AIAA JOURNAL
Vol. 31, No. 9, September 1993

Sparse Distributed Associative Memory for the Identification of Aerospace Acoustic Sources

E. A. Scott,* C. R. Fuller,† W. F. O'Brien,‡ and R. H. Cabell§
Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061

A pattern recognition system has been developed to classify five different aerospace acoustic sources. The system consists of one microphone for data acquisition, a preprocessor, a feature selector, and a classifier. In this paper the performances of an associative memory classifier and a neural network classifier are compared with the performance of a previously designed system. Source noises are classified using features calculated from the time and frequency domain. Each classifier is trained to classify source noises correctly using a set of known sources. After training, the classifier is tested with unknown sources. Results show that over 96% of the sources were identified correctly with the new associative memory classifier. The neural network classifier identified over 81% of the sources correctly.

Nomenclature

d_j = desired output for node j
$d(y)$ = decision function
e_j = excitation of jth processing element
t = time
u_j = input feature or activation of node on previous layer
w_{ij} = weight connecting nodes i and j
x = known pattern vector
x_i = ith feature of known pattern vector
y = unknown pattern vector
y_i = ith feature of unknown pattern vector
z_j = actual output for node j
δ_j = error term for node j
θ_j = threshold of node j
ω_n = class n

Introduction

IN recent years an increased volume of air traffic at airports has significantly increased noise annoyance, as well as the difficulty in monitoring the offending noise sources. Therefore, a system that could automatically identify aerospace acoustic sources would be useful for monitoring airport noise. The system must be able to distinguish between surface and aerospace vehicles. Potentially, this system could monitor airport noises and identify excessively noisy aircraft.

An automatic pattern recognition system was previously developed[1,2] to identify jets, propeller planes, helicopters, wind turbines, and trains. This original system was based on linear discriminant functions and successfully identified the five sources about 91% of the time. The present paper discusses improvements to the original pattern recognition system[1,2] by the use of an associative memory classifier based on sparse distributed memory[3,4] and a classifier based on artificial neural networks.[5,6] These classifiers use sophisticated pattern recognition techniques to classify the different acoustic sources. Each system is simulated with a Fortran program and tested with data from recordings of the acoustic signatures of the five vehicles.

Presented as Paper 90-3992 at the AIAA 13th Aeroacoustics Conference, Tallahassee, FL, Oct. 22–24, 1990; received Sept. 17, 1991; revision received Oct. 9, 1992; accepted for publication Oct. 20, 1992.

*Graduate Project Assistant, Department of Mechanical Engineering.
†Professor, Department of Mechanical Engineering. Associate Fellow AIAA.
‡Associate Dean, College of Engineering.
§Research Associate, Department of Mechanical Engineering.

Pattern recognition systems (PRSs) have many applications, from reading the account numbers on checks to classifying handwritten characters.[7,8] PRSs have also been used for medical research[9] and nondestructive evaluation of various structures.[10,11] PRSs can be used for identifying different acoustic sources[1,2,12,13] as well as source types, such as different helicopters.[14]

The classifiers presented in this paper are designed to classify an unknown aerospace acoustic source as being either a jet plane, propeller plane, helicopter, train, or wind turbine, using data gathered from a single microphone. Given the complex nature of the classes under consideration, the pattern classifiers are trained to recognize the sources by presenting known patterns and allowing the classifiers to learn to identify these patterns correctly. Different training algorithms available for each classifier will be discussed in later sections.

Pattern Recognition Theory

In pattern recognition theory there are three basic methods of classification[5]: membership roster, common property, and clustering. The membership-roster method compares an unknown pattern with known training patterns. The unknown pattern is classified if it is sufficiently similar to one of the known patterns, as shown in Fig. 1. An unknown pattern that is corrupted by noise or that has been scaled to a different size may be incorrectly classified using this method.

The common-property and clustering methods do not identify the entire pattern but instead identify a set of features that quantifies characteristics of the pattern. Several features are chosen to describe the patterns, and these features are assembled into a vector. The resulting feature vector reduces the degrees of freedom of the problem and reduces the sensitivity of the classifiers to insignificant variations in the patterns.

The common-property method compares the feature vector of an unknown pattern y to the feature vector of each known pattern x. The class of an unknown pattern is the same as the class of a matching known pattern.

The clustering method uses the n-element feature vector to locate a point corresponding to the pattern, or noise source for the present work, in n-dimensional space. Patterns of the same class should cluster in the same compact region of n-dimensional space if the features are chosen well. Figure 2 shows an example of class clusters in a two-dimensional feature space, where each axis corresponds to one of the features. The clustering method is least sensitive to variations in the patterns; hence it is the basis for all three classifiers considered in this work.

