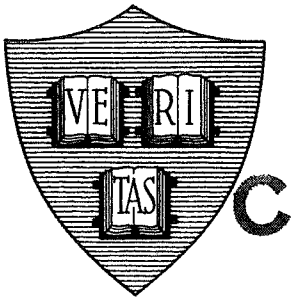
PATTERN CLASSIFICATION APPLIED TO ELECTRO-ENCEPHALOGRAPHS

Technical Report No. 1



By

J. L. Poage and K. P. S. Prabhu

September 1969

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Prepared under Grant NGR 22-007-143

Division of Engineering and Applied Physics

Harvard University • Cambridge, Massachusetts

PATTERN CLASSIFICATION APPLIED TO ELECTRO-ENCEPHALOGRAPHS

By

J. L. Poage and K. P. S. Prabhu

Technical Report No. 1

September 1969

Prepared under Grant NGR 22-007-143
Division of Engineering and Applied Physics
Harvard University Cambridge, Massachusetts

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

PATTERN CLASSIFICATION APPLIED TO ELECTRO-ENCEPHALOGRAPHS

By

J. L. Poage and K. P. S. Prabhu

Division of Engineering and Applied Phyiscs

Harvard University Cambridge, Massachusetts

ABSTRACT

Pattern classification techniques are employed to classify an electro-encephalograph (EEG) record into two parts - one part containing evoked responses to photic stimuli and the other part not containing such evoked responses. In Part I a feature reduction method is suggested to decrease the amount of data to be handled in the context of linear classificatory procedures.

In Part II a sequential algorithm is presented to classify unknown samples into one of two linearly inseparable classes. The algorithm is formed from training sets of known classification. An estimate (on an expected value basis) of the probability of making an error in classification is given.

PART I

LINEAR SEPARATING SURFACES AND FEATURE REDUCTION METHODS

1. INTRODUCTION

Electro-encephalography is a field where pattern recognition techniques can be usefully employed. An electro-encephalograph (EEG) is a continuous record of neuro-electric activity, usually in a human subject. Light may be periodically flashed (photic stimulus) into the eyes of the subject, in which case, the portion of the record between two successive flashes is called an 'evoked response' or, simply a 'response'. Various authors[1] have studied the statistical properties of the random process represented by the EEG records. In most cases, EEG records are taken in order to infer something about the state of the subject (examples - normal vs. diseased brain, sleepy vs. alert subject). It is here that the pattern recognition aspect of the problem becomes apparent. The different states of interest are the pattern classes and, each evoked response is a pattern vector, no doubt corrupted by "noise". The present application is simpler; the task is merely to determine the presence or absence of evoked responses. During part of the recording, stroboscopic light is flashed into the eye of the subject at the frequency of the 'alpha rhythm'* (roughly equal to 10 c. p. s.) and hence this part contains evoked responses. For the remaining part of the recording, the flash of light is absent and also the evoked responses; however, the EEG is still roughly periodic at the frequency of the alpha rhythm. (See appendix A for details). The task is to determine whether an evoked response is

---

* The alpha rhythm can be defined as the frequency of the largest spectral component in the power spectrum of the EEG with no stimulus applied.

present or not in any given part of the EEG record, when this information is not available a priori. Here we have the makings of a simple two-class recognition problem.

It is well-known[1] that the presentation of a photic stimulus gives rise, on the average, to a "peaked response which is time-locked to the stimulus". However, this peak is not easily detectable in any single evoked response due to the corrupting influence of "noise". The following method has been employed by physiologists to solve the recognition problem using this time-locking feature.

Let t_p be the time taken by the EEG record in Fig. 1 to attain its maximum positive value after the reference mark (which coincides with the stimulus if one is present). t_p is evaluated for a large number of responses known to belong to the same class. t_p is treated as a random variable; its mean and variance are then computed. If the stimulus were present at the beginning of each waveform, then the variance of t_p tends to be smaller because of the time-locking feature. On the contrary, if the stimulus were absent, there is no time-locking and so the variance of t_p tends to be large. The disadvantage of this method is that a fairly large number of responses are needed to get a good estimate of the variance and hence the decision cannot be reached quickly. Typically, the number of responses needed to make a reliable decision is of the order of 600. The purpose of the present work is to apply pattern recognition techniques in order to reach a decision as quickly as possible.

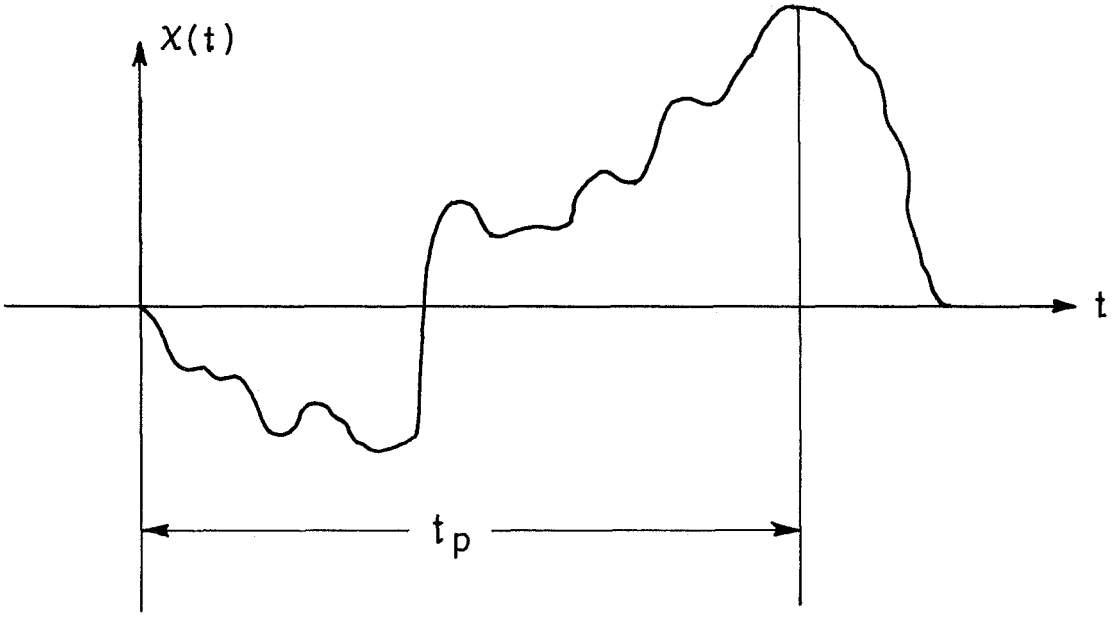


FIG. 1

2. DECISION BASED ON A SINGLE RESPONSE

2.1. Let $\{x(t) : t \in (0, T)\}$ represent any single response. The functions $x(t)$ are assumed to be elements of the normed space C_2 of functions square-integrable on $(0, T)$. This space is to be divided into two sub-classes-class H^0 is the collection of all possible responses when the stimulus is not on and, class H^1 is the collection of all possible responses when the stimulus is on. The usual method of discrimination is to evaluate a scalar functional $\mathcal{F}$ of $x(t)$ and compare it against a threshold. That is

$$\text{if } \mathcal{F}\{x(\cdot)\} < \eta \quad \text{then} \quad x(\cdot) \in H^0$$
$$\text{if } \mathcal{F}\{x(\cdot)\} > \eta \quad \text{then} \quad x(\cdot) \in H^1$$

In this work, attention is restricted to linear functionals only, that is, functionals $\mathcal{L}$ such that

$$\mathcal{L}\{ax(\cdot) + by(\cdot)\} = a\mathcal{L}\{x(\cdot)\} + b\mathcal{L}\{y(\cdot)\}$$

To facilitate digital computer work, the EEG record was sampled at N equi-spaced points between every two successive reference marks. Thus each response $x(t)$ is represented by a N-dimensional vector $(x_1, x_2 \ldots x_N)$. When this is done, discrimination based on a linear functional reduces to the usual hyperplane decision method in N-dimensional space.

$$\text{If } \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_N x_N < \eta \quad \text{then} \quad x \in H^0$$
$$> \eta \quad \text{then} \quad x \in H^1$$

where the α's are a set of weights and η is the threshold.

Now, because of the time-locking characteristic discussed before, we might expect that only a few components of the vector $(x_1, x_2 \ldots x_N)$ play an important part in the decision process. In other words, the decision procedure might work well in a subspace of much lower dimension than N. The procedures which look for such subspaces are known as 'feature reduction procedures' in pattern recognition literature.

2.2. The Feature Reduction Method.

One possible figure of merit which measures the effectiveness of any particular feature for discrimination purposes is the normalised square of the distance between the means (which is also a distance measure between the distributions of the two pattern classes)

$$d_i = \frac{[\mu_i^{(1)} - \mu_i^{(2)}]^2}{\sigma_{ii}^{(1)} + \sigma_{ii}^{(2)}} \qquad (2.1)$$

where $\mu_i^{(j)}$ and $\sigma_{ii}^{(j)}$ denote the mean and the variance of feature x_i under class j.

When we examine the combined effectiveness of a group of features, we have to taken into account the correlations between features. The distance measure generalises to

$$d = (\underline{\mu}^{(1)} - \underline{\mu}^{(2)})^T (\sum^{(1)} + \sum^{(2)})^{-1} (\underline{\mu}^{(1)} - \underline{\mu}^{(2)}) \qquad (2.2)$$

where $\underline{\mu}^{(j)}$ and $\sum^{(j)}$ are the mean vector and covariance matrix of the features under consideration for the distribution of pattern class j. If we let $\underline{\mu} = \underline{\mu}^{(1)} - \underline{\mu}^{(2)}$ and $\sum = \sum^{(1)} + \sum^{(2)}$ equation (2) can be written as

$$d = \underline{\mu}^T \sum^{-1} \underline{\mu} \qquad (2.3)$$

(The quantity of real interest is the error rate (probability of error) in recognition. If the two pattern classes are normally distributed with equal covariance matrices, it can be proved[2] that the error rate is a monotonically decreasing function of the distance measure d).

