ACKNOWLEDGMENT

The author wishes to thank Dr. D. Dutta Majumdar, Head of the Laboratory, and Prof. C. R. Rao, F.R.S., Director of the Institute, for their interest in the work and Miss K. Dutta Roy and Mr. R. Kundu for their assistance at different levels.

REFERENCES


A Binary Feature Extraction Technique

KALWANT S. GREWAL AND JOHN D. PATTERSON

Abstract—Numerous schemes are available for feature selection in a pattern recognition problem, but the feature extraction process is largely intuitive. A sequential feature extraction scheme is proposed for binary features. A decision function, which is linear and near optimal, is developed concurrently with each feature. Performance bounds are developed for several design strategies. Experimental results are given to illustrate the use of the scheme and the effectiveness of the performance bounds.

Index Terms—Binary features, character recognition, feature extractions, linear decision.

I. INTRODUCTION

In most pattern recognition problems, the trend is to first extract, intuitively or otherwise, a sufficiently large number of features. Having designed a set of features, an appropriate feature selection technique is applied to select the best subset of features. This set is then used in a decision scheme to classify the patterns. Several feature selection schemes have been developed for different types of features (1), (3), (8), (9), and a large number of classification schemes are also available (2), (3), (5)–(7). However, for most practical problems, the process of feature extraction is primarily intuitive and no general techniques exist for the extraction of features from a given sample set.

In this correspondence a sequential feature extraction procedure is developed. Certain requirements which the feature to be extracted should satisfy at each stage are determined using this scheme. In a feature selection process, a smaller subset of features is selected, sequentially or otherwise, from a group of features which have already been extracted. The technique developed here is used to empirically extract a set of features from a given training set of samples. The feature extraction process is not completely automatic but requires the designer to extract a feature with specified requirements by examining certain samples of the training data.

The decision scheme to be used in the classification process is developed concurrently with each feature. The discussion of the feature extraction scheme in this correspondence is concerned with the two class problem. The extension of the principles to the multiclass case is straightforward and generally consists of dividing the multiclass problem into different two class problems and then solving each of these problems separately.

II. THE FEATURE EXTRACTION SCHEME

The objective is to extract a set of features that can be used to classify samples from classes A and B within a specified percentage error using a feasible decision scheme. The basic approach of this procedure is the following. The first feature is designed to maximize the correct classification using only this feature. In general, the resulting error is too high so that a second feature is designed to separate the misclassified samples of the class with a greater number of errors from the correctly classified samples of the other class. This procedure is repeated with the third and subsequent features until the desired percentage error is achieved.

Let $D_1$ be the decision after j features $x_1, x_2, \ldots, x_j$ have been extracted. A sample is assigned to class A if $D_1 = 1$ and to class B if $D_1 = 0$. If $x_j$ completely separates the two classes and if $x_j$ is the first feature for samples of class A, then the optimal decision point is a very simple one, i.e., $D_j = x_j$. If $x_j$ does not separate A and B completely, then the sample set can be divided into four groups as shown in Fig. 1. A, B, and BC are the misclassified samples using the binary features $x_j$.

To obtain further separation, a second feature $x_2$ is extracted. If $x_2$ is to completely separate the group $A_i$ from $A_j$ and the group $B_i$ from $B_j$, then it should distribute the samples in one of the four ways shown in Fig. 2.

If $x_j$ was the best feature that minimized $P_e$, the percentage error with one feature, then $x_j$ alone cannot separate A and B completely. This implies that $x_j$ cannot separate the groups as shown in Fig. 2. In fact, $x_j$ does not separate the groups as shown in Fig. 2. This cannot be the case if a second feature is to be designed to correct the samples $B_i$. The feature $x_1$ should transfer the samples in $B_i$ to $B_j$ as shown by the solid arrow in Fig. 1. In doing so, it should not transfer $A_i$ to $A_j$ as shown by the dotted arrow. If $a_j$ is the number of samples transferred from $A_i$ to $A_j$ and $b_j$ is the number of samples transferred from $B_i$ to $B_j$, then the feature $x_j$ will reduce the total error only if $b_j > a_j$. It should be noted that only the samples in groups $A_i$ and $B_i$ are used to design $x_j$. Using the first two features, four new groups $A_1, B_1, A_2,$ and $B_2$ are formed. Now if $A_1 > B_1$, then the third feature is to be designed to remove the errors in class A. So $x_2$ is designed using only samples from $A_1$ and $B_1$. It should transfer samples from $A_1$ to $A_2$, and in doing so, avoid transferring samples from $B_1$ to $B_2$. This procedure can be continued to extract features $x_2, x_3$, and so on.

