Kernel Nonparametric Weighted Feature Extraction for Hyperspectral Image Classification

Bor-Chen Kuo, Cheng-Hsuan Li, and Jinn-Min Yang

***Abstract*—In recent years, many studies show that kernel methods are computationally efficient, robust, and stable for pattern analysis. Many kernel-based classifiers were designed and applied to classify remote-sensed data, and some results show that kernel-based classifiers have satisfying performances. Many studies about hyperspectral image classification also show that nonparametric weighted feature extraction (NWFE) is a powerful tool for extracting hyperspectral image features. However, NWFE is still based on linear transformation. In this paper, the kernel method is applied to extend NWFE to kernel-based NWFE (KNWFE). The new KNWFE possesses the advantages of both linear and nonlinear transformation, and the experimental results show that KNWFE outperforms NWFE, decision-boundary feature extraction, independent component analysis, kernel-based principal component analysis, and generalized discriminant analysis.**

***Index Terms*—Feature extraction, image classification.**

I. Introduction

In RECENT years, many studies [1]–[7] show that kernel methods are computationally efficient, robust, and stable for pattern analysis. Many kernel-based classifiers were designed and applied to classify remote-sensed data, and some results show that kernel-based classifiers have satisfying performances [5]–[7]. The main idea of kernel methods is to map the input data from the original space to a convenient feature space by a nonlinear mapping where inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly, and linear relations are sought among the images of the data items in the feature space.

Some studies [8]–[13] have also shown that nonparametric weighted feature extraction (NWFE) [14] is powerful in reducing dimensionality of hyperspectral image data. In this paper, we try to combine the advantages of kernel method and NWFE and develop a kernel-based NWFE (KNWFE) for hyperspectral image classification.

This paper is organized as follows. Introduction to the kernel trick is discussed in Section II. The reviews of some unsupervised and supervised feature extractions and their kernel versions are introduced in Section III. KNWFE is proposed in Section IV. In order to reduce the influence of the singularity of the kernel matrix, the eigenvalue resolution is discussed in Section V. For evaluating the performance of the proposed method on a real hyperspectral image data, experiment is designed in Section VI and experimental results are also reported in this section. Section VII contains comments and conclusions.

Manuscript received April 22, 2008; revised July 3, 2008 and September 4, 2008. Current version published March 27, 2009.

B.-C. Kuo is with the Graduate Institute of Educational Measurement and Statistics, National Taichung University, Taichung 40306, Taiwan (e-mail: kbc@mail.ntcu.edu.tw).

C.-H. Li is with the Graduate Institute of Educational Measurement and Statistics, National Taichung University, Taichung 40306, Taiwan, and also with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan.

J.-M. Yang is with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2008.2008308

II. Kernel Trick

It is easier for classification if pixels are more sparsely distributed. Generally speaking, images with high dimensionality (the number of spectral bands) potentially have better class separability. The strategy of kernel method is to embed the data from original space R^d into a feature space H, a Hilbert space with higher dimensionality, where more effective hyperplanes for classification are expected to exist in this space than in the original space. From this, we can compute the inner product of samples in the feature space directly from the original data items using a kernel function. This is based on the fact that any kernel function $\kappa : R^d \times R^d \to R$ satisfies the Mercer's theorem [4], i.e., there is a feature map ϕ into a Hilbert space H such that $k(x,z) = \langle \phi(x), \phi(z) \rangle$, where $x, z \in X$, if and only if it is a symmetric function for which the matrices $K = [\kappa(x_i, x_j)]_{1 \le i,j \le N}$ formed by restriction to any finite subset $\{x_1, \ldots, x_N\}$ of the space X are positive semidefinite.

Suppose $x_1^{(i)}, \ldots, x_{N_i}^{(i)} \in R^d$ are the samples in class i, $i = 1, \ldots, L$, and $N = N_1 + \cdots + N_L$. Let $X_i^{\mathrm{T}} = [\phi(x_1^{(i)}), \ldots, \phi(x_{N_i}^{(i)})]$ and $X^{\mathrm{T}} = [X_1^{\mathrm{T}}, \ldots, X_L^{\mathrm{T}}]$, then the kernel matrix $K = [\kappa(x_i, x_j)]_{1 \le i,j \le N}$ with respect to κ on samples is XX^{T}, i.e., $K = XX^{\mathrm{T}}$.

The following are some popular kernels.

1) *Linear kernel*:

$$\kappa(x,z) = \langle x, z \rangle.$$

2) *Polynomial kernel*:

$$\kappa(x,z) = (\langle x, z \rangle + 1)^r, \qquad r \in Z^+.$$

3) *Gaussian radial-basis-function (RBF) kernel*:

$$\kappa(x,z) = \exp\left(\frac{-\|x-z\|^2}{2\sigma^2}\right), \qquad \sigma \in R - \{0\}$$

where x and z are the samples in R^d.

It is worth stressing here that the size of the kernel matrix is $N \times N$ and contains in each position K_{ij}, the information of distance among all possible pixel pairs (x_i and x_j) measured

with a suitable kernel function k, fulfilling the characterization of kernels, and if we use the linear kernel, then the feature mapping ϕ is an identity map, i.e., ϕ is linear. Otherwise, the feature mapping can be nonlinear. One important idea for using kernel method is without knowing the nonlinear mapping explicitly.

III. Reviews of Some Feature Extraction Methods

In this section, some well-known supervised and unsupervised feature-extraction methods and their kernel versions are reviewed.

A. Unsupervised Feature Extraction

Unsupervised feature-extraction methods do not require any prior knowledge for training data [15]. One typical method is the "principal component analysis" (PCA), a multivariate technique that allows us to reduce an original set of correlated observed variables into a smaller set [16], [17]. The purpose of PCA is to reduce dimensionality according to what percentage of the overall variance can be captured. The kernel-based PCA (KPCA) is to find the directions by performing the PCA in the kernel feature space [3].

Independent component analysis (ICA) is a statistical technique for separating the independent signals from overlapping signals [18]. ICA is related to PCA but is more powerful and capable of finding the underlying factors or sources when the principal-component approach fails. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non-Gaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA [18].

Further techniques, based on image-processing approaches, have been proposed in [12] and [19] by combining PCA/ICA and morphological transformations in the context of the classification of hyperspectral images of urban areas.

B. Supervised Feature Extraction

Supervised feature extraction directly takes into account the training information that is available for a given supervised-classification problem [15]. Linear discriminant analysis (LDA) is often used for dimension reduction in classification problems. It is also called the parametric feature extraction method in [20], since LDA uses the mean vector and covariance matrix of each class. Usually, within-class, between-class, and mixture scatter matrices are used to formulate the criterion of class separability. A kernel-based LDA was proposed by [2] in 2000 and called generalized discriminant analysis (GDA) using a kernel approach. There are three drawbacks of LDA. One is that it works well only if the distributions of classes are normal-like distributions. When the distributions of classes are nonnormal-like or multimodal mixture distributions, the performance of LDA is not satisfactory. The second disadvantage of LDA is that the rank of the between-class scatter matrix is less than or equal to $L-1$, so assuming sufficient observations and the rank of within-class scatter matrix is $v \leq d$, then only v features can be extracted. The third limitation is that, if the within-class covariance is singular, which often occurs in high-dimensional problems, LDA will have a poor performance on classification.

Lee and Landgrebe [21] proposed the "decision-boundary feature extraction" (DBFE) that can extract both discriminately informative features and discriminately redundant features from the decision boundary. The approach uses the training samples directly to determine the location of the decision boundary and employs information about the decision hypersurfaces associated with a given classifier to define an intrinsic dimensionality for the classification problem. Then, the corresponding optimal linear mapping can be obtained.

NWFE was proposed by [14] to solve the problems which LDA suffered. It also absorbs the idea of DBFE for determining the location of the decision boundary by training samples. The main ideals of NWFE are putting different weights on every sample to compute the "weighted means" and compute the distance between samples and their weighted means as their "closeness" to boundary, then defining nonparametric between-class and within-class scatter matrices which put large weights on the samples close to the boundary and deemphasize those samples far from the boundary.

The experimental results of [12] and [19] show that NWFE outperforms LDA and DBFE under morphological approach. In [9] and [13], the authors suggest replacing DBFE by NWFE to obtain more effective features. Other papers show NWFE outperforms LDA, approximated pairwise accuracy criterion linear dimension reduction, nonparametric discriminant analysis [14], and DBFE [8] in remote-sensing data sets.

IV. KNWFE

Although NWFE can improve the problems of LDA, the feature transformation of NWFE is still linear. To extend NWFE for nonlinear situation, a KNWFE is proposed in this section.

The between-class scatter matrix S_b^{NW} and the within-class scatter matrix S_w^{NW} of NWFE in original space R^d are

$$S_b^{\mathrm{NW}} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right) \times \left(x_\ell^{(i)} - M_j\left(x_\ell^{(i)}\right)\right)^{\mathrm{T}}$$

$$S_w^{\mathrm{NW}} = \sum_{i=1}^{L} P_i \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,i)}}{N_i} \left(x_\ell^{(i)} - M_i\left(x_\ell^{(i)}\right)\right) \times \left(x_\ell^{(i)} - M_i\left(x_\ell^{(i)}\right)\right)^{\mathrm{T}}$$

where the scatter-matrix weight $\lambda_\ell^{(i,j)}$ is defined by

$$\lambda_\ell^{(i,j)} = \frac{\mathrm{dist}\left(x_\ell^{(i)}, M_j\left(x_\ell^{(i)}\right)\right)^{-1}}{\sum_{t=1}^{N_i} \mathrm{dist}\left(x_t^{(i)}, M_j\left(x_t^{(i)}\right)\right)^{-1}}.$$

$M_j(x_\ell^{(i)}) = \sum_{k=1}^{N_j} w_{\ell k}^{(i,j)} x_k^{(j)}$ denotes the weighted mean with respect to $x_\ell^{(i)}$ in class j, $\mathrm{dist}(A, B)$ represents the distance between A and B, and

$$w_{\ell k}^{(i,j)} = \frac{\mathrm{dist}\left(x_\ell^{(i)}, x_k^{(j)}\right)^{-1}}{\sum_{t=1}^{N_j} \mathrm{dist}\left(x_\ell^{(i)}, x_t^{(j)}\right)^{-1}}.$$

The between-class scatter matrix S_b^{KNW} and the within-class scatter matrix S_w^{KNW} of KNWFE in the feature space H are

$$S_b^{\mathrm{KNW}} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j\neq i}}^{L} \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}}$$

$$S_w^{\mathrm{KNW}} = \sum_{i=1}^{L} P_i \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,i)}}{N_i} \left(\phi\left(x_\ell^{(i)}\right) - M_i\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_i\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}}$$

where the scatter-matrix weight $\lambda_\ell^{(i,j)}$ is defined by

$$\lambda_\ell^{(i,j)} = \frac{\mathrm{dist}\left(\phi\left(x_\ell^{(i)}\right), M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{-1}}{\sum_{t=1}^{N_i} \mathrm{dist}\left(\phi\left(x_t^{(i)}\right), M_j\left(\phi\left(x_t^{(i)}\right)\right)\right)^{-1}}.$$

$M_j(\phi(x_\ell^{(i)})) = \sum_{k=1}^{N_j} w_{\ell k}^{(i,j)} \phi(x_k^{(j)})$ denotes the weighted mean with respect to $\phi(x_\ell^{(i)})$ in class j and

$$w_{\ell k}^{(i,j)} = \frac{\mathrm{dist}\left(\phi\left(x_\ell^{(i)}\right), \phi\left(x_k^{(j)}\right)\right)^{-1}}{\sum_{t=1}^{N_j} \mathrm{dist}\left(\phi\left(x_\ell^{(i)}\right), \phi\left(x_t^{(j)}\right)\right)^{-1}}.$$

The following lemmas and theorems show that every part in S_b^{KNW} and S_w^{KNW} can be evaluated by the elements in kernel matrix K or the kernel function κ. The proofs of the following lemmas, theorems, and corollary will be given in the Appendix.

Lemma 1: The weighted mean in class j with respect to $\phi(x_\ell^{(i)})$ is

$$M_j\left(\phi\left(x_\ell^{(i)}\right)\right) = X_j^{\mathrm{T}} \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix}.$$

Lemma 2: Let $K^{(i,j)} = X_i X_j^{\mathrm{T}}$, the (i, j) block of the kernel matrix K, and

$$W^{(i,j)} = \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{N_i 1}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix}.$$

The scatter matrix weight $\lambda_\ell^{(i,j)}$ is shown at the bottom of the page.

