Application of the Karhunen–Loève Expansion to Feature Selection and Ordering

KEINOSUKE FUKUNAGA, MEMBER, IEEE, AND WARREN L. G. KOONTZ, MEMBER, IEEE

Abstract—The Karhunen–Loève expansion has been used previously to extract important features for representing samples taken from a given distribution. A method is developed herein to use the Karhunen–Loève expansion to extract features relevant to classification of a sample taken from one of two pattern classes. Numerical examples are presented to illustrate the technique.

Also, it is shown that the properties of the proposed technique can be applied to unsupervised clustering, where given samples are classified into two classes without a priori knowledge of the class.

Index Terms—Clustering, feature extraction, feature selection, Karhunen–Loève expansion, pattern recognition, unsupervised learning.

INTRODUCTION

THE Karhunen–Loève expansion is a well known technique for representing a sample function of a random process [1]–[6]. It has been shown to be optimal in that the mean-square error committed by approximating the infinite series with a finite number of terms is minimized [1], [2]. Thus the Karhunen–Loève expansion extracts a set of features that is optimal with respect to representing a pattern class whose observable is a random process.

Watanabe [1], [2] discusses application of the Karhunen–Loève expansion to the representation of a pattern class. Chien and Fu [3] derive a necessary condition under which more random processes than one class can be represented by a single expansion, assuming zero-means for all processes. They present further discussion and a character recognition example in [4].

In pattern recognition, however, we wish to extract features which represent the difference between one pattern class and another. These features do not necessarily coincide with the important features to represent the pattern classes. In particular, two classes may have principal features which are similar. Besides, different sets of features for different classes add further difficulty. Several approaches to extracting features for recognition appear in the literature. Kadota and Shepp [7] and Kullback [8] give results for discriminating between two zero-mean Gaussian random processes. Their approach is to find the linear transformation which maximizes divergence. Fukunaga and Krile [9] find the transformation which simultaneously diagonalizes the covariance matrices in a two-class problem. Tou and Heyden proposed an iteration procedure to find the transformation to maximize the divergence in the reduced dimension [10].

It is not our purpose here to derive optimal features for recognition via the Karhunen–Loève expansion. We will show, however, that if we transform our coordinate system before applying the expansion, we may extract "good" features for recognition. Thus we will in one way relate the pattern representation problem to the pattern recognition problem.

An important byproduct of this investigation is the development of an algorithm for "unsupervised" clustering. Here, a set of samples is dichotomized using only the a priori probabilities of the two classes assumed to exist.

KARHUNEN–LOÈVE EXPANSION FOR TWO-CLASS PROBLEMS

Let us, first of all, briefly introduce the Karhunen–Loève expansion.

A set of time functions, \( f_i(t) \) \( (i = 1, 2, \ldots, N) \), can be expanded as a linear combination of basis functions, \( \phi_j(t) \), \( (j = 1, 2, \ldots) \), as follows:

\[
f_i(t) = \sum_{j=1}^{\infty} \alpha_{ij} \phi_j(t).
\]

(1)

The basis functions are obtained by solving the integral equation

\[
\lambda_j \phi_j(t) = \int_{-\infty}^{\infty} R(t, \tau) \phi_j(\tau) d\tau
\]

(2)

where \( R(t, \tau) \) is the autocorrelation function of the \( f(t) \)'s, and is given by

\[
R(t, \tau) = E[f_i(t)f_i(\tau)],
\]

(3)

where \( E[\cdot] \) denotes expectation over the \( N \) samples.

Suppose we measure and use only \( n \) discrete sampling points to express each time function, instead of the original continuous function. Then the functions may be represented as column vectors

\[
F_i = \begin{bmatrix} f_i(t_1) \\ f_i(t_2) \\ \vdots \\ f_i(t_n) \end{bmatrix} \quad (i = 1, 2, \ldots, N)
\]

(4)

Manuscript received March 10, 1969; revised August 12, 1969.

This work was supported in part by the Joint Services Program under ARPA Contract N00014-67-A-0226 Modified AE, and the Naval Ship Systems Command under Contract N000-68-C-1193.

The authors are with the School of Electrical Engineering, Purdue University, Lafayette, Ind. 47907.
Equation (1) becomes a finite sum
\[ F_i = \sum_{j=1}^{n} \alpha_{ij} \Phi_j \] 
(5)
where \( \Phi_j \) is also a vector representation of the \( j \)th basis function as
\[ \Phi_j = \left[ \begin{array}{c} \Phi_j(t_1) \\ \Phi_j(t_2) \\ \vdots \\ \Phi_j(t_n) \end{array} \right] \quad (j = 1, 2, \ldots, n). \]
(6)

Also, (2) and (3) are replaced by
\[ \lambda_j \Phi_j = S \Phi_j, \]
(7)
and
\[ S = E[F_i F_i^T], \]
(8)
where \( F_i^T \) is the transposed vector of \( F_i \). Equations (7) and (8) show that \( S \) is an autocorrelation matrix, and \( \lambda_j \) and \( \Phi_j \) are the \( j \)th eigenvalue and eigenvector of \( S \).

