A GA-based feature selection approach with an application to handwritten character recognition

C. De Stefano, F. Fontanella*, C. Marrocco, A. Scotto di Freca

*Dipartimento di Ingegneria Elettrica e dell'Informazione (DIEI),
Università di Cassino e del Lazio Meridionale, Italy*

Abstract

In the framework of handwriting recognition, we present a novel GA–based feature selection algorithm in which feature subsets are evaluated by means of a specifically devised separability index. This index measures statistical properties of the feature subset and does not depends on any specific classification scheme. The proposed index represents an extension of the Fisher Linear Discriminant method and uses covariance matrices for estimating how class probability distributions are spread out in the considered $N-$dimensional feature space. A key property of our approach is that it does not require any a priori knowledge about the number of features to be used in the feature subset. Experiments have been performed by using three standard databases of handwritten digits and a standard database of handwritten letters, while the solutions found have been tested with different classification methods. The results have been compared with those obtained by using the whole feature set and with those obtained by using standard feature selection algorithms. The comparison outcomes confirmed the effectiveness of our approach.

1. Introduction

It is generally agreed that one of the main factors influencing performance in handwriting recognition is the selection of an appropriate set of features for representing input samples [1, 2, 3, 4, 5, 6, 7]. This has led to the development of a large variety of feature sets, which are becoming increasingly larger in terms of number of attributes. The aim is to address the problem of diversity in style, size, and shape, which can be found in handwriting produced by different writers [8]. The effect is that the efficiency of learning algorithms may degrade, especially in presence of irrelevant or redundant features.

To overcome this problem and maximize classification performance, many techniques have been proposed for reducing the dimensionality of the feature space in which data have to be processed. These techniques, generally denoted as feature *reduction* [9], may be divided in two main categories, called feature *extraction* and feature *selection*. Feature extraction–based methodologies transform the original feature space into a smaller one. The transformation can be any linear or nonlinear combination of the original features [10]. Feature selection–based approaches, instead, produce as output a feature subset from the original one, without any kind of transformation [11]. Such subset is supposed to include the best features

according to a certain criterion. The role of such criterion consists in identifying the subset providing the most discriminative power.

The choice of a good feature subset is a crucial step in any classification process for several reasons:

– 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.

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

– Irrelevant and noisy features improperly chosen may make the learning process ineffective.

– The computational cost of the classification process 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 whole feature set Y is high, the problem of finding the optimal feature subset 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 in the literature for finding near–optimal solutions: Greedy selection [12], branch and

---
*Corresponding author

Email addresses: `destefano@unicas.it` (C. De Stefano),
`fontanella@unicas.it` (F. Fontanella*),
`cristina.marrocco@unicas.it` (C. Marrocco), `a.scotto@unicas.it`
(A. Scotto di Freca)

bound (B&B) [13], floating search [14]. These algorithms use greedy stepwise strategies that incrementally generate feature subsets by adding the feature that produces the highest increment of the evaluation function. Since these algorithms do not take into account complex interactions among several features, in most of the cases they lead to sub–optimal solutions. An alternative way to cope with the search problem is that of using Genetic Algorithms (GAs), which have demonstrated to be an effective search tools for finding near–optimal solutions in complex and non–linear search spaces [15]. For this reason, GA-based search strategies have been widely used to solve feature selection problems [16, 17, 18, 19, 20, 21]. In [16] and [17], a Nearest Neighbor (NN) classifier has been used for evaluating feature subsets, while in [21] this goal is achieved by using a Neural Network and by combining the classification results with some costs associated to the features. In particular, in [17], a hybrid mechanism is proposed for finding better solutions in the neighborhood of each solution found by the GA. Moreover, comparative studies have demonstrated the superiority of GAs in feature selection problems involving large numbers of features [16]. In all the mentioned approaches, however, the cardinality of the subset to be found must be a priori fixed. Finally, in [22] a GA based method is presented, which uses a combination of Adaboost classifiers for evaluating the fitness of each individual in the evolving population. The analysis of the experiments shows that their feature selection method obtains results that are comparable with those obtained by considering all the features available. Thus, there is no performance increment, but only a reduction of the computational complexity.

Feature selection methods can be subdivided into two wide classes, *filter* and *wrapper*. Given a feature subset to be evaluated, filter functions take into account its statistical properties, while the wrapper ones use the performance achieved by a certain classifier trained on that subset. Filters methods generally involve a non-iterative computation on the dataset, which can be much faster than a classifier training session. In fact, implementing a classifier for evaluating the recognition rate attainable on a given subset, would require a costly training phase of such a classifier on a training set and a sample by sample labeling procedure on a test set. Moreover, filters methods evaluate intrinsic properties of the data, rather than the interactions of such data with a particular classifier: thus, the provided solutions should be more general, allowing good results to be obtained with a larger family of classifiers. The main drawback of filter methods is the fact that the objective function is generally monotonic, and this imply that the algorithm tends to select the full feature set as the optimal solution. This forces the user to select an arbitrary cut-off on the number of features to be selected. Wrapper methods generally achieve better recognition rates than filters ones since they are tuned tacking into account the specific interactions between the considered classifier and the dataset. These methods, however, are computationally

expensive since they require that the learning procedure must be repeated for each feature subset, and the obtained results will be specific for the considered classifier.

Most of the approaches proposed in the context of handwriting recognition use wrapper methods [3, 4, 5, 6]. Their main purpose is to reduce the number of features, keeping the recognition rate unchanged, or at most slightly worse.

Moving from these considerations, we propose a GA–based feature selection algorithm in which feature subsets are evaluated by means of a novel separability index. Our algorithm belongs to the filter method category and has been devised by extending the Fisher Linear Discriminant [23] method. Such method uses covariance matrices for estimating how the probability distributions of patterns are spread out in the considered $N-$dimensional space. Given a feature subset X, the Fisher's approach estimates the separability of the classes in X by taking into account 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. Moreover, in order to compare subsets with different number of attributes, and to balance the effects of the monotonic trend of the separability index, we have added to the objective function a further term, suitable weighted, that takes into account the cardinality of the subspace to be evaluated. The proposed approach, thanks to the devised separability index, presents two main advantages: (i) it does not require that the dimensionality of the searched subspace (i.e. the actual number of features to be used) is a priori fixed; (ii) its performances are independent from the classification scheme.

The effectiveness of the proposed approach has been tested by using three standard databases of handwritten digits and a standard database of handwritten letters (uppercase and lowercase, totaling 52 classes), while the solutions found have been used to train different classifiers. The results have been compared with those obtained by using the whole feature set and with those obtained by using standard feature selection algorithms. The comparison outcomes confirmed the effectiveness of our approach.

The remainder of the paper is organized as follows: Section 2 discusses the feature selection problem, while Section 3 illustrates the proposed GA–based feature selection method. Section 4 describes the fitness function based on a specifically devised separability index used for subset evaluation. In Section 5 the experimental results are detailed, while some conclusions are eventually left to Section 6.