The following theorem shows that S_b^{KNW} and S_w^{KNW} can be evaluated by matrices multiplication.

Theorem 3: Suppose that

$$\Lambda^{(i,j)} = \mathrm{diag}\left\{\frac{\lambda_1^{(i,j)}}{N_i}, \ldots, \frac{\lambda_{N_i}^{(i,j)}}{N_i}\right\}$$

$$W^{(i,j)} = \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & & \vdots \\ w_{N_i 1}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix}.$$

The within-class scatter matrix S_w^{KNW} becomes

$$S_w^{\mathrm{KNW}} = X^{\mathrm{T}} W X$$

where $W = W_1 - W_2 - W_2^{\mathrm{T}} + W_3$, and

$$W_1 = \mathrm{diag}\left\{P_1 \Lambda^{(1,1)}, \ldots, P_L \Lambda^{(L,L)}\right\}$$

$$W_2 = \mathrm{diag}\left\{P_1 \Lambda^{(1,1)} W^{(1,1)}, \ldots, P_L \Lambda^{(L,L)} W^{(L,L)}\right\}$$

$$W_3 = \mathrm{diag}\left\{P_1 W^{(1,1)^{\mathrm{T}}} \Lambda^{(1,1)} W^{(1,1)}, \ldots, P_L W^{(L,L)^{\mathrm{T}}} \Lambda^{(L,L)} W^{(L,L)}\right\}.$$

The between-class scatter matrix S_b^{KNW} becomes

$$S_b^{\mathrm{KNW}} = X^{\mathrm{T}}(B - W)X$$

where $B = B_1 - B_2 - B_2^{\mathrm{T}} + B_3$, and

$$B_1 = \mathrm{diag}\left\{P_1 \sum_{j=1}^{L} \Lambda^{(1,j)}, \ldots, P_L \sum_{j=1}^{L} \Lambda^{(L,j)}\right\}$$

$$B_2 = \begin{bmatrix} P_1 \Lambda^{(1,1)} W^{(1,1)} & \cdots & P_1 \Lambda^{(1,L)} W^{(1,L)} \\ \vdots & & \vdots \\ P_L \Lambda^{(L,1)} W^{(L,1)} & \cdots & P_L \Lambda^{(L,L)} W^{(L,L)} \end{bmatrix}$$

$$B_3 = \sum_{i=1}^{L} P_i \mathrm{diag}\left\{W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} W^{(i,1)}, \ldots, W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} W^{(i,L)}\right\}.$$

$$\lambda_\ell^{(i,j)} = \frac{\left[K_{\ell\ell}^{(i,i)} + \left(W^{(i,j)} K^{(j,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{\ell\ell} - 2\left(K^{(i,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{\ell\ell}\right]^{-1/2}}{\sum_{t=1}^{N_i} \left[K_{tt}^{(i,i)} + \left(W^{(i,j)} K^{(j,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{tt} - 2\left(K^{(i,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{tt}\right]^{-1/2}}$$

From earlier theorem, the linear transformation $A \in R^{d\times p}$ of KNWFE in the feature space H can be obtained by solving the following problem:

$$A = \arg\max_{A} \operatorname{tr}\left((A^{\mathrm{T}}X^{\mathrm{T}}WXA)^{-1}A^{\mathrm{T}}X^{\mathrm{T}}(B-W)XA\right).$$

Since our reduced space is a subspace spanned by all training samples in H, we can express A in dual form, i.e.,

$$A = X^{\mathrm{T}}\widetilde{A}$$

where $\tilde{A} \in R^{N\times p}$. Then, the earlier optimal problem is equivalent to the following optimization:

$$\widetilde{A} = \arg\max_{\widetilde{A}} \operatorname{tr}\left((\widetilde{A}^{\mathrm{T}}KWK\widetilde{A})^{-1}\widetilde{A}^{\mathrm{T}}K(B-W)K\widetilde{A}\right).$$

From the earlier discussion, we can have the following corollary, and its proof is stated in the Appendix.

Corollary 4: If the rank of $X^{\mathrm{T}} = [x_1, \ldots, x_N]$ is d, then under the Fisher criterion, KNWFE with linear kernel is the same as NWFE.

V. Eigenvalue Resolution of KNWFE

Note that the kernel matrix K is an $N \times N$ matrix. Hence, if $N > d$, then K is a positive semidefinite matrix, i.e., K is singular. In order to reduce the influence of the singularity of K, it is necessary to do a decomposition (eigendecomposition, QR decomposition, etc.) of K. Let us use the eigendecomposition of the kernel matrix K, a symmetric matrix, i.e.,

$$K = P\Gamma P^{\mathrm{T}}$$

where Γ is the diagonal matrix of all eigenvalues of K and P is a orthogonal matrix. Substituting K in the Fishier criterion $\operatorname{tr}((\tilde{A}^{\mathrm{T}}KWK\tilde{A})^{-1}\tilde{A}^{\mathrm{T}}K(B-W)K\tilde{A})$, we have

$$\operatorname{tr}\left((\widetilde{A}^{\mathrm{T}}P\Gamma P^{\mathrm{T}}WP\Gamma P^{\mathrm{T}}\widetilde{A})^{-1}\widetilde{A}^{\mathrm{T}}P\Gamma P^{\mathrm{T}}(B-W)P\Gamma P^{\mathrm{T}}\widetilde{A}\right).$$

Let us proceed to variable modification using U such that

$$U = P^{\mathrm{T}}\widetilde{A}.$$

Then, the problem

$$\widetilde{A} = \arg\max_{\widetilde{A}} \operatorname{tr}\left((\widetilde{A}^{\mathrm{T}}KWK\widetilde{A})^{-1}\widetilde{A}^{\mathrm{T}}K(B-W)K\widetilde{A}\right)$$

is equivalent to

$$U = \arg\max_{U} \operatorname{tr}\left((U^{\mathrm{T}}(\Gamma P^{\mathrm{T}}WP\Gamma)U)^{-1} \times U^{\mathrm{T}}\left(\Gamma P^{T}(B-W)P\Gamma\right)U\right).$$

Therefore, the process of finding $\tilde{A}$ can be divided into two parts. The first step is to find the eigenvalues λ and eigenvectors u for the following generalized eigenvalue problem:

$$\left(\Gamma P^{\mathrm{T}}(B-W)P\Gamma\right)u = \lambda(\Gamma P^{\mathrm{T}}WP\Gamma)u.$$

The extracted p features are the p eigenvectors with the largest p eigenvalues of the following matrix:

$$(\Gamma P^{\mathrm{T}}WP\Gamma)^{-1}\left(\Gamma P^{\mathrm{T}}(B-W)P\Gamma\right).$$

Here, the $\Gamma P^{\mathrm{T}}WP\Gamma$ is regularized by

$$0.5(\Gamma P^{\mathrm{T}}WP\Gamma) + 0.5\mathrm{diag}(\Gamma P^{\mathrm{T}}WP\Gamma).$$

One can remark that, using the linear kernel, the corresponding feature mapping is identity mapping. This is stated in the Corollary 4, the NWFE is a special case of KNWFE. However, using other kernel functions, the only one we can control is the kernel matrix K, not the samples in the original space. Hence, the process of the eigenvalue resolution is necessary. Hence, the regularized within-class matrix is not the same as that of NWFE. In our experiment, the algorithm of KNWFE must be done after eigenvalue resolution, except the algorithm of KNWFE with linear kernel.

Hence, we can build U. After U is calculated, we compute $\tilde{A}$ by

$$\widetilde{A} = PU.$$

We finally compute the projection of a point $\phi(z)$ by

$$A^{\mathrm{T}}\phi(z) = \widetilde{A}^{\mathrm{T}}X\phi(z) = \widetilde{A}^{\mathrm{T}}\begin{bmatrix}\phi(x_1)^{\mathrm{T}}\\ \vdots \\ \phi(x_N)^{\mathrm{T}}\end{bmatrix}\phi(z) = \widetilde{A}^{\mathrm{T}}\begin{bmatrix}\kappa(x_1,z)\\ \vdots \\ \kappa(x_N,z)\end{bmatrix}.$$

KNWFE procedure is summarized in the following steps.

1) Compute the distances between each pair of sample points in the feature space and form the distance matrix.
2) Compute $w_{\ell k}^{(i,j)}$ using the distance matrix and get the matrix $W^{(i,j)}$.
3) Compute the scatter-matrix weights $\lambda_{\ell}^{(i,j)}$ and get the matrix $\Lambda^{(i,j)}$.
4) By Theorem 2, compute B and W and, hence, $S_b^{\mathrm{KNW}} = X^{\mathrm{T}}(B-W)X$, $S_w^{\mathrm{KNW}} = X^{\mathrm{T}}WX$.
5) Do eigendecomposition for K, i.e., $K = P\Gamma P^{\mathrm{T}}$.
6) Extract features by solving $(\Gamma P^{\mathrm{T}}(B-W)P\Gamma)u = \lambda(\Gamma P^{\mathrm{T}}WP\Gamma)u$.
7) Compute $\tilde{A}$ and the projections of sample points.

VI. Experiments

A. Data Set

In this paper, for investigating the influences of training sample sizes to the dimension, three distinct cases, $N_i = 20 < N < d$ (case 1), $N_i = 150 < d < N$ (case 2), and $d < N_i = 300 < N$ (case 3), will be discussed. Due to these sample size constraints, some of the classes in selected hyperspectral images for the experiment are used. The MultiSpec [22] was used to select training and testing samples (100 testing samples per class) in our experiments which is the same method in [12], [15], and [22].

In this paper, three real data sets are applied to compare the performances of KNWFE and other famous feature-extraction methods. They are the Indian Pine, a mixed forest/agricultural

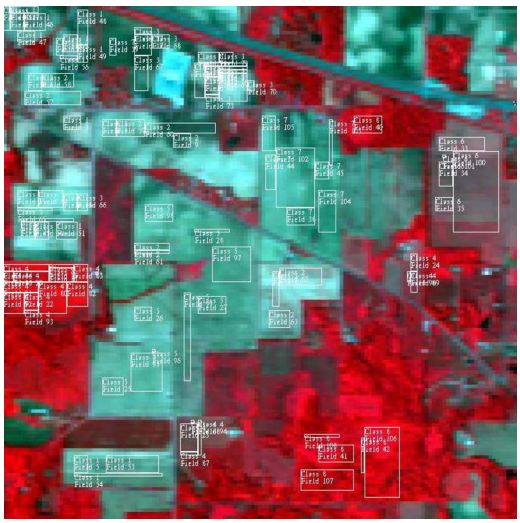


Fig. 1. Simulated grayscale IR image of the Indian Pine Site data set.

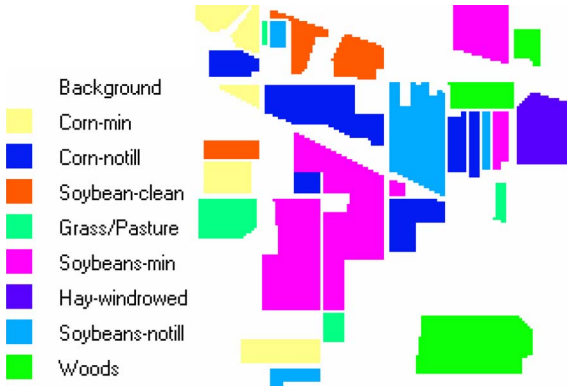


Fig. 2. Ground truth of the area with eight classes.

site in Indiana [22], Kennedy Space Center (KSC), FL [23], and the Washington, DC Mall hyperspectral image [22] as an urban site. The first two of these data sets were gathered by a sensor known as the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). The Indian Pine image, mounted from an aircraft flown at 65 000 ft altitude and operated by the NASA/Jet Propulsion Laboratory, with the size of 145 × 145 pixels has 220 spectral bands measuring approximately 20 m across on the ground. The simulated grayscale IR image and the ground-truth map are shown in Figs. 1 and 2, respectively. Since the size of samples in some classes are too small to retain enough disjoint samples for training and testing, only eight classes, Corn-min, Corn-notill, Soybean-clean, Grass/Pasture, Soybeans-min, Hay-windrowed, Soybeans-notill, and Woods, were selected for the experiments.

The KSC data set was acquired over the KSC by the NASA AVIRIS instrument on March 23, 1996. AVIRIS acquires data in 224 bands of 10-nm width with center wavelengths from 400–2500 nm. The KSC data, acquired from an altitude of approximately 20 km, have a spatial resolution of 18 m. After removing water absorption and low SNR bands, 176 bands were used for the analysis [23]. Due to the sample-size constraints, seven classes, Scrub, Graminoid marsh, Spartina marsh, Cattail marsh, Salt marsh, Mud flats, and Water, are selected.