Class clusters are generated using a set of training patterns of known class membership. After the class clusters are

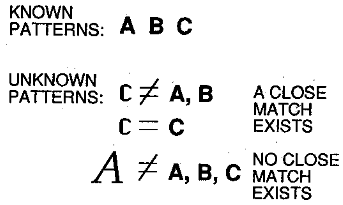

Fig. 1 Membership-roster classification.

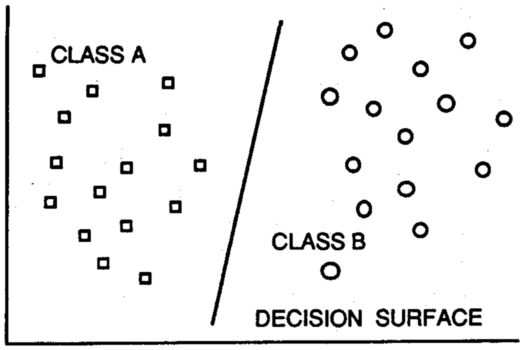

Fig. 2 Class clusters.

formed, the class of an unknown pattern can be determined. One method for identifying unknown patterns uses discriminant functions, also called decision surfaces, that are placed between the class clusters. The class of an unknown pattern is determined by its position relative to the decision surface. For example,

$$y \in \omega_1 \quad \text{if} \quad d(y) > 0 \quad \text{or} \quad y \in \omega_2 \quad \text{if} \quad d(y) < 0 \tag{1}$$

In the example shown in Fig. 2, a straight line or linear discriminant function completely separates the two clusters. Meshed clusters, shown in Fig. 3, are not linearly separable and must be separated with a nonlinear discriminant function. Artificial neural networks, described later in this paper, can form nonlinear separating surfaces.[5,6]

Distance functions can also be used for determining the class of an unknown pattern. Each class is represented by a prototype, and the class of an unknown pattern is determined by finding the closest prototype or prototypes.[3] The associative memory classifier discussed later is based on the distance function method.

System Design

A PRS consists of several parts as shown in Fig. 4. The first part is data acquisition, where properties of a pattern are measured by an appropriate method, such as a recording of the acoustic pressure at a microphone. The measured data are then preprocessed, filtered, digitized, and transformed as needed. In the current work, the output of a single, ground-based microphone was low-pass filtered at 2.5 kHz to reduce aliasing and to eliminate high frequencies that are less useful for identifying vehicle noise. The signal was digitized at a convenient sampling rate of 7812.5 Hz. The sampled data were broken into blocks of 4096 points, each of which contained about one-half second of sampled data. The source noise was considered to be stationary for this half-second period. Each block of data was then transformed into the frequency domain to produce the power spectral density (PSD) of the signal.

Three types of features were calculated from the PSD to describe the distribution of power as a function of frequency. Partial powers, ratios of partial powers, and accumulated power points were calculated.[10] Partial powers describe the percentage of power in a large band, such as 0–500 Hz, present in a portion of that band, such as 0–150 Hz. Ratios of partial powers describe the relationship between two partial powers. An accumulated power point is the frequency at which some percentage of the power in a band is located, when the power is accumulated up from the bottom of the band. For example, an accumulated power point could be used to describe the frequency at which 50% of the power in the 0–2500-Hz band is located, when the power is accumulated up from 0 Hz. The frequency ranges and percentage points were varied to produce 105 features. Two other features were calculated from the autocorrelation, and one feature described the number of peaks exceeding a threshold in the PSD, for a total of 108 features.

A large number of features could be calculated for the current problem; however, a small set of features is better because it has less redundancy among features and has reduced computational requirements.[15] A general purpose reduction intensive (GPRI) feature selector was used to select a good set of features from the 108 candidate features.[15] The feature selector tested many combinations of features using two small sets of data: one set to train a classifier and another to evaluate the performance of the classifier.

The feature selector optimized the classifier performance for a given number of features, which was varied from 4 to 10 features.[15] The best set of features was a function of the number selected from the candidate list. For example, the best set of five features chosen using the GPRI algorithm may or may not contain the same features as the best set of four features.

The fourth element of a PRS is the classifier that identifies a pattern as belonging to one of the pretrained classes.

Associative Memory Classifier

The first classifier discussed in this paper is based on an associative memory that recalls the contents of a memory location when an address "close to" that location is queried.

