PCA, Kernel PCA and Dimensionality Reduction in Hyperspectral Images

Aloke Datta, Susmita Ghosh and Ashish Ghosh

Abstract In this chapter an application of PCA, kernel PCA with their modified versions are discussed in the field of dimensionality reduction of hyperspectral images. Hyperspectral image cube is a set of images from hundreds of narrow and contiguous bands of electromagnetic spectrum from visible to near-infrared regions, which usually contains large amount of information to identify and distinguish spectrally unique materials. In hyperspectral image analysis, reducing the dimensionality is an important step where the aim is to discard the redundant bands and make it less time consuming for classification. Principal component analysis (PCA), and the modified version of PCA, i.e., segmented PCA are useful for reducing the dimensionality. A brief detail of these PCA based methods in the field of hyperspectral images with their advantages and disadvantages are discussed here. Also, dimensionality reduction using kernel PCA (one of the non linear PCA) and its modification i.e., clustering oriented kernel PCA in this field are elaborated in this chapter. Advantages and disadvantages of all these methods are experimentally evaluated over few hyperspectral data sets with different performance measures.

1 Introduction

Development of hyperspectral sensors [1] is a significant breakthrough in remote sensing. Hyperspectral sensors acquire a set of images from hundreds of narrow and contiguous bands of the electromagnetic spectrum from visible to infrared regions. Images captured by hyperspectral sensors have ample spectral information to identify and distinguish spectrally unique materials. There are various applications of hyperspectral images [2–5] like target detection, material identification, mineral mapping,

A. Datta
Department of CSE, NIT Meghalaya, Shillong, India

S. Ghosh
Department of CSE, Jadavpur University, Kolkata, India

A. Ghosh (✉)
Center for Soft Computing Research, Indian Statistical Institute, Kolkata, India
e-mail: ash@isical.ac.in

G. R. Naik (ed.), *Advances in Principal Component Analysis*,
https://doi.org/10.1007/978-981-10-6704-4_2

vegetation species identification, mapping details of surface properties etc. To perform these tasks, homogeneous pixels with defined similarity have to be grouped together (recognition/classification) in hyperspectral images.

Recognition/Classification of patterns is either of the two tasks: supervised classification (or simply known as classification) and unsupervised classification (also known as clustering) [6]. Classification task of hyperspectral images is a very challenging task in recent days due to the presence of a large number of features for each pixel. The performance of a classifier depends on the interrelationship between sample sizes, number of features and classifier complexity. The minimum number of training patterns required for proper training may be an exponential function of the number of features present in a data set [7]. It has been often observed that more features may not increase the performance of a classifier, if the number of training samples is small relative to the number of features. This phenomenon is termed as "curse of dimensionality" [6, 8]. Another fact of hyperspectral images is that the neighboring bands are generally strongly correlated. As a result, it is possible that very less relevant information is actually being added by increasing the spectral resolution. Thus, it can be concluded that large number of features is not always needed. In case of analysis of hyperspectral images, dimensionality reduction is an important issue [9–11].

The main two approaches of dimensionality reductions in hyperspectral images are feature selection and feature extraction [8, 12]. In brief, feature selection [6, 13–19] is nothing but selecting a subset of features from the original set of features to preserve crucial information and reduce redundancy among information. Feature selection methods preserve the original physical meaning of the features; whereas, transforming the original features into a reduced set of features, which preserves the class separability as much as possible in the transformed space, is called feature extraction [20–24]. The extracted features lose the meaning of the original features, but each of the original features may contribute to make a transformed feature. The main advantages of performing feature selection and feature extraction are to improve the classification accuracy by avoiding the "curse of dimensionality" and to reduce the computational cost for classification or clustering of data. Depending on the availability of labeled patterns, feature selection/extraction is categorized into supervised and unsupervised ones. Supervised methods use class label information of patterns and, when no labeled patterns are available, unsupervised method is used for dimensionality reduction.

In this chapter, our main aim is to represent principal component analysis and its various modifications in respect to feature extraction in hyperspectral images. Principal component analysis (PCA) [10], and the modified version of PCA, i.e., segmented PCA [20] are useful for reducing the dimensionality. A brief detail of these PCA based methods in the field of hyperspectral images with their advantage and disadvantages are discussed here. Also, dimensionality reduction using kernel PCA (one of the non linear PCA) [22, 25] and its modification i.e., clustering oriented kernel PCA [26] in this field are elaborated in this chapter. Advantages and disadvantages of all these methods are experimentally evaluated over few hyperspectral data sets in terms of different performance measures.

2 Principal Component Analysis (PCA) Based Feature Extraction Method

Principal component analysis (PCA) [10, 12, 27] is an orthogonal basis transformation with the advantage that the first few principal components preserve most of the variance of the data set. This method [27], initially, calculates the covariance matrix of the given data set, and then finds the eigenvalues and eigenvectors of this matrix. Next it selects a few eigenvectors whose eigenvalues are more to form the transformation matrix to reduce the dimensions of the data set.

Suppose, there are D number of band images. So, a pixel has D number of different responses over different wavelengths. As a consequences, a pixel may be treated as a pattern of D attributes. The main target is to reduce the dimensionality from D to d $(d \ll D)$of hyperspectral image pixel.

Let, there be a set of pattern x_i, where $x_i \in \Re^D$, $i = 1, 2, ..., N$. Assume that the data are centered, i.e., $x_i \Longleftarrow x_i - E\{x_i\}$. Conventional PCA formulates the eigenvalue problem by

$$\lambda V = \Sigma_x V \tag{1}$$

where λ is eigenvalue, V is eigenvector, Σ_x is the corresponding covariance matrix over data set x which is calculated by the following equation

$$\Sigma_x = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T. \tag{2}$$

The projection on the eigenvector V^k is calculated as

$$x_{pc}^k = V^k.x. \tag{3}$$

The principal component based transformation is defined as

$$y_i = W^T x_i; \tag{4}$$

where W is the matrix of first d normalized eigenvectors of highest eigenvalues of the image covariance matrix Σ_x. T denotes the transpose operation.

Here, a pattern x_i from original D-dimensional space is transformed into y_i, a pattern in reduced d-dimensional space by choosing only the first d components (eigenvectors of highest d eigenvalues).

The transformed data set has two main properties which are significant to the application here. The variance in the original data set has been rearranged and reordered so that first few components contain almost all of the variance in the original data, and the components in the new feature space are uncorrelated in nature [20].

3 Segmented Principal Component Analysis (SPCA) Based Feature Extraction Method

In hyperspectral images, the correlations between neighboring spectral bands are generally higher than for bands further apart. If conventional PCA based method is modified so that the transformation is carried out by avoiding the low correlations between the highly correlated blocks, the efficiency of PCA will be improved. Also, the computational load is a major consideration in the case of hyperspectral data transformation, i.e., it is inefficient to transform the complete data set. So, a segmented principal component analysis comes into picture.

In this scheme [20], the complete data set is first partitioned into several subgroups, depending on the correlations of neighboring features of hyperspectral images. Highly correlated features are selected as subgroups. Then, PCA based transformation is conducted separately on each subgroup of data.

At the onset, the D number of bands of a hyperspectral images is partitioned into a few number of contiguous intervals with constant intensities (i.e., K subgroups). Highly correlated bands should be in a subgroup. Let I_1, I_2, ..., I_k, be the number of bands in the 1st, 2nd, and Kth group, correspondingly. The purpose is to obtain a set of K breakpoints $P = \{p_1, p_2, \dots, p_K\}$, which defines the contiguous intervals $I_k = [p_k, p_{k+1})$. The partition should follow the principle that each band should be inside one block.

Let Γ be a correlation matrix of size $D \times D$, where D is the number of bands present in a hyperspectral image. Each element of Γ is γ_{ij}, where γ_{ij} represents the correlation between band images B_i and B_j. Let the size of each band image be $M \times N$. The correlation coefficient between B_i and B_j is defined as

$$\gamma_{i,j} = \frac{\Sigma_{x=1}^{M}\Sigma_{y=1}^{N}|B_i(x,y)-\mu_i||B_j(x,y)-\mu_j|}{\sqrt{(\Sigma_{x=1}^{M}\Sigma_{y=1}^{N}[B_i(x,y)-\mu_i]^2)(\Sigma_{x=1}^{M}\Sigma_{y=1}^{N}[B_j(x,y)-\mu_j]^2)}} \tag{5}$$

where μ_i and μ_j are the mean of band images B_i and B_j, respectively. $|B_i(x,y)-\mu_i|$ measures the difference between the reflectance value of pixel (x, y) from the mean value of the total image.

