R GE

Form Approved
OMB No. 0704-0188

1 AGENCY USE ONLY *(Leave blank)*	2 REPORT DATE	3 REPORT TYPE AND DATES COVERED
	June 1994	Professional Paper

4 TITLE AND SUBTITLE

TRANSIENT SONAR SIGNAL CLASSIFICATION USING HIDDEN MARKOV MODELS AND NEURAL NETS

5 FUNDING NUMBERS

PR: SUB8
WU: DN308105

6 AUTHOR(S)

G. C. Chen, A. Kundu, and C. E. Persons

7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152-5001

8 PERFORMING ORGANIZATION REPORT NUMBER

9 SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152-5001

10 SPONSORING/MONITORING AGENCY REPORT NUMBER

11 SUPPLEMENTARY NOTES

12a DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13 ABSTRACT *(Maximum 200 words)*

In ocean surveillance, a number of different types of transient signals are observed. These sonar signals are waveforms in one dimension (1-D). The hidden Markov model (HMM) is well suited to classification of 1-D signals such as speech. In HMM methodology, the signal is divided into a sequence of frames, and each frame is represented by a feature vector. This sequence of feature vectors is then modeled by one HMM. Thus, the HMM methodology is highly suitable for classifying the patterns that are made of concatenated sequences of micro patterns. The sonar transient signals often display an evolutionary pattern over the time scale. Following this intuition, the application of HMM's to sonar transient classification is proposed and discussed in this paper.

Published in *IEEE Journal of Oceanic Engineering*, vol. 19, no. 1, pp. 87-99, January 1994.

14 SUBJECT TERMS

acoustic surveillance
antisubmarine warfare (ASW)

15 NUMBER OF PAGES

16 PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT	18 SECURITY CLASSIFICATION OF THIS PAGE	19 SECURITY CLASSIFICATION OF ABSTRACT	20 LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

R GE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1994	Professional Paper

4. TITLE AND SUBTITLE

TRANSIENT SONAR SIGNAL CLASSIFICATION USING HIDDEN MARKOV MODELS AND NEURAL NETS

5. FUNDING NUMBERS

PR: SUB8
WU: DN308105

6. AUTHOR(S)

G. C. Chen, A. Kundu, and C. E. Persons

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152-5001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division
San Diego, CA 92152-5001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. ABSTRACT (Maximum 200 words)

In ocean surveillance, a number of different types of transient signals are observed. These sonar signals are waveforms in one dimension (1-D). The hidden Markov model (HMM) is well suited to classification of 1-D signals such as speech. In HMM methodology, the signal is divided into a sequence of frames, and each frame is represented by a feature vector. This sequence of feature vectors is then modeled by one HMM. Thus, the HMM methodology is highly suitable for classifying the patterns that are made of concatenated sequences of micro patterns. The sonar transient signals often display an evolutionary pattern over the time scale. Following this intuition, the application of HMM's to sonar transient classification is proposed and discussed in this paper.

Published in *IEEE Journal of Oceanic Engineering*, vol. 19, no. 1, pp. 87-99, January 1994.

14. SUBJECT TERMS

acoustic surveillance
antisubmarine warfare (ASW)

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

21a. NAME OF RESPONSIBLE INDIVIDUAL	21b TELEPHONE *(include Area Code)*	21c OFFICE SYMBOL
G. C. Chen	(619) 553-4924	Code 732

Transient Sonar Signal Classification Using Hidden Markov Models and Neural Nets

Amlan Kundu, *Member, IEEE*, George C. Chen, *Member, IEEE*, and Charles E. Persons

Abstract—**In ocean surveillance, a number of different types of transient signals are observed. These sonar signals are waveforms in one dimension (1-D). The hidden Markov model (HMM) is well suited to classification of 1-D signals such as speech [7], [8]. In HMM methodology, the signal is divided into a sequence of frames, and each frame is represented by a feature vector. This sequence of feature vectors is then modeled by one HMM. Thus, the HMM methodology is highly suitable for classifying the patterns that are made of concatenated sequences of micro patterns. The sonar transient signals often display an evolutionary pattern over the time scale. Following this intuition, the application of HMM's to sonar transient classification is proposed and discussed in this paper. Toward this goal, three different feature vectors based on an autoregressive (AR) model, Fourier power spectra, and wavelet transforms are considered in our work. In our implementation, one HMM is developed for each class of signals. During testing, the signal to be recognized is matched against all models. The best matched model identifies the signal class.**

The neural net (NN) classifier has been successfully used [2]–[4] for sonar transient classification. The same set of features as mentioned above is then used with a multilayer perceptron NN classifier. Some experimental results using "DARPA standard data set I" with HMM and MLP-NN classification schemes are presented. Finally, a combined NN/HMM classifier is proposed, and its performance is evaluated with respect to individual classifiers.

I. Introduction

THE classification of transient sonar signals has been widely studied [2]–[6]. The transient classification problem is deemed difficult for a number of reasons: 1) Short duration of the transients makes the classical frequency analysis difficult; 2) wide intraclass variations due to large variations in the structures and systems generating the transients; and 3) the effects of ambient ocean noise and the presence of biologics and merchant ships lead to poorly separated class boundaries. The most common type of classifier used for this task is the neural net [2]–[4] though other classifiers have been studied [2], [5], [6]. Fourier power spectral coefficients are widely used as feature vectors. Recently, the hidden Markov model has been studied for sonar signal classification [5], [6], [24]. In [5], [6], AR model parameters are used as feature vectors for the HMM classifier. It is relevant to note here that the HMM was originally introduced by the speech community [7], [8]. In speech, the linear predictive coefficients (LPC), i.e., AR coefficients, are successfully used as the feature vector. However, sonar signals have their own characteristics. It has been found that no single technique can adequately capture all feature information for all ocean acoustic transients of interest [2]–[6], [16]. So, it is expected that other features could lead to more interesting results. With this view in mind, we have experimented with the HMM classifier and three different feature vectors in this paper. The feature vector based on an AR model is a natural candidate. As the Fourier power spectrum is widely used by the NN community for their research, these features are also considered [4]. Finally, wavelet-transform-based features are considered. Interestingly, some features based on specific wavelet implementation have been used in [2], [3]. It is well known that sonar transients are nonstationary signals. The wavelet transform can properly represent such signals. In particular, Daubechies type wavelets are considered in our work. These wavelets are finite duration filters and quite easy to implement. Besides, these wavelets have not been tried in the context of transient sonar signal classification. It is our viewpoint that these three very different signal representations for feature extraction would reveal some of the latent characteristics of the signal for better classification.

