A GA-based Feature Selection Algorithm for Remote Sensing Images

C. De Stefano, F. Fontanella and C. Marrocco

Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e
Matematica Industriale (DAEIMI)
Università di Cassino
Via G. Di Biasio, 43 02043 Cassino (FR) – Italy
{destefano,fontanella,cristina.marrocco}@unicas.it

Abstract. We present a GA–based feature selection algorithm in which feature subsets are evaluated by means of a separability index. This index is based on a filter method, which allows to estimate statistical properties of the data, independently of the classifier used. More specifically, the defined index uses covariance matrices for evaluating how spread out the probability distributions of data are in a given $n-$dimensional space. The effectiveness of the approach has been tested on two satellite images and the results have been compared with those obtained without feature selection and with those obtained by using a previously developed GA–based feature selection algorithm.

1 Introduction

In the last years, the interest about the feature selection problem has been increasing. In fact, new applications dealing with huge amounts of data have been developed, such as data mining [1] and medical data processing [2]. In this kind of applications, a large number of features is usually available and the selection of an effective subset of them, i.e. a subset that allows to maximize the performance of the subsequent clustering or classification process, represents a very important task. The feature selection problem plays also a key role when set of features belonging to different domains are used: this typically happens in applications such as remote sensing [3] or handwriting recognition [4].

In a classification task in which the objects to be classified are represented by means of a set of features, the feature selection problem consists in selecting, from the whole set of available features, the subset of them providing the most discriminative power. The choice of a good feature subset is a crucial stage in any classification process since:

- If the considered set features does not include all the information needed to discriminate patterns belonging to different classes, the achievable performances may be unsatisfactory regardless the effectiveness of the learning algorithm employed.

- The features used to describe the patterns determine the search space to be explored during the learning phase. Then, irrelevant and noisy features make the search space larger, increasing both the time and the complexity of the learning process;
- The computational cost of classification depends on the number of features used to describe the patterns. Then, reducing such number results in a significant reduction of this cost.

When the cardinality N of the candidate feature set Y is high, the problem of finding the optimal feature subset, according to a given evaluation function, becomes computationally intractable because of the resulting exponential growth of the search space, made of all the 2^N possible subsets of Y. Many heuristic algorithms have been proposed for finding near–optimal solutions [5]. Among these algorithms, greedy strategies that incrementally generate feature subsets has been proposed. Since these algorithms do not take into account complex interactions among several features, in most of the case they lead to sub–optimal solutions.

GA's, which have demonstrated to be an effective search tools for finding near–optimal solutions in complex and non–linear search spaces [6] as is the case of feature selection problems, have been widely used to solve feature selection problems [7, 8]. Moreover, comparative studies have demonstrated the superiority of GA's in feature selection problems involving large numbers of features [9].

We propose a GA–based feature selection approach in which each individual, whose genotype encode the selection of a feature subset, is evaluated by means of a separability index. This index is based on a filter method which takes into account statistical properties of the input data in the subspace represented by that individual and is independent of the used classification scheme. The proposed index uses covariance matrices, which are the generalization of the variance of a scalar variable to multiple dimensions. These matrices estimate how spread out the probability distributions are in the considered $n-$dimensional space. Given a feature subset X, the proposed index estimates the separability of the classes in X by evaluating two aspects: (i) how patterns belonging to a given class are spread out around the corresponding class mean vector (the centroid); (ii) distances among class mean vectors.

The effectiveness of the proposed approach has been tested on two datasets extracted from satellite images. In both datasets image pixels are represented by feature vectors in high–dimensional spaces. These vectors describe textural measurements based on grey–level co–occurrence matrices (GLCM) [10]. The effectiveness of the selected features has been tested on a neural network classifier. The obtained results have been compared with those obtained without feature selection and with those obtained by using a previously presented approach [11].

The remainder of the paper is organized as follows: in Section 2 the problem of feature selection is described, Section 3 illustrates the GA–based method implemented, while the separability index used for subset evaluations is presented in Section 4. In Section 5 the experimental results are detailed. Finally, Section 6 is devoted to conclusions.

2 The Feature Selection Problem

Feature Selection (FS) is one of the first stages in any classification process in which data are represented by feature vectors. Its role is to reduce the number of features to be considered later in the classification stage. This task is performed by removing irrelevant and noisy features from the set of the available features. Feature selection is accomplished by reducing as much as possible the information loss due to the feature set reduction. Moreover, this selection process should not reduce classification performances.