It is observed that the correlation between neighboring spectral bands are generally higher than for bands further apart. Partitioning is performed based on the results obtained by first considering only correlations whose absolute value exceeds a given threshold, and simultaneously searching for edges in the “image” of the correlation matrix [20]. Each value of the correlation matrix is compared with a threshold (correlation). If the magnitude is greater than the threshold value (i.e., denoted by Θ), then replace it by 1; otherwise by 0. The value of Θ has been determined depending on the value of average correlation (μ_{corr}) and standard deviation (σ_{corr}) of correlation matrix Γ as

$$\Theta = \mu_{corr} + \sigma_{corr}; \tag{6}$$

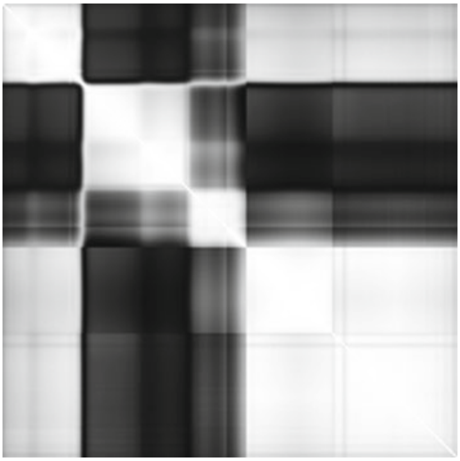


Fig. 1 Gray scale image representation of the correlation matrix of Indian data set

where,

$$\mu_{corr} = \frac{1}{D^2} \Sigma_{i=1}^{D} \Sigma_{j=1}^{D} \gamma_{i,j}; \tag{7}$$

and

$$\sigma_{corr} = sqrt(\frac{1}{D^2} \Sigma_{i=1}^{D} \Sigma_{j=1}^{D} (\gamma_{i,j} - \mu_{corr})). \tag{8}$$

The image of the thresholded correlation matrix will be a binary image with the square blocks of white color in diagonal direction. These square blocks of white color are treated as a subgroup or partition of bands. An example of the correlation matrix of AVIRIS Indian data in image form is shown in Fig. 1.

Now, PCA based transformation is conducted on each subgroup of data. Selection over obtained principal components from each subgroup is performed based on pairwise separability measure, such as the Bhattacharyya distance [20].

4 Kernel Principal Component Analysis (KPCA) Based Feature Extraction Method

PCA, basically, rotates the original axes, so that the new coordinate system aligns with the orientation of maximum variability of data. Rotation is a linear transformation and the new coordinate axes are then a linear combination of the original axes. So, PCA as a linear algorithm is inadequate to extract the non linear structures of the data. Also, PCA only considers variance between patterns which is a second order statistics, that may limit the effectiveness of the method. So, a non-linear version of PCA is considered, which is called kernel PCA (KPCA). It is capable of capturing a part of higher order statistics. So it is useful for representing the information from the original data set which is more useful to discriminate among themselves.

Kernel principal component analysis [22], a nonlinear version of the PCA is capable of capturing a part of higher order statistics, which may represent the information in a better way from the original data set to reduced data set [25]. This technique is used for reducing the dimensionality of hyperspectral images. Here, the data of the input space $\Re^D$ is mapped into another space, called feature space $\mathcal{F}$, to capture higher-order statistics. A non-linear mapping function Φ is used to transfer the data from input feature space to a new feature space by

$$\Phi : \Re^D \rightarrow \mathcal{F};$$

$$x \rightarrow \Phi(x). \tag{9}$$

The non-linear function Φ transforms a pattern x from D-dimensional input space to another feature space $\mathcal{F}$. The covariance matrix in this feature space is calculated as

$$\Sigma_{\Phi(x)} = \frac{1}{N}\sum_{i=1}^{N} \Phi(x_i)\Phi(x_i)^T. \tag{10}$$

The principal components are then computed by solving the eigenvalue problem

$$\lambda V = \Sigma_{\Phi(x)} V = \frac{1}{N}\sum_{i=1}^{N}(\Phi(x_i).V)\Phi(x_i). \tag{11}$$

Furthermore, all eigenvectors with nonzero eigenvalue must be in the span of mapped data, i.e., $V \in span\{\Phi(x_1), ..., \Phi(x_N)\}$, and there exists coefficients α_i $(i = 1, 2, ..., N)$ such that

$$V = \sum_{i=1}^{N} \alpha_i \Phi(x_i). \tag{12}$$

Here, $\boldsymbol{V}$ denotes the eigenvector and x_i denotes the ith pattern. Multiplying Eq. 11 by $\Phi(x_k)$ from left and substituting Eq. 12 into it, we get

$$\lambda \sum_{i=1}^{N} \alpha_i (\Phi(x_k)\Phi(x_i)) = \frac{1}{N}\sum_{i=1}^{N} \alpha_i \left(\Phi(x_k). \sum_{j=1}^{N}(\Phi(x_j).\Phi(x_i))\Phi(x_j)\right); \tag{13}$$

for $k = 1, ..., N$.

Calculation of principle components in feature space $\mathcal{F}$ is computationally prohibitive. It is possible to work implicitly in $\mathcal{F}$ while all computations is done in the input space using kernel trick. Using kernel function, the product in feature space is reduced to a possibly nonlinear function (denoted by ψ) in the input space

$$\Phi(x_i).\Phi(x_j) = \psi(x_i, x_j). \tag{14}$$

Now, the NXN matrix, termed as kernel matrix Ψ, is defined as

$$\Psi = \begin{pmatrix} \psi(x_1, x_1) & \psi(x_1, x_2) & \cdots & \psi(x_1, x_N) \\ \psi(x_2, x_1) & \psi(x_2, x_2) & \cdots & \psi(x_2, x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(x_N, x_1) & \psi(x_N, x_2) & \cdots & \psi(x_N, x_N) \end{pmatrix}.$$

Using the kernel matrix Ψ, Eq. 13 becomes

$$\lambda\alpha = \Psi\alpha; \tag{15}$$

where, $\alpha = (\alpha_1, ..., \alpha_N)^T$, T denotes the transpose operation, and one computes an eigenvalue for the expansion coefficient α_i, which is solely dependent on kernel function.

Like PCA algorithm, the data needs to be centered in F and it is done by substituting the kernel matrix Ψ by

$$\Psi_c = \Psi - 1_N\Psi - \Psi 1_N + 1_N\Psi 1_N; \tag{16}$$

where 1_N is a square matrix such as $(1_N)_{ij} = 1/N$.

For extracting features of a new pattern x with KPCA, one simply projects the mapped pattern $\Phi(x)$ into kth eigen vector V^k by

$$(V^k.\Phi(x)) = \sum_{i=1}^{M} \alpha_i^k(\Phi(x_i).\Phi(x)) = \sum_{i=1}^{M} \alpha_i^k \psi(x_i, x). \tag{17}$$

The KPCA incorporates nonlinearity in the calculation of the matrix elements of Ψ and the evaluation of the expansion.

The function ψ is a positive semi-definite function on $\Re^D$ which incorporates nonlinearity into processing. This is usually called a kernel. Selecting an appropriate kernel is a new scope of research. It is better to use Gaussian kernel if there are assumptions of the nature of clusters of data as Gaussian. Hyperspectral remote sensing data are known to be well approximated by a Gaussian distribution [28]. So, in this article, Gaussian kernel is used, which is described by following equation

$$\psi(x_i, x_j) = exp\left(-\frac{||x_i - x_j||}{2\sigma^2}\right). \tag{18}$$

In the Gaussian kernel, the parameter σ, controls the width of the exponential function. For a very small value of σ, each sample is considered as an individual cluster, and vice-versa. The value of σ depends on data set [25]. This KPCA based feature extraction method selects some percent of data from the total data set randomly to calculate the kernel matrix, i.e., value of σ [22]. The minimum distance of all representative patterns x_i with other patterns is calculated. Thus, if there are N patterns, then there will be N minimum distances. The average of this N minimum distances is calculated. σ is taken as five times of this minimum value. Thus, σ value is dependent on the nature of data set.

