WEIGHTED COMBINATION OF MULTIPLE CLASSIFIERS FOR THE CLASSIFICATION OF HYPERSPECTRAL IMAGES USING A GENETIC ALGORITHM

Y.Maghsoudi[a], A.Alimohammadi[b], M.J.Valadan Zoej[b] and B. Mojaradi[b]

[a]Islamic Azad University, Maybod branch, Maybod, Yazd, Iran- ymaghsoudi@yahoo.com

[b]Faculty of Geodesy and Geomatics Eng., KN Toosi University of Technology, Mirdamad Cross, Tehran, Iran

alimoh_abb@yahoo.com, valadanzouj@kntu.ac.ir, Mojaradi@alborz.kntu.ac.ir

KEYWORDS: hyperspectral, classification, multiple classifiers, weighted, genetic

ABSTRACT:

The improved spectral resolution of modern hyperspectral sensors provides effective means for discrimination of subtly different classes and objects. However, in order to obtain statistically reliable classification results, the number of required 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 possible or so easy. This problem may be overcome by using multiple classifiers. In this paper, we describe a weighted combination of multiple classifiers based on the genetic algorithm. Practical examinations on the AVIRIS data for discrimination of different land use/cover classes demonstrate the effectiveness of the proposed approach.

1. INTRODUCTION

Hyperspectral imaging is a fast growing area in remote sensing. It leads to expansion and improvement of the capabilities of multispectral image analysis. Hyperspectral images take advantage of hundreds of contiguous spectral channels to uncover materials that usually cannot be resolved by multispectral sensors. However, the data analysis approach that has been successfully applied to multispectral data in the past is not so effective for hyperspectral data. The classification performance in these images suffers from two important problems:

1. Curse of dimensionality; the accuracy of parameter estimation depends substantially on the ratio of the number of training samples to the dimensionality of the feature space. As the dimensionality increases, the number of training samples as needed for characterization of classes increases considerably. If the size of the training samples fails to satisfy the requirements, which is the case for the hyperspectral images, the estimated statistics, becomes very unreliable. As a result, for a given number of training samples, the classification accuracy first grows and then declines as a function of the increase in the number of spectral bands. This is often referred to as the Hughes Phenomenon or the curse of dimensionality [1]. As it is often difficult to provide adequate training samples for supervised classification, an ensemble of classifiers can be used to solve this problem.

2. Large hypothesis space; In general there are three spaces associated with any classification problem: (i) Input space , which is the space of all the features that are used in the classification process ,(ii) Output space which is the set of all observed classes. This space is the most powerful one from the standpoint of information extraction [2] and (iii) Hypothesis space which is the space of the models in which the desired classifier is sought. With increase in the input dimensionality, for a fixed number of classes and choice of a classifier family, the hypothesis space also grows exponentially. This problem makes the classification performance very unreliable. By using an ensemble of classifiers this problem can also be avoided.

An ensemble of classifiers requires two conditions to be met in order to reduce the generalization error of its constituent members [11]. Firstly the classifiers must be diverse. To be precise about what diversity means, classifiers should be independent i.e. containing uncorrelated errors. 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. If the classifiers are an average accurate and diverse then we would expect that most of the classifiers will not make the same mistake on the same example. A simple majority voting schema would ensure that the correct classification is made.

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 scheme for implementation of fusion mechanism that can optimally combine the classifications.

In this paper a weighted combination of multiple classifiers has been described. In practical situations, each classifier is not fully certain and complete. Therefore it is necessary to weight each of the classifiers so that the final ensemble reflects our knowledge of the reliability of each of the classifiers. These weighting factors can be obtained by different methods. In this study a genetic algorithm has been employed to define these weights.

The paper is organized into five sections. Section two presents a literature survey on different methods for creating and then fusing multiple classifiers. In section three different methods for fusing multiple classifiers have been reviewed. Section four includes the framework for weighted combination of classifiers and the ways that the weighting parameters can be obtained. The experimental results on AVIRIS data and the conclusions have been presented in sections five and six.

