(4)

Technical Document 1946
November 1990

Adaptive Mixture Approach to Pattern Recognition

C. E. Priebe
D. J. Marchette

Approved for public release; distribution is unlimited.

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

J. D. FONTANA CAPT, USN
Commander

H. R. TALKINGTON, Acting
Technical Director

ADMINISTRATIVE INFORMATION

This work was performed for the Office of the Chief of Naval Research, Arlington, VA 22217-5000 under program element 0601152N, project number ZW13, and agency accession number DN309032.

Released by
M. C. Mudurian
Advanced Concepts &
Development Branch

Under authority of
J. A. Salzmann, Head
Ashore Command &
Intelligence Centers
Division

CONTENTS

INTRODUCTION

A large number of pattern recognition tasks require the ability to recognize patterns within data when the character of the patterns may change with time. Examples of such tasks are remote sensing, autonomous control, and automatic target recognition in a changing environment. (Titterington *et al.* [1985], Chapter 2, gives a list of tasks to which mixture models have been applied. Many of these tasks, and their variants, fall into the above categories.) These tasks have a common requirement: the need to recognize new entities as they enter the environment. A pattern recognition system must be able to recognize and develop a representation of a new pattern in the environment as well as to change its representation of the statistics of the pattern dynamically.

The adaptive mixtures approach presented here uses density estimation to develop decision funcuons for supervised and unsupervised learning. Much work in performing density estimation in supervised and unsupervised situations has been done. For the most part, this research has centered on approaches that use a great deal of *a priori* information about the structure of the data. In particular, parametric assumptions are often made concerning the underlying model of the data. While these approaches yield impressive results, nonparametric approaches (Tapia and Thompson [1978], Prakasa Rao [1980]) free of *a priori* assumptions can be considered more powerful due to their increased generality and therefore wider applicability. Developing a system for performing unsupervised learning nonparametrically (that is, devoid of restricting assumptions) is a daunting task. In fact, there are many instances in which no system can be assured of performing properly. For example, two classes with identical distributions cannot be identified as such based on purely unsupervised learning. Nevertheless, a nonparametric density estimation approach to unsupervised learning can, in many cases, lead to a general and powerful pattern recognition tool.

In addition to the nonparametric assumption, we also consider the problem of recursive estimation (Titterington *et al.* [1985], Chapter 6; Nevel'son and Has'minskii [1973]; Young and Calvert [1974]). That is, it is assumed that, due to high data rates or time constraints, we must develop our estimates in such a way that they do not require the storage or processing of all observations to date. This also limits our ability to develop optimal estimates, but often is the only approach for a given application.

By virtue of addressing the types of applications that can be termed recursive and nonparametric, we have at once made the problem more difficult and more interesting. The recursive assumption eliminates the possibility of using iterative techniques. It is necessary, by hypothesis, to develop our estimate at time t only from our previous estimate and the newest observation. The nonparametric assumption implies that we cannot make any but the simplest assumptions about our data. Realistic restrictions on processing and memory, as might be imposed in automatic target recognition, remote sensing, and automatic control applications, in conjunction with high data rates, make such applications, and the procedure discussed here, an important subject in pattern recognition.

In this work, we apply statistical pattern recognition concepts to the problem of recursive nonparametric pattern recognition in dynamic environments. We begin with a formal description of pattern recognition in this context. Next, we present a formulation of the learning problem (supervised and unsupervised). While this concept has been studied by many authors (Yakowitz [1969, 1970], O'Neill [1978], Greblicki [1980], Postaire [1982], Niemann and Sagerer [1982], Gowda and Krishna [1979]), little work has been done in the recursive nonparametric unsupervised problem. Adaptive mixtures, a method for performing both supervised and unsupervised learning, is then developed. Similarities exist between adaptive mixtures and potential functions (Aizerman *et al.* [1966]), maximum penalized likelihood (Good and Gaskins [1971], Tapia and Thompson [1978]), and reduced kernel estimators (Fukunaga and Hayes [1989].) We conclude with a discussion of adaptive mixtures and their relationship to learning. The two-category problem is considered throughout. The results can be extended to multicategory problems easily (e.g., successive dichotomies). In addition, univariate assumptions are made in places for clarity. The following discussion will, we hope, allow for meaningful discussion of recursive nonparametric learning as well as provide a useful problem definition and approach from which to begin addressing specific applications.

MOTIVATION

Learning techniques are useful in a broad class of pattern recognition problems. In this section, we discuss their application to problems requiring recursive and nonparametric processing. We begin with a formulation of the general pattern recognition problem. Next, we present density estimation as an approach to pattern recognition. We will then consider the problem of pattern recognition in a dynamic environment and the need for both supervised and unsupervised learning techniques in its solution.

We will adopt the following conventions on notation. A parenthesized superscript, e.g., $C^{(i)}$, will denote an element of an array or set. Time is considered to be discrete, $t = 0,1,2,...$ and a subscript t, e.g., $F_{t,}$ will denote indexing by time. Thus, $D_t^{(i)}$ will indicate that the vector D may be a function of time, while $D^{(i)}$ will be used when D is independent of time.

PATTERN RECOGNITION

Let $\Omega = \{C^{(1)}, C^{(2)}, ..., C^{(N)}\}$ be a set of classes, or patterns. We wish to model the problem of determining which pattern is present at a given time t on the values of a number of attributes or features associated with these patterns. The particular time sequence of patterns presented is unknown. For this reason, it is standard to construct a probabilistic model of pattern recognition. The patterns will be considered as the elementary events in a probability space, constructed as follows: Define on B, the σ-algebra generated by Ω, a probability measure Π_t indexed by time, such that for t fixed, $\Pi_t(\{C^{(i)}\}) = \pi_t^{(i)}$, where

$$\sum_{i=1}^{N} \pi_t^{(i)} = 1$$

Thus, (Ω, B, Π_t) defines a probability space for each t, and $\pi_t^{(i)}$, called the class probability, will correspond to the probability that at time t a given pattern $C^{(i)}$ will be observed.