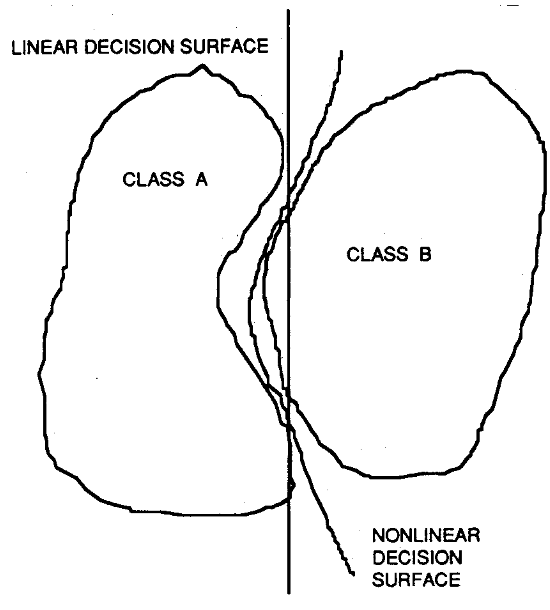

Fig. 3 Meshed clusters.

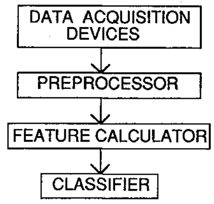

Fig. 4 Pattern recognition system components.

For the current application, the memory locations correspond to stored training patterns; hence an unknown pattern is classified based on the closest training pattern, much like a distance function method. A memory location or address can be noisy or have small errors and still recall the correct data in an associative memory, unlike in the random access memory of a computer.[3]

Memory locations are specified by n-bit addresses, where the address of a pattern is the feature vector of that pattern. The class of an unknown pattern is found by examining the classes of patterns with nearby addresses, as shown in Fig. 5. In a sparse distributed memory (SDM), the memory locations do not usually correspond to the exact locations of the training patterns but instead are sparsely distributed throughout the feature space.[3,4] The class of a training pattern is stored at memory locations near the address of that training pattern. In the current work, the SDM implementation was modified such that the memory locations did correspond to the addresses of the training patterns.

The SDM algorithm can be implemented in the structure shown in Fig. 6 to take advantage of parallel processing.[4] The structure consists of an input layer containing the feature vector of a pattern, a hidden layer containing one processing unit for each memory storage location, and an output layer. Each unit in the output layer corresponds to one of the pattern classes.

To store or recall the class of a pattern from the memory, memory storage locations near the address of that pattern are first activated. The feature vector of that pattern is applied to the inputs of the SDM, and the activation or excitation level of each memory storage location is calculated,

$$e_j = \sum_N \frac{1.0}{1.0 + \mathrm{abs}(x_i - y_i)} \tag{2}$$

The greater the excitation, the closer the memory location is to the input pattern. All of the memory locations with an excitation greater than a threshold are activated. The weights between the activated memory locations and the outputs are then summed and thresholded. The output unit with the highest value corresponds to the class of the input pattern.[3,4]

SDM training requires only two passes through the training data. First, each training pattern is assigned to a hidden unit. The feature vector of each training pattern is stored in the weights between the inputs and the hidden unit assigned to the pattern. After all of those weights are set, the entire set of training patterns is presented a second time to set the weights between the hidden units and the output units. Each training pattern is input to the SDM, activating a number of hidden units. The weights between active hidden units and the output unit corresponding to the class of the training pattern are incremented, whereas the weights between activated hidden units and output units corresponding to other classes are decremented.

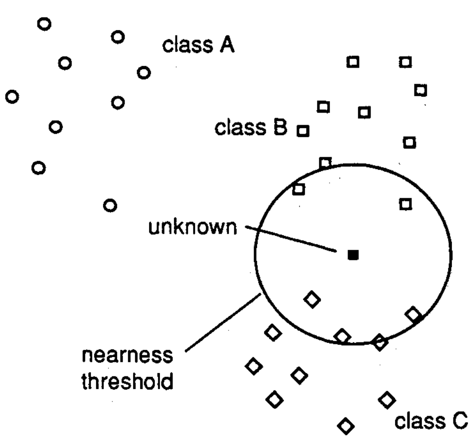

Fig. 5 Memory locations in n-dimensional space.

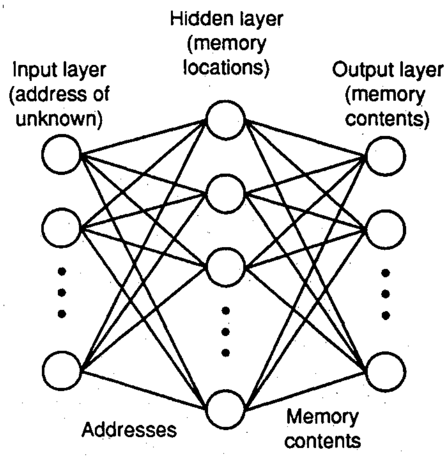

Fig. 6 Parallel structure for an SDM classifier.

