On Using Class-dependent Principle Component Analysis for Dissimilarity-Based Classifications

Sang-Woon Kim
Department of Computer Engineering, Myongji University, Yongin, 449-728 Korea

Abstract—The aim of this paper is to present an empirical evaluation on using class-dependent principle component analysis (PCA) for dissimilarity-based classifications (DBC) [18]. In DBC, the classification performance relies heavily on how well the dissimilarity space is constructed. In this paper, we study a way of constructing it in eigenspaces, spanned by the subset of principal eigenvectors, extracted from the training data set through the class-dependent PCA, instead of utilizing prototype selection methods and/or generalizing dissimilarity measures. In particular, we generate an eigenspace (i.e., a covariance matrix) per class, not for the entire data set, to compute distances in a vector space constructing a dissimilarity-based classifier. Our experimental results, obtained with well-known benchmark data and some UCI data sets, demonstrate that when the dimensionality of the eigenspaces has been appropriately selected, DBC, albeit not always, can be improved in terms of classification accuracies.

Keywords: statistical pattern recognition, dissimilarity-based classification, class-dependent principle component analysis

1. Introduction

Dissimilarity-based classifications (DBC) [18] are a way of defining classifiers among the classes, where the process is not based on feature measurements of individual objects (a set of features), but rather on a suitable dissimilarity measure among the individual objects (pairwise object comparisons). The advantage of this methodology is that since it does not operate on the class-conditional distributions, the problems associated with feature spaces, such as the curse of dimensionality and the small sample size problem, can be avoided [13]. Another salient advantage of such a paradigm is that it can exploit expert knowledge when measuring (dis)similarities between the pairwise objects [4].

The problem with this strategy, however, is that we need to measure the inter-pattern dissimilarities for all the training data to ensure there is no zero distance between objects of different classes. Consequently, the classes do not overlap, and therefore, the lower error bound is zero. The major issues we encountered when designing DBCs are summarized as follows: (1) How can prototype subsets be selected (or created) from the training samples [13], [18], [19]? (2) How can the dissimilarities between object samples be measured [17], [18]? (3) How can classifiers be designed in the dissimilarity space [12], [18]? and (4) How can the dissimilarity space be embedded in the pseudo-Euclidean space [4]?

Several strategies have been used to explore these issues. Specifically, with regard to solving the first problem, various methods have been proposed in the literature [19] as a means of selecting a representation subset of data that is both compact and capable of representing the entire data set. In these methods, a training set, T, is pruned to yield a subset of representative prototypes, P, where, without loss of generality, |P| ≤ |T|. However, it is difficult to find the optimal number of prototypes and, furthermore, selecting prototype stage may potentially lose useful information for discrimination. To avoid these problems, Riesen et al. [20] and Kim [11] prefer not to directly select the representative prototypes from the training samples; rather, they use a dimensionality reduction scheme after computing the dissimilarity matrix with the entire training samples.

On the other hand, subspace methods of pattern recognition [16] are a technique in which the object classes are not primarily defined as bounded regions in a feature space, but rather given in terms of eigenspaces defined by the basis vectors of the principal component analysis (PCA) [10]. However, unlike conventional PCA, an eigenspace can be generated per class, not for the entire data set, in order to maximize the discriminatory power of the subspace [21]. Because different classes may have different characteristics and hence the reliability of the estimated covariance matrices can be significantly different. However, as the objective of PCA is the best pattern reconstruction that may not be optimal for classification. In order to overcome this problem from the classification point of view, numerous strategies, including Probabilistic PCA [22], Rotational invariant l1-norm PCA [3], [14], and Asymmetric PCA [8], [9], have been proposed in the literature. In particular, in [8], Jiang addressed the problem of applying PCA on the asymmetric classes and/or the unbalanced data. From this perspective, in this paper the training data set is partitioned into several subsets (i.e., one per class ) and then performs PCA about each subset.