We assume there are m real-valued features measured at each occurrence of a pattern, yielding the random observation sequence $x_t : \Omega \dashrightarrow \Re^m$ (Euclidean m-space). In many problems of interest, the measurements of the feature values of a particular pattern may vary randomly from one presentation of the pattern to the next. When this occurs, write $x_t(C^{(i)}) = Z_t^{(i)}$, with $Z_t = (Z_t^{(1)}, ..., Z_t^{(N)})$, a (vector-valued) random variable, and the distribution function of $Z_t^{(i)}$ denoted by $F_t^{(i)}$ (we write $Z_t^{(i)} \sim F_t^{(i)}$). If m is the number of features measured at each observation, then each $Z_t^{(i)}$ is an m-vector. The distribution of feature vectors corresponding to a single class models (1) noise in the measurement of features and (2) possible systematic but unknown variations among exemplars of the same class. The randomness in the choice of class and the randomness of the feature measurements give

$$x_t \sim F_t = \sum_{i=1}^{N} \pi_t^{(i)} F_t^{(i)}$$

If $F_\tau = F_{\tau'}$ for all τ, τ', we say the stochastic process X is stationary. If, on the other hand, there exists τ, τ', such that $F_\tau \neq F_{\tau'}$, then X is a nonstationary stochastic process.

The pattern recognition system S is required to map an observation X_t to a pattern $C^{(i)}$. That is, S requires a decision function $d : \Re^m \dashrightarrow \Omega$. Furthermore, at any time $t = \tau$, S may be required to make a decision based on its processing of observations $\{X_t\}_{t=1}^{\tau}$ up to time τ. Generally there is a cost associated with misclassifications, and so d must be chosen to minimize the expected value of some given cost, or loss, function. These conditions are quite reasonable for real-time applications in autonomous control, remote sensing, or automatic target recognition.

The problem at hand then is the construction of the decision function d. This generally involves constructing estimates of the statistics of the problem. Following Kendall [1966], we define two distinct approaches to the tasks to be performed: classification and discrimination. Discrimination can be described as a supervised task. Based on a set of observations for which the true class of origin is known (the teaching set), we wish to construct a method for assigning a new observation of unknown origin to the correct class, yielding a decision function $d : \Re^m \dashrightarrow \Omega$. Classification, on the other hand, is an inherently unsupervised task. Based on a set of observations of unknown class, one decides whether groups exist within this data set. If so, one attempts to construct a method of assigning new observations to the correct class, again constructing a decision function.

In the supervised task, the pattern recognition system is intended to infer the statistics of the problem from the training set, for which it is told from which class the observations are drawn. That is to say, we observe X_t, with the added information that the observation is from a given class $C^{(i)}$. Thus, the supervised problem naturally breaks down to separate estimates, first, of the distribution of the individual classes and, second, of the class probabilities $\pi_t^{(i)}$. In the unsupervised learning problem, it is certainly possible to estimate the distribution F_t *of* x_t, but it is much harder to determine the decomposition of the underlying mixture

$$\sum_{i=1}^{N} \pi_t^{(i)} F_t^{(i)}$$

As one might guess, conditions must be placed on this underlying mixture in order for this problem to be solvable, or even well posed in the sense of the identifiability of the mixture (see Titterington *et al.* [1985].)

Much of what follows pertains to classification and discrimination. While the nature of tasks to be performed becomes more complicated as we build to the dynamic environment scenario, the requirements of our pattern recognition system S can normally be thought of as analogous to these two tasks. In general, we have available a teaching data set $\{x_t\}_{t=1}^{a}$, for which the true class is known, and untagged observations $\{x_t\}_{t=a+1}^{\infty}$, for which the true class is unknown. We wish to perform discrimination based on $\{x_t\}_{t=1}^{a}$ and use the decision function d derived during this process to assign a class to the observations $\{x_t\}_{t=a+1}^{\infty}$. We would like, if possible, to use $\{x_t\}_{t=a+1}^{\infty}$ to update (and improve) $d(\bullet)$. Using these data, for which the true class is unknown, entails unsupervised learning.

DENSITY ESTIMATION FOR STATISTICAL PATTERN RECOGNITION

While there are numerous approaches to constructing decision functions (Fukunaga [1972], Chen [1973]), we concentrate on density estimation techniques. This places the burden on the development of an overall estimate $\hat{D}$ for all observations, and estimates $\hat{D}^{(i)}$ for each $C^{(i)} \in \Omega$. Developing the $\hat{D}^{(i)}$ will allow the straightforward construction of a decision function $d(\bullet)$, namely

$$d(x) \equiv C^{(\alpha)} \ni \hat{D}^{(\alpha)}(x) = \max_{i} \hat{D}^{(i)}(x) \qquad (1)$$

or

$$d(x) \equiv C^{(\alpha)} \ni \hat{\pi}^{(\alpha)}\hat{D}^{(\alpha)}(x) = \max_{i} \hat{\pi}^{(i)}\hat{D}^{(i)}(x) \qquad (2)$$

if an estimate for the prior probabilities π can be made (see Lawoko and McLachlan [1989] for a discussion of bias in the estimation of these priors). Similar equations hold for the time-varying distributions. In the unlikely case where no unique maximum exists, a tie is said to occur and any arbitrary method may be used to break it (e.g., a coin flip). Note that we are not calculating the decision surface directly. Rather, we are evaluating each estimate $\hat{D}^{(i)}x$.

When we use density estimation techniques for developing a decision function, we obtain probability information as well as simple classification information. Furthermore, if we define the probability of misclassification when using a given decision function $d(\bullet)$ as $P_d(e)$, then the Bayesian discriminant d_B minimizes $P_d(e)$, where d_B is defined by

$$d_B(x) = \max_{i} \frac{\pi^{(i)}D^{(i)}(x)}{D(x)} \qquad (3)$$

Finally, convergent estimates, $\hat{D}_t^{(i)}(x)$ -> $D^{(i)}(x)$as n -> ∞, can yield $P_d(e)$ -> $P_B(e) = P_{\text{opt}}(e)$ (see e.g., Wolverton and Wagner [1969], Greblicki and Pawlak [1983]). This adds justification to the use of density estimates in constructing decision functions.

NONPARAMETRIC DENSITY ESTIMATION

There are two basic approaches to density estimation: parametric techniques and nonparametric techniques. The idea behind parametric estimation is to assume that the data come from a family of densities parameterized by a set of parameters (e.g., mean and variance), and to estimate the parameters from the data. This assumes some knowledge of the data which may or may not be available. In nonparametric estimation, the density is not assumed to be from a parameterized family, but is rather assumed to satisfy some other criteria (such as smoothness of the density). A familiar nonparametric technique is the histogram. In the histogram, the choice of bin width determines the "smoothness" of the estimate, in some sense.