2. The Feature Selection Problem

The goal of Feature Selection (FS) is that of reducing the number of features to be considered in the classification stage. This task is performed by removing irrelevant or noisy features from the whole set of the available ones. Feature selection is accomplished by reducing as much as

possible the information loss due to the feature set reduction: thus, at list in principle, the selection process should not reduce classification performance. The feature selection process consists of three basic steps (see Fig. 1): a search procedure, a subset evaluation and a stopping criterion. A typical search procedure uses a *search strategy* for finding the optimal solution, according to a given subset evaluation criterion previously chosen. The search procedure is repeated until a stopping criterion is satisfied.

Considering a generic application in which a set of samples (say $\mathcal{Z}$) must be classified, and assuming that the samples are represented by means of a set Y of N features, the feature selection problem can be formulated as follows: find the subset $X \subseteq Y$ of M features which optimizes an objective function J. Given a generic subset $X \subseteq Y$, $J(X)$ measures how well the patterns in $\mathcal{Z}$ are discriminated by using the features subset X.

Example of statistical measures used by filter methods are the following: Distance [24], correlation [11, 25], information [26] and consistency [27]. Distance-based criteria takes into account the geometrical characteristics of the class distributions in order to evaluate how well different classes are separated in the subset to be evaluated. Correlation measures use measures able to estimate the dependency between couple of variables. Such estimation can be used to find the correlation between a feature and a class. If the correlation between the feature x_1 and a given class c_i is higher than that between the feature x_2 and c_i, then the feature x_1 is preferred to x_2 for describing the class c_i. A slight variation of this criterion determines the dependence of a feature on the other ones; this value can be used to assess the redundancy degree of the features. Information measures, instead, evaluate the information gain from a given feature. The information gain of a feature x is defined as the difference between the a-priori uncertainty and the expected a-posteriori uncertainty of the class label given x; the entropy measure can be used to estimate these uncertainties. Finally, the consistency measure of a feature subset, is determined counting the number of samples with the same feature values, but belonging to different classes.

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. As
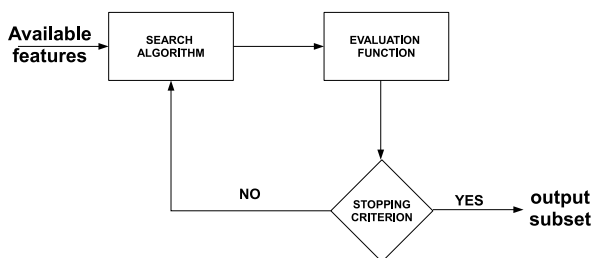
mentioned in the Introduction the size of this search space is exponential (2^N). As a consequence, the exhaustive search for the optimal solution becomes infeasible when a large number of features ($N > 50$) is involved. Search strategies like branch and bound [28] have been proposed to strongly reduce 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 flavors: forward selection and backward elimination. Forward selection strategies generate near–optimal feature subsets by a stepwise procedure which starts with an empty set. At each step the feature, among those not yet selected, that most increases the evaluation function J is added to the so far built subset; 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 [29] seems to be appropriate for finding near–optimal solutions which take into account multiple interactions among several features.

3. The proposed method

In the framework of filter approaches to the feature selection problem, we propose a new method based on the use of Genetic Algorithms (GAs). These algorithms belong to the *evolutionary computation* paradigm [15], which has shown to be very effective for solving optimization problems whose search spaces are high dimensional, discontinuous and complex. Mimicking the phenomena of natural evolution of species, GAs allows us to evolve a population of possible solutions, where each of them (denoted as an *individual* in the GA jargon) is represented as a binary string. Crossover and mutation operators are used to modify such strings in order to explore the search space, i.e. the set of all possible solutions. The method presented here has been implemented by using a generational GA, in which individuals are binary vectors each encoding a feature subset. More specifically, given a feature set Y having cardinality N, a subset X of Y ($X \subseteq Y$) is represented by an individual I having N elements whose i-th element is set to 1 if the i-th feature is included in X, 0 otherwise (see Fig. 2).

Besides the simplicity in the solution encoding, GAs are well suited for this class of problems because the search in this exponential space is very hard since interactions

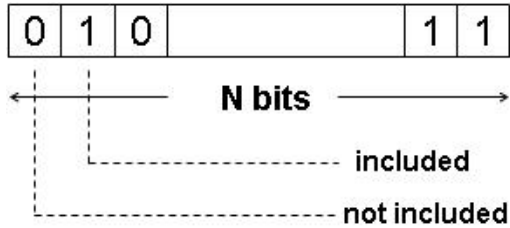Figure 1: The feature selection process

Figure 2: An example of feature subset encoding by means of a bit string.

among features can be highly complex and strongly non-linear. The algorithm starts by randomly generating a population of P individuals, whose values are set to 1 according a given probability (called *one_prob*). Such probability is usually set to low values (≈ 0.1) in order to force the early stage of the evolutionary search toward solutions having a small number of features. Then, the fitness of the generated individuals is evaluated by means of a suitably defined fitness function. This function takes into account how well the samples belonging to different classes are separated in the feature subset encoded by an individual, favoring at the same time the discovery of 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 current population in order to implement an elitist strategy. Then $(P - e)/2$ couples of individuals are selected using the tournament method, which allows both loss of diversity and selection intensity to be controlled [30]. The one point crossover operator is then applied to each of the selected couples, according to a given probabilit fator p_c. Afterwards, the mutation operator is applied. Then, the fitness funct o is computed according to the method illustrated in the next Section. Finally hese in ividuals are added to the new population. The process jus described is re eated for N_g generations. Note that it would be possible that some of the individuals generated according to the above process encode feature subset for which it is not possible to compute the fitness function. These solutions are simply discarded by the GA and a new offspring are generated by selecting other new parents in the current population.

4. Fitness Function

The proposed fitness function takes into account two terms: in the first one, a function J, called separability index, measures the separability of the patterns belonging to different classes in the feature subset encoded by an individual. The second term takes into account the cardinality of the subset so as to favor solutions containing a smaller number of features.

The separability index J has been derived from the Multiple Discriminant Analysis (MDA) approach. MDA is an extension to C–class problems ($C > 2$) of the Fisher's linear discriminant [23], which has been defined for finding the best linear combination of features in case of two class problems.

In our case, assuming that each feature can be modeled as a random variable, the separability index J can be computed by using the covariance matrix of the whole feature set. Before providing the definition of J, let us recall some general properties of covariance matrix, Fisher's Linear Discriminant and Multiple Discriminant Analysis.

Covariance matrix is the generalization of variance of a scalar variable to multiple dimensions. While variance measures the dispersion of the values of a random variable around its mean value, the covariance matrix of n variables measures how the joint probability distribution of the variables is spread out in the considered $n-$dimensional space around the mean vector. In particular, given n random variables $\{x_1, x_2, \ldots, x_n\}$, each sampled by considering m values (stored in a $m \times n$ matrix D), the covariance matrix $\mathbf{\Sigma}$ is a $n \times n$ matrix in which the element at row i and column j represents the covariance between the variables x_i and x_j:

$$\mathbf{\Sigma}[i,j] = Cov(x_i, x_j)$$

The covariance matrix is symmetric, in fact the generic element $Cov(x_i, x_j)$ is defined as follows:

$$Cov(x_i, x_j) = \frac{1}{m} \sum_{l=1}^{m} (D[l,i] - \mu_i)(D[l,j] - \mu_j) \quad (1)$$

where μ_i and μ_j are the mean values of the elements in the $i-$th and $j-$th column of D, respectively [1].