In this procedure, $x_j$ will be a perfect feature if it transfers all samples from $B_1$ to $B_2$ and does not transfer any sample of $A_1$ to $A_2$. If this is the case and further if $x_1 = 1$ for $B_1$ and $x_2 = 0$ for $A_1$, then the samples will be distributed as shown in Fig. 3. It is evident that the error $A_1$ does not change as $x_2$ was designed only to correct $B_1$. In the decision $D_1$ the point (1,1) is assigned to class $B_1$. If the point (0,1) is assigned to $B_1$, then the error at $x_1 = 0$ does not change. So, if $x_1 = 1$, the sample is assigned to class $B_1$ and if $x_1 = 0$, then it is assigned according to $D_1$. This implies that

$$D_1 = D_2 x_2' = x_2 x_1'.$$

(1)

The third feature $x_3$ is to be designed to separate $B_1$ from $A_1$. If $x_2$ can be found such that $x_2 = 1$ for all $A_1$ and $x_2 = 0$ for all $B_1$, or vice versa, then the problem can be solved with three features. The decision $D_3$ can be written as

$$D_3 = D_2 + x_3 = x_3 + x_2 x_1'.$$

(2)

In general, perfect separation is not achieved at each stage and the scheme has to be extended for more features. If $x_j$ is to be extracted to correct the group $A_i$, let $x_j$ be such that it is present in


Manuscript received February 7, 1973; revised August 15, 1973.

Manuscript received February 7, 1973; revised August 15, 1973.
samples of $A_{i}$ and absent in $B_{i}$. All points with $x_{i} = 1$ are then assigned to class $A$ and all points with $x_{i} = 0$ are assigned according to $D_{j-1}$. Then $D_{j}$ can be written as in (3a). Similarly, if $x_{j}$ corrects the group $B_{i}$ then the decision $D_{j}$ can be expressed as in (3b):

$$D_{j} = D_{j-1} + x_{j}, \quad A_{i} > B_{i} \tag{3a}$$

$$D_{j} = D_{j-1} - x_{j}', \quad A_{i} < B_{i} \tag{3b}$$

From Fig. 3 it can be seen that $x_{2}$ is designed to be present in $B_{i}$. Since samples of class $B$ are similar to one another and different from samples of class $A$, more samples of class $B$ can be expected at the point $(0,1)$, than class $A$. If this is true then the decision $D_{2}$ is optimal.

To show that the decision function is linear, the logical decision $D_{j}$ is converted to an algebraic decision $G_{j}$ such that if $G_{j} > 0$, the sample is assigned to class $A$ and if $G_{j} < 0$, the sample is assigned to class $B$. Since $D_{1} = x_{i}$, it is evident that

$$G_{1} = x_{2} - \frac{1}{2}. \tag{4}$$

We also know that $D_{2} = D_{2}x'$. This means that the decision is unaffected if $x_{2} = 0$; but if $x_{2} = 1$, then $G_{2}$ should be less than 0. So the weight of $x_{2}$ should be negative and its magnitude greater than the maximum value of $G_{2}$ which is $\frac{1}{2}$. Then one expression for $G_{2}$ is

$$G_{2} = -x_{2} + x_{1} - \frac{1}{2}. \tag{5}$$

From the expression for $D_{2}$ in (2), it can be seen that the decision is unaffected if $x_{2} = 0$, but if $x_{2} = 1$, then $G_{2}$ should be greater than 0. This implies that the weight of $x_{2}$ must be positive and its magnitude should be greater than the magnitude of the minimum value of $G_{2}$ which is $3/2$. Then one expression for $G_{2}$ is

$$G_{2} = 2x_{2} - x_{1} + x_{2} - \frac{1}{2}. \tag{6}$$

In general, if $x_{i}$ is designed to correct samples of class $A_{j}$ then $D_{j}$ is given by (3a). So, as with $G_{2}$, $G_{i}$ is written as in (7a). If $x_{j}$ is designed to correct samples of class $B_{j}$, then $D_{j}$ is given by (3b) and, as with $G_{2}$, $G_{j}$ is written as in (7b):

$$G_{j} = G_{j-1} - x_{j} \left( \min G_{j-1} - \frac{1}{2} \right), \quad A_{i} > B_{i} \tag{7a}$$

$$G_{j} = G_{j-1} - x_{j} \left( \max G_{j-1} + \frac{1}{2} \right), \quad B_{j} > A_{i}. \tag{7b}$$

It is evident that if $G_{j+1}$ is linear, so is $G_{j}$. Since $G_{i}, G_{j}$ are linear, then $G_{i}$ is linear.

