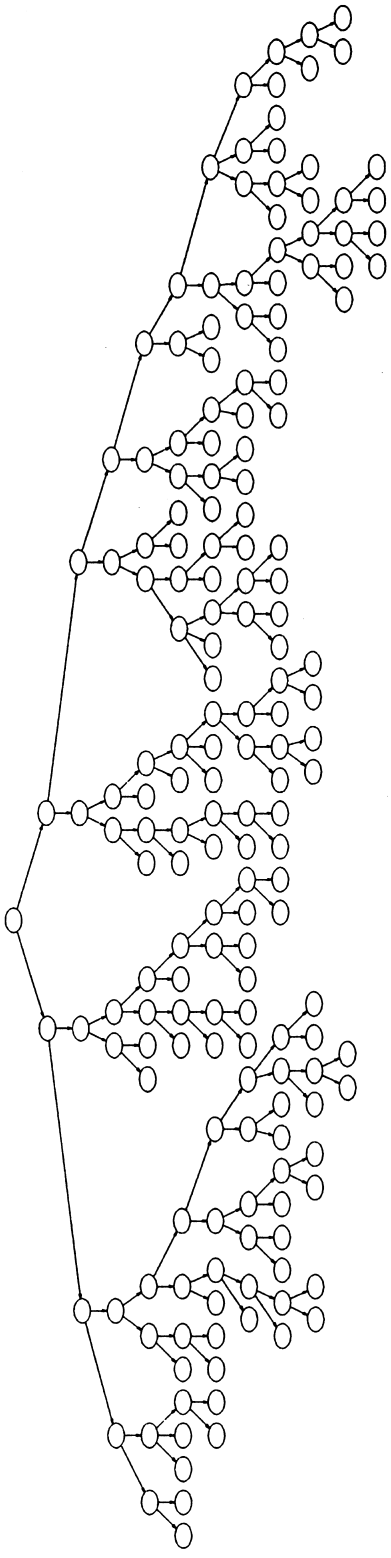

Fig. 6. New decision tree for *Clostridium* species.

c4.5 tree and a small predicted error rate, but the tree using acid production data was much larger than any of the others and was not useful for classification purposes. Given the success of the algorithm in creating classification systems under other circumstances, this would tend to indicate an insufficiency of relevant information in the acid production data subset.

4.3. Implications

The approach using the decision-tree creation system worked well. The biggest challenge to using it on a larger scale will be data collection and management. Until data on all microbes can be compiled, new machine learning-based microbial ID systems must be limited to certain subgroups of the original tree. This in essence substitutes new branches onto the stump of the empirical system. Such a method will still be highly useful in creating ID kits and other custom systems.

Further improvements could be made to the system by utilizing the knowledge that not all features are the same in microbiology. Tests vary widely from simple observations through a microscope, which can determine features such as cell size and shape, through various chemical reaction tests, to growth studies in which the organism's use of its environment and various growth media are determined. The time and the costs required for microbial tests can vary widely depending on the specific test, although they do tend to fall into groups of similar tests. The ideal tree, then, would be optimized not in terms of simple tree depth but in terms of average time or cost for ID, or preferably both. In some cases, industrial applications may minimize time requirements to improve overall food safety, while in other cases fiscal costs may be minimized.

A number of different metrics have been proposed (Norton, 1989; Nùñez, 1991; Tan & Schlimmer, 1989; Turney, 1995) for incorporating the feature cost into decision tree structures. All of these metrics use the cost of a test to weight the information gain or information gain ratio in some manner. These range from the simple (Norton, 1989)

$$\text{weighted information gain} = \frac{\text{information gain}}{\text{test cost}} \tag{5}$$

to the more complicated (Nùñez, 1991)

$$\text{weighted information gain} = \frac{2^{\text{information gain}} - 1}{(\text{test cost} + 1)^{\omega}}, \quad 0 \leqslant \omega \leqslant 1 \tag{6}$$

and would need to be tested over the specific domain to see which is most effective for this particular problem.