If samples belonging to different classes are represented in as points in a $N-$dimensional space, the $k-$th class can be described by using its covariance matrix $\mathbf{\Sigma}_k$, which is obtained by considering only the samples belonging to the class k. Such a matrix, in fact, reports information about the variability of k–th class samples around their mean vector $\vec{\mu}_k$.

Denoting with c_0 and c_1 the classes to be discriminated and with ω the label associated to a sample $\vec{x}$, the Fisher's Linear Discriminant (FLD) [23] approach assumes that the conditional probability density functions $p(\vec{x}|\omega = c_0)$ and $p(\vec{x}|\omega = c_1)$ are normally distributed with mean and covariance parameters $(\vec{\mu}_0, \mathbf{\Sigma}_0)$ and $(\vec{\mu}_1, \mathbf{\Sigma}_1)$, respectively. It is also assumed that both classes have the same a priori probability $(P(c_0) = P(c_1))$. The goal of FLD approach is to find the vector $\vec{w}^*$ representing the locus of the points where the samples are projected, which best separates the two classes. To this aim, the separation between the above distributions (denoted as class separability S in the following) has been computed as the ratio of the variance between the classes to the variance within the classes:

---

[1]Note that the element $\mathbf{\Sigma}[i,i]$ represents the variance of the variable x_i

$$S(\vec{w}) = \frac{\sigma_{between}^2}{\sigma_{within}^2} = \frac{(\vec{w}^T \vec{\mu}_1 - \vec{w}^T \vec{\mu}_0)^2}{\vec{w}^T \mathbf{\Sigma}_0 \vec{w} + \vec{w}^T \mathbf{\Sigma}_1 \vec{w}} = \qquad (2)$$

$$\frac{\vec{w}^T (\vec{\mu}_1 - \vec{\mu}_0)(\vec{\mu}_1 - \vec{\mu}_0)^T \vec{w}}{\vec{w}^T (\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1) \vec{w}} \qquad (3)$$

According to the FDL theory, it can be shown that the best separation occurs when:

$$\vec{w} = \vec{w}^* = (\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^{-1}(\vec{\mu_1} - \vec{\mu_0}) \qquad (4)$$

Note that the vector $\vec{w}^*$ is normal to the discriminant hyperplane and can be computed through eq. (4) only if the matrix resulting from the sum $(\Sigma_0 + \Sigma_1)$ is nonsingular, i.e. invertible.

When there are C classes to be discriminated, the analysis just described can be extended to find the $(C-1)$-dimensional subspace, which maximizes the class separability S. Note that if N is the number of available features, such subspace can be represented by means of a $N \times (C-1)$ matrix $\mathbf{W}$. Such a matrix is composed of $(C-1)$ N-dimensional vectors, representing the $(C-1)$ projections on the transformed space. This approach is usually denoted in the literature as *Multiple Discriminant Analysis*(MDA). In this case, the variances $\sigma_{between}^2$ and σ_{within}^2 can be expressed in terms of two matrices denoted as *within-class scatter matrix* $\mathbf{\Sigma}_W$ and *between-class scatter matrix* $\mathbf{\Sigma}_B$:

$$\mathbf{\Sigma}_W = \sum_{k=1}^{C} P(c_k)\mathbf{\Sigma}_k$$

$$\mathbf{\Sigma}_B = \sum_{k=1}^{C} P(c_k)(\vec{\mu}_k - \vec{\mu}_0)(\vec{\mu}_k - \vec{\mu}_0)^T$$

where $P(c_k)$ denotes the a priori probability of the k–th class, $\mathbf{\Sigma}_k$ and $\vec{\mu}_k$ are the covariance matrix and the mean vector of k–th class, respectively, and $\vec{\mu}_0$ denotes the overall mean:

$$\vec{\mu}_0 = \sum_{k=1}^{C} P(c_k)\vec{\mu}_k$$

Note that the *within-class scatter matrix* $\mathbf{\Sigma}_W$ measures the average spread of the classes about their mean vectors, while the *between-class scatter matrix* $\mathbf{\Sigma}_B$ measures the distances between each class mean vector and the overall mean.

According to the MDA theory, the class separability S can be measured as follows:

$$S(W) = \frac{|\mathbf{W}^T \mathbf{\Sigma}_B \mathbf{W}|}{|\mathbf{W}^T \mathbf{\Sigma}_W \mathbf{W}|} \qquad (5)$$

where $|\cdot|$ indicates the determinant.

In this case the best separation is obtained by selecting the projections that give the best separation among classes. Such projections individuate the subspace represented by the $N \times (C-1)$ matrix W^*, written as:

$$W^* = \mathbf{\Sigma}_W^{-1} \mathbf{\Sigma}_B$$

The matrix W^* allows a transformation of the original space in the projected space, thus a dimensionality reduction from a N to $C-1$. Such a mapping is a good way to handle the curse of dimensionality but, as explained in [23], it can not possibly allow to obtain the minimum achievable error rate, especially in case of very large data set. Moreover the computational complexity of finding the optimal W^* is dominated by the calculation of the inverse of the *within-class scatter matrix* $(\mathbf{\Sigma}_W^{-1})$. Note that the matrix $\mathbf{\Sigma}_W$ can be obtained by computing the covariances $\mathbf{\Sigma}_k$ $(k = 1, .., C)$ through equation (1), but the computation of $\mathbf{\Sigma}_W^{-1}$ requires that $|\mathbf{\Sigma}_W| \neq 0$ and this condition is not always verified. Finally, it is worth noticing that the Multiple linear discriminant method does not directly represent a classification method, but rather it provides the subspace in which the classes are best separated: in this subspace a classification scheme must be defined in order to classify the patterns.

Moving from the above consideration, the basic idea of our approach is that of managing the curse of dimensionality by finding the optimal mapping from the original N-dimensional space to a M-dimensional one obtained by considering only a subset of M features among the whole set of N available ones. In other words, we have reformulated the feature extraction problem as a feature selection one, in which the effectiveness of the selected feature subspace is measured by using the class separability S defined in eq. (5). Following this approach, the matrix $\mathbf{W}$ is composed of M N-dimensional vectors, representing the axes of the transformed M-dimensional subspace. Such axes constitute the orthonormal basis of this M-dimensional subspace, and each of them coincides with one of the axes of the original N-dimensional feature space. Note that we do not assume that the selected features are independent, even if the considered subspaces are orthogonal: in fact, the orthogonality of the subspaces is simply a direct consequence of the basic assumption of our approach, which try to solve a feature selection problem and not a feature extraction one.

