Combining single class features for improving performance of a two stage classifier

I. INTRODUCTION

In recent years sensors capable of acquiring a variety of features have been developed in fields such as remote sensing [1] and medical imaging [2]. On one hand, large numbers of features may result useful in applications requiring to discriminate between samples exhibiting slight differences. On the other hand, the large dimensional data spaces generated using such sensors give place to challenging methodological problems. In the context of supervised classification, the most important methodological facing issue is the Hughes effect, also known as "the curse of dimensionality", which occurs when number of features and number of available training samples are unbalanced [3]. This effect may make classifiers unsuccessful. In order to cope with such challenging and hard problem, feature selection techniques aiming at reducing the dimensionality of the original feature space have been used [4].

Feature selection implies searching, in the whole set of available features, the subset exhibiting the most discriminant power. When the dimensionality \( NF \) of the whole set is high, such search problem becomes computationally intractable because the number of possible solutions increases exponentially with \( NF \). Therefore, heuristic algorithms become necessary for finding near-optimal solutions [5].

Any feature selection algorithm is made of two parts: a search strategy for selecting feature subspaces and an evaluation function able to estimate how well classes are separated in a given feature subspace. As regards evaluation methods, those proposed in the literature can be divided in two wide classes: (i) \textit{filter} methods, which evaluate feature subsets on the basis of some statistical distance measures between the samples belonging to different classes. (ii) \textit{wrapper} methods, which are based on the classification results achieved by a given classifier. Filter methods are independent of the classifier used afterwards and are usually faster than wrapper ones, as these latter require a new training of the classifier at each evaluation.

As for search strategies, greedy approaches are computationally advantageous, but quite often lead to suboptimal solutions. Such kind of algorithms explore the available search space by a stepwise procedure that iteratively improves the current subset by choosing the feature that more increases the evaluation function. This strategy is optimal at each step, but it cannot discover complex interactions among several features, as it is the case in most real world feature selection problems. For this reason, heuristic search algorithms seem more appropriate because they are able to take into account complex interactions among several features. In this framework, Genetic Algorithms (GA) have demonstrated to be very effective [6] when a high number of features exhibiting complex interactions is involved.

In a previous paper [7], we presented a GA–based feature selection algorithm for determining the feature subset most convenient for classifying a set on \( N \) classes. Even if the results were quite good, the error analysis highlighted that errors mainly concentrated on some classes, suggesting that using features specific for those classes could improve the whole classification performance.

According to the above considerations, in this paper we propose an approach in which a problem with a set of \( N \) classes is subdivided into a set of \( N \) problems. In each problem, a GA–based feature selection algorithm is used for finding the best subset of features. These subsets are then used for training \( N \) classifiers. We chose Support Vector Machine (SVM) classifiers because they are less sensitive to the curse of dimensionality [8] and are specifically devised for two class problems. In the classification phase, unknown samples are given in input to each of the trained SVM’s by using the corresponding subspaces. To manage conflicting responses, we propose a two stage architecture: a sample that, for some reason, cannot be classified at the first stage is sent to a second stage made of a supplementary SVM.
The training of such SVM is made by describing each sample in the training set with the union of the feature sets corresponding to the classifiers that recognized the sample in the first stage. Those samples that, in the first stage, were not assigned to any class, are described by using the whole feature set.

The proposed approach has been tested on a real world dataset containing hyperspectral image data. The obtained results have been compared both with those obtained by some standard feature selection methods [9], [10] and with those presented in [11]. Specifically, in [11], an approach similar to ours is used: GA’s are used for finding near-optimal feature subsets for each class and these subsets are then used for training SVM-based classifiers. However, the wrapper method is used, and the “winner takes all” rule is adopted for combining conflicting responses in the classification phase. Compared to the considered methods, our method achieves better recognition rates.

II. OUR APPROACH

The system architecture we propose is shown in Fig. 1. According to our approach, system training is performed in three phases: first a problem with a set of $N$ classes $C = \{C_{11}, C_{12}, \ldots, C_{1N}\}$ is divided into a set of $N$ problems. For each problem, the best subset of features $FS_i$, which allows the system to better discriminate the class $C_{ii}$ ($i = 1, 2, \ldots, N$) against all the others (i.e., $C - C_{ii}$) is found by means of a GA. In the second phase, for each class $C_{ii}$, a different SVM is trained with samples described using the features of the subspace $FS_i$, that in this way is associated to each SVM. Thus, the $i$-th SVM is able to discriminate the samples of the $i$-th class from those belonging to any other.

