Solving Feature Subset Selection Problem by a Hybrid Metaheuristic

Miguel García-Torres, Félix García-López, Belén Melián-Batista,
José A. Moreno-Pérez and J. Marcos Moreno-Vega,
Dpto. de E.I.O. y Computación.
Universidad de La Laguna

February 10, 2005

Abstract

The aim of this paper is to develop a hybrid metaheuristic based on Variable Neighbourhood Search and Tabu Search for solving the Feature Subset Selection Problem in classification. Given a set of instances characterized by several features, the classification problem consists of assigning a class to each instance. Feature Subset Selection Problem selects a relevant subset of features from the initial set in order to classify future instances. The proposed hybrid metaheuristic is compared with a Genetic Algorithm proposed in the literature. Although he hybrid metaheuristic and the genetic algorithm had a similar performance according to the accuracy percentages, the hybrid metaheuristic provided a higher reduction in the set of features.

1 Introduction

The purpose of a classification problem is to classify instances, that are characterized by a set of features, by determining the class to which each instance belongs. In order to achieve this classification, a set of cases or instances of a known class is given.

The aim of the feature subset selection problem is to determine the subset of features that optimally carry out the classification task. Selecting the optimal feature subset is an NP-hard optimization problem [17]. Thus it is impossible to explore the whole solution space. Metaheuristics provide procedures for finding solutions with reasonable efficacy and efficiency. The application of evolutive procedures such as Genetic Algorithms [22] are described in the literature. In this work a hybrid metaheuristic (VNSTS) based on variable neighbourhood search (VNS) and tabu search (TS) is developed to solve the feature subset selection problem. It is then compared with a genetic algorithm proposed in the literature.

Variable Neighborhood Search [12], [13] is a recent metaheuristic for solving combinatorial and global optimization problems based upon a simple principle: systematic changes of neighborhood within the search. When a local minimum is reached, a shake procedure performs a random search. This determines a new starting point for running an improvement method. Basic Tabu Search (TS) [10] maintains a selective history H of the states encountered during the search, and replaces the neighborhood of the current solution $N(s)$ by a modified neighborhood, which may be denoted

$N(H, s)$. History therefore determines which solutions may be reached by a move from the current solution, selecting s' from $N(H, s)$.

Next section introduces the feature selection problem with a formal description of the problem and the most common solution strategies. Section 3 describes the main elements of the Variable Neighborhood Search and Tabu Search metaheuristics in the application to this problem. The outcomes of the computational experience are shown in section 4 for several data sets and motivate our conclusions, summarized in section 5.

2 The Feature Subset Selection Problem

The paradigms of learning *Instance-Based Learning* and *Bayesian Learning* [19] are considered in order to perform the classification task. *Instance-Based Learning* uses the nearest examples to predict the label of the instance, given a set of examples and an instance to be classified. In this work, the instance-based algorithm called IB1 [2], which classifies each instance with the label of the nearest example, is used. For this purpose, IB1 considers all the features, although, in general, only a few of them are highly relevant. *Bayesian Learning* uses probability as an approach for classification. The Naive Bayes classifier, which is based on the Bayes theorem, estimates "a posteriori" probabilities of all possible classifications. For each instance, the classification with the highest "a posteriori" probability is chosen.

With the purpose of selecting the subset of features, both the wrapper approach and the filter approach can be considered. The first strategy applies a different methodology in the training and test phases. The main disadvantage of this procedure is that it ignores the effect of the subset of features in the induction algorithm. In this approach, there are two relevant algorithms: FOCUS [3], which examines the entire subset of features selecting those that minimize the number of features among those classifying correctly all of the training instances, and RELIEF [16], which is a random algorithm that assigns a weight to each feature based on the two nearest instances. The wrapper strategy applies the same algorithm in the entire process. Two well known wrapper approaches are Sequential Forward Selection (SFS) and Sequential Backward Elimination (SBE) [6]. SFS starts with an empty subset of features and, at each step, adds the feature that improves the most the classification. This process is iterated until no improvement is possible. In SBE the initial subset consists of all the available features and, at each step, the worst feature is eliminated from the subset. As in SFS, this process is repeated until no improvement is possible. The VNSTS metaheuristic proposed in this work is based on the wrapper approach.

