A Feature Selection Algorithm for Class Discrimination Improvement

Claudio De Stefano, Francesco Fontanella, Cristina Marrocco and Gilda Schirinzi
Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale
Università di Cassino, via G. Di Biasio 43,
03043 Cassino (FR), ITALY
{destefano, schirinzi, cristina.marrocco, fontanella}@unicas.it

***Abstract*—We propose a new feature selection algorithm for remote sensing image classification. Our approach has been especially devised for applications in which there is a large number of different features that can be potentially selected, implying that the search space is complex and high-dimensional. In this framework, our proposal is that of reformulating the feature selection problem as the search for the optimal subspace in which the different classes are more effectively discriminated. The search has been performed by using a genetic algorithm in which each individual encode the choice of a subspace, and its fitness is a measure of the class seperability in that subspace. The experimental results, performed on two databases, confirmed the effectiveness of the approach.**

Remote sensing, image classification, feature selection

I. Introduction

Production of land cover maps from satellite images is an important application field in remote sensing. To solve this problem, several feature based classification techniques have been proposed in the literature. However, the complexity and the performance of systems based on these techniques strongly depends on the set of features that are used [1]. The selection of an effective set of features is then a key problem to be solved in remote sensing image classification [2-5].

Feature selection consists in taking a set of candidate features and selecting a subset of them providing the most discriminative power among different classes, under some classification systems. This procedure can reduce not only the cost of recognition, by reducing the number of features that need to be collected, but in some cases it can also provide better classification accuracy: in fact it allows to discard the features that contribute to increase the variability within specimen belonging to the same class without increasing the capability to discriminate specimen belonging to different classes.

The need for selecting effective feature sets is particularly relevant in all the problems where a large number of different features can be defined. Examples of such problems are:

- applications where data taken by multiple sensors are fused;
- integration of multiple models, where all the parameters from the different models can be used for classification;
- data mining applications, where the goal is to recover the hidden relationships among the features.

In this framework, our proposal is that of reformulating the feature selection problem as the definition of the optimal mapping between an initial, possibly very high-dimensional feature space of dimension *N*, and a *M*-dimensional subspace (with $M \leq N$), in which the different classes are more effectively discriminated**.** The mapping can be established by using a binary vector **b**, belonging to a *N*-dimensional binary space, whose *k-th* element is equal to one if the *k-th* feature is included in the feature subspace, while is zero in the opposite case. The selection of the optimal mapping is performed by finding the *M*-dimensional subspace which maximizing a specifically defined class separability index. Under this hypothesis, finding the optimal solution means to find the binary vector **b** corresponding to the optimal *M*-dimensional subspace, where the search space has cardinality 2^N. This search space is generally quite complex because, in the majority of remote sensing image classification problems, *N* is a high number (typically greater then 100) and this implies that efficient search algorithms must be used.

In this context, systems based on Evolutionary Algorithms (EAs) seem to offer an effective methodology, as they are based on a powerful tool for finding solutions in complex high dimensional search spaces, where there is no *a priori* information about the samples distribution [6-8]. They typically work on a population of individuals each one representing a possible solution of the problem to be solved, which can be encoded in many different way. The algorithm starts by generating an initial population of individuals. Then, the "goodness" of each individual as solution of the problem at hand is measured by means of a *fitness* function. After this evaluation process, a new population is generated by choosing in the current one the individuals to be modified by suitable operators. These choices are made by a stochastic process called *selection*, which favors the reproduction of individuals having higher fitness in order to generate, if possible, new and better, individuals. Nevertheless, the individuals having lower fitness are not completely excluded from the process generating the new population. This process is repeated until one or more stop conditions are not satisfied.

In our case, we have used a Genetic Algorithm (GA) in which each individual *I* is an *N*-dimensional binary vector representing the set of features included in the *M*-dimensional subspace. As regards the fitness function, we have used the above mentioned class separability index. According to this choice, at the end of the evolutionary process, the individual *I* representing the optimal set of features, should be obtained.

The remainder of the paper is organized as it follows: Section II introduces the basic concepts relative to Genetic Algorithms. Section III provides a detailed description of the considered set of features. Section IV illustrates the architecture of the method. Section V reports the experimental results and some concluding remarks.