In the third phase, a supplementary SVM classifier is trained using the same training set samples used in the second phase, each described according to the following criterion: if the sample was recognized by one or more SVM’s, it is described using the union of the feature subsets relative to those SVM’s; if the sample was not recognized by any SVM, it is described by using the whole set of features. The set of the $N$ SVM’s trained in the second phase, makes up the first stage of our classification system. The supplementary SVM classifier is the second classification stage. In the classification phase, an unknown sample is given in input to each of the SVM’s of the first stage by using for its description the feature subset associated to that SVM. The box $FS_i$ in figure 1 represents the filter selecting the proper feature subset for the $i$-th SVM. Different cases may occur: (i) only the $i$-th SVM recognizes the sample as belonging to the $i$-th class; (ii) more that one SVM recognize the sample; (iii) no SVM recognizes the sample. In case (i) the sample is assigned to the $i$-th class, while in cases (ii) and (iii) the sample is passed to the second stage. In case (ii) the sample is described by using the union of the feature subsets associated to those SVM’s that recognized it; in case (iii), the sample is described by using the whole set of features.

In order to find the $i$-th subset of features, able to discriminate the $i$-th class, we used the generational GA already presented in [7]. For the sake of clarity, the algorithm is summarized in the following. First, a population of $P$ individuals is randomly generated. Each individual is a binary vector encoding a feature subset that represents a solution of the problem. The value of the $i$-th vector element in the initial population is set to 1 according to a low probability value, in order to force the early stage of the search towards solutions having a small number of features. At any successive evolution step, the fitness of the individuals in the current population is evaluated and a new population is generated, by first copying the best $e$ individuals, in order to implement an elitist strategy. Then, $(P-e)/2$ couples of individuals are selected: the tournament method has been chosen to control loss of diversity and selection intensity. The one point crossover operator is then applied to each of the selected couples, followed by the application of the mutation operator. Both operators are applied according to given probabilities. Eventually, these individuals are added to the new population. The process just described is repeated for $N_g$ generations.

In a feature selection problem, the fitness function of an individual should be able to effectively evaluate how well samples belonging to different classes are discriminated in the subspace represented by that individual. In our case, the fitness function is made of two terms: a separability index $J$ and a term $W$ related to the number of used features.

The separability index is based on a generalization of the so-called Fisher Criterion used in Multiple Discriminant Analysis. It has been defined by using covariance matrices, which measure data scattering in the considered space. Given an $n$-dimensional space, and a set of samples belonging to different classes, the $i$-th covariance matrix $\Sigma_i$ contains information about variability of samples belonging to the $i$-th class around their mean value $\mu_i$. In particular, class information is concentrated in two scatter matrices $\Sigma_B$ and $\Sigma_W$:

$$\Sigma_W = \sum_i P(\omega_i)\Sigma_i; \quad \Sigma_B = \sum_i P(\omega_i)\Delta_i\Delta_i^T$$

where $P(\omega_i)$ and $\Sigma_i$ denote respectively the a priori probability and the covariance matrix of the $i$-th class. As regards $\Delta_i$, it denotes the vector $(M_i - M_0)$, where $M_i$ is the mean vector of the $i$-th class, and $M_0$ denotes the overall mean, computed on all the training set samples. Note that $\Sigma_W$ is a within-class scatter matrix, as it measures the dispersion of samples within a class, while $\Sigma_B$ is a between-class scatter matrix, since it measures distances between class mean vectors, i.e. centroids.

Given an individual $I$, representing a feature subset $X$, the
The separability index $J(I)$, has been defined as follows:

$$J(I) = \text{trace}(\Sigma_1^{-1}\Sigma_B)$$  \hspace{1cm} (1)

High values of the index $J$ indicate that the classes centroids are well separated in $X$ and, at the same time, samples are gathered around their centroid values.

The second term $W(I)$, making up the fitness function, is defined as:

$$W(I) = \frac{N_T - N_F(I)}{N_T}$$  \hspace{1cm} (2)

where $N_T$ is the total number of available features and $N_F(I)$ is the number of features in the considered individual (its value varies in the range $[1, N_T]$). $W$ is proportional to the number of features in the subset $X$ and it has been normalized with respect to the total number of available features. It assumes the maximum value when only one feature is used, and linearly decreases as the number of features increases, reaching the value 0 when the whole set of available features is used. Its role is to avoid an excessive increase of the number of features during the search process. Thanks to this term, for a given $J$, individuals having a smaller number of features are favoured.

In conclusion, the fitness value $F$ of an individual $I$ is given by:

$$F(I) = J(I) + C \cdot W(I)$$  \hspace{1cm} (3)

where $C$ is a constant value heuristically determined during the experiments.