In DBC, on the other hand, the classification performance

This work was supported by the National Research Foundation of Korea funded by the Korean Government (2012R1A1A2041661). The author is very grateful to Prof. Dum from the Delft University of Technology for the instructive discussions we had in Benicassim and his valuable comments.
relies heavily on how well the dissimilarity space, which is determined by the dissimilarity matrix, is constructed. Thus, to improve the classification performance, more robust dissimilarity matrices should be constructed. To achieve this goal, the prototype subset should be selected [19] and/or the dissimilarity could be generalized [17]. From this point of view, in this paper we perform DBC in eigenspaces spanned by the principal eigenvectors, expecting that the noise and outlier could be excluded from the dissimilarity representation, without the need to select prototypes or generalize the dissimilarity. Especially, we generate an eigenspace (i.e., a covariance matrix) per class, not for the entire data set.

The major goal of this paper is to demonstrate that the classification performance of DBC can be improved by measuring the dissimilarity in eigenspaces constructed with class-dependent PCA, not class-independent PCA that is usually employed in PCA-based applications. This goal can be achieved by appropriately projecting the data set on multiple eigenspaces, one per class, and effectively measuring the dissimilarity between the projected points. Experimental results, obtained with a well-known benchmark data and some UCI data sets, demonstrate that when the dimensionality of the eigenspaces has been appropriately selected, the DBC can be improved in terms of classification accuracies.

The remainder of the paper is organized as follows: In Section 2, after providing a brief introduction to DBC, we continuously present an explanation of the class-dependent PCA and that of the DBC in multiple eigenspaces. In Section 3, we present the experimental results obtained with the benchmark image data and UCI real-life data sets. Finally, in Section 4, we present our concluding remarks as well as some feature works that deserve further study.

2. Related Work

In this section, we briefly review dissimilarity-based classifications and distance measures in eigenspaces that are closely related to the present empirical study. The details of these methodologies can be found in the related literature, including [18] and [21].

2.1 Dissimilarity Representation [18]

A dissimilarity representation of a set of object samples, \( T = \{ \mathbf{x}_i \}_{i=1}^n \in \mathbb{R}^d \), is based on pairwise comparisons and is expressed, for example, as an \( n \times m \) dissimilarity matrix, \( D_{T,P}[:,:], \) where \( P = \{ \mathbf{p}_j \}_{j=1}^m \in \mathbb{R}^d \), a prototype set, is extracted from \( T \), and the subscripts of \( D \) represent the set of elements on which the dissimilarities are evaluated. Thus, each entry, \( D_{T,p}[i,j] \), corresponds to the dissimilarity between the pairs of objects, \( \mathbf{x}_i \) and \( \mathbf{p}_j \), where \( \mathbf{x}_i \in T \) and \( \mathbf{p}_j \in P \). Consequently, when given a distance measure between two vectors, \( d(\cdot,\cdot) \), an object, \( \mathbf{x}_i \), is represented as a column (or a row) vector, \( \delta(\mathbf{x}_i,P) \), as follows:

\[
\delta(\mathbf{x}_i,P) = [d(\mathbf{x}_i,\mathbf{p}_1), \cdots, d(\mathbf{x}_i,\mathbf{p}_m)], \quad 1 \leq i \leq n. \tag{1}
\]

Here, the dissimilarity matrix, \( D_{T,P}[\cdot,\cdot] \), defines vectors in a dissimilarity space on which the \( d \)-dimensional object, \( \mathbf{x}_i \), given in the input-feature space, is represented as an \( m \)-dimensional vector, \( \delta(\mathbf{x}_i,P) \) or simply \( \delta(\mathbf{x}_i) \).

On the basis of what we explained briefly, an conventional algorithm for DBC is summarized in the following:

1. Select the prototype subset, \( P \), from the training set, \( T \), by using one of the prototype selection methods.
2. Using Eq. (1), compute the dissimilarity matrix, \( D_{T,p}[\cdot,\cdot] \), in which each dissimilarity is computed on the basis of a distance metric.
3. For a test sample, \( z \), compute a dissimilarity vector, \( \delta(z) \), by using the same measure used in Step 2.
4. Achieve the classification by invoking a classifier built in the dissimilarity space and operating it on the dissimilarity vector \( \delta(z) \).

In the above algorithm, it can be seen that the performance of DBC relies heavily on how well the dissimilarity space, which is determined by the dissimilarity matrix, is constructed. Thus, to improve the performance, we need to ensure that the dissimilarity matrix is well assembled.

