International Journal of Remote Sensing
Vol. 00, No. 00, DD Month 200x, 1–10

Classification of hyperspectral remote-sensing images using discriminative linear projections

Lior Weizman and Jacob Goldberger*
School of Engineering, Bar-Ilan University, Ramat-Gan 52900, Israel

(*Received 00 Month 200x; in final form 00 Month 200x*)

In this study we apply a variant of a recently proposed linear subspace method, the Neighbourhood Component Analysis (NCA), to the task of hyperspectral classification. The NCA algorithm explicitly utilizes the classification performance criterion to obtain the optimal linear projection. NCA assumes nothing about the form of each class and the shape of the separating surfaces. In some cases we would like to weight the penalty function for different types of misclassifications of the algorithm. A modification of the NCA cost function is introduced for this case. Experimental studies are conducted on hyperspectral images acquired by two sensors: the Airborne Visible/Infrared Imaging Spectroradiometer (AVIRIS) and AISA-EAGLE. Experimental results confirm the superiority of the NCA classifier in the context of hyperspectral data classification over methodologies that were previously suggested.

Keywords - Classification, hyperspectral images, remote sensing, linear projection, NCA.

1 Introduction

A remarkable increase in spectral resolution has led to imaging sensors that can gather data in hundreds of contiguous narrow spectral bands to generate hyperspectral images. This ability of imaging sensors to acquire the reflectance spectrum of a pixel in significant detail, leads to substantial differences in the reflectance values of the pixels belonging to disparate materials on the Earth's surface. The automatic analysis of hyperspectral data, however, is not a trivial task. One of the major difficulties in hyperspectral classification is the high dimensionality of the data. The enormous number of features in a hyperspectral image is often a major drawback. A hyperspectral image generally consists of thousands of pixels over hundreds of spectral bands. Classification of this tremendous amount of data, is time consuming and requires significant computational effort, which may not be possible in many applications. Therefore, the traditional but still common approach for classification of a hyperspectral image consists of a feature extraction procedure followed by classification algorithms.

A basic and commonly used method for supervised feature extraction is the Fisher linear discriminant analysis (LDA). This method tries to find the feature space in which the trace of the inverse of the within-class scatter matrix times the between-class scatter matrix is maximized. Another linear transformation for feature extraction is the decision boundary feature extraction (DBFE) method (Lee *et al.* 1993). This method is based on the decision boundary between classes in order to define discriminately redundant features and discriminant informative features. Based on the LDA, the nonparametric weighted feature extraction (NWFE) method was developed (Kou *et al.* 2004). This method puts different weights on every sample in order to deemphasize samples close to the boundary in relation to those far away from the boundary. A regularized version of the NWFE that can handle better small sample size was recently suggested by Kou *et al.* (2007).

In addition to feature extraction, appropriate classification algorithms should be applied to the reduced dimension data, in order to produce accurate land-cover maps for the desired hyperspectral image. The K-nearest neighbour classifier is a simple but surprisingly accurate method. The relevant component analysis (RCA) (Bar-Hillel *et al.* 2005) can be used to calculate an appropriate distance measure in the reduced space that ignores the within class variability. There are, however publications on classification methods

Corresponding author. Email: goldbej@eng.biu.ac.il

International Journal of Remote Sensing
ISSN 0143-1161 print / ISSN 1366-5901 online
http://www.tandf.co.uk/journals
DOI: 10.1080/0143116YYxxxxxxxx

that do not follow the approach of feature extraction followed by a K-NN classifier. For example, the work of Melgani *et al.* (2004) that is based on applying the SVM classifier (Schölkopf *et al.* 2001) to hyperspectral images.

This paper will focus on implementation and adaptation of a recently proposed method for feature extraction, the Neighbourhood Component Analysis (NCA) (Goldberger *et al.* 2004) to the problem of hyperspectral image classification. The contribution of this paper is the application of the NCA method to the task of hyperspectral classification yielding improved results over previously suggested methods. Another contribution of the paper is a novel variant of the NCA that can be more adequate for remote-sensing classification tasks. The main advantage of the NCA method is that it explicitly utilizes the classification performance criterion to obtain the optimal linear projection. In the proposed method, the optimal transformation is selected such that using the Euclidean distance in the transformed space yields optimal classification results. We show that the adaptation of the NCA to the task of hyperspectral classification leads to an improvement in comparison to the other common subspace methods used today.

This paper is organized as follows: The theoretical development of NCA followed by a modified version of it is presented in Section 2. Experimental results on real hyperspectral images, acquired by two different sensors, are presented in Section 3. Conclusions are discussed in Section 4.