5 Clustering Oriented Kernel Principal Component Analysis (KPCA) Based Feature Extraction Method

The clustering oriented KPCA based feature extraction method [26] performs kernel principal component analysis to transform the original data set of dimension D into d dimensional space. The KPCA is non linear in nature and uses higher order statistics of data set to discriminate the classes. The most important thing is to select the proper training set for calculating kernel matrix for KPCA. A randomly selected training pattern may not represent the overall data set properly. Also, it should not be too large so that the method becomes computationally prohibitive. So, a proper subset of original hyperspectral data set which can represent the total data set properly should be selected and this training set should not contain any noisy data. DBSCAN clustering technique is used for choosing the proper representative training set. In this section, selection of N representative patterns using DBSCAN clustering technique is described and then discuss about the KPCA based transformation using these data.

KPCA shares the same properties as the PCA, but in a different space. Both PCA and KPCA need to solve eigenvalue problem, but the dimensions of the problem are different, $D \times D$ for PCA and $N \times N$ for KPCA, where D is the dimensions of data set and N is number of representative patterns required to calculate kernel matrix Ψ. The size of the matrix becomes problematic for large N. Number of pixel points (N) in hyperspectral images is huge, so it is difficult to perform KPCA by taking all the pixels. If some percentage of total pixels are selected randomly, then the selected pixels may not represent the characteristics of total data. So, it is better to make small group of pixels according to their similarity, and then take some representative pixels from each group to make the representative pattern set for KPCA.

Selecting of N Representative Pixels using DBSCAN Clustering

Pixels on homogeneous region have similar properties and make group or region in hyperspectral images by clustering. Each pixel of a hyperspectral image can be treated as a pattern with D attributes, where D represents the total number of features present in the images. In the proposed investigation a density based spatial clustering technique (DBSCAN) [29] is applied to obtain the region types. It does not require prior information regarding the number of clusters. DBSCAN treats a noisy pattern as an isolated point, rather than including it into any cluster. The main concept of DBSCAN clustering technique is that within each cluster, density of points is considerably higher than outside the cluster, whereas, the density around the noisy area is lower than the density in any of the clusters. So, if the neighborhood of a given radius of a pattern, contains at least a minimum number of patterns, i.e. the density in the neighborhood exceeds some threshold, then that pattern is in a cluster.

It requires two user-defined parameters, neighborhood distance (*Eps*) and the minimum number of points (*MinPts*). For a given point, the points within an Eps distance are called neighbors of that point. DBSCAN labels the data points as core points, border points, and outlier points. Core points are those which have at least *MinPts* number of points within the *Eps* distance in all directions. Border points can be defined as points that are not core points, but are the neighbors of core points. Outlier points are those which are neither core points nor border points.

The algorithm starts with an arbitrary starting point and then finds all the neighboring points within Eps distance of the starting point. If the number of points of its neighborhood is greater than or equal to *MinPts*, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The algorithm then repeats this process for all the neighbors iteratively. If the number of neighbors is less than *MinPts*, the point is marked as noise (i.e., isolated point). If a cluster is fully expanded, then the algorithm proceeds to iterate through the remaining unvisited points in the data set. The steps of DBSCAN are given in Algorithm 1.

Algorithm 1 Pseudo Code of DBSCAN Algorithm

```
Let $S = \{x_1, x_2, \ldots, x_n\}$
Let $class(x) = -1, \forall x \in S$
Choose $Eps$ and $MinPts$
$class\_no = 1$
for $i = 1$ to $n$ do
  $A_i = \{x \in S : d(x, x_i) \le Eps\}$
  if $(\mid A_i \mid \ge MinPts)$ then
    if $(class(x_i) == -1)$ then
      if $(max(class(x : x \in A_i)) > -1)$ then
        $new\_class\_no = min(class(x : x \in A_i$ and $class(x : x \in A_i) > -1))$
        $class(x_i) = new\_class\_no$
        $class(x : \forall x \in A_i) = new\_class\_no$
      else
        $class(x_i) = class\_no$
        $class(x : \forall x \in A_i) = class\_no$
        $class\_no = class\_no + 1$
      end if
    else
      $new\_class\_no = class(x_i)$
      $class(x : \forall x \in A_i) = new\_class\_no$
    end if
  end if
end for
return class
```

Let DBSCAN clustering technique produce C clusters. The isolated pixels identified by DBSCAN algorithm are discarded considering them as noise. Number of clusters (C) does not lie on any predefined range, it is dependent on the data set. Basically it is better that C be close to the number of regions/ land cover types present on the hyperspectral image. The value of C gives an approximation on the number of land cover types/ groups present in the images. DBSCAN clustering technique gives only the clusters present in the data set, but not the cluster centers.

From each cluster, a certain percentage of pixels are selected as representative patterns for calculating the kernel matrix of the KPCA based method. For example, if a cluster C_1 has N_1 pixels, then $N_1/10$ number of pixels are selected from that cluster. The first selected pixels of each cluster is the mean of all the pixels present in a cluster. If a cluster mean does not represent a physical pixel in that cluster, then the nearest pixel of cluster mean is selected from that cluster. Then the next pixels from another cluster is selected which has the maximum distance from other selected pixels of that cluster. The isolated pixels or noisy pixels, which is far away from any cluster (DBSCAN clustering technique detects them and considers them separately) would not be included in the representative pattern set, because KPCA is susceptible to noise.

Now, the KPCA based transformation is performed to reduce the dimensionality from D to d, as described in Sect. 4, where the set of representative patterns are selected by DBSCAN clustering technique to properly represent the characteristics of whole data set. This technique is called as clustering oriented KPCA based feature extraction method of hyperspectral images. An outline of the clustering oriented KPCA based feature extraction method is given in Algorithm 2.

Algorithm 2 Clustering oriented KPCA based feature extraction algorithm

1. Selecting N representative pixels

   - Perform DBSCAN clustering technique over pixels of hyperspectral images which is in D-dimensional space using Algorithm 1.
   - Choose some percentage of exemplar pixels from each cluster to make N representative pixels.
   - These N pixels are used as representative pixels for calculating the kernel matrix in KPCA.

2. Using kernel PCA, transform data into reduced d-dimensional space

   - Compute kernel matrix, Ψ, using Eq. 18
   - Center Ψ, using Eq. 16
   - Solve eigen value problem of Eq. 15
   - Extract the d first principal components using Eq. 17

6 Experimental Evaluation

6.1 Description of Data Sets

Experiments are carried out to evaluate the effectiveness of these feature extraction methods on three hyperspectral remotely sensed images namely, Indian Pine [30], KSC [31], and Botswana [31] images corresponding to the geographical areas of Indian Pine test site of Northwest Indiana, Kennedy Space Center of Florida and Okavango Delta of Botswana. The data sets are described here.

Indian Pine data:
Indian Pine image [30] data was captured by AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) over an agricultural portion of northwest Indiana's Indian Pine test site in the early growing season of 1992. The data has been taken within the spectral range from 400 to 2500 nm with spectral resolution of about 10 nm and has 220 spectral bands.

The size of the image is 145 × 145 pixels and spatial resolution is 20 m. Twenty water absorption bands (numbered 104–108, 150–163 and 220) and 15 noisy bands (1–3, 103, 109–112, 148–149, 164–165 and 217–219) were removed, resulting in a total of 185 bands. There are 16 classes in this image. Class name and the number of labeled samples for each class are given in Table 1. Among the 16 classes, seven classes contain fewer samples. For more details and ground truth information, see [30] and visit http://dynamo.ecn.purdue.edu/biehl/ (Figs. 2, 3, and 4).