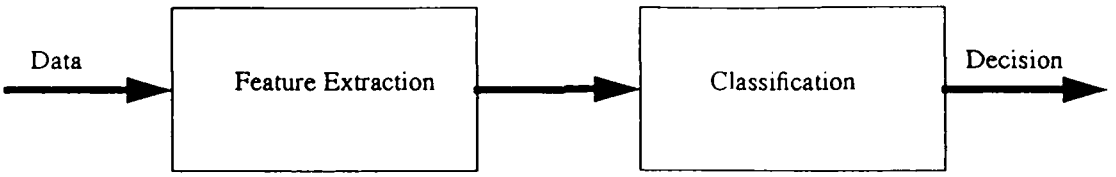
In speech, the spoken word manifests itself as a left-to-right concatenation of phonemes [7], [8], the fundamental speech unit. The states in HMM are identified with the phonemes. As a result, a left-to-right HMM topology is often preferred in the application of HMM to speech recognition. This argument, in our view, may not hold in all applications of HMM to sonar signal classification. We think of a particular sonar transient as a macro pattern that has evolved as a sequence of micro patterns. We identify the "states" with the "micro patterns." However, in the absence of any other *a priori* constraint, the macro pattern may be composed of any sequential combination of the basic micro patterns. In other words, a fully connected HMM topology, where the transition from any state to any other state is possible, could be more useful in such situations. For the dataset used in our experiment, the fully connected HMM topology performs consistently better than the left-to-right HMM topology. However, there are sonar signals where the utility of left-to-right HMM topology has been demonstrated [5].

Finally, we have studied the same set of features with a MLP–NN classifier with the express objective of finding out the complementary nature, if any, of these two classifiers—MLP–NN and HMM. So far, a comprehensive study involving NN and HMM and a number of feature sets has

Manuscript received May 1993. This work has been supported by NRaD's "wide area undersea surveillance" block program. The sponsor is T. Goldsberry, ONR 231.

The authors are with the Naval Command, Control, and Ocean Surveillance Center, RDT&E Division, Code 732, San Diego, CA 92152-5001.

IEEE Log Number 9215413.



- DARPA Data I
- Autoregressive Coefficients
- Spectral Coefficients
- Wavelet Coefficients
- Neural Network
- Hidden Markov Model
- Confusion Matrices

Fig. 1. Block diagram representation of classification scheme.

not been undertaken. In a recent article, Miller *et al.* [1] have pointed out the importance of exploring alternative technologies to NN in order to make comparative performance measurements and to obtain the best possible solutions to signal processing and classification problems. We show in the current paper that a combined classifier using HMM's and MLP–NN's is likely to outperform the individual classifiers. It is relevant to note here that the concept of a combined classifier for robust classification is well known in pattern recognition theory, and has already been tried with other classifiers for transient sonar signal classification [2]. Fig. 1 gives the block diagram of our scheme. Note that there is no preprocessing involved in our system. This is a deliberate decision. The preprocessing operations are often quite dependent on the given signal. Usually, these operations try to enhance or deemphasize certain aspects of the given signal for better classification. A potential drawback of such operations is that when the signal classes are changed, the old preprocessing schemes are often invalid. Thus, to design an automatic system for transient signal classification, we will not include any preprocessing operations. This design without preprocessing is expected to make our system suitable for a wide range of sonar transients.

The remaining sections of this paper are organized as follows: Section II describes the implementation of AR, FFT-based, and wavelet-based features. Section III presents the theory and implementation of HMM as applied to our classification problem. In this section, some discussions on the implementation of NN are also included. Section IV discusses some practical considerations in implementation. Section V gives the detailed experimental results using DARPA standard data set I. Section VI summarizes the conclusions.

II. Feature Representation

As described in Section I, we have three different feature representation schemes: one based on an autoregressive model, one based on Fourier power spectra, and the other based on the wavelet transform. In this section, these feature representation schemes are briefly discussed.

A. Autoregressive Model

In describing the AR model, we will use the notation $r(l)$ to denote the signal. The autoregressive model is a simple prediction of the current signal value by a linear combination of M previous signal values plus a constant term and an error term:

$$r(l) = \alpha + \sum_{j=1}^{M} \theta_j r(l-j) + \sqrt{\beta} w_l \qquad l = 1, \cdots, L \quad (2.1)$$

where:

$r(l)$: current signal value; $r(l-j)$: previous signal values; θ_j: autoregressive coefficients to be estimated; M: model order; α: constant to be estimated; $\sqrt{\beta}$: constant to be estimated; w_l: random number with zero mean and unit variance.

$\theta_1, \cdots, \theta_M$, α, β are the model parameters; β is the variance of prediction noise and reflects the accuracy of the prediction.

It is noted from (2.1) that to predict $r(1)$, we need M initial values of $r(l)$, i.e., $r(-M+1), r(-M+2), \cdots, r(0)$. It is easy to derive that:

$$\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_M \\ \alpha \end{bmatrix} = \begin{bmatrix} R_{11} & \cdots & R_{1M} & S_1 \\ \vdots & \ddots & \vdots & \vdots \\ R_{M1} & \cdots & R_{MM} & S_M \\ S_1 & \cdots & S_M & N \end{bmatrix}^{-1} \begin{bmatrix} R_{01} \\ \vdots \\ R_{0M} \\ S_0 \end{bmatrix} \quad (2.2)$$

where

$$R_{ij} = R_{ji} = \sum_{l=1}^{L} r(l-i) r(l-j), \qquad i, j = 1, \cdots, M \quad (2.3)$$

$$S_i = \sum_{l=1}^{L} r(l-i), \qquad i = 0, 1, \cdots, M \quad (2.4)$$

and

$$\beta = \frac{1}{L} \sum_{l=1}^{L} \left[r(l) - \left(\alpha + \sum_{j=1}^{M} \theta_j r(l-j) \right) \right]^2 \quad (2.5)$$

