CREATING MULTIPLE CLASSIFIERS FOR THE CLASSIFICATION OF HYPERSPECTRAL DATA: FEATURE SELECTION OR FEATURE EXTRACTION

Yasser Maghsoudi[a], Majid Rahimzadegan[b], M. J. Valadan zoej[b]

[a]Inha University, Department of Geoinformatic Engineering, Yonghyun-dong, Nam-ku, Incheon 402-751, KOREA
[b]Faculty of Geodesy and Geomatics Eng., KN Toosi University of Technology, Mirdamad Cross, Tehran, Iran
ymaghsoudi@yahoo.com, maj_r2002@yahoo.com, valadanzouj@kntu.ac.ir

ABSTRACT

Classification of hyperspectral images is challenging. A very high dimensional input space requires an exponentially large amount of data to adequately and reliably represent the classes in that space. In other words in order to obtain statistically reliable classification results, the number of necessary training samples increases exponentially as the number of spectral bands increases. However, in many situations, acquisition of the large number of training samples for these high-dimensional datasets may not be so easy. This problem can be overcome by using multiple classifiers. In this paper we compared the effectiveness of two approaches for creating multiple classifiers, feature selection and feature extraction. The methods are based on generating multiple feature subsets by running feature selection or feature extraction algorithm several times, each time for discrimination of one of the classes from the rest. A maximum likelihood classifier is applied on each of the obtained feature subsets and finally a combination scheme was used to combine the outputs of individual classifiers. Experimental results show the effectiveness of feature extraction algorithm for generating multiple classifiers.

Keywords: hyperspectral data, classification, feature selection, feature extraction, multiple classifiers

1. INTRODUCTION

Recent developments in sensor technology have made it possible to collect hyperspectral data from 200 to 400 spectral bands. These data can provide more effective information for monitoring of the earth surface and a better discrimination among ground cover classes than the traditional multispectral scanners [1].

Although the availability of hyperspectral images is widespread but the data analysis approaches that have been successfully applied to multispectral data in the past are not so effective for hyperspectral data. The major problem is high dimensionality which can impair classification due to the curse of dimensionality. In other words, as the dimensionality increases, the number of training samples as needed for the characterization of classes increases considerably. If the number of training samples fails to satisfy the requirements, which is the case for hyperspectral images, the estimated statistics becomes very unreliable. Although increasing the number of spectral bands potentially provides more capabilities for discrimination of classes, this positive effect can be diluted by poor statistics estimation. As a result, the classification accuracy first grows and then declines with the number of spectral bands when the number of the training samples is low, finite and remains constant. This is often referred to as the Hughes Phenomenon [2]. Studies aiming at reducing the data dimensionality while keeping most of the relevant information have been reported by many authors [3]-[9]. Feature selection and feature extraction are two frequently employed approaches. The main idea in feature selection algorithms is to find an optimal or suboptimal subset of features according to some given criterion. Feature extraction approaches are often used to transform data from the original higher dimensional space into a lower dimensional feature space.

In the present study in order to improve the classification accuracy, instead of using one classifier we exploit the theory of multiple classifiers. To have a good performance, two conditions should be met for an ensemble of classifiers [10].Firstly the classifiers must be diverse. Obviously ensembling identical classifiers will not lead to any improvements. Secondly the classifiers should be accurate. An accurate classifier is one that has an error rate of better than random guessing on a new data point. Therefore design of classifier ensembles consists of two parts. The first part is constructing multiple classifiers for creation of a set of diverse and accurate classifiers and the second part is the design of a combination to combine the outputs of the individual classifiers. In this study we concentrate on the former part i.e. creating multiple classifiers. We compared the effectiveness of two approaches for creating multiple classifiers, feature selection and feature extraction.