In a classification process involving a dataset $\mathcal{D}$, in which patterns are represented by means of a set $Y = \{1, 2, \ldots, N\}$ of N features, the feature selection problem can be formulated as follows: find the subset $S \subseteq Y$ of n features which optimizes the function J. Given a generic subset $X \subseteq Y$, $J(X)$ measures how well the patterns in D, belonging to different classes, are discriminated by using the n features in X. The methods implemented by the function J can be divided in two wide class:

- *filter methods* which evaluate a feature subset independently of the classifier but are usually based on some statistical measures of distance between the patterns belonging to different classes.
- *wrapper methods* which are based on the classification results achieved by a certain classifier.

Filter methods are usually faster than wrapper ones, as the latter requires the training of the classifier used for each evaluation and this process may make this approach unfeasible when a large number of features is involved. Moreover, while filter–based evaluations are more general, as they give statistical information on the data, wrapper–based evaluations may give raise loss of generality because they depend on the specific classifier used.

Once the evaluation function $J(X)$ has been chosen, the feature selection problem becomes an optimization problem whose search space is the set of all the subsets of Y. The size of this search space is exponential (2^N). As a consequence, the exhaustive search for the optimal solution is unfeasible for those problems involving a large number of features ($N > 50$). Search strategies like branch and bound [12] have been proposed to strongly reduce the amount of evaluations, but the exponential complexity of the problem still remains. The exponential size of the search space for the feature selection problem makes appropriate the use of heuristic algorithms, for finding near–optimal solutions. Among these search algorithms, greedy search strategies are computationally advantageous but may lead to suboptimal solutions. They come in two flavours: forward selection and backward elimination. Forward selection strategies generate near–optimal feature subsets by a stepwise procedure which starts with an empty set and adds to the so far built subset the feature, among those not yet selected, that more increases the evaluation function J, this procedure is repeated until a stop criterion is not satisfied. In backward elimination, instead, the whole subset of feature is initially considered, and at each step the feature that least reduce the evaluation function is eliminated. Both procedures are optimal at each step, but they cannot discover complex interactions among several features, as is the case in most of the real world feature selection problems. Then heuristic search algorithms, like genetic algorithms and simulated annealing seems to be appropriate for finding near–optimal solutions which take into account multiple interactions among several features.

3 Genetic Algorithms for Feature Selection

The principles governing the phenomena of natural evolution have been widely studied by mathematicians and computer scientist since the end of the 50's of the last century. These studies have led to the development of a new computational paradigm named evolutionary computation [6], which during the last decades has shown to be very effective as methodology for solving optimization problems whose search space are discontinuous and very complex. In this field, genetic algorithms (GA's in the following) represent a subset of these optimization techniques, in which solutions are represented as binary strings. To these strings, operators such as selection, crossover and mutation are applied to a population of competing individuals (problem's solutions). GA's have been applied to a wide variety of both numerical and combinatorial optimization problems [7].

GA's can be easily applied to the problem of feature selection, as given a set Y having cardinality equal to N, a subset X of Y $(X \subseteq Y)$ can be represented by a binary vector $\mathbf{b}$ having N elements whose i-th element is set to 1 if the i-th features is included in X, 0 otherwise. Besides the simplicity in the solution encoding, GA's are well suited for these class of problems as the search in this exponential space is very hard since interactions among features can be highly complex and strongly nonlinear. Some studies on the GA's effectiveness in solving features selection problems can be found in [7, 8].

The system presented here has been implemented by using a generational GA, which starts randomly generating a population of P individuals. Each individual's chromosome is a binary vector encoding a feature subset that represents an allowed solution of the problem. The value of the i-th element is set to 1 according a given probability (called *one_prob*), which represents a parameter algorithm and is usually set to 0.1 at the aim to force the early stage of the search toward solutions having a small number of features. Afterwards, the fitness of the generated individuals is evaluated. This fitness takes into account two terms, the former measures the separability of the patterns belonging to the different classes in the problem at hand, in the feature subset encoded by the individual, while the latter terms measures the cardinality of the subset so as to favour solutions containing a smaller number of features. After this evaluation phase a new population is generated, by first copying the best e individuals of the initial population in order to implement an elitist strategy. Then $(P - e)/2$ couples of individuals are selected using the tournament method, to control loss of diversity and selection intensity. The one point crossover operator is applied to each of the selected couples, according to a given probability factor p_c. Afterwards, the mutation operator is applied. As regards this operator, two different probability factor p_0 and p_1 have been defined. These factors represent the mutation probability respectively of the 0's and 1's in the chromosome. These two different probability factors have been adopted since in a chromosome the 0's and 1's occurrences can be very different: typically in a chromosome 0's are much more than 1's. This fact is due to both the generation procedure of the individuals in the initial population and the fitness function, which favours the individuals with a smaller number of features. As a consequence, as in an individual the number of the 1's is much smaller than the that of 0's the value of p_1 is set much smaller than that of p_0. The purpose is to make, on average, the probability mutation of the 1's about equal to that of the 0's. Finally these individuals are added to the new population. The process just described is repeated for N_g generations.

