A computationally efficient scheme for feature extraction with kernel discriminant analysis

Hwang-Ki Min, Yuxi Hou, Sangwoo Park, Iickho Song *

School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea

1. Introduction

When the dimension of data is large, a pattern recognition (PR) system is normally decomposed into two sub-systems of feature extraction and classification. Concisely speaking, feature extraction is a procedure of removing superfluous information from data, or equivalently, reducing the dimension of data. Once a feature extractor (FE) is derived, effective features can be extracted from the data, with which the efficiency and processing speed of the classification step can be improved.

In feature extraction, the independent component analysis (ICA) [1], linear discriminant analysis (LDA) [2,3], and principal component analysis (PCA) [4] are regarded as the most fundamental and powerful tools. The LDA is a supervised learning method and usually outperforms the PCA and ICA in most PR problems. In principle, the FE of the LDA is obtained by maximizing the ratio of the between-class distance to the within-class distance of data. Due to the singularity of the within-class scatter, unfortunately, the LDA becomes ill-posed for PR problems of small sample size (SSS). Here, a PR problem is called SSS when \( \frac{d}{N} \leq c \), where \( d \) is the dimension of data, \( N \) is the number of data, and \( c \) is the number of classes. Normally, \( d \) is in tens of thousands and \( N \) is up to several thousands in PR problems of SSS.

For applications in SSS problems, several LDA-related methods have been proposed in the literature. The typical examples are the PCA plus LDA [5], null space-based LDA (NLDA) [6], direct LDA [7], and LDA based on the generalized singular value decomposition (LDA/GSVD) [8]. It is noteworthy that the NLDA and LDA/GSVD are aiming at the maximization of the same objective function. Among the LDA-related methods, the NLDA, an extension of the LDA in SSS problems, attracts more and more attention due to its good performance. Meanwhile, some attempts on complexity reduction of the NLDA have been made in [9,10] by adopting eigen-decomposition and QR factorization, respectively. In order to reduce the complexity of the NLDA further, a novel scheme for the NLDA has been recently proposed in [11] based on the linear equation technique. Although the LDA and LDA-related methods can cope well with linearly separable problems, they generally result in degraded performance for linearly unseparable PR problems, in which case we may exploit kernel tricks. Interestingly, the kernel tricks can frequently provide improved PR performance for linearly separable problems also.

By extending the scope of linear discriminant methods into the kernel space, the kernel discriminant analysis (KDA) can be formulated, in which the objective is to maximize the ratio of the kernel between-class distance to the kernel within-class distance. The scheme in [12], where the KDA is viewed as an extension of the LDA, produces the FE of an approximated KDA based on a perturbation method and provides a complexity of \( \frac{1}{2}N^2d + \frac{1}{2}Nd + \frac{13}{6}N^3 \); the scheme will be called the PKDA in this paper. The generalized discriminant analysis (GDA) [13], a scheme
for finding the FE of the KDA, employs eigen-decomposition on the kernel matrix, and suffers from a complexity of \( \frac{1}{2}N^2d + \frac{1}{2}N^3 + \frac{3}{2}N^2d \). The kernel discriminative common vector (KDCV) scheme [14] adopts the null space-based criterion, which is to maximize the kernel between-class distance under the constraint of zero kernel within-class distance, and produces the FE from the viewpoint of the optimal discriminant subspace. Since eigen-decomposition is employed twice in the procedure, the KDCV suffers from a complexity of \( \frac{1}{2}N^2d + \frac{1}{2}N^3 + 9N^2 \).

Several KDA-related methods have been proposed in the literature to lower the complexity. For example, by avoiding the eigen-decomposition on the kernel matrix, the null space-based kernel Fisher discriminant analysis (NKFDA) [15] provides a complexity of \( \frac{1}{2}N^2d + \frac{1}{2}N^3d \). The method of the KDA via QR decomposition (KDA/QR) [16] exhibits a still lower complexity of \( \frac{1}{2}N^2d + \frac{1}{2}N^3d \). On the other hand, since the NKFDA takes into account neither the null space nor the non-null space of the kernel between-class scatter, the method possesses the inevitable drawback of PR performance limitation. Similarly, the KDA/QR also has the weakness of inferior PR performance because the step of removing the null space of the kernel between-class scatter unavoidably results in a partial loss of the null space of the kernel within-class scatter.

In this paper, by extending the scheme proposed in [11] into the kernel space, a new scheme is designed for the feature extraction with the KDA. The new scheme is proposed to provide a reduction of complexity without any degradation of PR performance. We first derive a reformulation of the constraint of zero kernel between-class distance into a linear expression in Theorem 1. Then, by employing Lagrange method, the problem of finding the FE of the KDA is transformed into an optimization problem, and eventually, into a linear equation problem in Theorem 2: here, it is noteworthy that Cholesky decomposition can be employed in solving the linear equation problem based on Theorem 3, which allows a reduction of complexity. At a complexity of \( \frac{1}{2}N^2d + \frac{1}{2}N^3d + \frac{1}{2}N^3 \), the proposed scheme is also shown to provide better or practically the same PR performance compared with the schemes in [12–16]. In addition, by an incorporation of regularization, we will attempt to improve the PR performance of the proposed scheme further.

### 2. Preliminary

#### 2.1. Pattern recognition

Since a multi-class PR problem can be decomposed into a multiple of two-class PR problems via the traditional one-versus-all approach [17], we will in this paper focus mainly on the two-class PR problems, in which an input vector is to be categorized into one of the two classes represented by corresponding labels.

When the dimension and the number of data get larger, building a PR system composed only of a classifier becomes more time-consuming. In such a case, to reduce the time consumption in building a PR system, an FE is usually added in front of the classifier. In short, building a PR system is divided into the two steps of building an FE and a classifier. We will concentrate only on EFS in this paper: one of the major reasons is that even a simple classifier performs well if effective features are extracted successfully from an FE.

#### 2.2. Kernel EFS

By reducing the dimension of data, an FE produces vectors of lower dimension, which are called the features. If the features are carefully extracted, containing a sufficient amount of relevant information on the inputs, the classifier in the subsequent step can process the features instead of the full size data, and save the processing time (Table 1).

A kernel FE can in general be described as

\[
F(x) = w^T \Phi(x) + b^T, 
\]

where \( x \in \mathbb{R}^d \) denotes the input, \( \Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d \) is called the mapping function, \( w \) is called the kernel discriminant vector (KDV) with dimension \( d \), and \( b^T \) is a scalar called the bias.