Let A be a set of given instances characterized by d features $X = \{X_j : j = 1, \ldots, d\}$, which are either a nominal or a linear attribute. An attribute is linear if the evaluation of the difference between two of its values has sense (being discrete or continuous); otherwise it is nominal. With the purpose of carrying out the classification task, we consider as training instances the subset of instances $T \subset A$ in which labels are known, and as validation instances the subset $V = A \setminus T$ of instances to be classified. The aim of the classification problem is to obtain the label of the instances using only a subset of features. Therefore the objective of our problem is to find the subset of features $S \subseteq \{X_j : j = 1, \ldots, d\}$ with the highest accuracy percentage.

In order to estimate the accuracy percentages of a subset of features S on a given set of instances $B \subseteq A$, we consider the method widely used in the literature called k-fold cross validation. This method divides the set of instances B at random into k disjoint subsets of equal size $B_1, B_2, \ldots, B_k$ and performs k executions. In the execution i, B_i is considered as test set and the union of the other subsets $T_i = B \setminus B_i$ as training set. In each trial i, the test instances B_i are classified using the learning algorithm. The estimated accuracy percentage of the classifier is the average of the accuracy percentages over all the trials. The estimated accuracy percentage of a subset of features S on a given set of instances B using *cross-validation* is stated as follows:

$$f_B(S) = 100 \frac{|a \in B : \tilde{c_a} = c_a|}{|B|} \tag{1}$$

where c_a is the class of each instance a and $\tilde{c_a}$ is the class assigned by the classifier.

For the purpose of guiding the search for the best subset of features (training) and measuring the effectiveness of a particular subset of features after the search algorithm has chosen it as solution of the problem (validation), the function (1) for 2-fold cross-validation is used. To guide the search, $f_T(\cdot)$ is considered and to measure the effectiveness, $f_V(\cdot)$ is used. In training and validation, we consider 5×2 cross-validation (5×2cv) [7] that consists of dividing the set V into 2 folds and then conducting two trials. This is done for 5 random arrangements of V.

3 Application of VNS and TS to the Feature Subset Selection Problem

The aim of this section is to describe the characteristics of the proposed hybrid metaheuristic applied to the Feature Subset Selection Problem.

Variable Neighborhood Search (VNS) [12], [13] is a recent metaheuristic for solving combinatorial and global optimization problems based upon the simple principle of systematically changing the neighborhoods within the search. Let $\mathcal{N}_k$, $(k = 1, \ldots, k_{max})$ be a finite set of neighborhood structures, and $\mathcal{N}_k(s)$ be the set of solutions in the k^{th} neighborhood of a solution s. Neighborhoods $\mathcal{N}_k$ may be induced from metric functions introduced into a solution space S. If $d(.,.)$ is this distance then take increasing values d_k, $k = 1, ..., k_{max}$ and set $N_k(s) = \{s' \in S : d(s, s') \leq d_k\}$. Most local search heuristics use only one neighborhood structure $\mathcal{N}$. Therefore a series of nested neighborhoods are obtained from a single neighborhood by taking $\mathcal{N}_1(s) = \mathcal{N}(s)$ and $\mathcal{N}_{k+1}(s) = \mathcal{N}(\mathcal{N}_k(s))$, for every solution s. This means that a move to the k-th neighborhood is performed by repeating k times a move to the original neighborhood. A solution $s' \in S$ is a *local minimum* with respect to $\mathcal{N}_k$ if there is no solution $s \in \mathcal{N}_k(s') \subseteq S$ better than s' (i.e., such that $f(s) < f(s')$ where f is the objective function of the problem).

A usual strategy with two neighborhoods consists in performing local searches using the first neighborhood from points s' that belong to the second neighborhood of the current solution (i.e. $s' \in \mathcal{N}_2(s)$). The Basic Variable Neighborhood Search (BVNS) method uses deterministic changes in the neighborhood structure for perturbation or shaking. Its steps are given in Figure 1. The stopping condition may be the maximum CPU time allowed, the maximum number of iterations, or the maximum number of iterations between two improvements.