2 Neighbourhood Component Analysis

In this section we review the NCA algorithm (Goldberger *et al.* 2004) and focus on a new variant found to be suitable for classification of hyperspectral images. We begin with a labelled data set consisting of n real-valued input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathcal{R}^D$ and corresponding class labels $c_1, ..., c_n$. In the case of hyperspectral images, the vectors are the spectral signatures of the pixels, and the labels are the land-cover classes. We want to find a low-dimensional linear transformation $\mathbf{A} : \mathcal{R}^D \rightarrow \mathcal{R}^d$ that maximizes the performance of nearest neighbour classification in the reduced space. Instead of imposing unjustified structure on the data points we can directly use this objective as the optimization criterion. Ideally, we would like to optimize performance on future test data, but as we do not know the true data distribution we instead attempt to optimize leave-one-out (LOO) performance on the training data. Given a finite set of linear transformations to choose from, we can easily select the best one, namely the one that minimizes the number of classification errors. Denote the point nearest to the point i in the reduced space by $NN(i)$, i.e.

$$NN(i) = \arg\min_{k \neq i} ||\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k||$$

The score we want to maximize, therefore, is the number of training points that are correctly classified. In other words, we want to find the linear transformation that maximizes the size of the following set: $\{i|c_i = c_{NN(i)}\}$.

The nearest-neighbour classification error, however, is a discontinuous function of the transformation $\mathbf{A}$, given that an infinitesimal change in $\mathbf{A}$ may change the neighbour graph and thus affect LOO classification performance by a finite amount. Hence we can not use this optimization criterion in our case where there is a continuously parameterized family of linear transformations which must be searched. Instead, we adopt a more well-behaved measure of nearest-neighbour performance, by introducing a differentiable cost function based on stochastic ("soft") neighbour assignments in the transformed space. In particular, each point i selects another point j as its neighbour with some probability p_{ij}, and inherits its class label from the point it selects. The probability p_{ij} is defined using a soft nearest-neighbour over Euclidean distances in the transformed space:

$$p_{ij}(\mathbf{A}) = \frac{\exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2)}{\sum_{k \neq i} \exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k\|^2)} \quad , \quad p_{ii}(\mathbf{A}) = 0 \tag{1}$$

In eq. (1), the probability of selecting a neighbour point is inversely proportional to its distance from

the point i. In comparison, in the case of a deterministic nearest-neighbour criterion, the probabilistic assignment is reduced to:

$$p_{ij}(\mathbf{A}) = \begin{cases} 1 & j = NN(i) \\ 0 & \text{else} \end{cases} \tag{2}$$

Note that the norm of the matrix $\mathbf{A}$ controls the softness of the neighbour assignments. By replacing $\mathbf{A}$ with $\lambda\mathbf{A}$, it can easily be shown that as λ tends to infinity, the probabilistic assignment (1) is reduced to a deterministic nearest-neighbour assignment (2) in the same transformed space. We use the notation $C_i = \{j|c_i = c_j\}$ to denote denotes the set of points in the same class as i. Under the stochastic selection rule (1), we can compute the probability $\sum_{j\in C_i} p_{ij}$ that a point i will be correctly classified.

The objective function we maximize is as follows:

$$C(\mathbf{A}) = \sum_i \log(\sum_{j\in C_i} p_{ij}) = \sum_i \log\left(\sum_{j\in C_i} \frac{\exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2)}{\sum_{k\neq i}\exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k\|^2)}\right) \tag{3}$$

Maximizing this objective would correspond to maximizing the probability of obtaining a *perfect (error free) classification of the entire training set*. Alternatively we can define the cost function as $\sum_i \sum_{j\in C_i} p_{ij}$. Maximizing this variant would correspond to maximizing the expected number of correct classifications. We have found empirically that both definition yields classification algorithm with similar performance.

Note that since $\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{A}^\top\mathbf{A})(\mathbf{x}_i - \mathbf{x}_j)$, the optimization criterion $C(\mathbf{A})$ depends only on $\mathbf{A}^\top\mathbf{A}$. Instead of looking for an optimal linear projection, we can equivalently look for an optimal d-dimensional Mahalanobis distance in the original space. Denoting the Mahalanobis distance induced from $\mathbf{\Sigma} = \mathbf{A}^\top\mathbf{A}$ by $d_\Sigma(\mathbf{x},\mathbf{y}) = \frac{1}{2}(\mathbf{x}-\mathbf{y})^\top\mathbf{\Sigma}(\mathbf{x}-\mathbf{y})$, we can write the cost function $C(\mathbf{A})$ as a function of the semi-positive definite matrix $\mathbf{\Sigma}$:

$$C(\Sigma) = \sum_i \log\left(\sum_{j\in C_i} \frac{\exp(-d_\Sigma(\mathbf{x}_i,\mathbf{x}_j))}{\sum_{k\neq i}\exp(-d_\Sigma(\mathbf{x}_i,\mathbf{x}_j))}\right) \tag{4}$$