In the training stage of feature extraction, given a function \( \Phi(\cdot) \) and a training set \( \mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \) of data \( x_i \) and labels \( y_i \in \{-1, 1\} \), the objective is to determine the KDV \( w \) and bias \( b \), which can be accomplished by employing the PKDA, GDA, KDCV, NKFDA, and KDA/QR, for example. Let \( X = [x_1, x_2, \ldots, x_N] \) denote the \( d \times N \) training data matrix, and

\[
\Phi(X) = \begin{bmatrix} \Phi(x_1) & \Phi(x_2) & \cdots & \Phi(x_N) \end{bmatrix},
\]

\( w = \Phi(X)\eta^T \)

the kernel FE in (1), after the training with \( \mathcal{D} \), can be rewritten as

\[
F(x) = (\eta^T)\Phi^T(X)\Phi(x) + b^T,
\]

where \( \eta^T \) is the coefficient vector of size \( N \times 1 \). From the relationship (3), the objective is now changed to determining the coefficient vector \( \eta^T \).

Assume a non-linear mapping function \( \Phi(\cdot) \) defined implicitly via the \( N \times N \) kernel matrix \( K = [K_{ij}] \), where

\[
K_{ij} = \Phi^T(x_i)\Phi(x_j) = \exp \left( -\frac{||x_i - x_j||^2}{2\sigma^2} \right)
\]

with \( \sigma \) being called the inner scale and \( || \cdot || \) denoting the Euclidean norm.

Then, the KDV of the KDA can be obtained as

\[
w_{KDA} = \arg \max_{||w||=1} w^T S_{W}^T w,
\]

where

\[
S_W = \sum_{i=1}^{N} (\Phi(x_i) - m^W)(\Phi(x_i) - m^W)^T
\]

is the \( d \times d \) kernel between-class scatter and

\[
S_{W} = \sum_{i=1}^{N} (\Phi(x_i) - m^W)(\Phi(x_i) - m^W)^T
\]

is the \( d \times d \) kernel within-class scatter. Here, \( m^W = \frac{1}{N} \sum_{i=1}^{N} \Phi(x_i) \) denotes the kernel sample mean vector for class \( l \), \( Z_l = \{i : x_i \in \text{class } l\} \) denotes the index set for class \( l \), and \( N_l \) denotes the number of elements in \( Z_l \). We will assume that the data \( x_i \) have been ordered so that the label vector can be expressed as \( y = [y_1, y_2, \ldots, y_N] = [1_{y_1=n_l} - 1_{y_1=n_l}]^T \), where \( 1_{y_1=n_l} \) is the \( a \times b \) matrix of all 1’s. It is well-known [12] that the problem (6) becomes ill-posed since the maximization in (6) makes the kernel within-class distance \( \sum_{i=1}^{N} S_{W}^T w \) approach zero. In order to circumvent such singularity, a perturbation method is employed in the PKDA.

By reformulating (6), the KDV in the proposed scheme is obtained as

\[
w_{KDA} = \arg \max_{||w||=1} w^T S_{W}^T w \text{ subject to }
\]

subject to
w' S_k^k w = 0. \hspace{2cm} (10)

Note that the criterion of (9) and (10), which is basically to maximize the kernel between-class distance while keeping the kernel within-class distance zero, is the same as those of the GDA and KDCV.

Using $m_0^k = \Phi(X) |x_{0}, 1_x X0_{1}, N_0 {1}_{0}, 1_x X0_{1}, N_0 ^{T}$, $m_1^k = \Phi(X) |x_{1}, 1_x X0_{1}, N_0 {1}_{0}, 1_x X0_{1}, N_0 ^{T}$, (3), (7), and (8), we can transform (9) and (10) into

$$\eta_k^p = \arg \max_{\eta^p \in \mathcal{K}} (\eta^p) ^{K} K \eta^p$$

and

$$(\eta^p)^{T} K L K \eta^p = 0.$$ \hspace{2cm} (12)

respectively, where $V = S_l S_l ^{T}$ and

**Table 1**

Nomenclature.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_i^0$</td>
<td>Solution vector of linear equation problem (18)</td>
</tr>
<tr>
<td>$a_i^{0,R}$</td>
<td>Solution vector of regularized linear equation problem (31)</td>
</tr>
<tr>
<td>$A_k$</td>
<td>Problem matrix of linear equation problem (18)</td>
</tr>
<tr>
<td>$\tilde{A}_k$</td>
<td>Modified problem matrix (20)</td>
</tr>
<tr>
<td>$\eta^p$</td>
<td>Bias of kernel FE (1)</td>
</tr>
<tr>
<td>$c$</td>
<td>Number of classes</td>
</tr>
<tr>
<td>$d$</td>
<td>Dimension of data</td>
</tr>
<tr>
<td>$\eta^{p,R}$</td>
<td>Coefficient vector of kernel FE (4)</td>
</tr>
<tr>
<td>$\eta^{p,R}$</td>
<td>Coefficient vector of proposed scheme with regularization</td>
</tr>
<tr>
<td>$\phi(\cdot)$</td>
<td>Mapping function from original data space to higher dimensional space</td>
</tr>
<tr>
<td>$\Phi(\cdot)$</td>
<td>Training data matrix in kernel space, $\Phi(\cdot) = [\Phi(x_1), \Phi(x_2), ..., \Phi(x_N)]$</td>
</tr>
<tr>
<td>$K$</td>
<td>Kernel matrix</td>
</tr>
<tr>
<td>$m_i^k$</td>
<td>Kernel sample mean vector for class $l$</td>
</tr>
<tr>
<td>$\nu_{G,CCV}$</td>
<td>Regularization parameter</td>
</tr>
<tr>
<td>$\nu_{G}$</td>
<td>Sub-optimal value of regularization parameter</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of training data</td>
</tr>
<tr>
<td>$N_l$</td>
<td>Number of training data for class $l$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Inner scale of Gaussian kernel</td>
</tr>
<tr>
<td>$S_i^k$</td>
<td>Kernel between-class scatter</td>
</tr>
<tr>
<td>$S_i^k$</td>
<td>Kernel within-class scatter</td>
</tr>
<tr>
<td>$w$</td>
<td>Kernel discriminant vector of kernel FE (1)</td>
</tr>
<tr>
<td>$x_l, y_l$</td>
<td>Training data and corresponding label</td>
</tr>
<tr>
<td>$X$</td>
<td>$d \times N$ training data matrix, $X = [x_1, x_2, ..., x_N]$</td>
</tr>
<tr>
<td>$y$</td>
<td>Label vector for training data, $y = [y_1, y_2, ..., y_N]$</td>
</tr>
<tr>
<td>$y_l$</td>
<td>Normalized label vector for linear equation problem (18)</td>
</tr>
<tr>
<td>$Z_l$</td>
<td>Index set for class $l$, $Z_l = {i : x_i \in \text{class} l}$</td>
</tr>
</tbody>
</table>