2. METHODS FOR CREATING MULTIPLE CLASSIFIERS

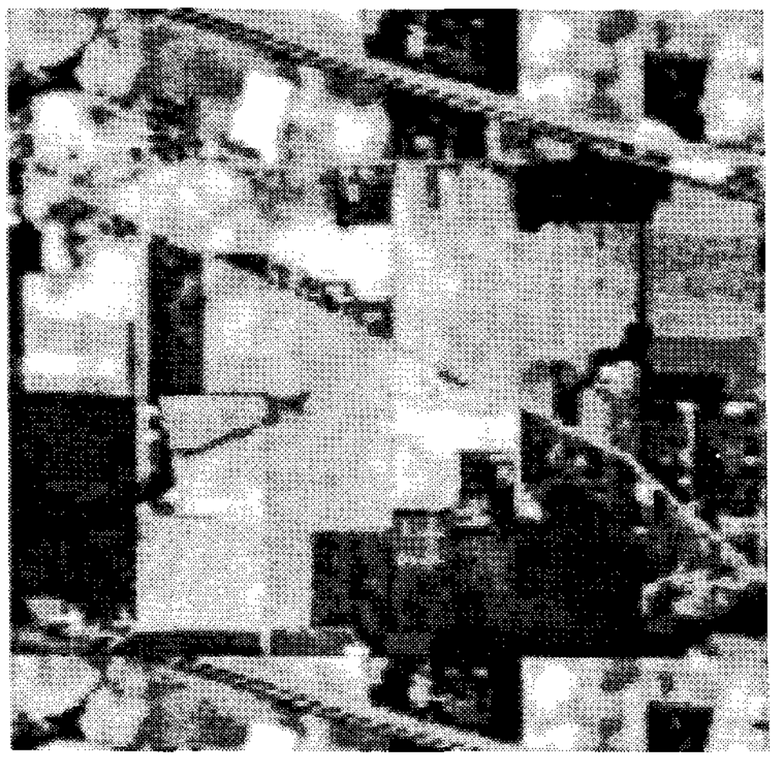

Figure 1. Band 12 of the hyperspectral image utilized in the experiments.

In general there are three categories of approaches for creating multiple classifiers with the above mentioned properties: manipulating the training data, manipulating the output classes and manipulating the input features. In the first category of methods an ensemble of classifiers is generated by training classifiers on different sets of training data. Bagging [11] which uses sampling with replacement is one of the best known methods for generating a set of classifiers. In bagging n different training sets are created by sampling with replacement from the original training set. Finally a classifier is trained on each set and the outputs of classifiers are combined using a simple voting. A popular alternative to Bagging is Boosting [12]. The second group of methods for generating an ensemble of classifiers is through manipulating the output classes. In these methods a multi-class problem is decomposed into multiple two class classifiers. Error Correcting Output Coding (ECOC) proposed by Dietterich and Bakiri [13] is one of these methods.

But the third and the most important category of methods-from the standpoint of hyperspectral data classification- is via manipulating input features. In these methods the input feature space is divided into multiple feature subsets. Then a classifier is trained on each of these newly-generated feature sets. Finally the outputs of these classifiers are combined via a combination schema. Obviously, this method works well when the input features are highly redundant [10]. In this process employing the most effective way for sampling of the features from the input feature space, that can provide us both diverse and accurate classifies, is the most challenging point. Random selection is one of these sampling strategies [14]. The feature subsets generated by this strategy are diverse but they are not accurate enough for having an effective ensemble of classifiers. In order to overcome this problem we exploit feature selection and also feature extraction techniques for the sampling strategy of features from the original feature space.

Table 1. List of classes, training and testing sample sizes used in the experiments.

Land cover classes	Number of Training	Number of Testing
1-Corn-notill	519	749
2-Corn-min	275	503
3-Grass/pasture	160	260
4-Grass/trees	219	504
5-Hay-windrowed	135	267
6-Soy-notill	231	454
7-Soy-mintill	623	1069
8-Soy-clean	168	212
9-Woods	310	424
Total	2640	4442

3. METHODOLOGIES AND EXPERIMENTAL RESULTS

3.1 Dataset Description

The dataset used in this study is an AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) dataset downloaded from [16].

The considered dataset referred to the agricultural area of Indian pie in the Northern part of Indiana. Images have been acquired by an AVIRIS in June 1992. The dataset was composed of 220 spectral channels (spaced at about 10 nm) acquired in the 0.4-2.5 um region. Figure 1 shows channel 12 of the sensor. The nine landcover classes used in our study are also shown in Table 1.

3.2 Class-based Feature Selection

The main idea of the method is that from the huge number of spectral bands in hyperspectral data there are some bands which can discriminate each class better than the others. Assume that there are k classes in the classification problem.

$$C = \{C(1), C(2),..., C(k), \quad k = 1,2,..., 9\} \qquad (1)$$

In order to find the best features for each of the classes we applied a feature selection process. In general the feature selection problem can be stated as follows: Given a set of N features find the best subset of m features to be used for classification. Feature selection algorithms generally involve both a search strategy and an evaluation function [3] [15]. The aim of the search algorithm is to generate subsets of features from the original feature space and the evaluation function compares these feature subsets in terms of discrimination.

In this paper the Fast Constrained (FC) search algorithm is used as the search strategy. It is the computationally reduced version of Steepest Ascent (SA) algorithm which is proposed by Serpico et al. [9]. SA is based on the representation of the problem solution by a discrete binary space and on the search for constrained local maximums of a criterion

function in such space. A feature subset is a local maximum of the criterion function if the value of that feature subset criterion function is greater than or equal to the value the criterion function takes on any other point of the neighborhood of that subspace. Unlike SA the number of iterations in FC algorithm is deterministic so it is faster and the quality of the selected features is comparable to many other search algorithms.