Basic Tabu Search (TS) [12], [13] maintains a selective history H of the states encountered during the search, and replaces the neighborhood of the current solution $N(s)$ by a modified

```
Initialization.
    Select the set of neighborhood structures $\mathcal{N}_k$, for $k = 1, \ldots, k_{max}$.
    Find an initial solution $s$.
    Choose a stopping condition.
Iterations.
    Repeat the following sequence until the stopping condition is met:
      (1) Set $k \leftarrow 1$.
      (2) Repeat the following steps until $k = k_{max}$:
          (a) Shaking.
              Generate a point $s'$ at random from the $k^{th}$ neighborhood of $s$ ($s' \in \mathcal{N}_k(s)$).
          (b) Local search.
              Apply some local search method with $s'$ as initial solution; denote as $s''$ the obtained
              local optimum.
          (c) Move or not.
              If this local optimum is better than the incumbent, then $s \leftarrow s''$, and continue the
              search with $\mathcal{N}_1$ ($k \leftarrow 1$); otherwise, set $k \leftarrow k + 1$.
```

Figure 1: Basic Variable Neighborhood Search.

neighborhood, which may be denoted $N(H,s)$. History therefore determines which solutions may be reached by a move from the current solution, selecting s' from $N(H,s)$.

In the TS strategies based on short term considerations, $N(H,s)$ characteristically is a subset of $N(s)$, and the tabu classification serves to identify elements of $N(s)$ excluded from $N(H,s)$. In the intermediate and longer term strategies, $N(H,s)$ may contain solutions not in $N(s)$, generally consisting of selected elite solutions (high quality local optima) encountered at various points in the solution process. Such elite solutions typically are identified as elements of a regional cluster in intermediate term intensification strategies, and as elements of different clusters in longer term diversification strategies.

TS also uses history to create a modified evaluation of currently accessible solutions. This may be expressed formally by saying that TS replaces the objective function $f(s)$ by a function $f(H,s)$, which has the purpose of evaluating the relative quality of currently accessible solutions. It is provided by the use of frequency based memory. The relevance of this modified function occurs because TS uses aggressive choice criteria that seek a best s'; i.e., one that yields a best value of $f(H,s)$, over a candidate set drawn from $N(H,s)$. Moreover, modified evaluations often are accompanied by systematic alteration of $N(H,s)$, to include neighboring solutions that do not satisfy customary feasibility conditions.

For large problems, where $N(H,s)$ may have many elements, or for problems where these elements may be costly to examine, the aggressive choice orientation of TS makes it highly important to isolate a candidate subset of the neighborhood, and to examine this subset instead of the entire neighborhood. Because of the significance of the candidate subset, we refer to it explicitly by the notation $N'(H, s)$.

Therefore, the Tabu Search procedure, instead of simply selecting the best neighbor s' of s with respect to f as the greedy local search does, selects the solution in $N'(H, s)$ that minimizes $f(H, s)$. The selected solution s' is called a highest evaluation candidate.

3.1 Initialization

In order to perform the initialization step of the hybrid metaheuristic, the neighborhood structure and the procedure that generates the initial solution must be defined.

The k^{th} neighborhood of the solution S, $\mathcal{N}_k(S)$, consists of all the solutions that can be reached from S by exchanging k features in the solution with k features out of the solution. Let $d(S, S')$ be the distance between the solutions $S = \{X_1, ..., X_r\}$ and $S' = \{X'_1, ..., X'_r\}$ defined as follows.

$$d(S, S') = r - |\{X_1, ..., X_r\} \bigcap \{X'_1, ..., X'_r\}|$$

Then the k^{th} neighborhood of the solution S, $\mathcal{N}_k(S)$, is stated as follows.

$$\mathcal{N}_k(S) = \{S' : d(S, S') \leq k\}$$

In this work, the initial solution S for the VNSTS procedure has been generated using the Sequential Forward Selection (SFS) and the Sequential Backward Elimination (SBE) algorithms, which are executed one after the other until no improvement is achieved. We proposed to initialization methods to develop the VNSTS, obtaining the hybrid metaheuristics called VNSTS+ and VNSTS, respectively. The first one selects the solution provided by running the SFS and the SBE algorithms one after the other until no improvement in achieved. The second one selects a solution generated at random, which has the maximum number of features between the number of features of the SFS-SBE solution and the 10% of the total number of features.

The Sequential Forward Selection (SFS) and the Sequential Backward Elimination (SBE) algorithms are stated as follows.