Table 1 Indian Pine data: class names and the number of samples

Class no	Class name	No. of samples
C1	Corn	191
C2	Corn-min	688
C3	Corn-notill	1083
C4	Soybean-clean	541
C5	Soybean-min	2234
C6	Soybean-notill	860
C7	Wheat	211
C8	Alfalfa	51
C9	Oats	20
C10	Grass/ Trees	605
C11	Grass/ Pasture	351
C12	Grass/ Pasture-mowed	17
C13	Woods	1293
C14	Hay-windrowed	477
C15	Bldg-Grass-Tree-Drives	380
C16	Stone-steel-towers	86

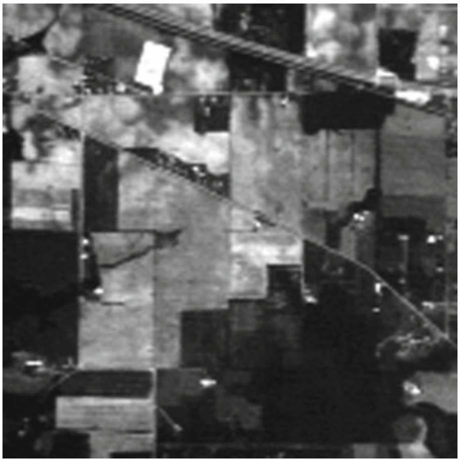


Fig. 2 Band 11 image of Indian Pine data

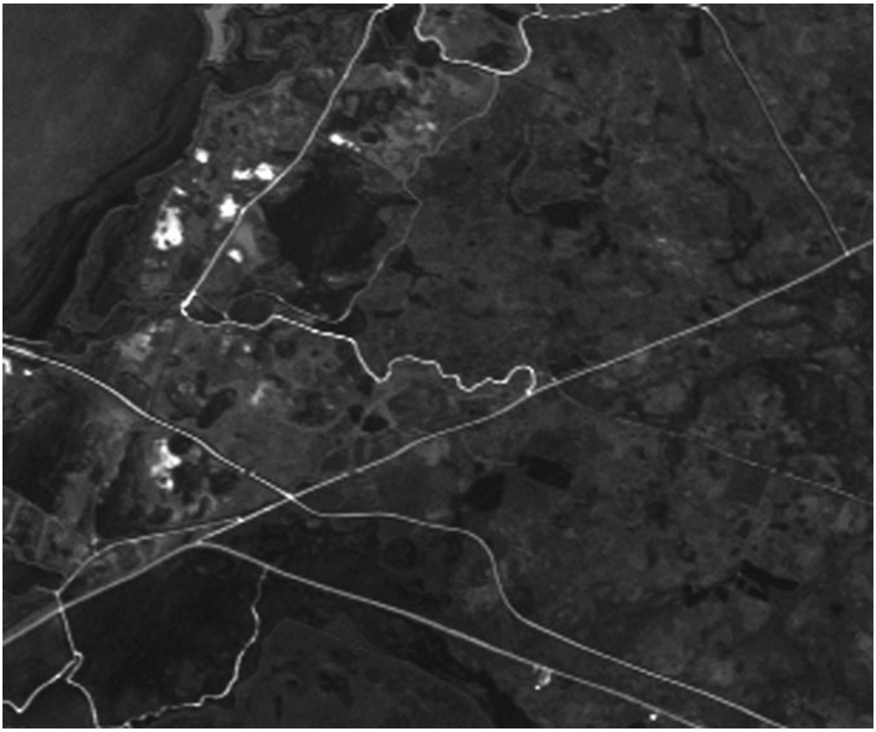
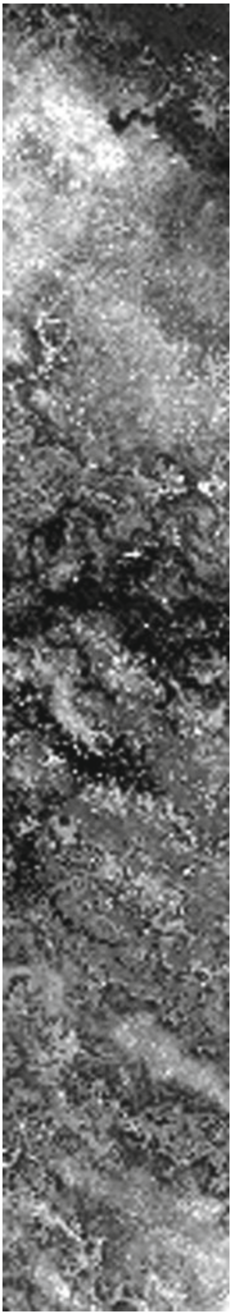


Fig. 3 Band 11 image of KSC data

Fig. 4 Band 11 image of Botswana data



KSC data:
The KSC [31] images, acquired over Kennedy Space Center (KSC), Florida on March 23, 1996 by NASA AVIRIS, is of size 512 × 614. AVIRIS acquires data in 224 bands of 10 nm width with wavelengths ranging from 400 to 2500 nm. The data is acquired from an altitude of approximately 20 km with a spatial resolution of 18 m. After removing the bands disturbed due to water absorption or with low signal-to-noise-ratio (SNR) value (numbered 1–4, 102–116, 151–172 and 218–224), 176 bands are used for analysis. Training data were selected using land cover maps derived from color infrared photography provided by the Kennedy Space Center and Landsat Thematic Mapper (TM) imagery. The vegetation classification scheme was developed by KSC personnel in an effort to define functional types that are discernable at the spatial resolution of Landsat and the AVIRIS data [31]. Discrimination of land cover for this environment is difficult due to similarity of spectral signatures for certain vegetation type. Details of the 13 land cover classes considered in the KSC data area are listed in Table 2. For more details and ground truth information, see [31] and visit http://www.csr.utexas.edu/.

Botswana data:
The NASA Earth Observing 1 (EO-1) satellite acquired a sequence of 1476 × 256 pixels over the Okavango Delta, Botswana in 2001–2004 [31]. The Hyperion sensor on EO-1 acquired data at 30 m pixel resolution over a 7.7 km × 44 km surface are in 242 bands from the 400–2500 nm portion of the spectrum in 10 nm windows. Preprocessing of the data was performed by the UT Center for Space Research to mitigate the effects of bad detectors, inter-detector miscalibration, and intermittent anomalies. Uncalibrated and noisy bands which cover water absorption features were removed, and the remaining 145 bands were included as candidate features: [10–55, 82–97, 102–119, 134–164, 187–220]. This data was acquired on May 31, 2001 and

Table 2 KSC data: class names and the number of samples

Class no	Class name	No. of samples
C1	Scrub	761
C2	Willow swamp	243
C3	Cabbage palm hammock	256
C4	Cabbage palm/oak hammock	252
C5	Slash pine	161
C6	Oak/broadleaf hammock	229
C7	Hardwood swamp	105
C8	Graminoid marsh	431
C9	Spartina marsh	520
C10	Cattail marsh	404
C11	Salt marsh	419
C12	Mud flats	503
C13	Water	927

Table 3 Botswana data: class names and the number of samples

Class no	Class name	No. of samples
C1	Water	270
C2	Hippo Grass	101
C3	FloodPlain Grasses 1	251
C4	FloodPlain Grasses 2	215
C5	Reeds	269
C6	Riparian	269
C7	Firescar	259
C8	Island Interior	203
C9	Acacia Woodlands	314
C10	Acacia Shrublands	248
C11	Acacia Grasslands	305
C12	Short Mopane	181
C13	Mixed Mopane	268
C14	Exposed Soils	95

consists of observations from 14 identified classes representing the land cover types in seasonal swamps, occasional swamps, and drier woodlands located in the distal portion of the Delta [31]. These classes were chosen to reflect the impact of flooding on vegetation in the study area. Class names and corresponding number of ground truth observations used in our experiment are listed in Table 3. For more details and ground truth information, see [31] and visit http://www.csr.utexas.edu/.

6.2 *Performance Measures*

In this section, four feature evaluation indices namely, class separability (S) [8], overall classification accuracy (OA) [32], kappa coefficient (κ) [32] and entropy (E) [33], have been described which are considered for evaluating the effectiveness of the extracted features. The first three measuring indices need class label information of the samples while the last one does not require the same. The details of the evaluation indices used in this thesis, are given below.

Overall Accuracy (OA)**:**
Overall accuracy [32] represents the ratio between the number of samples correctly recognized by the classification algorithm and the total number of test samples. To measure the overall accuracy, initially, confusion matrix is determined. The confusion matrix is a square matrix of size $C \times C$, where C represents the number of classes of the given data set. The element n_{ij} of the matrix denotes the number of samples of the jth ($j = 1, 2, ..., C$) category which are classified into ith ($i = 1, 2, ..., C$) category. Let N be the total number of samples; where $N = \sum_{i=1}^{C} \sum_{j=1}^{C} n_{ij}$. The

overall accuracy (*OA*) is defined as

$$OA = \frac{\sum_{i=1}^{C} n_{ii}}{N}. \tag{19}$$

***Kappa* Coefficient (κ):**
The kappa coefficient (κ) [32] is a measure defined on the difference between the actual agreement in the confusion matrix and the chance agreement, which is indicated by row and column totals of the confusion matrix. The kappa coefficient is widely adopted, as it also takes into consideration the off-diagonal elements of the confusion matrix and compensates for chance agreement. The value of κ lies in the range $[-1\ ,+1]$. Closer the value of κ to $+1$, better is the classification.