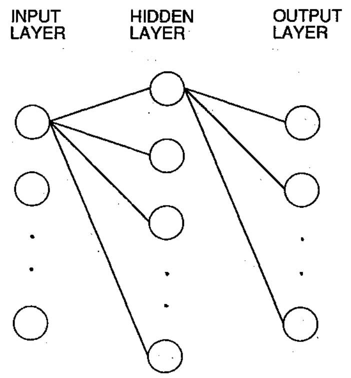

Fig. 7 Feed-forward neural network.

A set of 126 recordings was available for training and testing the system. The set contained recordings of 69 jets and 25 propeller planes, 9 recordings of a helicopter, 11 recordings of a train, and 12 recordings of a wind turbine. The propeller planes were recorded both landing and taking off.

The classification system was tested in two stages to achieve maximum use of the noise recordings. First, the classifiers were trained and evaluated using a leave-one-out procedure. For this procedure, 10 patterns were chosen from each class except the helicopter, which had only 9 patterns. Each classifier was trained with 48 of those patterns and then identified the 49th pattern. This procedure was repeated 49 times. Next, the system was trained with all 49 patterns, then tested with the remaining 77 patterns. The performance of each classifier is given as the total percentage of test patterns correctly identified by each classifier.

Neural Network Classifier

The second classifier is an artificial neural network that generates nonlinear surfaces to separate the class clusters.[5,6] The structure of a multilayer feed-forward neural network is shown in Fig. 7. The network consists of an input layer, a hidden layer, and an output layer, all of which are connected by weights. Although the structure of the neural network is similar to the parallel SDM, the computations are very different. The input units of the network are set to the feature vector of the current pattern, and each hidden and output unit computes a linear summation,

$$e_j = \sum_i w_{ij} u_{ij} \tag{3}$$

which is then passed through a squashing or sigmoid function to produce an output a_j

$$a_j = \frac{1.0}{1.0 + \exp[-(e_j - \theta_j)]} \tag{4}$$

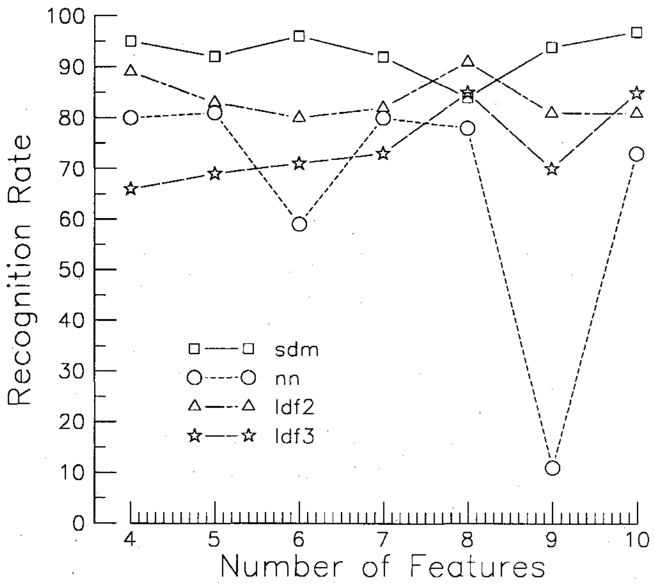

Fig. 8 Combined class results.

The nonlinear squashing functions in each hidden and output unit allow the network to form nonlinear surfaces to separate meshed clusters, like those shown in Fig. 3.

The weights in the neural network are calculated using a gradient descent training algorithm called the backpropagation algorithm.[5,6] Backpropagation calculates incremental weight corrections to reduce the error between the actual and the desired outputs of the network.

At the beginning of training, the weights in the network are randomly initialized with small values. The feature vector of a training pattern is set at the network input, the network output is computed, and an error is calculated between the output of the network and the desired output. The error between the network output z_j and the desired output d_j is used to modify the weights to reduce the error. Using Eq. (5), the new value for a weight w_{ij} is

$$w_{ij}^{t+1} = w_{ij}^{t} + \eta\delta_j u_i + \alpha(w_{ij}^{t} - w_{ij}^{t-1}) \tag{5}$$

where η and α are training parameters.[5] The term α is called a momentum term, set to 0.7 for training. The term η is a gain term set to 0.5 for the hidden unit to output unit weights and 0.7 for the input to hidden unit weights. The superscript t denotes the iteration number; hence w_{ij}^{t-1} is the value of weight w_{ij} at the previous pattern presentation. The term δ_j is called the error gradient for unit j. The error gradient for the hidden units to the output units is

$$\delta_j = z_j(1.0 - z_j)(d_j - z_j) \tag{6}$$

For the weights connecting input units and hidden units,

$$\delta_j = u_j(1.0 - u_j) \sum_k \delta_k w_{jk} \tag{7}$$

In Eq. (7), δ_j is a function of the error gradients δ_k in the next highest layer, hence the name backpropagation of errors.[5,6]