Equation (2.3) suggests a sequential algorithm for feature selection. At each step, the proposed algorithm would choose the feature which leads to a maximum increase in the distance measure. If n features have already been chosen,

$$d_n = \underline{\mu}^T \sum^{-1} \underline{\mu}$$

where $\underline{\mu}$ is a $n \times 1$ vector of the difference in means and $\sum$ is the $n \times n$ matrix of the sum of the covariances. If a new feature x_{n+1} is added,

$$d_{n+1} = (\underline{\mu}^T \mu_{n+1}) \left(\begin{array}{c|c} \sum & C \\ \hline C^T & \sigma_{n+1} \end{array} \right)^{-1} \left(\begin{array}{c} \underline{\mu} \\ \\ \mu_{n+1} \end{array} \right)$$

where μ_{n+1} is the scalar difference in means for x_{n+1}, σ_{n+1} is the scalar variance of x_{n+1} and C is a $n \times 1$ vector whose components are the covariances between the new feature x_{n+1} and the old features x_1 through x_n. Applying the Frobenius inversion formula† [3] it is easy to see that

† Let $P = \left[\begin{array}{c|c} \underset{m \times m}{A} & \underset{m \times n}{B} \\ \hline \underset{n \times m}{C} & \underset{n \times n}{D} \end{array} \right]$

Then $P^{-1} = \left[\begin{array}{cc} R + RB\Delta^{-1}CR & -RB\Delta^{-1} \\ -\Delta^{-1}CR & \Delta^{-1} \end{array} \right]$

where $\Delta = D - CRB$ and $R = A^{-1}$

$$d_{n+1} - d_n = (\underline{\mu}^T \mu_{n+1}) \left(\begin{array}{c|c} \Sigma^{-1} + \dfrac{\Sigma^{-1} C C^T \Sigma^{-1}}{\sigma_{n+1} - C^T \Sigma^{-1} C} & -\dfrac{\Sigma^{-1} C}{\sigma_{n+1} - C^T \Sigma^{-1} C} \\ \hline -\dfrac{C^T \Sigma^{-1}}{\sigma_{n+1} - C^T \Sigma^{-1} C} & \dfrac{1}{\sigma_{n+1} - C^T \Sigma^{-1} C} \end{array} \right) \left(\begin{array}{c} \underline{\mu} \\ \hline \mu_{n+1} \end{array} \right) - \underline{\mu}^T \Sigma^{-1} \underline{\mu}$$

$$= \frac{(\mu_{n+1} - C^T \Sigma^{-1} \underline{\mu})^2}{\sigma_{n+1} - C^T \Sigma^{-1} C}$$

The feature which maximises $d_{n+1} - d_n$ is chosen as the next feature to be included.

The following facts emerge from this expression for the increase in the distance measure.

i) Since any covariance matrix is at least positive semidefinite, it can be proved that

$$\sigma_{n+1} - C^T \Sigma^{-1} C \geq 0$$

which implies

$$d_{n+1} - d_n \geq 0$$

Therefore, bringing in an additional feature can never worsen the discriminating capacity of the features already chosen.

ii) The increase in distance is zero if and only if the new feature x_{n+1} is a linear combination of the old features $(x_1, x_2 \ldots x_n)$.

iii) The increase in distance is infinite if and only if the two classes are singularly distributed on separate hyperplanes in $(n+1)$ dimensional space.

iv) Since

$$\mu_{n+1} - C^T \Sigma^{-1} \underline{\mu} = E[x_{n+1} | x_1 = x_2 \ldots = x_n = 0]$$

and

$$\sigma_{n+1} - C^T \Sigma^{-1} C = \mathrm{Var}[x_{n+1} | x_1, x_2 \ldots x_n]$$

it is seen that the algorithm chooses the (n+1)th feature in the same manner as it chooses the first one, except that the mean and the variance now refer to the conditional distribution of x_{n+1} given that the previous features x_1 though x_n are all zero.

2.3. Details of the Algorithm.

i) Using a sufficiently large training set, estimates of the $N \times 1$ vector of the difference in means, and the $N \times N$ matrix of the sum of the covariances are obtained.

ii) The first feature x_i is chosen such that

$$\frac{\mu_i^2}{\sigma_{ii}} = \underset{j}{\text{Max}} \left(\frac{\mu_j^2}{\sigma_{jj}} \right)$$

iii) At each subsequent step, the increase in the distance measure $(d_{n+1} - d_n)$ is computed for each of the remaining features. The feature which gives rise to the maximum increase is chosen.

iv) Having chosen the best feature, the inverse covariance matrix $\sum^{-1}$ is updated according to the Frobenius inversion formula.

v) The best separating hyperplane in the subspace spanned by the features chosen so far is

$$\underline{\alpha}_{opt.}^T \underline{x} + \alpha_o = 0$$

where

$$\underline{\alpha}_{opt.} = \left(\sum^{(1)} + \sum^{(2)}\right)^{-1} \left(\underline{\mu}^{(1)} - \underline{\mu}^{(2)}\right) \qquad (2.4)$$

$$\alpha_o = -\frac{1}{2}\left(\underline{\mu}^{(1)} + \underline{\mu}^{(2)}\right)^T \left(\sum^{(1)} + \sum^{(2)}\right)^{-1} \left(\underline{\mu}^{(1)} - \underline{\mu}^{(2)}\right)$$

The weighting vector $\underline{\alpha}_{opt.}$ and the threshold α_o are computed. (The optimality of this hyperplane is discussed in the next section).

vi) The process is stopped either, after all features have been exhausted or, when the increase in distance is smaller than a preset value.

2.4. Optimality of the Algorithm.

It is an inherent characteristic of the feature reduction problem that no sequential algorithm can be truly optimal. This is so because the best subset of n features is not necessarily a subset of the best subset of (n + 1) features. Only by an exhaustive search of all $\binom{N}{n}$ possible feature combinations, at each step, one can construct a truly optimal scheme; however, such an exhaustive search rapidly becomes infeasible as the total number of features increases.

Subject to this qualification, the algorithm is optimal in the sense that, at each step, it picks the particular feature which adds most to the effectiveness of the feature set already chosen. Moreover, the particular weighting vector $\alpha_{opt.}$ given by equation (4) maximises the Fischer criterion[4], which is expressed by

$$\frac{[\underline{\alpha}^T(\underline{\mu}^{(1)} - \underline{\mu}^{(2)})]^2}{\underline{\alpha}^T(\sum^{(1)} + \sum^{(2)})\underline{\alpha}}$$

2.5. Results.

In the actual application, each EEG response was discretised into 100 values.(N = 100). A training set of 2000 responses of known classification, roughly equally divided between the two pattern classes, was used to compute the mean vectors and covariance matrices. The separating surfaces given by the algorithm were tested against a test set of 1000 responses. Fig. 2 shows how the actual error rate comes

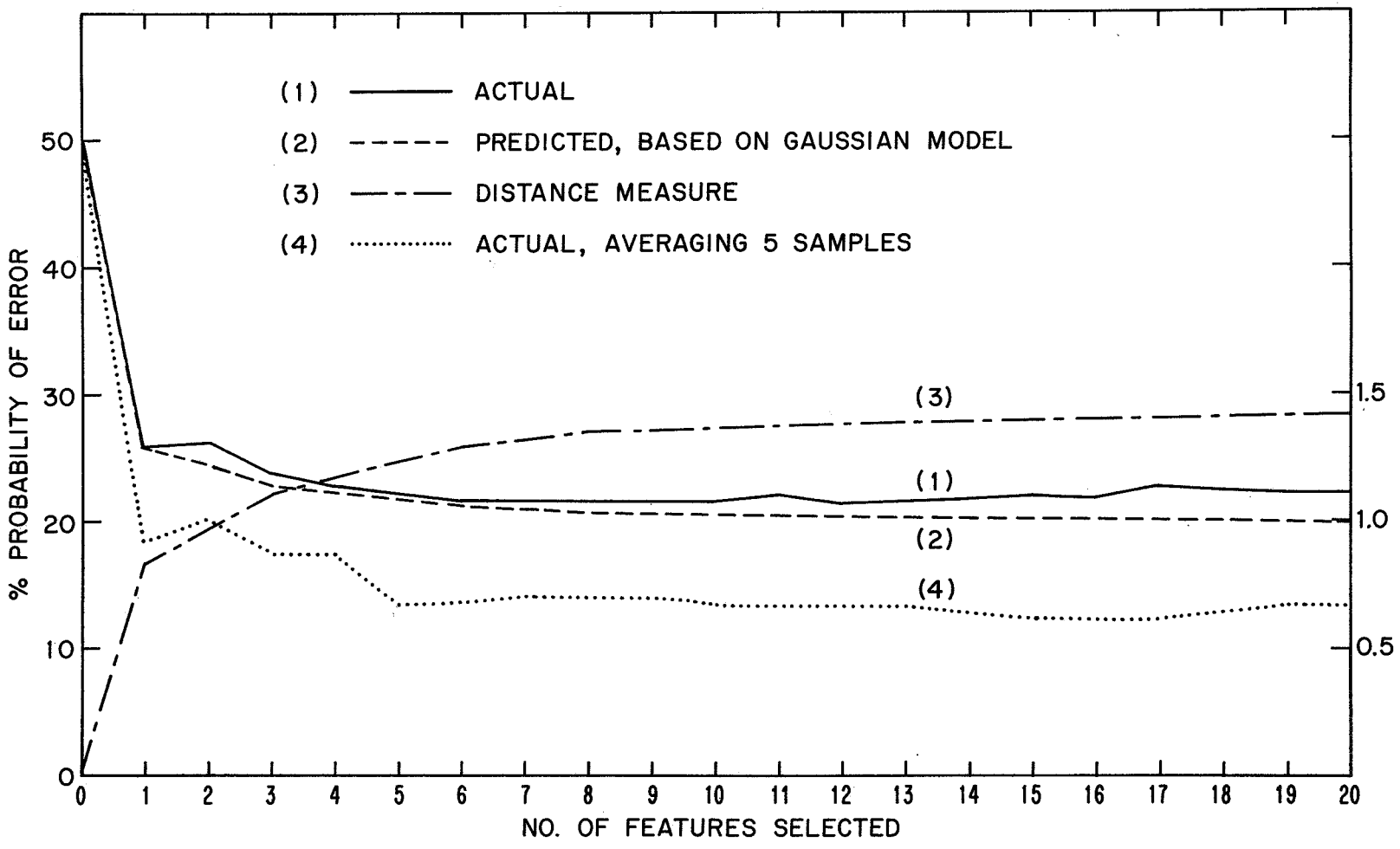

(1) ACTUAL
(2) PREDICTED, BASED ON GAUSSIAN MODEL
(3) DISTANCE MEASURE
(4) ACTUAL, AVERAGING 5 SAMPLES
% PROBABILITY OF ERROR
NO. OF FEATURES SELECTED
0
10
20
30
40
50
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
(3)
(1)
(2)
(4)
1.5
1.0
0.5

FIG. 2

close to the predicted error rate based on a Gaussian model. This fact also seems to justify the stationarity assumption implicit in this approach, since the training and test sets were separated by about two minutes of EEG record.

It is also clear that decisions based on a small number of important features do almost as well as those based on a much larger number of features. The best two features correspond to the positive and negative peaks of the average evoked response.

The error rate cannot be brought below about 20% by decisions based on a single response alone. Fig. 3, which is a plot of the patterns in the two-dimensional subspace spanned by the best two features, explains why this is so. There is considerable overlap between the two classes as a result of the large variances of the features within each pattern class.

3. DECISIONS BASED ON MORE THAN ONE RESPONSE

3.1. We have seen that the poor error rate in recognition results from the large deviations of the individual evoked responses from the average evoked response. One way of circumventing this difficulty would be to average several statistically independent responses known to belong to the same pattern class; the decision would be based on the average response. It is well known that averaging K statistically independent random variables reduces the variance by a factor of K.