There are other important machine learning issues related to microbial ID. One of these comes from the

difference between selecting tests and selecting features. In most decision tree inductions, once a desired feature is identified, the associated test is performed. With microbial ID, certain features can be determined from more than one test, and certain tests can determine multiple features. To further complicate matters, portions of these two sets overlap. Therefore, once a given feature has been selected, it will also be necessary to specify the test that should be used to obtain that feature. This complication was not important in this project, but it may become crucial when building cost-based trees. Presumably, the several possible tests that could give a desired feature would all have different costs. Once such a test had been performed to generate one feature, the other features given by that test would be available for little or no additional cost. Turney (1995) solves the multiple-feature problem by using a hybrid genetic decision tree algorithm that makes test costs interdependent within groups of interacting features or tests, but does not address the selection problem for features that can be determined by more than one test.

Another potential optimization concern is the distribution of occurrences of various microbes in the real world. Certain organisms are much more common than others. Therefore, truly efficient trees should move toward rapid (or inexpensive) ID of the most commonly occurring microbes. It might also be desirable to produce a tree that was biased to rapidly identify the most important pathogenic microbes, such as the FDA's "Bad Bug List" of major foodborne pathogens. In either case, a preference mechanism must be created to cause the tree to classify certain microbes most cost-effectively while still maintaining full accuracy over the entire domain. Finally, it might be advantageous to create specialized trees for different source areas. These could be made for specific food products, or other specialized areas in fields such as medicine or waste processing.

5. Conclusions

Based on the significant reduction in tree depth demonstrated for the *Clostridium* species, it is reasonable to conclude that the software tool developed in this study is successful in improving the efficiency of identification schemes. The conditions necessary for such improvement include a larger variable set than the original and as high a data density as possible. The software developed was also successful in creating specialized ID systems from subsets of the data.

Acknowledgements

The authors would like to thank Dr. Rick Millane of the Whistler Center for Carbohydrate Research, Food Science Department, Purdue University for the use of their computer facilities; and Dr. Carla Brodley and Mr. Jeffrey Bradford of the School of Electrical Engineering, Purdue University for their assistance in providing computer access and processing scripts. This research effort was partly supported by a grant from USDA (USDA National Needs Fellowship #94-38420-0972).

References

Kononenko, I., Bratko, I., & Roskar, E. (1984). Experiments in automatic learning of medical diagnostic rules (technical report). In J. R. Quinlan, *Induction of decision trees* (pp. 81–106). Jozef Stefan Institute, Ljubljana, Yugoslavia, 1986. Machine Learning 1. Boston: Kluwer Academic Publishers.

Norton, S. S. W. (1989). Generating better decision trees. In *Proceedings of the 11th international joint conference on artificial intelligence, IJCAI-98* (pp. 800–805). Detroit, MI.

Nùñez, M. (1991). *The use of background knowledge in decision tree induction* (pp. 231–250). Machine Learning 6. Boston, MA: Kluwer Academic Publishers.

Quinlan, J., (1986). *Induction of decision trees* (pp. 81–106). Machine Learning 1. Boston: Kluwer Academic Publishers.

Quinlan, J., (1991). *C4.5: Programs for machine learning*. San Mateo, California: Morgan Kaufmann.

Tan M., & Schlimmer, J., (1989). Cost-sensitive concept learning of sensor use in approach and recognition. In *Proceedings of the sixth international workshop on machine learning, ML-89* (pp. 392–395). Ithaca, New York.

Tansil, B. (ed.), (1984). *Bergey's manual of systematic bacteriology*, vols. I & II. (8th ed.). Baltimore, Maryland: Williams & Williams.

Turney, P. D., (1995). Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree algorithm. *Journal of Artificial Intelligence Research 2, 369–409*. Morgan Kaufmann, San Mateo, California.

Villarreal, D. T., & Sun, I. L. (1994). *Biology 221L (Introduction to microbiology) lab manual*. Department of Biological Sciences, Purdue University, West Lafayette, Indiana.