The weights are incrementally adjusted after each training pattern using Eq. (5), so several passes through the training data are usually required before the output error is significantly reduced. The momentum and gain terms affect the speed and stability of convergence.

Backpropagation learning is most efficient if the training set contains the same number of patterns per class and those patterns are presented in alternating order by class. The training set described in the previous section contained 10 jets and 5 each of the other sources, so the other sources were presented to the network twice as often during training. The neural network was tested with the remaining 96 sources after 100 passes through the entire training set, at which point the mean square output error was approximately one-fourth of the error at initialization. A threshold was used for interpreting the output of the neural network since the network was trained for such a short time. If the output of a unit was greater than 0.5, then it was considered high; if it was less than 0.5, it was considered low. A pattern was classified according to the unit with the high output.

The neural network required much longer training times and more training data than either the SDM or linear discriminant classifiers, and so the network performance was not optimized as the other classifiers were. In particular, feature selection was not done with the neural network; instead the features selected during SDM training were used. Additionally, the number of hidden units in the network was determined using a simple trial and error method instead of using a more thorough optimization process. The number of hidden units in the network varied from 12 units for the 4-feature case to 24 units for the 10-feature case.

Linear Discriminant Function Classifier

The third classifier discussed in this paper is a linear discriminant function (LDF) classifier, previously applied to this problem.[1,2] The training and testing of the classifier is described in Refs. 1 and 2. The classifier separates clusters in n-dimensional space using linear surfaces, described by Eq. (8),

$$d(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + w_{n+1} \tag{8}$$

Two configurations of the linear classifier were tested in the original work. The first variation, called LDF2 in the figures, used a tree classification structure. The tree structure consisted of several layers of cascaded classifiers; on the first layer a pattern was classified as either a ground source or an aerospace source. If classified as an aerospace source, the pattern was then classified as either a fixed-wing aircraft or a helicopter. If classified as a fixed wing, the pattern was then classified as either a jet plane or a propeller plane. If first classified as a ground source, the pattern was then classified as either a train or a wind turbine. The other linear classifier, called LDF3, used a single level classifier that immediately classified a pattern into one of the five source classes. The structure of the LDF3 classifier was similar to the structure of the associative memory and neural network classifiers.

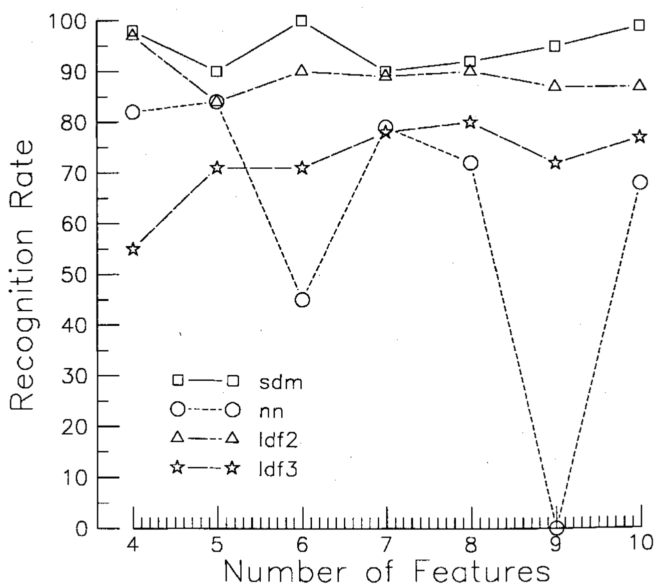

Fig. 9 Jet results.