Note that every orthogonal matrix $\mathbf{R}_{d\times d}$ yields a transformation $\mathbf{R}\cdot\mathbf{A}$ that is completely equivalent to $\mathbf{A}$ in the sense that $C(\mathbf{R}\mathbf{A}) = C(\mathbf{A})$. To keep the representation parsimonious we can use the Choleski decomposition representation by forcing the entries of $\mathbf{A}$ below the main diagonal to be zero and the entries on the diagonal to be non-negative. This makes the representation of $\mathbf{A}$ unique.

In some cases we would like to weight the penalty for different types of misclassification of the algorithm. For example, there are cases where misclassification between different types of vegetation is not as important as misclassification of road as soil. Therefore, a modification of the cost function defined in (3) is performed as follows: Assuming that there are s classes, then the matrix $\mathbf{M}_{s\times s}$ is defined as part of the problem setting, whose elements, $\mathbf{M}(c_1, c_2)$, varies from 0 to 1. These elements represent the score given for classification of pixel from class c_1 as class c_2. The score 1 stands for perfect classification and 0 stands for a complete misclassification. In the usual classification setup $\mathbf{M}$ is the identity matrix where either the classification is correct (score 1) or wrong (score 0).

Applying the proposed modification to the cost function (3), yields the following modified cost function:

$$D(\mathbf{A}) = \sum_i \log(\sum_j \mathbf{M}(c_i, c_j)p_{ij}) \tag{5}$$

We coin this variant of the NCA algorithm weighted-NCA. Note that if $\mathbf{M}$ is the identity matrix, the cost function $D(\mathbf{A})$ coincides with the un-weighted original cost function $C(\mathbf{A})$. Otherwise, the cost function $D(\mathbf{A})$ is a generalization of $C(\mathbf{A})$. Differentiating D with respect to the transformation matrix $\mathbf{A}$ yields a gradient rule which can be used for learning. Although expressions (3) and (5) look very complex we show

next that we can obtain an efficiently computed expression for the gradient. To simplify the derivation we use the notations $f_{ij}(\mathbf{A}) = \exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2)$ and $f_i = \sum_{k\neq i} f_{ik}$. It can easily seen from definition (1) that $p_{ij} = f_{ij}/f_i$. Given that the gradient of the quadratic expression $\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^\top \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)$ is $2\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$, the gradient of f_{ij} is:

$$\frac{\partial f_{ij}(\mathbf{A})}{\partial \mathbf{A}} = -f_{ij}\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top = -f_{ij}\mathbf{\Delta}_{ij} \tag{6}$$

where $\mathbf{\Delta}_{ij} = \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$. Hence:

$$\begin{aligned}\frac{\partial p_{ij}(\mathbf{A})}{\partial \mathbf{A}} &= \frac{\partial}{\partial \mathbf{A}}(\frac{f_{ij}}{f_i}) = \frac{f'_{ij}f_i - f_{ij}f'_i}{f_i^2} = \frac{f'_{ij}}{f_i} - p_{ij}\frac{f'_i}{f_i} = -p_{ij}\mathbf{\Delta}_{ij} - p_{ij}\sum_{k\neq i}\frac{f'_{ik}}{f_i} \\ &= -p_{ij}\mathbf{\Delta}_{ij} + p_{ij}\sum_{k\neq i}\frac{f_{ik}}{f_i}\mathbf{\Delta}_{ik} = p_{ij}\left(\sum_k p_{ik}\mathbf{\Delta}_{ik} - \mathbf{\Delta}_{ij}\right)\end{aligned} \tag{7}$$

The gradient of the cost function $D(\mathbf{A})$ defined in (5) is:

$$\frac{\partial D(\mathbf{A})}{\partial \mathbf{A}} = \sum_i \frac{\sum_j \mathbf{M}(c_i, c_j)p'_{ij}}{\sum_j \mathbf{M}(c_i, c_j)p_{ij}} \tag{8}$$