The implementation of the averaging operation would be rather difficult. This is so because the responses to be averaged

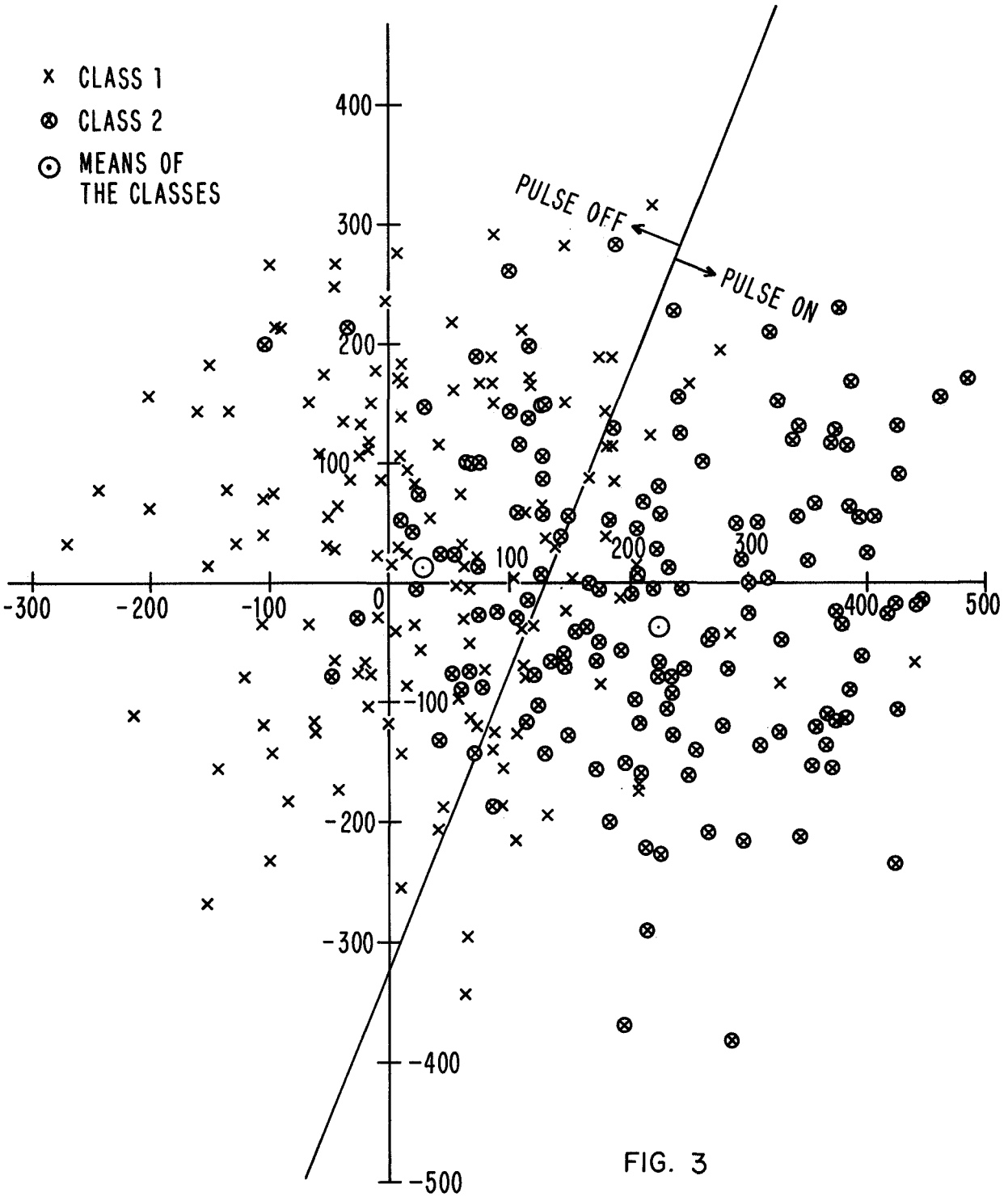

× CLASS 1
⊗ CLASS 2
⊙ MEANS OF THE CLASSES
PULSE OFF
PULSE ON
400
300
200
100
-100
-200
-300
-400
-500
-300
-200
-100
0
100
200
300
400
500

FIG. 3

will have to be picked out at random points in time from the EEG record, if they are to be statistically independent. In other words, decisions cannot be made in real time. And real time operation would be a very desirable characteristic of any practical scheme.

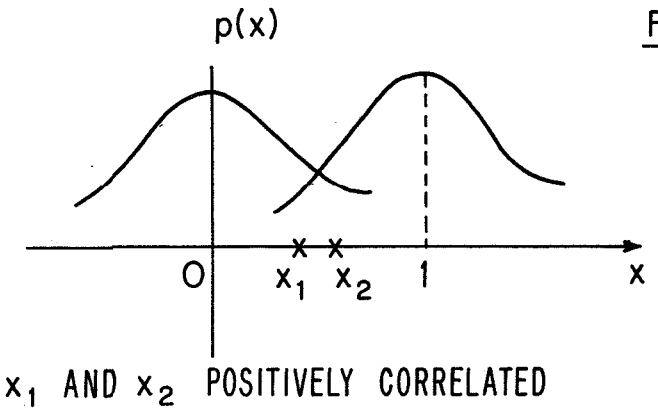
If the decision is to be based on several consecutive responses an the EEG record, we cannot afford to disregard the correlations between them. A simple example in Fig. 4 will illustrate the effects of positive and negative correlations upon the decision-making process. Let us consider the case where the two pattern classes are normally distributed N(0, 1) and N(1, 1) respectively. x_1 and x_2 are two successive measurements from the same class. It is obvious that if x_1 and x_2 are positively correlated, two measurements are not much better than one for discrimination; whereas, if x_1 and x_2 are negatively correlated, they tend to lie on either side of the mean and contain much greater information than x_1 alone.

3.2. The Algorithm.

The algorithm is essentially the same as in the single response case, except that the available set of features now extends over several consecutive responses. Computation of some more second order statistics is necessary. Specifically, the correlation matrices

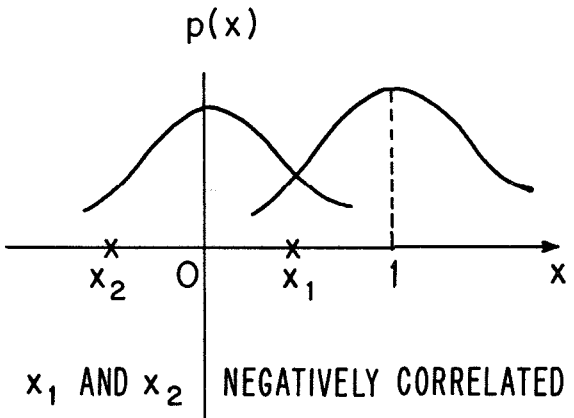
$$C_\tau = E[\underline{x}(t)\underline{x}^T(t+\tau)], \quad \tau = 0, 1, 2, \ldots . T$$

are needed. Here $\underline{x}(t)$ stands for the t^{th} pattern vector (N-dimensional) starting from an arbitrary one and, T is the time beyond which correlations are judged to be insignificant.



x_1 AND x_2 POSITIVELY CORRELATED

FIG. 4



x_1 AND x_2 NEGATIVELY CORRELATED

The single-response algorithm is applied to $\underline{x}(1)$ and the optimum linear combination

$$y_1 = \underline{\alpha}^T_{opt.}(1)\underline{x}(1)$$

is obtained. y_1 is now combined with $\underline{x}(2)$ to yield a $(N+1)$-dimensional pattern vector $(y_1, \underline{x}(2))$ for which the vector of the differnce in means is $(\underline{\alpha}^T_{opt.}(1)\underline{\mu}, \underline{\mu})$ and the matrix of the sum of covariances is

$$\left(\begin{array}{c|c} E[\underline{\alpha}^T(1)\underline{x}(1)]^2 & E[\underline{\alpha}^T(1)\underline{x}(1)\underline{x}^T(2)] \\ \hline E[\underline{x}(2)\underline{x}(1)^T\underline{\alpha}(1)] & E[\underline{x}(2)\underline{x}(2)^T] \end{array}\right) = \left(\begin{array}{c|c} \underline{\alpha}^T(1)C_0\underline{\alpha}(1) & \underline{\alpha}^T(1)C_1 \\ \hline C_1^T\underline{\alpha}(1) & C_0 \end{array}\right)$$

The single response algorithm is again applied to produce the best linear combination of the $(N+1)$ features. The process continues, the pattern vector always being of dimension $(N+1)$

3.3. Results

In the actual application, the responses were discretised into 20 values ($N = 20$) to facilitate storage of several correlation matrices ($T = 19$). Fig. 5 shows how the distance measure increases and predicted error rate (based on a Gaussian model) decreases as more responses are used to arrive at a decision. The actual error rate deviates somewhat from the predicted one (unlike in the single response case). Thus the Gaussian model may not be adequate to describe the joint distribution of several pattern vectors.

Fig. 6 compares the performance of this scheme to the simple averaging technique using the same number of responses and features. It is seen that the averaging method performs poorly for small number