The third data set, Washington, DC Mall from an urban area, is a Hyperspectral Digital Imagery Collection Experiment airborne hyperspectral data flight line over the Washington, DC Mall. Two hundred and ten bands were collected in the 0.4–2.4 m region of the visible and infrared spectrum. Some water-absorption channels are discarded, resulting in 191 channels. The data set is available in the student CD-ROM of [22]. There are seven information classes in the Washington, DC data, roofs, roads, trails, grass, trees, water, and shadows, in the data set.

B. Methods

The purpose of this experiment is to compare the multiclass-classification performances using maximum likelihood (ML), 1-nearest neighbor (1NN), and soft-margin support vector machine (SVM) classifiers with the original hyperspectral data and NWFE, DBFE, KPCA, GDA, and KNWFE features. Three kernel-based feature extractions, KPCA, GDA, and KNWFE, and one kernel-based classifier, soft-margin SVM, are applied with two types of kernels, polynomial kernels of degree d ($d = 1$ and 2) and the RBF kernel with σ. Here, we use the five-fold cross validation to find the best σ within the given set $\{2^5, 2^6, \ldots, 2^{20}\}$ of parameters.

In this paper, PRTools [24] and LIBSVM [25] were used to implement 1NN and soft-margin SVM classifiers, respectively. DBFE, ICA, and KPCA were implemented by MultiSpec [26]–[28], and GDA was implemented by [2]. For the soft-margin SVM classifier, there is a parameter C to control the tradeoff between the margin and the size of the slack variables. Again, we use the five-fold cross validation to find the best C within the given set $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$ (suggested by [25]) of parameters.

C. Results

Tables I–III display the classification accuracies of testing data in cases 1, 2, and 3, respectively. In those tables, "poly-d" indicates the polynomial kernel when degree d is used, and "RBF" means the RBF kernel is applied. Note that the best accuracy of each data set (in column) is highlighted in shadow cell. From Tables I–III, we can find the following conditions.

1) In the small sample size situation (case 1, $N_i = 20 < N < d$), the highest accuracies among all methods are 0.8 (KNWFE–RBF with SVM–RBF classifier), 0.937 (KNWFE–RBF with 1NN classifier), and 0.840 (KNWFE–RBF with 1NN classifier) in Indian Pine, KSC, and Washington, DC data sets, respectively.
2) In the case 2 ($N_i = 150 < d < N$), the highest accuracies among all methods are 0.903 (KNWFE–poly1 with ML classifier), 0.971 (KNWFE–RBF with 1NN classifier), and 0.851 (KNWFE–RBF with SVM–poly1 and SVM–RBF classifiers) in Indian Pine, KSC, and Washington, DC data sets, respectively.
3) In the case 3 ($d < N_i = 300 < N$), the highest accuracies among all methods are 0.954 (KNWFE–poly1 with ML classifier), 0.987 (KNWFE–RBF with 1NN classifier), and 0.959 (KNWFE–poly1 with ML classifier) in Indian Pine, KSC, and Washington, DC data sets, respectively.

TABLE I
HIGHEST ACCURACIES USING EXTRACTED FEATURES (PUTTING IN BRACKETS) APPLIED TO THREE DIFFERENT DATA SETS ($N_i = 20$, CASE 1)

Feature Extraction	Classifier	Data Set		
		Indian Pine	KSC	DC
None	ML	0.125	0.143	0.246
	1NN	0.681	0.854	0.331
	SVM-poly1	0.604	0.861	0.263
	SVM-poly2	0.569	0.864	0.263
	SVM-RBF	0.623	0.627	0.245
DBFE	ML	0.720(7)	0.914(8)	0.328(10)
	1NN	0.711(12)	0.811(6)	0.508(10)
	SVM-poly1	0.673(10)	0.861(6)	0.496(10)
	SVM-poly2	0.595(10)	0.788(15)	0.496(10)
	SVM-RBF	0.674(10)	0.804(15)	0.498(12)
ICA	ML	0.773(9)	0.911(9)	0.651(14)
	1NN	0.666(9)	0.676(4)	0.747(14)
	SVM-poly1	0.672(7)	0.630(13)	0.7(13)
	SVM-poly2	0.681(13)	0.703(12)	0.638(13)
	SVM-RBF	0.517(5)	0.553(8)	0.69(11)
NWFE	ML	0.783(10)	0.926(6)	0.724(13)
	1NN	0.769(8)	0.913(9)	0.822(4)
	SVM-poly1	0.614(8)	0.873(9)	0.745(9)
	SVM-poly2	0.580(6)	0.876(9)	0.69(7)
	SVM-RBF	0.641(6)	0.931(8)	0.728(10)
KPCA-poly1	ML	0.750(14)	0.914(8)	0.7(4)
	1NN	0.681(15)	0.853(11)	0.7(4)
	SVM-poly1	0.744(15)	0.797(7)	0.713(9)
	SVM-poly2	0.715(11)	0.799(7)	0.762(12)
	SVM-RBF	0.736(15)	0.894(14)	0.735(6)
KPCA-poly2	ML	0.766(14)	0.854(9)	0.709(12)
	1NN	0.663(14)	0.840(14)	0.757(12)
	SVM-poly1	0.783(15)	0.639(11)	0.681(11)
	SVM-poly2	0.759(12)	0.631(13)	0.734(11)
	SVM-RBF	0.775(9)	0.539(6)	0.704(12)
KPCA-RBF	ML	0.684(7)	0.920(12)	0.7(4)
	1NN	0.681(15)	0.840(13)	0.777(11)
	SVM-poly1	0.644(15)	0.793(12)	0.715(10)
	SVM-poly2	0.678(15)	0.804(14)	0.741(10)
	SVM-RBF	0.771(15)	0.903(12)	0.73(8)
GDA-poly1	ML	0.664(7)	0.931(5)	0.143(1)
	1NN	0.689(7)	0.933(6)	0.659(6)
	SVM-poly1	0.665(7)	0.893(6)	0.434(5)
	SVM-poly2	0.495(6)	0.911(6)	0.367(5)
	SVM-RBF	0.660(6)	0.904(6)	0.434(5)
GDA-poly2	ML	0.654(6)	0.524(5)	0.142(1)
	1NN	0.744(7)	0.884(6)	0.801(6)
	SVM-poly1	0.734(7)	0.866(6)	0.748(6)
	SVM-poly2	0.673(6)	0.841(5)	0.462(4)
	SVM-RBF	0.738(7)	0.880(6)	0.748(6)
GDA-RBF	ML	0.547(4)	0.664(6)	0.7(8)
	1NN	0.647(7)	0.930(6)	0.777(11)
	SVM-poly1	0.546(7)	0.840(6)	0.712(9)
	SVM-poly2	0.494(7)	0.926(6)	0.741(9)
	SVM-RBF	0.666(7)	0.840(6)	0.73(7)
KNWFE-poly1	ML	0.654(7)	0.920(5)	0.544(1)
	1NN	0.761(9)	**0.937(15)**	0.703(6)
	SVM-poly1	0.586(9)	0.903(14)	0.596(7)
	SVM-poly2	0.615(10)	0.844(7)	0.434(7)
	SVM-RBF	0.658(8)	0.763(7)	0.617(7)
KNWFE-poly2	ML	0.743(6)	0.843(11)	0.760(4)
	1NN	0.764(6)	0.894(12)	0.831(9)
	SVM-poly1	0.608(8)	0.917(12)	0.796(9)
	SVM-poly2	0.579(12)	0.823(15)	0.7343(9)
	SVM-RBF	0.783(11)	0.724(12)	0.809(8)
KNWFE-RBF	ML	0.695(6)	0.917(5)	0.689(5)
	1NN	0.778(10)	**0.937(15)**	**0.840(12)**
	SVM-poly1	0.649(14)	0.854(15)	0.833(8)
	SVM-poly2	0.735(10)	0.881(6)	0.831(9)
	SVM-RBF	**0.800(12)**	0.899(5)	0.731(14)

TABLE II
HIGHEST ACCURACIES USING EXTRACTED FEATURES (PUTTING IN BRACKETS) APPLIED TO THREE DIFFERENT DATA SETS ($N_i = 150$, CASE 2)

Feature Extraction	Classifier	Data Set		
		Indian Pine	KSC	DC
None	ML	0.125	0.143	0.400
	1NN	0.758	0.926	0.445
	SVM-poly1	0.751	0.870	0.228
	SVM-poly2	0.705	0.881	0.360
	SVM-RBF	0.781	0.956	0.375
DBFE	ML	0.853(11)	0.911(10)	0.71(15)
	1NN	0.849(8)	0.948(7)	0.828(15)
	SVM-poly1	0.839(10)	0.883(11)	0.142(1)
	SVM-poly2	0.773(10)	0.934(15)	0.644(15)
	SVM-RBF	0.830(7)	0.894(15)	0.382(1)
ICA	ML	0.888(9)	0.837(15)	0.768(15)
	1NN	0.741(8)	0.894(15)	0.77(15)
	SVM-poly1	0.680(8)	0.846(15)	0.761(5)
	SVM-poly2	0.680(8)	0.823(15)	0.744(15)
	SVM-RBF	0.518(10)	0.841(15)	0.738(6)
NWFE	ML	0.876(8)	0.899(5)	0.767(15)
	1NN	0.856(14)	0.960(15)	0.833(5)
	SVM-poly1	0.805(10)	0.884(15)	0.728(15)
	SVM-poly2	0.805(10)	0.889(15)	0.847(11)
	SVM-RBF	0.815(10)	0.933(15)	0.782(11)
KPCA-poly1	ML	0.879(7)	0.839(15)	0.77(13)
	1NN	0.750(15)	0.924(15)	0.768(13)
	SVM-poly1	0.759(15)	0.763(14)	0.72(12)
	SVM-poly2	0.823(13)	0.776(15)	0.788(11)
	SVM-RBF	0.754(14)	0.587(11)	0.727(10)
KPCA-poly2	ML	0.878(15)	0.527(14)	0.76(15)
	1NN	0.746(14)	0.787(15)	0.753(14)
	SVM-poly1	0.749(15)	0.564(15)	0.774(9)
	SVM-poly2	0.788(15)	0.609(15)	0.768(13)
	SVM-RBF	0.750(15)	0.467(15)	0.777(9)
KPCA-RBF	ML	0.816(15)	0.879(7)	0.77(13)
	1NN	0.734(15)	0.750(15)	0.775(12)
	SVM-poly1	0.746(15)	0.593(13)	0.702(11)
	SVM-poly2	0.774(15)	0.620(15)	0.721(8)
	SVM-RBF	0.835(14)	0.799(15)	0.788(12)
GDA-poly1	ML	0.865(7)	0.964(6)	0.832(6)
	1NN	0.860(7)	0.967(6)	0.837(6)
	SVM-poly1	0.836(6)	0.936(6)	0.717(2)
	SVM-poly2	0.825(6)	0.939(6)	0.477(4)
	SVM-RBF	0.826(7)	0.944(6)	0.715(2)
GDA-poly2	ML	0.883(7)	0.943(6)	0.142(1)
	1NN	0.880(7)	0.960(6)	0.841(5)
	SVM-poly1	0.856(7)	0.954(6)	0.817(6)
	SVM-poly2	0.839(7)	0.956(6)	0.827(6)
	SVM-RBF	0.840(7)	0.954(6)	0.838(6)
GDA-RBF	ML	0.891(7)	0.946(6)	0.841(6)
	1NN	0.891(7)	0.953(6)	0.837(6)
	SVM-poly1	0.855(6)	0.851(6)	0.797(6)
	SVM-poly2	0.796(7)	0.881(6)	0.802(6)
	SVM-RBF	0.870(7)	0.924(6)	0.68(4)
KNWFE-poly1	ML	**0.903(15)**	0.846(6)	0.840(13)
	1NN	0.886(12)	0.954(15)	0.839(10)
	SVM-poly1	0.721(12)	0.881(15)	0.751(14)
	SVM-poly2	0.773(11)	0.833(15)	0.840(9)
	SVM-RBF	0.866(8)	0.947(15)	0.829(13)
KNWFE-poly2	ML	0.894(10)	0.614(15)	0.789(6)
	1NN	0.878(15)	0.881(15)	0.839(8)
	SVM-poly1	0.701(13)	0.789(15)	0.836 (15)
	SVM-poly2	0.630(12)	0.524(14)	0.831(6)
	SVM-RBF	0.893(10)	0.841(11)	0.826(8)
KNWFE-RBF	ML	0.886(13)	0.960(9)	0.847(9)
	1NN	0.885(14)	**0.971(15)**	0.827(5)
	SVM-poly1	0.834(11)	0.931(6)	**0.851(6)**
	SVM-poly2	0.769(8)	0.934(6)	0.849(7)
	SVM-RBF	0.871(15)	0.961(8)	**0.851(12)**