Let, in the confusion matrix, the sum of the elements of ith row be denoted as n_{i+} (where, $n_{i+} = \sum_{j=1}^{C} n_{ij}$) and the sum of the elements of column j be n_{+j} (where $n_{+j} = \sum_{i=1}^{C} n_{ij}$). The kappa coefficient is then defined as

$$\kappa = \frac{N \sum_{i=1}^{C} n_{ii} - \sum_{i=1}^{C} n_{i+} n_{+i}}{N^2 - \sum_{i=1}^{C} n_{i+} n_{+i}}; \tag{20}$$

where N denotes the total number of samples and C denotes the number of classes of the given data set.

Class Separability:
Our aim is to look for a feature space where the inter-class distance is large and at the same time the intra-class distance is as small as possible. Let there be C classes $\omega_1, \omega_2, \ldots, \omega_C$. Assume S_w and S_b to be the intra-class and inter-class scatter matrices, respectively and can be defined as

$$S_w = \sum_{i=1}^{C} p_i \,\Xi\{(\mathbf{x} - \mu_i)(\mathbf{x} - \mu_i)^T \mid \omega_i\} = \sum_{i=1}^{C} p_i \Sigma_{\omega_i}; \tag{21}$$

$$S_b = \sum_{i=1}^{C} p_i (\mu - \mu_i)(\mu - \mu_i)^T; \tag{22}$$

where p_i is the a priori probability that a pattern belongs to class ω_i, $\mathbf{x}$ is a pattern vector, μ_i represents the sample mean vector of class ω_i, Σ_{ω_i} is the sample covariance matrix of class ω_i, and $\Xi\{\cdot\}$ calculates the expectation value. The overall mean vector (μ) for the entire data set is defined as

$$\mu = \sum_{i=1}^{C} p_i \mu_i. \tag{23}$$

Class separability [8], S, of a data set is defined as

$$S = trace(S_b^{-1} S_w). \tag{24}$$

A lower value of the separability measure S ensures that the classes are well separated.
Entropy:

The distance L_{pq} between the patterns $\mathbf{x}_p$ and $\mathbf{x}_q$ can be defined as:

$$L_{pq} = \left(\sum_{j=1}^{D} \left(\frac{x_{p,j} - x_{q,j}}{max_j - min_j} \right)^2 \right)^{1/2}, \tag{25}$$

where x_{pj} denotes the jth feature value of pattern $\mathbf{x}_p$, max_j and min_j are the maximum and the minimum values computed over all the patterns along the jth direction. Similarity between $\mathbf{x}_p$ and $\mathbf{x}_q$, represented as S_{pq}, can be defined as

$$S_{pq} = e^{-\alpha L_{pq}}; \tag{26}$$

where α is a positive constant. A possible value of $\alpha = \frac{-ln0.5}{\widehat{L}}$, where $\widehat{L}$ is the mean distance among patterns computed over the entire data set. Hence α is determined by the given data and can be calculated automatically.

Entropy [33] of a pattern $\mathbf{x}_p$ with respect to all other patterns is calculated as

$$E_p = - \sum_{\mathbf{x}_q \in \Upsilon}^{\mathbf{x}_p \neq \mathbf{x}_q} \left(S_{pq} log_2 S_{pq} + (1 - S_{pq}) log_2 (1 - S_{pq}) \right). \tag{27}$$

Here Υ is a set of all patterns. Entropy of overall data set is defined by

$$E = \sum_{\mathbf{x}_p \in \Upsilon} E_p = - \sum_{\mathbf{x}_p \in \Upsilon} \sum_{\mathbf{x}_q \in \Upsilon}^{p \neq q} \left(S_{pq} log_2 S_{pq} + (1 - S_{pq}) log_2 (1 - S_{pq}) \right). \tag{28}$$

It is to be noted that, entropy is less for stable configuration of patterns (data has well formed clusters), and is more for disordered configuration, i.e., data is uniformly distributed in the feature space.

6.3 Parameter Details

Experiments are conducted on three hyperspectral data sets, namely, Indian Pine, KSC and Botswana. Details about the data sets are given in Sect. 6.1. As already mentioned in the previous section, the clustering oriented KPCA based method first perform DBSCAN clustering technique on pixels to choose N representative patterns and then perform KPCA based transformation on the data set to reduce the dimensionality.

DBSCAN clustering algorithm uses two parameters, namely, minimum distance with respect to a point for which neighborhood is calculated (denoted as *Eps*) and the minimum number of points in an *Eps*-neighborhood of that point (denoted by *MinPts*). Ester et al. [29] suggested to use *MinPts* equal to 4 and used a method which considers the variation of the number of points with respect to their 4th nearest neighbor distance to calculate the value of *Eps*. Although higher values for *MinPts* have also been tested, it did not produce better results. The value of *Eps* is taken to be the location of the first valley of this graph. In the clustering oriented KPCA based strategy, *MinPts* and *Eps* are calculated in accordance to Ester et al. [29]. For Indian Pine data set, *Eps* value is 110, which is the 4th nearest neighbor distance of the first valley of the graph described at Ester et al. [29] with *MinPts* equal to 4. There are about 19 clusters of pixels and few isolated pixels which do not belong to any cluster. It is better to discard the isolated pixels and not consider them in formation of representative patterns, because KPCA is susceptible to noise. Generally, the principle for selecting representative patterns from each cluster is discussed in the proposed method section. But the percentage of total patterns which are selected for representative patterns, is needed to determine. Here, 2–12% of total patterns are selected for representative patterns for calculating kernel matrix of KPCA and the performance of the clustering oriented KPCA based method in terms of overall accuracy for 18 number of extracted features for Indian Pine data is depicted in Table 4. From the table, it is observed that 8–10% data patterns are sufficient for calculating kernel matrix. Similar observations are also found for the other data sets. So, 10% data from each cluster are selected for making representative patterns. So in the set of representative patterns, a small cluster has less number of pixels and vice verse. For example, the number of representative patterns for Indian Pine data is about 850.

To assess the performance of the above mentioned methods, classification of pixels is performed using transformed features. After completing the feature extraction, fuzzy *k*-NN based classification (in theory, any good classification algorithm can be used) is performed using the transformed features in 10-fold cross validation manner. 10-fold cross validation is a well-known technique for choosing training and testing data for classification. In this method, the whole data set is randomly partitioned into 10 blocks. Each time one block of data is treated as a testing data, and the remaining 9 blocks are training data. The whole process is repeated 10 times with different training and test data sets and the average overall accuracy is calculated. There may be overlapping of information between neighboring pixels of the hyperspectral

Table 4 Performance of the clustering oriented KPCA based method in terms of OA for 18 number of extracted features with different number of small representative samples for calculating kernel matrix of KPCA for Indian Pine data

N (%)	2	5	8	10	12
OA (%)	65.57	79.45	86.36	87.69	87.58

images. Fuzzy k-NN, rather than other classification techniques, is used to take care of the fuzziness present in the hyperspectral images.

The desired number of transformed features is not known apriori, because it varies with data set. In the present investigation, experiments are carried out for different number of features ranging from 4 to 30 with a step size of 2. Overall classification accuracy (OA), kappa coefficient (κ), class separability (S) and entropy (E) are calculated for the transformed set of features to assess the effectiveness of the feature extraction methods.

6.4 Analysis of Results

The cumulative eigenvalues of PCA, KPCA and clustering oriented KPCA based methods are depicted in Table 5 in percentage for Indian Pine data set. The cumulative eigenvalues represent the cumulative variance of the data [22, 34]. It shows that ninety five percent of cumulative variance of PCA is retained by the first six components, while KPCA and clustering oriented KPCA based methods need 14 to 18 components. In PCA most of the information content is retained in the first few features, where as, KPCA and clustering oriented KPCA based methods require more number of components.

