Pattern Recognition Letters 121 (2019) 77–86

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A ranking-based feature selection approach for handwritten character recognition

Nicole Dalia Cilia, Claudio De Stefano*, Francesco Fontanella, Alessandra Scotto di Freca

Dipartimento di Ingegneria Elettrica e dell'Informazione, University of Cassino and Southern Lazio, Via Di Biasio 43, Cassino, (FR) 03043, Italy

ARTICLE INFO

Article history:
Available online 10 April 2018

MSC:
41A05
41A10
65D05
65D17

Keywords:
Feature selection
Handwritten character recognition

ABSTRACT

Feature selection is generally considered a very important step in any pattern recognition process. Its aim is that of reducing the computational cost of the classification task, in an attempt to increase, or not to reduce, the classification performance. In the framework of handwriting recognition, the large variability of the handwriting of different writers makes the selection of appropriate feature sets even more complex and have been widely investigated. Although promising, the results achieved so far present several limitations, that include, among others, the computational complexity, the dependence on the adopted classifiers and the difficulty in evaluating the interactions among features. In this study, we tried to overcome some of the above drawbacks by adopting a feature-ranking-based technique: we considered different univariate measures to produce a feature ranking and we proposed a greedy search approach for choosing the feature subset able to maximize the classification results. In the experiments, we considered one of the most effective and widely used set of features in handwriting recognition to verify whether our approach allows us to obtain good classification results by selecting a reduced set of features. The experimental results, obtained by using standard real word databases of handwritten characters, confirmed the effectiveness of our proposal.

1. Introduction

In the framework of handwriting recognition, one of the main factors influencing the obtainable performance is the selection of an appropriate feature set for representing input samples. 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 use of a huge quantity of features, however, may produce a decay in the efficiency of learning algorithms, especially in the presence of irrelevant or redundant features. Thus, feature selection methods are used for searching the optimal subset of features from the whole set of available ones, in order to obtain best recognition results.

The procedures for searching the optimal subset of features may require a combinatorial time, and this is the reason why heuristic or greedy search algorithms are often used instead of exhaustive ones [22]. Such algorithms typically require both the definition of a search strategy for selecting feature subsets, and the definition of an evaluation function to measure the effectiveness of each selected feature subset, i.e. how well classes are separated in the corresponding feature subspace. One of the main drawbacks of these algorithms regards the limitations in evaluating the effects of the interactions among features. Moreover, the computational cost, which is determined by both the searching algorithm and the evaluation criteria, may be be still relevant since the effectiveness of each feature subset in the searching procedure should be compared with that of all the others selected.

A less expensive alternative in term of computational time is that of applying the evaluation criteria to each single feature and then comparing the effectiveness of all the features using the measures obtained. In this way, each feature can be positioned in a ranking according to the value of the above measure but, obviously, the effects of the interactions among features are not taken into account.

In this context, the aim of our work is twofold: on the one hand, we have tried to overcome some of the above-mentioned drawbacks by adopting a feature-ranking-based technique and a greedy search approach for choosing the feature subset able to provide the best classification results. On the other hand, we have considered one of the most effective and widely used set of features in handwriting recognition [8,10,13,42], i.e. the one proposed in [38], to verify whether our approach allows us to obtain good classification results by selecting a reduced set of features. Finally, considering the experiments performed on three real world datasets, we have also characterized the features that exhibit

* Corresponding author.
E-mail address: destefano@unicas.it (C. De Stefano).

https://doi.org/10.1016/j.patrec.2018.04.007
0167-8655/

higher discriminant power among the three feature groups defined in the above feature set, namely the concavity, the contour information and the character surface. We have shown that, in the large majority of cases, the character surface features were included in the best feature subset. For comparison purposes, we have also applied our strategy to another standard set of features [4], obtaining similar results. Such features have been widely used to represent handwritten digits [1,12,17]. It is worth noticing that in our experiments we have only considered databases containing characters of Latin alphabet: the representation of characters belonging to other alphabets may require, in principle, the use of different features.

As regards the feature ranking, we have considered different univariate measures, each producing a different ranking according to a criterion that evaluates the effectiveness of a single feature in discriminating samples belonging to different classes. In the search procedure, the different feature subsets were obtained adding each time some features according to their position in the ranking.

In the experiments we also compared the performance of the proposed strategy with that obtained by considering other effective and widely used search strategies: the results confirmed the effectiveness of our approach.

The remainder of the paper is organized as follows: in Section 2 we will discuss the main works published in this field, in Section 3 we will illustrate the considered sets of features, while in Section 4 we will describe the feature evaluation methods. The experimental results will be illustrated in Section 5, while discussion an conclusions will be left eventually to Section 6.

2. Related works

The feature selection process typically consists of three basic steps: a search procedure for searching candidate feature subsets, a feature subset evaluation strategy and a stopping criterion. The search procedure is repeated until the stopping criterion is satisfied.

The procedure for searching candidate feature subsets must explore a search space, whose size depends on the total number of available features. Since for N features, there are 2^N potential feature subsets, an exhaustive search becomes computationally intractable as the number N increases, because of the resulting exponential growth of the search space. This is the reason why many heuristic algorithms for finding near-optimal solutions have been proposed in the literature [22]. In this framework, greedy selection [31], branch and bound [44] and floating searches [34] belong to the exponential search category. These algorithms use greedy stepwise strategies that generate feature subsets incrementally by adding the feature that produces the highest increment of the evaluation function. The sequential search category includes Sequential forward feature selection (SFFS), sequential backward feature elimination (SBFE) and bidirectional selection, which are greedy search algorithms that add or remove one feature at a time. Finally, in the context of stochastic search, Genetic Algorithms (GA) or Particle Swarm Optimization (PSO) have demonstrated themselves effective search tools for finding near-optimal solutions in complex and non-linear search spaces and have been widely used to solve feature selection problems [7,11,29,36,47,48].