TABLE III
HIGHEST ACCURACIES USING EXTRACTED FEATURES (PUTTING IN BRACKETS) APPLIED TO THREE DIFFERENT DATA SETS ($N_i = 300$, CASE 3)

Feature Extraction	Classifier	Data Set		
		Indian Pine	KSC	DC
None	ML	0.716	0.884	0.446
	1NN	0.833	0.947	0.547
	SVM-poly1	0.846	0.871	0.207
	SVM-poly2	0.844	0.877	0.470
	SVM-RBF	0.814	0.973	0.471
DBFE	ML	0.929(15)	0.917(15)	0.936(15)
	1NN	0.906(9)	0.984(10)	0.957(5)
	SVM-poly1	0.844(8)	0.883(15)	0.91(15)
	SVM-poly2	0.869(7)	0.937(15)	0.88(8)
	SVM-RBF	0.874(10)	0.930(14)	0.952(13)
ICA	ML	0.912(9)	0.524(15)	0.854(15)
	1NN	0.836(10)	0.850(15)	0.874(15)
	SVM-poly1	0.646(9)	0.734(15)	0.817(15)
	SVM-poly2	0.735(8)	0.734(15)	0.76(15)
	SVM-RBF	0.580(13)	0.730(15)	0.804(15)
NWFE	ML	0.923(14)	0.916(6)	0.933(15)
	1NN	0.915(9)	0.949(15)	0.934(6)
	SVM-poly1	0.849(11)	0.853(13)	0.807(15)
	SVM-poly2	0.895(8)	0.856(12)	0.877(15)
	SVM-RBF	0.629(8)	0.919(13)	0.905(14)
KPCA-poly1	ML	0.900(13)	0.530(8)	0.7(4)
	1NN	0.838(6)	0.814(15)	0.777(11)
	SVM-poly1	0.756(15)	0.703(14)	0.712(9)
	SVM-poly2	0.792(15)	0.726(14)	0.762(11)
	SVM-RBF	0.786(15)	0.634(14)	0.734(5)
KPCA-poly2	ML	0.900(15)	0.520(11)	0.708(12)
	1NN	0.837(15)	0.687(15)	0.757(12)
	SVM-poly1	0.772(14)	0.454(3)	0.684(12)
	SVM-poly2	0.801(15)	0.354(5)	0.734(9)
	SVM-RBF	0.799(14)	0.391(6)	0.704(12)
KPCA-RBF	ML	0.899(15)	0.886(10)	0.848(14)
	1NN	0.823(9)	0.929(14)	0.854(15)
	SVM-poly1	0.774(11)	0.773(14)	0.731(15)
	SVM-poly2	0.648(15)	0.769(6)	0.764(15)
	SVM-RBF	0.640(3)	0.943(15)	0.837(14)
GDA-poly1	ML	0.918(7)	0.984(6)	0.143(1)
	1NN	0.886(7)	0.979(6)	0.659(6)
	SVM-poly1	0.888(7)	0.973(6)	0.434(5)
	SVM-poly2	0.885(7)	0.971(6)	0.367(5)
	SVM-RBF	0.896(7)	0.980(6)	0.434(5)
GDA-poly2	ML	0.926(7)	0.956(6)	0.142(1)
	1NN	0.926(7)	0.966(6)	0.801(6)
	SVM-poly1	0.933(7)	0.949(6)	0.748(6)
	SVM-poly2	0.931(7)	0.947(6)	0.581(5)
	SVM-RBF	0.926(7)	0.957(6)	0.748(6)
GDA-RBF	ML	0.918(7)	0.974(6)	0.94(5)
	1NN	0.886(7)	0.973(6)	0.938(6)
	SVM-poly1	0.918(7)	0.937(6)	0.901(6)
	SVM-poly2	0.924(7)	0.976(6)	0.864(4)
	SVM-RBF	0.894(7)	0.930(6)	0.854(5)
KNWFE-poly1	ML	**0.954(11)**	0.800(14)	**0.959(15)**
	1NN	0.936(10)	0.970(9)	0.953(14)
	SVM-poly1	0.949(13)	0.897(9)	0.884(15)
	SVM-poly2	0.931(12)	0.877(15)	0.723(13)
	SVM-RBF	0.905(12)	0.941(15)	0.930(14)
KNWFE-poly2	ML	0.941(11)	0.550(11)	0.916(7)
	1NN	0.934(11)	0.861(15)	0.946(6)
	SVM-poly1	0.894(14)	0.707(11)	0.911(7)
	SVM-poly2	0.776(12)	0.593(15)	0.899(6)
	SVM-RBF	0.939(14)	0.884(15)	0.926(15)
KNWFE-RBF	ML	0.944(11)	0.980(7)	0.951(15)
	1NN	0.941(14)	**0.987(11)**	0.947(15)
	SVM-poly1	0.916(15)	0.967(7)	0.923(9)
	SVM-poly2	0.891(8)	0.960(9)	0.916(8)
	SVM-RBF	0.929(14)	0.979(15)	0.923(9)

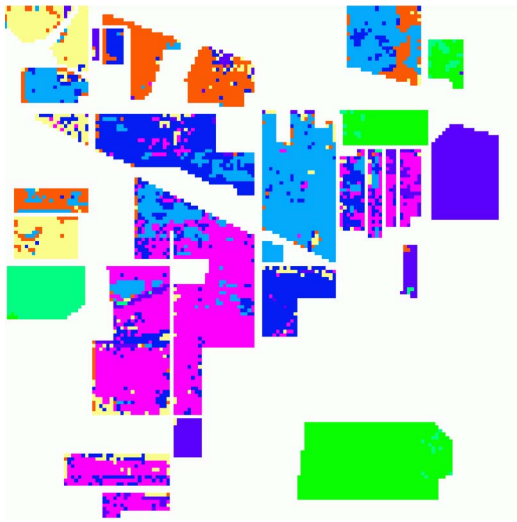


Fig. 3. Thematic map resulting from the classification of the area of Fig. 1 by SVM–poly1 classifier without feature extraction ($N_i = 300$, $p = 220$).

4) We can observe that the best accuracy in each column all occurs when applying KNWFE. It means that KNWFE outperforms other feature extraction under the best condition of all feature extractions.
5) The performance of applying only single classifier is worse than the others "FE + classifier" methods. Many studies show that kernel-based classifiers are robust and insensitive about hyperspectral images and may outperform traditional statistical classifiers. However, in the experimental results of this paper, many "FE + classifier" methods outperform single SVM.
6) SVM classifier did not perform well in Washington, DC data set, but if using feature extraction as a preprocessing process, then SVM has a great improvement.
7) For KSC data set, KNWFE–RBF with 1NN classifier is the best choice through the three cases. However, for the other two data sets, the best combination of feature extraction and classifier is not consistent.

Due to the length of this paper, we choose the well-known Indian Pine Site image as an example, and only some classified images are shown for comparison. The best classification mechanisms under case 3 ($N_i = 300$) and seven feature-extraction conditions (none, DBFE, ICA, NWFE, KPCA, GDA, and KNWFE) are selected to generate the classified images.

Figs. 1 and 2 are the Indian Pine Site image and the ground truth, respectively. Figs. 3–9 are the thematic map resulting from the classification of the area of Fig. 1 using the SVM–poly1 classifier (without feature extraction) and applying DBFE, ICA, NWFE, KPCA, GDA, and KNWFE with different classifiers which are the combination with highest classification accuracy. Here, p is the number of features extracted by these methods with the highest accuracies in the corresponding methods. For instance, KNWFE–poly1 with ML classifier has highest accuracy among all combinations of KNWFE and classifiers, and Fig. 9 is the classification result of this combination. From Figs. 3–9, we can find that KNWFE outperforms other feature-extraction methods in "Corn-min," "Corn-notill," and "Soybean-notill" parts.

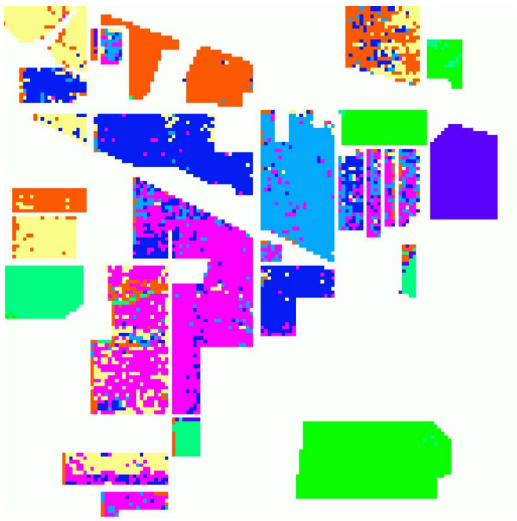


Fig. 4. Thematic map resulting from the classification of the area of Fig. 1 by DBEF with ML classifier ($N_i = 300$, $p = 15$).

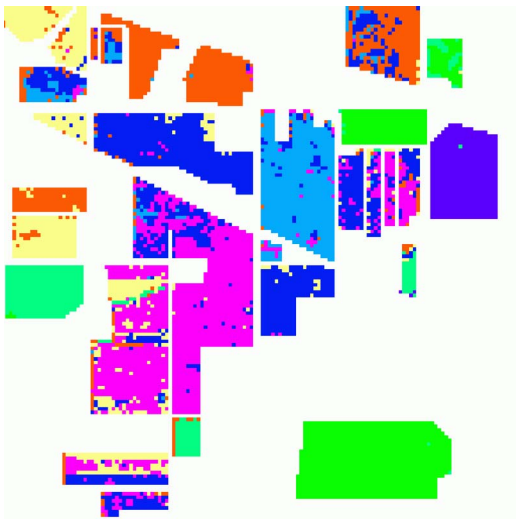


Fig. 5. Thematic map resulting from the classification of the area of Fig. 1 by ICA with ML classifier ($N_i = 300$, $p = 9$).

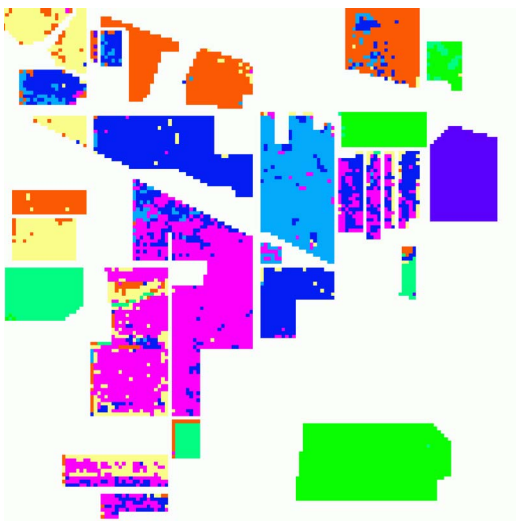


Fig. 6. Thematic map resulting from the classification of the area of Fig. 1 by NWEF with ML classifier ($N_i = 300$, $p = 14$).

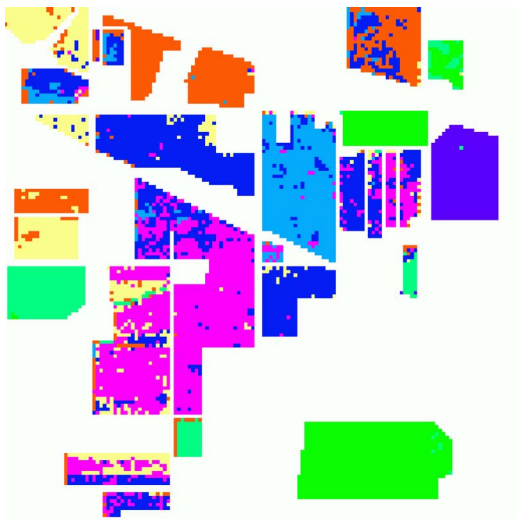


Fig. 7. Thematic map resulting from the classification of the area of Fig. 1 by KPCA–poly1 with ML classifier ($N_i = 300$, $p = 13$).