### III. CLASSIFICATION PERFORMANCE

Let $A_{ij}, B_{ij}$ represent the four groups of samples after $j$ features have been extracted. These groups are shown in Fig. 4. Assuming that $B_{ij} > A_{ij}$, the next feature $x_{j+1}$ is designed to remove the error in class $B$. After $x_{j+1}$ is extracted, let the error in class $B$ be $b_{j}$. Also let $a_{j}$ be the additional error generated in class $A$ due to the feature $x_{j+1}$. In the following discussion, $P_{j}$ is the percentage error after extracting $j$ features. Let $\Delta_{j}$ be the ratio of the number of samples in error in the class with the greater number of errors to the total number of samples in error. It can be seen that $\frac{1}{2} \leq \Delta_{j} \leq 1$. Let $r_{j}$ be the ratio of the net errors corrected to the number of errors that the feature $x_{j}$ was designed to correct. For the case shown in Fig. 4,

$$r_{j+1} = \frac{B_{ij} - a_{j} - b_{j}}{A_{ij} + B_{ij}} = \frac{a_{j} + b_{j}}{B_{ij}}.$$ 

The parameter $r_{j}$ is a measure of the effectiveness of a feature and shall be referred to as the power of the feature.

The fractional reduction of error with $x_{j+1}$ is

$$B_{ij} - a_{j} - b_{j} = r_{j+1} \Delta_{j} = \frac{P_{j} - P_{j+1}}{P_{j}}.$$ 

Thus, $P_{j+1} = P_{j}(1 - r_{j+1} \Delta_{j})$. It follows then that the percentage error after $n$ features is

$$P_{n} = P_{1} \prod_{j=1}^{n} (1 - \Delta_{j} r_{j+1}). \tag{8}$$

Simple expressions and bounds for $P_{n}$ can be obtained if either the
power \( r \) or the fractional reduction \( \Delta P_{ji+1} \) is kept constant. In the
following discussion three useful bounds are developed.

A. Constant Fractional Reduction

If each feature has a fractional reduction of \( k \) or higher, then an upper bound on \( P_{a} \) can be obtained by replacing the product \( r_{ji+1}a_{ji} \) by the constant \( k \) in (8). This yields the inequality \( P_{a} \leq P_{1}(1- k)^{n-1} \). So, if the number of features to be designed to achieve a given percentage error is selected, the minimum value of \( k \) required by each feature can be easily computed from this inequality. If each feature is designed such that a fractional reduction of \( k \) or higher is attained, then the percentage error curve will always be below the constant \( k \) curve.

B. Constant Power with Error Reduction in One Class

As discussed in Section II, the first feature is extracted to separate the two classes with a minimum percentage error. The strategy of the design scheme is used only after the first feature has been designed and \( P_{1} \) and \( \Delta \) are known. If \( \Delta \) is very close to unity, then this implies that the error is primarily in one class, say class A. Further if no new errors are allowed to be generated, then the features are designed to remove the errors in only one class. In practice, such a situation would arise if it is considerably more difficult to correct the errors in one class than it is in another.

Let the features be designed to have a constant power \( r \). Let the error in class A after \( j \) features be \( a_{ji} \) and the constant error in class B be \( b_{ji} \). The error corrected with feature \( x_{ji+1} \) is \( r_{ji}a_{j} \); so the error in class A after \( j+1 \) features is

\[
a_{ji+1} = a_{ji} - (1-r) a_{ji}.
\]

Then,

\[
\Delta_{ji} = \frac{(1-r) a_{ji}}{a_{ji} - b_{ji} (1-r)a_{ji} + b_{ji} (1-r) a_{ji} + (1 - r) \Delta_{i} + 1 - \Delta_{i}}.
\]

If the features have a power \( r \) or higher, the percentage error becomes

\[
P_{a} < P_{1} \frac{\prod_{j=0}^{n-1} (1 - \frac{r(1-r)}{(1-r) \Delta_{i} + 1 - \Delta_{i}})}{1 - \frac{r(1-r)}{(1-r) \Delta_{i} + 1 - \Delta_{i}}}.\]

C. Constant Power with Alternating Error Reduction

After extracting the first feature, if \( \Delta \) is not close to unity then the error in class A is generally reduced alternately. Using the design strategy that no new errors are to be generated and each feature is to have a constant power \( r \), then with \( a_{ji} < b_{ji} \) the error in class A after \( j + 1 \) features is \( (1-r)a_{ji} \). If \( a_{ji} > b_{ji} \) then the error in class B after \( j + 2 \) features is \( (1-r)b_{ji} \). Since no new errors are being generated, then

\[
\Delta_{ji+2} = \frac{a_{ji+2}}{a_{ji+2} + b_{ji+2}} = \frac{(1-r)a_{ji}}{(1-r)a_{ji} + (1-r)b_{ji} \Delta_{i}} \tag{9}
\]

and similarly \( \Delta_{ji+1} = \Delta_{ji+2} \). So, if \( a_{ji} < b_{ji} \), then

\[
\Delta_{i} = \frac{a_{ji}}{a_{ji} + b_{ji}}.
\]