Substituting equation (7) in equation (8), we obtain:

$$\begin{aligned}\frac{\partial D(\mathbf{A})}{\partial \mathbf{A}} &= \sum_i \frac{1}{\sum_j \mathbf{M}(c_i, c_j)p_{ij}}\left(\sum_k p_{ik}\mathbf{\Delta}_{ik}(\sum_j \mathbf{M}(c_i, c_j)p_{ij}) - \sum_j \mathbf{M}(c_i, c_j)p_{ij}\mathbf{\Delta}_{ij}\right) \\ &= \sum_i \left(\sum_k p_{ik}\mathbf{\Delta}_{ik} - \frac{\sum_j \mathbf{M}(c_i, c_j)p_{ij}\mathbf{\Delta}_{ij}}{\sum_j \mathbf{M}(c_i, c_j)p_{ij}}\right)\end{aligned} \tag{9}$$

We can derive in a similar way the gradient of the un-weighted cost function $C(\mathbf{A})$:

3 Experimental Results

This Section exhibits the results of three experiments that are based on two hyperspectral datasets. In the experiments the NCA results are compared to other well known feature extraction and classification methods. The first hyperspectral dataset used in our experiments is a section of a scene taken over northwest Indian Pines in Indiana by the AVIRIS sensor in 1992 (see AVIRIS 1992). From the 220 spectral channels acquired by the AVIRIS sensor, 20 channels were discarded as they were affected by atmospheric interference. The nine land-cover classes in the image consist of 9345 pixels. In the first experiment, the data was divided equally into training samples (used for teaching the classifiers) and test samples (exploited for assessing their accuracies). The setting of this experiment is identical to the first experiment published by Melgani *et al.* (2004). In the second experiment, only 10% of the data was used for training while the rest was used as a test set. For the third experiment we used a subset of a scene acquired by the AISA-EAGLE sensor over southern Israel in 2004. This dataset has spatial dimensions of 294×850 pixels, and 59 spectral bands. The ground truth labeling of this image consists of 8 major land-cover classes. Ten percent of the pixels in this dataset were used as training samples, while the other pixels in this set were used for testing. A detailed description of the dataset for the first experiments can be found in Melgani *et al.* (2004), while the datasets used for second and third experiment are presented in Table 1 and Table 2, respectively.

Training:
Input: A set of n labelled points: $\{\mathbf{x}_i \in R^D, c_i\}$, a class-confusion matrix $\mathbf{M}$ and the reduced dimension d.

Output: A linear projection $\mathbf{A}_{d\times D} : R^D \rightarrow R^d$ that maximizes the objective:

$$D(\mathbf{A}) = \sum_i \log\Big(\sum_{j\in C_i} \mathbf{M}(c_i, c_j) p_{ij}\Big) \quad \text{with} \quad p_{ij}(\mathbf{A}) = \frac{\exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2)}{\sum_{k\neq i}\exp(-\frac{1}{2}\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k\|^2)}$$

Method:

- Apply optimization algorithm (e.g. conjugate-gradient) to find the maximum of $D(\mathbf{A})$. The derivative of the $D(\mathbf{A})$ has the following simple form:

$$\frac{\partial D}{\partial \mathbf{A}} = \sum_i \left(\sum_k p_{ik}\mathbf{\Delta}_{ik} - \frac{\sum_j \mathbf{M}(c_i, c_j) p_{ij}\mathbf{\Delta}_{ij}}{\sum_j \mathbf{M}(c_i, c_j) p_{ij}} \right) \qquad \text{where} \qquad \mathbf{\Delta}_{ij} = \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$$

- Store $\mathbf{A}$ and the projected training set $\{\mathbf{A}\mathbf{x}_i, c_i\}$.

Testing:
classify unlabelled point $\mathbf{x}$.

- find the nearest neighbour training point $i = \arg\min_j \|\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{x}_j\|$ and set the label of x to be c_i.

Figure 1. The proposed featureexraction+classfication method based on a linear subspace-projection learning algorithm.

Table 1. Number of training and test samples used in experiment 2 (AVIRIS).

Class	Training	Test
w_1 - Corn-no till	143	1291
w_2 - Corn-min till	83	791
w_3 - Grass/Pasture	49	448
w_4 - Grass/Trees	75	672
w_5 - Hay-windrowed	49	440
w_6 - Soybean-no till	97	871
w_7 - Soybean-min till	247	2221
w_8 - Soybean-clean till	61	553
w_9 - Woods	129	1165
Total	933	8412

3.1 *Performance of the NCA*

In our experiments we examined the performance of several feature extraction methods, including NCA. Each feature extraction method extracts features. The classification accuracy of using a variant number of features is computed by using the NN classifier. The following feature-extraction methods were tested in the experiments: LDA, RCA (Bar-Hillel *et al.* 2005), NWFE (Kou *et al.* 2004), DBFE (Lee *et al.* 1993) and NCA. In order to compare classification results in the feature subspace with a classification in the original feature space, the SVM-Linear and SVM-RBF methods were also tested. In Melgani *et al.* (2004), the SVM exhibited the best performance over other classifiers in the original feature space. Therefore, in order to make an unbiased comparison, we used the SVM method in our experiments with the same parameters as reported in Melgani *et al.* (2004), and without performing any prior feature extraction/selection method.