The Jeffries-Matusita distance, which is an inter-class measure, is used as a criterion for the evaluation of feature subsets. The Jeffries-Matusita distance is as follows:

$$JM = 2\sum_{n=1}^{k}\sum_{m>n}^{k} JM_{mn} \qquad (2)$$

$$JM_{mn} = \sqrt{2(1-e^{-bmn})} \qquad (3)$$

$$b_{mn} = \frac{1}{8}(M_m - M_n)^T \left(\frac{C_m + C_n}{2}\right)^{-1}(M_m - M_n) + \frac{1}{2}\ln\left(\frac{\left|\frac{C_m + C_n}{2}\right|}{\sqrt{|C_m||C_n|}}\right) \qquad (4)$$

where k is the number of classes, b_{ij} is the Bhattacharyya distance between class i and j and M_i and C_i are the mean vector and covariance matrix of the class i respectively.

Each time for the evaluation of each feature subset for each class the Jeffries-Matusita distance from that class to the rest of classes is computed and the sum of them is set as the evaluation for that feature subset for that specified class.

Before running the class-based feature selection procedure we must know the best number of feature to be selected. In order to find the best number of features for each subset we run the feature selection algorithm with different number of features (from 5 to 50 features). In this study we used Bayesian classifier as the classification algorithm. The highest classification accuracy is provided using 24 features. Although the overall accuracy is used as measure for finding the best number of features for each subset, it can provide us a rough estimate of the appropriate number of features.

After finding the best number for the features we run the feature selection algorithm for the first class. In this manner the feature subset with the highest evaluation function for that class is selected. This process is repeated for all other classes. Subsequently the Bayesian classifier is trained on each of those selected feature subsets. Finally a combination schema is used to combine the outputs of the individual classifiers. Due to the nature of the individual classifiers for emphasizing one class more than the others so a simple voting schema can't work in this case.

For this reason we propose the following combination rule. Since the first classifier's emphasis is on the first class so it shares its first class values in the final classified image, the second classifier plays this role by sharing its second class values and so on. Three different cases might happen in the final classified image:

1) Those pixels in the final classified image for which there is only one decision; these pixels get their final label from the only decision made by the related classifier.

2) Those pixels in the final classified image for which there are more than one decisions; in this case there is a conflict between two or more classifiers for deciding the final label of the pixel. In order to settle down this problem we exploit the output probabilities of the conflicting classifiers. In other words we hold a competition among those conflicting classifiers and the class with the highest value of probability is selected as the final decision.

3) Those pixels in the final classified image for which there isn't any decision. In this time we hold a competition among the output probabilities of all the classifiers.

Table 2. Classification accuracy for each class in single feature selection and class-based feature selection

Classes	Classification Accuracy (Percent)	
	Feature selection	Class-based Feature selection
Corn-notill	90.254	93.858
Corn-min	82.505	82.903
Grass/pasture	98.077	98.462
Grass/trees	96.627	99.206
Hay-windrowed	99.625	99.625
Soy-notill	81.278	81.718
Soy-mintill	72.872	72.311
Soy-clean	83.962	85.849
Woods	98.349	98.821
Overall Accuracy	86.493	87.506

Table 2 shows the classification accuracies for the test samples in different classes. The first column of the results represents the classification accuracy using a general feature selection algorithm and the second column of the results is associated with the class-based feature selection.

3.3 Class-based Feature Extraction

In class-based feature extraction methodology we repeated the above-mentioned procedure, with this difference that in the latter the feature selection process is replaced by feature extraction. Feature extraction methods transform a large amount of information on classes' separability into a small number of extracted features and consequently minimizing the dimensions of the feature space while enhancing the overall accuracy of classification. The feature extraction process is usually based on finding features that optimize a particular criterion. For example, in discriminant analysis within class and between class scatter matrices are estimated by using the training samples, and then features that optimize a function of these matrices are obtained[21].

Current feature extraction algorithms while effective in some circumstances have significant limitations [17]. A new Nonparametric Weighted Feature Extraction method (NWFE) is developed to solve limitations of the current methods [20].