II. GENETIC ALGORITHMS

Genetic Algorithms are a class of search algorithms inspired by the mechanisms of Biological Evolution and Adaptation of species [6]. They have been successfully applied to a large variety of both numerical and combinatorial optimization problems with noisy, real-valued functions. To this purpose, the search space points are preliminarily coded into bit strings, and the function to optimize is interpreted as a fitness function, i.e. as a measure of the ability of the individual to survive and reproduce.

The features that make GA suitable for optimization problems can be summarized as follows:

- they do not require any specific knowledge about the problem at hand, but only values of the function to be optimized;
- they can explore several regions of the configuration space simultaneously and by means of the selection the search process is concentrated on the most promising regions.
- by using probabilistic transition rules, they are able to manage landscapes with a wide number of local optima.

Starting from a population of tentative solutions for the problem at hand, GA iteratively generates new solutions by means of a *selection mechanism* together with the genetics-inspired operators of *crossover* and *mutation*, hoping to evolve the population towards the most promising regions of the solution space. The algorithm is repeated until a termination criterion is satisfied. The solutions are encoded by means of “chromosomes” which consist of strings of “genes”, e.g. bits, whose values are called “allele”. The selection mechanism is aimed to choosing the chromosomes in the population in such a way that better chromosomes, i.e. those having higher fitness values, have higher chances to be chosen for reproduction and for genetic manipulation.

As regards the genetic operators, the *crossover* exchanges parts of two selected chromosomes, thus generating two offspring, while *mutation* works by randomly changing the allele in some location of the chromosome. It is worth noticing that these two operators must be applied with probabilities, called crossover rate and mutation rate respectively, whose values typically depend on the specific considered problem. This implies that preliminary experiments must be performed for selecting effective values for these probabilities. Finally, as termination criterion, it is possible to consider only the maximum number of generations or to add other criteria based on specific requirements on the fitness function to be optimized.

III. THE CONSIDERED FEATURES

There are variants of texture analysis methods, but the texture measures based on the grey-level co-occurrence matrices (GLCM) [9], are among the most widely used in the analysis of remote sensed imagery [5]. Previous studies indicated that GLCM is very suitable for finding texture information in images of natural scenes and performs well in classification applications [10]. Therefore, GLCM based textures should be appropriate also for the analysis of Landsat images.

Co-occurrence texture features are extracted from an image in two steps. First, in a user defined moving kernel (window) a grey level co-occurrence matrix (GLCM) are computed by measuring the spatial frequency of co-occurrence of pixel grey levels separated by a given displacement vector $\mathbf{d}=(d_x,d_y)$. Second, a set of different scalar quantities (features) can be computed for summarizing the information contained in a matrix GLCM.

A co-occurrence matrix is a two-dimensional array, **P**, in which both the rows and the columns represent a set of possible image values. Given a direction **d** in the moving window (0°, 45°, 90°, 135°), each element P_d[i,j] of the grey-level co-occurrence matrix represents the relative frequency with which two neighbouring pixels separated by a distance of d_x columns and d_y lines occur, one with grey tone *i* and the other with grey tone *j*.

The co-occurrence matrix P_d has dimension $L\times L$, where *L* is the number of grey levels in the kernel selected. Because the size of a GLCM matrix depends on the data range of pixel grey values, images of large numbers of data bits may result in large GLCM matrix sizes and require a large amount of computer resources (memory and CPU cycles). As a result, it is a practical necessity to reduce the co-occurrence matrix size for better computational performance. More importantly, because GLCM approximates the joint probability distribution of two pixels, reducing the matrix size will also reduce the number of zero-value cells in a matrix, which in turn will improve the statistical validity. A common technique to reduce GLCM matrix sizes is to rescale image grey levels to a lower data bit number. It has been demonstrated that reduction of grey levels causes only minor degradation (about 3%) in classification accuracy [11]. Therefore, the original 256 image grey level used in this study was rescaled to 32 level data before GLCM processing.