2. METHODS FOR CREATING AND FUSING MULTIPLE CLASSIFIERS

There are three methods for creating an ensemble with the above mentioned properties The first method of generating an ensemble of classifiers is to train classifiers on different sets of training data. Bagging [4] which uses sampling with replacement is one of the best known methods for generating a set of classifiers. In bagging we create n different training sets by sampling with replacement from the original training set. We then train a classifier on each set and combine their outputs using a simple voting. A popular alternative to Bagging is Boosting [5]. In Boosting the classifiers in the ensemble are trained serially, with the weights on the training instances set according to the performance of the previous classifiers.

Another method for generating multiple classifiers is to manipulate the set of input features. In this method different feature subspaces are passed to different classifiers. Obviously, this method works well when the input features are highly redundant. On the other hand this approach does not suffer from curse of dimensionality.

The third method for generating a good ensemble of classifiers is through manipulating the output classes. Error Correcting Output Coding (ECOC) proposed by Dietterich and Bakiri [6] is one of these methods.

Methods for fusing multiple classifiers can be classified according to the type of information produced by the individual classifiers. Three levels can be defined [7]:

1) Abstract-level: The output of each classifier is a unique class label for each input pattern.

2) Rank-level: The output of each classifier is a list of possible classes with ranking for each input pattern.

3) Measurement-level: The outputs are confidence levels, for each class, for each input pattern.

In this study, the Bayesian classifier which can provide information at the measurement-level has been used.

Let all individual classifiers follow the Bayesian rule. In this rule, the classification of an input pattern x is based on the calculation of posterior probabilities:

$$P(w_i \mid x) \qquad i = 1,2,3,\cdots,k \tag{2}$$

where $P(w_i \mid x)$ represents the probability that x comes from each of k classes under the condition x. If we take $P(w_i \mid x_j)$ as the postprobability of the j^{th} classifier to the class w_i then some of the important combination rules are as follow [8]:

1- Sum rule: It is based on computing the sum of the outputs of the individual classifiers:

$$P_i = \sum_{j=1}^{N} p(w_i \mid x_j) \qquad i = 1,2,3,\cdots,k \tag{3}$$

The winner class is the class with the largest P_i.

2- Product rule: The product of the posterior probabilities related to each class is calculated:

$$P_i = \prod_{j=1}^{N} P(w_i \mid x_j) \qquad i = 1,2,3,\cdots,k \tag{4}$$

The unknown sample x is assigned to the class with the largest P_i.

3- Max Rule: This rule is based on finding the maximum in the outputs of the individual classifiers:

$$P_i = \mathrm{Max}_{j=1}^{N} P(w_i \mid x_j) \quad i = 1,2,3,\cdots,k \tag{5}$$

The winner is the class with the largest P_i.

4- Min Rule: it is based on finding the Minimum in the outputs of the individual classifiers:

$$P_i = \mathrm{Min}_{j=1}^{N} P(w_i \mid x_j) \quad j = 1,2,3,\cdots,k \tag{6}$$

The winner is the class with the largest P_i.

3. WEIGHTED COMBINATION OF MULTIPLE CLASSIFIERS

In practical circumstances, fully certain and complete classifier does not exist. Therefore it is necessary to weight each of the classifiers so that the final ensemble reflects our knowledge of the reliability of each of the classifiers. In other words, one can assign different weights to different classifiers in order to achieve a more satisfactory ensemble of classifiers. One simple way to adjust the contribution of each classifier is to append exponents to the postprobabilities of the classifiers [10]. This modification to equations 3 to 6 gives:

$$P_i = \sum_{j=1}^{N} p(w_i \mid x_j)^{\alpha_j} \qquad i = 1,2,3,\ldots,k \qquad (7)$$

$$P_i = \prod_{j=1}^{N} P(w_i \mid x_j)^{\alpha_j} \qquad i = 1,2,3,\ldots,k \qquad (8)$$

$$P_i = Max_{j=1}^{N}\ P(w_i \mid x_j)^{\alpha_j} \qquad i = 1,2,3,\ldots,k \qquad (9)$$

$$P_i = Min_{j=1}^{N}\ P(w_i \mid x_j)^{\alpha_j} \qquad i = 1,2,3,\ldots,k \qquad (10)$$

where $0 \le \alpha_j \le 1$ is the classifier-specific weighting parameter which allows one to adjust the contribution for the j^{th} classifier.