**Table 2**

Objectives and complexities of KDA-related schemes in two-class PR problems.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Objective (criterion)</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKDA</td>
<td>Maximize $w' S_k^k w$ under the condition $</td>
<td></td>
</tr>
<tr>
<td>GDA</td>
<td>Maximize $w' S_k^k w$ under the conditions $</td>
<td></td>
</tr>
<tr>
<td>KDCV</td>
<td>Maximize $w' S_k^k w$ under the conditions $</td>
<td></td>
</tr>
<tr>
<td>NKFDA</td>
<td>Maximize $w' S_k^k w$ under the conditions $</td>
<td></td>
</tr>
<tr>
<td>KDA/QR</td>
<td>Maximize $w' S_k^k w$ under the conditions $</td>
<td></td>
</tr>
<tr>
<td>KPCA (1st principal component)</td>
<td>Maximize $w' S_k^k w$ under the conditions $</td>
<td></td>
</tr>
<tr>
<td>Proposed scheme</td>
<td>Maximize $w' S_k^k w$ under the conditions $</td>
<td></td>
</tr>
</tbody>
</table>

**3. Reduction of complexity for the FE**

In this section, we will establish a new scheme to obtain the coefficient vector $\eta_k^p$ shown in (11). The new scheme will be shown to require a lower complexity than the PKDA, GDA, KDCV, NKFDA, and KPCA in obtaining the FE.

**3.1. FE of the KDA**

Let us first derive relationships equivalent to the constraint (12).

**Theorem 1.** When $\zeta^p = s_k ^{T} K \eta^p$ is non-zero and $K$ is of full-rank, the constraint (12) of the problem $(\eta^p)^{T} K L K \eta^p = 0$ is equivalent to...
\[ a^0 K \phi^0 + b^0 1_{N \times 1} = y. \] (16)

where \( a^0 = \frac{2}{c^0} \) and \( b^0 = - \frac{s^2_1 K \phi^0}{c^0} \), with \( s_2 = \left[ \frac{1}{N^2} 1_{N \times N} - \frac{1}{N} 1_{N \times 1} \right]^T \).

Using Theorem 1, the solution \( \eta_k^B \) of (11) under the constraint (12) can be obtained as follows.

**Theorem 2.** Under the constraint (12), the solution \( \eta_k^B \) of (11) can be computed via

\[ \eta_k^B = \pm \left( \left( a_k^0 \right)^T K a_k^0 \right)^{-1/2} a_k^0, \] (17)

where the \( N \times 1 \) vector \( a_k^0 \) denotes the solution to the linear equation problem

\[ A_k a_k^0 = y_k \] (18)

with

\[ A_k = K - \frac{1}{N} 1_{N \times N} K - \frac{1}{N} K 1_{N \times N} \] (19)

and \( y_k = \left[ - \frac{N}{N^2} 1_{N \times N} + \frac{N}{N^2} 1_{N \times N} \right]^T \).

In short, once \( a_k^0 \) is obtained, the coefficient vector \( \eta_k^B \) can be evaluated via (17), with which the bias \( \lambda_k^B \) can be evaluated via (15). The features \( \{ F(x_i) \} \) can subsequently be calculated via (4). The proofs of Theorems 1 and 2 are given in Appendices A and B, respectively.

We would like to note that the linear equation problem (18) is derived under the assumption that the scalar \( c^0 \) is known and the matrix \( A_k \) is of full-rank. Now, when \( s^2_1 K \phi^0 \) is zero, the problem (11) becomes ill-posed because \( (\eta_k^B) \) \( K \phi \) is singular, we can acquire \( \lambda_k^B \), where \( \lambda_k^B \) will be taken into account in the complexity reduction of the proposed scheme.

### 3.3. Analysis of complexity

Denote one multiplication operation by complexity 1, and assume one square root or division operation as complexity 1 also [23,22]. For convenience, let

\[ H = x^T x + x^T x - 1_{N \times 1} n_i - n_i 1_{1 \times N}, \] (24)

where

\[ n_i = \left[ \left\| x_1 \right\|^2 \left\| x_2 \right\|^2 \ldots \left\| x_N \right\|^2 \right] \] (25)

denotes the \( N \times 1 \) vector of the Euclidean 2-norms of \( \{ x_i \} \).

**Step 1:** Calculation of the matrix \( A_k \). \( A_k \) requires a complexity of \( \frac{1}{2} N^2 d + \frac{1}{2} N d + \frac{3}{2} N^2 + \frac{11}{2} N + 1 \) as described herein. Consider the calculation of \( A_k \) via \( \tilde{A}_k \). First, the kernel matrix \( K = \exp \left( \frac{H_0}{2} \right) \), where \( H_0 \) denotes the (i, j)th element of the matrix \( H \), can be obtained with \( \frac{1}{2} N^2 d + \frac{3}{2} N d + \frac{3}{2} N^2 + \frac{3}{2} N^2 \) multiplications and \( \frac{1}{2} N^2 d + \frac{3}{2} N d + 2 N^2 + 4 N \) additions, based on the following five observations: (i) due to the symmetry of \( x^T x \), only \( \frac{1}{2} N(N + 1) \) positive elements in \( x^T x \) are to be calculated. Since \( d \) multiplications and \( d - 1 \) additions are required for obtaining each of the \( \frac{1}{2} N(N + 1) \) elements, we require a complexity of \( \frac{1}{2} N(N + 1) d \) for calculating \( x^T x \). (ii) Since \( \left\| x_i \right\|^2 \) can be obtained with \( d \) multiplications and \( d - 1 \) additions, \( N d \) multiplications and \( N(d - 1) \) additions are required to obtain \( n_i \) in (25). (iii) When \( x^T x \) and \( n_i \) are available, we can obtain \( H \) from (24) with \( \frac{1}{2} N(N + 1) \) additions because of the symmetry of \( H \) since 3 additions are necessary for the calculation of each of the \( N(N + 1) \) elements of \( H \). (iv) To obtain \( \frac{n_i}{\sqrt{n_i}} \) from \( H \), \( \frac{1}{2} N(N + 1) \) divisions are required, resulting in a complexity of \( \frac{1}{2} N(N + 1) \). (v) Each element of the \( N \times N \) symmetric matrix \( \exp \left( \frac{H_0}{2} \right) \) can be evaluated with relative error smaller than \( 2^{-n} \) from \( \sqrt{n} \) multiplications and \( \sqrt{n} \) additions, where \( n \) is the number of digits to store an element [24]. Thus, choosing \( n = 64 \), sufficiently large for most practical applications, \( 4N(N + 1) \) multiplications and \( 4N(N + 1) \) additions are required to obtain \( \exp \left( \frac{H_0}{2} \right) \). Second, the matrix \( A_k \) can be obtained with \( N \) multiplications and \( N(N - 1) + 2 \times \frac{1}{2} N(N + 1) = N^2 \) additions from \( K \) by (19) based on the following three observations: (i) the \( N \) rows of \( \frac{1}{2} N(N + 1) \) are all the same while the jth element \( \mu_j^B \) of a row can be obtained with 1 multiplication and \( N - 1 \) additions by adding all the elements of the jth column of \( K \) and then dividing the result by \( N^2 \) for \( j = 1, 2, \ldots, N \). (ii) The matrix \( \frac{1}{2} N(N + 1) \) is simply the transpose of \( \frac{1}{2} N(N + 1) K \). (iii) Since \( A_k \) is symmetric, only \( \frac{1}{2} N(N + 1) \) elements of \( A_k \) need to be calculated, each with 2 additions when \( K, \frac{1}{2} N K 1_{N \times N} \), and \( \frac{1}{4} N(K 1_{N \times N}) \) are available. Finally,
after choosing a number satisfying (21) for \(e^c_i\), we can obtain \(A_k\) with \(\frac{1}{2}N(N+1)\) additions from \(A_k\) by (20). Note that, because of (23), the calculation of \(e^c_i\) requires only one division and \(N-1\) additions when the numbers \((b^T)^T_{i+1}\) are available.