A visual presentation of the datasets, the ground truth and the classification results using the NCA method is shown in Figs. 2 and 3 for experiments 1 and 3, respectively.

The results of our experiments in terms of classification accuracy provided by the different methods are summarized in Table 3. The number in the parentheses is the number of features used for obtaining the

Table 2. Number of training and test samples used in experiment 3 (AISA-EAGLE).

Class	Training	Test
w_1 - Road	726	6542
w_2 - Pure vegetation	6331	56 979
w_3 - Mixed vegetation with soil	130	1177
w_4 - Greenhouse with crops	2441	21 971
w_5 - Greenhouse without crops	1061	9553
w_6 - Fine dune sand	13 818	124 362
w_7 - Coarse sand	419	3777
w_8 - Buildings	61	552
Total	24 987	224 913

(a) (b) (c)

Figure 2. Classification results of experiment 1. (a) The hyperspectral image in true color, (b) The ground truth, (c) The NCA classification results.

Table 3. Best overall accuracies, achieved on the test set by different classifiers.

Method	Experiment 1	Experiment 2	Experiment 3
NCA	94.23 (14)	88.32 (14)	91.09 (10)
SVM-Linear	91.26	85.77	92.40
SVM-RBF	91.04	85.69	92.29
NWFE	92.70 (14)	88.19 (9)	89.96 (14)
DBFE	89.87 (14)	N/A	89.88 (14)
RCA	88.62 (7)	82.13 (8)	90.75 (7)
LDA	83.73 (14)	76.15 (9)	89.17 (14)

Table 4. The Kappa statistics values, achieved on the test set by different classifiers for the experiment presented in Table 3.

Method	Experiment 1	Experiment 2	Experiment 3
NCA	0.93	0.86	0.86
SVM-Linear	0.90	0.83	0.88
SVM-RBF	0.89	0.83	0.87
NWFE	0.91	0.86	0.84
DBFE	0.88	N/A	0.84
RCA	0.87	0.79	0.85
LDA	0.81	0.72	0.82

best classification accuracy. In addition to the overall accuracy we are also able to compute the Kappa statistics in order to asses the classification quality. Table 4 presents the Kappa statistics values for the classification results presented in Table 3. Note that the DBFE algorithm results are not available for the second experiment. This is due to the fact that the DBFE requires that the number of training pixels in every class should be greater than the number of bands in the image. From the experiments it can be clearly seen that the NCA method significantly outperforms previously suggested linear dimensionality-reduction methods. In the experiments with dataset 1 (i.e., experiments 1 and 2), the NCA exhibits better results than the SVM algorithm. In experiment 3 the SVM results are slightly better than the NCA results.

3.2 *Performance of the weighted NCA*

The performance of the weighted-NCA algorithm was also tested in our experiments. The matrix $\mathbf{M}$, which was the identity matrix in the Section 3.1, was selected to be a general weight matrix in order to examine the weighted-NCA algorithm. In the first dataset we chose to provide the NCA algorithm more freedom in misclassification between different types of corn, different types of grass and different types of soybean.