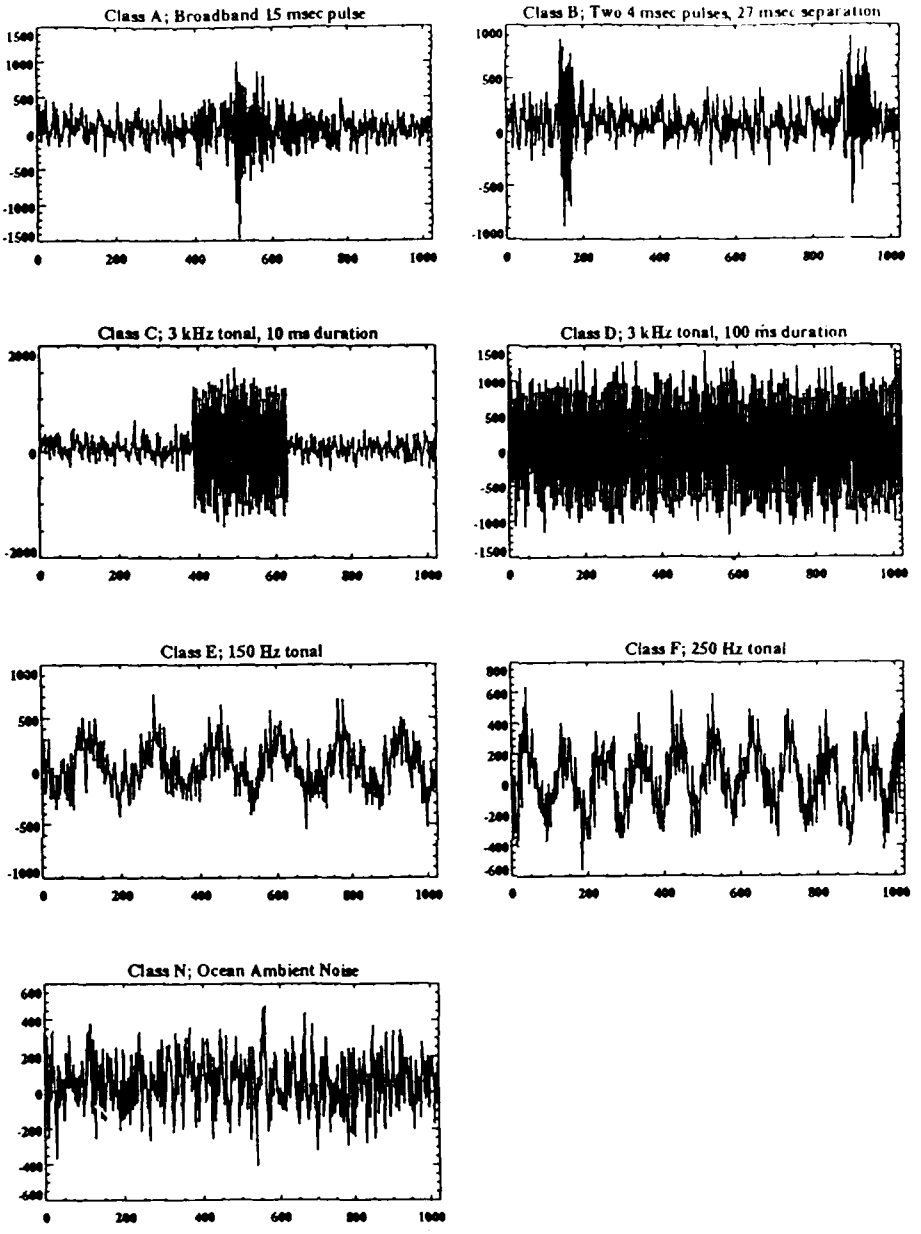


Fig. 2. An example of the different classes of signals used in our experiment.

If α is zero, (2.2) takes the form of Yule–Walker equations. The R matrix is then Hermitian and Toeplitz [19]. A straightforward approach is to replace the autocorrelation functions in R by sample autocorrelations. Often, a more efficient algorithm, known as Burg's algorithm, is used to compute θ_j. For details regarding Burg's algorithm, please see [19]. It should be noted that the choice of optimal model order, M, is application-dependent and is usually determined empirically.

B. Fourier Power Spectrum

From the given data segment, its FFT is computed. Before FFT computation, each data segment is windowed with a Kaiser–Bessel window function. The magnitude square of the FFT coefficients gives the Fourier power spectrum of the data.

C. Wavelet Transform

In the short-time Fourier transform, time and frequency resolutions are fixed. Because of Heisenberg's principle, the time and frequency resolution product cannot be better than a threshold ($1/4\pi$). In the wavelet representation, it is possible to achieve high time resolution at the cost of frequency resolution, and vice versa. This is easily demonstrated by making the frequency resolution proportional to frequency. In the wavelet transform, this compromise leads to very high time resolution for high-frequency signals, and high-frequency resolution for low-frequency signals. Since the sonar signals almost always display an evolving frequency profile with time, wavelet transform representation is philosophically very appealing.

In the wavelet transform, the transform space is defined by the basis functions, which are all derived from one basic

wavelet via scaling and translation; i.e., if $h(t)$ is the basic wavelet and $h_{a,\tau}(t)$ is a generic wavelet basis function, then [20], [21]:

$$h_{a,\tau}(t) = 1/\sqrt{(a)}h((t-\tau)/a).$$

When a and τ are continuous, there are infinite possibilities for a and τ. Consequently, the transformation of a signal $x(t)$ using these basis functions, and subsequent reconstruction of $x(t)$ from the transform, is a simpler task. The interesting task is to appropriately discretize the time-scale parameter a and τ such that a true orthonormal basis function is obtained. The solution depends on the choice of wavelet $h(t)$. So, our problem is:

$$\text{discretize } h_{j,k}(t) = a_0^{-j/2}h(a_0^{-j}t - kT)$$

where T is the sampling period of the discrete signal. Of course, if we choose $a_0 \approx 1$ and T small, we are close to the continuous case. For implementation advantage, our interest is in the dyadic wavelet that has $a_0 = 2$, i.e.,

$$h_{j,k}(t) = 2^{-j/2}h(2^{-j}t - kT)$$

where j and k belong to the set of natural numbers. The Daubechies wavelets are a class of discrete orthonormal dyadic wavelets. An M-order Daubechies wavelet [20] is given by M coefficients denoted by C_j, $j = 0, \cdots, M-1$. Then, the convolution of the signal with a FIR filter of length M (C_j, $j = 0, \cdots M-1$) gives the smooth component. On the other hand, the convolution of the signal with a FIR filter of length M and coefficients $(-1)^{-j}C_{M-1-j}$, $j = 0, \cdots M-1$ gives the detail component. After one pass of this algorithm, the smooth and detail components are decimated by 2. The smooth components are then transformed again, and the procedure continues until we have only two smooth components left. The output, at this stage, is the wavelet transform of the original signal. The coefficients in Daubechies wavelets are obtained from orthonormality conditions and "smoothness constraints." For an M-order wavelet, these conditions and constraints lead to exactly M linear equations. Thus, M coefficients are uniquely determined. For more discussions and details about the coefficients, see [20], [21]. For an excellent exposition related to theory and applications of wavelet transforms, please see [23].

D. Feature Selection

The feature representation schemes described so far transform the original signal into feature space. Since some features may be more useful than others, only the important features should be selected for a compact representation of the signal for classification purposes. This is a necessary data reduction stage. The idea behind this stage is that only a few features can discriminate one class from the others.

In our scheme, the signal is divided into a number of overlapping frames. For the AR-model-based feature representation, the AR coefficients are taken as the feature vector. Since relatively few AR coefficients are needed to represent a frame, AR feature representation is already in very compact form. For FFT power spectrum and the wavelet transform, the spectral and transform coefficients with relatively higher magnitude are selected as features. For instance, 256-point real data will give 128 distinct FFT power spectral coefficients. For a particular signal class, all such FFT power spectral coefficients are analyzed, and the top few, say L of them, in terms of magnitude, are selected as features for that signal class. The union of all individual feature sets, each one belonging to one signal class, gives the global feature vector set for all signals. A similar procedure is used to select the feature vector from wavelet transform coefficients. The details regarding the number of frames in a signal template, the percentage of overlap among successive frames, and the number of features in each feature vector are described in Section V.

III. Classifier Design

In our work, we have used two classifiers: HMM and MLP-NN. In this section, these two classifiers are described.

A. Continuous Density HMM

A first order N-state Markov chain is defined by an $N \times N$ state transition probability matrix A and an $N \times 1$ initial probability vector Π, where:

$$A = \{a_{ij}\}, \qquad a_{ij} = \Pr(q_{t+1} = j | q_t = i); \qquad i, j = 1, 2, \cdots, N$$

$$\Pi = \{\pi_i\}; \qquad \pi_i = \Pr(q_1 = i), \qquad i = 1, 2, \cdots, N$$

$$Q = \{q_t\} - \text{state sequence.} \qquad q_t \in \{1, 2, \cdots, N\}, \qquad t = 1, 2, \cdots, T$$

N—number of states

T—length of state sequence.