Let us now provide a formal definition of the separability index J: given an individual I, representing a feature subset X having dimensionality M, the separability index $J(I)$ is computed as follows:

$$J(I) = tr(\frac{\mathbf{W}^T \mathbf{\Sigma}_B \mathbf{W}}{\mathbf{W}^T \mathbf{\Sigma}_W \mathbf{W}}) \qquad (6)$$

where $\mathbf{W}$ is the transformation matrix[2] from the original N-dimensional space to the M-dimensional subspace corresponding to the subset X, while the symbol $tr(.)$ is the

---

[2]For example, if $N = 4$ is the cardinality of the whole feature space and the individual $I = (0, 1, 0, 1)$ encodes the subspace X

trace operator. High values of the separability index $J(I)$ indicate that, in the subspace represented by the individual I, the centroids of the classes are well separated and, at the same time, the patterns are not too much spread out around their mean values. Without loosing generality, we have modified the eq. (5) using the trace operator: in fact, even if there is no mathematical equivalence between eq. (5) and eq. (6), it has been demonstrated in [10] that they provide the same set of optimal features. Therefore, we have chosen eq. (6), which is computationally more effective.

Eventually, let us now define the fitness function used by our GA-based feature selection method: given an individual I, its fitness value $F(I)$ is computed by applying the formula:

$$F(I) = \frac{J(I)}{N} + K\frac{N - N_I}{N} \qquad (7)$$

where N is the total number of features available, N_I is the cardinality of the subset represented by I (i.e. the number of bits equal to 1 in its chromosome) and K is a constant value used to weight the second term.

Note that in the first term, the separability index J has been divided by N so as to assure that both terms of eq. (7) range from 0 to 1. The second term has been added since the first one exhibits a monotonic trend with the number of features. In fact, starting from a certain set of features, if we add any further feature, the separability index J do not decrease its value unless the new feature set makes the matrix $\mathbf{\Sigma}_W$ not invertible. In this case, obviously, the separability index cannot be computed and the new solution must be discarded. Thus, using as fitness only the first term of eq. (7), the GA may produce solutions in which there are features not contributing to increase separability index J. These solutions are penalized by the presence of the second term. Finally, as regards the constant K, its role is to weight the second term in such a way that individuals having more features are favored only if they exhibits higher values for the separability index.

5. Experimental Results

In order to ascertain the effectiveness of the proposed approach, four real world datasets involving handwritten characters have been taken into account. Since our approach is stochastic, as well as all the EC–based algorithms, 20 runs have been performed for each experiment

---

having cardinality 2 (features 2 and 4) the matrix $\mathbf{W}$ is the folowing:

$$\mathbf{W} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Note that if $\mathbf{\Sigma}$ is the 4×4 covariance matrix, the product $\mathbf{W}^T\mathbf{\Sigma}\mathbf{W}$ gives as result the 2×2 matrix $\mathbf{\Sigma}'$, which represents the projection of $\mathbf{\Sigma}$ on the subspace encoded by the individual I.

carried out. The ability of our system in finding good subsets has been evaluated by measuring the performance obtained with three classification algorithms: Support Vector Machine (SVM) [31], Multiple Layer Perceptron (MLP) and k–Nearest Neighbor (k–NN) [23]. To this purpose, at the end of each run, the best feature subset found has been used for training the considered classifiers and for evaluating their performance.

As regards the SVM, we used the implementation provided by the LIBSVM public domain software [32]. In the experiments described in the following we used three different kernels: Radial Basis Function (RBF), Polynomial and Sigmoidal. For the k–NN and MLP classification algorithms, we used the implementation provided by the WEKA tool [33]. As regards the evolutionary parameters, shown in Table 1, they have been heuristically found performing a set of preliminary trials; this set of parameters has been used for all the experiments reported below.

The main objectives of the experiments are the following:

- To investigate the influence of the cost factor K on the performance of the system.

- To study if the selected feature subsets are independent of the choice of the used classifier. To this aim, the subsets found have been used to train the considered classification algorithms with different sets of parameters.

- To perform a comparison of the obtained results with those of other feature selection algorithms reported in the literature.

In the following, the datasets used and the experiments performed are detailed.

5.1. The Data Sets

We have used in the experiments three standard databases of handwritten digits and a standard database of handwritten letters (uppercase and lowercase, totaling 52 classes). Two of them (OPTODIGIT and MFEAT) are publicly available from the UCI machine learning repository [34], the third one (MNIST) has been made available by the New York University [35], while the last one (NIST-SD19) is provided by the National Institute of Standard Technologies [36].

Table 1: The evolutionary parameters

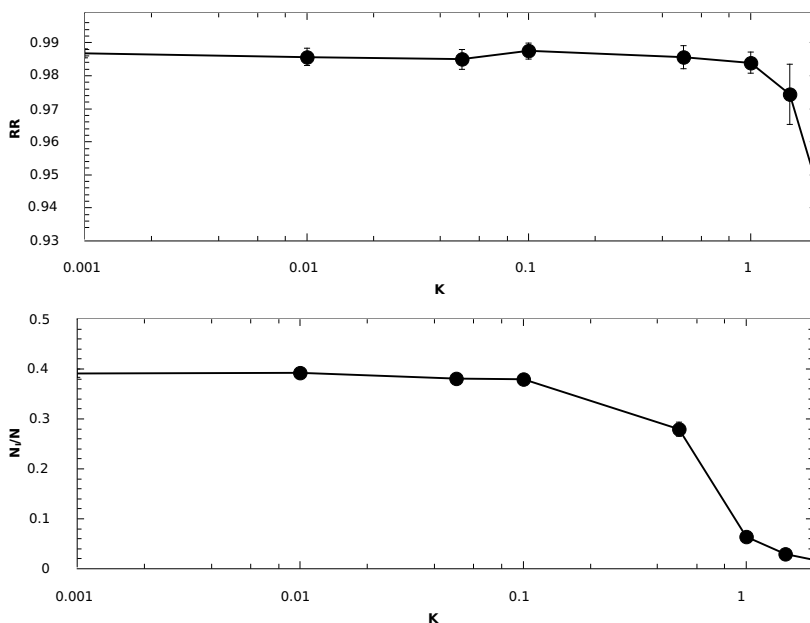| Parameter	symbol	value
population size	P	100
tournament size	t	5
elitism size	e	1
crossover probability	p_c	0.6
mutation probability	p_m	$1/N$
generation number	N_g	1000

6

Figure 3: Recognition rate RR (top) and number of features N_I (bottom) as a function of the constant K for MFEAT dataset.

The MFEAT dataset (Multiple Features Data Set) contains 2000 instances of handwritten digits, 200 for each digit, extracted from a collection of Dutch utility maps. Data are described by using six different sets of features, totaling 649 features. Each set of features has been used to describe all the handwritten digits, and arranged in separate datasets. This implies that we have 6 datasets (DS1, ..., DS6), each containing 2000 samples. For each dataset, the type of features and their number are the following:

DS1: 76 Fourier coefficients of the character shapes;
DS2: 47 Zernike moments;
DS3: 6 morphological features.
DS4: 64 Karhunen-Love coefficients;
DS5: 240 pixel averages in 2 x 3 windows;
DS6: 216 profile correlations.

More details about the feature sets can be found in [37]. Starting from the provided datasets we generated a further dataset (DS) obtained by merging all the descriptions included in the previous ones, in such a way to describe each sample by the whole set of 649 available features. From the generated dataset DS, 70 samples per class have randomly extracted to build a training set (TR). The remaining data have been used to build a test set (TS) including 130 samples per class. Summarizing, TR contains 700 samples, while TS contains 1300 samples.