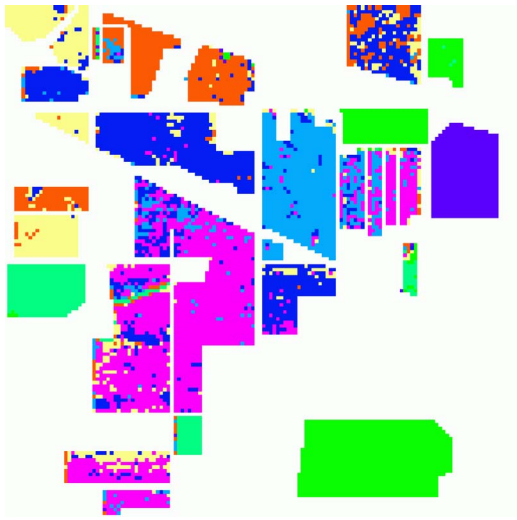


Fig. 8. Thematic map resulting from the classification of the area of Fig. 1 by GDA–poly2 with SVM–poly1 classifier ($N_i = 300$, $p = 7$).

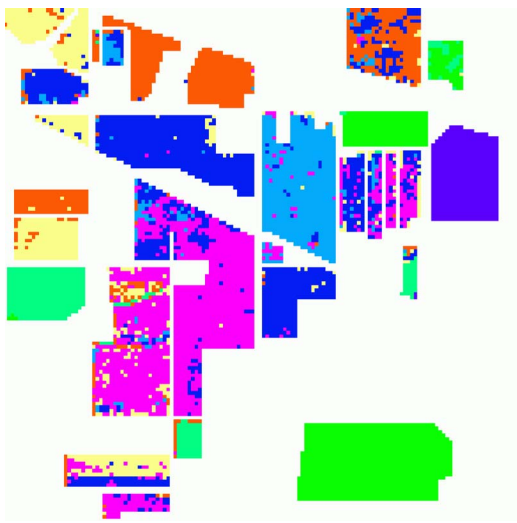


Fig. 9. Thematic map resulting from the classification of the area of Fig. 1 by KNWFE–poly1 with ML classifier ($N_i = 300$, $p = 11$).

TABLE IV
BEST COMBINATION IN DIFFERENT NUMBER OF FEATURES ($N_i = 20$, CASE 1)

# of features	Feature Extraction	Classifier	Overall Accuracy
1	KPCA-RBF	ML	0.545
2	NWFE	1NN	0.683
3	KNWFE-RBF	1NN	0.673
4	ICA	ML	0.738
5	KPCA-poly2	ML	0.741
6	KNWFE-poly2	1NN	0.764
7	ICA	ML	0.772
8	KNWFE-poly2	SVM-RBF	0.775
9	KPCA-poly2	SVM-RBF	0.775
10	KNWFE-RBF	SVM-RBF	0.783
	NWFE	ML	
11	KNWFE- poly2	SVM-RBF	0.783
	NWFE	ML	
12	KNWFE-RBF	SVM-RBF	0.800
13	KNWFE-RBF	SVM-RBF	0.781
14	KNWFE-RBF	SVM-RBF	0.809
15	KNWFE-RBF	SVM-RBF	0.809

From Tables I–III, we can also find that different classifiers are needed to achieve the highest results with the KNWFE. For example, classifying data set 1 (in Table I), the highest accuracy is based on the KNWFE–RBF followed by a SVM–RBF. In contrast to this, on data set 2, KNWFE–RBF followed by a 1NN classifier achieved the highest result.

We think that the reason of occurring this situation is that the distributions of data sets may be very different. Some may be simple or single mode, and the others may be mixture and complex. From many related papers [39], [41], [42] and the experimental results of this paper, we can find that there is no the "best" classifier which can have the highest accuracy on "every data set." Similarly, we also think that it is hard to find the best combination of feature extraction + classifier which can have the highest accuracy on "every data set." The experimental results (Tables I–III) also show that DBFE, KPCA, ICA, and GDA also have this unstable condition. Therefore, we think that this unstable situation of KNWFE is normal and acceptable. Under this unstable situation, KNWFE with suitable classifier can achieve the best classification accuracy.

Due to the length of this paper, we also choose the Indian Pine Site image as an example to explore the performances of different methods if the same number of features is extracted. Tables IV–VI demonstrate the feature extraction and classifier combination with highest classification accuracies of Indian Pine data set when different number of extracted features is applied.

From Tables IV–VI, we have the following findings.

1) KNWFE outperforms the other feature extraction methods when the number of extracted features is greater or equal than eight. When the number of extracted features is less than eight, the best feature extraction is hard to define.
2) As the number of training samples increases (Table VI), KNWFE can have a better performance even in small number of features condition.

TABLE V
BEST COMBINATION IN DIFFERENT NUMBER OF FEATURES ($N_i = 150$, CASE 2)

# of features	Feature Extraction	Classifier	Overall Accuracy
1	GDA-RBF	ML	0.623
2	GDA-RBF	ML	0.680
3	KNWFE-poly1	ML	0.755
4	NWFE	ML	0.838
5	NWFE	ML	0.856
6	GDA-RBF	ML	0.866
7	GDA-RBF	ML, 1NN	0.891
8	KNWFE-poly1	ML	0.894
9	KNWFE-poly1	ML	0.890
10	KNWFE-poly1	ML	0.898
11	KNWFE-poly1	ML	0.896
12	KNWFE-poly1	ML	0.895
13	KNWFE-poly1	ML	0.903
14	KNWFE-poly1	ML	0.899
15	KNWFE-poly1	ML	0.903

TABLE VI
BEST COMBINATION IN DIFFERENT NUMBER OF FEATURES ($N_i = 300$, CASE 3)

# of features	Feature Extraction	Classifier	Overall Accuracy
1	KNWFE-RBF	SVM-RBF	0.519
2	NWFE	1NN	0.658
3	KNWFE-poly1	1NN	0.823
4	KNWFE-RBF	1NN	0.868
5	KNWFE-poly1	ML	0.903
6	KNWFE-RBF	ML	0.919
7	KNWFE-RBF	ML	0.936
8	KNWFE-poly1	ML	0.941
9	KNWFE-RBF	ML	0.944
10	KNWFE-poly1	ML	0.951
11	KNWFE-poly1	ML	0.954
12	KNWFE-poly1	ML	0.953
13	KNWFE-poly1	ML	0.951
14	KNWFE-poly1	ML	0.950
15	KNWFE-poly1	ML	0.945

3) From Indian Pine data set part of Table I, the highest accuracy among all combinations except KNWFE family is 0.783 achieved by NWFE + ML with ten features. However, Table IV shows that KNWFE–RBF + SVM–RBF can obtain the same accuracy with the same number of features and achieve better performance as the number of features increases.
4) Table IV shows that KNWFE–RBF with SVM–RBF can have a better performance in ill-posed condition (case 1). However, when the number of training samples increases (cases 2 and 3), KNWFE–poly1 with ML can be a better choice.

In a more insightful observation, we can see that, in Fig. 10 for case 1, the training and testing samples of three classes of Indian Pine data set are projected into the feature space formed by the first two eigenvectors of the six feature-extraction methods. The distributions of KNWFE-projected data are more separable as compared with those of DBFE, ICA, NWFE, KPCA, and GDA. The training and testing data of ICA and

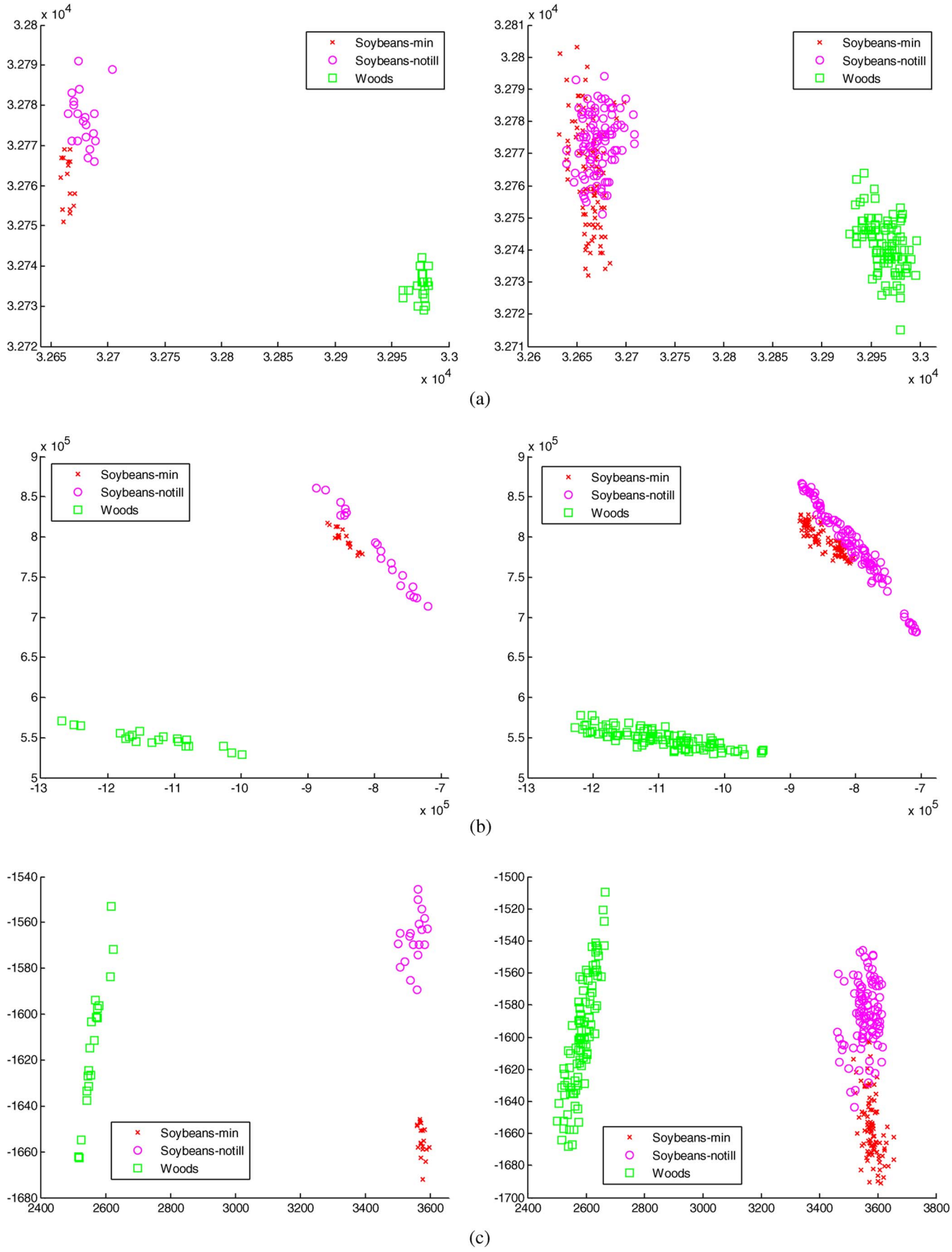
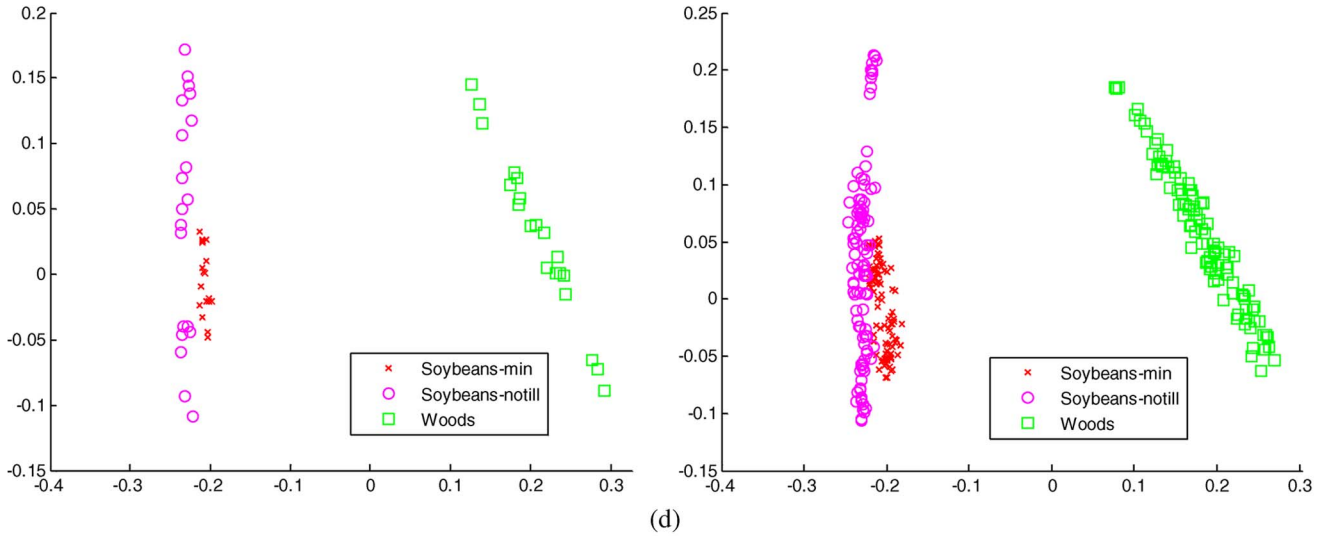
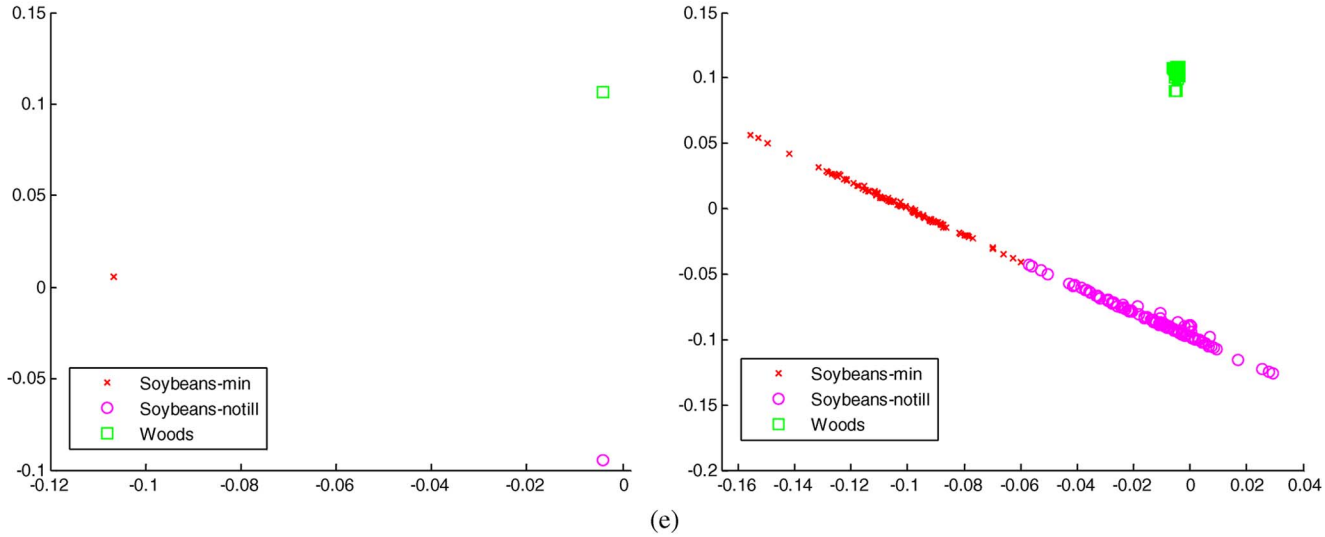