4 The Separability Index

In any EC–based algorithm the design of a suitable fitness function for the problem to be solved is a crucial task. To be successful in feature selection problems, the fitness function must be able to effectively evaluate how well patterns belonging to different classes are discriminated in the subspace represented by the solution to be evaluated. The fitness function adopted is based on a well known class separability index J, which is usually adopted in Multiple Discriminant Analysis for finding a linear transformation that allows to reduce the dimension of the data.

According to so-called Fisher Criterion, the separability index J has been defined by using covariance matrices, which measure how spread out the probability distribution of the data is in the considered space. In a particular $n-$dimensional space, given a set of patterns belonging to different classes, the $i-$th class can be described by using its covariance matrix Σ_i, which is obtained only considering the patterns belonging to class i. Note that the covariance matrix Σ_i contains information about variability of data points belonging to the i–th class around their mean value μ_i.

The separability index J used for finding the most discriminative features, makes use of this dispersion concept. In particular, the classes information is condensed in two scatter matrices Σ_B and Σ_W:

$$\Sigma_W = \sum_i P(\omega_i)\Sigma_i$$

$$\Sigma_B = \sum_i P(\omega_i)(M_i - M_0)(M_i - M_0)^T$$

where $P(\omega_i)$ denotes the prior probability of the i–th class, Σ_i and M_i are respectively the covariance matrix and the mean vector of i–th class, and M_0 denotes the overall mean:

$$M_0 = \sum_i P(\omega_i)M_i \tag{1}$$

Note that Σ_W is a *within-class scatter matrix*, as it measures the spread of the classes about their means, while Σ_B is a *between-class scatter matrix*, since it measures distances between class mean vectors, i.e. centroids.
Given a feature subset X, the separability index $J(X)$, has been defined as the ratio between a *within-class scatter matrix* Σ_W and a *between-class scatter matrix* Σ_B:

$$J(X) = trace(\Sigma_W^{-1}\Sigma_B) \tag{2}$$

High values of the separability index $J(X)$ indicate that in the subspace represented by the feature subset X the class means are well separated and, at the same time, patterns appear to be not much spread out around their means values.

5 Experimental Results

The proposed approach has been tested on data represented by feature vectors in high dimensional spaces (> 200). These feature vectors describe textural measures based on the grey–level co–occurrence matrices (GLCM) [10], computed on patterns belonging to datasets extracted from Landsat Satellite images. Two datasets have been considered and for each of them 20 runs have been performed. The reported results are those obtained using the individual having the highest fitness among those obtained during the 20 performed runs. Some preliminary trials have been performed to set the basic evolutionary parameters reported in Table 1. This set of parameters has been used for all the experiments reported below.

Given an individual I, its fitness value F has been computed by applying the formula:

$$F(I) = J(I) + k\frac{N_{FT} - N_F(I)}{N_{FT}} \tag{3}$$

where $J(I)$ is the separability index, described in Section 4, computed on the subset of features represented by I, while N_{FT} is the maximum number of features available and $N_F(I)$ is the cardinality of the subset represented by I, i.e. number of bits equal to 1 in its chromosome. Finally, k is a constant value; this constant is used so as to weight the second term in the (3), which is in inverse proportion to the number of features in I. The role of this second term is essential in order to avoid an excessive increase of the number of features, as may result from the selection process because of the monotonic trend of the index. Thanks to this term individuals having a smaller number of features are favoured.

5.1 The Datasets

The first dataset (DS1 in the following) used for testing the proposed approach is the standard dataset Satimage included in the UCI dataset repository. This dataset was generated from 4–band Landsat Multi-Spectral Scanner image data.