Generally speaking, parametric methods are preferable to nonparametric methods if there is good reason to believe the model is correct and a suitable method is available with which to estimate the parameters. Often, however, little is known about the true distribution of the different classes, and so a nonparametric technique is required. Also, for the purposes of classification and discrimination, a fair approximation of the density is often sufficient, and as such, the extra accuracy that might be afforded by a parametric model may not be worth the risk associated with an incorrect model or the computational effort required.

The algorithm described below evolved from the study of two common density estimates. The first of these, the kernel estimator (Parzen [1962], Silverman [1986]), is a truly nonparametric technique that places a fixed density, the kernel, at every data point. This essentially requires the storage of all the data, and an evaluation of the kernel estimator can be quite computationally intensive (although there are techniques to speed up the evaluation, see Silverman [1986]).

$$\hat{D}(x) = \frac{1}{nh}\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right) \qquad (4)$$

In Eq. 4, the kernel K is a probability density function (though this requirement can be relaxed) and the points x^i are the observations. Note that if K is a density, then so is $\hat{D}$. Another useful feature of the kernel estimator is that the smoothness properties (differentiability) of the kernel is inherited by the estimate.

A related technique is called the reduced kernel estimator (RKE) (Fukunaga and Hayes [1989]). Here the idea is to reduce the original data set to a smaller representative set, and then compute the kernel estimator for this smaller set. This reduces the computational complexity of the estimator once the reduced data set has been chosen, but the reduction of the data set can be quite computationally intensive. This idea of reducing the number of kernels in the kernel estimator is the basis for the algorithm described below, though it will be developed from a different perspective than the kernel estimator.

In addition to the storage and computational requirements of the kernel estimator, there is the necessity of the choice of h. This is critical to the performance of the estimator, and a large body of literature addresses this subject. Although the storage and computational requirements are less for the RKE, the choice of h is still critical.

Another technique for density estimation, at least superficially related to the kernel estimator, is known as the method of mixtures:

$$\hat{D}(x) = \sum_{i=1}^{q} \pi_i \, K(x, \hat{\theta}_i) \qquad (5)$$

This is really a parametric technique, which fits the data to a mixture of a fixed number of densities. The resultant estimate requires a knowledge of q, the number of components, or subpopulations, of the data. There is also a nonparametric technique: fitting the data to a mixture. This is nonparametric in the same sense that the histogram is. Rather than fitting a "mixture" of fixed non–overlapping rectangles to the data, as the histogram does, this fits a mixture of a given density to the data. Discussions can be found in Titterington *et al.* [1985], Everitt and Hand [1981], and McLachlan and Basford[1988].

The adaptive mixture approach is a combination of these two ideas. It approximates the density as a mixture, as in the nonparametric version of the mixture method. The number of components, q, is chosen from the data. Like the RKE, the adaptive mixture chooses a subset of the data at which to place a kernel. Unlike the RKE, this is done recursively. Also, unlike the RKE, the points which do not get kernels are still used to update the estimate.

For many problems, it is impractical to process the entire data set at once. The data set may be very large, and a technique which iterates through the data repeatedly would be impractical. The data may be nonstationary, with no *a priori* model of character of nonstationarity available. The problem may require real–time processing, forcing the system to respond to each input with very little processing time allowed. In situations such as these, a recursive technique is preferable to an iterative one. In an iterative technique, the entire data set is processed for each update of the system. Typically, many updates are required for the system to converge to an acceptable solution. In a recursive system, each update requires only the current state of the system and the most recent data point. This greatly reduces the computational complexity, and can allow the tracking of nonstationary distributions, as will be seen below. As intuition indicates, on a fixed data set, an iterative technique which sees the data many times will generally outperform a recursive technique which sees the data once. Often, an iterative technique can be modified to run recursively, and vice versa, so this is more a concern of implementation than of algorithm choice.

The above considerations, in the presence of high data rates, indicate the suitability of a recursive stochastic approximation procedure (as opposed to an iterative procedure) for estimating the various densities $D^{(i)}$. A parametric approach, such as finite mixture models, is well suited to such a task (Titterington [1984]). These procedures will take the form

$$\hat{\theta}_{t+1} = \Phi_t(x_{t+1}; \hat{\theta}_t), \qquad t = 0,\ 1, \ldots \tag{6}$$

where θ represents the parameters to be estimated and $D(x;\theta)$ is the corresponding density. $\hat{\theta}_t$ denotes the estimated parameters after t observations and x_{t+1} denotes the $(t+1)_{st}$ observation. If we are required to give an estimate for $D(x;\theta)$ at time $t = \tau$, our response would therefore be $D(x;\hat{\theta}_\tau)$. Φ_t is the (time-varying) parameter update function, about which much will be said later.

The parametric assumption in Eq. 6 is valid only for situations in which we know that the densities being modelled all belong to a common family, D, or we are willing to make this assumption.

DYNAMIC ENVIRONMENT

Let us now consider the extended problem in which the total number of classes, and thus the distributions $D^{(i)}$ from which our x_t can be drawn, is finite but not constant over time. For simplicity of exposition, we will assume that the densities $D^{(i)}$ are stationary. In order to retain our formalism, we will consider Ω to be the set of all classes which appear during the operation of the classifier, with $|\Omega| = N$. Let N_t be the number of classes present at time t, that is, the number of classes $C^{(i)}$ for which the class probability $\pi_t^{(i)}$ is nonzero. Then a new class entering the environment at time $t = \tau$ corresponds to $N^\tau = N_{\tau-1} + 1$. Let class $C^{(N_\tau)}$ enter into Ω at time $t = \tau$, and remain a member of Ω until time $t = \tau'$. Then $X_t \sim \sum_{i=1}^{N_\tau} \pi_t^{(i)} D^{(i)}$ for $t \in [\tau, \tau')$. That is to say, the observations x_t can be drawn from distribution $D^{(N_\tau)}$ for $\tau \leq t < \tau'$. For $t \notin [\tau, \tau'), X_t \sim \sum_{i=1}^{N_{\tau-1}} \pi_t D^{(i)}$, and x_t will not be drawn from distribution $D^{(N_\tau)}$. Note that, since we are assuming $\pi_t^{(i)} > 0$, the proportions $\pi_t^{(i)} (i = 1, \ldots, N_\tau)$ must be adjusted during the period of time that $C^{(N_\tau)}$ is in our environment $(t \in [\tau, \tau'))$. For the simplest case, the class probabilities $\pi_t^{(i)}$ remain constant in the regions $(t \in [0, \tau - 1],$ and $t \in [\tau, \tau')$. This corresponds to a simple kind of nonstationarity in the overall distribution D that can be termed a "jump" nonstationarity. A good deal of work has been done in detecting changes such as these in stochastic processes in the simplest case (Bansal and Papantoni-Kazakos [1986, 1989]).