The number of sampling points, \( n \), should be determined independently so as to retain enough information to reconstruct the original time functions.

Thus, when two sets of time functions, \( F_{i1} \) and \( F_{i2} \), are expanded by two sets of basis functions, \( \Phi_{j1} \) and \( \Phi_{j2} \), \( F_{i1} \) and \( F_{i2} \) are expressed by
\[ F_{ik} = \sum_{j=1}^{n} \alpha_{ij}^{(k)} \Phi_j^{(k)} \quad (k = 1, 2; i = 1, 2, \ldots, N_k) \]
(9)
where it is assumed that the number of sampling points is \( n \) for both class 1 and class 2. We will use \( \omega_1 \) and \( \omega_2 \) in this paper to represent class 1 and class 2. Since the basis functions are the eigenvectors of the autocorrelation matrix as shown in (7), they are mutually orthonormal, i.e.,
\[ \Phi_j^{(k)} \Phi_i^{(k)} = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j. \end{cases} \]
(10)

Thus, the coefficients of the expression \( \alpha_{ij}^{(k)} \) can be obtained by
\[ \alpha_{ij}^{(k)} = \Phi_j^{(k)} F_i^{(k)}, \]
(11)
or in terms of matrix notation,
\[ A_{ik}^{(k)} = B^T F_i^{(k)}, \]
(12)
where
\[ A_{ik}^{(k)} = \begin{bmatrix} \alpha_{i1}^{(k)} \\ \alpha_{i2}^{(k)} \\ \vdots \\ \alpha_{in}^{(k)} \end{bmatrix} \quad \text{and} \quad B = [\Phi_1 \Phi_2 \cdots \Phi_n]. \]
(13)

When \( n \) basis functions are used, the \( F_i^{(k)} \)'s are expanded without error. However, if we select fewer, say \( n_k \) basis functions, the expansion becomes an approximation of the \( F_i^{(k)} \)'s. The mean-square error is given by
\[ \hat{e}^2 = \sum_{i=1}^{n} \left( \sum_{j \in J_k} \alpha_{ij}^{(k)} \Phi_j^{(k)} \right)^T \left( F_i^{(k)} - \sum_{j \in J_k} \alpha_{ij}^{(k)} \Phi_j^{(k)} \right) \cdot \left( F_i^{(k)} - \sum_{j \in J_k} \alpha_{ij}^{(k)} \Phi_j^{(k)} \right) \] \( P(\omega_k) \)
(14)
where \( P(\omega_k) \) is the a priori probability of \( \omega_k \), and \( J_k \) is the set of \( n_k \)'s such that \( \Phi_j^{(k)} \) is used in the expansion. Using (10) and (11), we see that (14) becomes
\[ \hat{e}^2 = \sum_{i=1}^{n} \sum_{j \in J_k} \Phi_j^{(k)} S_k \Phi_j^{(k)} \]
(15)
where \( S_k \) is the autocorrelation matrix multiplied by the a priori probability, i.e.,
\[ S_k = P(\omega_k) E[F_i^{(k)} F_i^{(k)}]^T \quad (k = 1, 2). \]
(16)
The mean-square error given by (15) can be minimized when the \( \Phi_{j1}^{(1)} \)'s and \( \Phi_{j2}^{(2)} \)'s are the eigenvectors of \( S_1 \) and \( S_2 \), respectively, and the minimized \( \hat{e}^2 \) can be expressed by the linear summation of eigenvalues whose eigenvectors are not selected to approximate \( F_i^{(k)} \)'s. Hence
\[ \hat{e}^2 = \sum_{j \in J_1} \lambda_j^{(1)} + \sum_{j \in J_2} \lambda_j^{(2)}. \]
(17)
The \( \lambda_j^{(1)} \)'s and \( \lambda_j^{(2)} \)'s are the eigenvalues of \( S_1 \) and \( S_2 \), respectively.

Thus, the basis functions can be selected in order of eigenvalues until \( \hat{e}^2 \) of (17) becomes smaller than some specified value.

From the above discussion we see that the Karhunen-Loève expansion is a powerful technique to represent or estimate a set of time functions. In particular, the mechanism of extracting features from the set of time functions suggests that we can apply the technique to pattern recognition. However, one of the big disadvantages of the technique, as far as pattern recognition is concerned, is that the technique does not necessarily select important features for separating pattern classes. When two pattern classes share similar important features, the corresponding eigenvalues are “large” and dominant. But these eigenvectors or basis functions are not important for separating the pattern classes. Therefore, a modification is needed in the Karhunen-Loève expansion in order to apply the expansion to pattern recognition and this problem will be discussed in the next section.