**Step 2:** It requires a complexity of \(\frac{1}{2}N\) to acquire \(a^c_i\) from \(A_k\) as described herein. We need a complexity of \(\frac{1}{2}N^2 + \frac{1}{2}N - \frac{2}{3}N\) for the decomposition of \(A_k\) into

\[
A_k = R_k^T R_k
\]

by using the method of Gaussian elimination with partial pivoting, where \(R_k\) is an upper triangular matrix with positive diagonal elements. With Cholesky decomposition (26) completed, we can acquire \(a_i^c\) from (22) with \(N(N+1)\) multiplications and \(N(N-1)\) additions [21]; specifically, an intermediate vector \(x_i\) will first be obtained by solving \(R_k^T x_i = y_i\) with \(\frac{1}{2}N(N+1)\) multiplications and \(\frac{1}{2}N(N-1)\) additions via the backward substitution algorithm. We can then solve \(R_k^T x_i = y_i\) for \(a_i^c\) with \(\frac{1}{2}N(N+1)\) multiplications and \(\frac{1}{2}N(N-1)\) additions via the forward substitution algorithm.

**Step 3:** A complexity of \(N(N+2)\) is necessary for calculating the KDA coefficient vector \(\tilde{Y}_N\) from (17) when \(a_i^c\) is available. First, to obtain \(\tilde{K}_{NN}\) from \(a_i^c\) requires \(N^2\) multiplications and \(N(N-1)\) additions. Second, with \(N\) multiplications, \(N-1\) additions, and 1 square root operation, we can calculate \(\sqrt{(a_i^c)^T \tilde{K}_{NN}}\). Thus, the coefficient vector \(\tilde{Y}_N\) can eventually be acquired with additional \(N\) divisions from dividing each element of \(a_i^c\) by \(\sqrt{(a_i^c)^T \tilde{K}_{NN}}\).

In summary, the complexity requirement of the proposed scheme is \(\frac{1}{2}N d + \frac{1}{2}N d + \frac{1}{2}N^2\) for two-class PR problems. From Table 2 shown in Section 2, it is anticipated that the proposed scheme will clearly require a lower complexity than the PKDA, GDA, KDCV, NKFDA, and KPCA. Although the proposed scheme is anticipated to require a higher complexity than the KDA/QR, the difference is negligibly small in SSS problems and the proposed scheme can provide a better PR performance than the KDA/QR as we shall see shortly in Section 4.

Let us now consider the complexity requirement of the proposed scheme for multi-class PR problems. Each of the \(c\) two-class PR problems decomposed from a \(c\)-class problem can be transformed into a linear equation problem (22) with the same \(A_k\) but different \(y_i\). Therefore, once one of the two-class PR problems is solved, we can solve the remaining two-class PR problems with an additional complexity \((c-1)N(N+1)\) in Step 2 and \((c-1)(N(N+2)+1)\) in Step 3; note that the calculation of \(A_k\) in Step 1 and Cholesky decomposition of \(A_k\) in Step 2 are not required after the first two-class problem. In short, the complexity of the proposed scheme for \(c\)-class problems is \(\frac{1}{2}N d + \frac{3}{2}N d + \frac{1}{2}N^2\), the same as that for two-class problems except for minor terms such as \(N^2\). Similarly, it can be anticipated that the complexities of the other schemes for multi-class problems are not different significantly from those for two-class problems shown in Table 2.

### 4. Results of experiments

Let us address the experimental complexity and PR performance of the proposed scheme in comparison with other schemes. All the schemes considered here are implemented in MATLAB program and run on an Intel (R) Core (TM) i7 PC with 3.40 GHz CPU and 8 GB RAM.

#### 4.1. Sample sets

Five sample sets, two for non-SSS problems and the other three for SSS problems, are employed in the experiments. In terms of the number of classes, three sample sets are two-class PR problems, and the other two are multi-class PR problems. Some of the important statistics of the five sample sets are summarized in Table 3:

<table>
<thead>
<tr>
<th>Sample set</th>
<th>Total number of samples</th>
<th>(d)</th>
<th>(c)</th>
<th>(\sigma)</th>
<th>Number of training samples per class (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAVEFORM</td>
<td>5000</td>
<td>21</td>
<td>2</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>RINGNORM</td>
<td>7400</td>
<td>20</td>
<td>2</td>
<td>2</td>
<td>0.06</td>
</tr>
<tr>
<td>CRANMED</td>
<td>2431</td>
<td>300</td>
<td>0.2</td>
<td>200</td>
<td>0.4</td>
</tr>
<tr>
<td>TDT2</td>
<td>9394</td>
<td>30</td>
<td>1</td>
<td>30</td>
<td>0.4</td>
</tr>
<tr>
<td>ORL</td>
<td>400</td>
<td>10</td>
<td>0.08</td>
<td>3</td>
<td>3.5, 7, 9</td>
</tr>
</tbody>
</table>

#### 4.2. Experimental results

#### 4.2.1. Non-SSS problems

(a) WAVEFORM: the sample set WAVEFORM [25] consists of 5000 samples \(\{x_i, y_i\}_{i=1}^{5000}\) of dimension 21 generated by

\[
x_{ij} = \begin{cases} 
uh_{ij}(1) + (1 - u)h_{ij}(2) + e_i, & \text{(class 'a')} 

uh_{ij}(1) + (1 - u)h_{ij}(2) + e_i, & \text{(class 'b')} 

uh_{ij}(1) + (1 - u)h_{ij}(1) + e_i, & \text{(class 'c')} 
\end{cases}
\]

where \(e_i\) denotes the \(j\)th element of the \(i\)th sample; \(u\) is a uniform random variable over \([0, 1]\); \(e_i\) is a Gaussian random variable with mean 0 and variance 1; and \(h_{ij}(1) = \max(6 - y_i - 11u, 0)\), \(h_{ij}(2) = h_{ij}(1) - 4\), and \(h_{ij}(3) = h_{ij}(2) + 4\) are triangular waveforms. Since we employ this sample set as a two-class PR problem in this paper, we will take 1647 samples from class 'a' as the first class and 3353 samples from classes 'b' and 'c' as the second class. In essence, we have 5000 vectors of size \(21 \times 1\) and the corresponding labels.