Table 1. Values of the basic evolutionary parameters used in the experiments. Note that p_0 and p_1 depend on the chromosome length. For the experiments involving the first dataset (DS1) the 0's probability mutation has been set equal to 0.0047 (0.047 for the 1's), while for the second dataset (DS2) this probability has been set equal to 0.003 (0.03).

Parameter	**symbol**	**value**
Population size	$\mathcal{P}$	100
Tournament size	$\mathcal{T}$	6
elithism size	$\mathcal{E}$	5
Crossover probability	p_c	0.4
Mutation probability of 0's	p_0	$1/N_F$
Mutation probability of 1's	p_1	$10/N_F$
Number of Generations	N_g	1500

Each Landsat 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 patterns belong to 6 different classes, namely: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble and very damp grey soil.
DS1 contains 6435 patterns, organized in two sets of data: a training set (TR1 in the following) containing 4435 patterns and a test set (TS1 in the following) containing 2000 patterns. Each pattern corresponds to a 3×3 square neighbourhood 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 dataset is represented by a feature vector of 36 integer values in the range [0,255].
For each original pattern, a new feature vector has been built by computing its Grey Level Co–occurrence Matrix (GLCM) [10] with a moving window equal to the 3×3 neighbourhood. For each of the four spectral bands, 4 GLCM in the directions $(0°, 45°, 90°, 135°)$ has been computed. As a consequence for each pattern 16 GLCM has been computed. Afterwards, for each GLCM 13 textural features have been computed. Finally, to each new feature vector the values of the 4 spectral bands and the spectral feature NDVI (Normalized Difference Vegetation Index) has been added. Then in our experiments each pattern in DS1 has been described by using a feature vectors of 213 elements (13 texture features × 4 directions × 4 spectral bands + 4 pixel values in the spectral bands + 1 NDVI). The constant k in 3 has been set to 0.2 for this dataset.

The second dataset (DS2 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 on the Landsat 7 satellite, 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

Table 2. Number of features found by GA1 and GA2.

	Best ind.		**Average**		**std. dev.**	
	GA1	GA2	GA1	GA2	GA1	GA2
DS1	8	9	6.05	9.1	1.1	0.3
DS2	11	4	9.4	4.5	1.8	0.5

square neighbourhood 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 patterns and a test set (TS2) with 712 patterns, 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. As for DS1, also patterns in DS2 has been described by using the GLCM textural features mentioned above. In this case, as six spectral bands are available, feature vectors of 319 elements (13 texture features × 4 directions × 6 spectral bands + 6 pixel values in the spectral bands + 1 NDVI) has been built. In this case the value of the constant k has been set to 4.0.

5.2 Comparison Findings

In order to asses the effectiveness of the implemented system, the selected features have 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.
The results of our GA–based method (GA2 in the following) have been compared with those obtained by another GA-based feature algorithm previously proposed in [11] (GA1 in the following), which used a different separability index for feature subset evaluations. That index was computed by using a training set Tr, containing C classes, of labelled patterns represented as feature vectors in the initial N-dimensional feature space. Given an individual I, representing a subset X, its separability index was computed as follows: first, for each class the corresponding centroid is computed in X by averaging the components of the feature vectors belonging to that class; Then, each pattern in Tr is labelled with the label of the nearest centroid (nearest neighbour rule) in X; Finally, the percentage of patterns correctly labelled is assumed as separability index of X.

Table 2 shows the number of features obtained by GA1 and GA2. As regards the results on DS1, GA1 and GA2 have found, on average, respectively 6.05 and 9.1 features. Then, on this dataset our method has found a higher number of features. However, the number of features of the best individuals (8 for GA1 and 9 for GA2) are comparable. Probably this means that though GA2 finds, on average, smaller feature subsets, it obtains good performances only when larger subsets are found. Moreover, in GA2 the number of features of the best individual is about equal to the average one. As regards the results on DS2, the average number of features found by GA1 and GA2 is equal respectively to 9.4 and 4.5.

Then, in this case GA2 has been able to find smaller feature subsets than GA1. Moreover, also in this case the number of features of the GA2 best individual is lower than the average one, while for GA1 the best individual has a number of features larger than the average. This fact seems to confirm the hypothesis stated above that GA2 obtains good performances only when a larger number of features is used. Finally, it is worth noting that GA2 standard deviations are lower than those of GA1. This fact indicates that the separability index used in GA2 is able, almost always, to find the smaller subsets providing, according to that index, the most discriminative power.