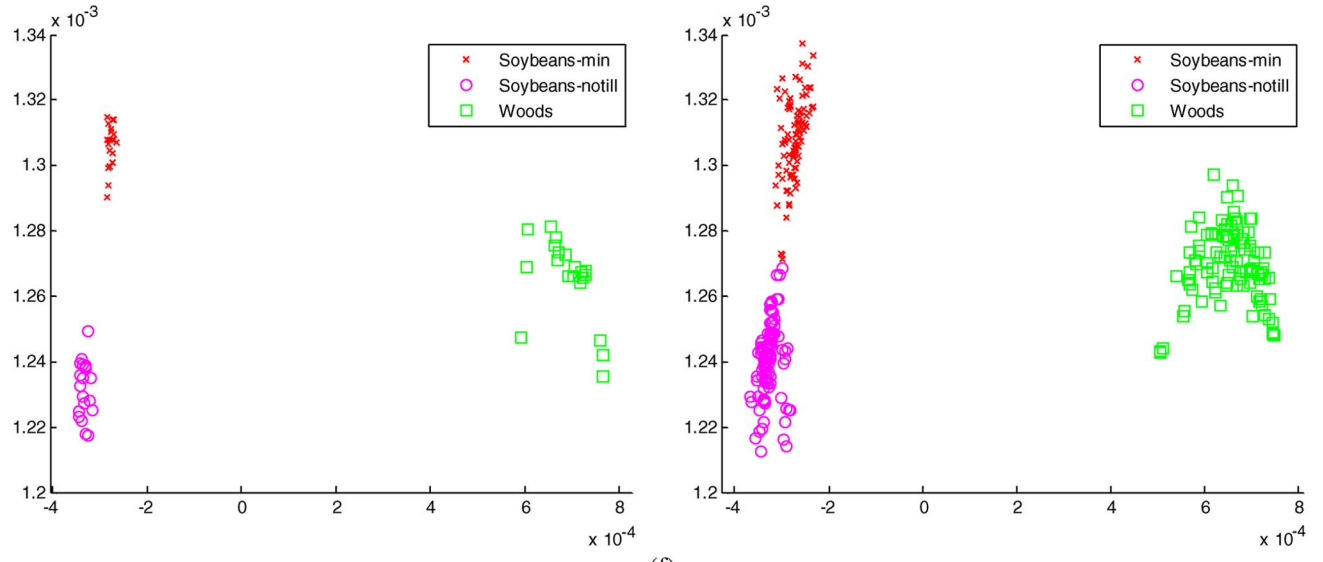
Fig. 10. Distributions of training samples and testing samples for "Soybeans-min," "Soybeans-notill," and "Woods" of Fig. 2 using the first two significant features obtained from the methods. (a) DBFE. (b) ICA. (c) NWFE.



(d)



(e)



(f)

Fig. 10. (*Continued*.) Distributions of training samples and testing samples for "Soybeans-min," "Soybeans-notill," and "Woods" of Fig. 2 using the first two significant features obtained from the methods. (d) KPCA–RBF. (e) GDA–RBF. (f) KNWFE–RBF. In each method, the left scatter plot is for training data and the right one is for testing data ($N_i = 20$, case 1).

KPCA are projected in a parallel form; therefore, it is difficult for classification. Fig. 10(e) shows that GDA has an overfitting problem.

VII. Conclusion

In this paper, a new KNWFE was proposed, and we have analyzed and compared KNWFE and other kernel-based methods both theoretically and experimentally. From a theoretical point of view, NWFE is a special case of KNWFE with linear kernel. With KNWFE, more kernels can be used to obtain better classification results. The experimental results of three hyperspectral images show that KNWFE can have the highest classification accuracy under three training-sample-size conditions. Experimental results also show that the performance of KNWFE is not consistently better than those of the other methods under small number of features and training-sample condition. However, when the number of training samples is large enough, KNWFE outperforms other methods under both small and large numbers of feature conditions. Because the proposed method is kernel based, it is disadvantaged in that it is time consuming when the training sample size is large.

In the next steps, we will consider the use of composite kernels and the development of semisupervised-version KNWFE. Furthermore, we will try to develop a feature extraction which can achieve the best result no matter what kind of classifier is applied.

Appendix
Proofs of Lemmas, Theorems, and Corollary

Lemma 1: The weighted mean in class j with respect to $\phi(x_\ell^{(i)})$ is

$$M_j\left(\phi\left(x_\ell^{(i)}\right)\right) = X_j^{\mathrm{T}} \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix}.$$

Proof:

$$\begin{aligned} M_j\left(\phi\left(x_\ell^{(i)}\right)\right) &= \sum_{k=1}^{N_j} w_{\ell k}^{(i,j)} \phi\left(x_k^{(j)}\right) \\ &= \left[\phi\left(x_1^{(j)}\right), \ldots, \phi\left(x_{N_j}^{(j)}\right)\right] \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix} \\ &= X_j^{\mathrm{T}} \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix}. \end{aligned}$$

■

Lemma 2: Let $K^{(i,j)} = X_i X_j^{\mathrm{T}}$, the (i,j) block of the kernel matrix K and

$$W^{(i,j)} = \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{N_i 1}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix}.$$

The scatter-matrix weight $\lambda_\ell^{(i,j)}$ is shown at the bottom of the page.

Proof: First, compute the numerator of $\lambda_\ell^{(i,j)}$

$$\begin{aligned} &\mathrm{dist}\left(\phi\left(x_\ell^{(i)}\right), M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^2 \\ &\quad = \left\|\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right\|^2 \\ &\quad = \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}} \\ &\qquad \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \\ &\quad = \phi\left(x_\ell^{(i)}\right)^{\mathrm{T}} \phi\left(x_\ell^{(i)}\right) \\ &\qquad + M_j\left(\phi\left(x_\ell^{(i)}\right)\right)^{\mathrm{T}} M_j\left(\phi\left(x_\ell^{(i)}\right)\right) \\ &\qquad - 2\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}} M_j\left(\phi\left(x_\ell^{(i)}\right)\right) \\ &\quad = K_{\ell\ell}^{(i,i)} + \left[w_{\ell 1}^{(i,j)}, \ldots, w_{\ell N_j}^{(i,j)}\right] X_j X_j^{\mathrm{T}} \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix} \\ &\qquad - 2\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}} X_j^{\mathrm{T}} \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix} \\ &\quad = K_{\ell\ell}^{(i,i)} + \left(W^{(i,j)} K^{(j,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{\ell\ell} \\ &\qquad - 2\left(K^{(i,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{\ell\ell}. \end{aligned}$$

By the same way

$$\begin{aligned} &\mathrm{dist}\left(\phi\left(x_t^{(i)}\right), M_j\left(\phi\left(x_t^{(i)}\right)\right)\right)^2 \\ &\quad = K_{tt}^{(i,i)} + \left(W^{(i,j)} K^{(j,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{tt} \\ &\qquad - 2\left(K^{(i,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{tt}, \qquad t = 1, \ldots, N_i. \end{aligned}$$

Hence, we have $\lambda_\ell^{(i,j)}$ shown at the bottom of the page. ■

$$\lambda_\ell^{(i,j)} = \frac{\left[K_{\ell\ell}^{(i,i)} + \left(W^{(i,j)} K^{(j,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{\ell\ell} - 2\left(K^{(i,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{\ell\ell}\right]^{-1/2}}{\sum_{t=1}^{N_i} \left[K_{tt}^{(i,i)} + \left(W^{(i,j)} K^{(j,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{tt} - 2\left(K^{(i,j)} \left(W^{(i,j)}\right)^{\mathrm{T}}\right)_{tt}\right]^{-1/2}}$$

Theorem 3: Suppose that

$$\Lambda^{(i,j)} = \operatorname{diag}\left\{\frac{\lambda_1^{(i,j)}}{N_i}, \ldots, \frac{\lambda_{N_i}^{(i,j)}}{N_i}\right\}$$

$$W^{(i,j)} = \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & & \vdots \\ w_{N_i1}^{(i,j)} & \cdots & w_{N_iN_j}^{(i,j)} \end{bmatrix}.$$

The within-class scatter matrix S_w^{KNW} becomes

$$S_w^{\mathrm{KNW}} = X^{\mathrm{T}} W X$$

where $W = W_1 + W_2 - W_3 - W_3^{\mathrm{T}}$ and

$$\begin{aligned} W_1 &= \operatorname{diag}\left\{P_1\Lambda^{(1,1)}, \ldots, P_L\Lambda^{(L,L)}\right\} \\ W_2 &= \operatorname{diag}\left\{P_1 W^{(1,1)^{\mathrm{T}}}\Lambda^{(1,1)}W^{(1,1)}, \ldots, \right. \\ &\qquad \left. P_L W^{(L,L)^{\mathrm{T}}}\Lambda^{(L,L)}W^{(L,L)}\right\} \\ W_3 &= \operatorname{diag}\left\{P_1\Lambda^{(1,1)}W^{(1,1)}, \ldots, P_L\Lambda^{(L,L)}W^{(L,L)}\right\}. \end{aligned}$$

The between-class scatter matrix S_b^{KNW} becomes

$$S_b^{\mathrm{KNW}} = X^{\mathrm{T}}(B - W)X$$

where $B = B_1 + B_2 - B_3 - B_3^{\mathrm{T}}$, and

$$\begin{aligned} B_1 &= \operatorname{diag}\left\{P_1\sum_{j=1}^{L}\Lambda^{(1,j)}, \ldots, P_L\sum_{j=1}^{L}\Lambda^{(L,j)}\right\} \\ B_2 &= \sum_{i=1}^{L} P_i \operatorname{diag}\left\{W^{(i,1)^{\mathrm{T}}}\Lambda^{(i,1)}W^{(i,1)}, \ldots, W^{(i,L)^{\mathrm{T}}}\right. \\ &\qquad \left.\Lambda^{(i,L)}W^{(i,L)}\right\} \\ B_3 &= \begin{bmatrix} P_1\Lambda^{(1,1)}W^{(1,1)} & \cdots & P_1\Lambda^{(1,L)}W^{(1,L)} \\ \vdots & \ddots & \vdots \\ P_L\Lambda^{(L,1)}W^{(L,1)} & \cdots & P_L\Lambda^{(L,L)}W^{(L,L)} \end{bmatrix}. \end{aligned}$$