If every feature designed has a power of \( r \) or higher, then from (8) the percentage error becomes

\[
P_{a} \leq \frac{P_{1}(1- \Delta_{i}r)^{n}(1- \Delta_{i}r)^{n-1}}{1 - \Delta_{i}r}, \quad n = 2m + 1
\]

Equation (10) is also an upper bound for features that have a constant power \( r \) and which generate new errors. If \( x_{ji} \) is a feature with power \( r \) that corrects the errors in class B and generates new errors in class A, then \( \Delta_{i}r > \Delta_{i} \) and hence \( P_{1}(1- \Delta_{i}r) > P_{1}(1- \Delta_{i}) \).

So a feature that generates new errors reduces the percentage error in the next stage. Thus (10) is an upper bound for features with a power \( r \) or more.

It is observed in (8) that with a constant \( r \), \( P_{a} \) will be a minimum if \( \Delta_{i}r \) is equal to unity at each stage. Thus a lower bound for \( P_{a} \) is obtained,

\[
P_{a} \geq P_{1}(1- \Delta_{i}r)^{n}.\]

The upper and lower bounds for \( \Delta_{i} = 0.7 \) and several values of \( r \) are plotted in Fig. 5.

IV. EXPERIMENTAL RESULTS

To illustrate these concepts an experiment with handwritten characters was performed. The classifier was to determine whether a character was a 9 or not. Class A consisted of 100 samples of handwritten 9's. Class B consisted of 100 samples of a mixture of the other 9 numerals and the symbols +, −, /, %. Each character was represented by a 12 × 16 binary array.

As the samples in class B differ among themselves, it is difficult to find a feature which is present in all the samples of class B and absent in the misclassified samples of class A, or vice versa. This makes it difficult to remove the errors in class A. For this reason an additional practical restriction was imposed on the features, namely, that the error in class A should be kept as close to 0 as possible.

The first feature was extracted to classify the two classes as well as possible. The feature \( x_{i} \) that did this best resulted in one error in class A and 55 errors in class B. Then an attempt was made to remove all the errors in class B which would result in a percentage error of less than 1 percent. To design features with a constant power \( r \) the curves of Fig. 6 were plotted using (9). In this experiment, \( P_{1}/P_{1}r = 0.037 \), to achieve the design goals with four features, a power of 0.75 is required for each feature. However, it seems impractical to obtain features with a power of 0.75 or better as the first feature had a power of only 0.73. A more reasonable approach would appear to be to design five features, each with a power of 0.63 or better, to obtain a percentage error of 1 percent. The number of samples to be corrected at each stage with a power of 0.63 is shown in Table I.

If the number of features to be designed is too small, then at some stage \( j \), it might become difficult to extract features with a given
power. In this case, the feature $x_i$ may become very complex or, in fact, impractical to design. So the design process may have to be restarted by changing the design objective if the power of each feature is required to be unreasonably high.

If the features are designed using a constant fractional reduction approach, then the value of $k$ required to reduce the percentage error to less than 1 percent with five features can be computed from (8) with $n = 5, P_n = 0.01, P_1 = 0.27$. The value of $k$ obtained is 0.57. Using this value of the fractional reduction, the value of $r$ for the given $\Delta$ is computed at each stage. From this, the minimum number of samples to be corrected to achieve the required $k$ is also computed.

The requirements for a constant $k$ at each stage and the results of the experiment are given in Table I. The percentage error at each step in the feature design is shown in Fig. 7.

The five features extracted by following the guidelines of the scheme were simple strokes and are shown in Fig. 8. The effectiveness of these features was tested on two test sets. The first consisted of 100 samples of 9's and 100 samples of the other numerals and symbols. Only one 9 was misclassified and none of the other samples were classified as being a 9. The second set consisted of 200 handwritten letters of the English alphabet. None of these characters was classified as a 9. This shows that the features extracted were sufficient to distinguish the 9's from other characters.

From Fig. 7 the constant $r$ bound seems to be better in the earlier
stages, whereas the constant $k$ bound is tighter in the latter stages. The constant $k$ bound is simpler, but the constant $r$ case gives the required power of a feature at the outset of the design process which may provide valuable insight as to the difficulty of the feature design. The required number of samples to be corrected appear to be nearly the same for both cases. However, in the constant $r$ case, the requirements were known before the experiment was performed, whereas in the constant $k$ case the requirements are known only one stage at a time.

V. CONCLUSIONS

The feature extraction scheme presented in this correspondence seems to be useful and practical in a two class problem in character recognition. The decision is developed concurrently with each feature and it is linear and near optimal. The theoretical error bounds developed in Section III appear to be reasonably tight as illustrated by the experiment in Section IV.

REFERENCES