Sequential Forward Selection (*SFS*)

1. Initialize the set of features $S = \emptyset$;
2. For each feature $X_j \notin S$, calculate $f(S \bigcup \{X_j\})$. Let j^* be such that $f(S \bigcup \{X_j*\}) = max_j\{f(S \bigcup \{X_j\})\}$ and $S^* = S \bigcup \{X_j*\}$. If $f(S^*) > f(S)$, take $S = S \bigcup \{X_j*\}$ and repeat step 2. Otherwise, stop.

Sequential Backward Elimination (*SBE*)

1. Initialize the set of features $S = X_j : j = 1, ..., d$;

2. For each feature $X_j \in S$, calculate $f(S \backslash \{X_j\})$. Let j^* be such that $f(S \backslash \{X_j*\}) = max_j\{f(S \backslash \{X_j\})\}$ and $S^* = S \backslash \{X_j*\}$. If $f(S^*) > f(S)$, take $S = S \backslash \{X_j*\}$ and repeat step 2. Otherwise, stop.

3.2 Shaking procedure

This procedure generates a solution S' at random from the k^{th} neighborhood of S ($S' \in \mathcal{N}_k(S)$). Let $X_{i(1)}, ..., X_{i(k)}$ be k features in the solution S chosen at random and $Y_{i(1)}, ..., Y_{i(k)}$ k features out of the solution S also chosen at random. The solution S' is then stated in the following way.

$$S' = S \setminus \{X_{i(1)}, ..., X_{i(k)}\} \bigcup \{Y_{i(1)}, ..., Y_{i(k)}\}$$

In this step, the features $\{X_{i(1)}, ..., X_{i(k)}\} \bigcup \{Y_{i(1)}, ..., Y_{i(k)}\}$ involved in the shaking procedure are added to a tabu list, whose elements cannot be used in the shaking procedure in the next iteration. An iteration is defined as the performance of the shaking procedure and the local search.

Since in each iteration there are $2k$ tabu features, which cannot be used for shaking purposes, the stopping condition is met when the number of features to be exchanged, k, is greater than the minimum between the number of non-tabu features in the solution and the number of non-tabu features out of the solution.

3.3 Improvement method

The improvement method applied to the solution S' in this metaheuristic is the execution of the Sequential Forward Selection and the Sequential Backward Elimination algorithms one after the other until no improvement is achieved. The local optimum reached by running the SFS and SBB is compared with the best solution obtained by the procedure changing then either to the $(k+1)^{th}$ neighborhood structure if there has not been improvement or to the first structure otherwise.

4 Computational Results

The objective of the computational experiments is to show the performance of the hybrid metaheuristic (VNSTS) in searching for a reduced set of features with high accuracy. Firstly, we execute the hybrid procedures VNSTS and VNSTS+ explained above. Then these procedures are compared with a Genetic Algorithm using two standard classifiers (IB1 and Naive Bayes). The data showed a superiority of the VNSTS strategies over the genetic algorithm in reducing the number of features used for the classification task. Moreover, the computational experience carried out corroborates that the comparisons of the VNSTS strategies and the genetic algorithm is similar when using any of these classifiers, although Naive Bayes provides a higher reduction of the number of features.

Although the optimal selection of parameters is still an open problem on Genetic Algorithms, guided by the recommendations of Bäck [4], the probability of crossover is set to 1.0 and the mutation probability to $1/d$ being d the number of variables of the domain (these values are so common in the literature). Fitness proportionate selection is used to select individuals for crossover. The population size is set to 1000 and the new population is formed by the best members from both the old population and the offspring. The criterion for halting the genetic search is the following: stop when in a sampled new population of solutions no individual is found with an evaluation function value that improves the best individual found in the previous generation. Thus the best solution of the previous population is returned as the result of the genetic search. These parameters were already used in feature subset selection by Inza et al. (2001) [15].

The data sets considered in our computational experiments were obtained from the UCI repository [20], from which full documentation about all data sets can be obtained. We chose them taking into account their size and use in machine learning research. The selected data sets have more than 200 instances because small data sets can motivate overfitting. An induction algorithm overfits the data set if it models the training examples too accurately and its predictions are poor. Table 1 summarizes the characteristics of the chosen data sets. The first two columns correspond to the name of the data sets as it appears in the UCI repository and the identifier (Id) used in forthcoming tables. The intermediate three columns show the total number of features, the number of nominal features and the number of (numerical) linear features. Finally, the last two columns summarize the number of instances and classes in the data set.