Proof: For every class i, the summation

$$\sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}} \qquad i = 1, \ldots, L, \quad j = 1, \ldots, L$$

in scatter matrices S_b^{KNW}, and S_w^{KNW} can be evaluated by

$$\begin{aligned} &\sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \\ &\quad\times\left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}} \\ &\quad= \sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\left(\phi\left(x_\ell^{(i)}\right)\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}}\right. \\ &\qquad\qquad + M_j\left(\phi\left(x_\ell^{(i)}\right)\right)M_j\left(\phi\left(x_\ell^{(i)}\right)\right)^{\mathrm{T}} \\ &\qquad\qquad - \phi\left(x_\ell^{(i)}\right)M_j\left(\phi\left(x_\ell^{(i)}\right)\right)^{\mathrm{T}} \\ &\qquad\qquad \left. - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}}\right) \\ &\quad= \sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\phi\left(x_\ell^{(i)}\right)\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}} \\ &\qquad + \sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}X_j^{\mathrm{T}}\begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)}\end{bmatrix}\left[w_{\ell 1}^{(i,j)}, \ldots, w_{\ell N_j}^{(i,j)}\right]X_j \\ &\qquad - \sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\phi\left(x_\ell^{(i)}\right)\left[w_{\ell 1}^{(i,j)}, \ldots, w_{\ell N_j}^{(i,j)}\right]X_j \\ &\qquad - \sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}X_j^{\mathrm{T}}\begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)}\end{bmatrix}\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}}. \end{aligned}$$

Now, we compute earlier expression term by term. We rewrite the first term in the matrix form by the following computation:

$$\begin{aligned} &\sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\phi\left(x_\ell^{(i)}\right)\phi\left(x_\ell^{(i)}\right)^{\mathrm{T}} \\ &\quad= \left[\phi\left(x_1^{(i)}\right), \ldots, \phi\left(x_{N_i}^{(i)}\right)\right]\begin{bmatrix} \frac{\lambda_1^{(i,j)}}{N_i} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_{N_i}^{(i,j)}}{N_i}\end{bmatrix}\begin{bmatrix} \phi\left(x_1^{(i)}\right)^{\mathrm{T}} \\ \vdots \\ \phi\left(x_{N_i}^{(i)}\right)^{\mathrm{T}}\end{bmatrix} \\ &\quad= X_i^{\mathrm{T}}\begin{bmatrix} \frac{\lambda_1^{(i,j)}}{N_i} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_{N_i}^{(i,j)}}{N_i}\end{bmatrix}X_i = X_i^{\mathrm{T}}\Lambda^{(i,j)}X_i. \end{aligned}$$

The second term can be computed by the following process:

$$\begin{aligned} &\sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}X_j^{\mathrm{T}}\begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)}\end{bmatrix}\left[w_{\ell 1}^{(i,j)}, \ldots, w_{\ell N_j}^{(i,j)}\right]X_j \\ &\quad= X_j^{\mathrm{T}}\sum_{\ell=1}^{N_i}\frac{\lambda_\ell^{(i,j)}}{N_i}\begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)}\end{bmatrix}\left[w_{\ell 1}^{(i,j)}, \ldots, w_{\ell N_j}^{(i,j)}\right]X_j \end{aligned}$$

$$= X_j^{\mathrm{T}} \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{N_i 1}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{1N_j}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix} \begin{bmatrix} \frac{\lambda_1^{(i,j)}}{N_i} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_{N_i}^{(i,j)}}{N_i} \end{bmatrix} \times \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{N_i 1}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix} X_j$$

$$= X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} W^{(i,j)} X_j.$$

The calculation of third term is

$$\sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \phi\left(x_\ell^{(i)}\right) \left[w_{\ell 1}^{(i,j)}, \ldots, w_{\ell N_j}^{(i,j)}\right] X_j$$

$$= \left[\phi\left(x_1^{(i)}\right), \ldots, \phi\left(x_{N_i}^{(i)}\right)\right] \begin{bmatrix} \frac{\lambda_1^{(i,j)}}{N_i} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_{N_i}^{(i,j)}}{N_i} \end{bmatrix} \times \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{1N_j}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{N_i 1}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix} X_j$$

$$= X_i^{\mathrm{T}} \Lambda^{(i,j)} W^{(i,j)} X_j.$$

The evaluating process of the final term is

$$\sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} X_j^{\mathrm{T}} \begin{bmatrix} w_{\ell 1}^{(i,j)} \\ \vdots \\ w_{\ell N_j}^{(i,j)} \end{bmatrix} \phi\left(x_\ell^{(i)}\right)^{\mathrm{T}}$$

$$= X_j^{\mathrm{T}} \begin{bmatrix} w_{11}^{(i,j)} & \cdots & w_{N_i 1}^{(i,j)} \\ \vdots & \ddots & \vdots \\ w_{1N_i}^{(i,j)} & \cdots & w_{N_i N_j}^{(i,j)} \end{bmatrix} \begin{bmatrix} \frac{\lambda_1^{(i,j)}}{N_i} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_{N_i}^{(i,j)}}{N_i} \end{bmatrix} \begin{bmatrix} \phi\left(x_1^{(i)}\right)^{\mathrm{T}} \\ \vdots \\ \phi\left(x_{N_i}^{(i)}\right)^{\mathrm{T}} \end{bmatrix}$$

$$= X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} X_i.$$

Hence, we have

$$\sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}}$$

$$= X_i^{\mathrm{T}} \Lambda^{(i,j)} X_i + X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} W^{(i,j)} X_j - X_i^{\mathrm{T}} \Lambda^{(i,j)} W^{(i,j)} X_j - X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} X_i$$

for $i = 1, \ldots, L$, $j = 1, \ldots, L$.

The within-class scatter matrix S_w^{KNW} becomes the expression shown at the bottom of the page. The process of S_b^{KNW} is more complex. Note that

$$S_b^{\mathrm{KNW}} = \sum_{i=1}^{L} P_i \sum_{\substack{j=1 \\ j \neq i}}^{L} \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}}$$

$$= \sum_{i=1}^{L} P_i \left(\sum_{j=1}^{L} \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}} - \sum_{\ell=1}^{N_i} \frac{\lambda_\ell^{(i,j)}}{N_i} \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right) \times \left(\phi\left(x_\ell^{(i)}\right) - M_j\left(\phi\left(x_\ell^{(i)}\right)\right)\right)^{\mathrm{T}} \right)$$

$$= \sum_{i=1}^{L} P_i \sum_{j=1}^{L} \left(X_i^{\mathrm{T}} \Lambda^{(i,j)} X_i + X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} W^{(i,j)} X_j - X_i^{\mathrm{T}} \Lambda^{(i,j)} W^{(i,j)} X_j - X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} X_i \right) - X^{\mathrm{T}} W X.$$

$$S_w^{\mathrm{KNW}} = \sum_{i=1}^{L} P_i \left(X_i^{\mathrm{T}} \Lambda^{(i,i)} X_i + X_i^{\mathrm{T}} W^{(i,i)^{\mathrm{T}}} \Lambda^{(i,i)} W^{(i,i)} X_i - X_i^{\mathrm{T}} \Lambda^{(i,i)} W^{(i,i)} X_i - X_i^{\mathrm{T}} W^{(i,i)^{\mathrm{T}}} \Lambda^{(i,i)} X_i \right)$$

$$= \left[X_1^{\mathrm{T}}, \ldots, X_L^{\mathrm{T}}\right] \times \left(\begin{bmatrix} P_1 \Lambda^{(1,1)} & & 0 \\ & \ddots & \\ 0 & & P_L \Lambda^{(L,L)} \end{bmatrix} + \begin{bmatrix} P_1 W^{(1,1)^{\mathrm{T}}} \Lambda^{(1,1)} W^{(1,1)} & & 0 \\ & \ddots & \\ 0 & & P_L W^{(L,L)^{\mathrm{T}}} \Lambda^{(L,L)} W^{(L,L)} \end{bmatrix} - \begin{bmatrix} P_1 \Lambda^{(1,1)} W^{(1,1)} & & 0 \\ & \ddots & \\ 0 & & P_L \Lambda^{(L,L)} W^{(1,1)} \end{bmatrix} - \begin{bmatrix} P_1 W^{(1,1)^{\mathrm{T}}} \Lambda^{(1,1)} & & 0 \\ & \ddots & \\ 0 & & P_L W^{(1,1)^{\mathrm{T}}} \Lambda^{(L,L)} \end{bmatrix} \right) \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix}$$

$$= X^{\mathrm{T}} \left(W_1 + W_2 - W_3 - W_3^{\mathrm{T}}\right) X = X^{\mathrm{T}} W X$$

Note that we have the following expressions.

1)

$$\sum_{j=1}^{L} X_i^{\mathrm{T}} \Lambda^{(i,j)} X_i = X_i^{\mathrm{T}} \left(\sum_{j=1}^{L} \Lambda^{(i,j)} \right) X_i.$$

2)

$$\begin{aligned}
&\sum_{j=1}^{L} X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} W^{(i,j)} X_j \\
&\quad = \left[X_1^{\mathrm{T}}, \ldots, X_L^{\mathrm{T}} \right] \\
&\qquad \times \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} W^{(i,1)} & & 0 \\ & \ddots & \\ 0 & & W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} W^{(i,L)} \end{bmatrix} \\
&\qquad \times \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix} \\
&\quad = X^{\mathrm{T}} \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} W^{(i,1)} & & 0 \\ & \ddots & \\ 0 & & W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} W^{(i,L)} \end{bmatrix} X.
\end{aligned}$$

3)

$$\begin{aligned}
&\sum_{j=1}^{L} X_i^{\mathrm{T}} \Lambda^{(i,j)} W^{(i,j)} X_j \\
&\quad = X_i^{\mathrm{T}} \sum_{j=1}^{L} \Lambda^{(i,j)} W^{(i,j)} X_j \\
&\quad = X_i^{\mathrm{T}} \left[\Lambda^{(i,1)} W^{(i,1)}, \ldots, \Lambda^{(i,L)} W^{(i,L)} \right] \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix} \\
&\quad = X_i^{\mathrm{T}} \left[\Lambda^{(i,1)} W^{(i,1)}, \ldots, \Lambda^{(i,L)} W^{(i,L)} \right] X.
\end{aligned}$$

4)

$$\begin{aligned}
&\sum_{j=1}^{L} X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} X_i \\
&\quad = \left(\sum_{j=1}^{L} X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} \right) X_i \\
&\quad = \left[X_1^{\mathrm{T}}, \ldots, X_L^{\mathrm{T}} \right] \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} \\ \vdots \\ W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} \end{bmatrix} X_i \\
&\quad = X^{\mathrm{T}} \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} \\ \vdots \\ W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} \end{bmatrix} X_i.
\end{aligned}$$

From earlier, we obtain the expression shown at the bottom of the page. From the following calculations, we can obtain the formula of S_b^{KNW}.