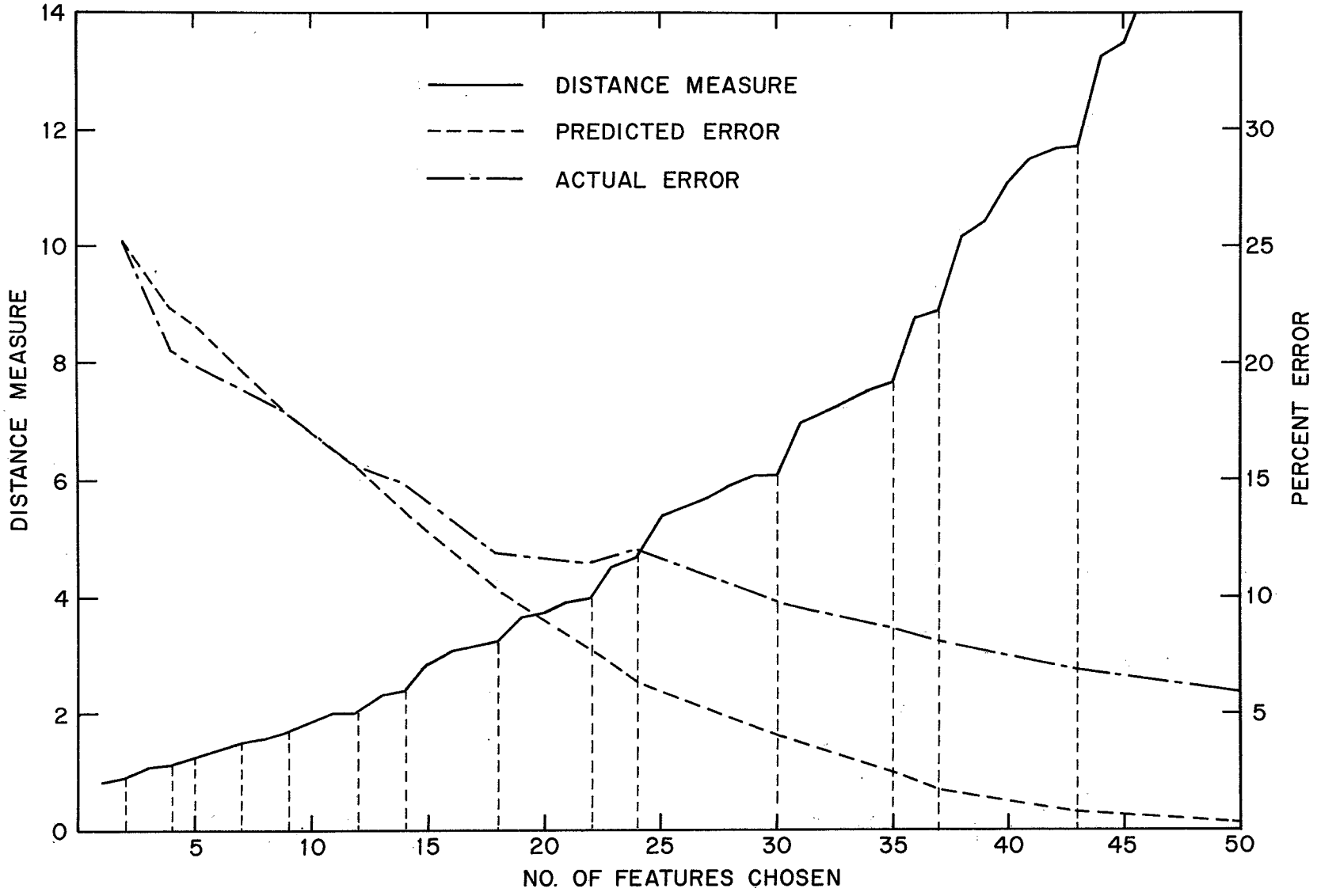

DISTANCE MEASURE
PREDICTED ERROR
ACTUAL ERROR
DISTANCE MEASURE
PERCENT ERROR
NO. OF FEATURES CHOSEN
0
2
4
6
8
10
12
14
5
10
15
20
25
30
35
40
45
50

FIG. 5

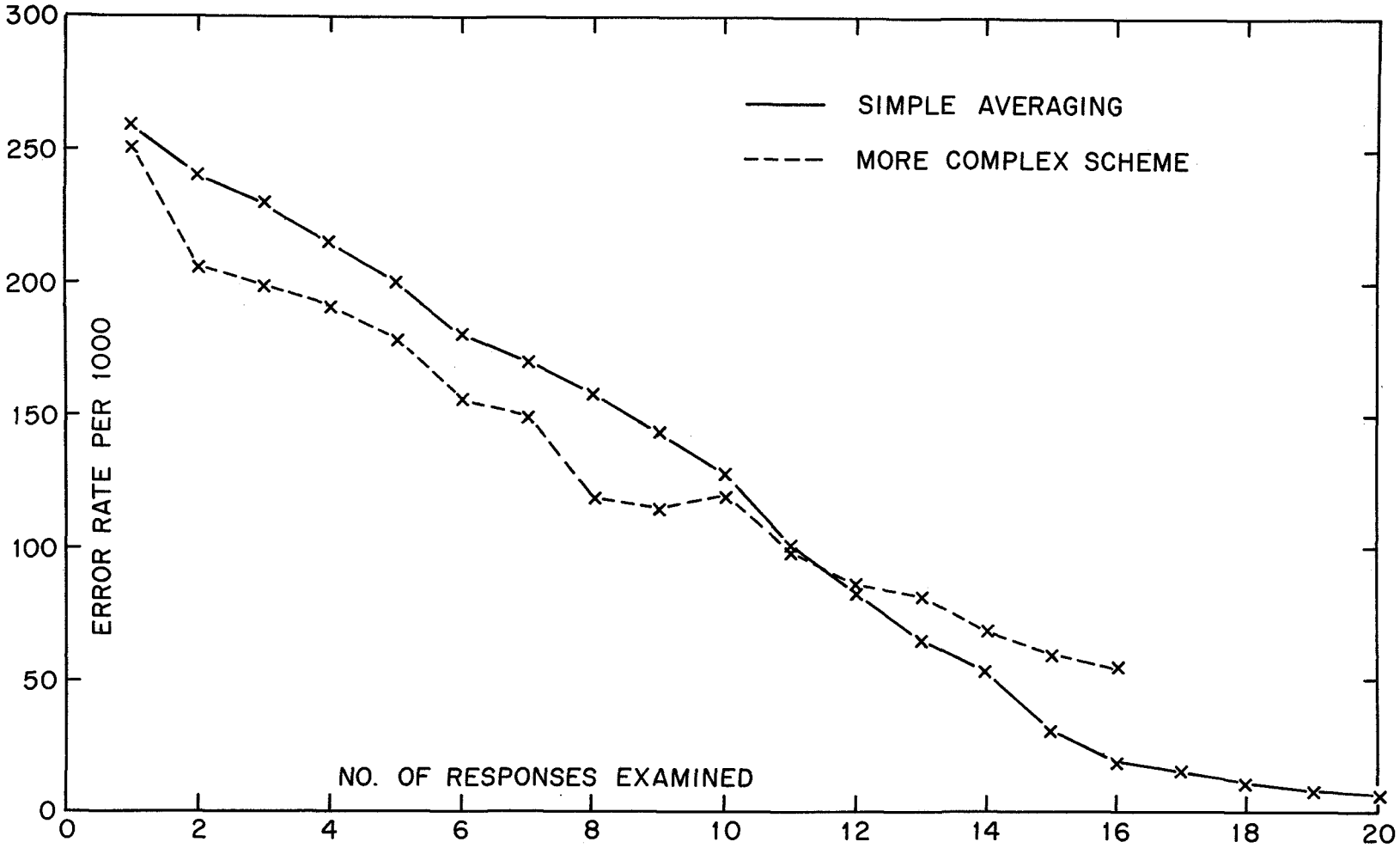

SIMPLE AVERAGING
MORE COMPLEX SCHEME
ERROR RATE PER 1000
NO. OF RESPONSES EXAMINED
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
20

FIG. 6

of responses, but does better than the complex scheme when the number of samples averaged exceeds 12. The explanation for this phenomenon perhaps lies in the assumptions underlying the two schemes. The averaging method requires that the density function $p[\underline{x}(t)]$ be independent of t. The more complex scheme requires that the joint density function $p[\underline{x}(t), \underline{x}(t+1) \ldots \underline{x}(t+T)]$ be independent of t, which is certainly a more stringent assumption (stationarity of higher order) which is less likely to be satisfied in practice. What is more, the order of stationarity required increases as more responses are used to make a decision and, some deterioration in performance is only to be expected.

4. CONCLUSIONS

The techniques of feature reduction and pattern classification have been found useful in detecting the presence of evoked responses in the EEG data. Decisions based on the average of 20 successive responses, using only two features roughly corresponding to the positive and negative peaks of the average evoked responses, produced an error rate of 7/1000.

REFERENCES FOR PART I

1. W. A. Rosenblith, "Processing Neuroelectric Data," M. I. T. Press (1962) and references given therein.

2. T. Marill and D. M. Green, "On the Effectiveness of Receptors in Recognition Systems," IEEE Transactions on Information Theory 1963, pp. 11-17.

3. E. Bodewig, "Matrix Calculus".

4. T. W. Anderson, "An Introduction to Multivariate Statistical Analysis".

PART II

SEQUENTIAL TEST FOR LINEARLY INSEPARABLE CLASSES USING TRAINING SETS

1. INTRODUCTION

Much work has been done in pattern classification with methods formulated with a set of training samples of known classification. As no information is known about the underlying distribution, distribution free methods must be employed. The use of a linear separating hyperplane has been a popular method of solving this problem. When the classes to be recognized are linearly inseparable, there will of course be errors in classification. This report presents a method which attempts to improve the error in classifying linearly inseparable classes by using several of the unknown samples in deciding on a classification. The unknown samples are used sequentially.

The EEG signal described in Appendix A is an example of where this method might be applied. The strobe may be flashed several times so several samples are available for a sequential test. The data obtained with the strobe light on and the data obtained with it off are linearly inseparable.

Sequential methods of hypothesis testing involving the distributions of each class have been used for some time. The sequential probability ratio test (SPRT) developed by Wald[1] has been used extensively. Unknown samples, all from the class to be determined, are taken sequentially and applied to the decision mechanism until classification has been made. The average number of samples required for a decision can be computed, but this does not guarantee that the decision process will terminate in a reasonable number of steps in every case. Chien and Fu[2] have developed a method of

using time varying stopping boundaries with the SPRT to assure termination in a finite time. These methods require knowledge of the underlying distribution of each class and hence are not directly applicable to the case where only training samples of known classification exist. Chien and Fu[2] have formulated a nonparametric sequential recognition method using ordered statistics. The method does give an idea of error probabilities but does not make use of training samples of each class. Rather, it compares unknown samples with samples known to come from one class and decides if the unknown samples are from that class.

This report presents a method which is sequential and makes use of training samples from both classes whose classification is known. The method gives an estimate of the probability of error. Use is made or ordered statistics which find broad application in nonparametric methods.

2. SEQUENTIAL TEST FOR LINEARLY INSEPARABLE CLASSES USING TRAINING SETS

2.1. Assumptions

The method is formulated to decide if an unknown sample belongs to one of two classes which shall be referred to as class 1 and class 2. The following assumptions are made:

i) That a training set is available from each class.

ii) That the samples are independently identically distributed in each class.

iii) That the density function of each class is continuous, so the probability of any two samples being equal is zero.

iv) That several samples all from the same unknown class are available, as the method is to be sequential.

2.2. Ordered Statistics

Use is made in this report of several properties of ordered statistics. A detailed discussion of ordered statistics, including some nonparametric properties and tolerance regions, is given in appendix B. Some properties of ordered statistics and tolerance regions to be used in the report will now be stated without proof[4, 5].

Given a set of n scalar random variables, $(Z_1, Z_2, \ldots, Z_n)$, the points can be arranged in ascending order, $Z_{i_1} < Z_{i_2} < Z_{i_3} < \ldots < Z_{i_n}$. Let $Y_1 = Z_{i_1}, Y_2 = Z_{i_2}, \ldots, Y_n = Z_{i_n}$ so that $Y_1 < Y_2 < \ldots < Y_n$. As Y_i is a random variable, $F(Y_i)$, where $F(x)$ is the distribution function of x, is a random variable. The $P(F(Y_j) - F(Y_i) \geq p)$ can be found and the expection of $F(Y_j) - F(Y_i)$ can be shown to be

$$E(F(Y_j) - F(Y_i)) = \frac{j - i}{n + 1} \qquad j > i \,. \tag{2.1}$$

Thus

$$E(F(Y_{j+1}) - F(Y_j)) = \frac{1}{n + 1} \,. \tag{2.2}$$

It is observed that n ordered statistics partition the density function into n + 1 parts and that the expected value of the per cent of random points that will fall in each segment of the partitioning is $100/(n+1)$. i. e., $F(Y_i)$ is uniformly distributed.