Starting from GLCM, different features can be computed. Haralick [9] originally proposed 14 different features; however, typically only a subset of these are used. He also suggested calculating each of the 14 measures for the four directions.

The set of features adopted in this paper include textural features such as Contrast (Inertia), Homogeneity, Entropy, Energy or Angular second moment (ASM), Correlation, Sum Average, Sum Variance, Sum Entropy, Difference Average, Difference Variance, Difference Entropy, Information Measure of Correlation 1, Information Measure of Correlation 2, and the spectral feature NDVI (Normalized Difference Vegetation Index). A detailed description of these features can be found in [9].

IV. THE ARCHITECTURE OF THE METHOD

As anticipated in the Introduction, we have reformulated the feature selection problem as the problem of finding the optimal mapping between the initial *N*-dimensional feature space and a *M*-dimensional subspace in which the different classes are more effectively discriminated**.** More specifically, let **Y** be the initial set of features, with cardinality *N* and let **X** a subset of **Y** with cardinality $M \leq N$. Selecting the optimal set of features means to find a subset $\mathbf{X} \subset \mathbf{Y}$ which maximizes a class separability index *J(***X***)* [5]:

$$\mathbf{X} = \underset{\mathbf{Z}\subset\mathbf{Y},\ |Z|\leq N}{\arg\max}\ J(Z) \tag{1}$$

The mapping between the initial *N*-dimensional feature space and a *M*-dimensional subspace, can be established by using a binary vector **b**, belonging to a *N*-dimensional binary space, whose *k-th* element is equal to one if the *k-th* feature is included in the feature subspace, while is zero in the opposite case. According to this assumption, the class separability index can be expressed as a function of **b** and eq. (1) becomes as:

$$\hat{\mathbf{b}} = \underset{\mathbf{b}\in B^N}{\arg\max}\, J(\mathbf{b}) \tag{2}$$

We have defined the class separability index *J(***b***)* in the following way:

$$J(\mathbf{b}) = \sum_{k=1}^{K} \sum_{i=1}^{n_K} \prod_{\substack{j=1 \\ j\neq k}}^{K} u\left[d\left(\mathbf{b}\circ Y_{ki}, \mathbf{b}\circ Y_{cj}\right) - d\left(\mathbf{b}\circ Y_{ki}, \mathbf{b}\circ Y_{ck}\right)\right] \tag{3}$$

where *u*() denotes the Heaviside function (unitary step function), *d*() denotes the Euclidean distance between two feature vectors and ∘ denotes the Hadamard product. In the formula *K* is the number of classes, n_k is number of elements belonging to the class *k*, Y_{ki} is the feature vector representing the i-th element of class *k*, and Y_{ck} is the centroid of class *k*. The centroids of all the classes have been determined by considering a training set of labeled pattern, represented as feature vectors in the initial *N*-dimensional feature space. For each class, the corresponding centroid is computed by averaging the components of all the feature vectors representing patterns of that class.

According to the above definition, the value of *J(***b***)* coincides with the recognition rate achievable by considering a Nearest Neighbor classifier which uses as prototypes the centroids Y_{ck} in the subspace identified by the vector **b**. The maximization of this function aims to find the subspace in which both the distances between each element of a class and the centroids of all the other classes are maximized, and the distances between each element of a class and the corresponding centroid are minimized.

The selection of the optimal vector **b** has been performed by using an evolutionary algorithm. In particular we have used a Genetic Algorithm in which the individuals are encoded as binary vectors belonging to the space 2^N. The fitness of each individual *I* is evaluated by computing the value *J*(*I*). Once the best solution is found, the corresponding features are used to implement a classifier in that subspace.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed approach has been tested by using two different Landsat Satellite databases. The first one (DB1 in the following) is the standard database *Satimage* included in the UCI database repository [12]. This database was generated from Landsat Multi-Spectral Scanner image data. Each frame consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum) and two are in the (near) infra-red. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m×80m. The database contains 6435 patterns belonging to 6 different classes, namely: *red soil*, *cotton crop*, *grey soil*, *damp grey soil*, *soil with vegetation stubble* and *very damp grey soil*. The patterns are organized in two sets of data: a training set (TR1 in the following) containing 4435 samples and a test set (TS1 in the following) containing 2000 samples. Each pattern corresponds to a 3×3 square neighborhood of pixels and is described by considering the pixel values in the four spectral bands of each of the 9 pixels in that neighbourhood. To each pattern is assigned as label the class of the central pixel. Thus, each pattern of the database is represented by a feature vector of 36 integer values in the range [0,255].