Given the above definition of α_j it is clear that if α_j is equal to the value 1 then the classifier j is fully reliable. As α_j tends to 0 then the classifier j is considered to be less reliable. The choice of classifiers weighting factors will have a significant effect on the results of the classification because the contribution of each classifier will be reduced or enhanced in proportion to its weight. Several possible methods for measuring the classifiers weighting parameters may be employed as described below:

3.1 Using classification accuracy: in this method the overall accuracy of each classifier is used for generating the weighting parameters. A classifier is assigned a higher weight if the resulting classification accuracy is high. Therefore, if the overall accuracy of a classifier is low, the classifier is assigned a lower weight:

$$\alpha_i \propto \text{overall accuracy(i)} \qquad i = 1,2,\ldots,n \qquad (11)$$

where n is the number of classifiers.

3.2 Using Class Accuracy: The overall accuracy of a classifier can only provide a general overview of the performance of a classifier and it is only a rough guide. In order to overcome this problem the weighting process can be adapted locally. In other words the weighting parameters are obtained using the performance of a classifier in a small part of an image rather than the whole image. Therefore a class-based weighting can be applied. The main idea is that the classes which are poorly classified by a specified classifier receive a lower weight during the combination process:

$$\alpha_{ij} \propto \text{class accuracy(i, j)} \quad i = 1,2,\ldots,n \quad j = 1,2,\ldots,k \qquad (12)$$

where n and k are the number of classifiers and the number of classes respectively. The class accuracies can be derived by means of the confusion matrix.

3.3 Using Genetic Algorithm: Using the performance of a classifier as its weight is based on the intuitive assumption that classifiers with high classification accuracy are more trustworthy than the classifiers that perform poorly. However, there is no objective proof that this strategy is optimal. Under a more general approach, we may consider the set of weights as free parameters in a multiple classifier system, and try to find the combination of values that lead to the best performance of the whole system. Out of many possible optimization procedures it was decided to use a genetic algorithm [11] for weighted optimization. Among the reasons to favor a genetic approach over other methods was the simplicity and elegance of genetic algorithms as well as their demonstrated performance in many other complex optimization problems [12][13][14]. The genetic algorithm for obtaining the weight factors has been described in Section 5 in greater detail.

4. EXPERIMENTS

The dataset used in this study is an AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) dataset downloaded from [9]. 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.

In our previous study[15] the Random Subspace Method was employed for construction of the multiple classifiers in which different feature subsets were randomly selected and passed on to the Bayesian classifiers. The outputs of each Bayesian classifier are

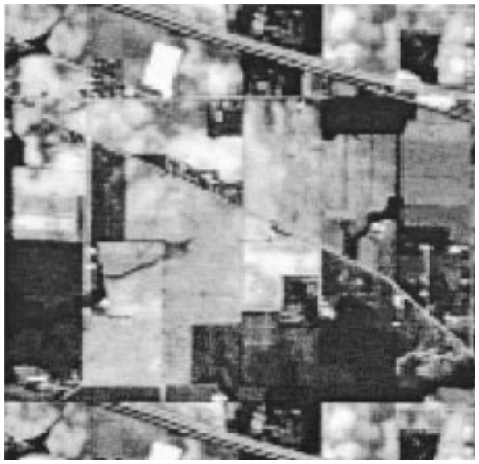

Figure 2. Band 12 of the hyperspectral image utilized in the experiments.

vectors of probabilities for different classes. Four operators including the Max, Min, Sum and Product were then used to combine the outputs of the individual classifiers. It was shown that the best classification accuracy was for the case that 10 features were used in each of the 22 subspaces (Table 2 shows the results of this experiment).