2.2 Class-dependent PCA [21]

The data set, \( T \), can be decomposed into subsets, \( T_i \), \( (i = 1, \cdots, c) \), as follows: \( T = \bigcup_{i=1}^c T_i \), \( T_i = \{ \mathbf{x}_j \}_{j=1}^{n_i} \in \mathbb{R}^d \), with \( n = \sum_{i=1}^c n_i \), \( T_i \cap T_j = \emptyset, \forall i \neq j \). Our goal is to perform DBC in multiple eigenspaces constructed with this training data set, \( \bigcup_{i=1}^c T_i \), and to classify a new sample into an appropriate class. More specifically, we generate an eigenspace for each class, i.e., class-dependent PCA, instead of a single eigenspace for the whole data (as is usually done in PCA-based methods). To achieve this, for each \( T_i \), we first find eigenvectors and eigenvalues, \( \mathbf{\mu}_k^{(i)} \) and \( \lambda_k^{(i)} \), \( (k = 1, \cdots, d) \), of the covariance matrix, \( \Sigma_i \), using \( \Sigma_i \mathbf{\mu}_k^{(i)} = \lambda_k^{(i)} \mathbf{\mu}_k^{(i)} \), and sort them in a decreasing order according to the corresponding eigenvalues, i.e., \( \lambda_1^{(i)} \geq \cdots \geq \lambda_d^{(i)} \). Next, these eigenvectors are transposed and selected to form the row vectors of a transformation matrix, \( A_i = \{ \mathbf{\mu}_k^{(i)} \}_{k=1}^d \in \mathbb{R}^d \). We then project the data samples, \( \mathbf{x}_j \) \( (j = 1, \cdots, n_i) \), into \( c \) \( q \)-dimensional subspaces, called eigenspaces, spanned by the arranged principal eigenvectors, using a transformation formula for each subset (i.e., class/cluster) as follows [12]:

\[
y_j^{(i)} = A_i^T (\mathbf{x}_j - \mathbf{m}_i), \quad 1 \leq i \leq c, \tag{2}
\]

where \( y_j^{(i)} \) denotes the \( j \)-th \( q \)-dimensional vector generated in the \( i \)-th eigenspace, \( A_i^T \) is the transpose of \( A_i \), and \( \mathbf{m}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_j \), where \( \mathbf{x}_j \in T_i \).

From Eq. (2), it can be seen that all the samples of classes are projected around the common center which is the origin of the \( q \)-dimensional space (refer to [21] for more details). Therefore, instead of \( \mathbf{m}_i \), it is required to find a reference vector such that its direction and displacement...
from the origin provides maximum separable projections in the eigenspace. The reference vector, \( m_{ref} \), can be computed as a production of the maximum eigenvalue and the corresponding eigenvector, i.e., \( m_{ref} = \mu_1 \lambda_1 \), from \( S_B \mu = \lambda \mu \), where \( S_B = \sum_{i=1}^c n_i (m_i - m)(m_i - m)^T \); \( m = \frac{1}{c} \sum_{i=1}^c m_i \); \( \mu \) and \( \lambda \) are eigenvectors and their corresponding eigenvalues, respectively.

As mentioned previously, PCA is applied separately to each subset \( T_i \) of the training data, where \( i \) runs from 1 to \( c \), the number of classes. Hence, according to Eq. (2), there should be for each sample as many projections as classes. Among the \( c \) projections, a projection, \( y_j^{(i)} \), is selected as the final projection, \( y_j \), for example, using the \( k \)-nearest neighbor rule. That is, for a test sample, \( z \), out of the \( k \)-closest neighbors to \( z \), \( \{ x_l \}_{l=1}^k \), we first identify the number \( k_i \) of the samples that belong to the class \( i \), where \( \sum_{i=1}^c k_i = k \). Then, we select \( y_j^{(i)} \) as \( y_j \), for which \( k_i > k_l \), \( \forall l \neq i \).