2.3. Expressing Data as Scalars

The use of ordered statistics requires the use of scalars while most data points encountered in pattern recognition are vector quantities. To reduce the sample points to scalars, a linear combination of the features of each data point is taken. Let $(x_1, x_2, \ldots, x_m)$ represent sample point where the x_i's are the features. Take

$$z = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_m x_m \tag{2.3}$$

so z is now a scalar. The α_i's are chosen by using the coefficients from already existing linear separating algorithms. It is beyond the scope of this paper to discuss the various known methods of finding $\{\alpha_i\}$ $i = 1, \ldots, m$ for linear separating hyperplanes. Ho and Agrawala[3] give a survey of many linear separating algorithms. The α_i's used in this paper for the applications were taken from the results of the first part of this report.

In linear separating algorithms, the scalar z,

$$z = \alpha_1 x_1 + \ldots\ldots + \alpha_m x_m$$

is compared to a threshold to decide classification. This report is devoted to finding two thresholds to be used in sequential classification. All data points in this paper, both training samples and unknown samples, will be assumed to have been reduced to scalars. z_i will be used to denote the scalar linear combination for the data points.

2.4. Use of Two Thresholds

While the two classes may be linearly inseparable, one class must be largely above the other class in order for an unknown sample to be classified with at least some degree of accuracy. If the overlap region between the classes is quite large, distinguishing between the classes becomes difficult. The samples of class 2 are taken to lie largely above those of class 1 with a region of overlap between the classes. See Fig. 2.1 for an example. The scalar unknown z is compared with two thresholds which are placed in the overlap region.

If z lies above both thresholds, the unknown sample is as assigned to one class; if z lies below both thresholds, it is assigned to the other class; and if z lies between both thresholds, another sample is taken as z lies in the region of overlap. The procedure is applied to the new sample which is compared with a new set of thresholds. It is assumed that all new samples come from the same class. New samples are taken until a decision is made, and then the algorithm is terminated. Each new sample is compared with a new pair of thresholds.

The thresholds are calculated by using ordered statistics from a training set of each class to give an estimate on the underlying probability distribution of each class. n ordered statistics, n scalar samples arranged in acsending order $z_1 < z_2 < \ldots < z_n$, partition the density function into n+1 parts. On the average the value of the per cent of sample points that fall in each segment of

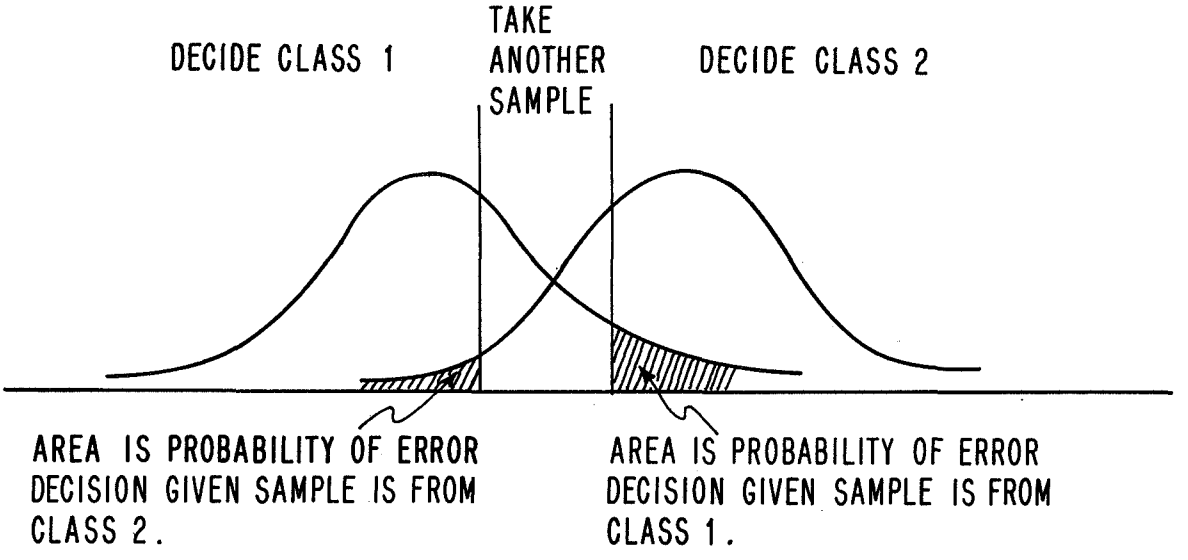

DECIDE CLASS 1
TAKE ANOTHER SAMPLE
DECIDE CLASS 2
AREA IS PROBABILITY OF ERROR DECISION GIVEN SAMPLE IS FROM CLASS 2.
AREA IS PROBABILITY OF ERROR DECISION GIVEN SAMPLE IS FROM CLASS 1.

FIG. 2.1

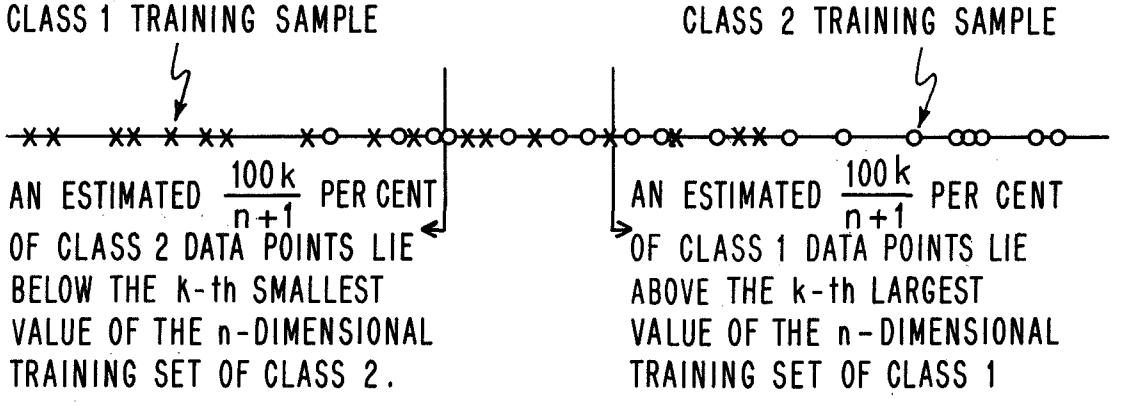

CLASS 1 TRAINING SAMPLE
CLASS 2 TRAINING SAMPLE
AN ESTIMATED $\frac{100k}{n+1}$ PER CENT OF CLASS 2 DATA POINTS LIE BELOW THE k-th SMALLEST VALUE OF THE n-DIMENSIONAL TRAINING SET OF CLASS 2.
AN ESTIMATED $\frac{100k}{n+1}$ PER CENT OF CLASS 1 DATA POINTS LIE ABOVE THE k-th LARGEST VALUE OF THE n-DIMENSIONAL TRAINING SET OF CLASS 1

FIG. 2.2

the partitioning, that is between z_i and z_{i+1}, is $100/(n+1)$.[4] As an example let us say that the densities for each class look like those in Fig. 2.1. As the distributions are not known, it is desired to estimate the error decision regions using the training sets. The scalar samples resulting from taking a linear combination of the features for each class are arranged in ascending order to yield an ordered statistic. The k-th highest number in the ordered statistic gives an estimate, which is an expected value, that $100k/(n+1)$ per cent of all points of this class will lie above that value, or $100(n+1-k)/(n+1)$ per cent lie below that value. See Fig. 2.2. So if the error probability is desired to be p, the $(n+1)p$-th highest number of the ordered statistic should be chosen as the threshold at the upper end of class 1. An estimated

$$\frac{100p(n+1)}{n+1} = 100p$$

per cent of all points of this class will lie above the threshold. As class 2 lies largely above class 1, a threshold to give a desired error region at the lower end of class 2 can be similarly found. Two thresholds are now known. Similar calculations are made for several iterations which give a set of thresholds to be used at each iteration of the sequential algorithm.

The expected value of the per cent of sample points that fall between two neighboring points of the ordered statistic, $100/(n+1)$ can be expected to occur over an average of a large number of different sets of ordered statistics. Work is being done to determine

the effect of the size of the training sets on the accuracy of the estimated probability of error and will appear in the future.

2.5. Setting Thresholds for First Iteration

The n samples, now scalars, from each of the two training sets are ordered separately in ascending magnitude, only one class is shown,

$$z_{i_1} < z_{i_2} < \ldots < z_{i_n}$$

Let the training samples be relabeled

$$y_1 = z_{i_1}, y_2 = z_{i_2}, \ldots, y_n = z_{i_n} .$$

The training samples are now in ascending order,

$$y_1 < y_2 < \ldots < y_n .$$

If z is an unknown sample,

$$\begin{aligned} p(\text{classification error}) = {} & p(\text{classification error} \mid z \in \text{class 1}) p(z \in \text{class 1}) \\ & + p(\text{classification error} \mid z \in \text{class 2}) p(z \in \text{class 2}) \end{aligned} \tag{2.4}$$

The error probabilities for each class, p(classification error $\mid z \in$ class j) j = 1, 2, can be calculated separately. The setting of thresholds will be examined in detail for one class, say class 1. The ordered training set, $y_1 < y_2 < \ldots < y_n$, will be considered to be from that class. The following discussion of setting thresholds applies to either class.

Given the set of ordered statistics from one class,

$$Y_1 < Y_2 < \ldots < Y_n ,$$

the probability that a sample from this class is less than any of the ordered statistics, Y_i, is $F(Y_i)$. From equation (2.1) we have

$$E(F(Y_i)) = \frac{i}{n+1} . \qquad (2.5)$$

An estimated $100i/(n+1)$ per cent of all points lie below Y_i (or $100(n+1-i)/(n+1)$ per cent exceed Y_i.) The overlap region of the inseparable training sets has been taken to be at the higher end of the class 1 ordered statistics and the lower end of the class 2 ordered statistics. See Fig. 2.3. The unknown samples are to be used iteratively. $A(k,1)$ represents the upper threshold, where k represents the number of the iteration of the sequential test, and $A(k,2)$ the lower threshold.

Let the unknown samples to be classified be from class 1 so that p(classification error$|\, z \varepsilon$ class 1) can be calculated. If the first unknown sample lies above both thresholds, it will be classified as belonging to class 2 which would be an error. If it lies below both thresholds a correct classification of class 1 would be made. If it falls between the thresholds, another sample should be taken. See Fig. 2.4. To get an estimate of the probability of a sample from class 1 falling above both thresholds, the number of training samples from class 1 that fall above the threshold $A(k=1,1)$ can be used.

If the threshold $A(k=1,1)$ is set equal to the value of the $n_{e_1}^1$-th largest ordered statistic of the training set of class 1, $A(k=1,1) = Y_{n_1 - n_{e_1}^1 + 1}$, then $n_{e_1}^1 - 1$ training samples lie above $A(k=1,1)$. The superscript on n represents the number of iterations and the subscript the class. This means $100\, n_{e_1}^1/(n_1+1)$ per cent of the density of class 1 lies beyond $A(k=1,1)$ and is in the error region.