By definition, $\sum_{j=1}^{N} a_{ij} = 1$ for $i = 1, 2, \cdots, N$ and $\sum_{i=1}^{N} \pi_i = 1$. A state sequence Q is a realization of the Markov chain with probability:

$$\Pr(Q|A, \Pi) = \pi_{q_1} \prod_{t=2}^{T} a_{q_{t-1}q_t}. \tag{3.1}$$

A hidden Markov model is a Markov chain whose states cannot be observed directly, but can be observed through a sequence of observation vectors [7]. Each observation vector, also called a symbol, manifests itself as states through certain probability distributions. In other words, each observed vector is generated by an underlying state with an associated probability distribution. For solving our problem, we will consider only the observations with continuous probability density. A continuous density HMM is characterized by the state transition probability A, the initial state probability Π, and an $N \times 1$ observation density or symbol probability density vector B, where:

$$B = \{b_j(o_t)\},$$

$$b_j(o_t) = a \textit{ posteriori} \text{ density of observation } o_t \text{ given } q_t = j$$

$$O = \{o_t\} - \text{observation sequence.} \qquad t = 1, 2, \cdots, T$$

In many practical problems, it is reasonable to assume that the observation density is Gaussian. In this case, the density is completely specified by the mean and covariance of o_t, i.e., $b_j(o_t) = N(\mu_j, V_j)$, where μ_j and V_j are the conditional mean vector and the conditional covariance matrix, respectively, of o_t given state $q_t = j$. It should be noted that, in our application, the states of an HMM may not have specific physical meaning. They may just reflect some clustering properties of the observation vectors in the feature space.

We can more compactly denote the parameter set by $\lambda = (A, \Pi, B)$. Then, an HMM is completely specified by λ. Three problems associated with HMM are of our concern:

1) Based on what optimization criterion should our model be built?

2) Given the model and an observation sequence $O = \{o_1, o_2, \cdots, o_T\}$, how can we classify the observation efficiently? This is the classification problem.

3) Given a number of observation sequences of a known class, how can we obtain the optimal model estimate $\hat{\lambda}$? This is the training problem.

Problem 1—Optimization Criterion: Suppose we are given a model λ and an observation sequence $O = \{o_1, o_2, \cdots, o_T\}$. Then, the density function of O is given by:

$$p(O|\lambda) = \sum_{Q} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t). \tag{3.2}$$

A direct choice of optimization criterion is the maximum likelihood criterion that maximizes $P(O|\lambda)$. The estimation of the parameters by this criterion can be solved using the Baum–Welch reestimation algorithm [7]. The algorithm is an iterative procedure that guarantees a monotonic increase of the likelihood function for a given set of training samples.

Another optimization criterion is to maximize the state-optimized likelihood function defined by:

$$p(O, Q^*|\lambda) = \max_{Q} p(O, Q|\lambda)$$

$$= \max_{Q} \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t) \tag{3.3}$$

where $Q^* = \{q_1^*, q_2^*, \cdots, q_T^*\}$ is the optimal state sequence associated with the state-optimized likelihood function, and q_i^* is the ith state in this optimal state sequence. Equation (3.3) is the density of the optimal or most likely state sequence path among all possible paths. The estimation of the parameters using this criterion is given by the segmental K-means algorithm [9], [10]. This algorithm is an iterative procedure that guarantees, under some conditions described later, the monotonic increase of the state-optimized likelihood function for a given set of training samples.

Comparing (3.2) and (3.3), we find out that (3.2) involves computation along all possible state paths, while (3.3) tracks only the most likely path. Therefore, the computation required by (3.3) is much less than that of (3.2). Also, since $b_{q_t}(o_t)$ often has a large dynamic range, overflow or underflow is more likely to happen in evaluation of (3.2) than in evaluation of (3.3). Furthermore, in a particular application, if the data fit the model very well, all the observation samples of one class are likely to have few dominant state sequences. It means that the optimal state sequence in (3.3) carries a lot of information that may discriminate one class from another. For these reasons, we have chosen maximization of (3.3) as our criterion.

Problem 2—Classification: To solve our signal classification problem, we create one HMM for each class. For a classifier of P classes, we denote the P models by λ_p, $p = 1, 2, \cdots, P$. When a signal O of unknown class is given, we calculate:

$$p^* = \arg\max_{p} p(O, Q^*|\lambda_p) \tag{3.4}$$

and classify the signal as belonging to class p^*.

Now we can immediately see one of the advantages of HMM. The model for one class is independent of the model for any other class, i.e., the training for one class is not related to the training for any other class. It follows that when a new class is added to the classifier, we need only to train for this new class, but do not have to retrain for any other class. In general, this advantage is not associated with a neural net classifier.

For a given λ, an efficient method to find $p(O, Q^*|\lambda)$ is the well-known Viterbi algorithm [11], [12] as described below.

Viterbi Algorithm

Step 1. Initialization

For $1 \le i \le N$,

$$\delta_1(i) = \pi_i b_i(o_1) \tag{3.5}$$

$$\psi_1(i) = 0. \tag{3.6}$$

Step 2. Recursive computation

For $2 \le t \le T$, for $1 \le j \le N$,

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(o_t) \tag{3.7}$$

$$\psi_t(j) = \arg\max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}]. \tag{3.8}$$

Step 3. Termination

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{3.9}$$

$$q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)]. \tag{3.10}$$

Step 4. Tracing back the optimal state sequence

For $t = T - 1, T - 2, \cdots, 1$,

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{3.11}$$

P^* is the state-optimized likelihood function, and $Q^* = \{q_1^*, q_2^*, \cdots, q_T^*\}$ is the optimal state sequence.

In practice, as t increases, the value of $\psi_t(j)$ could be very large or very small so that an overflow or underflow may occur during computation on a computer. To avoid this problem, we take the logarithm of all probabilities and densities, and replace all multiplications by additions. Obviously, the result of tracting the optimal state sequence is not affected by this modification. If any particular value is zero, we set it to a very small number such that it does not affect the result.

Problem 3—Training: In creating the model for each class, we should guarantee that the parameters we obtain are the optimum for a given set of training samples. Since our decision rule is the state-optimized likelihood function, it requires that the estimated parameter $\hat{\lambda}$ be such that $p(O, Q^*|\hat{\lambda})$ is maximized for the training set. It is shown in [9] that the segmental K-means algorithm [10] converges to the state-optimized likelihood function for a wide range of observation density functions, including the Gaussian density we have assumed. The algorithm is described below.

1) Cluster all training vectors into N clusters using the minimum distance rule with random initial clustering centroids. Each cluster is chosen as a state and numbered from 1 to N. The tth vector, o_t, of a training sequence O is assigned to state i, denoted as $o_t \in i$, if its distance to state i is smaller than its distance to any other state j, $j \neq i$. The distance measure we have used is the unweighted Euclidean distance. This step is to get a good initialization for the complete training procedure.

2) Calculate the mean vector and covariance matrix for each state. For $1 \leq i \leq N$,

$$\hat{\mu}_i = \frac{1}{N_i} \sum_{o_t \in i} o_t \tag{3.13}$$

$$\hat{V}_i = \frac{1}{N_i} \sum_{o_t \in i} (o_t - \hat{\mu}_i)^t (o_t - \hat{\mu}_i) \tag{3.14}$$

where N_i is the number of vectors assigned to state i.