(b) RINGNORM: in the sample set RINGNORM [26], 3664 and 3736 column vectors are generated from the multivariate normal distributions \(N(0, e_{\times 1}, 4 I_{20})\) and \(N(\frac{1}{2} I_{20}, 1, I_{20})\), respectively. Subsequently, we form a \(20 \times 7400\) matrix \(\tilde{X} = \{x_i\}_{i=1}^{7400}\) by arranging the 7400 vectors \(\{x_i\}_{i=1}^{7400}\) column by column. Then, the data matrix \(\tilde{X} = [x_i]\) are defined by the standardization of the elements \(\{\tilde{x}_{ij}\}\) of \(\tilde{X}\) in a ‘row-wise’ sense as \(\tilde{x}_{ij} = x_{ij} - \bar{x}_j\), where \(\bar{x}_j = \frac{1}{7400} \sum_{i=1}^{7400} x_{ij}\); \(\tilde{C}_j = \frac{1}{7400} \sum_{i=1}^{7400} (\tilde{x}_{ij} - \bar{x}_j)^2\) are the sample-mean and sample-variance, respectively, of the \(j\)th row of \(\tilde{X}\). The labels are determined by the corresponding distributions.

#### 2. Sample sets for SSS problems

(a) CRANMED: the sample set CRANMED [27] consists of 2431 documents of abstracts, 1398 aerodynamic samples from the Cranfield collection and 1033 biomedical samples from the Medlar collection. After 41,681 professional terms appearing frequently in aerodynamics and biomedicine are collected, each of the 2431 documents is transformed into a 41,681 \(\times 1\) vector associated with a label by counting the frequency that each of the 41,681 terms appears in the document. In essence, we have 2431 column vectors of dimension 41,681 and the corresponding labels.

(b) TDT2: the sample set of the second phase of the topic detection and tracking (TDT2) project [28] consists of 9394 samples in 30 classes collected from 2 news wires (APW, NYT), 2 radio programs (VOA, PRI), and 2 television programs (CNN, ABC). By counting the frequency of 36,771 terms appearing at least twice in the 9394 samples, each document is
transformed into a 36,771 × 1 vector. In short, we have 9394 column vectors of dimension 36,771 and the corresponding labels for 30 classes in the sample set of TDT2.

(c) ORL: the Olivetti-Oracle Research Lab (ORL) sample set [29] consists of 400 face images of 10,304 pixels for 40 individuals with variations in pose, illumination, facial expression, and facial details. By measuring the grey level in each pixel, an image can be transformed into a 10,304 × 1 vector. We then have 400 column vectors and the corresponding labels for 40 classes.

4.2. CPU time consumption

As discussed in [30], the value of the inner scale σ in (5) influences significantly on the PR performance of kernel methods. In this paper, therefore, we have first explored the influence of σ on the PR performance for several values of σ ranging from 0.01 to 1000, and determined the values of σ at which the average PR accuracies reach the highest. These values are listed also in Table 3.

The CPU time consumption in constructing the kernel FEFs by various schemes are then measured. Specifically, by randomly choosing k samples from each of the c classes, we have built matrices of training data of sizes 21 × 2k, 20 × 2k, 41, 681 × 2k, 36, 771 × 30k, and 10, 304 × 40k from the sample sets WAVEFORM, RINGNORM, CRANMED, TDT2, and ORL, respectively: here, note that the number N of training samples is ck. To make the comparison results reliable, the CPU time consumed to obtain the kernel FE for every value of k is averaged over 250 repetitions. The average values of CPU time consumed to obtain the kernel FEs are shown in Tables 4–8. In addition, for easy comparison of relative CPU time, the ratios of the CPU time consumed for various kernel schemes relative to the proposed scheme are provided in Figs. 1–5.

From the simulation results shown in Tables 4–8 and Figs. 1–5, it is clearly observed that the relative CPU time consumptions for obtaining the kernel FEs are generally in good accordance with the expectation from the analytical results shown in Table 2. Specifically, we can make the following observations:

(A) The CPU time consumption is the most demanding for the KDCV, and followed by that for the GDA, PKDA, NKFDA, KPCA, proposed scheme, and KDA/QR. The proposed scheme consumes less CPU time than the KDCV, GDA, PKDA, NKFDA, and KPCA.

(B) The ratios (16.25–32.85, 17.07–36.62, 1.09–1.37, 1.40–2.27, and 1.65–2.03 in Figs. 1, 2, 3, 4 and 5, respectively) of the CPU time consumed for the GDA to that for the proposed scheme are generally higher than what (25.09–26.20, 25.18–26.24, 108–133, 114–135, and 110–130 for Figs. 1, 2, 3, 4 and 5, respectively) we can expect from Table 2: This is due to the fact that the eigen-decomposition in the simulations consumes more CPU time than we expect. Because of a similar reason, the KDCV generally consumes more CPU time than we can expect from Table 2.

(C) For non-SSS problems, Tables 4 and 5 and Figs. 1 and 2 clearly show that the CPU time consumed for the proposed scheme is more than that for the KDA/QR. On the other hand, for SSS problems, it is clearly observed in Tables 6–8 and Figs. 3–5 that the CPU time consumption for the proposed scheme is practically the same as that for the KDA/QR, which is in good agreement with the theoretical analysis shown in Table 2.

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Average CPU time (in seconds) consumed to obtain the kernel FEs for sample set WAVEFORM.</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>PKDA</td>
</tr>
<tr>
<td>400</td>
<td>0.167</td>
</tr>
<tr>
<td>600</td>
<td>0.507</td>
</tr>
<tr>
<td>800</td>
<td>1.122</td>
</tr>
<tr>
<td>1000</td>
<td>2.091</td>
</tr>
</tbody>
</table>

![Fig. 1. Ratio of CPU time relative to the proposed scheme for WAVEFORM.](image-url)
that the ratio \( \frac{1}{\sqrt{2^{N_{d}^3}}} + \frac{1}{2^{Nd}} \approx \frac{N^3}{32(N+1)} \) of the third dominant term to the sum of the first two dominant terms of the complexity of the proposed scheme is not higher than 2% for the values of \( N \) and \( d \) in the simulations of SSS problems. For example, the ratio is about 1.3% when \( N = 2k = 1600 \) and \( d = 36,771 \).