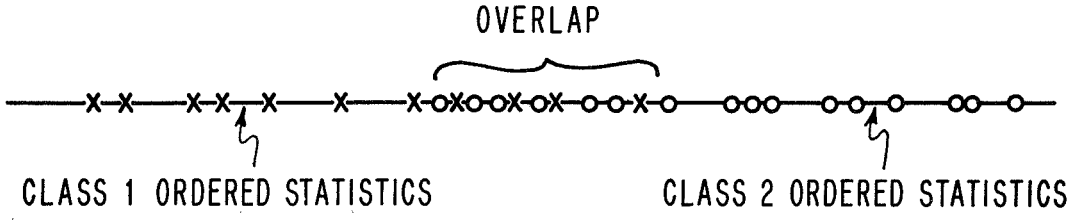


FIG. 2.3

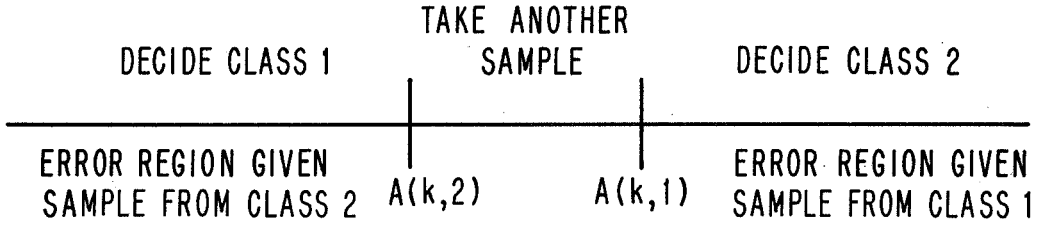


FIG. 2.4

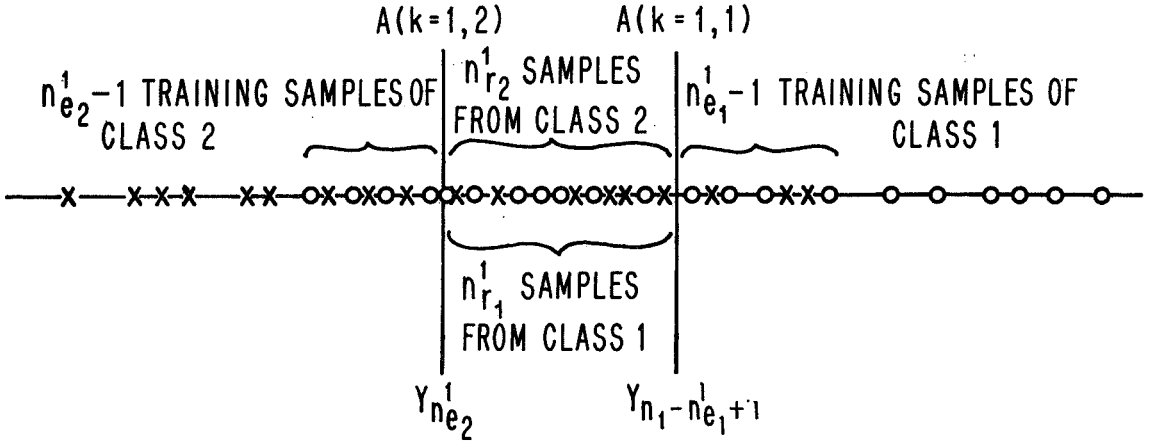


FIG. 2.5

From equation (2.1),

$$E(1 - F(Y_{n_1 - n^1_{e_1} + 1})) = 1 - E(FY_{n_1 - n^1_{e_1} + 1}))$$

$$= 1 - \frac{n_1 - n^1_{e_1} + 1}{n_1 + 1}$$

$$E(1 - F(Y_{n_1 - n^1_{e_1} + 1})) = \frac{n^1_{e_1}}{n_1 + 1} \tag{2.6}$$

If p is the desired probability of error for class 1 on this iteration, then $n^1_{e_1}$ should be chosen so that

$$\frac{n^1_{e_1}}{n_1 + 1} = p$$

$$n^1_{e_1} = (n_1 + 1)p \, . \tag{2.7}$$

$A(k = 1, 1)$ is then set equal to $Y_{n_1 - n^1_{e_1} + 1}$.

$A(k = 1, 2)$, the error threshold for class 2, is determined similarly. As the error region for class 2 lies at the lower end of the ordered statistic, $A(k = 1, 2)$ is set equal to the $n^1_{e_2}$-th lowest ordered statistic of class 2, $A(k = 1, 2) = Y_{n^1_{e_2}}$,

$$E(F(Y_{n^1_{e_2}})) = \frac{n^1_{e_2}}{n_2 + 1} \tag{2.8}$$

where $n^1_{e_2}$ is chosen such that p is the desired error probability for a sample from class 2 on the first iteration.

$$E(F(Y_{n_{e_2}^1})) = p$$

$$\frac{n_{e_2}^1}{n_2 + 1} = p$$

$$n_{e_2}^1 = (n_2 + 1)p \tag{2.9}$$

See Fig. 2.5.

2.6. Thresholds for the Second and Following Iterations

If the sample on the first iteration falls between the thresholds, a second sample is taken. An estimate of the probability of the first sample falling in this region is obtained by counting the number of training samples between the thresholds for each class. Again taking class 1, let $n_{r_1}^1$ be the number of training samples between the thresholds on the first iteration, see Fig. 5. Then an estimated $(n_{r_1}^1 + 1)/(n_1 + 1)$ per cent of the area under the density function for class 1 falls in the region between the thresholds.

Actually the lower threshold is based on class 2 so that there is not one whole interval between class 1 sample points but a fraction of one at the lower end of the region between the thresholds. In practice, $n_{r_1}^1$ is usually large enough that counting the interval as a whole has a negligible effect on $n_{e_1}^2$. This is true for any iteration k.

For a decision to be made resulting in a classification error on the second iteration, the first sample must fall in the region between the thresholds of the first iteration and the second sample in the error region of the second iteration. If p is the desired

probability of error for the second iteration, then we desire

p(1st sample between thresholds)p(2nd sample in error region) = p

$$p(A(k = 1, 2) < z_1 < A(k = 1, 1))p(z_2 < A(k = 2, 1)) = p$$

But we do not know $p(A(k= 1, 2) < z_1 < A(k = 1, 1))$ and $p(z_2 > A(k = 2, 1))$, and so they can only be estimated. So the number of training samples in the error region for the second iteration is chosen as

$$\frac{n^1_{r_1} + 1}{n_1 + 1} \frac{n^2_{e_1}}{n_1 + 1} = p$$

$$n^2_{e_1} = \frac{n_1 + 1}{n^1_{r_1} + 1}(n_1 + 1)p$$

$$n^2_{e_1} = \frac{n_1 + 1}{n^1_{r_1} + 1} n^1_{e_1} \quad \text{from eqn. (2.7)} . \qquad (2.10)$$

$A(k = 2, 1)$ is set equal to the $n^2_{e_1}$ - th largest training sample, $A(k = 2, 1) = Y_{n_1 - n^2_{e_1} + 1}$. $A(k = 2, 2)$ is set similarly using the training samples of class 2.

$$\frac{n^1_{r_2} + 1}{n_2 + 1} \frac{n^2_{e_2}}{n_2 + 1} = p$$

$$n^2_{e_2} = \frac{n_2 + 1}{n^2_{r_1} + 1}(n_2 + 1)p = \frac{n_2 + 1}{n^1_{r_2} + 1} n^1_{e_2} \qquad (2.11)$$

As $\frac{n_1 + 1}{n^1_{r_1} + 1} > 1$, $n^2_{e_1} > n^1_{e_1}$ and $A(k = 2, 1) \leq A(k = 1, 1)$,

also $A(k = 2,2) \geq A(k = 1,2)$. Thus the thresholds for the second iteration will be closer together than the thresholds for the first iteration. $n^2_{e_j}$, $j = 1,2$, may not be an integer. In this case the greatest integer less than $n^2_{e_j}$ is used in setting the threshold. The actual non-integer, $n^2_{e_j}$, though is used in calculating the other error regions.

The number of training samples between the thresholds for the second iteration are counted for each class, $n^2_{r_1}$ and $n^2_{r_2}$. The $n^3_{e_1}$ and $n^3_{e_2}$ can be calculated. For an error decision on the third iteration both, the first and second samples must fall between their respective thresholds, and the third sample must fall in the error region.

The algorithm as presented has taken the probability of error and ending on each iteration to be the same for each class,

$$p(\text{error decision and end on k-th iteration} \mid \text{unknown } \epsilon \text{ class 1})$$
$$= p(\text{error decision and end on k-th iteration} \mid \text{unknown } \epsilon \text{ class 2})$$
$$= p .$$

These could be set equal to different values if so desired. Although then the prior probabilities of which class the unknown sample belongs, p(unknown ϵ class 1) and p (unknown ϵ class 2), should be known in order to calculated the estimated error decision probabilities.

- For error decision on third iteration, $k = 3$,

$$p(A(k = 1,2) < z_1 < A(k = 1,1))p(A(k = 2,2) < z_2 < A(k = 2,1))$$
$$p(z_3 > A(k = 2,1)) = p$$

The estimated form of this equation is

$$\frac{(n_{r_1}^1 + 1)}{(n_1 + 1)} \frac{(n_{r_1}^2 + 1)}{(n_1 + 1)} \frac{n_{e_1}^3}{(n_1 + 1)} = p$$

$$n_{e_1}^3 = \frac{(n_1 + 1)}{(n_{r_1}^2 + 1)} \frac{(n_1 + 1)}{(n_{r_1}^1 + 1)} (n_1 + 1)p$$

$$n_{e_1}^3 = \frac{(n_1 + 1)}{(n_{r_1}^2 + 1)} n_{e_1}^2 \qquad (2.12)$$

$A(k = 3, 1) = Y_{n_1 - n_{e_1}^3 + 1}$.

Similarly, $n_{e_2}^3$ is found, $n_{e_2}^3 = \frac{(n_2 + 1)}{(n_{r_2}^2 + 1)} n_{e_2}^2$ and $A(k = 3, 2) = Y_{n_{e_2}^3}$.

The calculation of the thresholds continues, with the thresholds for each iteration being calculated simultaneously. In general,

$$p(A(k = 1, 2) < z_1 < A(k = 1, 1)) \ldots p(A(k - 1, 2) < z_{k-1} < A(k - 1, 1))$$

$$p(z_k > A(k, 1)) = p \qquad (2.13)$$

The estimated form is

$$\frac{(n_{r_1}^1 + 1)}{(n_1 + 1)} \frac{(n_{r_1}^2 + 1)}{(n_1 + 1)} \cdots \frac{(n_{r_1}^{k-1} + 1)}{(n_1 + 1)} \frac{n_{e_1}^k}{(n_1 + 1)} = p$$

$$n_{e_1}^k = \frac{(n_1 + 1)}{(n_{r_1}^{k-1} + 1)} \frac{(n_1 + 1)}{(n_{r_1}^{k-2} + 1)} \cdots \frac{(n_1 + 1)}{(n_{r_1}^1 + 1)} (n_1 + 1)\, p$$

$$n_{e_1} = \frac{(n_1 + 1)}{(n_{r_1}^{k-1} + 1)} n_{e_1}^{k-1}. \qquad (2.14)$$

$$\text{Similarly} \qquad n_{e_2}^{k} = \frac{(n_2 + 1)}{(n_{r_2}^{k-1} + 1)} \, n_{e_2}^{k-1} . \tag{2.15}$$

A(k, 1) is set equal $Y_{n_1 - r_{e_1}^{k} + 1}$ and A(k, 2) equal $Y_{n_{e_2}^{k}}$.