In this paper the same process has been examined but by introducing weights to the classifiers or classes. All the methods for obtaining the weighting parameters have been implemented and tested on the described dataset. Tables 3 and 4 show the results of the first and second experiment which have been based on the use of classification accuracy and class accuracy respectively for definition of the weight factors.

Using genetic algorithm for obtaining the weighting factors is straightforward. In a genetic algorithm a possible solution of the problem under consideration is represented by a chromosome. In the initialization step of the algorithm a set of chromosomes are created randomly. The actual set of chromosomes is called the population. A fitness function is defined to represent the quality of the solution given by a chromosome. Only the chromosomes with the highest values of this fitness function are allowed to reproduce. In the reproduction phase new chromosomes are created by fusing information of two existing chromosome (crossover) and by randomly changing them (mutation). Finally the chromosomes with the lowest values of the fitness function are removed. This reproduction and elimination step is repeated until a predefined termination condition becomes true. In the following the genetic algorithm that is used in our multiple classifier system has been described in more details.

4.1 Chromosomes and the fitness function

In this study instead of using one weight factor for each classifier a weight is used for each class in each classifier. So each gene in each chromosome corresponds to the weight of each class in each

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

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

classifier. Therefore a $n \times k$ chromosome will be formed in which n and k are the number of classifiers and the number of classes respectively. The chromosomes are represented by an array of real numbers between 0 and 1. The fitness of a chromosome is defined as the overall accuracy of the whole ensemble when using weighted voting with the weights represented by the chromosome.

4.2 Algorithm initialization

A population of size 400 was used in the algorithm. All positions of the chromosomes are set to random real values between 0 and 1 at the beginning of the algorithm. Then the fitness function is calculated for all of these chromosomes.

4.3 Selection

The tournament selection was adopted in this study. Pairs of individuals are picked at random from the population. Whichever has the higher fitness is copied into a mating pool (and then both are replaced in the original population). This is repeated until the mating pool is full. In the selection process the two chromosomes with the highest value of fitness are copied to the mating pool without any tournament.

4.4 Crossover

A uniform crossover operator is used. First a crossover mask with random values of 0 and 1 is generated. Two parents are selected randomly from the mating pool. Where there is a 1 in the crossover mask, the offspring gene is copied from the first parent, and where there is a 0 in the mask, the offspring gene is copied from the second parent. The probability of the crossover was set to 90 %.

5.5 Mutation

The mutation operator is applied to all new chromosomes produced by the crossover operator. It randomly alters each gene with a small probability (In

this study 0.002). Mutation provides a small amount of random search, and helps ensure that no point in the search space has a zero probability of being examined.

Classes	Classification Accuracy(Percent)			
	Max	Min	Sum	Product
Corn-notill	84.112	87.316	94.126	94.126
Corn-min	68.191	70.577	75.547	75.348
Grass/pasture	93.846	96.923	97.308	97.308
Grass/trees	92.659	96.825	97.421	97.421
Hay-windrowed	99.251	99.625	99.625	99.625
Soy-notill	80.837	68.062	85.242	85.903
Soy-mintill	51.356	70.533	76.146	76.239
Soy-clean	70.283	78.302	91.509	90.566
Woods	97.642	98.821	98.585	98.585
Overall Accuracy	77.172	82.463	87.978	88.001

Table 2. Random Subspace Method without weight

Classes	Classification Accuracy(Percent)			
	Max	Min	Sum	Product
Corn-notill	79.306	84.379	91.055	91.055
Corn-min	67.793	69.781	74.155	74.155
Grass/pasture	93.846	96.923	97.308	97.308
Grass/trees	92.262	96.429	97.222	97.222
Hay-windrowed	99.251	99.625	99.625	99.625
Soy-notill	81.498	68.943	85.683	86.344
Soy-mintill	55.753	74.369	80.823	80.449
Soy-clean	70.283	77.83	91.509	91.509
Woods	97.642	98.821	98.585	98.585
Overall Accuracy	77.398	82.823	88.451	88.429

Table 4. Weighted combination based on class accuracy

The crossover and mutation operators are applied to the selected parents until there are enough offspring for the generation of a new population. At this time the old population is replaced by the new one. (Figure 3)