Table 1: Characteristics of the data sets

DataBase	Id	Features			Instances	Classes
		All	Nom	Lin		
Anneal	anneal	38	29	9	798	5
Audiology	audiology	70	70	0	226	24
Horse Colic	colic	21	14	7	368	2
Credit Approval	credit-a	15	9	6	690	2
Glass	glass	9	0	10	214	7
Heart (Cleveland)	heart-c	13	7	6	303	2
Heart (Disease)	heart-h	13	7	6	294	2
Ionosphere	ionosphere	34	0	34	351	2
Sonar	sonar	60	0	60	208	2

In order to analyze the effectiveness of the hybrid metaheuristics proposed in this work, we solved several problems with both hybrid algorithms and with the genetic algorithm using the IB1 and Naive Bayes classifiers. Table 2 shows the results obtained with VNSTS, VNSTS+ and Genetic Algorithm using the classifier IB1. For these three procedures we show the accuracy percentages, the standard deviations over 10 runs. In addition, for the hybrid strategies we report the p-values obtained using the statistical test explained below. This test compares each hybrid procedure with the genetic algorithm.

As said before, we use the 5×2 cross-validation method to measure the accuracy percentage of the resulting subset of features selected by the algorithms. With the purpose of performing the comparisons between methods we used the 5×2 F test, which performs 5 replications of the 2-fold cross validation. In each replication, the data set is divided into two equal-sized sets. We apply the F-test proposed by Alpaydin in 1999 [1] that consists in the following.

The value $p_i^{(j)}$ is the difference between the error rates of the two classifications on fold $j = 1, 2$ of replication $i = 1, ..., 5$. The average on replication i is $\overline{p}_i = (p_i^{(1)} + p_i^{(2)})/2$ and the estimated variance is $s_i^2 = (p_i^{(1)} - \overline{p}_i)^2 + (p_i^{(2)} - \overline{p}_i)^2$. Under the null hypothesis that there is not difference between the classifier procedures $p_i^{(j)}$ is the difference between tho identically distributed proportions.

Ignoring the dependence between the proportions, $p_i^{(j)}$ can be treated as approximately normal distributed with zero mean and unknown variance σ^2 (see Dietterich 1998 [7]). Then $p_i^{(j)}/\sigma$ is approximately unit normal and $(p_i^{(j)}/\sigma)^2$ can be treated as a chi-square with one degree of freedom. Ignoring now that the s_i^2 are not independent, the statistics $M = (1/\sigma^2)\sum_{i=1}^{5} s_i^2$ and $N = (1/\sigma^2)\sum_{i=1}^{5}\sum_{j=1}^{2}(p_i^{(j)})^2$ are approximately chi-squared distributed variables with 5 and 10 degrees of freedom, respectively. Therefore, assuming that they are independent, $F = (N/10)/(M/5) \sim F_{10,5}$. We use the approximated p-values of this statistic for testing the equality of the percentages of the classification methods in each data set. Note that, due to the approximations made, the true p-value would be even better when the hypothesis is rejected.

Table 2: Accuracy and standard deviations using IB1

	IB1							
	VNSTS			VNSTS+			GA	
Id.	f	σ	p-val	f	σ	p-val	f	σ
anneal	96.57	±1.76	0.53	96.43	±1.50	0.78	96.37	±2.40
audiology	66.28	±4.94	0.41	65.04	±6.25	0.57	69.03	±4.60
colic	80.60	±1.52	0.23	81.09	±2.92	0.12	83.48	±2.98
credit-a	80.96	±2.33	0.75	80.03	±5.47	0.67	80.58	±3.00
glass	66.16	±3.38	0.07	67.94	±5.41	0.56	67.57	±4.01
heart-c	74.85	±3.67	0.43	72.22	±8.62	0.58	75.71	±4.18
heart-h	74.56	±4.84	0.72	73.54	±9.41	0.53	75.24	±5.07
ionosphere	87.57	±2.55	0.54	88.03	±2.70	0.77	87.98	±2.30
sonar	78.56	±5.34	0.53	77.60	±4.10	0.41	80.77	±3.48

Table 3 reports the number of features and standard deviations obtained for the three procedures using the classifier IB1.