The obtained OA and κ for Indian Pine data after applying fuzzy k-NN classifier over the transformed set of features by PCA, segmented PCA (SPCA), kernel PCA (KPCA) and clustering oriented KPCA based methods are given in Table 6. For PCA based method, OA becomes saturated when the number of transformed feature is 10 and after that it is stabilized. For KPCA and clustering oriented KPCA based methods, OA saturated at 18 and 16 number of features, respectively. It is due to

Table 5 Percentage of cumulative eigenvalues of principal components of PCA, KPCA and clustering oriented KPCA based methods for Indian Pine data

No. of PCs	PCA	KPCA	Clustering oriented KPCA
	(Cum.%)	(Cum.%)	(Cum.%)
2	72.32	57.74	63.18
4	85.89	68.11	73.74
6	96.69	76.41	81.54
8	98.37	83.37	88.62
10	99.06	87.16	91.97
12	99.23	89.78	94.24
14	99.33	91.71	95.86
16	99.37	93.48	96.92
18	99.42	94.82	97.60
20	99.46	95.84	98.15

Table 6 Overall accuracy and kappa coefficients of PCA, SPCA, KPCA and clustering oriented KPCA based methods for different number of extracted features for Indian Pine data

No. of features	PCA		SPCA		KPCA		Clustering oriented KPCA	
	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ
4	64.91	0.5952	52.19	0.4502	60.95	0.5516	**61.15**	**0.5539**
6	75.36	0.7167	66.24	0.6106	73.49	0.6948	**74.86**	**0.7102**
8	82.84	0.8031	74.31	0.7044	78.62	0.7541	**80.18**	**0.7713**
10	83.87	0.8144	78.92	0.7578	81.36	0.7859	**82.58**	**0.7997**
12	83.96	0.8154	82.31	0.7769	83.39	0.8092	**84.27**	**0.8187**
14	84.01	0.8159	83.68	0.8128	84.21	0.8180	**86.49**	**0.8436**
16	83.84	0.8140	84.78	0.8245	85.14	0.8287	**87.61**	**0.8559**
18	83.93	0.8149	85.16	0.8288	85.56	0.8332	**87.73**	**0.8573**
20	83.82	0.8137	85.02	0.8273	85.59	0.8336	**87.66**	**0.8565**
22	84.01	0.8159	84.98	0.8269	85.78	0.8356	**87.49**	**0.8546**
24	83.71	0.8124	85.13	0.8286	85.53	0.8328	**87.58**	**0.8556**
26	83.54	0.8102	85.01	0.8272	85.64	0.8341	**87.82**	**0.8584**
28	82.56	0.7995	84.91	0.8261	85.51	0.8324	**87.46**	**0.8542**
30	83.78	0.8132	85.10	0.8282	85.43	0.8316	**87.38**	**0.8533**

the fact that the number of principal components for PCA, KPCA and clustering oriented KPCA methods, for containing most of the variance of data, are 10, 18 and 16, respectively (shown in Table 5). It is noticed from Table 6 that Kernel PCA based methods (i.e., KPCA and clustering oriented KPCA) give better results than PCA and segmented PCA based methods. From Table 6, it is also observed that clustering oriented KPCA method achieves better results in terms of OA and κ for different number of transformed features. The reason behind this finding is that all the four methods transform the original set of features into a new set of features considering the maximum variance of data. Moreover, KPCA based methods incorporate the non linearity in transformation. The clustering oriented KPCA method gives better results than KPCA, because the representative patterns, for calculating kernel matrix for KPCA, are not selected randomly (like KPCA). The DBSCAN clustering technique is used to select the representative patterns so that it properly represents all the clusters of the data set, as well as, discard noisy pattern.

Figure 5 depicts the variation of average *OA* (in percentage) with number of features for all the methods used in the experiment. The graph corroborates to our earlier findings. For Indian Pine data, ground truth image with 16 classes is shown in Fig. 6, where different colors are used to distinguish the pixels among classes. Figure 7a–d shows the pictorial representation of the classified image with the best subset of features extracted using PCA, SPCA, KPCA and clustering oriented KPCA based techniques, correspondingly. A view of the classified images show that the clustering oriented KPCA based technique transforms a better set of features for classification

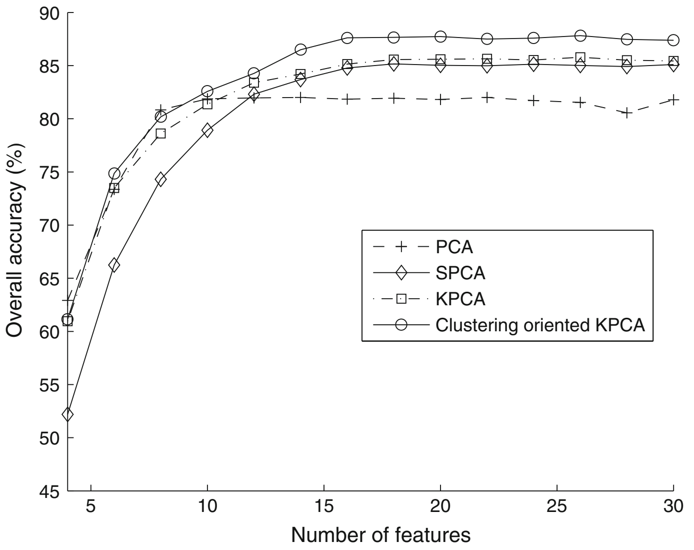


Fig. 5 Comparison of the performance of PCA, SPCA, KPCA and clustering oriented KPCA based methods in terms of overall accuracy with respect to the number of features used for Indian Pine data

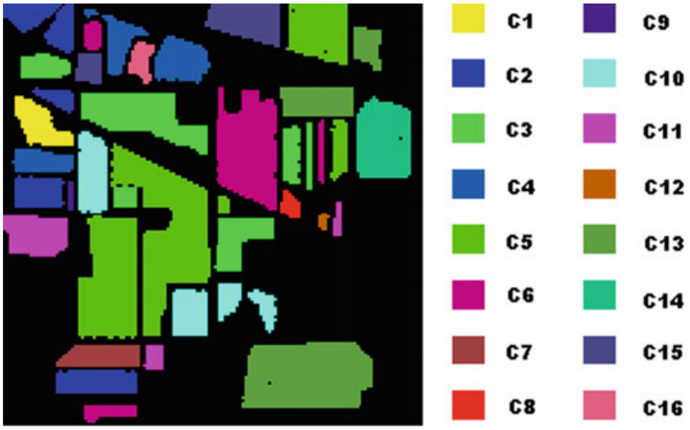


Fig. 6 Ground truth image of Indian Pine data

of given hyperspectral images as compared to other methods. It is clearly observed that the classified Indian Pine image with transformed feature set using clustering oriented KPCA based method has very less misclassified pixels compared to other methods. Table 10 contains the optimum value of OA, κ, S and E for all three hyperspectral data sets for all four methods. From this table, it is noticed that clustering oriented KPCA based method gives less value of S and E, which is better with respect to the other three methods used in our experiments. It shows that the clustering oriented KPCA method transforms better subset of features which gives well separated classes as well as stable configuration of patterns compared to other methods.

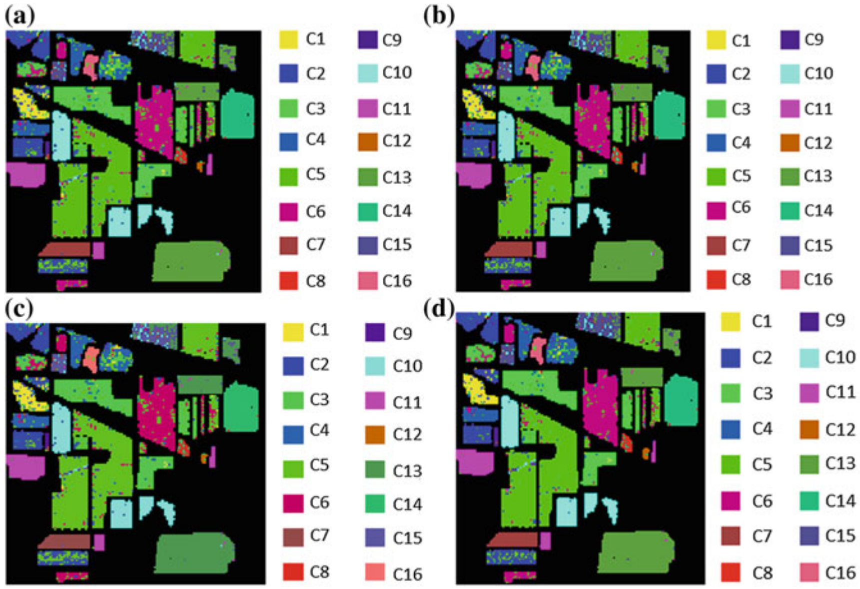


Fig. 7 Classified images of Indian Pine data with extracted feature set using (a) PCA, (b) SPCA, (c) KPCA, and (d) clustering oriented KPCA based methods

Table 7 CPU time for PCA, SPCA, KPCA and clustering oriented KPCA based methods using Indian Pine data

	PCA	SPCA	KPCA	Clustering oriented KPCA
CPU time (s)	15.29	48.24	79.12	61.31

For comparing the computational costs, using an Intel(R) Core(TM) i7 2600 CPU @ 3.40-GHz processor and an Indian Pine image with 185 features of 145 × 145 pixels, clustering oriented KPCA method required about 61.31 s. Programs are developed in C. Table 7 gives a simple quantitative analysis of the computational cost of each method for Indian Pine data. The clustering oriented KPCA method takes much less time than KPCA, where all the patterns are used for kernel matrix, but it takes little more time than PCA based methods (i.e., PCA and SPCA).