The second database (DB2 in the following) contains data relative to a satellite image of a residential area (city of Anzio, Italy) and was recorded by the ETM sensor, which has a ground spatial resolution of about 30m×30m and six spectral bands. Each pixel is a 8-bit binary word, with 0 corresponding to black and 255 to white. Each pattern corresponds to a 3×3 square neighborhood of pixels and contains 54 attributes (6 spectral bands × 9 pixels in the neighbourhood), resulting in a feature vector of 54 integer values in the range [0,255]. The data were divided into a training set (say TR2) with 800 samples and a test set (say TS2) with 712 samples, randomly extracted from the original Landsat 7 scene. In this scene five classes must be discriminated, namely: *water*, *grey soil*, *wood*, *urban area*, *sea sand* and *bare soil*.

In our experiments, we have considered the whole set of features described in Section III. In particular, the GLCM matrices were computed from the original data using a 3×3 moving window in four directions (0°, 45°, 90°, 135°) and for 13 textural measurements. As a result, each pattern in database DB1 has been described by using a feature vectors of 213 elements (4 pixel values in the spectral bands + 13 texture features × 4 directions × 4 spectral bands + 1 NDVI). Similarly, each pattern in database DB2 has been described by using a feature vectors of 319 elements (6 pixel values in the spectral bands + 13 texture features × 4 directions × 6 spectral bands + 1 NDVI).

Some preliminary experiments have been performed to tune the parameters of the Genetic Algorithm. Moreover, we have modified the fitness function adding a term which penalizes the individuals with a high number of bit 1 in the chromosome (i.e. those including a high number of feature). Thus, the fitness function F assume the form:

$$F(I) = J(I) + k\frac{N_{FT} - N_F(I)}{N_{FT}} \qquad (4)$$

where N_{FT} is the total number of features, $N_F(I)$ is the number of features considered by individual I and k is a constant assuming in our experiment the value 0.5. The GA has been executed 15 times for each run with different initial populations, in order to reduce the effects of the stochastic fluctuations due to the randomness of the search. At the end of a run, the best individual has been stored. Finally, the best individual discovered over the 15 runs has been selected and the corresponding set of features assumed as the result of the feature selection algorithm. In Table I we have reported the results relative to both databases.

TABLE I. NUMBER OF FEATURES DISCOVERED BY THE GA

	Statistics relative to 15 runs of the GA		
	Number of features of the best individual	***Average number of features***	***Standard deviation of the number of feature***
DB1	8	6,2	1,22
DB2	11	9,4	1,85

The effectiveness of the selected features has been tested by using them to implement a simple and widely adopted neural network classifier: the Multi Layer Perceptron (MLP) trained with the Back Propagation algorithm [13]. For the sake of comparison, we have also implemented a MLP classifier using as features only the values of the central pixel in each spectral band, and a MLP classifier considering also the information of the 3×3 neighbourhood. The results relative to DB1 (see Table II) show that the overall accuracy is improved when information relative to the neighbourhood of each pixel are added to the feature set. They also show that the proposed GA has selected an effective set of features which allows to obtain a further slight improvement in the recognition rate, but using only 8 features rather than 36, as in the previous case. Similar considerations can be repeated for the results relative to DB2 (see Table III), but in this case there are no improvements in the recognition rate adding neighbouring information. This is mainly due to the fact that the considered image is very fragmented and characterized by the presence of many small adjacent regions belonging to different classes. As it obvious, in this situation the probability that adjacent pixels belong to different classes becomes higher, making less reliable the use of neighbouring information for pixel description. On the contrary, the set of features selected by the GA produces a considerable improvement in the recognition rate with a negligible increase of the number of features.