### III. EXPERIMENTAL RESULTS AND DISCUSSION

For our experiments, we have used data extracted from hyper-spectral images captured by AVIRIS sensor [12]. This database contains 4757 training samples and 4588 test samples. Each sample is described by using values relative to 200 of the 220 spectral channels used by the AVIRIS sensor: data relative to the remaining 20 spectral channels have been discarded because affected by noise due to atmospheric problems. The classification problem consists of nine land cover classes. The training set has been used to select the feature subsets and to train both the first and the second classification stages. The test set has been used to assess the performance of the whole system.

As discussed in Section 2, we subdivided a $N$ classes problem in $N$ two class problems. For this reason, a separate set of experiments have been performed for each of the $N$ classes to be recognized. In particular, for each class, 20 runs of the GA have been executed 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 such runs, the best subset obtained has been used for training the SVM that is associated to that class.

To set the values of the GA basic evolutionary parameters, some preliminary trials have been carried out obtaining the results shown in Table I. As regards the constant $C$ in eq. 3, the best performance has been obtained for a value equal to 1.3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>population size</td>
<td>$P$</td>
<td>500</td>
</tr>
<tr>
<td>tournament size</td>
<td>$t$</td>
<td>5</td>
</tr>
<tr>
<td>elitism size</td>
<td>$e$</td>
<td>1</td>
</tr>
<tr>
<td>initial probability of symbol 1</td>
<td>$p_1$</td>
<td>0.1</td>
</tr>
<tr>
<td>crossover probability</td>
<td>$p_c$</td>
<td>0.6</td>
</tr>
<tr>
<td>mutation probability</td>
<td>$p_m$</td>
<td>$1/N_F$</td>
</tr>
<tr>
<td>generation number</td>
<td>$N_g$</td>
<td>1500</td>
</tr>
</tbody>
</table>

Table I
VALUES OF THE GA EVOLUTIONARY PARAMETERS

As for the SVM’s, we used the standard algorithm of regularized Support Vector Classification (C-SVC) with a Radial Basis Function kernel [13]. The results achieved by our system have been compared with those obtained by other standard feature selection or feature extraction methods [9], [10], [11], [13].

Table II shows recognition rates ($R$) and number of selected features ($N_F$) achieved by our system and by the comparison methods. Row 1 shows the results achieved by our system. Row 2 reports the results exhibited by a multiclass SVM using the whole feature set. Rows from 3 to 9 show the results of seven feature selection methods: the first six are standard techniques implemented in the WEKA Machine Learning Project [14], while the seventh, denoted as Bazi, reports the results presented in [11]. The last row shows the results obtained by using the Kernel Principal Component Analysis method [13], which allows to compute
a set of orthogonal directions capturing most of the variance of the data distribution in the feature space. Note that the number of features $NF$ in the Table has to be intended as the average number per class for both our method and the Bazi method.

For each WEKA method, the acronyms of both the evaluation method and the search strategy used, are reported. In the following, a brief description of such techniques is given. The Feature–Class Correlation method [9] (FCC) evaluates a feature subset by measuring the correlation among its features and the classes: it prefers the subset highly correlated with the classes but having low correlation among features. The Consistency Criterion (CC) [10] evaluates the worth of the feature subset by using a consistency index measuring how well samples belonging to different classes are separated. The Information Gain (IG) evaluates every feature by measuring its information content for each class.

As regards search strategy, four different algorithms have been chosen. The Greedy Forward algorithm (GF) starts with no features and iteratively build the feature subset by adding to the current one the feature that more increases the evaluation function. The procedure stops when every new addition does not improve the current subset. The Best First algorithm (BF) searches the solution by applying the beam search heuristics. The Rank Search algorithm (RS) ranks features by means of an evaluation function: subsets of increasing size are considered and the best features set is given as result. Also the Ranker algorithm (RK) ranks single features according to a chosen evaluation function. The number of selected features depends on a threshold heuristically determined.

Table II shows that the results obtained using our method are better than those obtained by the ones used for comparison as regards the recognition rate achieved. Moreover, the reduction of the number of features used is in the average higher than 50%. More particularly, it is interesting to note that the percentage of samples of the test set classified in the first stage is 91.57% with a not recoverable error rate of 2.77%. The remaining 8.43% of samples are passed to the second stage that correctly classifies more than 65% of them, thus the 5.49% of the whole test set. This leads to the shown recognition rate of 94.29% and demonstrates the advantage of using a two stage architecture.

**REFERENCES**