The second considered dataset is the Optical Recognition of Handwritten Digits Data Set (OPTODIGIT). It contains 5620 samples equally distributed among the ten classes. Each sample is described by 64 features. Such data have been obtained by preprinted forms, extracting normalized 32x32 bitmaps of handwritten digits. Each bitmap is divided into non-overlapping blocks of 4x4 and the number of black pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range $[0, 16]$. As a consequence, a character is represented by a feature vector of 64 elements where each element contains a value of a 8x8 matrix. In order to build a training set, 3820 samples have been randomly picked up from the original dataset, while the remaining ones (1800) have been used as test set.

The third dataset taken into account is MNIST. Such dataset was constructed from NIST's Special Database 3 and Special Database 1 which contain binary images of handwritten digits [35]. The original black and white images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field. Finally, the MNIST training set is composed of $60,000$ samples, while the test set contains $10,000$ samples. The total number of features used to describe samples is 780, even if some of them assume values different from zero in less than 1% of samples. For this reason, we discarded such features considering in our experiment only 485 features.

Finally, the NIST-SD19 database (NIST in the following), contains binary alphanumeric characters. In particular, we have considered handwritten letters, uppercase and lowercase, corresponding to 52 classes. The handwriting sample form *hsf4*, containing 23941 characters, has been used as training set, while the handwriting sample
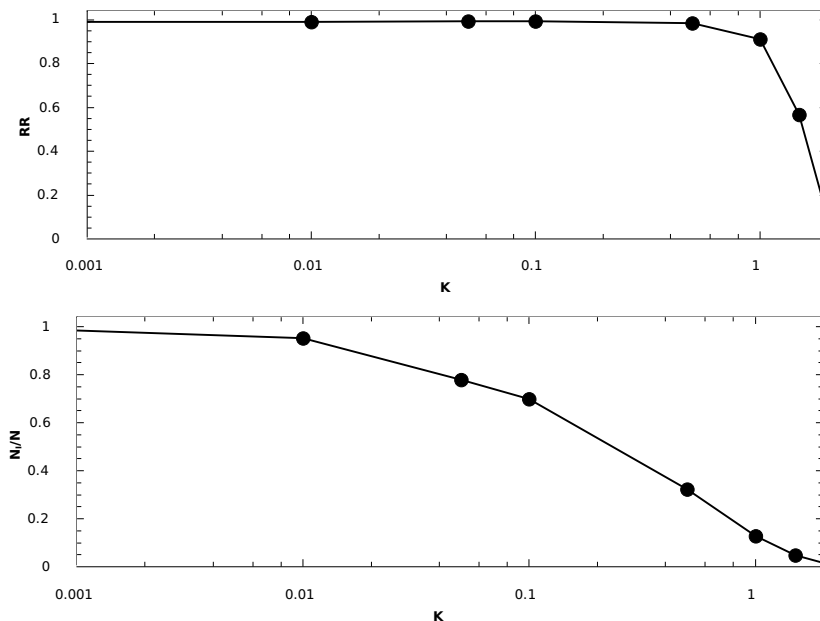
Figure 4: Recognition rate RR (top) and number of features N_I (bottom) as a function of the constant K for OPTODIGIT dataset.

form *hsf7*, containing 23670 characters, has been used as test set. *hsf4* has 11941 uppercase characters and 12000 lowercase ones, while *hsf7* has 12092 uppercase characters and 11578 lowercase ones. In each form, characters are segmented and stored in 128x128 pixel images, each associated to one of the 52 classes to be discriminated. Samples are represented by using the features proposed in [38]. Each character is described by a feature vector containing the measures associated to different parts of the image. More specifically, the image is divided in six parts and, for each part, 22 features are computed, totaling 132 features.

5.2. The Constant K

Several experiments have been performed for analyzing how the constant K affects the behavior of our feature selection method, in terms of both number of features and classification performance. Such a performance refers to the use of RBF SVM classifiers. As expected, the higher K the lower the number of selected features and the obtainable classification results. For the sake of clarity, we have not reported in the following plots the values of K for which the performance is too much degraded.

In the figures 3, 4, 5 and 6 we have reported both the recognition rate (RR) and the number of features (N_I/N) for the datasets MFEAT, OPTODIGIT, MNIST and NIST, respectively. For each value of K we have reported the average result over the 20 performed runs. Note that we have shown in the figures only the standard deviations, which do not assume negligible values.

As regards the MFEAT dataset, the figures 3 shows that for a large interval of K values, the performance is almost constant obtaining the highest value for $K = 0.01$. This value has been chosen hereafter in the experiments for MFEAT dataset, even if values up to 1.0 may be used slightly reducing classification performance, but strongly reducing also the dimension of the feature space. Similar considerations hold for the other two datasets: in particular for OPTODIGIT dataset the optimal value chosen for K is 0.05, even if the value $K = 0.5$ results in a very small reduction of the recognition rate, but using only less than 30% of the available features. For the MNIST dataset, the optimal value chosen for K is 0.001, even if the value $K = 0.5$ corresponds to a small reduction of the recognition rate, but using only the 10% of the available features. Finally, as regards NIST, the optimal value chosen for K is 0.1, even if the value $K = 0.1$ corresponds to a small reduction of the recognition rate but using only less than 20% of the available features.

Summarizing, as shown in figures 3, 4, 5 and 6, the trend of RR as a function of K is very regular: there is an interval of values of K in which RR assumes an almost constant or slightly increasing trend and then it rapidly decreases. Thus, exploiting the regularity of RR curve, a general experimental procedure for setting the value of the parameter K, is the following: increase the values of K until RR exhibits slight variations; as soon as RR starts to rapidly decrease for a certain value of K, select the previous K value as the optimal one.

5.3. Behavior Analysis

Since our feature selection method does depend on any specific classifiers, we want to evaluate both the generality and the effectiveness of our results by using different classification algorithms. As a consequence, for each dataset,
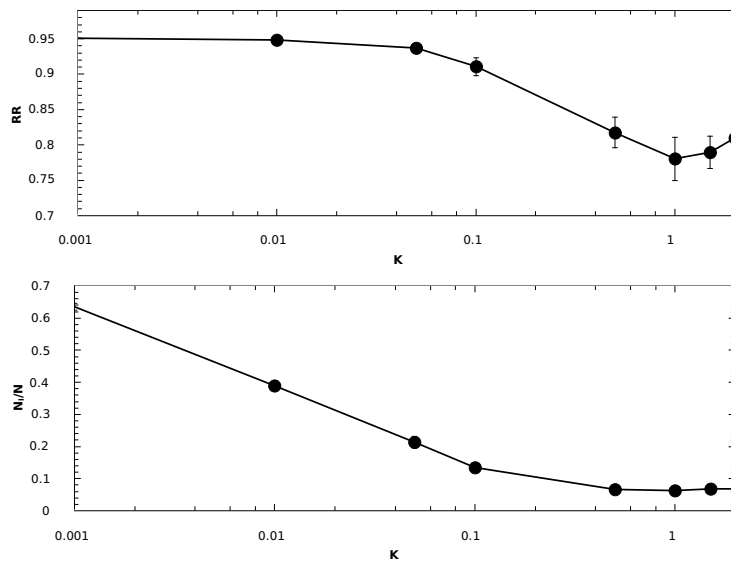
Figure 5: Recognition rate RR (top) and number of features N_I (bottom) as a function of the constant K for MNIST dataset.

the best feature subset provided by the GA has been tested by using different classifiers obtained varying the configuration parameters of the three considered classification schemes (SVM, MLP and k–NN). The obtained performances have been compared with those achieved by using the whole feature sets.