Let us now consider additionally that the individual $D^{(i)}$ be nonstationary (that is, time dependent, or drifting). Thus, $D^{(i)}$ is a function of time as well, and we write it $D_t^{(i)}$. Now we allow $D_t^{(i)} \neq D_{\tau'}^{(i)}$ for times $t \neq \tau'$. The density $D_t^{(i)}$ is allowed to change with time, and therefore the ability to track such a change (or drift) is necessary. This condition, together with a dynamic N_t, yields what will be termed a dynamic environment.

FORMULATION OF THE LEARNING PROBLEM

The development of the decision function $d(\bullet)$ can be partitioned into two distinct phases; analogous to the classification, discrimination distinction: teaching supervised and unsupervised learning.

TEACHING SUPERVISED

The first phase is supeıvised learning. A knowledge of the characteristics of the individual densities $D^{(i)}$ (or $D_t^{(i)}$ if the densities are nonstationary) is incorporated into S during supervised learning. This knowledge can be either *a priori* information about the densities or derived from a teaching data set $\{x_t\}_{t=1}^{\alpha}$ for which the true class is known. Depending on a and the time constraints placed on the supervised learning process, it can be performed either iteratively or recursively. There has been much work on this supervised learning problem (Duda and Hart [1973], Sklansky and Wassel [1981]), though a satisfactory general approach for the recursive nonparametric scenario has yet to be developed. The adaptive mixtures described in the next section are developed to help fill this void.

Once this supervised learning has been accomplished, the decision function $d(\bullet)$ can easily be obtained. The decision surfaces generated by the densities are exactly the decision surfaces required by $d(\bullet)$. Note that a loss function L defined on $\Omega \; X \; \Omega$ can easily be incorporated into the construction of the loss fınction d, as sketched in Duda and Hart [1973].

S can now classify new observations x_t $(t > \alpha)$, for which the true class is unknown, using $d(\bullet)$. However, if our teaching set $\{x_t\}_{t=1}^{\alpha}$ is not representative of the untagged observations $\{x_t\}_{t=a+1}^{\infty}$, classification based on the derived decision function d will not be especially good. In fact, if there is information in the untagged observations, then a system which does not use this information to improve its decision function cannot be considered optimal. Furthermore, if the distributions of the known classes drift, or if new classes enter the environment, unsupervised learning must be performed if the system is to perform well in this changing environment.

UNSUPERVISED LEARNING

In the dynamic environment scenario described in the previous section, there are three cases of unsupervised learning that may be considered. These all approach the task of updating the decision function d by using information derived from observations for which the true class is unknown. The first case involves updating the distribution estimate $\hat{D}_t^{(i)}$ corresponding to a known class $C^{(i)}$ for which the teaching observations were not representative. This nonrepresentative assumption implies that one can perhaps improve $\hat{D}_t^{(i)}$, and hence d, by using information contained in the untagged observations $\{X_t\}_{t=a+1}^{\infty}$. Let $\Psi(d_1, d_2)$ be some error function defined on densities d_1 and d_2. Let $D_t^{(i)}$ be the density associated with some $C^{(i)} \in \Omega$. Let $\hat{D}_{\alpha}^{(i)}$ be the estimate of $\hat{D}_{\alpha}^{(i)}$ obtained via $\{x_t\}_{t=1}^{\alpha}$ (supervised learning). If for some time $\beta > \alpha$ we have

$$\Psi(D_t^{(i)}, \hat{D}_{\beta}^{(i)}) < \Psi(D_t^{(i)}, \hat{D}_{\alpha}^{(i)}) \tag{7}$$

then we say unsupervised learning has improved our estimate of $D_t^{(i)}$ relative to Ψ. If, on the other hand

$$\Psi(D_t^{(i)}, \hat{D}_{\beta}^{(i)}) < \Psi(D_t^{(i)}, \hat{D}_{\alpha}^{(i)}) \tag{8}$$

we say that the estimate of $D_t^{(i)}$ has been degraded (relative to Ψ) by the u.l. process.

Also of particular interest in unsupervised learning is the modelling of the classes $C^{(j)} (j > N_{\alpha})$ that entered Ω after teaching was finished. It is clear that based on the teaching set $\{x_t\}_{t=1}^{\alpha}$, there exists no information in $d(\bullet)$ for these classes. All such classes will therefore be termed unknown classes and denoted by $U^{(j)} (j > N_{\alpha})$, and we desire that our *u*.l. algorithm develop model densities for these $U^{(j)}$.

This requires the system to first recognize that an observation does not belong to any of the current classes, and then construct a model for the new class, which it can then update as new points are received.

Modelling the $U^{(j)}$ includes both developing an overall density $\hat{D}^U$ for all observations classified as unknown and partitioning $\hat{D}^U$ into individual densities $\hat{D}^{(j)}$ corresponding to estimates of unknown classes $\hat{U}^{(j)}(j > N_\alpha)$ for which teaching observations are not available. (Notational convention: $U^{(j)}$ indicates a member of Ω for which no teaching observations are available. $\hat{U}^{(j)}$ indicates an unknown class developed by an unsupervised learning system in the course of its modelling of $\hat{D}^U$.)

There are two levels of recognition that S must now perform: recognition that an untagged observation x_t is not from one of the known classes $C^{(i)}$ (and therefore tagging x_t as "unknown"), and distinguishing between $\hat{U}^{(j)}$ and $\hat{U}^{(k)}(j \neq k)$ as the correct unknown class of origin for the observation, if possible. These two recognition levels correspond to using the $\hat{D}^U$ against individual $\hat{D}^{(j)}$ in construction d. The latter task is evidently more difficult than the former, due to the added complexity of partitioning $\hat{D}^U$.

To recap, we have three types of unsupervised learning we would like to incorporate into our pattern recognition system S after the initial supervised phase: updating our density estimates $\hat{D}_t^{(i)}$ associated with known classes $C^{(i)}$, developing an estimate $\hat{D}^{(U)}$ for the overall density of unknown observations, and partitioning $\hat{D}^{(U)}$ into $\hat{D}^{(j)}$ corresponding to individual unknown classes $U^{(j)}$. The adaptive mixtures developed next are applicable to each of these tasks.