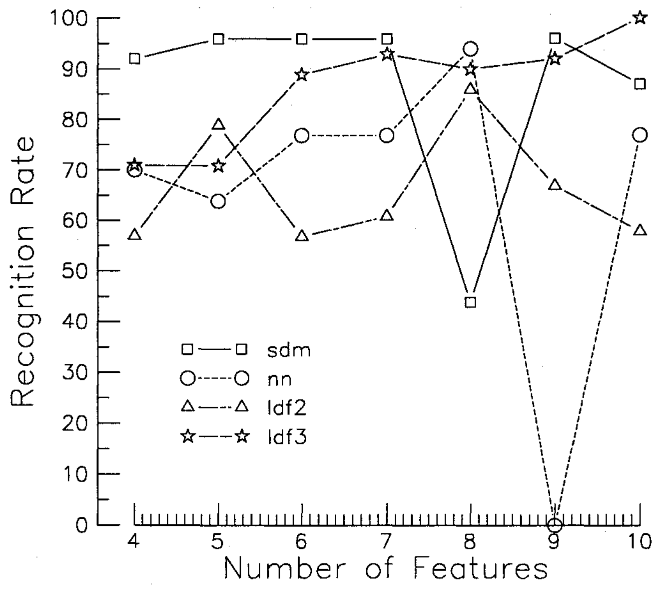

Fig. 10 Propeller plane results.

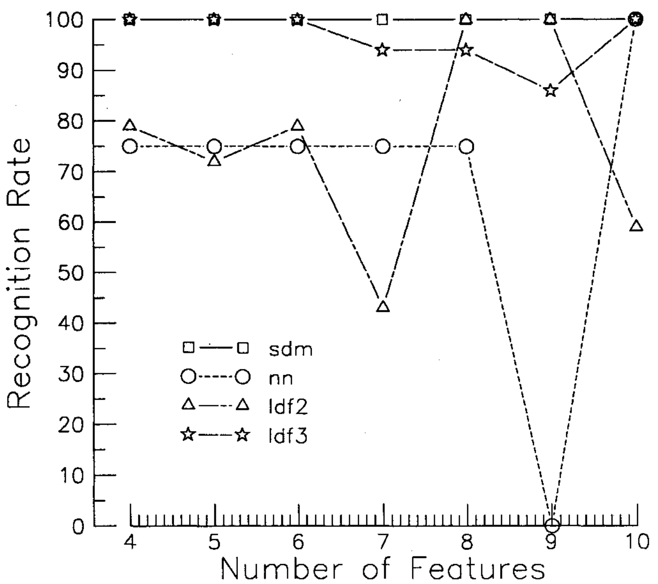

Fig. 11 Helicopter results.

Results

Combined Classes

The SDM recognized a higher percentage of patterns from all five source classes than any of the other classifiers, as shown in Fig. 8. The plot in Fig. 8 indicates the percentage of test patterns correctly identified as a function of the number of features. The recognition rate of the SDM classifier ranged from 92.1 to 96.8% correct and dropped slightly when using eight features. The drop in recognition rate will be discussed in the section on recognition of propeller planes.

The performance of the neural network was more sporadic than the SDM on all of the source classes, with the percentage of the correct recognitions ranging from 11.5 to 81.3%. The wide range of recognition rates could have been caused by insufficient training time and poor feature selection. In spite of this drawback, the neural network still outperformed LDF3 in three of the seven cases.

Jets

The SDM recognized between 89.9 and 98.6% of the jets, as shown in Fig. 9, exceeding the performance of the other classifiers in all of the cases. These results are very encouraging because of the very large number and diversity of recordings of jet planes. The results are not as good for the neural network classifier, ranging from a high of 84.7% correct with five features to a low of 0.0% correct with nine features. Again, the wide range of rates is probably due to incomplete training and poor feature selection.

Propeller Planes

The results for the propeller planes are shown in Fig. 10. The performance of the associative memory classifier exceeded that of the linear discriminant classifiers except for the 8-feature case. Using practical knowledge gained during the experiment, eight alternative features were selected and the associative memory classifier retested. The recognition rate improved from 44% correct to 88% correct, demonstrating the sensitivity of classifier performance to the feature set.

Classifier performance is also sensitive to variations in patterns not accounted for in the training set. For example, the SDM was trained with takeoff recordings of propeller planes, then tested with the remaining takeoff and landing recordings. The signature of a plane will vary depending on whether or not it is ascending or descending. The SDM correctly classified 89.3% of the takeoffs and 83.3% of the landings, which demonstrates that the SDM is robust to some variations in source characteristics.

Although the performance of the neural network was again sporadic, the network did outperform the SDM for the 8-feature case. This difference in recognition rates is significant because the classifiers used the same eight features but achieved very different recognition rates.

Helicopters

Recognition rates for the helicopters are shown in Fig. 11. The SDM classifier correctly identified all of the helicopter patterns regardless of the number of features. The neural network was not as consistent, correctly identifying 100% of the patterns for the 10-feature case but not recognizing any for the 9-feature case. The performance of both classifiers was most likely affected by the small number of helicopter recordings available for training and testing.