In Table 3 the recognition rates achieved by the MLP classifier on the best feature subsets found by GA1, GA2 and on the 3×3 neighbourhood feature set of the original datasets are reported. On both the databases analyzed, GA2 has achieved better results than those obtained by GA1. As regards DS1, GA2 has further improved the good rates obtained by GA1. Also for DS2 the MLP has obtained better results on the features selected by GA2 than on those selected by GA1. It is also worth noting that, for both datasets, GA2 has obtained better results than GA1 for each value of hidden nodes. This fact demonstrates that GA2, whatever the number of nodes in the hidden layer, always gives better results than GA1. Finally, on both datasets, the selected features has been able to significantly improve the performances obtained by the feature sets made of the spectral band values in the 3×3 neighbourhood of each pixel.

6 Conclusions

A new GA-based algorithm for feature selection in high dimensional feature spaces has been presented. The proposed approach uses a separability index for evaluating feature subsets. This index is based on a filter method, which estimates statistical properties of the data and is independent from the classifier used. The index is able to effectively measure both the spreading of the patterns around their mean vectors and the distances of the mean vectors of the different classes.

Table 3. Classification rates (expressed in percentages) of the MLP classifier on DS1 (left) and DS2 (right) on the best feature subsets found by GA1, GA2 and the 3×3–neighbourhood feature sets of the original datasets. The actual number of features used is reported in parenthesis. In the column N the number of hidden nodes of the MLP classifier is reported.

MLP results on DS1						
N	**3×3 neigh. (36)**		**GA1 (8)**		**GA2 (9)**	
	TR1	TS1	TR1	TS1	TR1	TS1
30	90.38	87.60	89.10	87.40	89.99	89.25
40	90.14	87.20	89.16	**88.00**	89.63	88.85
50	90.38	**87.80**	89.26	87.40	89.99	**89.50**

MLP results on DS2						
N	**3×3 neigh.(54)**		**GA1 (8)**		**GA2 (9)**	
	TR1	TS1	TR1	TS1	TR1	TS1
30	92.37	**74.80**	87.73	77.20	83.62	**78.65**
40	92.28	74.60	87.75	**77.60**	84.50	77.67
50	92.90	72.20	88.98	77.00	84.00	78.09

The proposed approach has been tested on two datasets extracted from satellite images. From these data a wide set of GLCM textural features has been computed and from this set the near–optimal subsets has been found by using the GA–based method. The results have been compared with those obtained by another GA–based method. The comparison has been done by implementing a MLP, which has been trained and tested on the best subsets found by the two methods. The accuracies on the test sets have been compared. For both the datasets, the comparison has demonstrated that our method is able to found subsets that yields better accuracies than those obtained by using the subsets found by the other method.

References

1. Martin-Bautista, M., Vila, M.A.: A survey of genetic feature selection in mining issues. In: Proc. 1999 Congress on Evolutionary Computation (CEC99). (1999) 1314–1321
2. Puuronen, S., Tsymbal, A., Skrypnik, I.: Advanced local feature selection in medical diagnostics. In: Proc. 13th IEEE Symp. Computer-Based Medical Systems. (2000) 25–30
3. Hung, C., Fahsi, A., Tadesse, W., Coleman, T.: A comparative study of remotely sensed data classification using principal components analysis and divergence. In: Proc. IEEE IntlConf. Systems, Man, and Cybernetics. (1997) 2444–2449
4. I.S. Oh, J.L., Suen, C.: Analysis of class separation and combination of class-dependent features for handwriting recognition. IEEE Trans. Pattern Analysis and Machine Intelligence **21** (1999) 1089–1094
5. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3** (2003) 1157–1182
6. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley (1989)
7. Lee, J.S., Oh, I.S., Moon, B.R.: Hybrid genetic algorithms for feature selection. IEEE Trans. Pattern Anal. Mach. Intell. **26** (2004) 1424–1437
8. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. IEEE Intelligent Systems **13** (1998) 44–49
9. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern recognition. Pattern Recognition **33** (2000) 25–41
10. Haralick, R.M.: Textural features for image classification. IEEE Trans. System, Man and Cybernetics **3** (1973) 610–621
11. De Stefano, C., Fontanella, F., Marrocco, C., Schirinzi, G.: A feature selection algorithm for class discrimination improvement. In: IEEE International Geoscience and Remote Sensing Symposium. (2007)
12. Yu, B., Yuan, B.: A more efficient branch and bound algorithm for feature selection. Pattern Recognition **26** (1993) 883–889