DEVELOPMENT OF ADAPTIVE MIXTURE APPROACH

A formulation of the learning problem in four categories has been developed. We will now introduce an approach capable of performing recursive nonparametric learning in each of these categories: adaptive mixtures.

We will develop the adaptive mixture from density estimation techniques of kernel estimation, finite mixture modelling, and stochastic approximation (s.a.). The extension of the adaptive mixture beyond these techniques will allow for the modelling of dynamic environments. For simplicity of exposition, we will focus on the estimation of a single density. This should be thought of as one of the class densities $\hat{D}_t^{(i)}$.

FINITE MIXTURES

Consider for the moment the problem of estimating the components of a Gaussian mixture. That is, we assume that our density is of the form

$$D_t(x) = \sum_{i=1}^{N} \lambda_t^{(i)} \Phi_t(x; \mu_t^{(i)}, \sigma_t^{(i)}) \qquad (9)$$

where n is known, and the $\lambda_t^{(i)}$ sum to 1. We are implicitly assuming here that the data come from a single class, and we are trying to estimate the density for that class. We wish to estimate the parameter vector θ^T which consists of $\lambda, \mu,$ and σ. Let us also assume for the moment that D is stationary. A standard technique for estimating the parameter vector θ^T is to maximize the loglikelihood. We will write an estimate for $D(x)$ with parameter vector θ as $D(x;\theta)$. Following Titterington [1984], we set

$$S(x,\theta) = \frac{\partial}{\partial\theta} \log\ [D(x,\theta)] \qquad (10)$$

and use these likelihood equations to obtain the update formula

$$\hat{\theta}_{t+1} = \hat{\theta} + \alpha_1 S(x_{t+1}; \hat{\theta}_{t)} \tag{11}$$

This can be seen to be a gradient ascent on the loglikelihood surface, and under certain conditions on α_t at and D, we will have convergence to the target density (see Titterington [1984]). An example of this kind of approximation formula, which will be used below, is the following set of recursive update equations (written in component form):

$$\rho_t^{(i)} = \lambda_t^{(i)} \frac{\hat{D}_t^{(i)}(x_t)}{\hat{D}_t(x_t)} \tag{12}$$

$$\lambda_{t+1}^{(i)} = \lambda_t^{(i)} + a_{t+1}^{(i)} \ (\rho_{t+1}^{(i)} \ - \ \lambda_t^{(i)}) \tag{13}$$

$$\mu_{t+1}^{(i)} = \mu_t^{(i,j)} + a_{t+1}^{(i)} \ \rho_{t+1}^{(i)}(x_{t+1}^{(j)} - \mu_t^{(i,j)}) \tag{14}$$

$$\sigma_{t+1}^{(i,j,k)} = \sigma_t^{(i,j,k)} \ + \ a_{t+1}^{(i)} \ \rho_{t+1}^{(i)}[(x_{t+1}^{(j)} - \mu_t^{(i,j)})(x_{t+1}^{(k)}) - \mu_t^{(i,k)}) - \sigma_t^{(i,j,k)}] \tag{15}$$

We use the following convention: the first element of a superscripted list corresponds to the number of the mixture component, and following elements correspond to the indices of the parameter in question. Thus, $\mu^{(i,j,)}$ refers to the j^{th} element of the i^{th} mean vector, and $\sigma^{(i,j,k)}$ is the $(j,k)^{th}$ entry in the i^{th} covariance matrix. In vector notation, the mean and covariance update rules become

$$\mu_{t+1}^{(i)} = \mu_t^{(i)} + a_{t+1}^{(i)} \ \rho_t^{(i)}(x_{t+1} - \mu_t^{(i)}) \tag{16}$$

$$\sigma_{t+1}^{(i)} = \sigma_t^{(i)} + a_{t+1}^{(i)} \ \rho_t^{(i)}[(x_{t+1} - \mu_t^{(i)})(x_{t+1} - \mu_t^{(i)})^T - \sigma_t^{(i)}] \tag{17}$$

We will call Eq. 12-15 and Eq. 17 the "update rule" $U_t(x_{t+1}; \hat{\theta}_t)$.

An obvious choice for $a_{t+1}^{(i)}$ (in the stationary case) is $\left(\sum_{j=1}^{t+1} \rho_j^{(i)}\right)^{-1}$. If n, the number of components in the mixture, is 1, this is just $1/t + 1$, which is the inverse of the number of data points. In general, this can be thought of as the number of points used to update component i. If the density D is not known to be a mixture of Gaussians, however, one might still wish to use the above formulation to find an approximation to the density by such a mixture. In some sense, the kernel estimator is an extreme of this point of view. Thus, one could choose m "large enough," start the estimate with some initial θ, and then recursively update the estimate using the above formula. Assuming that the density is well approximated by such a mixture (which is the case if m is large) and that a reasonable initial estimate is used, this procedure will result in a good estimate of the density.

If an approximation of the density D by a mixture as above is used, the number of components, m, and an initial estimate must be chosen. It would be helpful (and in fact is essential in the nonstationary case) if the algorithm could choose m and the initial estimate recursively from the data. It is this which motivates the algorithm described below.

In order to develop an estimate of the form given by Eq. 9, we will use a combination of the above finite mixture modelling algorithm (the update rule) and a dynamic allocation procedure which allows the algorithm to increase the number of terms in our model if our current estimate fails to account for the current observation. That is, we will add a new term to the mixture, with mean $\mu = x_t$, if circumstances indicate that this is necessary. Otherwise, we will update our estimate $\hat{\theta}_t$ (and hence $\hat{D}_t$). We will call this "create rule" $C_t(x_t + 1; \hat{\theta}_t, k)$ and will describe it shortly. Our s.a. procedure now becomes

$$\hat{\theta}_{t+1} = \hat{\theta}_t + [1 - P_t(x_{t+1}; \hat{\theta}_t)]\ U_t(x_{t+1}; \hat{\theta}_t) + P_t(x_{t+1}; \hat{\theta}_t)\ C_t(x_{t+1}; \hat{\theta}_t, k) \qquad (18)$$

$P(\bullet)$ in Eq. 18 is the "decision-to-add-component" function and takes on values 1 or 0, depending on whether the decision is to add a component or not.