3) Calculate the transition and initial probabilities. For $1 \leq i \leq N$,

$$\hat{\pi}_i = \frac{\text{Number of occurrences of } \{o_1 \in i\}}{\text{Number of training sequences}}. \tag{3.15}$$

For $1 \leq i \leq N$ and $1 \leq j \leq N$,

$$\hat{a}_{ij} = \frac{\text{Number of occurrences of } \{o_t \in i \text{ and } o_{t+1} \in j\} \text{ for all } t}{\text{Number of occurrences of } \{o_t \in i\} \text{ for all } t}. \tag{3.16}$$

4) Calculate density functions of each training vector for each state. For $1 \leq j \leq N$,

$$\hat{b}_j(o_t) = \frac{1}{(2\pi)^{M_2/2} |\hat{V}_j|^{1/2}} \cdot \exp\left[-\frac{1}{2}(o_t - \hat{\mu}_j)\hat{V}_j^{-1}(o_t - \hat{\mu}_j)^t\right]. \tag{3.17}$$

Here M_2 is the dimension of the feature vector.

5) Use the Viterbi algorithm and the new probabilities to trace the optimal state sequence Q^* for each training sequence. A vector is reassigned a state if its original state assignment is different from the tracing result, i.e., assign $o_t \in i$ if $q_t^* = i$.

6) If any vector is reassigned a new state in Step 5, use the new state assignment and repeat Step 2 through Step 5; otherwise stop.

B. Multilayer Perceptrons

Multilayer perceptrons (MLP) are feedforward nets with one or more layers of nodes between the input and output layers. The lowest layer is the input layer, which does not have any processing capability. The highest layer is the output layer, and any layer between the input and output layers is called the hidden layer. All the nodes in a layer are connected to the nodes in the layer above it, and there is no connection within a layer or from the higher layer. For example, a three-layer perceptron is shown in Fig. 4. The perceptron processing unit performs a weighted sum of its input values a_i

$$x_j = f\left(\sum_{i=0}^{I} a_i w_{ij}\right) \qquad y_k = f\left(\sum_{j=0}^{K} x_j v_{jk}\right)$$

where $\{w_{ij}\}$, $\{v_{jk}\}$ are the weight matrices and $f(\cdot)$ is usually a nonlinear function such as the sigmoid function

$$f(x) = 1/(1 + e^{-x}).$$

Generally, the multilayer perceptrons are trained with the error backpropagation (EBP) algorithm [15] which is an iterative gradient algorithm designed to minimize the mean square error (MSE) between the desired output y_k^* and actual output y_k. Sometimes, a momentum term is also included in the training procedure. The details of this algorithm can be found in [14]. In addition to MLP's, other neural networks such as Kohonen feature maps have also been used in pattern recognition. A good introduction to the neural nets is given by Lippmann [13], and an excellent exposition of the NN is given by Hecht–Nielsen [17]. For a useful survey on NN's and their foundations, paradigms, applications, and implementations, see [18], [22].

IV. Implementation Considerations

A. Training of Classifiers

Each signal template, i.e., exemplar, is divided into a sequence of partially overlapping segments. Each segment is then represented by one feature vector. The sequence of feature vectors is used as one training/testing observation sequence for the HMM. For the MLP–NN, the whole sequence of feature vectors is used as the training vector. For example, if there are 20 four-dimensional vectors in the sequence, these 80 features are used as training feaures for the MLP–NN, and the MLP–NN is designed with 80 input nodes. The MLP–NN has one hidden layer, and it is trained using the backpropagation algorithm and sigmoidal nonlinearity. The HMM's are trained using segmental k-means algorithm [11] as described in the previous section. For each signal class, one HMM is designed. During recognition, the test signal is matched against all models to find the best match. The matching is done by the Viterbi algorithm [1], [12]. Fig. 3 depicts the implementation of HMM. For more details about the number of points in each signal template, the number of frames in a signal template, the percentage of overlap among successive frames, and the number of signal classes, refer to Section V.

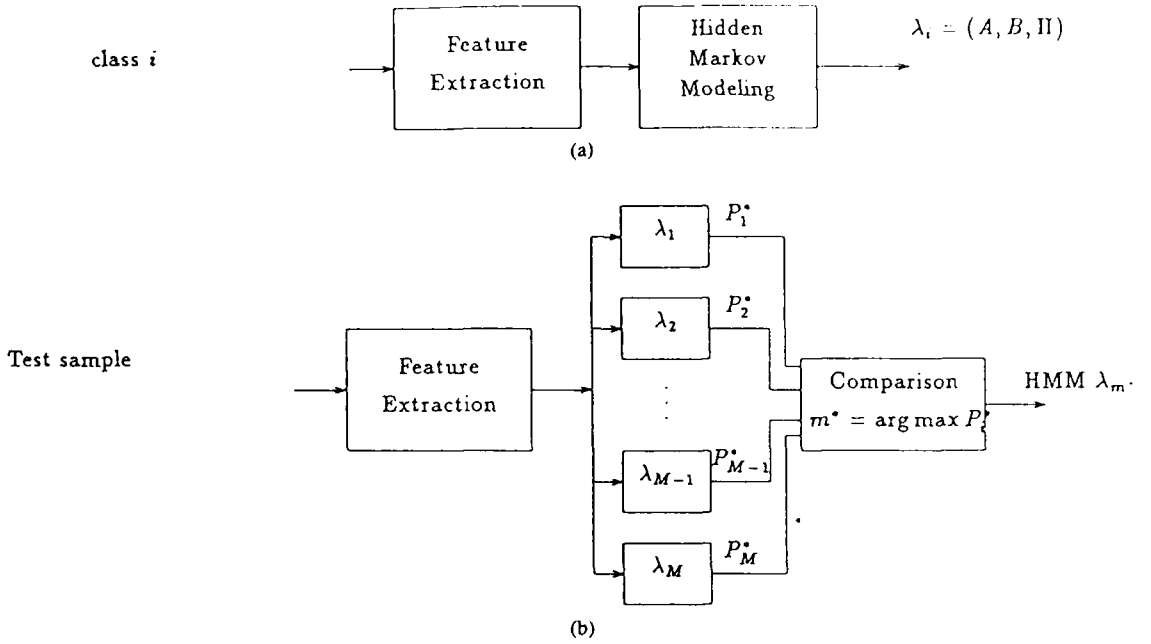


Fig. 3. Implementation of the HMM classifier. (a) Training phase. (b) Testing.

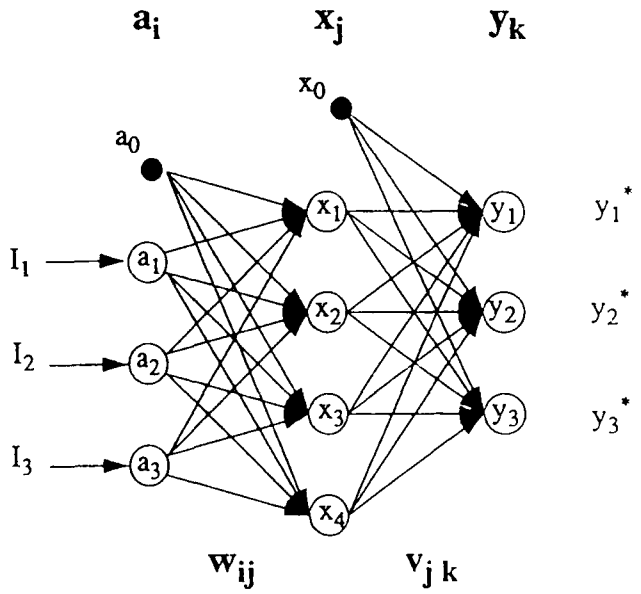


Fig. 4. A three-layer perceptron.