Table 3: Number of features and standard deviations using IB1

	IB1					
	VNSTS		VNSTS+		GA	
Id.	f	σ	f	σ	f	σ
anneal	9.3	±0.8	8.9	±1.0	15.2	±2.5
audiology	12.5	±3.5	11.8	±3.1	31.0	±4.4
colic	8.2	±3.4	6.2	±2.6	7.4	±2.4
credit-a	5.9	±2.1	4.0	±2.1	5.2	±1.7
glass	4.9	±1.1	4.4	±1.1	4.8	±0.6
heart-c	6.2	±0.9	5.8	±2.5	6.9	±1.9
heart-h	6.1	±2.5	3.4	±1.5	4.8	±1.9
ionosphere	7.8	±2.7	5.5	±2.4	8.5	±2.3
sonar	13.3	±3.1	10.5	±2.8	20.0	±4.1

The results reported in Table 2 show that there are not significant differences in the accuracy percentages between the hybrid metaheuristics and the genetic algorithm using the classifier IB1. However, the results summarized in Table 3 show that the hybrid metaheuristics proposed in this work provide a higher reduction in the set of features than the genetic algorithm. Moreover, the metaheuristic VNSTS+, in which the initial solution is obtained running SFS and SBE until no improvement is achieved, provides the smallest numbers of features.

Although the computational CPU times have not been reported in this work because are not considered a crucial subject in learning, the computational times provided by the hybrid metaheuristics is a level of magnitude smaller than the provided by the genetic algorithm.

The accuracy percentages obtained using the classifier Naive Bayes reported in Tables 4 and 5 are similar to the accuracy percentages obtained using IB1. However, the Naive Bayes classifier reduces more the number of features selected to perform the classification.

Table 4: Accuracy and standard deviations using Naive Bayes

	Naive-Bayes							
	VNS-TS			VNS-TS+			GA	
Id.	f	σ	p-val	f	σ	p-val	f	σ
anneal	90.51	±2.00	0.38	88.13	±1.31	0.19	92.18	±2.51
audiology	69.38	±4.50	0.64	69.11	±4.67	0.61	69.73	±2.20
colic	83.75	±2.41	0.65	83.37	±2.71	0.53	83.70	±1.76
credit-a	85.51	±1.34	0.58	84.78	±1.41	0.55	85.18	±1.34
glass	58.90	±6.67	0.53	57.29	±7.00	0.30	59.44	±6.28
heart-c	80.86	±2.88	0.53	77.56	±4.01	0.39	80.14	±2.87
heart-h	81.48	±2.33	0.13	81.09	±2.64	0.52	81.36	±3.03
ionosphere	90.20	±3.36	0.33	89.86	±3.55	0.33	91.45	±2.52
sonar	70.67	±4.40	0.30	68.94	±5.50	0.73	69.33	±4.76

Table 5: Number of features and standard deviations using Naive Bayes

	Naive-Bayes					
	VNS-TS		VNS-TS+		GA	
Id.	f	σ	f	σ	f	σ
anneal	7.6	±1.4	3.2	±2.7	13.3	±2.8
audiology	10.0	±3.0	6.6	±2.6	14.7	±2.2
colic	3.9	±1.4	4.5	±1.6	5.8	1.23
credit-a	5.2	±0.9	3.8	±2.6	5.3	±1.4
glass	3.8	±0.6	3.6	±1.1	4.0	±0.7
heart-c	5.4	±1.7	3.8	±2.7	5.5	±1.8
heart-h	4.5	±0.9	3.1	±1.5	4.4	±0.8
ionosphere	7.4	±2.6	5.8	±2.3	11.1	±2.6
sonar	4.9	±1.6	3.8	±2.1	15.2	±5.1

5 Conclusions

In this paper, we propose two hybrid procedures based on the variable neighborhood search and tabu search metaheuristics to solve the Feature Subset Selection Problem. These two hybrid algorithms are obtained by combining the characteristics of both metaheuristics and generating initial solutions for the variable neighborhood search in two different ways.

The obtained computational results corroborate the effectiveness of our hybrid procedures when compared to a genetic algorithm. They both get accuracy percentages similar to the genetic algorithm using the classifiers IB1 and Naive Bayes. However, the hybrid metaheuristics proposed in this work provide a higher reduction in the set of features than the genetic algorithm. In addition, the metaheuristic VNSTS+ provides the smallest numbers of features. Finally, the Naive Bayes classifier reduces more the number of features selected to perform the classification.