Assuming that the system has decided to add a component, the create rule $C(\bullet)$ for the single-class case is

$$\mu_{t+1}^{(m+1)} = x_{t+1} \qquad (19)$$

$$\sigma_{t+1}^{(m+1)} = \sigma_t^0 \qquad (20)$$

$$\lambda_{t+1}^{(i)} = \lambda_{t+1}^{(i)}(1 - w_t) \quad (\mathrm{i} = 1, \ldots, \mathrm{m}) \qquad (21)$$

$$\lambda_{t+1}^{(m)} = w_t \qquad (22)$$

$$m = m + 1 \qquad (23)$$

An obvious choice for w_t is a_t. Thus, the new component is centered at the observation, given an initial covariance (which may be user-defined or derived from the components in the neighborhood of the observation) and a small proportion. All the other proportions must be updated so that they sum to 1, but otherwise the other components are unaffected. For the multiclass modelling case, $C(\bullet)$ becomes a bit more involved. This situation will be discussed below.

When the decision is made to add a component for each data point, the estimate is similar to the kernel estimator (one-dimensional densities are used for clarity):

$$D_t(x) \ = \ \frac{1}{t}\sum_{i=1}^{N}\frac{1}{\sigma_i}\Phi(x; x_i, \sigma_i) \qquad (24)$$

Putting this into a more familiar notation, we have

$$D_n(x) \ = \ \frac{1}{n}\sum_{i=1}^{N}\frac{1}{h_i}k\ (\frac{x - x_i}{h_i}) \qquad (25)$$

$D_n(x)$ is the estimator considered in Wolverton and Wagner [1969] and Wegman and Davies [1979]. Its consistency is easily established. Thus, in this extreme case, the algorithm is consistent for reasonable choices of the system variables (in this case, w, σ, and K). It is reasonable, therefore, that since the update rule is a recursive maximum likelihood estimator (see Titterington [1984], Dempster *et al.* [1977], Redner and Walker [1984]), and so in some sense improves the estimate between the addition of new components, that if the decision to add a component is properly chosen, the overall system will be consistent. The performance of the estimator obtained by using recursive updates, as opposed to merely always adding another term, is important. The reduction in the number of terms required in the estimate is a storage and computational advantage.

The decision to add a component $P(\bullet)$ an be made in a number of ways. One way is to check the Mahalanobis distance from the observation to each of the components, and if the minimum of these exceeds a threshold (called the create threshold), then the point is in some sense "too far away" from the other components, and a new component should be created. Recall that the Mahalanobis distance between a point x and a component with mean $\mu^{(i)}$ and covariance $\sigma^{(i)}$ is defined by

$$M^{(i)}(x) = (x - \mu^{(i)})^T \sigma^{(i)^{-1}} (x - \mu^{(i)}) \tag{26}$$

Thus, if the create threshold is C, then we create a new component at the point x_{t+1} if

$$M(x_{t+1}) = \min_i \; [M^{(i)}(x_{t+1})] > C \tag{27}$$

Another approach is to let $P(\bullet)$ be determined probabilistically. That is, let $q_{t+1}^{(i)}(x)$ be the probability that a point drawn from the distribution defined by the i^{th} component is further from the mean than x:

$$q_{t+1}^{(i)}(x) = \text{Prob}(|X^{(i)} - \mu^{(i)}| > |x - \mu^{(i)}|) \tag{28}$$

Let $q_{t+1}(x_{t+1})$ be the maximum of these, and let $p_{t+1}(\bullet)$ be a random variable which takes on the value 0 with probability $q_{t+1}(x_{t+1})$ and 1 with probability $1 - q_{t+1}(x_{t+1})$. Thus, if the point is close to the mean of an existing component, the system is unlikely to create a new component, and if it is far from any existing component, it will most likely create a new component. This is the same type of rule as the one described above, but it eliminates the need for the user-defined create threshold.

Other approaches could use the estimated density directly, or the density of the observations in the region of the new observation (computed recursively, e.g., by some function of the proportions of the neighboring components and their relative distances from the new observation).

Windowing

The most common technique for modifying a recursive system to allow the estimation of a nonstationary distribution is to use a window on the observations. This amounts, in the simple case, to setting a_t to some small constant. This puts an exponential window on the data, forcing the system to always treat the newest observation with a certain amount of respect. This approach obviously precludes a system from being consistent, but consistency in modelling nonstationary distributions is meaningless. Consider the general update formula mentioned above, namely

$$\hat{\theta}_{t+1} = \hat{\theta} + \alpha_t \, S(x_{t+1}; \hat{\theta}_t) \tag{29}$$

The conditions placed upon α_t in order for Eq. 29 to be a consistent estimation routine are basically (1) $\Sigma\alpha_t = \infty$ and (2) $\Sigma\alpha_t{}^2 < \infty$. For example, $\alpha_t = t^{-1}$. To implement a windowed estimator and address nonstationarities, consider the perturbation of Eq. 29 to be

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \beta_t S(x_{t+1}; \hat{\theta}_t) \tag{30}$$

where the β_t are such that $B > 0$ is a lower bound for β_t. Then Eq. 30 is a windowed s.a. scheme suitable for modelling nonstationary densities. Note that $\Sigma \beta_t^2 = \infty$, so consistency is unobtainable. However, this provides a window on the data, allowing the estimator to adapt to changes in the underlying density. As an example, let $\beta_t = \max \{t^{-1}, B^{-1}\}$ for constant $B > 0$.

It is important to note that asymptotic considerations, as detailed above, become moot when dealing with nonstationarities. What is of import is the level of performance that can be expected under given conditions. For instance, the variance and bias that can be expected for given window widths under stationary assumptions. This information allows one to evaluate the output of the system.

Extension To The Multiclass Case

The preceding discussion involved modelling a single-class distribution. For the general pattern recognition problem this is clearly insufficient. We must have a method for modelling N classes within the framework of Eq. 9. Consider N separate distributions of the form of Eq. 9 That is

$$D_t(x) = \sum_{j=1}^{N} \pi_t^{(i)} D_t^{(i)}(x) \tag{31}$$

Each of the component densities will be modelled as a mixture as above. Assume that a supervised training set is available, so that an initial estimate can be made for each of the classes. Assume further that all the classes are represented in the initial training set. We can then model the individual class densities as mixtures as above. However, in the general case, where there may be classes which are not represented by the initial training set, we need a mechanism for determining whether a point belongs to an existing class, or whether a new "unknown" class should be created.