NWFE takes advantages of desirable characteristics of pervious methods, while avoiding their shortcomings. The main idea of NWFE is putting different weights on each sample to compute the "weighted means" and defining new nonparametric between-class and within-class scatter matrices to obtain more than N–1 features. In NWFE, the nonparametric between-class scatter matrix for L classes is defined as:

$$S_b = \sum_{i=1}^{N} \frac{p_i}{N-1} \sum_{\substack{j=1 \\ j \neq i}}^{N} \sum_{k=1}^{N_j} \lambda_k^{(i,j)} (x_k^{(i)} - M_j(x_k^{(i)}))(x_k^{(i)} - M_j(x_k^{(i)}))^T \qquad (5)$$

where $x_k^{(i)}$ refers to the k[th] sample from class i, N_i is training sample size of class i, p_i denotes the prior probability of class i. The scatter matrix weight $\lambda_k^{(i,j)}$ is a function of $x_k^{(i)}$ and $M_j(x_k^{(i)})$, and defined as:

$$\lambda_k^{(i,j)} = \frac{dist\ (x_k^{(i)}, M_j(x_k^{(i)}))^{-1}}{\sum_{l=1}^{N_j} dist\ (x_l^{(i)}, M_j(x_l^{(i)}))^{-1}} \qquad (6)$$

where $dist(a,b)$ denotes the Euclidean distance from a to b.

$M_j(x_k^{(i)})$ denotes the weighted mean $x_k^{(i)}$ in class j and defined as:

$$M_j(x_k^{(i)}) = \sum_{l=1}^{N_j} W_l^{(i,j)} x_l^{(i)} x_l^{(j)} \qquad (7)$$

where

$$W_l^{(i,j)} = \frac{dist\ (x_k^{(i)}, x_l^{(j)})^{-1}}{\sum_{l=1}^{N_j} dist\ (x_k^{(i)}, x_l^{(j)})^{-1}} \qquad (8)$$

The nonparametric within-class scatter matrix is defined as:

$$S_w = \sum_{i=1}^{N} p_i \sum_{k=1}^{N_i} \lambda_k^{(i,j)} (x_k^{(i)} - M_i(x_k^{(i)}))(x_k^{(i)} - M_i(x_k^{(i)}))^T \qquad (9)$$

In this paper NWFE algorithm is used for constructing individual classifiers. In order to inject the class-based nature to NWFE algorithm we modified the algorithm in the following way:

For each class m:

$$S_b = \frac{p_m}{N-1} \sum_{\substack{j=1 \\ j \neq m}}^{N} \sum_{k=1}^{N_m} \lambda_k^{(m,j)} (x_k^{(m)} - M_j(x_k^{(m)}))(x_k^{(m)} - M_j(x_k^{(m)}))^T$$

$$+ \sum_{\substack{i=1 \\ i \neq m}}^{N} \frac{p_i}{N-1} \sum_{k=1}^{N_i} \lambda_k^{(i,m)} (x_k^{(i)} - M_m(x_k^{(i)}))(x_k^{(i)} - M_m(x_k^{(i)}))^T \qquad (10)$$

$$S_w = p_m \sum_{k=1}^{N_m} \lambda_k^{(m,j)} (x_k^{(m)} - M_m(x_k^{(m)}))(x_k^{(m)} - M_m(x_k^{(m)}))^T \qquad (11)$$

in which m is the number of class. As can be seen the between-class scatter matrix here is defined as the distance between one of the classes and the rest of classes and for the within-class scatter matrix only one class is considered. We applied this modified NWFE for N times each time for extracting the features associated to one of the classes.

Table 3. Best number of extracted features for class-based feature extraction

Class number	Number of extracted features
1	14
2	1
3	13
4	36
5	5
6	5
7	132
8	7
9	86

The same as class-based feature selection, in class-based feature extraction, we must find the best number of features that can be used for the classification of each class which can provide maximum producer accuracy for that class. In this respect, we used the class accuracy as a criterion for finding the optimum number of features for that class and the NWFE algorithm was run several times (from 1 to 133). The best number of extracted features for each class is represented in table 3.

Table 4. Classification accuracy for each class in single feature extraction and class-based feature extraction

Classes	Classification Accuracy (Percent)	
	Feature extraction	Class-based Feature extraction
Corn-notill	90.5207	90.6542
Corn-min	71.7694	85.2883
Grass/pasture	98.0769	98.0769
Grass/trees	96.2302	97.2222
Hay-windrowed	99.6255	99.6255
Soy-notill	79.0749	84.8018
Soy-mintill	72.4977	73.1525
Soy-clean	91.0377	93.3962
Woods	97.8774	98.5849
Overall Accuracy	85.6067	88.0122

Finding the best number of features for each class, a Bayesian classifier is trained on each of those subsets. Owing the identical nature of output in both class-based feature selection and extraction, the same combination schema was performed here as well.