Trains

The associative memory classifier had its worst performance classifying trains, as seen in Fig. 12. The recognition rate reached 100% for the 8- and 10-feature cases but dropped to 81.8% for all of the other cases. For the cases in which only

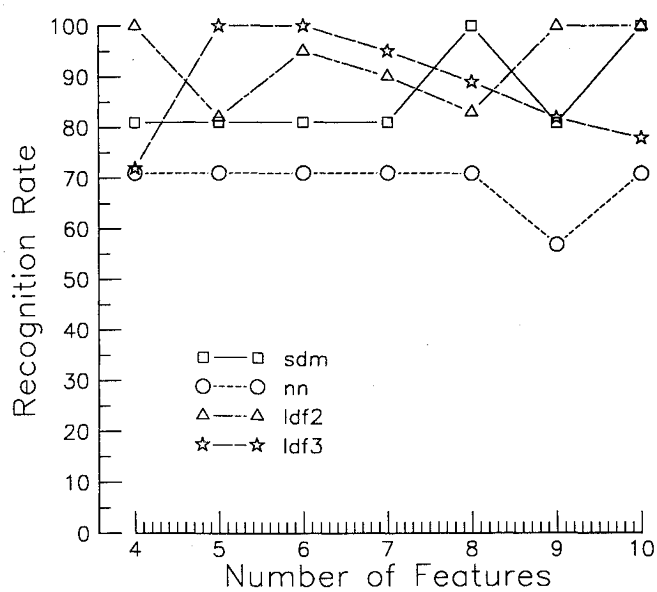

Fig. 12 Train results.

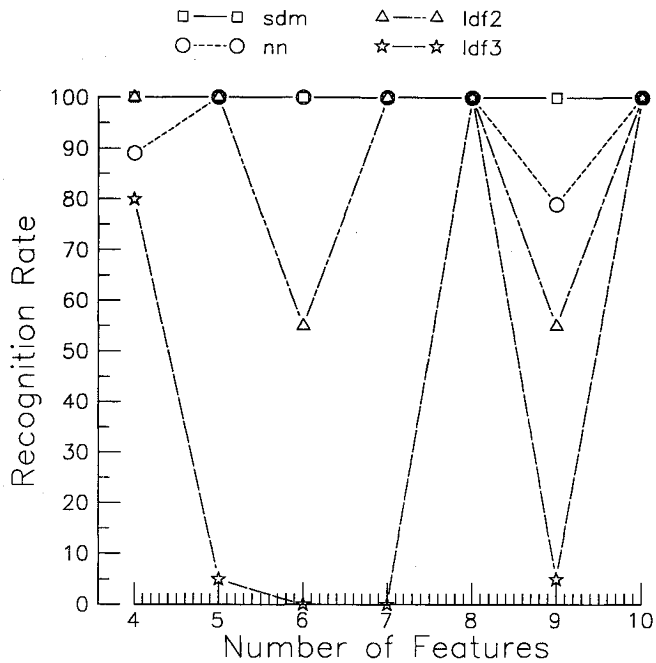

Fig. 13 Wind turbine results.

81.8% of the trains were correctly identified, the classifier misidentified the same set of train recordings, indicating that those recordings did not cluster near the rest of the train recordings but instead formed their own cluster when certain features were used for identification. The performance of the associative memory was not as good as the performance of either LDF2 or LDF3.

The recognition rates for the neural network were also worse than LDF2 and LDF3. As with the helicopters, the recognition rate was affected by the small number of recordings available for training and testing.

Wind Turbines

The associative memory classifier is a definite improvement on the linear discriminant classifiers for identifying wind turbine recordings, as seen in Fig. 13. The SDM correctly recognized all of the patterns regardless of the number of features used. The neural network also outperformed the two linear discriminant function classifiers.

Conclusions

Associative Memory Classifier

An associative memory pattern recognition system was designed to identify the sounds of jets, propeller planes, helicopters, trains, and wind turbines. The memory classifier was studied as an improvement to a previously designed and tested linear discriminant function classifier.

1) The associative memory classifier is capable of classifying over 96% of the sources correctly. The linear discriminant function classifier could only identify 91% of the sources correctly.

2) The memory classifier can generalize well to novel patterns. The system classified at least 90% of the jets correctly after being trained with samples from all nine types of jets. Patterns generated from takeoff recordings of the propeller planes were correctly identified 89.3% of the time, and those taken from landing recordings were correctly identified 83.3% of the time, even though the system was trained only with patterns from takeoff recordings. The linear discriminant function classifier, which classified 91% of the takeoff recordings correctly, only classified 37% of the landing recordings correctly.[1,2]

3) The memory classifier had trouble identifying certain patterns regardless of the features used. For example, one jet recording was consistently classified as a propeller plane, presumably because the pattern from this recording was not located near a cluster of other jet patterns.