Let Λ be the scaled normal: $\Lambda = (2\pi)^{m/2} \det[\Sigma^{(i)}]\Phi^{(i)}$. We now require that the create function utilize an inclusion test $\mathrm{I}(\Lambda^{(i)}, x_{t+1})$, which will be used to allow Eq. 31 to develop new, unknown classes $\hat{U}_t^{(j)}$ recursively. (Note that, for our purposes, a decision to update $[P(\bullet) = 0]$ implies no need for any consideration of inclusion: proportional update takes care of itself.) $I(\bullet)$ can be thought of as a "coveredness-coefficient" and is used to determine if the present observation is predicted by one of the terms in the summation of Eq. 31. (This is analogous to a tail-test, with the proviso that we are testing individual terms in the mixture of Eq. 31 rather than the classes.) If the model of Eq. 31 fails this test for all classes, and the creation of a new term is indicated by Eq. 27, then the newly created term will be considered the first member of a new unknown class $\hat{U}^{(N+1)}$. In this case, $C(\bullet)$ is as above, for the single-class case. If, on the other hand, the model passes the inclusion test for the current observation for one or more classes, then the new term will be incorporated into the class or classes for which the observation passes the test. This case will be discussed further.

Note that for D_n defined as in Eq. 25 $\lim_{n \to \infty} P_{D_n}(e) = P_{d_{opt}}(e)$. Thus, from an asymptotic point of view, there is justification in using this density-estimation technique for constructing our decision function in the fully supervised, multiclass case. In addition, we have implications for phase one of unsupervised learning: the development of the overall density estimate $\hat{D}$.

Specifically, we let $I(\Lambda^{(i)}, x_{t+1})$ be a random variable such that $\mathrm{I}(\Lambda^{(i)}, x_{t+1}) = 1$ if $\Lambda^{(i)}, (x_{t+1}) \geq T_I$ and $\mathrm{I}(\Lambda^{(i)}, (x_{t+1}) = 0$ if $\Lambda^{(i)}(x_{t+1}) < T_I$ for some threshold $T_I \leq 1$. If $\sum_i \mathrm{I}(\Lambda_{(i)}, x_{t+1}) = 0$, an "unknown class" will be created, as described above. If, on the other hand, $\sum_i \mathrm{I}(\Lambda_{(i)}, x_{t+1})$ is nonzero, $C(\bullet)$ then is as in the single-class case (Eq. 19 and 23), with the following exception: $C_{t+1} = \sum_i \mathrm{I}(\Lambda^{(i)}, x_{t+1})$ terms are created, one corresponding to each class $C^{(i)} \ni \mathrm{I}(\Lambda^{(i)}, x_{t+1}) = 1$, each with $\lambda = W_t/C_{t+1}$ (compare Eq. 22).

$\mathrm{I}(\Lambda^{(i)}, x_t)$ attempts to recursively identified the modes or terms in the unsupervised data, and as such cannot be perfect. In implementation, it is possible to use a number of parameters to develop $I(\bullet)$. In particular, a "minimum variance" parameter, σ^*, can be used to aid in making the inclusion decisions. For nonasymptotic reasons, a minimum distance (Mahalanobis) can be useful. (Note that these parameters need not be constant over the entire feature space!) Finally, uncertainty considerations can be made. For instance, when many observations (supervised or unsupervised) have been made in a given sector of feature space, we have more confidence in our estimate. We can therefore reduce our dependence on u.l. in these cases. This is for the stationary case. In the nonstationary case, we may indeed wish to suspend u.l. in certain instances until a "change detector" (such as in Bansal and Papantoni-Kazakos [1986, 1989]) indicates that our estimate is no longer valid. While these considerations are indeed important, they deal mainly with application-specific issues. The point of mentioning them is that they are imperative precisely because there can be no "perfect" u.l. machine!

As an aside, a *classification threshold* T may be beneficial for the ultimate response of the system, although this is technically unnecessary. The problem arises when no known class is "close" to the current observation. In this case (which, by the way, may or may not imply the creation of a new, unknown class), the probability that the observation originates from the closest class (in a Mahalanobis sense) can be high while the likelihood is quite low. This scenario, not uncommon in practice, can yield misleading system responses. However, a classification threshold $0 \leq T$ can be implemented such that $\max_i \hat{D}^{(i)}(x_t) < T$ implies that the system response will include, along with the probabilistic ranking of the classes of origin, the proviso that the observation is either an outlyer or originates from an unknown and as yet unmodeled class.

DISCUSSION

We have presented a statistical pattern recognition approach to both supervised and unsupervised learning which allows a system to update and improve its model of the dynamic environment based on observations for which truth is unknown. The adaptive mixtures presented here are interesting in terms of basic density estimation as well as learning applications. While unsupervised learning can never be completely reliable, the approach outlined above can be tailored to individual applications in such a way as to allow a new dimension in pattern recognition systems.

For the supervised case, analysis of the asymptotic performance of Eq. 18 as a density-estimation technique is found in Priebe and Marchette [1990a]. The relationship of this density estimation to discrimination is discussed in Priebe and Marchette [1990b]. Analysis of the extension of this work to the more general task that we call classification (unsupervised learning) is ongoing. While general performance statements in this area will require severe restrictions on the true densities $D^{(i)}$, we have found that in specific instances, Eq. 18 is capable of all three stages of unsupervised learning discussed at the end of the section titled Formulation of the Learning Problem: (1) updating density estimates $\hat{D}_t^{(i)}$ associated with known classes $C^{(i)}$, (2) developing estimate $\hat{D}^U$ for the overall density of unknown observations, and (3) partitioning $\hat{D}^U$ into $\hat{D}^{(j)}$ corresponding to individual unknown classes $U^{(j)}$. However, in the first and third of these especially, there are no guarantees, based solely on unsupervised learning, that the estimated model corresponds to the true $D^{(i)}$.

REFERENCES

Aizerman, M.A., Braverman, E.M., and Rozonoer, L.T., "The Probability Problem of Pattern Recognition Learning and the Method of Potential Functions," Autom. and Remote Control, Vol. 26, pp 1175-1190, 1966.

Bansal, R.K., and Papantoni-Kazakos, P., "An Algorithm for Detecting a Change in a Stochastic Process," IEEE Trans. Inf. Theory, Vol. IT-32, No. 2, pp 227-235, 1986.

Bansal, R.K., and Papantoni-Kazakos, P. "Outlier-Resistant Algorithms for Detecting a Change in a Stochastic Process," IEEE Trans. Inf. Theory, Vol. IT-35, No. 3, pp 521-535, 1989.