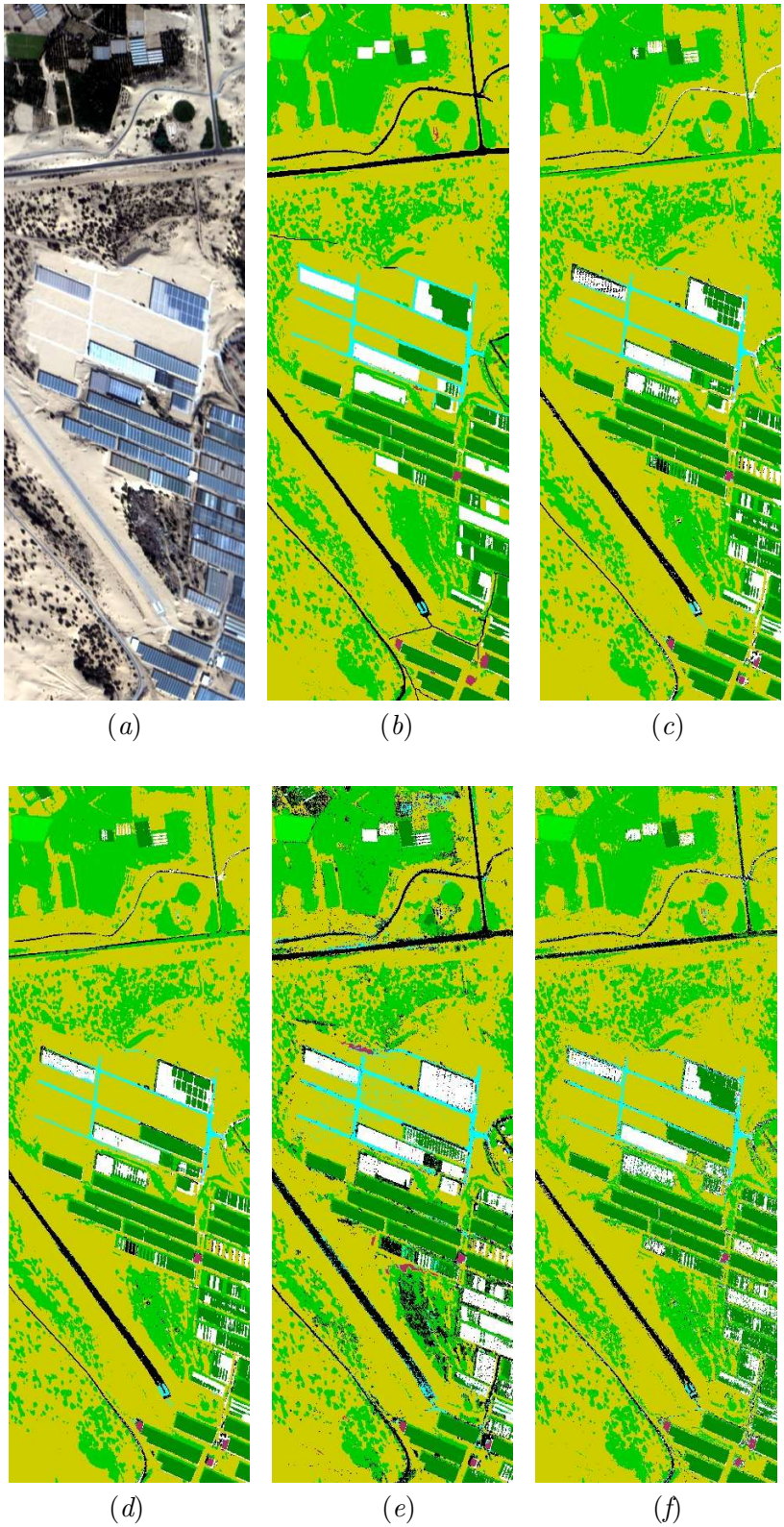

Figure 3. Classification results of experiment 3. (*a*) The hyperspectral image in true color, (*b*) The manually obtained, ground truth, (*c*) The SVM-RBF classification results, (*d*) The Linear-SVM results, (*e*) The DBFE classification results (for 8 extracted features) and (*f*) The NCA classification results (for 8 extracted features).

Table 5. The weight matrix used in experiments 1 and 2 ($\mathbf{M}_1$) and the weight matrix used in experiment 3 ($\mathbf{M}_2$).

$$\mathbf{M}_1 = \begin{pmatrix} 1 & .5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ .5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & .5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .5 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & .5 & .5 & 0 \\ 0 & 0 & 0 & 0 & 0 & .5 & 1 & .5 & 0 \\ 0 & 0 & 0 & 0 & 0 & .5 & .5 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & .5 & .5 & 0 & 0 & 0 & 0 \\ 0 & .5 & 1 & .5 & 0 & 0 & 0 & 0 \\ 0 & .5 & .5 & 1 & .5 & 0 & 0 & 0 \\ 0 & 0 & 0 & .5 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & .5 & 0 \\ 0 & 0 & 0 & 0 & 0 & .5 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Table 6. Class-by-class accuracies, achieved on the test set by NCA and weighted-NCA for 14 features, (experiment 1).

Method	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
NCA	90.32	87.21	99.16	98.88	100	90.02	92.56	89.32	99.53
Weighted-NCA	91.04	87.72	99.16	98.88	100	90.44	93.05	88.67	99.69

Table 7. Class-by-class accuracies, achieved on the test set by NCA and weighted-NCA for 14 features, (experiment 2).