Overall accuracy (OA) and kappa coefficient (κ) for KSC and Botswana data sets are put in Tables 8 and 9, respectively. From the table, it is observed that clustering oriented KPCA based method is producing better results than the other methods for both the data sets. A variation of OA for these methods with the number of transformed features are depicted graphically in Figs. 8 and 9, respectively, for KSC and Botswana data. Results for these data sets corroborate to our earlier findings. It is also observed that KPCA based transformation (KPCA and clustering oriented KPCA) are found to be better than PCA based methods (PCA and Segmented PCA).

Table 8 Overall accuracy and kappa coefficients of PCA, SPCA, KPCA and clustering oriented KPCA based methods for different number of extracted features for KSC data

No. of features	PCA		SPCA		KPCA		Clustering oriented KPCA	
	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ
4	69.44	0.6483	59.07	0.5311	65.84	0.6054	**67.21**	**0.6213**
6	78.37	0.7513	73.41	0.6943	77.31	0.7390	**78.19**	**0.7493**
8	86.21	0.8407	79.87	0.7703	85.37	0.8311	**85.81**	**0.8359**
10	88.50	0.8658	85.31	0.8304	87.21	0.8514	**88.47**	**0.8655**
12	88.72	0.8682	89.01	0.8714	88.79	0.8690	**89.21**	**0.8736**
14	88.84	0.8695	90.10	0.8835	89.38	0.8755	**89.98**	**0.8822**
16	88.89	0.8701	89.92	0.8815	90.61	0.8898	**90.72**	**0.8910**
18	88.69	0.8679	90.03	0.8827	90.72	0.8910	**91.92**	**0.9042**
20	88.42	0.8649	90.21	0.8847	90.89	0.8929	**91.93**	**0.9044**
22	88.51	0.8659	90.16	0.8841	90.81	0.8920	**91.87**	**0.9036**
24	88.10	0.8614	89.77	0.8798	90.64	0.8901	**91.89**	**0.9039**
26	88.27	0.8633	90.01	0.8825	90.32	0.8859	**91.75**	**0.9023**
28	88.48	0.8656	89.98	0.8822	90.57	0.8893	**91.87**	**0.9036**
30	88.46	0.8654	90.08	0.8833	90.61	0.8898	**91.82**	**0.9031**

If higher order statistics of a hyperspectral data set are considered with variance of data, then the methods give better results than others.

Class separability and entropy values are also calculated for both the KSC and Botswana data sets. Results of these data sets provide similar findings with the results of Indian Pine data. Table 10 incorporates the optimum values (for all the three data sets) in terms of OA, κ, S and E. The optimum value of all the methods are achieved in different number of extracted features which are also depicted in this table. The different numbers of extracted features for different methods (for optimum results) are in between 14 and 22, because different methods follow different extraction principles. The best results are marked in bold. This table also confirms the fact that clustering oriented KPCA based feature extraction algorithm gives better transformed set of features for classification than the other methods used in our experiment.

It also has been noticed that richness of the information of hyperspectral data is not fully handled using only variance of the data (by PCA method), it needs variance as well as higher order statistics of the data (like KPCA based methods). The KPCA based methods can extract more information from the hyperspectral data than the conventional PCA. Also, a proper choice of representative patterns for kernel matrix calculation, like clustering oriented KPCA based methods, produces better subset of features.

Table 9 Overall accuracy and kappa coefficients of PCA, SPCA, KPCA and clustering oriented KPCA based methods for different number of extracted features for Botswana data

No. of features	PCA		SPCA		KPCA		Clustering oriented KPCA	
	OA(%)	κ	OA(%)	κ	OA(%)	κ	OA(%)	κ
4	69.82	0.6552	58.31	0.5228	66.12	0.6085	**68.02**	**0.6302**
6	78.51	0.7529	72.42	0.6834	76.53	0.7244	**77.23**	**0.7321**
8	86.94	0.8486	79.98	0.7715	83.32	0.8084	**85.21**	**0.8293**
10	88.38	0.8645	86.31	0.8416	86.17	0.8439	**88.19**	**0.8624**
12	89.30	0.8746	87.83	0.8533	89.92	0.8815	**90.71**	**0.8909**
14	89.32	0.8748	89.39	0.8756	90.67	0.8904	**91.32**	**0.8976**
16	89.26	0.8740	90.43	0.8872	91.23	0.8966	**91.78**	**0.9026**
18	89.33	0.8749	90.72	0.8908	91.25	0.8988	**92.68**	**0.9125**
20	89.17	0.8730	90.74	0.8910	91.19	0.8961	**92.89**	**0.9151**
22	89.22	0.8735	90.63	0.8894	91.31	0.8975	**92.71**	**0.9130**
24	89.23	0.8736	90.68	0.8899	91.07	0.8968	**92.34**	**0.9089**
26	89.08	0.8719	90.21	0.8847	91.24	0.8987	**92.21**	**0.9075**
28	89.14	0.8726	90.19	0.8845	91.17	0.8979	**92.46**	**0.9103**
30	89.21	0.8734	90.52	0.8881	91.21	0.8983	**92.61**	**0.9119**

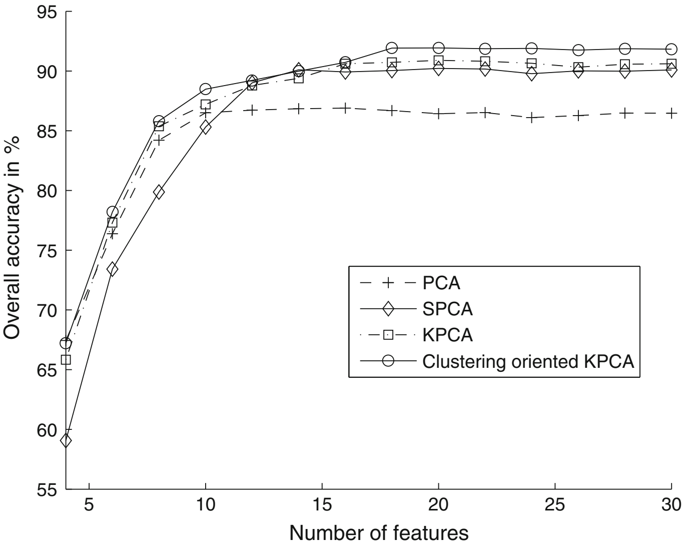


Fig. 8 Comparison of the performance of PCA, SPCA, KPCA and clustering oriented KPCA based methods in terms of overall accuracy with respect to the number of features used for KSC data

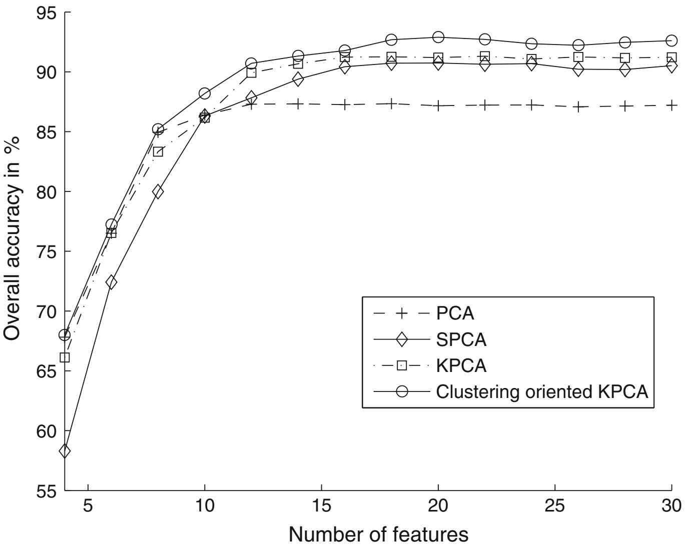


Fig. 9 Comparison of the performance of PCA, SPCA, KPCA and clustering oriented KPCA based methods in terms of overall accuracy with respect to the number of features used for Botswana data