In conclusion, the experiments confirmed the effectiveness of the proposed approach, which allows to reduce very much the number of selected features without reducing, or in some case increasing, the obtainable performance: for the database DB1, the initial feature set includes 213 features and the GA has selected only 8 features. Similarly, for database DB2, the initial feature set includes 319 features and the GA has selected only 11 features: the use of such features produced an increase of the recognition rate from 74.8 % to 77.6%.

TABLE II. RESULTS OF THE MLP CLASSIFIER RELATIVE TO DB1

hidden nodes	***4 features Spectral Bands***		***36 features 3x3 Neighbourhood***		***8 features Selected by GA***	
	TR1	**TS1**	**TR1**	**TS1**	**TR1**	**TS1**
30	85,19%	**84,00%**	90,38%	87,60%	89,10%	87,40%
40	85,10%	83,80%	90,14%	87,20%	89,16%	**88,00%**
50	85,15%	83,80%	90,38%	**87,80%**	89,26%	87,40%

TABLE III. RESULTS OF THE MLP CLASSIFIER RELATIVE TO DB2

hidden nodes	***6 features Spectral Bands***		***54 features 3x3 Neighbourhood***		***11 features Selected by GA***	
	TR2	**TS2**	**TR2**	**TS2**	**TR2**	**TS2**
30	84,95%	74,80%	92,37%	**74,80%**	87,73%	77,20%
40	84,85%	74,80%	92,28%	74,60%	87,75%	**77,60%**
50	85,25%	**75,00%**	92,90%	72,20%	88,98%	77,00%

REFERENCES

[1] L. O. Jimenez and D. A. Landgrebe, “Supervised classification in high dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data,” *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 39–54, Mar. 1998.

[2] H. S. Solberg, A. K. Jain. „Texture Fusion and Feature Selection Applied to ***SAR*** Imagery”. *IEEE Trans. Geosci. Remote Sens*, 35(2), pp. 475-479, 1997.

[3] B. Serpico, L. Buzzone, “A New Search Algorithm for Feature Selection in Hyperspectral Remote Sensing Images”, *IEEE Trans. Geosci. Remote Sens*., pp. 1360-1367, 2001

[4] R. Huber, L. V. Dutra, “Feature Selection for ERS-1/2 in SAR Classification: High Dimensionality Case”**,** *Proc. of IGARSS’98*, pp. 1907-1605, 1998.

[5] A. Jain, D. Zongker, “Feature Selection: Evaluation, Application, and Small Sample Performance” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 153-158, 1997..

[6] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley, 1989

[7] L.P. Cordella, C. De Stefano, F. Fontanella, A. Marcelli, “Evolutionary Generation of Prototypes for a Learning Vector Quantization Classifier”, in *Lecture Notes in Computer Sciences*, F. Rothlauf et al. eds., Springer-Verlag, vol. 3907, 2006, pp. 391-401.

[8] C. De Stefano, A. Della Cioppa, A. Marcelli, “An Evolutionary Approach for Dynamic Configuration of Multi-expert Classification Systems “ Proc. of the IEEE World Congress on Computational Intelligence, Vancouver BC, Canada, July 16-21, 2006.

[9] R. M. Haralick, K. Shanmugam, I. Dinstein, “Textural features for image classification", *IEEE Trans. Syst., Man, Cybern*, pp. 610-621, 1973.

[10] T. R. Reed, J. M. Hans-Du-Buf, "A Review of Recent Texture Segmentation and Feature Extraction Techniques," *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol 57(3), PP. 359-372, 1993.

[11] D. J. Marceau, P. J. Howarth, J.-M. Dubois, D. J. Gratton, “Evaluation of the Grey-Level Co-Occurrence Matrix Method For Land-Cover Classification Using SPOT Imagery”, *IEEE Trans. Geosci. Remote Sens.*, vol. 28, pp. 513-519, 1990.

[12] C. L. Blake and C. J. Merz, “Uci repository of machine learning databases, University of California, Irvine.” [Online]. Available: http://www-ics.uci.edu/mlearn/MLRepository.html.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no.9, pp. 533–536, 1986.