Method	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
NCA	82.41	76.67	94.20	99.55	99.32	81.86	88.16	82.10	97.60
Weighted-NCA	82.88	76.80	93.97	99.55	99.32	82.20	88.34	81.37	97.60

Table 8. Class-by-class accuracies, achieved on the test set by NCA and weighted-NCA for 10 features, (experiment 3).

Method	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
NCA	75.85	87.44	97.71	91.76	83.92	94.87	65.48	57.79
Weighted-NCA	76.03	87.47	97.62	91.73	83.87	94.88	65.32	57.79

In the second dataset we chose to provide the NCA algorithm more freedom in misclassification between different vegetation types, different soil types and different greenhouses types, in order to achieve better classification results. To do so, we utilized the matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ defined in Table 5 for the first and second dataset, respectively. As explained in Section 2, the class-confusion matrix elements, $\mathbf{M}(c_1, c_2)$, represent the weight given for classification of a pixel from class c_1 to class c_2. For example, in the above defined $\mathbf{M}_2(2,3) = 0.5$, meaning: a accuracy score of 0.5 is given for misclassification of w_2 (pure vegetation) as w_3 (mixed vegetation). In this example we define that correct classification of w_1 and w_8 is vital and that they should be given priority over misclassification that is defined in $\mathbf{M}_2$. Tables 6-8 show the results of the NCA algorithm and the weighted-NCA for experiments 1-3, respectively.

From Tables 6-8 it can be seen that applying the weighted-NCA algorithm to the datasets leads to the expected results of improving the requested class' accuracy. For example, as can be seen from the matrix $\mathbf{M}_2$ defined in Table 5, class w_1 in the second dataset was given priority in order to improve its classification accuracy. As can be seen from Table 8, the classification accuracy of this class is improved with comparison to the NCA. In addition, it can be seen from the matrix $\mathbf{M}_1$ defined in Table 5 that classes w_1 and w_2 were given associated priority in the first database. Applying the weighted-NCA led to better classification accuracies of these classes with comparison to the NCA, on the account of other classes (as can be clearly seen from Tables 6 and 7).

4 Discussion and Conclusion

In this paper, we addressed the problem of classifying hyperspectral remote sensing data using the Neighbourhood Components Analysis approach. This method is now in wide use, particularly in areas of image classification. The method proposed in this study is actually an alternative to the common approach of

hyperspectral classification, which is based on two separated steps of feature selection or extraction procedure followed by a classifier. In our approach these two steps are combined into a singe process. The main contribution of this paper is presenting this alternative and implementing it. The experimental results indicate that NCA can achieve dimensionality reduction and classification concurrently. Furthermore, from our experiments we conclude that, this approach mostly exhibits results that are better than the state-of-the-art classifiers currently being used. This paper also contains a modification of the method that can be used to take into account more important and less important mis-classifications during the classifier learning process. The modified method is useful in cases where a high accuracy is required only for some of the classes in the scene. An important point that has to be addressed is the effect of the number of features on the classification results. Figures 4-6 present the classification accuracies versus the number of features for experiments 1-3, respectively. From these experimental results we see that increasing the number of features above a certain level (8 features in dataset 1, and 7 features in dataset 2) does not lead to a major improvement in the performance of the feature extraction methods. This ability of the NCA to obtain high accuracy classification in a relatively low-dimension space, can lead to a specifically designed multi-spectral sensor for a specific scene or a task. These sensors are much more financially attractive and their acquired data is much less time consuming than the usage of hyperspectral sensors. This way, much money and time can be saved if the type of scene or mission is well specified before acquiring the image.

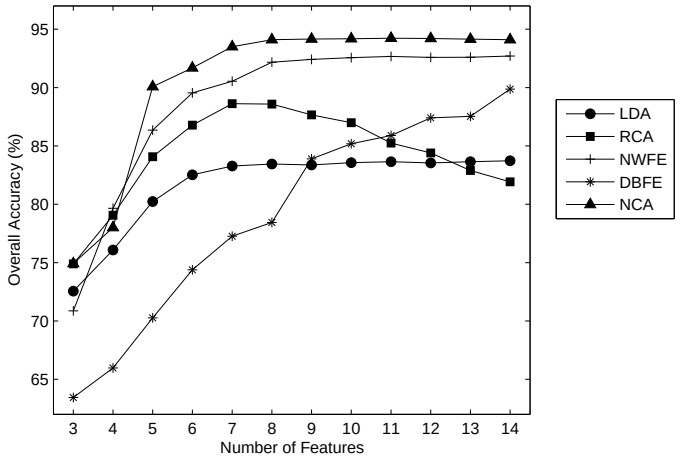

Figure 4. Experiment 1, Nearest-neighbour classification accuracy as a function the extracted features.