As can be inferred from Table 2, the class-based method has superior performance comparing to a general feature selection methods. In a general feature selection algorithm the main goal is to find the best features to be used for the classification of the whole image whereas the class-based methodology tries to find the features locally. Different regions in the image are classified with different sets of features. That's also the case about class-based feature extraction. As can be seen in table 4 the classification accuracy is improved in almost all the classes.

The comparison between table 2 and table 4 also demonstrates that class-based feature extraction

methodology outperforms the class-based feature selection. A possible explanation is that the feature extraction reduces the dimensionality without sacrificing significant information whereas in feature selection the dimensionality reduction is via picking up a subset of bands and ignoring the rest so the feature extraction can be more useful than feature selection.

4. CONCLUSIONS

The class-based methods can be very effective for creating an ensemble of classifiers. In this regard we applied two methods of class-based feature selection and class-based feature extraction for creating multiple classifiers. Employing these approaches not only solved the small training sample size problem but they also improved the classification accuracy. In comparison with a single feature selection or feature extraction applying a class-based methodology in feature selection or feature extraction can lead to an increase in classification accuracy. Experimental results also confirmed the suitability of class-based feature extraction in comparison with class-based feature selection for creating multiple classifiers.

REFRENCES

[1] C. Lee and D. A. Landgrebe, Analyzing high-dimensional multispectral data, IEEE Trans. On Geoscience and Remote Sensing, vol. 31, pp. 792–800, July 1993.

[2] G.F. Hughes. On the SUM accuracy of statistical pattern recognizers, IEEE Trans. On Information Theory, pp.55-63, 1968.

[3] K. Fukunaga. Introduction to Statistical Pattern Recognition, 2nd edition, Academic Press, New York, 1990.

[4] X. Jia, J. A. Richards, Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification, IEEE Trans. on Geoscience and Remote Sensing, vol. 37, no. 1 pp.538-542, 1999.

[5] C. Lee and D. A. Landgrebe, Decision boundary feature extraction for neural networks, IEEE Trans. Neural Networks, vol. 8, pp. 75–83, Jan. 1997.

[6] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp 388-400, 1993.

[7] S. Kumar, J. Ghosh, and M. M. Crawford, Best-bases feature extraction algorithms for classification of hyperspectral data, IEEE Trans. on Geoscience and Remote Sensing, vol. 39, pp. 1368-1379, 2001.

[8] A. Jain, D. Zongker, Feature selection: evaluation, application and small sample performance, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, No. 2, pp. 153-158, Feb. 1997.

[9] S. B. Serpico and L. Bruzzone, A new search algorithm for feature selection in hyperspectral remote sensing images, IEEE. Trans. on Geoscience and Remote Sensing, Special Issue on Analysis of Hyperspectral Image Data, vol. 39, No. 7, pp. 1360-1367, 2000.

[10] T.G. Dietterich, Ensemble methods in machine learning, In Proc. of MCS 2000, Lecture Notes in Computer Science, 2000.

[11] L. Breiman, Bagging predictors, Machine Learning, pp.123-140, 1996.

[12] R. E. Schapire, The strength of weak learnability, Machine Learning, pp.197-227, 1990.

[13] T.G. Dietterich and G. Bakiri, Solving multiclass learning problems via error correcting output codes, Journal of Artificial Intelligence Research, pp. 263-286, 1995.

[14] Ho, T.K. The random subspace method for constructing decision forests, IEEE Transactions, Pattern Analysis and Machine Intelligence, pp. 832-844, 1998.

[15] P. H. Swain and S. M. Davis, Remote sensing: the quantitative approach. New York: McGraw-Hill, 1978.

[16] http://dynamo.ecn.purdue.edu/~biehl/multispec/documentation.html.

[17] B-C. Kuo and D. A. Landgrebe, "Improved Statistics Estimation and Feature Extraction For Hyperspectral Data Classification", PhD Thesis, School of Electrical & Computer Engineering Technical Report. TR-ECE 01-6, December 2001.

[18] R, P. W. Duin and R. Haeb-Umbach, "Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, 2001, pp. 762-766.

[19] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 388–400, Apr . 1993.

[20] B-C. Kuo and D. A. Landgrebe, "Nonparametric Weighted Feature Extraction for Classification", IEEE Transactions on Geoscience and Remote Sensing, Volume 42 No. 5, pp 1096-1105, May, 2004.

[21] B. M. Shahshahani and D. A. Landgrebe, "The Effect Of Unlabeled Samples In Reducing The Small Sample Size Problem And Mitigating The Hughes Phenomenon", IEEE Transactions on Geoscience and Remote Sensing, Vol. 32, No. 5, pp 1087-1095, September 1994.