As regards the way in which feature subsets are evaluated, feature selection methods are generally subdivided into three wide classes, namely filter, wrapper and embedded methods [5,33]. Given a feature subset, filter methods take into account its statistical or geometrical properties, while wrapper ones use the performance achieved by a certain trained classifier by considering that feature subset. Finally, embedded methods try to reduce the computational cost of repeating training and classification processes for each feature subset, by incorporating the feature selection as part of the training process. According to the above definition, the use of wrapper methods for feature selection implies that the evaluation of each feature subset is obtained through a costly process, requiring both the training of the considered classifier, and the obtained classification results. Thus, the main drawback of the wrapper method is the computational cost and, therefore, they typically require the use of near-optimal search strategies, which may produce acceptable results with a reasonable computational cost. Filters methods imply a non-iterative computation on the dataset, which can be much faster than a classifier training session. Moreover, filters methods evaluate intrinsic properties of the data, rather than the interactions of such data with a particular classifier: thus, the solutions provided should be more general and applicable to a larger family of classifiers. Ranking methods belong to the filter method category and perform feature selection in two steps, before classification or clustering tasks. In the first one, the features are ranked according to a certain statistical measure. Then, in the second one, the features with the highest rankings are selected. Examples of widely used statistical measures for filter methods are distance [26], correlation [22,24], information gain [41] and consistency [9]. One of the drawbacks of ranking methods is that there is no one general criterion for choosing the dimension of the feature space: this implies that it is difficult to select a cut-off on the number of features to be selected. Moreover, the selected feature subset might not be optimal, since important features that are less informative on their own but are highly informative when combined with other ones, could be discarded.

In the field of handwritten character recognition the feature selection process plays a key role for obtaining satisfactory performance [6,35,37]. As it is well known, in fact, the differences in writing styles due to many factors such as age, culture, degree of education and origin, result in a very large shape variability, which has led to the development of a large variety of feature sets. To cope with such a variability and to manage the increasing number of available features, a large number of feature extraction and feature selection methods have been proposed [49]. In [27], authors applied a suite of index-based and wrapper methods for feature selection, with reference to the problem of handwritten digits recognition and printed musical notation recognition. In [16] the authors proposed a Class Dependent Features (CDFs) technique for identifying and extracting the features innate to each class. In [45] a hybrid feature selection method for historical Document Image Analysis is proposed. In this study, an adapted greedy forward selection and a genetic selection are used in cascade. In [18] the authors proposed a user-guided feature subset selection algorithm, which uses a filter approach for an initial feature clustering and a wrapper approach for selecting the most suitable feature from each cluster. Finally, in [12] the authors presented a novel GA-based feature selection algorithm in which feature subsets are evaluated by means of an evaluation index, based on the Fisher Linear Discriminant method.

In this framework, the aim of our study is that of exploiting the properties of standard ranking algorithms to build up an experimental protocol for choosing the feature subset able to provide the best classification results. As anticipated in the Introduction, we have combined the use of such ranking techniques with a greedy search strategy to select feature subsets with increasing number of features, obtained by adding features progressively according to their position in the ranking. It is worth to remark that, at our best knowledge, there are no other works that propose similar ranking-based strategies for feature selection.

3. The feature sets considered

The two sets of features considered in this study are those proposed in [4,38].

The features of the first set (*SET*1 hereafter) measure three properties of a segmented image representing an input sample, related to the concavity, to the contour and to the character surface. The image is divided into 6 zones arranged on three rows and two columns. For each zone, 13 concavity measurements are computed using the 4-Freeman directions as well as other 4 auxiliary directions, totalizing 78 concavity features, normalized between 0 and 1. Then, in each zone, 8 contour features are extracted from a histogram of contour direction obtained by grouping the contour line segments between neighboring pixels based on the 8-Freeman direction. Therefore, there are 48 contour features for each image, normalized between 0 and 1. Finally, the last part of the feature vector is related to the character surface. In particular, the number of black pixels in each zone is counted and normalized between 0 and 1, thus obtaining 6 values for each image. Summarizing, the total number of features is $78 + 48 + 6 = 132$.

The second set of features (*SET*2 hereafter) is that used for describing the MFEAT (Multiple Features) dataset, publicly available from the UCI machine learning repository [14].[1] The MFEAT dataset consist in handwritten digits extracted from a collection of Dutch utility maps. Data are described by using six different groups of features, totaling 649 features. Each group of features was used to describe all the handwritten digits, which were arranged in separate databases: thus, there are 6 databases available:

- FOU: 76 Fourier coefficients of the character shapes;
- ZER: 47 Zernike moments;
- MOR: 6 morphological features.
- KAR: 64 Karhunen-Love coefficients;
- PIX: 240 pixel averages in 2×3 windows;
- FAC: 216 profile correlations.

4. Feature evaluation

As anticipated in the previous section, our protocol requires a feature ranking technique and a greedy search approach for choosing the feature subset able to provide the best classification results. In this study, we have considered five standard univariate measures, namely Chi-square, Relief, Gain ratio, Information Gain and Symmetrical uncertainty. Each univariate measure ranks the available features according to a criterion, which evaluates the effectiveness in discriminating samples belonging to different classes. Moreover, in order to compare our results with those attainable by other heuristic or greedy searching algorithms defined in the literature, we considered one of the most effective and widely used algorithm for searching feature subsets, the Best First (BF) search strategy (based on the beam search heuristics) [46], combined with two different criteria for feature evaluation, namely the Consistency Criterion, and the Correlation-based Feature Selection criterion. It is worth noticing that the aim of the feature-subset measures is that of evaluating the discriminant power of groups of features, taking also into account the complementarity and the redundancy of a single feature with respect to the others, while the univariate measures, were designed to estimate the discriminative power of each single feature at a time. The measures adopted are described in the following subsections.

4.1. Univariate measures

The Chi-Square (CS) measure estimates feature merit by using a discretization algorithm on the CS statistic [32]. For each feature, the related values are initially sorted by placing each observed value into its own interval. The next step uses the Chi-square statistic CS to determine whether the relative frequencies of the classes in adjacent intervals are similar enough to justify the merge. The formula for computing the CS value for two adjacent intervals is the following [32]:

$$CS = \sum_{i=1}^{2} \sum_{j=1}^{C} \frac{\left(A_{ij} - E_{ij}\right)^2}{E_{ij}} \tag{1}$$

where C is the number of classes, A_{ij} is the number of instances of the jth class in the ith interval and E_{ij} is the expected frequency of A_{ij} given by the formula:

$$E_{ij} = R_i C_i / NT \tag{2}$$

where R_i is the number of instances in the ith interval and C_j and NT are the number of instances of the jth class and total number of instances, respectively, in both intervals. The extent of the merging process is controlled by a threshold; whose value represent the maximum admissible difference among the occurrence frequencies of the samples in adjacent intervals. The value of this threshold was set heuristically during preliminary experiments.