The AR coefficients are computed using Burg's algorithm. The gain coefficient is not used due to a scaling problem. The range of the "gain" coefficient is much much larger than the AR coefficients. The gain is given by $\sqrt{\beta}$ [(2.1)]. There are sophisticated techniques to overcome this problem and get better results. For instance, in [5] a product-code HMM is used that can take the gain coefficient into account. Daubechies-4 and Daubechies-20 wavelet coefficients are used to compute the wavelet transform. As explained in Section II, for each signal, only a few wavelet coefficients with high magnitude are used as features. Similarly, for each signal, only a few FFT power spectral coefficients with high magnitude are used as features. The union of the coefficients for all different signal types constitutes the feature vector.

B. Initial Clustering Center and Local Maxima for HMM

In Section III, we have assumed that the feature vectors have a normal distribution within each state. The global convergence property of the segmental K-means algorithm is based on this assumption. Although this is a practical and reasonable assumption, when the number of training samples is not sufficiently large, the data may not conform to this assumption very well. A better solution to this problem is replacing the Gaussian density by a mixture of Gaussian densities, but this will greatly increase the complexity of the model and therefore will be computationally very costly. If we do not change our model but carefully choose the initial cluster centers in Step 1 of the training algorithm (Section III) [11], we may still reach the global maximum. Thus, in the training procedures, we will try different initial cluster centers and select the set of parameters that results in the largest average P^* over all training samples of that class.

V. Experimental Results

A. Signal Description

We have used DARPA standard data set I for our experiments. This data set provides seven classes of signals to test our algorithm. A typical example, one from each class, is shown in Fig. 2. We denote these signal classes as:

Class A: Broadband 15-ms pulse
Class B: Two 4 ms pulses, 27 ms separation
Class C: 3 kHz tonal, 10 ms duration

TABLE I
CONFUSION MATRIX, HMM CLASSIFIER, AND AR COEFFICIENT FEATURES (SIXTH-ORDER)

True Class \ Chosen Class	A	B	C	D	E	F	N
A	17	1	0	0	1	0	4
B	0	15	0	0	1	0	0
C	0	0	22	0	0	0	0
D	0	0	2	20	0	0	0
E	1	0	0	0	15	5	1
F	0	0	0	0	9	13	0
N	0	3	0	0	2	6	11

- 6th-order AR model
- 6-state HMM; 8-state is worse
- Recognition accuracy = 73.3 %
- AR model may not be a good fit for this data
- AR model has problem modeling a pure sinusoid

TABLE II
CONFUSION MATRIX, HMM CLASSIFIER, AND AR COEFFICIENT FEATURES (TENTH-ORDER)

True Class \ Chosen Class	A	B	C	D	E	F	N
A	17	1	0	0	1	0	3
B	0	17	0	0	1	0	4
C	2	0	18	2	0	0	0
D	0	0	2	20	0	0	0
E	0	1	0	0	14	7	0
F	0	1	3	0	5	11	2
N	0	2	0	0	6	7	7

- 10th-order AR model
- 6-state HMM
- Recognition accuracy = 67.5%

Class D: 3 kHz tonal, 100 ms duration
Class E: 150 Hz tonal, 1 s duration
Class F: 250 Hz tonal, 8 s duration
Class N: Ocean ambient noise.

We have created 45 templates, i.e., exemplars, for each class, of which 23 are used as training templates and 22 as test templates. Each signal template contains 1024 data points. The sampling rate for the signal is 24.576 kHz. For this sampling rate, 1024 data points are enough to capture the essential characteristics of all transient types including the Class B type signal, which has the most time spread. This 1024-point signal template is divided into 21 frames of 256 data points with an overlap of 218 points (approximately 85%) between two successive frames. Once the feature vectors are computed from each frame, the signal template is represented by a feature vector sequence. The training/testing sets include exemplars from five different SNR groups. The lowest SNR is 24 dB down with respect to the highest SNR. The first group is the reference, i.e., 0 dB, group. The other groups are created adding background noise to this reference group, and the SNR values for these groups are -6, -12, -18, and -24 dB, respectively, with respect to the reference 0 dB group. The SNR is computed as the ratio of the peak signal power to background noise power expressed in dB. As a result, some very noisy exemplars are included in our experiments. Most classifiers can handle signals with relatively high SNR quite well, but fail with low SNR signal. A meaningful evaluation of a classifier is possible only when the classifier can classify low SNR exemplars with high accuracy. Another important distinction in our experiment is that we have included ocean noise as a separate class. In DARPA data set I, ambient noise has a frequency spectrum that substantially overlaps with that of types A, B, E, and D signals. Thus, the inclusion of ambient noise as a separate class makes our classification problem much harder.

We have tried a different number of states for HMM, from $N = 2$ to $N = 12$, and a different number of nodes, from 10 to 30, in the hidden layer of the MLP–NN. We have also compared the results of AR models of different orders, from $M = 2$ to $M = 10$. Only the best results are reported in the paper and the accompanying tables. Table I shows the number of errors in classifying the total 154 test exemplars using AR features and HMM's. The best result is obtained with the six-state HMM, and sixth-order AR model. As stated before, the gain coefficient is not used mainly because of the scale problem. Table II gives the result of the same experiment with ten-order AR model. The recognition accuracy, i.e., percentage of correctly classified test exemplars, in both these experiments is rather poor. This poor showing of AR-model-based features could be attributed to two possible explanations: 1) the AR model may not be a good fit for DARPA data set I; 2) the AR model implemented with Burg's algorithm has a problem modeling a pure sinusoid [19]. It can be clearly seen that the AR model has great difficulty in discriminating type E and type F signals—two single-frequency tonals with close frequencies. We have also found that the AR order beyond 6 is not helpful as the extra poles try to match the spurious peaks due to ocean

TABLE III
CONFUSION MATRIX, HMM CLASSIFIER, AND FFT FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	21	0	0	0	1	0	0
B	0	19	2	0	1	0	0
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	0	0	0	21	0	1
F	0	0	0	0	0	21	1
N	0	7	0	0	3	0	12

- 30 FFT features
- 8-state HMM
- Recognition accuracy = 89.6 %
- Recognition accuracy = 95.5 % when class for ocean noise is excluded.

TABLE IV
CONFUSION MATRIX, HMM CLASSIFIER, AND DAUBECHIES 4 FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	20	1	0	0	0	0	1
B	4	17	0	0	0	0	1
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	0	0	0	21	0	1
F	0	0	0	0	0	21	1
N	0	4	0	0	0	0	18

- 30 features
- 8-state HMM
- Recognition accuracy = 91.5 %

TABLE V
CONFUSION MATRIX, HMM CLASSIFIER, AND DAUBECHIES-20 FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	20	1	0	0	0	0	1
B	3	17	0	0	0	0	2
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	0	0	0	21	0	1
F	0	1	0	0	0	21	0
N	0	5	0	0	0	0	17

- 30 features
- 8-state HMM
- Recognition accuracy = 90.9 %

noise. It is conceivable that some performance improvement is still possible with AR-model-based features [5]; however, as we will show, the FFT power spectral features and Daubechies wavelet based features hold more promise for our classification task. Please note that the results reported in [2] and [6] are also not favorable for AR-model-based features.