Chen, C-h., *Statistical Pattern Recognition*, Hayden, H.J., 1973.

Dempster, A.P., Laird, N.M., and Rubin, D.B., "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. Royal Statist. Soc., Ser. B, Vol. 39, pp 1-38, 1977.

Duda, R.O., and Hart, P.E., *Pattern Classification and Scene Analysis*, John Wiley & Sons, NY, 1973.

Everitt, B.S., and Hand, D.J., *Finite Mixture Distributions*, Chapman and Hall, London, 1981.

Fukunaga, K., *Introduction to Statistical Pattern Recognition*, Academic Press, NY, 1972.

Fukunaga, K., and Hayes, R.R., "The Reduced Parzen Classifier," IEEE Trans. PAMI, Vol II, No. 4, pp 423-425, 1989.

Good, I.J., and Gaskins, R.A., "Nonparametric Roughness Penalties for Probability Densities, Biometrika," Vol. 58, pp 255-277, 1971.

Gowda, K. C., and Krishna, G., "Learning with a mutualistic teacher," Pat. Rec., Vol. 11, 1979.

Greblicki, W., "Learning to recognize patterns with a probabilistic teacher," Pat. Rec., Vol 12, 1980.

Greblicki, W., and Pawlak, M.,"Almost sure convergence of classification procedures using Hermite series density estimates," Pat. Rec. Letters 2, 1983.

Kendall, M.G., "Discrimination and Classification," in *Multivariate Analysis: Proceedings of an International Symposium Held in Dayton, Ohio, June 14-19*, edited by Krishnaiah, P.R., Academic Press, 1966.

Lawoko. C.R.O., and McLachlan, G.J., "Bias associated with the discriminant analysis approach to the estimation of mixing proportions," Patt. Recog., Vol. 22, No. 6, 1989.

McLachland, G.J., and Basford, K.E., *Mixture Models*, Marcel Dekker Inc., NY, 1988.

Nevel'son, M. B., and Has'minskii, R. Z., *Stochastic Approximation and Recursive Estimation*, Translations of Mathematical Monographs, Vol. 47, American Mathematical Society, R.I, 1973.

Niemann, H., and Sagerer, G., "An Experimental Study of Some Algorithms for Unsupervised Learning," IEEE Trans. Patt. Anal. and Mach. Intel., Vol. 4, No. 4, 1982.

O'Neill, T.J., "Normal Discrimination with Unclassified Observations," J. Amer. Stat. Soc., Vol. 73, No. 364, 1978.

Parzen, E., "On the Estimation of a Probability Density and Mode," Ann. Math. Stat., Vol 33, pp 1065-1076, 1962.

Postaire, J.-G., "An unsupervised Bayes classifier for normal patterns based on marginal densities analysis," Pat. Rec., Vol. 15, No. 2, 1982.

Prakasa Rao, B.L.S., *Nonparametric Functional Estimation*, Academic Press, Orlando, FL, 1983.

Priebe, C.E., and Marchette, D.J., "Adaptive Mixture Density Estimation," submitted to *Pattern Recognition*, 1990 (a).

Priebe, C.E., and Marchette, D.J., "Adaptive Mixture Discriminant Analysis," submitted to Pattern Recognition, 1990 (b).

Redner, R.A., and Walker, H.F., "Mixture Densities, Maximum Likelihood and the EM Algorithm," SIAM Review, Vol 26, No. 2, 1984.

Silverman, B.W., *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.

Sklansky, J., and Wassel, G.N., *Pattern Classifiers and Trainable Machines*, Springer-Verlag, NY, 1981.

Tapia, R. A., and Thompson, James R., *Nonparametric Probability Density Estimation*, The Johns Hopkins University Press, Baltimore, MD, 1978.

Titterington, D.M., "Recursive Parameter Estimation Using Incomplete Data," Royal Stat. Soc., Ser. B, Vol. 46, pp 257-267, 1984.

Titterington, D.M., Smith, A.F.M., and Makov, U.E., *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, 1985.

Wegman, E.J., and Davies, H.I., "Remarks on Some Recursive Estimators of a Probability Density," Annals of Statistics, Vol. 7, No. 2, pp 316-327, 1979.

Wolverton, C.T., and Wagner, T.J., "Asymptotically Optimal Discriminant Functions for Pattern Classification," IEEE Trans. Inf. Theory, Vol. IT-15, No. 2, pp 258-265, 1969.

Yakowitz, S., "Unsupervised Learning and the Identification of Finite Mixtures," IEEE Trans. Info. Th., Vol. 16, pp 330-338, 1970.

Yakowitz, S., "A Consistent Estimator for the Identification of Finite Mixtures," Ann. Math. Stat., Vol 40, pp 1728-1735, 1969.

Young, T.Y., and Calvert, T.W., *Classification, Estimation and Pattern Recognition*, American Elsevier, NY, 1974.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1990

3. REPORT TYPE AND DATES COVERED
Final: Oct 1989 — Sept 1990

4. TITLE AND SUBTITLE
ADAPTIVE MIXTURE APPROACH TO PATTERN RECOGNITION

5. FUNDING NUMBERS
0601152N
ZW13
DN309032

6. AUTHOR(S)
C. E. Priebe and D. J. Marchette

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Ocean Systems Center
San Diego, CA 92152–5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NOSC TD 1946

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of the Chief of Naval Research
Arlington, VA 22217–5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A method is developed of performing pattern recognition (discrimination and classification) that uses a recursive technique derived from mixture models, kernel estimation, and stochastic approximation.

14. SUBJECT TERMS

unsupervised learning	mixture model
density estimation	stochastic approximation
kernel estimator	recursive estimation

15. NUMBER OF PAGES
21

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

INITIAL DISTRIBUTION

Code 0012	Patent Counsel	(1)
Code 014	A. Gordon	(1)
Code 0144	R. November	(1)
Code 40	R. C. Kolb	(1)
Code 42	J. A. Salzmann	(1)
Code 421	M. Mudurian	(1)
Code 421	C. Priebe	(10)
Code 921	J. Puleo	(1)
Code 961	Archive/Stock	(6)
Code 964B	Library	(3)

Defense Technical Information Center
Alexandria, VA 22304-6145 (4)

NOSC Liaison Office
Washington, DC 20363-5100 (1)

Center for Naval Analyses
Alexandria, VA 22302-0268 (1)