The second measure considered is the Relief (RF), which uses instance-based learning to assign a relevance weight to each feature [28]. The assigned weights reflect the feature ability to distinguish among the different classes at hand. The algorithm works by sampling instances randomly from the training data. For each sampled instance, the nearest instance of the same class (nearest hit) and that of a different class (nearest miss) are found. A feature weight is updated according to how well its values distinguish the sampled instance from its nearest hit and nearest miss. Feature will receive a high weight if it differentiates between instances from different classes and has the same value for instances of the same class. Given a feature X, which can take the discrete values $\{x_1, x_2, \ldots, x_n\}$, its RF measure is computed according to the following formula [28]:

$$RF(X) = \frac{I_G(X) \sum_{x_i \in X} p(x_i)^2}{(1 - \sum_{c \in C} p(c)^2) \sum_{c \in C} p(c)^2} \tag{3}$$

where C is the class variable and I_G is a modified version of the Gini index [28].

Before introducing the last three univariate measures taken into account, let us briefly recall the information-theory concept of entropy. Given a discrete variable X, which can take the values $\{x_1, x_2, \ldots, x_n\}$, its entropy H(X) is defined as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i) \tag{4}$$

where $p(x_i)$ is the probability mass function of the value x_i. The quantity $H(X)$ represents an estimate of the uncertainty of the random variable X. The entropy concept can be used to define the conditional entropy of two random variables X and Y taking values x_i and y_j respectively, as:

$$H(X|Y) = -\sum_{ij} p(x_i, y_j) log_2 \frac{p(x_j)}{p(x_i, y_j)} \tag{5}$$

where $p(x_i, y_j)$ is the joint probability that $X = x_i$ and $Y = y_j$. The quantity in Eq. (5) represents the amount of randomness in the random variable X when the value of Y is known.

The above-defined quantities can be used to estimate the usefulness of a feature X to predict the class C of unknown samples. More specifically, such quantities can be used to define the Information Gain (IG) measure[2] [23, p. 56]:

$$IG(X) = H(C) - H(C|X) \tag{6}$$

[1] The dataset can be downloaded from the following link: https://archive.ics.uci.edu/ml/datasets/Multiple+Features.

[2] Note that the IG quantity is also known as *Mutual Information* between the feature and the class [43].

IG represents the amount by which the entropy of C decreases when X is given, and reflects additional information about C provided by the feature X.

The last two considered univariate measures uses the information gain defined in Eq. (6). The first one, called Gain Ratio (GR), is defined as the ratio between the information gain and the entropy of feature X to be evaluated [40]:

$$GR(X) = \frac{IG(X)}{H(X)} \tag{7}$$

Finally, the last univariate measure taken into account, called Symmetrical Uncertainty (SU), compensates for information gain bias toward attributes with more values and normalizes its value to the range [0, 1] [23, p. 57]:

$$SU(X) = 2.0\frac{IG(X)}{H(C) + H(X)} \tag{8}$$

4.2. Subset measures

As subset evaluation measures, we chose the Consistency Criterion [9] and the Correlation-based Feature Selection criterion [23]. The Consistency Criterion (CC) provides an effective measure of how well samples belonging to different classes are separated in a feature sub-space [32], whereas the Correlation-based Feature Selection criterion (CFS) measures the feature subset value by using a correlation-based heuristic.

The CC criterion is based on the concept of *inconsistency*. In the following description the term *pattern* will denote a part of an instance without a class label. A pattern is considered *inconsistent* if there are at least two instances that match all but their class labels; for example, an inconsistency is caused by instances (01, 1) and (01, 0) where the two features take the same values in the two instances while the class attribute varies, which is the last value in the instance.

The *inconsistency count* $I_c(p)$ for a pattern p is the number of times it appears in the data minus the largest number among different class labels. Let us assume that in a subset S and a pattern p appears in n_p instances such that $\sum_{i=1}^{n_c} n_i = n_p$, where n_i is the number of instances from the ith class and n_c the number of class of the problem at hand. If $c_j > c_i \; \forall i \in \{1, \ldots, n_c\} - \{j\}$, then $I_c = (n_p - c_j)$. The *inconsistency rate* I_r of a subset S is computed according to following formula:

$$I_r(S) = \frac{\sum_{p \in P} I_c(p)}{N} \tag{9}$$

where P is the set of all available patterns is S and N the number of instances in the data. Once the inconsistency rate $I_r(S)$ has been computed, the consistency measure function $f_{CC}(S)$ of S is defined as [9]:

$$f_{CC}(S) = 1 - I_r(S) \tag{10}$$

This CFS, instead, takes into account the usefulness of individual features for predicting class labels along with the level of intercorrelation among them. The idea behind this approach is that good subsets contain features highly correlated with the class and uncorrelated with each other. Denoting with X and Y two features, their correlation r_{XY} is computed as follows:

$$r_{XY} = 2.0\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)} \tag{11}$$

Given a feature selection problem in which the patterns are represented by means of a set Y of N features, the f_{CFS} function computes the merit of the generic subset $S \subseteq Y$, made of k features, as follows [23, p. 69]:

$$f_{CFS}(S) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{12}$$

where $\overline{r_{cf}}$ is the average feature-class correlation, and $\overline{r_{ff}}$ is the feature-feature correlation. Note that the numerator estimates the discriminative power of the features in X, whereas the denominator assesses the redundancy among them.

5. Experimental results

In order to assess the effectiveness of the proposed approach in handwritten character recognition problems, we performed three sets of experiments. In the first one, we used the feature set *SET*1 [38] for representing the samples of three real word databases, namely the well-known NIST-SD19 (NIST in the following, [21]), the Rimes database (RIMES in the following, [20]) and a database of characters segmented from postal addresses (PD in the following). We applied the proposed strategy to these databases and compared the results with those obtained by using the feature subset evaluation methods reported in Section 4. In the second sets of experiments, we performed a similar analysis but using the set of features *SET*2 [4]. As previously discussed, these features have been used to represent the samples of the publicly available MFEAT database. Finally, in the third sets of experiments, we characterized the groups features exhibiting higher discriminant power for both the sets *SET*1 and *SET*2. The effectiveness of the selected feature subsets was evaluated by using three different classification schemes, namely K-Nearest Neighbor (K-NN), Bagging and Random Forest.

The adopted classification schemes, the experimental protocol adopted and the three sets of experiments performed are described in the following subsections.