4.3. Comparison of PR performance

In terms of the PR accuracy, we have investigated the PR performance of several schemes when the FEs are combined with a classifier. Clearly, a higher PR accuracy of a PR system implies a better FE. Here, the PR accuracy denotes the ratio of the number of vectors (called testing samples) whose labels are estimated correctly by a PR system to the number \( M \) of vectors used in the estimation, where \( M = 5000 - 2k \), \( 7400 - 2k \), \( 2431 - 2k \), \( 9394 - 30k \), and \( 400 - 40k \) for WAVEFORM, RINGNORM, CRANMED, TDT2, and ORL, respectively. The PR accuracies of several schemes for the five sample sets are shown in Figs. 6–10, respectively.
For the results shown in Figs. 6–10, we have employed the nearest neighbor (NN) classifier because the NN classifier is simple, intuitive, and shown to be quite effective [31] in many cases. Briefly speaking, a PR system with the NN classifier operates in the following way: the feature $F(x_{\text{test}})$ of a testing sample $x_{\text{test}}$ and features $F(x^j)_{j=1}^{N}$ of training samples $x^j_{j=1}^{N}$ are calculated via (4) after the acquirement of the kernel FE. Then, based on the Euclidean distance between $F(x_{\text{test}})$ and $F(x^j)$, the NN classifier estimates the label for the testing sample $x_{\text{test}}$ as $y_{\text{test}} = y^m$, where $y^m$ is the label of the training sample whose feature is closest to $F(x_{\text{test}})$. Note that the PR accuracy of a PR system will be 100% for any subset of training samples due to the intrinsic characteristics of the NN classifier.

Based on the simulation results shown in Figs. 6–10, we can make the following observations:

(A) The PR accuracies of the proposed scheme, PKDA, GDA, and KDCV are very close to each other, and in particular, practically the same for the two-class PR problems.

(B) The proposed scheme provides a better PR accuracy than the NKFDA and KPCA. Specifically, the maximum gain reaches 4% and 14% over the NKFDA and KPCA, respectively.

(C) The proposed scheme provides us with an average gain of around 3% in the PR accuracy over the KDA/QR for the sample sets WAVWFORM, RINGNORM, and CRANMED. For the sample sets TDT2 and ORL, the gain ranges from 1% to 2%.

5. Incorporation of regularization

Let us now try to incorporate regularization into the proposed scheme in order to improve the PR performance further. First, (22) can be expressed equivalently as

$$a^p = \arg\min_{a^p} \frac{1}{N} \| y^p - \tilde{A}_k a^p \|_2^2. \tag{28}$$

Next, by adjusting (28) into the regularization framework [32], it is straightforward to see that the solution $a_{\nu,k}^p$ with regularization can be obtained as

$$a_{\nu,k}^p = \arg\min_{a^p} \left( \frac{1}{N} \| y^p - \tilde{A}_k a^p \|_2^2 \right) + \frac{\nu_k}{N} (\alpha^p)^T \tilde{A}_k (\alpha^p)^p. \tag{29}$$
where $\mu_R > 0$ is the regularization parameter. Recall that the framework (29) with regularization has been proven [33] to be more useful than (28) without regularization based on statistical learning theory.

Next, setting the derivative of the objective function in (29) with respect to $\alpha^k$ equal to zero, we can show that the solution $\alpha^k_R$ satisfies

$$A^T_k A^k_R + \mu_k \alpha^k_R = A^T_k y_h,$$

which can easily be simplified as

$$(A_k + \mu_k I) \alpha^k_R = y_h$$

(31)

since $A^k_R = A_k$. Note that we can employ Cholesky decomposition on $A_k + \mu_k I$ in obtaining $\alpha^k_R$ from (31) since $A_k + \mu_k I$ is positive definite when $A_k$ is positive definite. As we have done for (17), once the solution $\alpha^k_R$ is obtained, the coefficient vector can be evaluated via

$$n^k_R = \pm \left((\alpha^k_R)^T K_\rho \alpha^k_R\right)^{-1/2} \alpha^k_R$$

Clearly, if the regularization parameter $\mu_k$ is pre-specified or chosen arbitrarily, the complexity of the proposed scheme with regularization will be the same as that without regularization since the additional computation incurred by the regularization is only $N$ additions in the proposed scheme: in fact, the advantage of incorporating regularization lies in an improvement of the PR performance as we shall see shortly.

The optimal value of the regularization parameter has been addressed in [34], with which we can obtain the best PR performance: unfortunately, the optimal value is not determined easily in real applications because it is based on the distribution generating the observed data. Aiming for a sub-optimal value of the regularization parameter, the generalized cross-validation (GCV) [35] provides us with a reasonable way of choosing a sub-optimal value for the regularization parameter as

$$\mu^\text{GCV}_R = \arg \min_{\mu_R > 0} \frac{\left\|A^T_k \alpha^k_R - y_h\right\|^2}{\text{tr}^2\left(I_N - A_k (A_k + \mu_k I)^{-1}\right)}$$

(33)

When the regularization with the parameter $\mu^\text{GCV}_R$ is incorporated, the coefficient of $N^2$ in the complexity of the proposed scheme shown in Table 2 will change from $\frac{1}{\pi}$ to $\frac{1}{3}$. In order to evaluate the performance of the proposed scheme with regularization, we have first obtained the values of $\mu^\text{GCV}_R$ with (33) by employing 100, 300, and 120 samples randomly selected from the two-class sample sets (WAVEFORM, RINGNORM, and CRANMED), TDT-2, and ORL, respectively. The values of $\mu^\text{GCV}_R$ are listed in Table 3. The PR accuracies of the proposed scheme with these values of the regularization parameter $\mu^\text{GCV}_R$ are shown in Figs. 6–10 for the five sample sets. In the experiments, the average CPU time consumed for the proposed scheme with regularization is practically the same as that without regularization shown in the last columns of Tables 4–8, because we use the pre-calculated values of $\mu^\text{GCV}_R$ for regularization. Strictly speaking, when we evaluate the complexity of the proposed scheme with regularization, we should include the complexity of calculating the regularization parameter: yet, there exist many situations in real applications where we do not need to consider this additional complexity. For example, if we construct an FE for various subsets from a common sample set, the sub-optimal regularization parameter $\mu^\text{GCV}_R$ needs to be estimated only once; that is, only for the first subset.