In order to provide an illustrative example for the reason why using the multiple eigenspaces of the class-dependent PCA, we consider two artificially generated 2-dimensional and 2-class data sets, named Simple and Highleyman [5], respectively. Here, the two classes of them are differently defined by two Gaussian distributions, \( \text{Gauss}(m_1^{(i)}, S_1^{(i)}) \) and \( \text{Gauss}(m_2^{(i)}, S_2^{(i)}) \), \( (i = 1, 2 \) for the two data sets), respectively. \(^2\) where

\[
\begin{align*}
m_1^{(1)} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, & m_2^{(1)} &= \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \\
S_1^{(1)} &= S_2^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
S_1^{(2)} &= \begin{bmatrix} 1 & 0 \\ 0 & 0.25 \end{bmatrix}, & S_2^{(2)} &= \begin{bmatrix} 0.01 & 0 \\ 0 & 4 \end{bmatrix}.
\end{align*}
\]

Then, the class priors are \( P(1) = P(2) = \frac{1}{2} \). That is, both classes of Highleyman have different distributions for each class, while those of Simple are all Gaussian distributed with identity matrix as covariance matrix as shown in Fig. 1.

Fig. 2 shows Highleyman and its PCA projections in eigenspaces computed with the class-independent PCA and the class-dependent PCA methods. From the figures, it should be observed that the number of the overlapped points of Fig. 2 (b) is larger than that of Fig. 2 (c).

In order to compare discriminatory powers of two eigenspaces computed with the class-independent PCA and the class-dependent PCA methods, respectively. \(^2\)See gendats and gendath functions in [5] for more details.
Table 1: Classification error rates (mean ± std) (%) obtained with two classifiers (knnc and ldc) in the two eigenspaces.

<table>
<thead>
<tr>
<th>datasets</th>
<th>PCA</th>
<th>knnc</th>
<th>ldc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple</td>
<td>class-independent</td>
<td>27.20 ± 3.74</td>
<td>17.63 ± 3.25</td>
</tr>
<tr>
<td></td>
<td>class-dependent</td>
<td>19.93 ± 3.41</td>
<td>19.93 ± 3.41</td>
</tr>
<tr>
<td>Highleyman</td>
<td>class-independent</td>
<td>12.03 ± 2.72</td>
<td>12.70 ± 3.87</td>
</tr>
<tr>
<td></td>
<td>class-dependent</td>
<td>12.57 ± 2.67</td>
<td>12.57 ± 2.67</td>
</tr>
</tbody>
</table>

the class-dependent PCA methods, we carry out classifications on PCA projections of Simple and Highleyman in the subspaces and then obtain a result as summarized in Table 1. Here, the k-nearest neighbor classifier / the regularized normal density-based linear classifier (i.e., knnc / ldc) [5] and the holdout method are chosen.

From the result, the reader should observe that the discriminatory power in the eigenspaces of the class-dependent PCA is different from that of the class-dependent PCA when the data have different distributions for each class.

2.3 DBC in Multiple Eigenspaces

The basic strategy of the proposed technique is to solve the classification problem by first mapping the input-feature space to eigenspaces using the class-dependent PCA, one for each class, and then performing a DBC in the eigenspaces. Therefore, the mean squared error between a feature vector, \( \mathbf{x} \), and its projection, \( \mathbf{y} = \mathbf{E} \mathbf{x} \), is: \( e^2(q; \mathbf{\mu}_k) = E \left[ \| \mathbf{x} - \mathbf{y} \|^2 \right] \), where \( E[\cdot] \) and \( \| \cdot \| \) imply the expected value and the 2-norm, respectively. Here, since \( \| \mathbf{x} \|^2 - 2 \mathbf{x}^T \mathbf{x} = - \sum_{k=1}^q (\mathbf{\mu}_k^T \mathbf{\mu}_k) = - \sum_{k=1}^q (\mathbf{\mu}_k^T \mathbf{\mu}_k) \), we have an expression for \( e^2 \) as follows:

\[
e^2(q; \mathbf{\mu}_k) = E \left[ \| \mathbf{x} \|^2 \right] - \sum_{k=1}^q \mathbf{\mu}_k^T \Sigma \mathbf{\mu}_k, \tag{3}
\]

where \( \Sigma = E \left[ \mathbf{x} \mathbf{x}^T \right] \).