5.1. The classification schemes

The *K-Nearest Neighbor* algorithm (K-NN) is a well-known non-parametric method that can be used for both classification and regression [25]. According to this approach, an unknown sample is labeled with the most common label among its k nearest neighbors in the training set. The rationale behind the k-NN classifier is that, given an unknown sample $\mathbf{x}$ to be assigned to one of the c_i classes of the problem at hand, the a-posteriori probabilities $p(c_i|\mathbf{x})$ in the neighborhood of $\mathbf{x}$ may be estimated by looking at the class labels of the k nearest neighbors of $\mathbf{x}$. Despite its simplicity, K-NN has shown itself able to provide good results [19,30,39]. The following results were achieved by using the Mahalanobis distance, which, in a preliminary set of experiments, proved to be more effective than the Euclidean one.

Bagging is an ensemble method, whose name is derived from bootstrap aggregation. It generates multiple training sets (denoted as bootstrap sets) by sampling the original training set with replacement uniformly [2]. The effect is that, in these sets, some of the original examples can be repeated while others may be left out. Each bootstrap set is used to train a different component classifier and the final classification decision is based on the vote of each component classifier. The component classifiers are of the same general form, in our case decision trees, while their decisions can differ due to the different bootstrap set used for their training.

The term *Random Forest* does not refer to a single algorithm, but rather to a family of methods for building an ensemble of tree-based classifiers. The original algorithm proposed by Breiman in [3] is usually referred to in the literature as *Forest-RI* and it is used as reference method in most of the papers dealing with RF. Given a training set that contains N feature vectors, each consisting of M features, the forest-RI algorithm consists of the following steps applied to each tree:

1. Draw, from the dataset, N samples at random with replacement. The resulting set will be the training set associated with the starting node of the tree.

2. Set a number $K \ll M$.
3. At each node, randomly draw K features from the set of available ones.
4. For each of the K features drawn, consider its values in the training set and choose the best binary split value according to the Gini index [15]. Then, select the feature with the best index value and generate two new nodes by splitting the samples associated with the original node according to such a value.
5. Grow the tree to its maximum size according to the stopping criterion chosen.[3]
6. Leave the tree unpruned.

Once the forest has been built, an unknown sample is labeled according to the Majority Vote rule: i.e., it is labeled with the most popular class among those provided by the ensemble trees. It is worth noting that in [2] it has been proved that RF does not over-fit as more trees are added, but rather its generalization error tends to a limiting value.

5.2. The experimental protocol

In our experiments, the methods for feature evaluation illustrated in Section 4 have been investigated applying the experimental protocol described as follows.

We applied the univariate measures illustrated in Section 4 to these data, obtaining 5 different feature rankings. Let us consider the ranking provided by the first univariate measure, namely CS: by using this feature ranking we generated different representations for each database, each containing an increasing number of features. More specifically, if N is the number of available features, we generated $(N \div n_s)$ datasets in the following way: in the first one, the samples were represented by using the first n_s features in the ranking, in the second one by using the first $n_s{*}2$ features, in the third one the first $n_s{*}3$ features and from there on by adding each time the successive n_s features in the ranking. In the last dataset, the samples were represented by using all the available N features. The same procedure has been repeated for the other univariate measures taken into account. Summarizing, for each database, we obtained 5 different feature rankings, each used to generate $(N \div n_s)$ different sets of data with increasing number of features. Each of them was used in the experiments for evaluating the obtainable classification results.

As regards the classification process, we considered the three classification schemes described previously, using a 10-fold validation strategy. Moreover, to deal with the randomness of the classification algorithms, we performed 20 runs for each experiment. The results reported in the following were obtained by averaging the values over the 20 runs.

5.3. Experiments with feature set SET1

In these experiments the feature set *SET*1 was used to represent the samples of NIST, Rimes and PD databases. The NIST database contains binary images of Handwriting Sample Forms(HSFs) and segmented handprinted digit and alphabetic characters from those forms. The database contains eight series of images, denoted by *hsf1, hsf2, ... , hsf8*. We have considered handwritten uppercase and lowercase letters (52 classes). In particular we taken into account the image series hsf4, containing 23,941 characters (11941 uppercase and 12,000 lowercase), and the image series hsf7, containing 23,670 characters (12092 uppercase and 11,578 lowercase). We merged them to form a unique database of 47,611 samples. Finally, the characters are isolated, labeled and stored in 128 × 128 pixel images.

The Rimes dataset contains real world handwritten words, in French, written by more than 1300 volunteers. RIMES is a publicly available database and it has been largely adopted for performance evaluation of handwriting recognition systems. The 4047 word images of RIMES were processed in order to extract a sub-image containing connected components of ink. These words were then shown to 6 human experts, who labeled the connected components extracted previously, providing also the corresponding transcript. At the end of this process, from the 9869 samples that were manually classified and transcribed, a subset of 4768 samples, corresponding to isolated characters has been extracted and used for our experiments.

Finally, the PD database is composed of 16,064 characters (5486 uppercase and 10,578 lowercase letters), which were extracted and manually labeled from a database of variable size images of postal addresses. It is worth noticing that this database belongs to a private company, thus they are not publicly available.

In this set of experiments, the protocol described above has been applied, with $n_s = 5$. As a consequence, 65 sets of data were generated for each database. The classification results obtained are shown in Figs. 1–3. In each plot, x-axis reports the number of features used to represent the input samples, while y-axis reports the corresponding classification results, expressed in terms of recognition rates.

It is interesting to note that, accepting a reduction of the recognition rate of about 5% with respect to its maximum value, it is possible to select a very small subset of features, namely, the first 30 ones in the rankings, strongly reducing the computational complexity of the classification problem. The plots in the figures also show that, using the first 60 features in the rankings (i.e. less than 50% of all the available ones), the reduction of the recognition rate is less than 2%.

For the sake of clearness, we have also summarized the classification results obtained in Tables 1–3. The first row in the tables reports the recognition rates (RR, expressed as percentages) obtained with all the available features, while the other ones report the

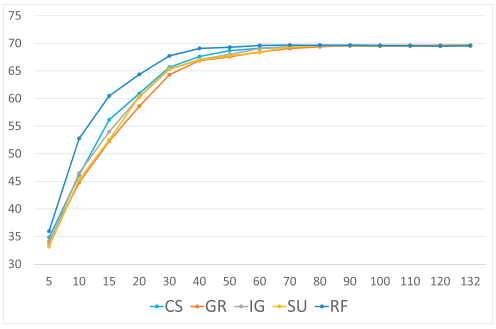
Table 1
Best recognition rates on NIST database using K-NN, bagging and random forest classifiers.