The purpose of such a comparison is twofold: on the one hand, we want to verify if the selected features allows us to improve the performance with respect to those relative to all the available features. On the other hand, we want to understand if the differences among the considered classifiers actually depend on the selected features.

For each of the three algorithm, three different configuration parameter values have been considered, totaling 9 classifiers. More specifically, for the SVM three different kernels have been used: Radial Basis Function (RBF), Polynomial (with the degree set to three), and the sigmoid. For the MLP, different classifiers have been obtained by changing the number of neurons in the hidden layer. The following values have been used: 50, 100 and 200. Finally, for the k–NN, the following k values have been considered: 1, 3, 5. The performances of the 9 classifiers have been compared by means of the 10-fold cross validation method. The results are shown in Figure 7. The acronyms SVM-R, SVM-P and SVM-S, stand for RBF SVM, Polynomial SVM and sigmoidal SVM,respectively. As regards the acronyms MLP-1, MLP-2 and MLP-3 they refer to the MLPs with 50, 100 and 200 hidden neurons, respectively. The acronyms of the k–NN are self-explanatory.

The figure shows that the feature subsets selected by our method always give better results than those obtained by using the whole set of features. Moreover, the trend of the performance obtained by using of the selected feature subsets is very similar to that of the whole feature set,

confirming the generality of the proposed feature selection method.

5.4. Comparison Findings

In order to test the effectiveness of the proposed system, our results have been compared with those obtained by four widely used feature selection techniques. Such techniques combines a search strategies and a subset evaluation criterion. Moreover, the performance obtained using the whole feature set has been also considered.

As regards the search strategies we used the Best First (BF) [39] and the Linear Forward (LF) [40] ones. The former strategy searches solutions by using a greedy hill–climbing technique. It starts with the empty set of features and adds new features according to the best first search algorithm [41]. The latter one, instead, represents an extension of the Best First strategy. Such technique reduces the number of attribute expansions in each forward selection step. It is faster than BF and it generally finds smaller subsets. Our search strategy, based on the use of a GA, exhibits an higher computational cost, since it requires a number of operations equal to the number of individuals in the population by the number of generations.

As subset evaluation criteria we have considered the following ones: Feature–Class Correlation, Consistency Criterion and three different wrapper evaluation functions.

The Feature–Class Correlation (FCC) [42] evaluates a feature subset by measuring the correlation among its features and the classes: it prefers subsets highly correlated with classes, but having low correlation among features. The Consistency Criterion (CC) [43] evaluates the worth of the feature subsets by using a consistency index measuring how well samples belonging to different classes are separated. As concerns the wrapper functions we used the

Table 2: Comparison results.

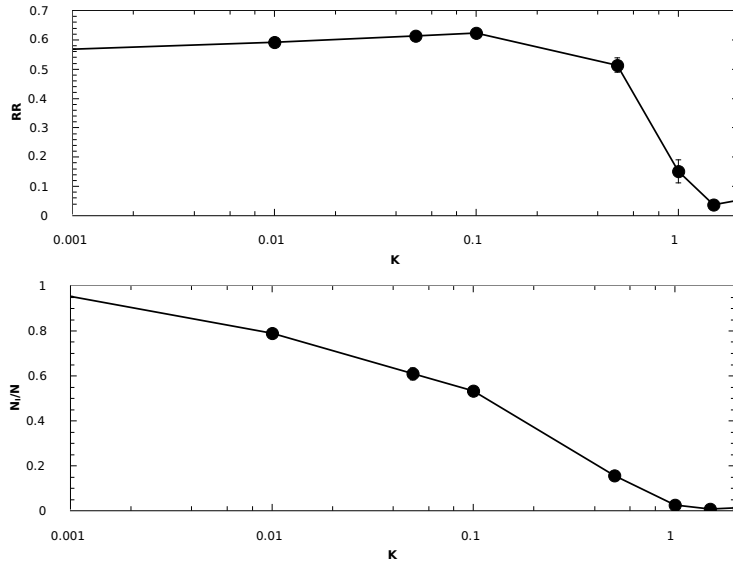|	Cl.	MFEAT		MNIST		OPTODIGIT		NIST	
		Acc.	#	Acc.	#	Acc.	#	Acc.	#
All	**SVM**	96.77		91.13		97.55		56.05	
	MLP	97.2	649	93.95	489	97	64	63.39	132
	KNN	96.19		93.47		97.5		60.25	
Our Method	**SVM**	97.69		92.94		98.61		66.16	
	MLP	98.38	233	96.15	272	98	44	71.06	61
	KNN	96.92		95.22		98.94		67.58	
FCC-BF	**SVM**	97.08		91.76		97.55		62.1	
	MLP	97.04	133	94.16	144	97.07	35	64.26	64
	KNN	96.35		93.92		97.72		63.58	
FCC-LF	**SVM**	97.15		88.36		97.55		62.21	
	MLP	97.54	106	88.23	58	97.07	35	64.57	57
	KNN	96.77		88.85		97.72		64.08	
CC-BF	**SVM**	93.46		73.65		89.32		58.55	
	MLP	91.92	6	68.09	13	83.81	9	64.39	17
	KNN	93.08		70.23		87.54		62.25	
CC-LF	**SVM**	93.92		77		90.43		58.95	
	MLP	93.08	6	72.79	20	83.92	9	64.89	14
	KNN	93.92		78.23		88.65		62.65	
MLP-BF	**SVM**	96.92		91.57		96.67		59.24	
	MLP	96.98	25	94.74	65	97.07	33	64.89	48
	KNN	95.94		94		96.56		64.11	
MLP-LF	**SVM**	97.19		91.68		96.85		58.48	
	MLP	97.28	14	94.85	43	97.1	26	65.07	41
	KNN	96.02		94.15		96.88		64.48	
SVM-BF	**SVM**	96.94		91.76		97.47		61.2	
	MLP	96.75	29	93.16	68	97.04	35	66.23	46
	KNN	95.94		93.14		96.63		65.25	
SVM-LF	**SVM**	97		91.83		97.67		61.4	
	MLP	96.92	13	93.27	47	97.44	30	66.45	40
	KNN	96.01		93.18		97.63		65.55	
KNN-BF	**SVM**	95.57		91.7		97.14		58.48	
	MLP	95.3	31	94.2	71	96.65	39	65.07	59
	KNN	95.88		94.33		97.5		64.48	
KNN-LF	**SVM**	95.57		91.85		97.6		58.68	
	MLP	97.3	17	94.27	41	96.83	32	65.7	48
	KNN	95.88		94.36		97.91		65.08	
Avg diff.		1.7		5.6		3.1		5.48	

Figure 6: Recognition rate RR (top) and number of features N_I (bottom) as a function of the constant K for NIST dataset.

same three classifiers used for the performance evaluation, namely RBF SVM, a MLP with 200 hidden neurons and the 3-NN.