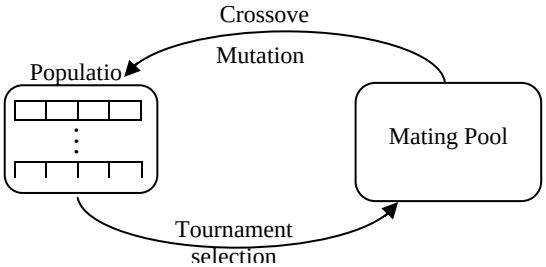

Figure 3. The scheme of the genetic algorithm used in the study

5.6 Convergence and Termination

If the value of the maximum fitness in the population is the same in 50 generations in a row the algorithm is terminated. This convergence observed in the 200th generation for all combination methods.

The weights of the chromosome with the highest fitness value are the final result and are used for the weighted voting combination. Table 5 shows the

Classes	Classification Accuracy(Percent)			
	Max	Min	Sum	Product
Corn-notill	83.044	87.45	94.126	94.126
Corn-min	68.588	71.372	75.547	75.348
Grass/pasture	93.462	96.923	97.308	97.308
Grass/trees	92.262	96.825	97.421	97.421
Hay-windrowed	99.251	99.625	99.625	99.625
Soy-notill	81.498	68.282	85.242	85.903
Soy-mintill	51.169	70.72	76.333	76.239
Soy-clean	70.755	78.774	91.509	90.566
Woods	97.642	98.821	98.585	98.585
Overall Accuracy	77.015	82.665	88.023	88.001

Table 3. Weighted combination: Using classification accuracy

Classes	Classification Accuracy(Percent)			
	Max	Min	Sum	Product
Corn-notill	84.513	83.445	90.254	88.652
Corn-min	63.221	73.956	77.734	78.131
Grass/pasture	94.231	96.923	97.692	97.692
Grass/trees	97.817	98.81	98.413	98.413
Hay-windrowed	99.625	99.625	99.625	99.625
Soy-notill	82.379	76.652	93.172	94.714
Soy-mintill	72.685	80.262	79.794	80.075
Soy-clean	55.66	75.472	88.679	88.208
Woods	98.113	98.821	98.585	98.585
Overall Accuracy	81.945	85.502	89.262	89.239

Table 5. Weighted combination based on Genetic algorithm

As can be inferred from Tables 3 and 4, using classification class accuracy for the weighted combination does not lead to a significant change in the classification performance, because both the classification accuracy and class accuracy are rough estimates of the weighting parameters so they are not so effective to be used directly as the weighting parameters. They are mainly related to mapping functions which are somehow converted into the weighting parameters. Definition of such a mapping function still relies on the analyst ad hoc decisions, which may not guarantee to obtain an optimal solution. Table 5 demonstrates that using genetic algorithm to find the weighting parameters can increase the classification accuracy more sensibly. Unlike the previous methods the genetic approach is a good tool for comprehensive search of the solution space and optimization of the weighting parameters.

6. CONCLUSIONS

Using an ensemble of classifiers based on different feature subsets can solve the two problems in the classification of hyperspectral images. In this paper in

order to improve the accuracy of simple combination of multiple classifiers a weighted combination has been used. Three methods for obtaining the weighting parameters have been implemented and tested on the