From the performance comparisons of the proposed scheme shown in Figs. 6–10, it is clearly observed that the regularization can provide us with higher PR accuracies for the proposed scheme in all the problems. Specifically, when the regularization is incorporated into the proposed scheme, we can expect an enhancement of the PR accuracy by up to 2% on the average.

6. Conclusion

In this paper, we proposed a computationally efficient scheme for the feature extraction with the KDA. By representing the null space of the kernel within-class scatter via linear equations and employing Lagrange technique, we have first transformed the problem of deriving the FE of the KDA into a linear equation problem. When the proposed scheme seeks for the FE of the KDA by solving the linear equation problem, Cholesky decomposition is employed as a means of lowering the complexity. The proposed scheme provides us with a reduction of complexity without degradation of PR performance.

The complexity analysis and simulation results have confirmed that the proposed scheme offers us a lower complexity than the PKDA, GDA, and KDCV at practically the same PR performance. Compared with the NKFDA and KPCA, the proposed scheme provides not only a lower complexity but also a better PR performance. Although the proposed scheme requires a higher complexity than the KDA/QR, simulation results have revealed that a significantly better performance can be achieved by the proposed scheme than by the KDA/QR. Finally, we have also shown that the PR performance of the proposed scheme can further be enhanced by incorporating regularization.

Conflict of interest

None declared.

Acknowledgement

The authors wish to thank the Associate Editor and anonymous reviewers for their constructive suggestions and helpful comments. This work was supported by the National Research Foundation of Korea, with funding from the Ministry of Science, Information and Communications Technology, and Future Planning, under Grant NRF-2015R1A2A1A01005868.

Appendix A. Proof of Theorem 1

It is shown in [36] that $z^T D z = 0 \Leftrightarrow z = 0$ for a positive semi-definite matrix $D$ and a real vector $z$. Thus, for the coefficient vector $\alpha^\rho$, we can prove $(\alpha^\rho)^T K L K \alpha^\rho = 0 \Leftrightarrow K L K \alpha^\rho = 0$ by showing that the matrix $K L K$ is positive semi-definite. Now, from the definition of $L$ in (13), we have $L L = L$. Then, the positive semi-definiteness of $K L K$ is obvious since $z^T K L K z = \|L z\|^2 \geq 0$ due to the symmetry of $K$ and $L$.

Let us next prove $K L K \alpha^\rho = 0 \Leftrightarrow \alpha^\rho K L K \alpha^\rho = 0 \Leftrightarrow a^\rho K L K \alpha^\rho = 0$.

(i) When $a^\rho K L K \alpha^\rho = 0 \Leftrightarrow a^\rho K L K \alpha^\rho + b^\rho 1_{N+1} = y$, we get

$$K L K \alpha^\rho = \frac{1}{2} s^T K \alpha^\rho K 1_{N+1} = \frac{1}{2} s^T K \alpha^\rho K y,$$

(34)

after substituting $a^\rho$ and $b^\rho$ with $\frac{s^T K \alpha^\rho}{c^\rho}$ and $\frac{s^T K \alpha^\rho}{c^\rho}$, respectively, and multiplying $\frac{c^\rho}{2} K$ from the left. We can then rewrite (34) as

$$K \left(I - \frac{1}{2} 1_{N+1} s^T - \frac{1}{2} y s^T\right) K L K \alpha^\rho = 0.$$

(35)
or as $K \Lambda K \gamma^p = 0$ after substituting $s_x, s_y,$ and $y$ with $\frac{1}{N} 1_s \gamma N - \frac{1}{N} 1_s \gamma N 1_y^*$, $\frac{1}{N} 1_s \gamma N - 1_y 1_y^*$, and $1_s \gamma N - 1_y 1_y^*$, respectively.

(ii) When $K \Lambda K \gamma^p = 0$, we have
$$K \left(2K \gamma^p - s_x^2 \gamma N K \gamma^p - y s_x^T \gamma N \right) = 0$$
(36)
after multiplying by $\frac{1}{2}$ since $L = \frac{1}{2} 1_s \gamma N 1_y^* - \frac{1}{2} 1_y \gamma N 1_y^*$ as implied in (35). With the definitions of $a^x, b^x,$ and $\zeta^x$, we can rewrite (36) as $a^x K \Lambda K \gamma^p + b^x K \Lambda 1_s \gamma N = K \gamma N y$, or as $a^x K \Lambda K \gamma^p + b^x 1_s \gamma N = y$. Combining the results above, we have Theorem 1.

Appendix B. Proof of Theorem 2

Noting that $V = s_x s_x^T$ and $K$ is symmetric, we can rewrite (11) into
$$\eta^x = \arg \min_{(4^p, \gamma^p)} \left( \Phi (X) \eta^x \right)^T \Phi (X) \eta^x$$
(37)
since $(\Phi (X) \eta^x)^T \Phi (X) \eta^x = (\eta^x)^T K \eta^x = 1$ and $\zeta^x = (\eta^x)^T K \eta^x - (\eta^x)^T K \eta x K \gamma^p \eta^x$. In addition, the constraint (12) can be expressed as
$$\frac{2}{\zeta^x} \Phi (X) \eta^x + b^x 1_s \gamma N = y$$
(38)
using (14) and (16), and the relationship $a^x = \frac{2}{\zeta^x}$. Letting $\gamma^p = \frac{2}{\zeta^x} \Phi (X) \eta^x$, the problem (37) and constraint (38) can be transformed into
$$\gamma^p = \arg \min \frac{1}{4} (\gamma^p)^T \gamma^p$$
(39)
and
$$\Phi (X) \gamma^p + b^x 1_s \gamma N = y$$
(40)
respectively, where $\gamma^p$ is related to $\eta^x$ as
$$\gamma^p = \frac{2}{\zeta^x} \Phi (X) \eta^x$$
(41)