**Normalization Process**

In this section, a kind of normalization process is proposed in order to extract the important features for separating two pattern classes. The notation used in what follows is in accordance with Table I.

Let us assume that the autocorrelation matrices of two pattern classes are \( R_1 \) and \( R_2 \) premultiplied by a
priori probabilities as defined in (16), and the autocorrelation matrix of the mixture of both classes, \( R_0 \), is

\[
R_0 = R_1 + R_2. \tag{18}
\]

Since \( R_0 \) is a symmetric matrix, there exists a transformation matrix, \( P \), such that

\[
P R_0 P^T = I. \tag{19}
\]

Thus, the transformed autocorrelation matrices \( S_1 = P R_1 P^T \) and \( S_2 = P R_2 P^T \) satisfy

\[
S_1 + S_2 = I. \tag{20}
\]

After the above transformation, the two sets of eigenvalues and eigenvectors of the two pattern classes are related as follows. The eigenvalues and eigenvectors of class 1 are given by

\[
S_1 \Phi_j^{(1)} = \lambda_j^{(1)} \Phi_j^{(1)}, \quad (j = 1, 2, \cdots, n). \tag{21}
\]

Assume that the eigenvalues are in ascending order, i.e.,

\[
\lambda_1^{(1)} \geq \lambda_2^{(1)} \geq \cdots \geq \lambda_n^{(1)}. \tag{22}
\]

Then, the eigenvalues and eigenvectors for the second class are

\[
S_2 \Phi_j^{(2)} = (I - S_1) \Phi_j^{(2)} = \lambda_j^{(2)} \Phi_j^{(2)}, \tag{23}
\]

or

\[
S_1 \Phi_j^{(2)} = (1 - \lambda_j^{(2)}) \Phi_j^{(2)}. \tag{24}
\]

From (21) and (24),

\[
\Phi_j^{(2)} = \Phi_j^{(1)} \tag{25}
\]

and

\[
\lambda_j^{(1)} = 1 - \lambda_j^{(2)}. \tag{26}
\]

Equations (25) and (26) show that both pattern classes have the same set of eigenvectors, and the corresponding eigenvalues are reversely ordered. That is, the eigenvector with the largest eigenvalue for class 1 has the smallest eigenvalue for class 2 and so on. On the other hand, from (17), the physical meaning of an eigenvalue is the mean-square error when the corresponding eigenvector is eliminated. Therefore, all eigenvalues for both classes should be positive. This fact derives the following relationship for eigenvalues:

\[
1 \geq \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0 \quad \text{for } \omega_1 \tag{27}
\]

\[
0 \leq 1 - \lambda_1 \leq 1 - \lambda_2 \leq \cdots \leq 1 - \lambda_n \leq 1 \quad \text{for } \omega_2. \tag{28}
\]

Here the superscripts which were used to indicate the class are eliminated to yield a simpler expression.

Thus, after the normalizing transformation, the important features or effective basis functions for class 1 are the least important features for class 2, and the important features for class 2 are the least important for class 1. Both classes cannot share common important features. Therefore, the selection of the basis functions can be performed by taking \( \Phi_1^{(1)}, \Phi_2^{(1)}, \cdots \) for class 1 and \( \Phi_n^{(1)}, \Phi_{n-1}^{(1)}, \cdots \) for class 2.

Unfortunately this technique does not extend to the general multiclass problem. However, the multiclass pattern classifier may be realized as a sequence of pairwise comparisons of likelihood functions. In this case each pair may be examined using the expansion described above.

### Table I

<table>
<thead>
<tr>
<th>Sample Vectors ( k = 1, \cdots, N ); ( \bar{k} = 1, 2 )</th>
<th>( X_i^{(k)} )</th>
<th>( F_i^{(k)} = PX_i^{(k)} )</th>
<th>( F_i^{(k)} )</th>
<th>( A_i^{(k)} = B^T F_i^{(k)} )</th>
<th>( A_i^{(k)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Vectors ( k = 1, 2 )</td>
<td>( M_\bar{k} )</td>
<td>( D_\bar{k} = P M_\bar{k} )</td>
<td>( D_\bar{k} )</td>
<td>( L_\bar{k} = B^T D_\bar{k} )</td>
<td>( L_\bar{k} )</td>
</tr>
<tr>
<td>Autocorrelation Matrices of (16) ( k = 0, 1, 2 )</td>
<td>( R_k )</td>
<td>( S_k = P R_k P^T )</td>
<td>( S_k )</td>
<td>( Q_k = B^T S_k B )</td>
<td>( Q_k )</td>
</tr>
<tr>
<td>Covariance Matrices ( k = 1, 2 )</td>
<td>( \Sigma_k )</td>
<td>( K_k = P \Sigma_k P^T )</td>
<td>( K_k )</td>
<td>( \Theta_k = B^T K_k B )</td>
<td>( \Theta_k )</td>
</tr>
</tbody>
</table>
where \( D_0 \) and \( D \) are the mean vector of the mixture and the vector from \( D_0 \) to the mean vector of the class 2, i.e.,

\[
\begin{align*}
D_1 &= D_0 \quad (33) \\
D_2 &= D_0 + D \quad (34)
\end{align*}
\]

or

\[
\begin{align*}
D_0 &= 1/2(D_2 + D_1) \quad (35) \\
D &= 1/2(D_2 - D_1). \quad (36)
\end{align*}
\]