Method	KNN		BAG		RF	
	RR	NF	RR	NF	RR	NF
–	69.56	132	72.51	132	74.52	132
SU	69.60	110	72.52	80	74.88	80
RF	69.70	70	72.57	110	74.86	80
CS	69.64	80	73.61	80	74.83	80
GR	69.64	120	72.58	110	74.90	80
IG	69.63	120	72.57	100	74.85	80
CFS	67.79	54	72.06	54	74.26	54
CC	54.66	16	65.46	16	66.79	16

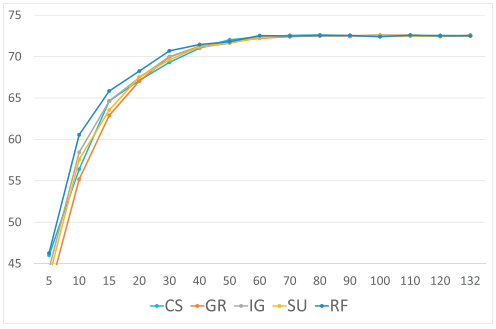
Table 2
Best recognition rates on RIMES database using K-NN, bagging and random forest classifiers.

Method	KNN		BAG		RF	
	RR	NF	RR	NF	RR	NF
–	71.85	132	73.40	132	75.26	132
SU	72.05	80	74.15	80	75.69	80
RF	72.27	90	74.07	80	75.70	80
CS	72.23	110	74.21	80	75.78	70
GR	72.11	110	74.02	70	75.61	80
IG	72.03	80	74.16	90	75.70	70
CFS	69.57	50	73.31	50	75.19	50
CC	52.34	16	66.65	16	68.49	16

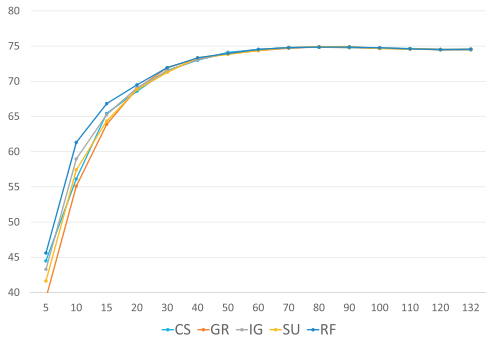
[3] Node splitting usually is stopped when one of the following conditions occur: (i) The number of samples in the node to be split is below a given threshold; (ii) all the samples in the node belong to the same class



(a)

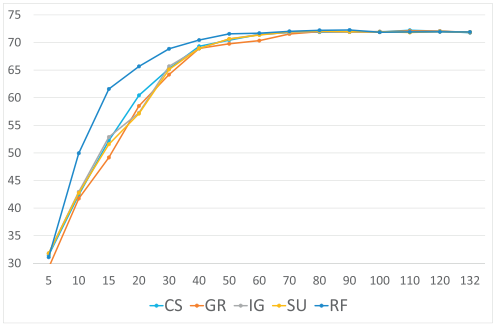


(b)

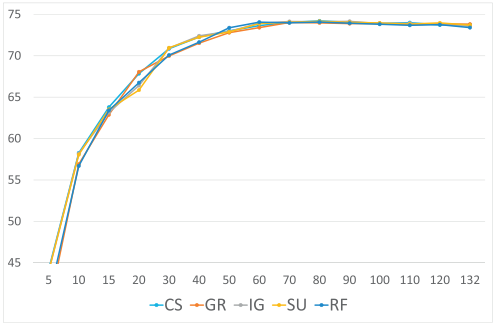


(c)

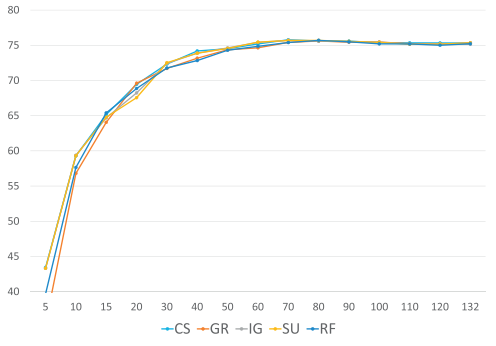
Fig. 1. Experimental results on NIST database using K-NN (a), Bagging (b) and Random Forest (c) classifiers.



(a)



(b)



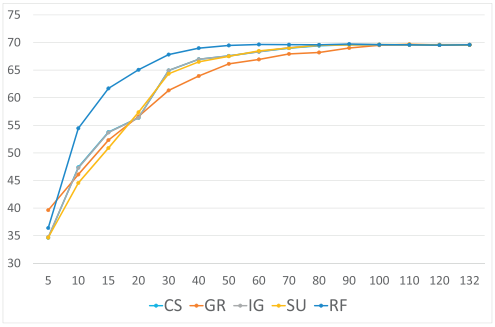
(c)

Fig. 2. Experimental results on RIMES database using K-NN (a), Bagging (b) and Random Forest (c) classifiers.

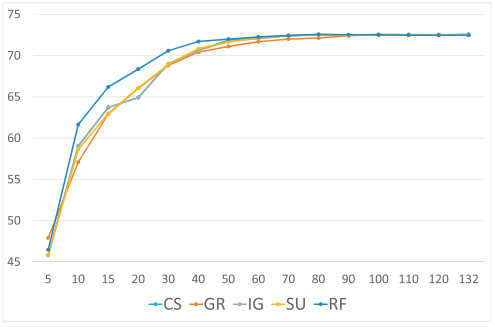
Table 3
Best recognition rates on PD database using K-NN, bagging and random forest classifiers.