Let us solve the problem (39) under the constraint (40) via Lagrange technique. Set the Lagrange function as
$$L^p = \frac{1}{4} (\gamma^p)^T \gamma^p + (\gamma^p)^T (\Phi (X) \gamma^p + b^x 1_s \gamma N - y)$$
(42)
where the $N \times 1$ vector $a^x$ is the Lagrange multiplier. By setting the partial derivatives of $L^p$ with respect to $\gamma^p$ and $b^x$ equal to zero, we get
$$\gamma^p = -2 \Phi (X) a^x$$
(43)
from $\frac{\partial L^p}{\partial b^x} = 0$, and
$$b^x 1_s \gamma N = 0$$
(44)
from $\frac{\partial L^p}{\partial b^x} = 0$. Then, substituting (43) and (44) into (42), $L^p$ can be rewritten as $L^p = - (\gamma^p)^T (K a^x + y)$. Eventually, we have the optimization problem
$$a^x = \arg \min \gamma^p (\gamma^p)^T (K a^x + y)$$
(45)
subject to (44), where the vectors $\gamma^x$ and $a^x$ have the relationship
$$\gamma^x = -2 \Phi (X) a^x$$
(46)
from (43). Following the steps similar to those shown in (42)–(45) with Lagrange technique, we can reformulate the problem (45) under the constraint (44) into
$$\gamma^x = \arg \min \left( (\gamma^x)^T I a^x + y \gamma^x - \frac{2}{N} 1_s \gamma N a^x \gamma N K a^x \right)$$
$$\frac{1}{N} 1_s \gamma N a^x \gamma N y$$
(47)
By setting the derivative of the objective function in (47) with respect to $a^x$ equal to zero, we have $(K - \frac{1}{N} 1_s \gamma N K - \frac{1}{N} 1_s \gamma N a^x = 0, \gamma N y)$. If we combine (41) and (46), we can show that $$(\gamma^x)^T K a^x = 1$$
(48)
from (47). Combining (48) and (49), we have Theorem 2.

Appendix C

Theorem 3. Assume that $K$ is of full-rank. Then, among the $N$ eigenvalues of $A_K$, $N - 1$ eigenvalues are positive and one eigenvalue is negative: the negative eigenvalue is $-1/2 I_s \gamma N K I_s \gamma N$ with the corresponding normalized eigenvector $\frac{1}{\sqrt{N}} 1_s \gamma N$.

Proof. (i) Let us first show that rank$(\Phi (X) \Phi (X)) = N - 1$, where $\Phi (X) = \Phi (X) \Phi (X)$ and $S = I_s \gamma N - \frac{1}{N} 1_s \gamma N$. When the $N \times N$ matrix $K = \Phi (X) \Phi (X)$ is of full-rank, we have rank$(\Phi (X)) = \text{rank}(\Phi (X) \Phi (X)) = N$. In addition, $\text{rank}(\Phi (X)) \leq \text{rank}(S) = N - 1$ and $\text{rank}(\Phi (X)) \geq \text{rank}(\Phi (X) \Phi (X)) = N$. Thus, $\text{rank}(\Phi (X) \Phi (X)) = N - 1$.

(ii) Let us next show that, among the $N$ eigenvalues of $\Phi (X) \Phi (X)$, $N - 1$ eigenvalues are positive and the remaining one eigenvalue is zero with the corresponding normalized eigenvector $\frac{1}{\sqrt{N}} 1_s \gamma N$. First, $\Phi (X) \Phi (X)$ is positive semi-definite since $z^T \Phi (X) \Phi (X) z = \| \Phi (X) z \|^2 \geq 0$. Thus, the eigenvalues of $\Phi (X) \Phi (X)$ are all non-negative. Recollecting that $\text{rank}(\Phi (X) \Phi (X)) = N - 1$, it is straightforward that $\Phi (X) \Phi (X)$ has $N - 1$ positive eigenvalues $\lambda_1^x, \lambda_2^x, ..., \lambda_{N-1}^x$ with the remaining eigenvalue $\lambda_0^x = 0$. Second, let $V_i^x$ be the normalized eigenvector of $\Phi (X) \Phi (X)$ corresponding to $\lambda_i^x$. It is clear that $V_i^x = \frac{1}{\sqrt{N}} 1_s \gamma N$ since $\Phi (X) \Phi (X) \frac{1}{\sqrt{N}} 1_s \gamma N = \frac{1}{\sqrt{N}} \text{rank}(\Phi (X) \Phi (X)) S 1_s, 1_s = 0$ and $\frac{1}{\sqrt{N}} 1_s \gamma N 1_s = 1$.

(iii) Since the normalized eigenvectors of a symmetric matrix are orthonormal, we have $(V_i^x)^T V_j^x = \delta_{ij}$ for $i, j = 1, 2, ..., N$ where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$. Now, from the definition of $\Phi (X)$ and (19) we can rewrite $\Phi (X) \Phi (X)$ as
$$\Phi (X) \Phi (X) = \left( I_N - \frac{1}{N} 1_s \gamma N \right) \Phi (X) \Phi (X) \left( I_N - \frac{1}{N} 1_s \gamma N \right)$$
$$= \Phi (X) \Phi (X) - \frac{1}{N} \text{rank}(\Phi (X) \Phi (X)) S 1_s, 1_s$$
$$+ \frac{1}{N} \text{rank}(\Phi (X) \Phi (X)) S 1_s, 1_s - \frac{1}{N} \text{rank}(\Phi (X) \Phi (X)) S 1_s, 1_s = 0$$
(50)
Since $V_i^x = \frac{1}{\sqrt{N}} 1_s \gamma N$, we can rewrite (50) further as $\lambda_0^x = \Phi (X) \Phi (X) - \frac{1}{N} \text{rank}(\Phi (X) \Phi (X)) V_i^x (V_i^x)^T$. 

Thus, from
\[ A_{\xi} v_i = \phi(X) \phi(x) v_i^\phi - \left( \frac{1}{N} I_{s \times N} K_{1_{s \times 1}} \right) v_i^{\phi} \]
we can conclude that \( v_i^\phi, v_i^\phi, \ldots, v_i^\phi, \) and \( v_i^\phi \) are the eigenvectors of \( A_{\xi} \) corresponding to the eigenvalues \( \lambda_1^\phi, \lambda_2^\phi, \ldots, \lambda_{s-1}^\phi, \) and \( -\frac{1}{N} I_{s \times N} K_{1_{s \times 1}} \) respectively.

Combining the results in (i)–(iii), we have Theorem 3.

References


[22] H. Alt, Square rooting is as difficult as multiplication, Computing 21 (3) (1979) 211–232.


H.-K. Min received the B.S.E. and M.S.E. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2004 and 2006, respectively. He is currently working toward the Ph.D. degree at KAIST. His research interests include pattern recognition, machine learning theory, and statistical signal processing.

Yuxi Hou received the B.S.E. and M.S.E. degrees in electrical engineering from Xi’an Jiaotong University, Xi’an, China, in 2001 and 2005, respectively, and the Ph.D. degree in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2012. His research interests include Gaussian mixture model, pattern recognition, feature extraction, and function regression.

Sangwoo Park received the B.S. degree in physics from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2014. He is currently working toward the M.S.E. degree at KAIST. His research interests include information theory, machine learning theory, and statistical signal processing.

Ickcho Song received the B.S.E. and M.S.E. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1982 and 1984, respectively, and the M.S.E. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 1985 and 1987, respectively. In 1988, he joined the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, where he is presently a Professor. He is a Fellow of IEEE and IET and a co-author of Advanced Theory of Signal Detection (Springer, 2002).