The eigenvalues and eigenvectors of \( S_1 \) follow from

\[
S_1 \Phi_j = 1/2 \{ (I - (D_0 D^T + DD_0^T)) \Phi_j = \lambda_j \Phi_j \} \quad (37)
\]

or

\[
(D_0 D^T + DD_0^T) \Phi_j = (1 - 2\lambda_j) \Phi_j. \quad (38)
\]

Therefore, \( (D_0 D^T + DD_0^T) \) has the same eigenvalues as \( S_1 \) and its eigenvalues \( \mu_i \) are related with \( \lambda_i \) as follows:

\[
\lambda_i = 0.5 - 0.5\mu_i. \quad (39)
\]

Since the rank of the matrix \( (D_0 D^T + DD_0^T) \) is 2, \( n - 2 \) eigenvalues are zero, and the other eigenvalues can be easily obtained by solving the following equations:

\[
\begin{align*}
\mu_1 + \mu_2 + \cdots + \mu_{n-1} &= 0 \\
\mu_1 + \mu_2 &= -2d_0^2 \cos \theta \\
\mu_1^2 + \mu_2^2 &= 2d_0^2 d_1^2 (1 + \cos^2 \theta)
\end{align*}
\]

where

\[
\begin{align*}
d_0 &= \sqrt{D_0^2 - D_0^2}, \quad \text{(length of the vector \( D_0 \))} \\
d &= \sqrt{D^2 - D_0^2}, \quad \text{(length of the vector \( D \))}
\end{align*}
\]

and

\[
\cos \theta = D^T D_0/d_0d_1 \quad \text{(angle between \( D_0 \) and \( D \)).} \quad (45)
\]

The solution of (41) and (42) is

\[
\begin{align*}
\mu_1 &= \frac{d_0d_1}{d_0d_1} (\cos \theta - 1) \leq 0 \\
\mu_2 &= \frac{d_0d_1}{d_0d_1} (\cos \theta + 1) \geq 0.
\end{align*}
\]

Thus when \( K_1 = K_2 \) and \( P(\omega_1) = P(\omega_2) = 0.5 \), \( (D_0 D^T + DD_0^T) \) has \( n - 2 \) zero eigenvalues, \( \mu_1, \cdots, \mu_{n-1} \), and one positive and one negative eigenvalue, \( \mu_2 \) and \( \mu_1 \). From (39) \( S_1 \) has the following eigenvalues:

\[
\begin{align*}
\lambda_1 &= 0.5 + 0.5d_0d_1 (1 - \cos \theta) \\
\lambda_2 &= \lambda_3 = \cdots = \lambda_{n-1} = 0.5 \\
\lambda_n &= 0.5 - 0.5d_0d_1 (1 + \cos \theta).
\end{align*}
\]

Equations (48), (49), and (50) suggest that we select \( \Phi_1 \) for class 1 whose eigenvalue is between 0.5 and 1, and \( \Phi_n \) for class 2 whose eigenvalue is between 0.5 and 0. Since \( \lambda_2, \lambda_3, \cdots, \lambda_{n-1} \) are not zero but 0.5, \( \Phi_2, \cdots, \), \( \Phi_{n-1} \) contribute information for representing or estimating both class 1 and class 2. However, these eigenvalues do not contribute any information for separating two pattern classes. This fact will be discussed in the next section.

IEEE TRANSACTIONS ON COMPUTERS, APRIL 1970

CLASSIFICATION

In the previous section, it was shown that after transforming the coordinate system, we can select the common set of eigenvectors for both class 1 and 2. We now select a subset of the eigenvectors to represent the sample and study the effectiveness of these eigenvectors in classification. We select those eigenvectors whose eigenvalues are closer to 0 or 1.