Method	KNN		BAG		RF	
	RR	NF	RR	NF	RR	NF
–	69.59	132	72.45	132	74.68	132
SU	69.63	110	72.55	110	75.38	80
RF	69.2	90	72.58	80	75.34	80
CS	69.61	100	72.54	100	75.90	80
GR	69.67	110	72.55	110	74.74	100
IG	69.64	90	72.55	110	75.84	90
CFS	67.29	52	71.82	52	74.10	52
CC	51.41	18	67.05	18	68.30	18

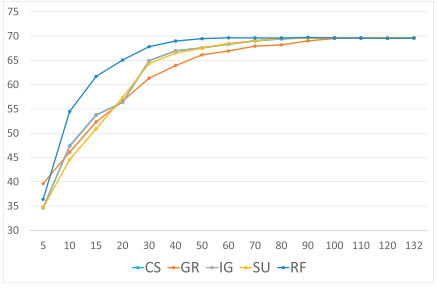
best classification results for each feature ranking, together with the corresponding number of selected features (NF). The last two rows of the tables show the classification results obtained with CFS and CC feature selection methods, and the corresponding number of selected features. The data in the tables show that, in the large majority of cases, the recognition rates obtained by the proposed feature selection strategy outperform those obtained by using the other feature subset selection methods considered, as well as those obtained by using of all the available features. Moreover, the best results are always achieved using a number of features significantly smaller than 132. In the average, the number of features allowing us to obtain the best results is about 90, i.e. about 70% of the total number of features. Finally, as regards the performance of the subset feature selection methods, CFS provides recognitions slightly worse, but using in the average a smaller number of features. The CC feature selection method, instead, performs significantly worse



(a)

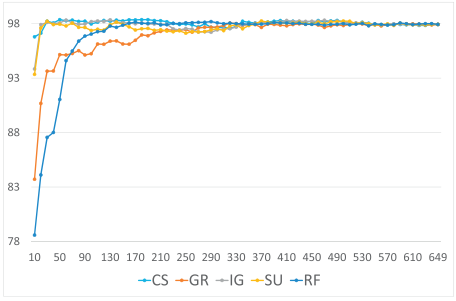


(b)

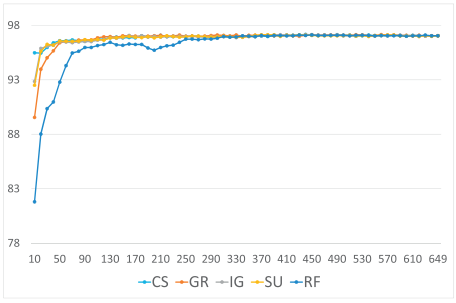


(c)

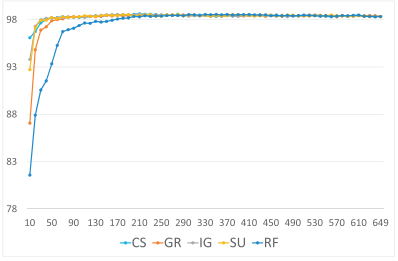
Fig. 3. Experimental results on PD database using K-NN (a), Bagging (b) and Random Forest (c) classifiers.



(a)



(b)



(c)

Fig. 4. Experimental results on MFEAT database using K-NN (a), Bagging (b) and Random Forest (c) classifiers.

than all the other methods, selecting in the average a too small number of features.

5.4. Experiments with feature set SET2

In order to verify the generality of the proposed strategy, we also applied our experimental protocol to a standard database of handwritten digits, the MFEAT database, publicly available from the UCI machine learning repository [14]. The samples of this database are represented by using the feature set *SET2* described in Section 3, which includes six groups of features, totaling 649 features. The database contains 2000 instances of handwritten digits, 200 for each digit. Following the experimental setup illustrated in Section 5.2, we generated 5×65 sets of data, obtained by setting $n_s = 10$.

In this set of experiments, the protocol described above has been applied, with $n_s = 5$. As consequence, 325 sets of data were generated. The classification results obtained are shown in Fig. 4. In each plot, x-axis reports the number of features used to represent the input samples, while y-axis reports the corresponding classification results, expressed in terms of recognition rates.

These results are in good accordance with those illustrated in the previous Subsection. A reduced number of features in the top positions of the rankings (less than 100 in the large majority of cases) allowed us to obtain high performance, with a reduction of the best recognition rate less than 2%. In particular, considering the rankings provided by both the Symmetrical Uncertainty (SU) measure and the Chi-Square (CS) measure, it is possible to select an even smaller set of features (including the first 40 features in the ranking) with a negligible reduction of the best recognition rate.

For the sake of clearness, in Table 4 we have also summarized the classification results obtained with the three classification schemes. The first row in the table reports the recognition rates (RR, expressed as percentages) obtained with all the available features, while the other ones report the best classification results for each feature ranking, together with the corresponding number of selected features (NF). The last two rows of the tables

Table 4
Best recognition rates on MFEAT database using K-NN, bagging and random forest classifiers.

Method	KNN		BAG		RF	
	RR	NF	RR	NF	RR	NF
–	97.8	649	94.75	649	98.32	649
SU	98.25	370	97.14	380	98.54	280
RF	98.19	290	97.13	450	98.55	350
CS	98.37	50	94.90	130	98.61	210
GR	98.06	320	97.15	560	98.52	220
IG	98.36	130	97.15	450	98.55	210
CFS	98.51	147	97.17	147	98.89	147
CC	94.63	7	93.85	7	95.47	7

show the classification results obtained with CFS and CC feature selection methods, and the corresponding number of selected features.

The data in the table show that the best results are always achieved using a number of features significantly lower than 649. On average, the number of features allowing us to obtain the best results are fewer than 220, i.e. about 34% of the total number of features. As regards the performance of feature subset selection methods, CFS provides recognition rates slightly better than those of the other methods, using, on average, a smaller number of features. It is worth noticing, however, that the rankings provided by both the Symmetrical Uncertainty (SU) measure and the Chi-Square (CS) measure allowed us to obtain very similar results selecting less than 40 features. The CC feature selection method, instead, performs significantly worse than all the other methods and selects, on average, a too small number of features, thus confirming the results of the previous Subsection.

5.5. Feature characterization

Another interesting aspect to point out, is the analysis of how many times the features of each group are ranked among the initial positions of the rankings. The aim is that of characterizing the groups of features exhibiting higher discriminant power for both the sets *SET*1 and *SET*2.

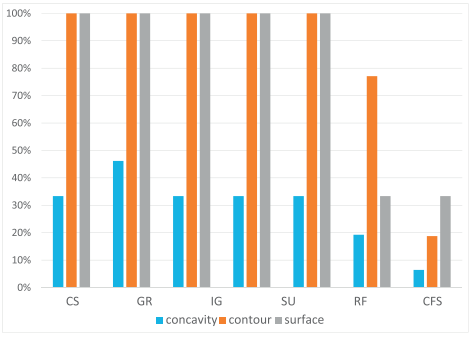
As regards the feature set *SET*1, Fig. 5 shows the histograms reporting the percentage of features of each category, namely those representing concavity information, contour information and character surface information, which have been included in the best feature subsets obtained by using the univariate measures and the methods for feature subset evaluation, with reference to the RF classifiers. The histograms in the figure refer to NIST, RIMES and PD data, respectively.