4) Other sources were easily classified because their acoustic signatures are very different from the other source classes but do not vary much among themselves. The wind turbines, for example, form a compact cluster, separated from the other class clusters.

5) The feature selector works well; however, the chosen feature set is not necessarily optimal, as demonstrated by the drop in performance for the memory classifier using eight features. Overall, the number of features did not significantly effect the performance of the classifiers, indicating the features and the selector need to be optimized.

Neural Network Classifier

The neural network classifier was also studied as an improvement on the linear discriminant function classifier. Although the neural network could not identify as many patterns as the SDM, certain conclusions can be drawn.

1) The performance of the neural network system was adversely effected by incomplete training, as demonstrated by the drastic changes in performance as the number of features changed. This occurred because a neural network trains at different rates, depending on the number of features and on the initial weight values, so some networks converge more quickly than other networks.

2) The feature sets that work well for one type of classifier will not necessarily work well for another type. This is illustrated in Fig. 10. With both classifiers using the same eight features, the neural network outperforms the associative memory classifier by almost 50%, but with nine features the memory outperforms the neural network by over 90%.

3) Further work should be done with the neural network classifier to study the effects of training time, feature selection, and a more thorough testing process.

Acknowledgment

The authors gratefully acknowledge the financial support of this work by the Applied Acoustics Branch at NASA Langley Research Center under task NAG1-762.

References

[1]Cabell, R. H., and Fuller, C. R., "A Smart Pattern Recognition System for the Automatic Identification of Aerospace Acoustic Sources," AIAA Paper 89-1114, April 1989.

[2]Cabell, R. H., "The Automatic Identification of Aerospace Acoustic Sources," Master's Thesis, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, Feb. 1989.

[3]Kanerva, P., *Sparse Distributed Memory,* MIT Press, Cambridge, MA, 1988.

[4]Rogers, D., "Kanerva's Sparse Distributed Memory: An Associative Memory Algorithm Well-Suited to the Connection Machine," Research Inst. for Advanced Computer Science, RIACS-TR-88.32, NASA Ames Research Center, Moffett Field, CA, 1988.

[5]Lippman, R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine,* Vol. 4, No. 2, April 1987, pp. 4–22.

[6]Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Foundations,* edited by D. E. Rumelhart and J. L. McClelland, Vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318–364.

[7]Tou, J. T., and Gonzalez, R. C., *Pattern Recognition Principles,* Addison-Wesley, Reading, MA, 1974.

[8]Tappert, C. C., "Cursive Script Recognition by Elastic Matching," *IBM Journal of Research and Development,* Vol. 26, No. 6, 1982, pp. 765–771.

[9]Moser, J. M., and Aunon, J. J., "Classification and Detection of Single Evoked Brain Potentials Using Time-Frequency Amplitude Features," *IEEE Transactions on Biomedical Engineering,* Vol. BME-33, No. 12, 1986, pp. 1096–1106.

[10]Chan, R. W. Y., and Hay, R. D., "A Case Study of Sensitivity of

Some Pattern Classifiers Used in Sorting Acoustic Emission Signals," *Review of Progress in Quantitative Nondestructive Evaluation,* Vol. 1, Plenum Press, New York, 1982, pp. 413–419.

[11]Haran, S., Finch, R. D., and Jansen, B. H., "Application of an Automated Package of Pattern Recognition Techniques to Acoustic Signature Inspection of Railroad Wheels," *Journal of the Acoustical Society of America,* Vol. 85, No. 1, 1989, pp. 440–449.

[12]Yamada, I., Yokata, A., Yamamoto, K., and Shimizu, S., "Acoustic Recognition of Aircraft Types in Flight," *Journal of the Acoustical Society of Japan,* Vol. 6, No. 3, 1985, pp. 203–213.

[13]Moukas, P., Simson, J., and Norton-Wayne, L., "Automatic Identification of Noise Pollution Sources," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-12, No. 5, 1982, pp. 622–634.

[14]Cabell, R. H., Fuller, C. R., and O'Brien, W., "The Identification of Helicopter Noise Using a Neural Network," AIAA Paper 90-3973, Oct. 1990.

[15]Klassen, G. S., "A General Purpose Reduction Intensive Feature Selector for Pattern Classification," Master's Thesis, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, June 1986.