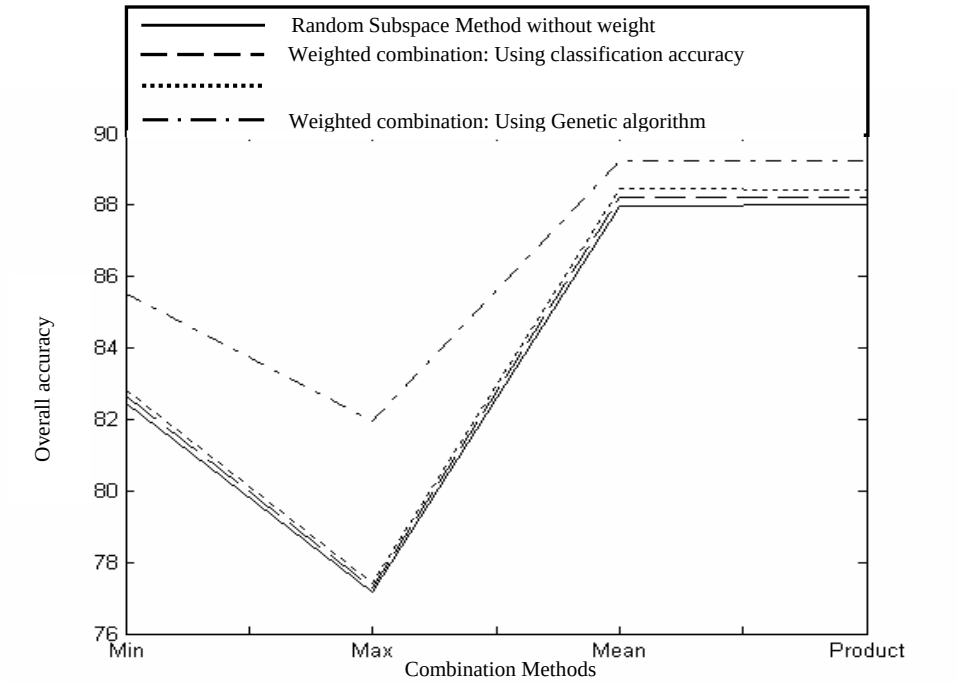

Figure 4. Overall accuracy of different weighting methods

dataset. The results have been compared to the simple random subspace method. Experimental results have shown that classification accuracy and class accuracy are only rough estimates of weighting parameters. Therefore, use of these parameters as the weight factors does not lead to significant enhancement of the classification performance. Results of this research confirm the suitability of genetic algorithms to find optimal weighting parameters for the weighted combination of multiple classifiers.

REFERENCES

G.F. Hughes. On the SUM accuracy of statistical pattern recognizers, *IEEE Transactions Information Theory*,pp.55 63, 1968.

David Landgrebe. On Information Extraction Principles for Hyperspectral Data, *School of Electrical and computer Engineering*, Purdue University, pp.168-173,July 1997.

T.G. Dietterich. Ensemble methods in machine learning. In Proc. Of MCS 2000, Lecture Notes in Computer Science, pp.1 15, 2000.

L. Breiman. Bagging predictors, *Machine Learning*, pp.123 140, 1996.

R. E. Schapire. The strength of weak learnability, *Machine Learning*, pp.197 227, 1990.

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

Lei Xu, Adam Kryzak, and Ching Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition, *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 418 435, May/June 1992.

J. Kittler, M. Hatef, Duin, R.P.W, and J. Matas. On combining classifiers. *IEEE Transactions Pattern Analysis and Machine Intelligence*, pp.226 239, 1998.

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

Lee, T., Richards, J. A. and Swain, P. H. (1987) 'Probabilistic and evidential approaches for multisource data analysis', *IEEE Transactions on Geoscience and Remote Sensing* 25,283-93

J.Holland. Adaption in Natural and Artificial Systems. University of Michigan Press, 1975.

F.J. Ferri, V. Kadirkamanathan, and J. Kittler. Feature subset search using genetic algorithms. *In IEE/IEEE Workshop on Natural Algorithms in Signal Processing* (NASP 93), pages 23/1–23/7, 1993.

J. Morris, D. Deaven, and K. Ho. Genetic-algorithm energy minimization for point charges on on a sphere.Physical Review B, 53(4):1740–1743, 1996.

[14] H. Zhang, B.-T. M¨uhlenbein. Genetic programming of minimal neural nets using Occam's razor. In S. Forrest, editor, Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93, pages 342–349, 1993.

[15] Y.Maghsoudi, A.Alimohammadi, M.J.Valadan Zoej and B. Mojaradi, 2006. Application of a Random Subspace Method for the Classification of Hyperspectral Data. Map India, New Delhi, India.