Table III shows the number of errors in classifying the total 154 test exemplars using FFT power spectral features and HMM's. The best result is obtained with the eight-state HMM's. The recognition accuracy is now close to 90%. When the ocean ambient noise is excluded, the recognition accuracy is over 95.5%. Tables IV and V show the number of errors in classifying the total 154 test exemplars using wavelet-transform-based features and HMM's. The best result is obtained with an eight-state HMM's. The recognition accuracy is now above 90%.

Table VI shows the number of errors in classifying the total 154 test exemplars using FFT power spectral features and MLP–NN. The best result is obtained with 20 nodes at the hidden layer. The recognition accuracy is now above 90%. Tables VII and VIII show the number of errors in classifying the total 154 test exemplars using wavelet-transform-based features and MLP–NN. Once again, the recognition accuracy is above 90%. In particular, the MLP–NN classifier and Daubechies-20 transform feature combination has achieved the best individual performance—93.5% classification accuracy.

We have also experimented with left-to-right HMM's. Table IX shows the number of errors in classifying the total 154 test exemplars using wavelet-transform-based features. The best result is achieved with Daubechies-4 transform features, and is reported in Table IX. This best result—88.9% classification accuracy—is somewhat inferior to the results achieved by

TABLE VI
CONFUSION MATRIX, NN CLASSIFIER, AND FFT FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	18	2	0	0	0	0	2
B	1	19	0	0	0	0	2
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	0	0	0	20	0	2
F	0	1	0	0	0	21	0
N	0	3	0	0	1	0	18

- 30 features
- 20 nodes for the hidden layer
- Recognition accuracy = 90.9 %

TABLE VII
CONFUSION MATRIX, NN CLASSIFIER, AND DAUBECHIES-4 FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	20	1	0	0	0	0	1
B	2	18	0	0	0	0	2
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	0	0	0	21	0	1
F	0	0	0	0	0	22	0
N	1	6	0	0	0	0	15

- 30 features
- 20 nodes for the hidden layer
- Recognition accuracy = 90.9 %

fully connected HMM's. Another important point is that the initialization process as described in Section III is only good for fully connected HMM's. For left-to-right HMM's, the initialization process needs to be defined in terms of an initial guess of A and B probability parameters. This latter initialization process is considerably more difficult, and needs more intimate knowledge of the data.

TABLE VIII
CONFUSION MATRIX, NN CLASSIFIER, AND DAUBECHIES-20 FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	20	0	0	0	0	0	2
B	0	20	0	0	2	0	0
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	0	0	0	20	0	2
F	0	0	0	0	0	22	0
N	1	0	0	0	2	1	18

- 30 features
- 20 nodes for the hidden layer
- Recognition accuracy = 93.5 %

B. Combined Classifier

From the confusion matrices given by Tables III–VIII, it is clear that every feature/classifier combination has a somewhat different performance. A pertinent question is—can we combine the evidence of all the feature/classifier combinations to yield results that would be superior to any specific feature/classifier combination? Such a combined classifier would also be more robust. One simple way to combine the feature vectors is to extract AR, FFT-based, and wavelet coefficients from each frame, and then form a large vector which would be the input to either a HMM or MLP–NN based classifier. In our case, this solution would mean a 66-dimensional floating point feature vector for each frame. This tremendous increase in computational complexity can be avoided by intelligent use of the classifier. A product-code HMM as described in [5] can incorporate all three different feature vectors in one classifier without substantial increase in the complexity. Unfortunately, these three feature sets are not independent of each other as required by the theory of product-code HMM.

We have devised a simple classifier, henceforth called the majority classifier, that would take the output of each specific feature/classifier combination and assign the test exemplar the class with the majority votes only when the vote exceeds a threshold. Since we have six votes per test exemplar, we choose a threshold of 3 and 4. If the majority vote is below this threshold, that test exemplar is not classified. The detailed

TABLE IX
CONFUSION MATRIX, L–R HMM CLASSIFIER, AND DAUBECHIES-4 FEATURES

True Class \ Chosen Class	A	B	C	D	E	F	N
A	20	2	0	0	0	0	0
B	2	18	0	0	0	0	2
C	0	0	22	0	0	0	0
D	0	0	0	22	0	0	0
E	0	1	0	0	20	0	1
F	0	1	0	0	0	21	1
N	1	6	0	0	0	0	15

- 30 features
- 8-state HMM
- Recognition accuracy = 88.9 %

TABLE X
FUSED CONFUSION MATRIX, MAJORITY VOTE NEEDED=4

True Class \ Chosen Class	A	B	C	D	E	F	N	X
A	20	0	0	0	0	0	0	2
B	0	17	0	0	0	0	0	5
C	0	0	22	0	0	0	0	0
D	0	0	0	22	0	0	0	0
E	0	0	0	0	21	0	1	0
F	0	0	0	0	0	21	0	1
N	0	1	0	0	0	0	17	4

- Misclassified = 2
- No decision = 12
- X means no decision

TABLE XI
FUSED CONFUSION MATRIX, MAJORITY VOTE NEEDED=3

True Class \ Chosen Class	A	B	C	D	E	F	N	X
A	20	0	0	0	0	0	1	1
B	2	19	0	0	0	0	0	1
C	0	0	22	0	0	0	0	0
D	0	0	0	22	0	0	0	0
E	0	0	0	0	21	0	1	0
F	0	0	0	0	0	22	0	0
N	0	1	0	0	0	0	20	1

- Misclassified = 5
- No decision = 3
- X means no decision

experimental results are given in Tables X and XI. When the threshold is 4, only two test exemplars are misclassified, but 12 are not classified. When the threshold is 3, five test exemplars are misclassified, but only three are not classified. It is very clear that the exemplars that would otherwise be classified erroneously are now classified as "nonclassified" by the combined classifier. Also, very few test exemplars are misclassified by the combined classifier. Thus, in the case of a definitive decision, the combined classifier has close to 100% recognition accuracy. For test signals with low SNR, the combined classifier would not make a wrong decision. For the lack of consistent evidence, it would term the signal as "nonclassified."

It is possible to implement the classified-versus-nonclassified decision for the individual classifier by finding an optimum threshold. For a HMM classifier, the computation of this optimal threshold requires some knowledge about the distribution of the observation sequence corresponding to optimal Viterbi state sequence given the HMM parameters. For an MLP–NN, the distribution at the output node is needed. While these distributions are very difficult to find, the majority classifier simplifies the threshold computation problem enormously especially when the output of each classifier/feature combination is given equal weight. Also, the best performance for a classifier/feature combination is given by the NN/Daubechies-20 combination. In this case, 10 out of 154 exemplars are wrongly classified. In the case of a majority classifier with threshold 3, five exemplars are wrongly classified and three exemplars are not classified. Even if we consider the "nonclassified" exemplars as wrongly classified, the performance of the majority classifier is still superior to any individual classifier.