The computational complexity of our evaluation function is lower than those exhibited by both FCC and CC. In fact, we have used the training set data only for computing, once and forever, the within-class and the between-class scatter matrices Σ_W and Σ_B. The other evaluation functions, FCC and CC, on the contrary, compute the feature correlation and the consistency index, respectively, for each subset to be evaluated. Finally, the computational complexity of the wrapper functions is the highest one, since they require to perform, for each subset to be evaluated, the training of the classifier and the evaluation of the corresponding recognition rate.

The results are shown in Table 2. In the table, the methods considered for the comparison are denoted by acronyms. For each acronym the first part denotes the evaluation function, while the second one the search strategy. For instance, FCC-BF denotes a comparing method using the Feature–Class Correlation as evaluation function and the Best First search strategy.

Note that the accuracy results have been obtained by using the 10–fold cross validation. For each dataset the number of selected features are also reported. The last row shows, for each dataset, the average values of the performance differences between the proposed approach and the methods taken into account for the comparison.

The data reported in the table shows that our approach achieves better results than those obtained by the comparing methods. The last row of this table, reports the mean values of such improvements, computed averaging the differences between our method and the others for each dataset. It can be seen that such improvements

varies from about 1.7% to 5.6%. As regards the number of selected features, our approach selects more features than the compared ones. This is due to the fact that we chose the value of the constant K, which maximizes the recognition rate even if larger values of K allowed a slight performance reduction, but a considerable smaller number of features.

In order to have a fair comparison with methods selecting small numbers of features, and to test the performance of our system when as less as possible features should be used, we performed a further set of experiments. More specifically, for each dataset, we chose in the fitness function a value for the parameter K, such as to obtain a number of features similar to that provided by the methods selecting the smallest number of features (CC–LF and CC-BF). The results of this comparison are shown in Table 3. Also in this case, accuracy results have been obtained by using the 10–fold cross validation.

The data in the table show that even when high values of K are used (strong feature reduction), our method performs almost always better than the other considered ones. In particular, as concerns MFEAT and OPTODIGIT datasets, our approach selects the same number of feature, or a slightly higher one, but achieves better results. The performance increments vary from 1% (MFEAT, KNN classifier) to 5% (OPTODIGIT, MLP classifier). As for MNIST dataset, our method selects a number of features higher than the that of CC–BF method (34 vs 13), but in this case the performance improvements are more relevant, ranging from 8.8% (SVM) to 16.1% (KNN). Finally, as regards NIST dataset, the results of our method are slightly worst than those of CC–LF method (approximatively from 1% to 2% less). It should be considered, however, that these are the less favorable conditions for our method: in fact, as
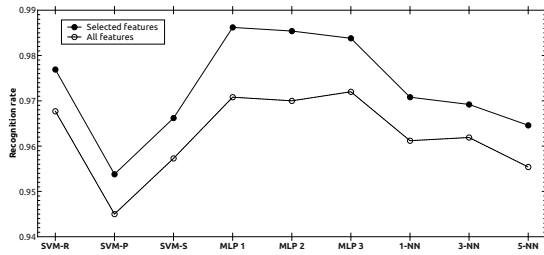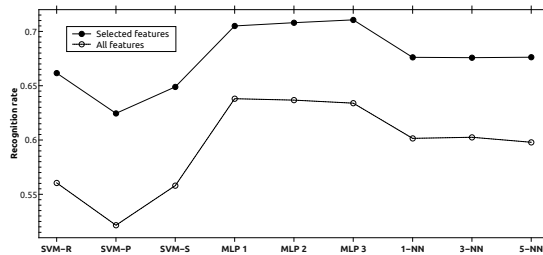
(a) MFEAT
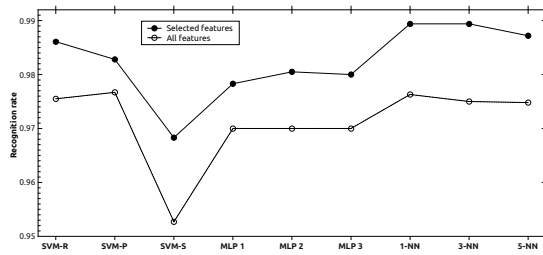
(b) OPTODIGIT
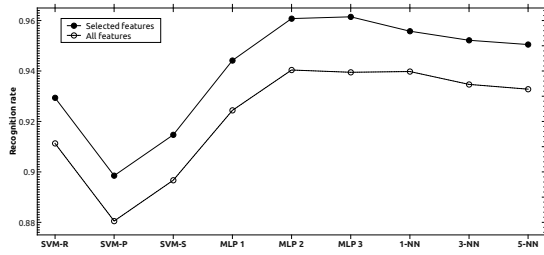
(c) MNIST

(d) NIST

Figure 7: Behavior analysis.

the value of K grows, the second term in the fitness function becomes more and more relevant with respect to the first one (see eq. 7), and the evolutionary algorithm tends to favor solutions using a small number of features even if they exhibit a low value of the separability index. This behavior has been analyzed in subsection 5.2 discussing the set of experiments for finding the optimal value of K.

6. Conclusions

In the framework of handwriting recognition problems, we have presented a feature selection method for improving classification performance. The devised approach uses a GA–based feature selection algorithm for detecting feature subsets where the samples belonging to different classes are well discriminated. The proposed method does not require that the dimensionality of the searched subspace is a priori fixed. Candidate feature subsets are evaluated by means of a novel evaluation function based on the Fisher linear discriminant. Such evaluation function uses covariance matrices for estimating how the probability distributions of patterns are spread out in the considered representation space. Moreover, in order to balance the effects of the monotonic trend of the evaluation function, a further term, suitable weighted, has been added which takes into account the cardinality of the subspace to be evaluated. The experiments have been performed by using four standard databases and the solutions found have been tested with different classification methods. The results have been compared with those obtained by using the

whole feature set and with those obtained by using standard feature selection algorithms. The comparison outcomes confirmed the effectiveness of our approach

From the experimental results we can also draw the following observations:

1. The choice of an appropriate value for the constant K is not critical. In fact, the trend of the recognition rate as a function of K is nearly constant for a wide range of K values and, starting from a certain point, it rapidly decreases. This behavior allows us to easily identify an appropriate range of values for K. More specifically, in application for which it is mandatory to maximize performance, the value of K corresponding to the highest recognition rate will be chosen. On the contrary, if we can accept a slight performance decrease, values of K corresponding to a lower number of features may be chosen.

2. The accuracy gain obtained by our method with respect to the compared methods justifies the computational time required by the feature selection algorithm proposed.

3. The separability index assumes that class distributions are normal. Even if this assumption is difficult to verify for real data, it represents a reasonable and a widely used approximation. Moreover, the results obtained on different datasets seems to prove that this assumption does not negatively affect the obtainable performance in handwriting recognition applications.

Table 3: Further comparison results.