From Eq. (3), it can be seen that the larger value of the second term, \( \sum_{k=1}^q \mathbf{\mu}_k^T \Sigma \mathbf{\mu}_k \), the smaller \( e^2 \). That is, we can reduce the mean squared error, \( e^2 \), when choosing appropriate basis vectors (eigenvectors) for each class. From this point of view, we employ the class-dependent PCA approach, instead of class-independent PCA that is usually done in PCA-based methods. The rationale of this strategy is presented in a later section together with experimental results. The proposed approach, which is referred to as an eigenspace DBC (EDBC), is summarized in the following:

1. Select the entire training set, \( T = \bigcup_{i=1}^n T_i \), as the prototype subset, \( P \).
2. For each subset \( T_i \), after computing a transformation matrix \( A_i \) and a mean vector \( m_i \), transform the input-feature vectors \( \mathbf{x}_j \in T_i \) (\( j = 1, \ldots, n_i \)), into the \( q \)-dimensional feature vectors \( \mathbf{y}_j^{(i)} \) using Eq. (2).
3. Using Eq. (1), compute the dissimilarity matrix \( D_{T, T} \), in which each individual dissimilarity \( d(\mathbf{x}_j, \mathbf{x}_k) \) is measured with \( d(\mathbf{y}_j^{(i)}, \mathbf{y}_k^{(i)}) \), where if \( \mathbf{x}_j \in T_i \), then \( \mathbf{y}_j^{(i)} \) and \( \mathbf{y}_k^{(i)} \) belong to the same eigenspace generated with \( T_i \), and \( d(\cdot, \cdot) \) denotes a distance measure.
4. This step is the same as Step 3 in DBC.
5. This step is the same as Step 4 in DBC.

As in the case of DBC, almost all the processing CPU-time of EDBC is also consumed in computing the transformation matrix and the dissimilarity matrix. So, the difference in magnitude between the computational complexities of DBC and EDBC depends on the computational costs associated with the two matrices. More specifically, Step 1 requires \( O(1) \) time in both algorithms. Then, in DBC, Step 2 of computing the \( n \times n \) dissimilarity matrix requires \( O(\text{dn}^2) \) time. On the other hand, the computation of that of EDBC needs \( O(d^3 + cn^2 + d^2n^2) \) time in executing Steps 2 and 3. Here, the first and the second terms, \( d^3 \) and \( cn^2 \), are for computing the eigenvalue decomposition and the transformation matrix, respectively. Next, computing a test column vector, \( \delta(x) \), requires \( O(\text{dn}) \) time at Step 3 in DBC and \( O(\text{dn} + 3dn^2) \) time at Step 4 in EDBC. Finally, in both algorithms, classification requires \( O(\gamma) \) time, where \( \gamma \) is the time for classifying the test vector with a classifier designed in the dissimilarity space. From this analysis, it can be seen that the required time for EDBC is more sensitive to the number of samples \( n \), the dimensionality \( d \), and the number of classes \( c \) than that for DBC.

3. Experimental Results

The run-time characteristics of DBCs in eigenspaces spanned by the class-independent and class-dependent PCA methods are reported below. The results obtained with a well-known benchmark data are first presented and then followed by the results achieved with certain kinds of UCI data sets.

3.1 Experiment #1 (Benchmark data: Nist38)

The proposed approach has been tested and compared with conventional methods. As a baseline experiment, we first investigate the mean squared error, \( e^2 \), with a normally distributed and well represented data, namely Nist38. The data set chosen from the NIST database [23] consists of two kinds of digits, 3 and 8, for a total of 1000 binary images. The size of each image is \( 32 \times 32 \) pixels, for a total dimensionality of 1024 pixels. Fig. 3 shows the mean squared error values of Eq. (3) of \( e^2 \) computed with the class-independent PCA and class-dependent PCA methods for the data set, where the values are represented by three lines, the dashed, solid, and dotted lines, in different colors. Also, the \( x \)-axis and the \( y \)-axis denote the selected dimensions and the \( e^2 \) values, respectively.