The results indicate that the features representing contour information and those representing character surface information have very high discriminant power and are almost always selected. On the contrary, the features associated with concavity information, whose number is higher than that of the other categories, seem to be less distinctive and, in most cases, more than 50% of such features have been discarded.

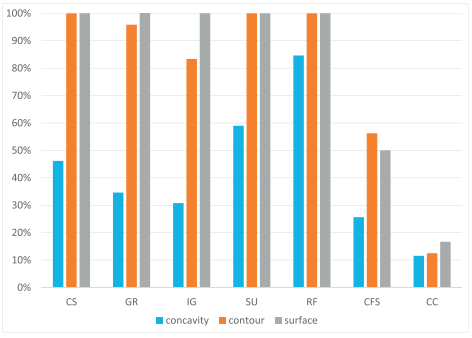
Similarly, Fig. 6 shows the histograms reporting the percentage of features of each category included in the best feature subsets with reference to the feature set *SET*2, the MFEAT database and the RF classifiers. In this case, morphological and profile correlations features seem much more distinctive than the others and have been included in most of the experiments in the best selected feature subset.

6. Discussion and conclusions

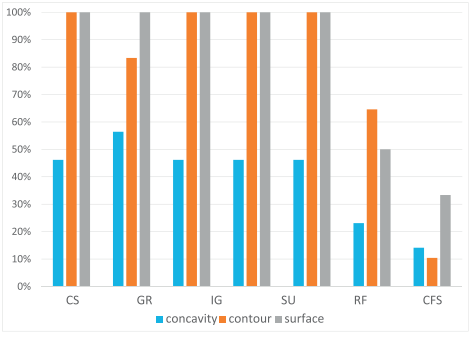
In this study we have tried to overcome the main drawbacks found in common feature selection algorithms, namely the high



(a)



(b)



(c)

Fig. 5. The percentage of features of each category (concavity, contour and surface) that have been included in the best feature subsets for PD (a) RIMES (b) and NIST (c) databases.

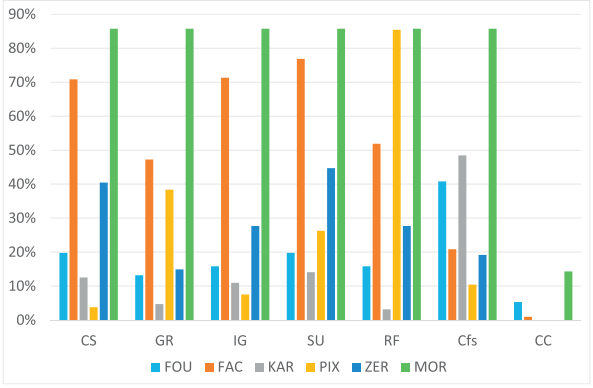


Fig. 6. The percentage of features of each category that have been included in the best feature subsets for MFEAT database.

computational cost of the procedures for searching effective feature subsets, and the difficulties to take into account the effects of the interactions among features. As discussed in the Introduction, to reduce the computational time, heuristic or greedy searching algorithms have been proposed instead of exhaustive ones, but they generally imply some limitations in evaluating the interactions among features and have a computational cost, which can still be important. An alternative approach requiring less computational time is that of applying the evaluation criteria to each single feature: in this way, each feature can be positioned in a ranking according to the value of the above measure but, obviously, the effects of the interactions among features are completely ignored.

In this framework, we have presented a study in which the properties of ranking algorithms have been used to build up an experimental protocol for finding feature subsets, which allow classification performance improvements in handwriting applications. To this aim, we have combined the use of such ranking techniques with a greedy search strategy to select feature subsets with an increasing number of features, obtained by adding features progressively according to their position in the ranking. In our study, we have considered one of the most effective and widely used set of features in handwriting recognition, and we have used these features for representing samples of three real word databases. The experimental results confirmed that it is possible to choose a reduced set of features without affecting the overall classification rates. The results have also shown that it is possible to reduce significantly the number of features, and consequently the complexity of the classification tasks, accepting a limited reduction of the recognition rate. The amount of such reduction can be set by the user, depending on the application scenarios, which in some cases may require very limited time for character classification (e.g postal sorting). In particular, it is possible to reduce the number of features up to 30%, without any loss in terms of recognition rate, while a higher reduction of the feature number (up to 70%) implies a reduction of the recognition rate less than 2%. For comparison purposes, we have also applied our experimental protocol to another standard database of handwritten digits represented by using a very large set of features, obtaining a similar behavior: in this case the loss in terms of recognition rate using a reduced set of features was even more limited.

Finally, considering the whole set of experiments performed on the above real world databases, we have also characterized the type of features that exhibit higher discriminant power among the whole set of available ones.

In conclusion, the results of these experiments suggest that the idea of using a reduced feature set, namely that obtained by discarding the features in the lower positions of the ranking, can provide very interesting results, significantly reducing the computational complexity of the whole recognition system with very limited effects on the classification performance.

Following this suggestion, as a future work, we would like to use other more complex and computationally expensive feature selection techniques on reduced feature subsets, obtained by selecting the features in the highest positions in the ranking.