Intuitively, the above selection of eigenvectors is appealing to use, as was discussed in the preceding section. However, in order to give a more rigorous justification, we have to set up an index of performance which measures the separability of two pattern classes, and to observe the contribution of each eigenvector to the index. Although there are several indices of performance, let us take the divergence in this paper [11]. The divergence for two Gaussian distributions is given by

\[
\begin{align*}
\text{Div} &= 1/2(\xi_1 - \xi_2)^T(\Theta_1^{-1} + \Theta_2^{-1})(\xi_1 - \xi_2) \\
&+ 1/2 \text{tr}[\Theta_1 - \Theta_2(\Theta_1^{-1} - \Theta_2^{-1})]
\end{align*}
\]

(51)

where, as seen in Table I, \( L_4 \) and \( \Theta_k \) are the mean vector and covariance matrix of the coefficients of the Karhunen–Loève expansion. Equations (11) or (12) show that these coefficients, \( A \)'s, can be considered as a new set of variables which are transformed from \( F \)'s by the eigenmatrix \( B \) of (12) and (13). Since the divergence of (51) is invariant under the linear transformation of Table I, if we select \( n \) eigenvectors, the distributions of \( A^{(1)} \)'s and \( A^{(3)} \)'s have the same divergence as the original distribution of \( X^{(1)} \)'s and \( X^{(3)} \)'s.

Thus, the evaluation of eigenvectors for classification purposes becomes the evaluation of variables in the distributions of \( A^{(3)} \)'s.

For calculating the divergence for Gaussian distributions, \( L_4 \), \( Q_8 \), and \( \Theta_k \) of Table I can be obtained as follows:

\[
\begin{align*}
L_4 &= E[A^{(1)}] = B^T D_8 \\
Q_8 &= P(\omega_k)E[A^{(1)} A^{(1)T}] = B^T P(\omega_k)E[F^{(1)} F^{(1)T}] B \\
\Theta_k &= Q_8/P(\omega_k) = L_4 L_4^T
\end{align*}
\]

(52)

(53)

(54)

where \( \Lambda \) is a diagonal matrix with \( \lambda_1, \cdots, \lambda_n \) as its components.

From (53)

\[
P(\omega_k) E[\sigma_i^{(1)}] = \begin{cases} 
\lambda_i & \text{for } k = 1 \\
1 - \lambda_i & \text{for } k = 2
\end{cases}
\]

(55)

Equation (55) shows that selecting eigenvectors by \( \lambda_i \) means the selection of eigenvectors whose coefficients have larger differences between \( \omega_1 \) and \( \omega_2 \) in the mean-square values. In the following discussion, we will show theoretically that this selection is the best in terms of the divergence when two covariance matrices are equal. For unequal covariance cases we will show by two ex-
Equal Covariance Cases

Let us assume \( K_1 = K_2 \) and \( P(\omega_1) = P(\omega_2) = 0.5 \). Since the \( \Phi_j \)'s are selected as the eigenvectors of \( (D_0D^T + DD_0^T) \)
as is shown in (38), and since \( \mu_i = \cdots = \mu_{n-1} = 0 \) from (40), \( \Phi_2, \cdots, \Phi_{n-1} \) should satisfy

\[
(D_0D^T + DD_0^T)\Phi_j = 0 \quad (j = 2, \cdots, n - 1)
\]

where \( \mathbf{0} \) is a column vector with all zero components.

Or,

\[
D_0(D^T\Phi_j) = -D(D_0^T\Phi_j). \tag{57}
\]

It is easily shown that (57) holds for finite and different vectors \( D_0 \) and \( D \) if and only if

\[
D^T\Phi_j = \Phi_j^T D = 0 \quad \text{and} \quad D_j^T\Phi_j = \Phi_j^T D_0 = 0 \quad (j = 2, \cdots, n - 1). \tag{58}
\]

That is, in this case, \( n - 2 \) eigenvectors are orthogonal to the plane of \( D \) and \( D_0 \).

From (33), (34), (52), and (58), the components of \( L_k, L_k^{(0)} \) are

\[
l_j^{(0)} = \Phi_j^T D_0 = \Phi_j^T (D_0 \mp D) = 0 \\
\quad \text{for } j = 2, \cdots, n - 1 \tag{59}
\]

\[
\quad \text{or} \\
\lambda_k = \begin{bmatrix}
l_1^{(0)} \\
0 \\
\vdots \\
0
\end{bmatrix}. \tag{60}
\]

The covariance matrix, \( \Theta = \Theta_1 = \Theta_2 \), can be calculated from (53) and (54) with \( P(\omega_1) = P(\omega_2) = 0.5 \) and \( \lambda_2 = \lambda_3 = \cdots = \lambda_{n-1} = 0.5 \) of (49) as follows:

\[
\Theta = 2\lambda - L_1L_1^T = 2(I - \Lambda) - L_2L_2^T
\]

\[
= \begin{bmatrix}
\theta_{11} & 0 & \cdots & 0 \\
0 & \theta_{22} & & \\
\vdots & \ddots & \ddots & \\
0 & \cdots & 0 & \theta_{nn}
\end{bmatrix} \tag{61}
\]

where

\[
\theta_{ii} = 2\lambda_i - l_i^{(1)^2} \quad (i = 1, n) \\
\theta_{ij} = \theta_{ji} = -l_i^{(1)} l_j^{(1)} \quad (i, j = 1, n). \tag{62}
\]