It is interesting to compare our data and results to those of Ghosh *et al.* [2] and Desai *et al.* [3]. Although, the same DARPA data set is used in [2], [3], only four signal types are considered in [3]. In [2], noise as a separate class is

not considered. As we have mentioned before, the addition of ocean noise as a separate class makes the classification of DARPA data set I very difficult. Another important point is that, in our technique, no preprocessing is used; but such processings are used in [2], [3]. Also, the experiments in [2] use signals from DARPA data set I test data set. Since we have no access to the "truthing" of these data, we have focused our experiment on DARPA data set I training data set. The "truthing" of all the signals in this set is known. Thus, a proper comparison is not possible at this point though our technique has performed quite well, which validates our approach. Also, a number of conclusions from our work confirm two main observations made in [2]. These are: 1) both FFT- and wavelet-based features are promising; and 2) a combined classifier is likely to yield better results.

VI. Concluding Remarks and Future Research

Based on the experimental results, the following conclusions are in order:

1) Both FFT- and wavelet-based features are promising.

2) To a certain extent, the wavelet-based features complement the FFT-based features.

3) To a certain extent, the HMM classifier complements the NN classifier.

4) The combined classifier has the best result. Also, the combination is very robust. Only a simple combination is described in the paper. Other possible combinations of HMM/NN classifier should be explored.

5) For longer signals, HMM-based classification could prove more effective. The longer signals are likely to have more sequence information. The exploitation of this sequence information is the rationale for using HMM.

6) Features based on other wavelets, such as Gabor wavelets, may prove more effective. This is another interesting future research topic.

Acknowledgment

The authors would like to thank Dr. B. Yoon of DARPA, Dr. T. McKenna of ONR and Code 535 at NRaD, San Diego for the DARPA dataset I. The authors would also like to thank the reviewers for many thoughtful comments.

References

[1] W. Miller, T. McKenna, and C. Lau, "Office of naval research contributions to neural networks and signal processing in oceanic engineering," *IEEE J. Ocean. Eng.*, vol. 17, no. 4, pp. 299–307, 1992.

[2] J. Ghosh, L. M. Deuser, and S. D. Beck, "A neural network based hybrid system for detection, characterization and classification of short-duration oceanic signals," *IEEE J. Ocean. Eng.*, vol. 17, no. 4, pp. 351–363, 1992.

[3] M. Desai and D. J. Shazeer, "Acoustic transient analysis using wavelet decomposition," in *Proc. IEEE Neural Netw. Conf. Ocean. Eng.*, 1991, pp. 29–40.

[4] Y. H. Pao and T. H. Hemminger, "An episodal neural-net computing approach to the detection and interpretation of underwater acoustic transients," in *Proc. IEEE Neural Netw. Conf. Ocean. Eng.*, 1991, pp 21–28.

[5] J. P. Woodard, "Modeling and classification of natural sounds by product code hidden Markov models," *IEEE Trans. Acoust., Speech, and Signal Proces.*, vol. 40, no. 7, pp. 1833–1836, 1992.

[6] M. K. Shields and C. W. Therrien, "A hidden Markov model approach to the classification of acoustic transients," in *Proc. ICASSP*, San Francisco, CA, 1992, pp. 2731–2734.

[7] L. R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, pp. 4–16, June 1986.

[8] L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Tech. J.*, vol. 64, no. 6, pp. 1211–1234, July/Aug. 1985.

[9] B. H. Juang and L. R. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoust., Speech, and Signal Proces.*, vol. ASSP-38, no. 9, pp. 1639–1641, Sept. 1990.

[10] L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A segmental k-means training procedure for connected word recognition," *AT&T Tech. J.*, vol. 65, no. 3, pp. 21–31, May/June 1986.

[11] Y. He and A. Kundu, "2-D shape classification using HMM," *IEEE Trans. Patt. Analysis and Mach. Intell.*, vol. 13, no. 11, pp. 1172–1184, 1991.

[12] G. D. Forney, Jr., "The Viterbi algorithm," *IEEE Proc.*, vol. 61, no. 3, pp. 263–278, Mar. 1973.

[13] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, pp. 4–21, Apr. 1987.

[14] J. E. Dayhoff, *Neural Network Architecture: An Introduction.* New York: Van Nostrand Reinhold, 1990, ch. 4, p. 58.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: M.I.T. Press, 1986.

[16] T. Brotherton, T. Pollard, R. Barton, A. Krieger, and L. Marple, "Application of time-frequency and time-scale analysis to underwater acoustic transients," in *Proc. IEEE Int. Symp. Time-freq. and Time-scale Analysis*, Victoria, B.C., Canada, Oct. 1992.

[17] R. Hecht-Nielsen, *Neurocomputing.* Reading, MA: Addison-Wesley, 1990.

[18] P. K. Simpson, *Artificial Neural Systems: Foundations, Paradigms, Applications and Implementations.* Elmsford, NY: Pergamon, 1990.

[19] S. Kay, *Modern Spectral Estimation.* Englewood Cliffs, NJ: Prenti ce Hall, 1988.

[20] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure and Applied Math*, vol. 41, no. 7, pp. 909–996, 1988.

[21] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE ASSP Mag.*, vol. 8, no. 4, pp. 14–38, 1991.

[22] C. Lau, *Neural Networks: Theoretical Foundations and Analysis.* New York: IEEE Press, 1992.

[23] R. K. Young, *Wavelet Theory and Its Applications.* Boston, MA: Kluwer, 1993.

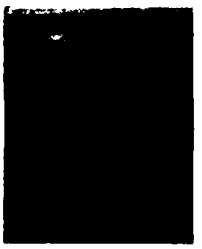
[24] W. Y. Huang and R. C. Rose, "Integrated models of signals and background for an HMM/neural net ocean acoustic events classifier," in *Proc. 25th Ann. Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 1991.

Amlan Kundu received the B.Tech. degree in electronics from Calcutta University in 1979, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara in 1982 and 1985, respectively.

After graduation, he joined the faculty of the State University of New York at Buffalo. From 1991 to 1992, he was affiliated with the Center of Excellence for Document Analysis and Recognition (CEDAR), SUNY Buffalo. Since June 1992, he has been with NCCOSC, San Diego, CA. His research interests include signal/image processing, optimal (Kalman type) filtering in non-Gaussian environments, pattern recognition, and communications. He is particularly interested in pioneering applications of hidden Markov modeling to nonspeech problems such as handwriting recognition, computer vision, and SONAR signal identification. He has written more than 40 refereed scientific papers in his various fields of research interest.

George C. Chen (M'89) received the M.S. degree in computer science from the University of California at Los Angeles in 1981, and the Ph.D. degree in electrical engineering from the University of California at San Diego in 1992.

Prior to 1985, he was a Software Engineer with the Computer Sciences Corporation, El Segundo, CA, and worked on computer networking and database systems. Since 1985, he has been a Research Engineer with the Naval Command, Control, and Ocean Surveillance Center's RDT&E Division (formally Naval Ocean Systems Center), San Diego, CA. His research interests include digital signal processing, communication systems, and neural networks.



Charles E. Persons received the B.A. degree from Hamilton College in 1955, the B.S. and M.S. degrees from the Massachusetts Institute of Technology in 1956 and 1958, respectively, and the Ph.D. degree from the University of California, San Diego, in 1973.

Since 1959, he has worked for the Naval Command, Control, and Ocean Surveillance Center's RDT&E Division (and its predecessors), San Diego, CA. His research interests include characterization of underwater acoustic channels and signals, and enhancement of broadband or narrowband signal detection. He has been responsible for the management of exploratory development projects in passive acoustic detection, classification, and localization of undersea surveillance targets.