References

[1] J. Bins, B.A. Draper, Feature selection from huge feature sets, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, 2, 2001, pp. 159–165 vol.2.
[2] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[3] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[4] M.P.W. van Breukelen, D.M.J. Tax, J.E. den Hartog, Handwritten digit recognition by combined classifiers, Kybernetika 34 (1998) 381–386.
[5] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28. 40th-year commemorative issue.
[6] 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. 128–131.
[7] L.P. 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: 2010 20th International Conference on Pattern Recognition, 2010, pp. 4352–4355.
[8] R.M. Cruz, G.D. Cavalcanti, I.R. Tsang, R. Sabourin, Feature representation selection based on classifier projection space and oracle analysis, Expert Syst. Appl. 40 (9) (2013) 3813–3827.
[9] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (1–2) (2003) 155–176.
[10] C. De Stefano, F. Fontanella, A. Marcelli, A. Parziale, A. Scotto di Freca, Rejecting both segmentation and classification errors in handwritten form processing, in: 2014 14th International Conference on Frontiers in Handwriting Recognition, 2014, pp. 569–574.
[11] C. De Stefano, F. Fontanella, C. Marrocco, A GA-Based Feature Selection Algorithm for Remote Sensing Images, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 285–294.
[12] C. De Stefano, F. Fontanella, C. Marrocco, A. Scotto di Freca, A GA-based feature selection approach with an application to handwritten character recognition, Pattern Recognit. Lett. 35 (2014) 130–141. Frontiers in Handwriting Processing.
[13] M. Diem, S. Fiel, A. Garz, M. Keglevic, F. Kleber, R. Sablatnig, ICDAR 2013 competition on handwritten digit recognition (HDRC 2013), in: 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 1422–1427.
[14] A. Frank, A. Asuncion, UCI machine learning repository, 2010.
[15] C. Gini, Measurement of inequality of incomes, Econ. J. 31 (121) (1921) 124–126.
[16] K. Goel, R. Vohra, A. Bakshi, A novel feature selection and extraction technique for classification, in: 2014 14th International Conference on Frontiers in Handwriting Recognition, 2014, pp. 104–109.
[17] S. Goswami, A.K. Das, A. Chakrabarti, B. Chakraborty, A feature cluster taxonomy based feature selection technique, Expert Syst. Appl. 79 (C) (2017) 76–89.
[18] S. Goswami, A.K. Das, A. Chakrabarti, B. Chakraborty, A feature cluster taxonomy based feature selection technique, Expert Syst. Appl. 79 (2017) 76–89.
[19] M. Govindarajan, R. Chandrasekaran, Evaluation of k-nearest neighbor classifier performance for direct marketing, Expert Syst. Appl. 37 (1) (2010) 253–258.
[20] E. Grosicki, M. Carre, J.-M. Brodin, E. Geoffrois, RIMES evaluation campaign for handwritten mail processing, in: ICFHR 2008 : 11th International Conference on Frontiers in Handwriting Recognition, Concordia University, Montreal, Canada, 2008, pp. 1–6.
[21] J. Grother, NIST special database 19: hand printed forms and characters database, Technical Report, National Institute of Standards and Technology, Gaithersburg, 1995.
[22] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[23] M. Hall, Correlation-based Feature Selection for Machine Learning, Ph.D. thesis, University of Waikato, 1999.
[24] 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.
[25] P.E. Hart, R.O. Duda, D.G. Stork, Pattern Classification, John Wiley & sons, Inc., New York, USA, 2001.
[26] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Trans. Pattern Anal. Mach. Intell. 24 (3) (2002) 289–300.
[27] W. Homenda, A. Jastrzebska, A practical study on feature selection methods in pattern recognition: examples of handwritten digits and printed musical notation, in: 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), 2017, pp. 1035–1038.
[28] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: European Conference on Machine Learning, 1994, pp. 171–182.
[29] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern recognition, Pattern Recognit. 33 (1) (2000) 25–41.
[30] M. Kumar, M. Jindal, R. Sharma, k-nearest neighbor based offline handwritten gurmukhi character recognition, in: 2011 IEEE International Conference on Image Information Processing (ICIIP), IEEE Computer Society, Shimla, Himachal Pradesh, India, 2011, pp. 1–4.
[31] N. Kwak, C.-H. Choi, Input feature selection for classification problems, IEEE Trans. Neural Netw. 13 (1) (2002) 143–159.
[32] H. Liu, R. Setiono, Chi2: feature selection and discretization of numeric attributes, in: Seventh International Conference on Tools with Artificial Intelligence (ICTAI), IEEE Computer Society, Washington, DC, USA, 1995, pp. 388–391.
[33] J. Miao, L. Niu, A survey on feature selection, Procedia Comput. Sci. 91 (2016) 919–926. Promoting Business Analytics and Quantitative Management of Technology: 4th International Conference on Information Technology and Quantitative Management (ITQM 2016).
[34] S. Nakariyakul, D.P. Casasent, An improvement on floating search algorithms for feature subset selection, Pattern Recognit. 42 (9) (2009) 1932–1940.
[35] 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.
[36] 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.

[37] 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. J. Pattern Recognit. Artif. Intell. (IJPRAI) 17 (2003) 2003.
[38] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1438–1454.
[39] J. Prez-Cortes, R. Llobet, J. Arlandis, Fast and accurate handwritten character recognition using approximate nearest neighbours search on large databases, in: F. Ferri, J. Iṣesta, A. Amin, P. Pudil (Eds.), Advances in Pattern Recognition, Lecture Notes in Computer Science, 1876, Springer Berlin – Heidelberg, 2000, pp. 767–776.
[40] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[41] L.E. Raileanu, K. Stoffel, Theoretical comparison between the Gini index and information gain criteria, Ann. Math. Artif. Intell. 41 (1) (2004) 77–93.
[42] J.M. Saavedra, Handwritten digit recognition based on pooling svm-classifiers using orientation and concavity based features, in: E. Bayro-Corrochano, E. Hancock (Eds.), Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer International Publishing, Cham, 2014, pp. 658–665.
[43] C.E. Shannon, A mathematical theory of communication, SIGMOBILE Mob. Comput. Commun. Rev. 5 (1) (2001) 3–55.
[44] P. Somol, P. Pudil, J. Kittler, Fast branch and bound algorithms for optimal feature selection, Pattern Anal. Mach. Intell. IEEE Trans. 26 (7) (2004) 900–912.
[45] H. Wei, K. Chen, R. Ingold, M. Liwicki, Hybrid feature selection for historical document layout analysis, in: 2014 14th International Conference on Frontiers in Handwriting Recognition, 2014, pp. 87–92.
[46] L. Xu, P. Yan, T. Chang, Best first strategy for feature selection, in: 9th International Conference on Pattern Recognition (ICPR 1988), IEEE Computer Society, Rome, Italy, 1988, pp. 706–708 Vol.2.
[47] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, IEEE Trans. Cybern. 43 (6) (2013) 1656–1671.
[48] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2016) 606–626.
[49] M. Arif Mohamad, H. Hassan, D. Nasien, H. Haron, A review on feature extraction and feature selection for handwritten character recognition, Int. J. Adv. Comput. Sci. Appl. (IJACSA) 6 (2) (2015) 204–212.