Thus, the divergence for equal covariance cases is

\[
\text{Div} = \frac{1}{2} \left[ \theta_{11}(l_1^{(2)} - l_1^{(1)^2}) + \theta_{1n}(l_2^{(2)} - l_1^{(1)^2}) \right.
\]

\[\left. - 2\theta_{1n}(l_2^{(2)} - l_1^{(1)^2})(l_n^{(2)} - l_1^{(1)^2}) \right]. \tag{63}
\]

Equation (63) shows that the divergence for \( n \) variables can be calculated by using only \( \lambda_1, \lambda_n, \Phi_1^T D_0, \) and \( \Phi_n^T D_n \) (\( 1 \geq \lambda_1 \geq 0.5 \) and \( 0.5 \geq \lambda_n \geq 0 \), and there is no contribution from \( \Phi_j \) \( (j = 2, 3, \cdots, n - 1) \) \( \lambda_2 = \lambda_3 = \cdots = \lambda_{n-1} = 0.5 \)). Thus, by selecting only two eigenvectors \( \Phi_1 \) and \( \Phi_n \), the two dimensional distributions of \( \alpha \)'s retains the same divergence as the \( n \)-dimensional distributions of \( \alpha \)'s. On the other hand, sufficient statistics tells that two eigenvectors are essentially necessary for two class problems with different non-zero mean vectors. Therefore, the procedure of this paper gives the best selection of eigenvectors for equal covariance cases.

Unequal Covariance Case

The unequal covariance case is difficult to discuss theoretically. Therefore, the properties of our feature selection and ordering procedure have been studied via numerical examples. We show that those coefficients which are important for representation after normalization (i.e., those with eigenvalues near 0 or 1) all are also important for classification. The divergence of the coefficients is used as the measure of effectiveness for classification.

Experiment 1: Marrill and Green give the mean vectors and covariance matrices for eight dimensional samples measured from the characters \( A \) and \( B \) \([11]\).

We applied the procedure outlined above to these data. Table II lists the eigenvalues, which are ordered by \( |\lambda - 0.5| \) from the largest at the top to the smallest at the bottom. Also, Table II shows the divergence for the coefficient of individual eigenvector, \( \text{Div}_j \). The order of \( \text{Div}_j \) is the same as the order of \( |\lambda - 0.5| \), except one exchange of \( j = 2 \) and 3. However, it should be noticed that \( |\lambda_3 - 0.5| \) and \( |\lambda_2 - 0.5| \), and also \( \text{Div}_2 \) and \( \text{Div}_3 \) are very close.

When a set of eigenvectors is selected, the divergence of the set includes the correlation terms between the


| Selection Order (j) | Eigenvalue ($\lambda_j$) | $|\lambda_j - 0.5|$ | Div$_{ij}$ |
|---------------------|--------------------------|----------------------|-----------|
| 1                   | 0.9748                   | 0.4748               | 18.48     |
| 2                   | 0.0443                   | 0.4670               | 10.13     |
| 3                   | 0.0759                   | 0.4241               | 5.146     |
| 4                   | 0.9129                   | 0.4129               | 4.337     |
| 5                   | 0.8244                   | 0.3244               | 1.453     |
| 6                   | 0.1815                   | 0.3185               | 1.365     |
| 7                   | 0.8131                   | 0.3131               | 1.290     |
| 8                   | 0.1871                   | 0.3129               | 1.288     |
| 9                   | 0.2076                   | 0.2824               | 1.039     |
| 10                  | 0.7683                   | 0.2683               | 0.8130    |
| 11                  | 0.2401                   | 0.2599               | 0.7405    |
| 12                  | 0.2683                   | 0.2317               | 0.5469    |
| 13                  | 0.7294                   | 0.2294               | 0.5332    |
| 14                  | 0.2922                   | 0.2078               | 0.4175    |
| 15                  | 0.6981                   | 0.1981               | 0.3724    |
| 16                  | 0.6632                   | 0.1632               | 0.2385    |
| 17                  | 0.6619                   | 0.1619               | 0.2342    |
| 18                  | 0.3569                   | 0.1431               | 0.1786    |
| 19                  | 0.6024                   | 0.1024               | 0.0876    |
| 20                  | 0.4051                   | 0.0949               | 0.0748    |
| 21                  | 0.4263                   | 0.0737               | 0.0444    |
| 22                  | 0.5657                   | 0.0657               | 0.0352    |
| 23                  | 0.4356                   | 0.0644               | 0.0337    |
| 24                  | 0.5154                   | 0.0514               | 0.0020    |
| 25                  | 0.4849                   | 0.0515               | 0.0019    |
| 26                  | 0.5040                   | 0.0004               | 0.0002    |

Coefficients in addition to the individual divergences. However, the individual divergence still gives a measure for effectiveness of each eigenvector.