Table 10 Comparison of feature extraction methods for hyperspectral data sets

Data set used	Method	Selected feature no.	Evaluation criterion			
			E	*S*	*OA*	*κ*
Indian Pine D = 185	PCA	14	0.6013	0.2659	84.01	0.8159
	SPCA	18	0.5929	0.2607	85.16	0.8288
	KPCA	22	0.5815	0.2559	85.78	0.8356
	Clustering oriented KPCA	18	**0.5567**	**0.2413**	**87.82**	**0.8584**
KSC D = 176	PCA	16	0.5637	0.1307	88.89	0.8701
	SPCA	20	0.5529	0.1279	90.21	0.8847
	KPCA	20	0.5496	0.1241	90.89	0.8929
	Clustering oriented KPCA	20	**0.5403**	**0.1193**	**91.93**	**0.9044**
Botswana D = 145	PCA	16	0.4734	0.1002	89.33	0.8749
	SPCA	20	0.4561	0.0913	90.74	0.8910
	KPCA	22	0.4493	0.0896	91.25	0.8988
	Clustering oriented KPCA	20	**0.4376**	**0.0809**	**92.89**	**0.9151**

7 Conclusions

PCA and KPCA based feature extraction techniques for hyperspectral images in unsupervised manner has been presented in this chapter, which transform the original data to a lower dimensional space. PCA is a linear transformation, whereas KPCA is non linear in nature and advantageous to attain the higher order statistics of data. In clustering oriented KPCA, the DBSCAN clustering technique is used to select proper training patterns for calculating kernel matrix for KPCA. To measure the effectiveness of these methods, four evaluation measures (namely, overall accuracy, kappa coefficient, class separability and entropy value) have been used. It is observed from the results that clustering oriented KPCA technique has a significant improvement, and a more consistent and steady behavior for different hyperspectral image data sets (Indian Pine, KSC and Botswana data) with respect to the other methods, i.e., PCA, SPCA and KPCA based methods in terms of all four evaluation measures.

It can be concluded from the above mentioned experimental results that clustering oriented KPCA based method gives better performance with respect to other methods, because the technique considers variance of the data set as well as other higher order statistics by using kernel PCA based transformation. The method also takes necessary steps for choosing the representative patterns as well as avoid noisy patterns for calculating kernel matrix of KPCA, which is a proper representation of the original data set by using DBSCAN clustering algorithm.

References

1. Varshney, P.K., Arora, M.K.: Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data, 2nd edn. Springer, Berlin (2004)
2. Landgrebe, D.: Hyperspectral image data analysis. IEEE Signal Processing Magazine, pp. 17–28, 2002
3. Manolakis, D., Marden, D., Shaw, G.A.: Hyperspectral image processing for automatic target detection applications. Lincoln Lab. J. **14**(1), 79–116 (2003)
4. Shippert, P.: Introduction to hyperspectral image analysis. Online Journal of Space Communication, 2003
5. Shippert, P.: Why use hyperspectral imagery? Photogrammetric Engineering and Remote Sensing, pp. 377–380, April 2004
6. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Macine Intelligence **22**(1), 4–37 (2000)
7. Bishop, C.M.: Neural Networks for Pattern Recognition, 1st edn. Oxford University Press, New Delhi (1995)
8. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach, 1st edn. Prentice-Hall International, New Delhi (1982)
9. Ghosh, A., Datta, A., Ghosh, S.: Self-adaptive differential evolution for feature selection in hyperspectral image data. Appl. Soft Comput. **13**(4), 1969–1977 (2013)
10. Jia, X., Kuo, B.-C., Crawford, M.M.: Feature mining for hyperspectral image classification. Proc. IEEE **101**(3), 676–697 (2013)
11. Datta, A., Ghosh, S., Ghosh, A.: Band elimination of hyperspectral imagery using partitioned band image correlation and capacitory discrimination. Int. J. Remote Sens. **35**(2), 554–577 (2014)

12. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Acacdemic Press, San Diego (1990)
13. Datta, A., Ghosh, S., Ghosh, A.: Combination of clustering and ranking techniques for unsupervised band selection of hyperspectral images. IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. **8**(6), 2814–2823 (2015)
14. Jia, S., Ji, Z., Shen, L.: Unsupervised band selection for hyperspectral imagery classification without manual band removal. IEEE J. Sel. Top. Appl. Earth Observations Remote Sens. **5**(2), 531–543 (2012)
15. Datta, A., Ghosh, S., Ghosh, A.: Wrapper based feature selection in hyperspectral image data using self-adaptive differential evolution. In: *Proceedings of the International Conference on Image Information Processing*, pp. 1–6 (2011)
16. Datta, A., Ghosh, S., Ghosh, A.: Clustering based band selection for hyperspectral images. In: *Proceedings of the International Conference on Communications, Devices and Intelligent Systems*, pp. 101–104 (2012)
17. Mojaradi, B., Abrishami-Moghaddam, H., Zoej, M.J.V., Duin, R.P.W.: Dimensionality reduction of hyperspectral data via spectral feature extraction. IEEE Trans. Geosci. Remote Sens. **47**(7), 2091–2105 (2009)
18. Datta, A., Ghosh, S., Ghosh, A.: Band elimination of hyperspectral imagery using correlation of partitioned band image. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pp. 412–417 (2013)
19. Jain, A., Zongker, D.: Feature selection: Evaluation, application, and small sample performance. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 153–189 (1997)
20. Jia, X., Richards, J.A.: Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification. IEEE Trans. Geosci. Remote Sens. **37**, 538–542 (1999)
21. Datta, A., Ghosh, S., Ghosh, A.: Supervised band extraction of hyperspectral images using partitioned maximum margin criterion. IEEE Geosci. Remote Sens. Lett. **14**(1), 82–86 (2017)
22. Fauvel, M., Chanussot, J., Benediktsson, J.A.: Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. J. Adv. Sig. Process. **2009**, 1–14 (2009)
23. Datta, A., Ghosh, S., Ghosh, A.: Maximum margin criterion based band extraction of hyperspectral imagery. In *Proceedings of the Fourth International Conference on Emerging Applications of Information Technology*, pp. 300–304 (2014)
24. Kuo, B.-C., Landgrebe, D.A.: Nonparametric weighted feature extraction for classification. IEEE Trans. Geosci. Remote Sens. **42**, 1096–1105 (2004)
25. Schölkopf, B., Smola, A., Müller, K.-R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**(5), 1299–1319 (1998)
26. Datta, A., Ghosh, S., Ghosh, A.: Unsupervised band extraction for hyperspectral images using clustering and kernel principal component analysis. Int. J. Remote Sens. **38**(3), 850–873 (2017)
27. Rodarmel, C., Shan, J.: Principal component analysis for hyperspectral image classification. Surveying Land Inf. Syst. **62**(2), 115–122 (2002)
28. Richards, J.A., Jia, X.: Remote Sensing Digital Image Analysis: An Introduction, 1st edn. Springer, New York (1999)
29. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231 (1996)
30. Jimenez, L.O., Landgrebe, D.A.: Hyperspectral data analysis and supervised feature reduction via projection pursuit. IEEE Trans. Geosci. Remote Sens. **37**, 2653–2667 (1999)
31. Ham, J., Chen, Y., Crawford, M.M., Ghosh, J.: Investigation of the random forest framework for classification of hyperspectral data. IEEE Trans. Geosci. Remote Sens. **43**(3), 492–501 (2005)
32. Congalton, R.G., Green, K.: Assessing the Accuracy of Remotely Sensed Data, 2nd edn. CRC Press, London (2009)

33. Yao, J., Dash, M., Tan, S.T., Liu, H.: Entropy-based fuzzy clustering and fuzzy modeling. Fuzzy Sets Syst. **113**, 381–388 (2000)
34. Licciardi, G., Marpu, P.R., Chanussot, J., Benediktsson, J.A.: Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles. IEEE Geosci. Remote Sens. Lett. **9**(3), 447–451 (2012)