As $\frac{n_2 + 1}{n_{r_1}^{k-1} + 1} > 1$ and $\frac{n_1 + 1}{n_{r_1}^{k-1} + 1} > 1$, the bounds move closer together.

Eventually, for some k, the thresholds will cross. This happens when $n_{e_1}^{k}$ and $n_{e_2}^{k}$ become sufficiently large that A(k, 2) > A(k, 1). The algorithm will be terminated for this value of k, and the two thresholds are replaced by a common threshold. Let this terminal value of k be called N. A decision will be made at k = N if the algorithm proceeds this far. In the examples to follow the common threshold was set by averaging the k = N - 1 thresholds,

$$A(N, 1) = A(N, 2) = [A(N - 1, 1) + A(N - 1, 2)]/2. \tag{2.16}$$

This could of course be set in other ways.

2.7. Application of Algorithm

In applying the algorithm to unknown samples, each sample is first reduced to a scalar by taking a linear combination of the features. The first sample, z_1 is compared to the thresholds A(k = 1, 1) and A(k = 1, 2). If

$z_1 < A(k = 1, 2)$	decide class 1
$z_1 > A(k = 1, 1)$	decide class 2
$A(k = 1, 2) < z_1 < A(k = 1, 1)$	take another sample

If another sample is taken, z_2, then it is similarly compared to $A(k = 2, 1)$ and $A(k = 2, 2)$. At each iteration that is needed, the bounds for that iteration are used. For any iteration k,

$z_k < A(k, 2)$	decide class 1
$z_k > A(k, 1)$	decide class 2
$A(k, 2) < z_k < A(k, 1)$	take another sample

If the procedure goes until k = N, a decision will be made then as there is only one threshold.

2.8. Estimate of Error Probability

The probability of error can be estimated. The algorithm can only end on one iteration so the events of error decision and end on k-th iteration and of error decision and end on j-th iteration are mutually exclusive for $k \neq j$. Thus the probability of error decision can be expressed as

$$p(\text{error decision}) = \sum_{k=1}^{N} p(\text{error decision and end on k-th iteration}) \tag{2.17}$$

where

$$\begin{aligned} &p(\text{error decision and end on k-th iteration}) \\ &= p(\text{error decision and end on k-th iteration} \mid \text{unknown} \in \text{class 1}) \\ &\qquad p(\text{unknown} \in \text{class 1}) \\ &+ p(\text{error decision and end on k-th iteration} \mid \text{unknown} \in \text{class 2}) \\ &\qquad p(\text{unknown} \in \text{class 2}) \end{aligned} \tag{2.18}$$

by the design of the algorithm, the estimated

$$p(\text{error decision and end on k-th iteration} \mid \text{unknown } \epsilon \text{ class i}) = p \quad \text{for } i = 1, 2 \tag{2.19}$$

So, the expected

$$p(\text{error decision and end on k-th iteration}) = p\, p(\text{unknown } \epsilon \text{ class 1}) + p\, p(\text{unknown } \epsilon \text{ class 2}) = p$$

$$p(\text{error decision}) = \sum_{k=1}^{N} p$$

$$p(\text{error decision}) = Np\,. \tag{2.20}$$

The estimated probability of error is Np. If Np is not near the desired value, p can be varied, which will change N and hence Np.

N is dependent on the value of p chosen. Generally for smaller p, N becomes larger. N is also dependent on the two sets of training samples. If the training sets have a large overlap, N will be large. This is to be expected as the region of indecision is large so more iterations will result. A mathematical bound on Np to assure $Np \leq 1$ has not been found, but for this to occur almost all of the training sets must overlap so that N becomes so large that $Np > 1$. But this problem can not be expected to be solved so a result that $Np > 1$ reflects the unreasonableness of the orginal problem.

Np is the expected value of the probability of error over a large number of training sets. The accuracy of the estimate depends on the size of the training sets and work is being done on this aspect of the problem.

2.9. Remarks

An intuitive explanation for the closing in of the thresholds can be given. In order for the algorithm to proceed to the second iteration, the first sample must fall in the reject region. For a decision to be made resulting in an error or the second iteration, the second sample must fall in the error region. Let p be the desired probability of making a decision which ends in an error at each iteration. To obtain p on the first iteration the probability of falling in the error region should be p. For an error decision to be made on the second iteration, the first sample must fall in the reject region and the second sample in the error regions. The probability of this is $p(z_1 \in$ reject region for k = 1) $p(z_2 \in$ error region for k = 2) = p. As $p(z_2 \in$ reject region for k = 1) < 1, $p(z_2 \in$ error region for k = 2) is greater than $p(z_1 \in$ error region for k = 1). Thus a larger error region for k = 2 than for k = 1 is allowed. The same argument applies for larger k.

3. EXPERIMENTAL RESULTS

The method was tested on the electro-encephalograph (EEG) signals explained in Appendix A of this report. Two data points out of the hundred for each waveform were used. The two features to be used were selected using the feature reduction scheme developed in the first part of this report. Of the hundred features, features eighty-five and fifty-seven were the two most significant features. A linear combination of these features was taken,

$$Y = \alpha_{57}x_{57} + \alpha_{85}x_{85} ,$$

where α_{57} and α_{85} were given by the method explained in the first part of this report with $\alpha_{57} = -.001777$ and $\alpha_{85} = .004431$ being used.

The algorithm was trained on one section of the EEG data and tested on another section. In each section, the samples were taken serially as they were recorded from the patient. Table 3.1 gives error rates on the testing samples for several parameter values. Five hundred testing samples were used in all cases.

From the chart it can be observed that the experimental error rates for class 1 generally do not agree very well with the estimate error rate. In an effort to find an explanation, the roles of the training section of the data and the testing section were reversed with the algorithm trained on what was the testing section and tested on what was the training section. This was done for the case of 199 training samples. The error rates were .04 for class 1 and .158 for class 2 with an estimated rate of .15. The error rate for class 1 is now lower. An examination of the date indicated that there were more higher values in one section of the data than the other for class 1. This would indicate that the data is non-stationary.

Also in the analysis of the first part of the report, it is mentioned that the samples are correlated. The independence assumption of the algorithm is violated. The correlation along with the nonstationarity of the data contribute to the higher than estimated error rates.

TABLE 3.1

Parameters		Experimental Results					
p	Number of training samples	N = maximum number of iterations	Average number of experimental iterations for decision		Estimated error rate=Np	Class 1 (no strobe) experimental error rate	Class 2 (strobe on) experimental error rate
			Class 1	Class 2			
p = .01	99	6	1.9	1.8	.06	.209	.0757
p = .01	199	7	2.22	2.55	.07	.186	.0612
p = .01	399	8	3.42	3.05	.08	.199	.0548
p = .005	199	12	3.68	6.25	.06	.11	.0875
p = .005	399	11	3.91	4.38	.055	.132	.0789
p = .001*	999	40	13.9	20.8	.04	.0556	.0833

* Five features instead of two were used for the p = .001 experiment.

To test the algorithm on data which was independent and uncorrelated, one thousand serial samples of EEG waveforms were mixed together so they no longer appeared serially as they were recorded from the patient. The results for the mixed samples appear in Table 3.2.

The experimental error rates in this case agree more closely with the estimated error rates. This indicates that all the assumptions of the algorithm are not met by the EEG waveforms as they are recorded from the patient.

4. CONCLUSION

The algorithm presented in this report is a sequential variation of the linear separating hyperplane approach to pattern recognition. When several unknown samples are available, the sequential algorithm can be used to give improved results when the two classes are linearly inseparable. The training sets that are used to form a linear separating hyperplane are also used to design a sequential mechanism involving two thresholds. The method gives an estimate (on an expected value basis) of the probability of making an error in classification.

TABLE 3.2

Parameters		Experimental Results					
p	Number of training samples	N = maximum number of iterations	Average number of experimental iterations for decisions		Estimated error rate=Np	Class 1 (no strobe) experimental error rate	Class 2 (strobe on) experimental error rate
			Class 1	Class 2			
p = .01	99	10	4.81	5.15	.1	.0962	.103
p = .01	199	10	4.63	5.88	.1	.0925	.1295
p = .01	399	10	3.68	4.58	.1	.054	.11
p = .005	199	16	5.95	8.33	.08	.0357	.12
p = .005	399	15	5.05	7.58	.075	.0303	.166

REFERENCES FOR PART II

1. Wald, A., Sequential Analysis, Wiley, New York, 1947.

2. Fu, K. S., Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968.

3. Ho, Y. C. and Agrawala, A. K., "On Pattern Classification Algorithms—Introduction and Survey," Proc. IEEE, Vol. 56, pp. 2101-2114, December 1968; Transactions of Automatic Control, IEEE, Vol. AC-13, No. 6, pp. 676-690, December 1968.

4. Hogg, Robert V. and Craig, Allen T., Introduction to Mathematical Statistics, MacMillan, New York, 1965.

5. Fraser, D. A. S., Nonparametric Methods in Statistics, Wiley, New York, 1957.

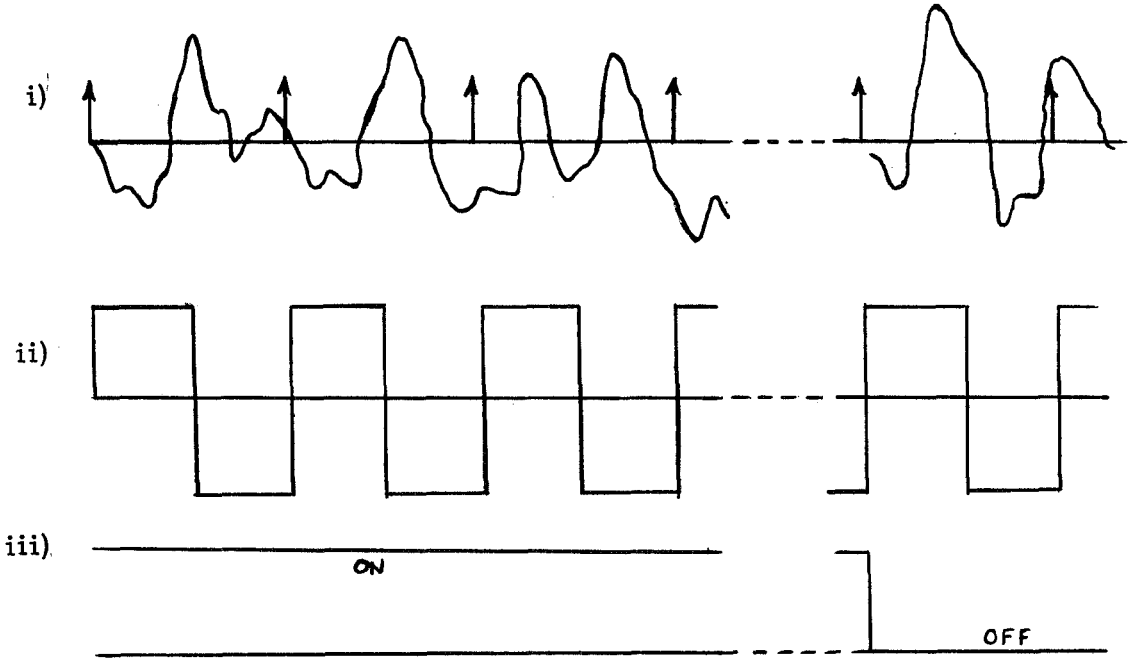
ACKNOWLEDGEMENTS

The authors are grateful to Professor Y. C. Ho for his guidance in this report. The authors also express their appreciation to Dr. J. Anliker of Bio-Technology Division, NASA-ERC for providing the EEG data that stimulated this work.