Datasets	Cl.	Our method		Others		
		Acc.	#	Method	Acc.	#
MFEAT	**SVM**	95.0	6	**CC-LF**	93.92	6
	MLP	95.07			93.08	
	KNN	94.85			93.92	
MNIST	**SVM**	82.5	34	**CC-BF**	73.65	13
	MLP	83.49			68.09	
	KNN	86.26			70.23	
OPTODIGIT	**SVM**	92.82	12	**CC-LF**	90.43	9
	MLP	88.87			83.92	
	KNN	92.43			88.65	
NIST	**SVM**	57.88	26	**CC-LF**	58.95	14
	MLP	62.57			64.89	
	KNN	61.45			62.65	

Finally, it is worth noting that the proposed approach also shows the following interesting properties: (i) it is independent of the classification system used (ii) its computational time is independent of the training set size.

The second property derive from the fact that we have used the training set only for computing, once and forever, the covariance matrices $\mathbf{\Sigma}_k$ $(k = 1, \ldots, C)$, the mean vectors $\vec{\mu}_k$ $(k = 1, \ldots, C)$ and the overall mean vector $\vec{\mu}_0$. These information are needed for computing the within-class and the between-class scatter matrices $\mathbf{\Sigma}_W$ and $\mathbf{\Sigma}_B$. This property is another consequence of the adopted filter approach, which makes our method suitable when large datasets are taken into account.

References

[1] R. Ahlgren, H. Ryan, C. Swonger, A character recognition application of an iterative procedure for feature selection, IEEE Transactions on Computers C-20 (9) (1971) 1067–1075.

[2] D. Shi, W. Shu, H. Liu, Feature selection for handwritten chinese character recognition based on genetic algorithms, in: 1998 IEEE Int. Conference on Systems, Man, and Cybernetics, Vol. 5, 1998, pp. 4201–4206.

[3] K. Chung, J. Yoon, Performance comparison of several feature selection methods based on node pruning in handwritten character recognition, in: Proceedings of the 4th Int. Conf. on Document Analysis and Recognition (ICDAR 97), IEEE Computer Society, 1997, pp. 11–15.

[4] G. Kim, S. Kim, Feature selection using genetic algorithms for handwritten character recognition, in: Proc. of the 7th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR 2000), Amsterdam, 2000, pp. 103–112.

[5] L. S. Oliveira, R.Sabourin, F. Bortolozzi, C. Suen, A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition, Int. Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 17 (2003) 2003.

[6] C. M. Nunes, A. d. S. Britto Jr., C. A. A. Kaestner, R. Sabourin, An optimized hill climbing algorithm for feature subset selection: Evaluation on handwritten character recognition, in: Proceedings of the 9-th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR'04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 365–370.

[7] L. Cordella, C. De Stefano, F. Fontanella, C. Marrocco, A feature selection algorithm for handwritten character recognition, in: 19th International Conference on Pattern Recognition (ICPR 2008), 2008, pp. 1–4.

[8] G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real-time applications, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 366–379.

[9] I. Fodor, A survey of dimension reduction techniques, Tech. rep., Center for Applied Scientic Computing, Lawrence Livermore National Laboratory (2002).

[10] K. Fukunaga, Introduction to statistical pattern recognition (2nd ed.), Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[11] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Mach. Learn. Res. 3 (2003) 1157–1182.

[12] N. Kwak, C.-H. Choi, Input feature selection for classification problems, IEEE Trans. on Neural Network 13 (1) (2002) 143–59.

[13] P. Somol, P. Pudil, J. Kittler, Fast branch and bound algorithms for optimal feature selection, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (7) (2004) 900–912.

[14] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recogn. Lett. 15 (11) (1994) 1119–1125.

[15] D. E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley, 1989.

[16] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern recognition, Pattern Recognition 33 (1) (2000) 25–41.

[17] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Trans. Pattern Anal. Mach. Intell. 26 (11) (2004) 1424–1437.

[18] L. Cordella, C. De Stefano, F. Fontanella, C. Marrocco, A. Scotto di Freca, Combining single class features for improving performance of a two stage classifier, in: 20th International Conference on Pattern Recognition (ICPR 2010), 2010, pp. 4352–4355.

[19] C. De Stefano, F. Fontanella, C. Marrocco, G. Schirinzi, A feature selection algorithm for class discrimination improvement, in: Geoscience and Remote Sensing Symposium, 2007 (IGARSS07), 2007, pp. 425 –428.

[20] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recogn. Lett. 10 (5) (1989) 335–347.

[21] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems 13 (1998) 44–49.

[22] H. Chouaib, O. R. Terrades, S. Tabbone, F. Cloppet, N. Vincent, Feature selection combining genetic algorithm and adaboost classifiers, in: Proc. of International Conference on Pat-

tern Recognition (ICPR 2008), IEEE, 2008.

[23] P. E. Hart, R. O. Duda, D. G. Stork, Pattern Classification, John Wiley & sons, Inc., 2001.

[24] T. K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 289–300.

[25] M. A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, pp. 359–366.

[26] M. Ben-Bassat, Pattern recognition and reduction of dimensionality, in: P. Krishnaiah, L. Kanal (Eds.), Handbook of Statistics-II, North Holland, 1982, pp. 773–791.

[27] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1-2) (2003) 155–176.

[28] B. Yu, B. Yuan, A more efficient branch and bound algorithm for feature selection, Pattern Recognition 26 (6) (1993) 883–889.

[29] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, European Journal of Operational Research 171 (3) (2006) 842–858.

[30] T. Blickle, L. Thiele, A comparison of selection schemes used in evolutionary algorithms, Evol. Comput. 4 (4) (1996) 361–394.

[31] V. N. Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.

[32] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, available at http://www.csie.ntu.edu.tw/ cjlin/libsvm (2001).

[33] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, SIGKDD Explorations 11 (1) (2009) 10–18.

[34] A. Frank, A. Asuncion, UCI machine learning repository (2010). URL http://archive.ics.uci.edu/ml

[35] Y. LeCun, C. Cortes, MNIST handwritten digit database (2010). URL http://yann.lecun.com/exdb/mnist

[36] P. J. Grother, Nist special database 19 (1995). URL http://www.nist.gov/srd/nistsd19.cfm

[37] M. V. Breukelen, R. Duin, D. Tax, J. den Hartog, Combining classifiers for the recognition of handwritten digits, in: 1st international workshop on statistical techniques in pattern recognition, 1997, pp. 13–18.

[38] L. Oliveira, R. Sabourin, F. Bortolozzi, C. Suen, Automatic recognition of handwritten numerical strings: A recognition and verification strategy, IEEE Trans. Pattern Analysis and Machine Intelligence 24 (11) (2002) 1438–1454.

[39] L. Xu, P. Yan, T. Chang, Best first strategy for feature selection, in: 9th International Conference on Pattern Recognition, IEEE, 1988, pp. 706–708.

[40] M. Gütlein, E. Frank, M. Hall, A. Karwath, Large-scale attribute selection using wrappers, in: Proc IEEE Symposium on Computational Intelligence and Data Mining, IEEE, 2009, pp. 332–339.

[41] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, Reading, MA, 1984.

[42] M. A. Hall, Correlation-based feature subset selection for machine learning, Ph.D. thesis, University of Waikato, Hamilton, New Zealand (1998).

[43] H. Liu, R. Setiono, A probabilistic approach to feature selection - a filter solution, in: 13th International Conference on Machine Learning, 1996, pp. 319–327.