References

[1] E. Alpaydin, Combined 5 × 2cv f test for comparing supervised classification learning algorithms, Neural Computation 11 (1999) 1885–1892.

[2] D. W. Aha, D. K. amd M. K. Albert, Instanced-based learning algorithms, Machine Learning 6 (1991) 37–66.

[3] J. R. Anderson, M. Matessa, Explorations of an incremental, bayesian algorithm for categorization, Machine Learning 9 (1992) 275–308.

[4] T. Bäck. *Evolutionary Algorithms is Theory and Practice.* Oxford University Press, 1996.

[5] V. Campos, F. Glover, M. Laguna, R. Martí, An experimental evaluation of a scatter search for the linear ordering problem, Journal of Global Optimization 21 (2001) 397–414.

[6] P. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, 1982.

[7] T. G. Dietterich, Approximate statistical test for comparing supervised classification learning algorithms, Neural Computation 10 (7) (1998) 1895–1923.

[8] F. Ferri, V. Kadirkamanathan, J. Kittler, Feature subset search using genetic algorithm, in: IEE/IEEE Workshop on Natural Algorithms in Signal Processing, IEE Press, 1993, p. Essex.

[9] F. García-López, B. Melián-Batista, J. Moreno-Pérez, J. M. Moreno-Vega, Parallelization of the scatter search for the p-median problem, Parallel Computing 29 (2003) 575–589.

[10] Glover, F., Laguna, M. *Tabu Search*, Kluwer Academic Publishers, 1997.

[11] L. Hyafil, R. L. Rivest, Constructing optimal binary decision trees is np-complete, Information Processing Letters 5 (1) (1976) 15–17.

[12] Hansen, P., Mladenovic, N.: Variable neighborhood search. *Computers & Operations Reserach.* (1997) 24, 1097–1100.

[13] Hansen, P., Mladenovic, N.: Variable neighborhood search: Principles and applications. *European Journal of Operational Research.* (2001) 130, 449–467.

[14] I. Inza, P. Larrañaga, R. Etxeberria, B. Sierra, Feature subset selection by bayesian networks based optimization, Artificial Intelligence 123 (2000) 157–184.

[15] I. Inza, M. Merino, P. Larraaga, J. Quiroga, B. Sierra, M. Girala. Feature subset selection by genetic algorithms and estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS. Artificial Intelligence in Medicine, (2001) 23/2, 187-205.

[16] K. Kira, L. Rendell, The feature selection problem: Traditional methods and a new algorithm, in: In Tenth National Conference Conference on Artificial Intelligence (AAAI-92), MIT, 1992, pp. 129–134.

[17] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1-2) (1997) 273–324.

[18] M. Laguna, R. Martí, Scatter Search: Metodology and Implementations in C, Kluwer Academic Press, 2003.

[19] T. Mitchell, Machine Learning, Series in Computer Science, McGraw-Hill, 1997.

[20] P. M. Murphy, D. W. Aha, Uci repository of machine learning.
URL `http://www.ics.uci.edu/ mlearn/MLRepository.html`

[21] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[22] C.R. Reeves, J.E. Rowe *Genetic Algorithms: Principles and Perspectives*, Kluwer, 2002.

[23] W. Siedlicki, J. Sklansky, A note on genetic algorithm for large-scale feature selection, Pattern Recognition Letters 10 (1989) 335–347.

[24] H. Vafaie, K. D. Jong, Robust feature selection algorithms, in: Proceedings of the 5th IEEE International Conference on Tools for Artificial Intelligence, IEE Press, 1993, pp. 356–363.

[25] D. R. Wilson, T. R. Matinez, Improved heterogeneous distance functions, Journal of Artificial Intelligence Research 6 (1997) 1–34.

[26] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2000.

[27] J. Yang, V. Honavar, Feature Subset Selection using a Genetic Algorithm. Genetic Programming 1997: Proceeding of the Second Annual Conference, Morgan Kaufmann, 1997.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Technology through the projects TIC2002-04242-C03-01 (70% of which are FEDER founds) and TIC2002-10886E.