**Experiment 2**: In this example, the mean vectors and the covariance matrices are calculated from 26 dimensional data supplied by Du Pont Corporation, Wilmington, Delaware. These data are measurements made on certain fabrics.

Table III shows the list of $\lambda_j$, $|\lambda_j - 0.5|$ and Div$_{ij}$. In this example, the ordering of $|\lambda_j - 0.5|$ is exactly the same as the ordering by individual divergences.

**Unsupervised Clustering**

We now show how the properties discussed in the preceding sections can be used to develop an algorithm for classifying samples given very limited a priori knowledge.

Suppose we have $N$ samples which we know to come from two pattern classes. Further, assume that the a priori probabilities $P(\omega_1)$ and $P(\omega_2)$ are known. Transform coordinates so that the mixture autocorrelation matrix is $I$. Then, for all dichotomies such that $NP(\omega_1)$ samples are assigned to $\omega_1$ and $NP(\omega_2)$ to $\omega_2$, (20) is satisfied. Thus, for all of these dichotomies, both classes share common eigenvectors and have reversely order eigenvalues in the range 0 to 1.

Now our aim is to choose the best dichotomy according to some criterion. Given the criterion, $C$, a simple algorithm for clustering is as follows.

1) Assign $NP(\omega_1)$ samples to $\omega_1$ and $NP(\omega_2)$ to $\omega_2$ arbitrarily.
2) Exchange equal numbers of samples between $\omega_1$ and $\omega_2$ so as to increase $C$.
3) Repeat until $C$ is maximized.

In what follows we assume $P(\omega_1) = P(\omega_2) = 0.5$.

**A Criterion $C$**

One property of a "good" dichotomy is that each feature should be effective for classification. According to what we have shown, this means that the eigenvalue of each feature should be near 0 or 1. Thus, our criterion function should increase as each eigenvalue approaches 0 or 1. A proposed criterion function is

$$C = \frac{1}{n} \sum_{j=1}^{n} (\lambda_j - 0.5)^2 + \frac{1}{n} \sum_{j=1}^{n} (1 - \lambda_j - 0.5)^2$$

$$\frac{2}{n} \left( \sum_{j=1}^{n} \lambda_j^2 - \sum_{j=1}^{n} \lambda_j \right) + 0.5.$$  (64)

That is, the eigenvalues which are far from 0.5 are more appreciated.

Equation (64) can be rewritten in terms of the elements of $S_1, s_{ij}$, rather than the eigenvalues, as follows:

$$C = \frac{2}{n} \left( \text{tr} S_1^2 - \text{tr} S_1 \right) + 0.5$$

$$\frac{2}{n} \left( \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij}^2 - \sum_{i=1}^{n} s_{ii} \right) + 0.5.$$  (65)

If a sample $F_i$ in class 1 is exchanged for a sample $F_k$ in class 2, then the change in $C$, $\Delta C$, is given by

$$\Delta C = \frac{2}{n} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} (2s_{ij} \Delta s_{ij} + \Delta s_{ii}) - \sum_{i=1}^{n} \Delta s_{ii} \right\}$$

(66)

where

$$\Delta s_{ij} = \frac{1}{N} \left\{ f_k(t_i)f_k(t_j) - f_i(t_i)f_i(t_j) \right\},$$

(67)

and where $f_k(t_i)$ is, from (4), the $i$th component of $F_k$. When $N$ is large, the $\Delta s_{ii}$ term of (66) can be neglected such that (66) becomes

$$\Delta C = \Delta C_k + \Delta C_l$$

(68)

where

$$\Delta C_k = \frac{2}{nN} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} 2s_{ij} f_k(t_i)f_k(t_j) - \sum_{i=1}^{n} f_k(t_i)^2 \right\}$$

(69)

and

$$\Delta C_l = - \frac{2}{nN} \left[ 2F_k^T S_1 F_k - F_k^T F_k \right].$$

(70)

It should be noted that neither (66) nor (68) requires eigenvalue calculations and autocorrelation calculations for sample exchanges.
A Sample Exchange Algorithm

A sample exchange algorithm proposed in this paper is based on the assumption that we have a large enough number of samples to justify (68). Also, $P(\omega_1) = P(\omega_2) = 0.5$ is assumed. The flow chart of the actual algorithm is shown in Fig. 1. As is seen in Fig. 1, for a given class assignment, $S_i$ is calculated, and $\Delta C_{jk}'s$ for all $F_k's$ of class 2 and $\Delta C_j's$ for all $F_i's$ of class 1 are calculated and ordered as

$$\Delta C_{i1} \geq \Delta C_{i2} \geq \cdots \geq \Delta C_{iN/2}$$

(71)

$$\Delta C_{2i} \geq \Delta C_{22} \geq \cdots \geq \Delta C_{2N/2}$$

(72)

Then

$$\Delta C_1 \geq \Delta C_2 \geq \cdots \geq \Delta C_{N/2}$$

(73)

where

$$\Delta C_i = \Delta C_{ki} + \Delta C_{ti},$$

(74)