APPENDIX A

The EEG record of about 10 minutes duration was obtained through NASA. The EEG was recorded from a single person in one sitting. Stroboscopic light is flashed into the eye of the subject at the frequency of the Alpha rhythm (approximately 10 c. p. s.). Thus a flash occurs once every 100 msec. approximately. The portion of the EEG record between successive flashes is considered to be a response. The Stroboscopic light is periodically blocked so it does not reach the eye of the subject. Thus the entire EEG record is split up into groups of two kinds of responses - one with the stroboscopic flash on and the other with the stroboscopic flash off. The on and off periods each last for about 25 seconds and hence contains about 250 responses of the same kind.

The original data were in analog form comprising 3 different waveforms as shown below



i) Is the EEG record itself

ii) Is a square wave at approximately 10 c. p. s. At every leading edge of it, a stroboscopic flash occurs.

iii) Is a 'on-off' waveform which indicates whether or not the light from the stroboscopic flash is reaching the eye of the subject.

To facilitate digital computer work, each of these waveforms was discretised by sampling every millisecond. It is observed that the waveform (ii) serves only to demarcate evoked responses in the continuous EEG record. Each evoked response, being approximately 100 msec. in duration, contains approximately 100 sampled values. It was observed that some of the evoked responses contained more than 100 values due to variations in the frequency of the stroboscope. In such cases only the first 100 values were retained to give pattern vectors of uniform dimension. Waveform (iii) contains only "on-off" information. Therefore, a number 1 or 0, depending on whether the stroboscopic flash is on or off, was augmented to each 100 dimensional pattern vector. The vector of 101 numbers contains all the information for classification purposes.

APPENDIX B

This section presents information on nonparametric methods involving ordered statistics and the density function of the distribution function, F(x). The material in this section can be found in references such as Hogg and Craig[4] and Fraser[5].

First the density function of the distribution function, F(X), will be examined. Let X be a continuous random variable with density function f(x) and distribution function F(x). Then the random variable Z = F(X) has a uniform density function,

$$h(z) = \begin{cases} 1 & 0 < z < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{B.1}$$

This can be shown simply:

$$F(x) = 0 \qquad x \leq a$$

$$= \int_b^x \qquad a < x < b$$

$$= 1 \qquad b \leq x$$

Let z = F(x), then

$$h(z) = f(x)\left|\frac{dx}{dz}\right| = f(x)\,\frac{dx}{dz} > f(x)\,\frac{1}{\frac{dz}{dx}} = f(x)\,\frac{1}{F'(x)} = f(x)\,\frac{1}{f(x)} = 1$$

$$\text{for } 0 < z < 1\,. \tag{B.2}$$

Let X be a random variable. $F(x) = p(X \leq x)$, and as F(X) is a random variable from above $p(F(X) \leq p) = p$. That is, the probability that the random interval $(-\infty, X)$ contains no more than 100p per cent of the probability for the distribution is the same as the probability that F(X) is less than or equal to p.

Now ordered statistics, which play an important role in non-parametric analysis, will be considered. Let $X_1, X_2, \ldots, X_n$ be n independent random samples from a continuous density function. The random variables $F(X_1), F(X_2), \ldots, F(Z_n)$ are independent and each has a uniform distribution on the internal (0, 1).

Let the samples $X_1, X_2, \ldots, X_n$ be rearranged and ordered in ascending order so that $X_{i_1} < X_{i_2} < \ldots < X_{i_n}$. As the density is continuous, $P(X_i = X_j)$ for $i \neq j$ is zero, and the posibility $X_i = X_j$ for $i \neq j$ need not be considered. Let the notation be changed by letting $Y_1 = X_{i_1}, Y_2 = X_{i_2}, \ldots, Y_n = X_{i_n}$. Now $Y_1, Y_2, \ldots Y_n$ is an ordered statistic where $Y_1 < Y_2 < \ldots < Y_n$.

Let $Z_i = F(Y_i)$. As F(x) is monotonically increasing, $Z_1 = F(Y_1) < Z_2 = F(Y_2) < \ldots < Z_n = F(Y_n)$. The joint and marginal distributions of the $\{Z_i\}$ are known and do not depend on the distribution of $\{X_i\}$. The joint density function of the random variables $Z_i = F(Y_i)$, $i = 1, 2, \ldots n$ is

$$f(z_1, z_2, \ldots, z_n) = \begin{cases} n! & 0 < z_1 < z_2 < \ldots < z_n < 1 \\ 0 & \text{elsewhere .} \end{cases} \tag{B. 3}$$

The marginal density of $Z_i = f(Y_i)$ is

$$f(z_i) = \begin{cases} \dfrac{n!}{(i-1)!(n-i)!}\, z_i^{i-1}(1 - z_i)^{n-i} & 0 < z_i < 1 \\ 0 & \text{elsewhere.} \end{cases} \tag{B. 4}$$

The joint density function of $Z_i = F(Y_i)$ and $Z_j = F(Y_j)$, with $i < j$, is

$$f(z_i, z_j) = \begin{cases} \dfrac{n!}{(i-1)!(j-i-1)!(n-j)!}\, z_i^{i-1}(z_j - z_i)^{j-i-1}(1 - z_j)^{n-j} \\ \qquad\qquad 0 < z_i < z_j < 1 \\ 0 \qquad\qquad \text{elsewhere .} \end{cases} \tag{B. 5}$$

A distribution free method to compute the probability that a certain random interval includes or covers a preassigned percentage of the probability of distribution under consideration will be consider. Consider $Z_j - Z_i = F(Y_j) - F(Y_i)$, $i < j$. Now $F(Y_j) - F(Y_i)$ is the probability that x lies between Y_i and Y_j. $F(Y_j) - F(Y_i)$ is a random variable. If $F(Y_j) - F(Y_i) \geq p$, then at least 100p per cent of the probability for the distribution of x lies between y_i and y_j. $\nu = P(F(Y_j) - F(Y_i) > P)$ is the probability that at least 100p per cent of the probability for the distribution if x lies in the random interval (Y_i, Y_j). y_i and y_i are called the 100ν per cent tolerance limits for 100p per cent of the probability for the distribution of x, and (y_i, y_j) is called the 100ν per cent tolerance interval.

As the joint density function of Y_i, Y_j is known, $P(Z_j - Z_i \geq P)$ can be calculated.

$$P(Z_j - Z_i \geq p) = \int_0^{1-p} \int_{p+z_i}^{1} f(z_i, z_j)\, dz_j\, dz_i \tag{B. 6}$$

This is a rather tedious computation so coverages are used. Consider the random variable

$$\begin{aligned} W_1 &= Z_1 = F(Y_1) \\ W_2 &= Z_2 - Z_1 = F(Y_2) - F(Y_1) \\ W_3 &= Z_3 - Z_2 = F(Y_3) - F(Y_2) \\ &\vdots \\ W_n &= Z_n - Z_{n-1} = F(Y_n) - F(Y_{n-1}) . \end{aligned} \tag{B. 7}$$

The random variable W_1 is called a coverage of the random interval $\{x| -\infty < x < y_1\}$ and the random variable W_i, $i = 2, 3, \ldots, n$, is called a coverage of the random interval $\{x| y_{i-1} < x < y_i\}$. The inverse functions of the above transformation are

$$\begin{aligned} z_1 &= w_1 \\ z_2 &= w_1 + w_2 \\ z_3 &= w_1 + w_2 + w_3 \\ &\vdots \\ z_3 &= w_1 + w_2 + \ldots + w_n \end{aligned} \tag{B.8}$$

The Jacobian is equal to one. The joint density function of $Z_1, Z_2, \ldots, Z_n$ is

$$f(z_1, z_2, \ldots z_n) = \begin{cases} n! & 0 < z_1 < z_2 < \ldots < z_n < 1 \\ 0 & \text{elsewhere .} \end{cases} \tag{B.9}$$

The joint density function of n coverages is then

$$f(w_1, w_2, \ldots, w_n) = \begin{cases} n! & 0 < w_i, i - 1, \ldots, n \\ & w_1 + w_2 + \ldots + w_n < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{B.10}$$

Because the density $f(w_1, w_2, \ldots, w_n)$ is symmetric in $w_1, w_2, \ldots w_n$, the distribution of every sum of r, $r < n$, of these n coverages is exactly the same for each fixed value of r. For example, if $i < j$ and $r = j - i$, the distribution of $Z_j - Z_i = F(Y_j) - F(Y_i) = W_{i+1} + W_{i+2} + \ldots + W_j$ is the same as that of $Z_{j-i} = F(Y_{j-i}) = W_1 + W_2 + \ldots + W_{j-i}$. The density of Z_{j-i} is the beta density of the form

$$f(z_{j-i}) = f(z_j - z_i) = \begin{cases} \dfrac{\Gamma(n+1)}{\Gamma(j-1)\Gamma(n-j \in i+1)} z_{j-i}^{j-i+1} (1 - z_{j-i})^{n-j+i} \\ \qquad 0 < z_{j-i} < 1 \\ 0 \qquad \text{elsewhere} \end{cases} \tag{B.11}$$

Consequently

$$P(F(Y_j) - F(Y_i) \geq p) = \int_p^1 f(z_{j-i})\, dz_{j-i} \tag{B.12}$$

Each of the coverages W_i, $i = 1, 2, \ldots, n$, the beta density function

$$f(w) = \begin{cases} n(1-w)^{n-1} & 0 < w < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{B.13}$$

because $W_1 = Z_1 = F(Y_1)$ has this density function. The expectation of each W_i is

$$\int_0^1 nw(1-w)^{n-1}\, dw = \frac{1}{n+1} \ . \tag{B.14}$$

The coverage W_i can be thought of as the area under the graph of the density function between the lines $x = y_{i-1}$ and $x = y_i$. The expected value of each of these random areas W_i, $i = 1, 2, \ldots n$, is $1/(n+1)$. The ordered statistics $y_1, y_2, \ldots, y_n$ partition the probability for the distribution into $n+1$ parts and the expected value of each of these parts is $1/(n+1)$. The expected value of $F(Y_j) - F(Y_i)$, $i < j$, is $(j-i)/(n+1)$ since $F(Y_j) - F(Y_i)$ is the sum of $j - i$ of these coverages.