From the picture shown in Fig. 3, it can be observed that there is a difference in the values between the class-independent PCA and class-dependent PCA (class-1 and
class-2) methods; the mean squared error values of the two
class-dependent PCA methods are smaller than that of the
class-independent PCA for a wide range of \( q \) (\( 8 \leq q \leq 512 \));
when \( q = d \), the three values are consistent. For Nist38, the
differences are marginal, which means that subspaces
generated with eigenvectors can be further optimized by means
of class-dependent PCA, rather than class-independent one.
From this observation, we further experiment on classification
accuracies of a classifier, namely the \( k \)-nearest neighbor
classifier (NN, where \( k=1 \)), designed in six fashions: ORG, ORG-DI, DBC, DBC-NFL, EDBC-s, and EDBC-m, which
are summarized in the following:

First, two NNs of ORG and ORG-DI are designed in
the original input-feature space, but different metrics. The
former is the Euclidean distance (\( l_2 \) metric) based NN
rule, while the latter is a maximum a posteriori (MAP)
approach using a high order dissimilarity called dissimilarity
increments (DI) [7]. In the interest of brevity, the details of the
MAP-DI based NN rule are omitted here, but can be
found in the related literature [1], [2].

Secondly, two NNs of DBC and DBC-NFL are all de-
signed in the dissimilarity space. The former is that of the
conventional DBC, but the latter is a generalized DBC by
using the nearest feature lines (NFL) [15] as prototypes to
enrich a given dissimilarity representation. In the interest of
brevity, the details of DBC-NFL are also omitted here, but
can be found in the literature, including [17].

Thirdly, two NNs of EDBC-s and EDBC-m are all of
DBC's performed in eigenspaces. NN of EDBC-s is designed
in the eigenspace generated with class-independent PCA
(i.e., single eigenspace), while that of EDBC-m is designed in
the eigenspace of class-dependent PCA - one per class
(i.e., multiple eigenspaces). Here, in Eq. (2), \( m_i \) is replaced by the reference vector, \( m_{ref} \), as in [21].

The goal of this experiment is to show that classification
accuracies of DBCs can be improved by utilizing the class-
dependent PCA, without the need to select prototypes or
generalize the dissimilarity. Therefore, ORG and DBC were
included as a reference scheme, while ORG-DI and DBC-
NFL were considered as state-of-the-art techniques of the
two schemes. Also, in DBC (and DBC-NFL), a selection
strategy of RandomC was used and the number of prototypes
(and NFL), \( m \) (and \( f \)), was defined as: \( m = \min(q, n_i) \),
where \( \min \) is a function of finding the smallest component;
\( q \) and \( n_i \) are the dimension of eigenspaces and the number
of examples per class, respectively.

Fig. 4 shows a plot of the classification error rates (%)
obtained with NN classifiers designed with the six fashions: ORG, ORG-
DI, DBC, EDBC-NFL, EDBC-s, and EDBC-m, for Nist38.

The observations obtained from the plot shown in Fig. 4
are the followings. First of all, it should be pointed out
that the estimated error rate, marked with a \( \circ \) symbol,
obtained with EDBC-s decreases sharply as the cardinality
of \( q \) (and \( m \)) increases, while that of EDBC-m, marked
with a \( \square \) symbol, generally maintains a consistent height
within the entire range of \( q \), which means that, with regard
to selecting the dimensionality of subspaces or the number
of prototypes, the latter approach is more robust than the former is. Secondly, two error rates obtained with EDBC-s and EDBC-m are almost the same in the intermediate range of $q$, but, when $q = d$, classification accuracy of the latter is marginally better than that of the former as well as the others, such as ORG and DBC. (Here, the error rate of ORG-DI is almost the same as that of EDBC-m.)

Finally, it should be observed that two error rates of DBC and EDBC-s are the same when we select all of the eigenvectors as basis vectors of the eigenspace. This observation is coincident with the fact that, formally, Euclidean distances should not change after an eigenvalue decomposition that involves all eigenvectors, in which the space is just rotated. Consequently, the error rates of EDBC-s come to be the same as that of DBC, as shown in Fig. 4.