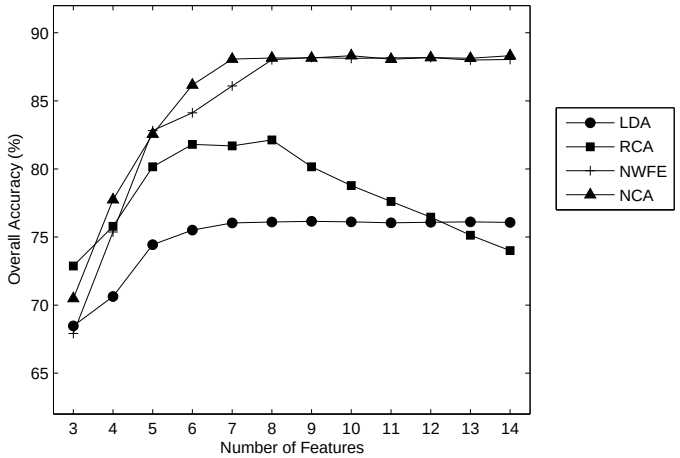

Figure 5. Experiment 2, Nearest-neighbour classification accuracy as a function the extracted features.

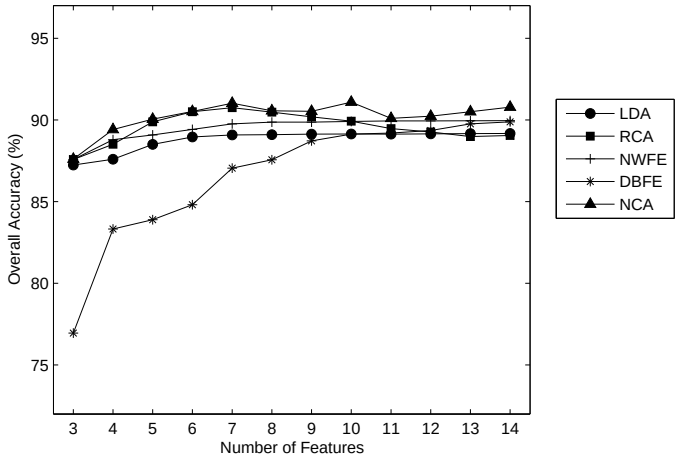

Figure 6. Experiment 3, Nearest-neighbour classification accuracy as a function the extracted features.

As a future work we will treat some additional issues. First, we did not use spatial or morphological information in our experiments. It is obvious, however, that using such information will improve the results and one should find the way to embed this information efficiently according to the specific scene or application. Second, since the NCA is an iterative method its complexity is higher than the variants of the LDA algorithm. The running time is comparable with that of the SVM algorithm and does not meet the requirements of real-time applications. Therefore, our future work will focus on improving the results while embedding spatial and morphological information. Another direction for future research is to reduce the computational complexity of the NCA algorithm.

References

[1] AVIRIS NW Indianas Indian Pines 1992 data set [Online]. Available: ftp://ftp.ecn.purdue.edu/biehl/MultiSpec/92AV3C.lan (original files) and ftp://ftp.ecn.purdue.edu/biehl/PC_MultiSpec/ThyFiles.zip (ground truth).

[2] Bar-Hillel A., Hertz T., Shental N., and Weinshall D., 2005, Learning a mahalanobis distance from equivalence constraints. *Journal of Machine Learning Research*, pages 937–965.

[3] Camps-Valls G., Gomez-Chova L., Munoz-Mari J., Vila-Frances J., and Calpe-Maravilla J., 2006, Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 3:93–97.

[4] Goldberger J., Roweis S., Hinton G., and Salkhutdinov R., 2004, Neighbourhood components analysis. *Advances in Neural Information Processing Systems (NIPS)*.

[5] Kou B. C. and Landgrebe D. A., 2004, Nonparametric weighted feature extraction for classification. *IEEE Transactions on Geoscience and Remote Sensing*, 42(5):1096–1105.

[6] Kou B. C. and Chang K. Y., 2007, Feature extraction for sample size classification problem. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3): 756–764.

[7] Lee C., and Landgrebe D. A., 1993, Feature extraction based on decision boundaries. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 15:388–400.

[8] Melgani F. and Bruzzone L., 2004, Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1796.

[9] Schölkopf B. and Smola. A., 2001, Learning with kernels support vector machines, regularization, optimization and beyond. MIT Press.