Exchanges are made for all pairs of $F_k's$ and $F_i's$ which give $\Delta C_i > 0$, since these exchanges increase $C$. Then $S_i$ is recalculated for the new class assignment after the above exchanges, and the same process is repeated until we obtain a class assignment which gives

$$\Delta C_1 = \Delta C_{k1} + \Delta C_{t1} \leq 0.$$  (75)

Equation (75) guarantees that this final class assignment gives the maximum value of $C$. Let us select $m$ samples $F_i's$ ($i \in I = \{i_1, i_2, \ldots, i_m\}$) from class 1 and $m$ samples $F_j's$ ($j \in J = \{j_1, j_2, \ldots, j_m\}$) from class 2. When (75) holds with (71) and (72), the exchange of these $2m$ samples gives

$$\Delta C = \sum_{i \in I} \Delta C_{ki} + \sum_{j \in J} \Delta C_{ij} \leq m(\Delta C_{k1} + \Delta C_{t1}) \leq 0.$$  (76)

Since (76) holds for all possible $m, I$, and $J$, (75) guarantees the maximum $C$.

Example 1: Again, Marill and Green's data are used [11]. According to the mean vectors and the covariance matrices in an eight-dimensional space, 200 samples for each class are generated so as to have a Gaussian distribution. The 400 samples from class 1 and 2 are normalized and mixed, and an initial class assignment is given randomly. The procedure of this paper reclassifies these samples into two classes. The classification error is 11 samples out of 200 (5.5 percent). Since Marill and Green gave 2.75 percent error experimentally and Fukunaga and Krule gave 1.85 percent error theoretically for the Bayes quadratic optimum classifier, 5.5 percent error is a reasonable result [9]. Table IV shows the number of exchanged sample pairs (the number of cycles in loop 2 of Fig. 1) and $C$ versus the number of class reassignments (the number of cycles in loop 1 of Fig. 1). Also, Fig. 2 shows the distribution of only 50 samples out of 200 in the most significant two-dimensional subspace of the original eight-dimensional spaces.
The misclassified samples appear in the region between the two distributions. There is apparently an overlap between correct and incorrect classified samples because we show only two dimensions. These results justify the criterion and the sample exchange algorithm of this paper.

Example 2: Several other examples have been studied. As a result, one significant property of our unsupervised clustering algorithm has been observed, which is shown in the most simplified examples as follows.

Fig. 3 is the typical simplified example where four samples are distributed symmetrically in a two dimensional space. If we assign the class to each sample, the procedure of this paper picks up the most important eigenvector for classification purposes. However, if we compare $C$'s among all possible combinations of class assignments under the assumption of $P(\omega_1) = P(\omega_2) = 0.5$, the circled class assignments in Fig. 3 gives the largest $C$. Therefore, the unsupervised clustering algorithms which maximize $C$ separate these samples as shown in Fig. 3, although a human being cannot tell how to separate these samples. The above class assignments are reasonable, if we look into the physical meaning as follows.

1) In Fig. 3(a), $X_1$ and $X_4$ are the negative images of $X_1$ and $X_2$, respectively. For example, if $X_1$ is a black and white image of a character on a mesh, $X_4$ is the same image with black and white reversed. Or if $X_1$ represents a time function $x_1(t)$, then $x_4(t) = -x_1(t)$. Therefore, $X_1$ and $X_4$ should have the common feature which is different from the common feature of $X_2$ and $X_3$.

2) In Fig. 3(b), $X_3$ and $X_4$ are approximately expressed by $ax_1$ and $bx_2$ where $a$ and $b$ are some constants. That is, the difference between $X_1$ and $X_4$ or $X_2$ and $X_4$ is found in the amplitude of signals. On the other hand, the difference between $X_1$ and $X_2$ or $X_3$ and $X_4$ is found in the shape of the time function. Thus, the unsupervised clustering algorithm in this paper is in favor of signal shape rather than signal level as its feature. When signals are normalized in amplitude, that is, $X_i/\sqrt{X_i^T X_i}$ is used as the signal representation, it becomes obvious to a human being that $X_1$ and $X_4$ should be separated from $X_2$ and $X_3$.

Conclusions

This paper discusses the application of the Karhunen-Loève expansion to pattern recognition. The major results obtained are as follows.

1) By a transformation which converts the auto-correlation matrix of the mixture to a unit matrix, two classes never share the important basis vectors for classification purposes.

2) Thus, the selection of the eigenvectors can be carried by the eigenvalues.

3) This selection picks up two eigenvectors, keeping the same divergence as for the eigenvectors, when the covariance matrices are equal. For unequal covariance cases, two examples are shown to justify the procedure.

4) Using the above properties, an unsupervised clustering algorithm is proposed. Examples are shown how samples are separated into classes using only limited a priori knowledge.

References