### 3.2 Experiment # 2 (UCI data)

To further investigate the advantage of using the proposed method, and, especially, to find out which kinds of significant data set are more suitable for the scheme, we repeated the above experiment # 1 with certain kinds of UCI data sets [6] as summarized in Table 2. Here, we chose four different numbers of classes of data: 2, 3, 6, and 10 classes. Moreover, the reader should observe that some data sets are class-imbalance problems, while other sets are small sample size problems.

Table 3 shows a numerical comparison of the averaged error rates and their standard deviations obtained with the NN classifiers built in the six methods, i.e., ORG, ORG-DI, DBC, DBC-NFL, EDBC-s, and EDBC-m, for the UCI data sets. Here, the experimental parameters shown in the second column, i.e., $m$ (and $f$), $q$, and $\Delta \epsilon^2$, are abbreviations for, respectively, the number of selected prototypes (and the number of nearest feature lines), the dimension of the eigenspaces, and the mean of the difference values of $\epsilon^2$ given by Eq. (3), i.e., $\Delta \epsilon^2 = mean(\{\epsilon_i^2 - \epsilon^2\}_{i=1}^c)$, where $mean$ is the mean value of the elements; $\epsilon_i^2$ and $\epsilon^2$ are computed with the class-dependent and class-independent PCAs. In order to facilitate the comparison, for each row, the lowest error rate is bold-faced. In particular, the values highlighted with a $*$ marker are the lowest ones among the four error rates of DBC and EDBC.

As shown in Fig. 4, the result in Table 3 shows again that EDBC-m generally obtains lower mean error than EDBC-s does. More specifically, consider first the values highlighted with a $*$ marker, which are the lowest ones among the four error rates of two DBCs and two EDBCs. Among them, EDBC-m has most of the lowest error rates (for the whole data sets), except for a low-dimensional and class-imbalanced data set, such as Glass. Then, the bold-faced values are of the lowest error rate among the entire six methods, including ORG and ORG-DI. Thus, it should be pointed out that, for certain kinds of data sets, such as Dermatology, Laryngeal1, and Liver, the approach of DBC, to say nothing of EDBC, does not work satisfactorily.

From the comparison, the reader should observe that, for the two EDBCs performed in the eigenspaces of class-dependent and class-independent PCAs, the error rates of the latter EDBC are generally higher than those of the former when the dimensionality of the eigenspace is appropriately chosen. From the table, and as also reported in [11], we can clearly observe the possibility of improving classification performance of DBCs by utilizing PCA, especially class-dependent PCA, rather than class-independent one.

In review, it is not easy to find out which kinds of significant data set are more suitable for EDBC-m. However, in terms of classification accuracies, DBC performed in multiple eigenspaces seems to be more useful for certain kinds of data sets than the traditional DBC does. Especially, the experimental results obtained, as shown in Fig. 4 and Table 3, demonstrate that the class-dependent PCA scheme, albeit not always, works more efficiently with the well-balanced and high-dimensional data sets than it does with the class-imbalanced and low-dimensional ones.

### 4. Conclusions

In an effort to improve the classification performance of DBC, we studied a way of utilizing class-dependent PCA, rather than employing the methods of effectively selecting prototypes or generalizing the dissimilarity. To achieve this improvement of DBC, we first computed eigenvectors and eigenvalues of the training data, one for each class. Then, we performed DBC in the eigenspaces spanned by the subset of principal eigenvectors. The proposed scheme was tested on a well represented image data set and some UCI data sets, and the results obtained were compared with those of other methods, including two state-of-the-art techniques using DI (dissimilarity increments) and NFL (the nearest feature lines). Our experimental results demonstrate that the classification accuracies of DBC in eigenspaces were improved when the dimensionality of the eigenspaces is
appropriately chosen. In particular, the experimental results demonstrate that, for the two DBCs in eigenspaces, performed in the single eigenspace for the whole data set and in the multiple eigenspaces, one per class, i.e., EDBC-s and EDBC-m, respectively, the error rates of the latter are generally lower than those of the former. Although we have shown that the classification accuracy of DBC can be improved by employing the class-dependent PCA approach, many tasks remain unchallenged. One of them is to theoretically further investigate the relationship between the improvement achieved in the multiple eigenspaces and the differences in the mean squared errors. An analysis of why using the multiple eigenspaces in DBC instead of selecting prototypes should also be performed in detail.

References