1)

$$\begin{aligned}
&\sum_{i=1}^{L} P_i X_i^{\mathrm{T}} \left(\sum_{j=1}^{L} \Lambda^{(i,j)} \right) X_i \\
&\quad = \left[X_1^{\mathrm{T}}, \ldots, X_L^{\mathrm{T}} \right] \\
&\qquad \times \begin{bmatrix} P_1 \sum_{j=1}^{L} \Lambda^{(1,j)} & & 0 \\ & \ddots & \\ 0 & & P_L \sum_{j=1}^{L} \Lambda^{(L,j)} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix} \\
&\quad = X^{\mathrm{T}} B_1 X.
\end{aligned}$$

2)

$$\begin{aligned}
&\sum_{i=1}^{L} P_i X^{\mathrm{T}} \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} W^{(i,1)} & & 0 \\ & \ddots & \\ 0 & & W^{(i,L)^{\mathrm{T}}} \Lambda(i,L) W^{(i,L)} \end{bmatrix} X \\
&\quad = X^{\mathrm{T}} B_2 X.
\end{aligned}$$

3)

$$\begin{aligned}
&\sum_{i=1}^{L} P_i X_i^{\mathrm{T}} \left[\Lambda^{(i,1)} W^{(i,1)}, \ldots, \Lambda^{(i,L)} W^{(i,L)} \right] X \\
&\quad = \left[X_1^{\mathrm{T}}, \ldots, X_L^{\mathrm{T}} \right] \\
&\qquad \times \begin{bmatrix} P_1 \Lambda^{(1,1)} W^{(1,1)} & \cdots & P_1 \Lambda^{(1,L)} W^{(1,L)} \\ \vdots & \ddots & \vdots \\ P_L \Lambda^{(L,1)} W^{(L,1)} & \cdots & P_L \Lambda^{(L,L)} W^{(L,L)} \end{bmatrix} X \\
&\quad = X^{\mathrm{T}} B_3 X.
\end{aligned}$$

$$\begin{aligned}
&\sum_{i=1}^{L} P_i \sum_{j=1}^{L} \left(X_i^{\mathrm{T}} \Lambda^{(i,j)} X_i + X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} W^{(i,j)} X_j - X_i^{\mathrm{T}} \Lambda^{(i,j)} W^{(i,j)} X_j - X_j^{\mathrm{T}} W^{(i,j)^{\mathrm{T}}} \Lambda^{(i,j)} X_i \right) \\
&\quad = \sum_{i=1}^{L} P_i \left(X_i^{\mathrm{T}} \left(\sum_{j=1}^{L} \Lambda^{(i,j)} \right) X_i + X^{\mathrm{T}} \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} W^{(i,1)} & & 0 \\ & \ddots & \\ 0 & & W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} W^{(i,L)} \end{bmatrix} X \right. \\
&\qquad \left. - X_i^{\mathrm{T}} \left[\Lambda^{(i,1)} W^{(i,1)}, \ldots, \Lambda^{(i,L)} W^{(i,L)} \right] X - X^{\mathrm{T}} \begin{bmatrix} W^{(i,1)^{\mathrm{T}}} \Lambda^{(i,1)} \\ \vdots \\ W^{(i,L)^{\mathrm{T}}} \Lambda^{(i,L)} \end{bmatrix} X_i \right)
\end{aligned}$$

4)

$$\sum_{i=1}^{L} P_i X^{\mathrm{T}} \begin{bmatrix} W^{(i,1)^{\mathrm{T}}}\Lambda^{(i,1)} \\ \vdots \\ W^{(i,L)^{\mathrm{T}}}\Lambda^{(i,L)} \end{bmatrix} X_i$$
$$= X^{\mathrm{T}} \begin{bmatrix} P_1 W^{(1,1)^{\mathrm{T}}}\Lambda^{(1,1)} & \cdots & P_L W^{(L,1)^{\mathrm{T}}}\Lambda^{(L,1)} \\ \vdots & \ddots & \vdots \\ P_1 W^{(1,L)^{\mathrm{T}}}\Lambda^{(1,L)} & \cdots & P_L W^{(L,L)^{\mathrm{T}}}\Lambda^{(L,L)} \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_L \end{bmatrix}$$
$$= X^{\mathrm{T}} B_3^{\mathrm{T}} X.$$

Therefore, we have

$$S_b^{\mathrm{KNW}} = X^{\mathrm{T}} \left(B_1 + B_2 - B_3 - B_3^{\mathrm{T}}\right) X - X^{\mathrm{T}} W X$$
$$= X^{\mathrm{T}} (B - W) X. \quad \blacksquare$$

Corollary 4: If the rank of $X^{\mathrm{T}} = [x_1, \ldots, x_N]$ is d, then under the Fisher criterion, KNWFE with linear kernel is the same as NWFE.

Proof: Since the rank of X is d, thus the columns of A is a linear combination of X, i.e.,

$$A = X^{\mathrm{T}} \widetilde{A}.$$

(Note that $\tilde{A}$ is not unique.) Then

$$\mathrm{tr}\left((A^{\mathrm{T}} X^{\mathrm{T}} W X A)^{-1} A^{\mathrm{T}} X^{\mathrm{T}} (B - W) X A\right)$$
$$= \mathrm{tr}\left((\widetilde{A}^{\mathrm{T}} X X^{\mathrm{T}} W X X^{\mathrm{T}} \widetilde{A})^{-1} \widetilde{A}^{\mathrm{T}} X X^{\mathrm{T}} (B - W) X X^{\mathrm{T}} \widetilde{A}\right).$$

The

$$X X^{\mathrm{T}} = \begin{bmatrix} x_1^{\mathrm{T}} \\ \vdots \\ x_N^{\mathrm{T}} \end{bmatrix} [x_1, \ldots, x_N] = \begin{bmatrix} x_1^{\mathrm{T}} x_1 & \cdots & x_1^{\mathrm{T}} x_N \\ \vdots & \ddots & \vdots \\ x_N^{\mathrm{T}} x_1 & \cdots & x_N^{\mathrm{T}} x_N \end{bmatrix}$$
$$= \begin{bmatrix} \kappa(x_1, x_1) & \cdots & \kappa(x_1, x_N) \\ \vdots & \ddots & \vdots \\ \kappa(x_N, x_1) & \cdots & \kappa(x_N, x_N) \end{bmatrix}$$
$$= K$$

where κ is linear kernel and K is the corresponding kernel matrix. Therefore

$$\mathrm{tr}\left((A^{\mathrm{T}} X^{\mathrm{T}} W X A)^{-1} A^{\mathrm{T}} X^{\mathrm{T}} (B - W) X A\right)$$
$$= \mathrm{tr}\left((\widetilde{A}^{\mathrm{T}} K W K \widetilde{A})^{-1} \widetilde{A}^{\mathrm{T}} K (B - W) K \widetilde{A}\right)$$

i.e., under the Fisher criterion, KNWFE with linear kernel is the same as NWFE.

ACKNOWLEDGMENT

The authors would like to thank Prof. Landgrebe for providing the Indian Pine and Washington, DC Mall data sets and Prof. Crawford for providing KSC data set.

REFERENCES

[1] S. Mika, G. Ratsch, B. Scholkopf, A. Smola, J. Weston, and K.-R. Muller, "Invariant feature extraction and classification in kernel spaces," in *Advances in Neural Information Processing Systems*, vol. 12. Cambridge, MA: MIT Press, 1999.

[2] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, Oct. 2000.

[3] B. Schölkopf, A. Smola, and K.R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[4] S. T. John and C. Nello, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[5] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[6] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[7] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. L. Rojo-Alvarez, and M. Martinez-Ramon, "Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1822–1835, Mar. 2008.

[8] P. F. Hsieh, D. S. Wang, and C. W. Hsu, "A linear feature extraction for multiclass classification problems based on class mean and covariance discriminant information, extraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 223–235, Feb. 2006.

[9] X. Song, G. Fan, and M. Rao, "Automatic CRP mapping using nonparametric machine learning approaches," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 888–897, Apr. 2005.

[10] J. A. Richards, "Analysis of remotely sensed data: The formative decades and the future," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 422–432, Mar. 2005.

[11] D. Landgrebe, "Multispectral land sensing: Where from, where to?" *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 414–421, Mar. 2005.

[12] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.

[13] M. M. Dundar and D. Landgrebe, "Toward an optimal supervised classifier for the analysis of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 1, pp. 271–277, Jan. 2004.

[14] B. C. Kuo and D. A. Landgrebe, "Nonparametric weighted feature extraction for classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 5, pp. 1096–1105, May 2004.

[15] B. S. Sebastiano and M. Gabriele, "Extraction of spectral channels from hyperspectral images for classification purposes," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 2, pp. 484–495, Feb. 2007.

[16] I. Jlliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[17] V. Zubko, Y. J. Kaufman, R. I. Burg, and J. V. Martins, "Principal component analysis of remote sensing of aerosols over oceans," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 730–745, Mar. 2007.

[18] A. Hyvrinen, J. Karhunen, and E. Oja, *Independent Copmonent Analysis*. New York: Wiley.

[19] J. A. Palmason, J. A. Benediktsson, J. R. Sveinsson, and J. Chanussot, "Classification of hyperspectral data from urban areas using morphological preprocessing and independent component analysis," in *Proc. IGARSS*, Jul. 2005, vol. 1, pp. 176–179.

[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, CA: Academic, 1990.

[21] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 388–400, Apr. 1993.

[22] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: Wiley, 2003.

[23] J. Ham, Y. Chen, M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 492–501, Mar. 2005.

[24] R. P. W. Duin, *PRTools, a Matlab Toolbox for Pattern Recognition*, Apr. 2008. [Online]. Available: http://www.prtools.org/

[25] C. C. Chang and C. J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

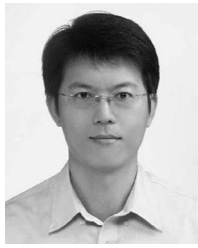
[26] A. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995.
[27] H. B. Nielsen, "UCMINF—An algorithm for unconstrained, nonlinear optimization," IMM, Tech. Univ. Denmark, Copenhagen, Denmark, AFFDL-TR-78, 2001. [Online]. Available: http://www.imm.dtu.dk/pubdb/views/edoc_download.php/642/ps/imm642.ps
[28] M. I. Schlesinger, V. Hlavac, and V. Franc, "*Statistical Pattern Recognition Toolbox,*" [Online]. Available: http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html
[29] G. Golub and C. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
[30] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 1, pp. 55–63, Jan. 1968.
[31] D. A. Landgrebe, "Information extraction principles and methods for multispectral and hyperspectral image data," in *Information Processing for Remote Sensing*, C. H. Chen, Ed. Singapore: World Scientific, 1999, ch. 1.
[32] S. Tadjudin and D. A. Landgrebe, "Classification of high dimensional data with limited training samples," Purdue Univ., West Lafayette, IN, ECE Tech. Rep. TR-EE 98-8, Apr. 1998.
[33] W. J. Krzanowski, P. Jonathan, W. V. McCarthy, and M. R. Thomas, "Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data," *Appl. Stat.*, vol. 44, no. 1, pp. 101–115, 1995.
[34] C. B. Moler and G. W. Stewart, "An algorithm for generalized matrix Eigenvalue problems," *SIAM J. Numer. Anal.*, vol. 10, no. 2, pp. 241–256, Apr. 1973.
[35] J. Duchene and S. Leclercq, "An optimal transformation for discriminant analysis and principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 978–983, Nov. 1988.
[36] B.-C. Kuo and D. A. Landgrebe, "Improved statistics estimation and feature extraction for hyperspectral data classification," School of Elect. Comput. Eng., Purdue Univ., West Lafayette, IN, Tech. Rep. TR-ECE 01-6, Dec. 2001. [Online]. Available: http://dynamo.ecn.purdue.edu/~landgreb/ publications.html
[37] K. Fukunaga and M. Mantock, "Nonparametric discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 6, pp. 671–678, Nov. 1983.
[38] B.-C. Kuo, D. A. Landgrebe, L.-W. Ko, and C.-H. Pai, "Regularized feature extractions for hyperspectral data classification," in *Proc. IGARSS*, Toulouse, France, Jul. 21–25, 2003, pp. 1767–1769.
[39] J. Munoz-Marf, L. Bruzzone, and G. Camps-Valls, "A support vector domain description approach to supervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2683–2692, Aug. 2008.
[40] L. O. Jimenez-Rodriguez, E. Arzuaga-Cruz, and M. Velez-Reyes, "Unsupervised linear feature-extraction methods and their effects in the classification of high-dimensional data," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 2, pp. 469–483, Feb. 2007.
[41] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.
[42] C. Marrocco, R. P. W. Duin, and F. Tortorella, "Maximizing the area under the ROC curve by pairwise feature combination," *Pattern Recognit.*, vol. 41, no. 6, pp. 1961–1974, Jun. 2008.

Bor-Chen Kuo received the B.S. and M.S. degrees from National Taichung Teachers College, Taichung, Taiwan, in 1993 and 1996, respectively, and the Ph.D. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2001.

He is currently an Associate Professor with the Graduate Institute of Educational Measurement and Statistics, National Taichung University, Taichung. His research interests include pattern recognition, remote sensing, image processing, and nonparametric functional estimation.

Cheng-Hsuan Li received the B.S. and M.S. degrees from National Chung Hsing University, Taichung, Taiwan, in 2001 and 2003, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan.

He is currently an Executive Assistant with the Graduate Institute of Educational Measurement and Statistics, National Taichung University, Taichung. His research interests include pattern recognition and machine learning.



Jinn-Min Yang received the M.S. degree from National Taichung University, Taichung, Taiwan, in 2000. He is currently working toward the Ph.D. degree in the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan.

His interests include pattern recognition and machine learning.