Information Sciences 326 (2016) 102–118

Contents lists available at ScienceDirect

Information Sciences

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

High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach

Miguel García-Torres[a,*], Francisco Gómez-Vela[a], Belén Melián-Batista[b], J. Marcos Moreno-Vega[b]

[a] Área de Lenguajes y Sistemas Informáticos, Universidad Pablo de Olavide, Ctra de Utrera, km. 1, Sevilla 41013, Spain
[b] Dpto. de Ingeniería Informática y de Sistemas, Universidad de La Laguna, La Laguna 38271, Spain

ARTICLE INFO

Article history:
Received 30 October 2014
Revised 15 July 2015
Accepted 17 July 2015
Available online 26 July 2015

Keywords:
Feature selection
High dimensionality
Metaheuristic
Feature grouping

ABSTRACT

In recent years, advances in technology have led to increasingly high-dimensional datasets. This increase of dimensionality along with the presence of irrelevant and redundant features make the feature selection process challenging with respect to efficiency and effectiveness. In this context, approximate algorithms are typically applied since they provide good solutions in a reasonable time. On the other hand, feature grouping has arisen as a powerful approach to reduce dimensionality in high-dimensional data. Recently, some authors have focused their attention on developing methods that combine feature grouping and feature selection to improve the model. In this paper, we propose a feature selection strategy that utilizes feature grouping to increase the effectiveness of the search. As feature selection strategy, we propose a Variable Neighborhood Search (VNS) metaheuristic. Then, we propose to group the input space into subsets of features by using the concept of Markov blankets. To the best of our knowledge, this is the first time in which the Markov blanket is used for grouping features. We test the performance of VNS by conducting experiments on several high-dimensional datasets from two different domains: microarray and text mining. We compare VNS with popular and competitive techniques. Results show that VNS is a competitive strategy capable of finding a small size of features with similar predictive power than that obtained with other algorithms used in this study.

1. Introduction

Feature selection for classification has become an increasingly important research area within machine learning and pattern recognition [42,44,45,59] due to rapid advances in data collection and storage technologies. These advances have allowed organizations from science and industry to create large, high-dimensional, complex and heterogeneous datasets that represent a new challenge to the existing methods in the feature selection field.

In high-dimensional spaces, in addition to the curse of dimensionality, the learning task suffers from the fact that usually not all the features have the same discriminative power. Moreover, as the number of dimensions becomes larger, not only the complexity of the datasets increases, but also the number of non informative features with respect to the class concept may increase, because of irrelevancy and redundancy [71]. In this context, feature selection plays a critical role for removing such

[*] Corresponding author. Tel.: +34 954977366.
E-mail addresses: mgarciat@upo.es (M. García-Torres), fgomez@upo.es (F. Gómez-Vela), mbmelian@ull.es (B. Melián-Batista), jmmoreno@ull.es (J.M. Moreno-Vega).

http://dx.doi.org/10.1016/j.ins.2015.07.041

features and may yield some of the following benefits: (i) reduction in the cost of acquisition of the data, (ii) improvement of the comprehensibility of the final classification model, (iii) a faster induction of the final classification model and (iv) an improvement in classification accuracy.

Classically, feature selection in classification tasks is defined as the process that seeks the minimal size of relevant features such that the classification error is optimized. A relevant feature is neither irrelevant nor redundant to the target concept; an irrelevant feature does not affect the target concept in any way, and a redundant feature does not add anything new to the target concept [13]. In order to identify the optimal subset of relevant features, different criteria have been proposed to evaluate the goodness of feature subsets. Feature subset selection strategies are essentially divided into wrapper, filter and embedded methods [9,23]. Wrappers use the learner as a black box to score the subsets of features according to their predictive power. Thus, the quality of feature subsets for classification is defined with respect to the induction algorithms. The main advantage is that they include the interaction between feature subset and model selection, and have the ability to take into account feature dependencies. However, they have a higher risk of overfitting than filters and are computationally expensive [9]. Filter approaches select subsets of features as a preprocessing step, and so assess each subset according to intrinsic properties of the data. The advantages of these methods are that they are computationally fast, so that they easily scale to high-dimensional datasets [57]. A disadvantage is that they ignore the interaction with the classifier, which may lead to worse classification performance. Since they are independent of the learning algorithm, feature selection needs to be performed only once for a given training dataset. In contrast to the filter and wrapper techniques, the embedded methods cannot separate the learning and the feature selection since the structure of the class of functions under consideration play a crucial role. In this approach, the search for an optimal subset of features is done during the induction of the classifier. The main advantage of these methods is that they combine the interaction with the classification model such as the wrapper methods. However, they are far less computationally intensive than wrappers [28,58,75].

Selecting the most relevant features is usually suboptimal for building the model due to redundancy [23]. Despite the recent achievements carried out in the field of feature selection, feature relevance and redundancy are still two challenging issues in the field. Researchers firstly focused on identifying relevant features [3,5,8,31,36,73]. Then they also focused on redundancy [18,52,53,69,77], especially in high dimensional data [19]. Furthermore, the number of possible feature subsets grows exponentially with the number of features and many problems related to feature selection have been shown to be $\mathcal{NP}-hard$ [6]. For all these reasons, finding the optimal subset is usually intractable [35] even for a moderate number of features d. Therefore, approximate algorithms are typically applied since they provide satisfactory solutions in a reasonable time (see, for example, [20,26,41]). Even if the obtained solution is suboptimal and there is no guarantee of the distance between such solution and the optimal one, in general, they provide satisfactory solutions in a reasonable computational time.

The idea of feature clustering or feature grouping is a powerful approach for reducing the dimensionality. Moreover, the grouping of features is highly beneficial in learning with high-dimensional data. It can reduce the variance of the estimator [60], improve the stability of feature selection [32], and also helps to reduce the complexity of the model. As far as we know, it has been applied to text mining [2,50,54,64] and microarray [4,15,43,72,74] domains since the late 90s. For finding feature groups, some approaches use learning algorithms like self-organizing map [62], K-means [74], or a reminiscent of it [17], logistic regression [16], etc. Other techniques make use of information–theory measures [37,65], graph theory [65], kernel density estimation [76], and regularization techniques [68].

Approaches based on regularization techniques are worth mentioning. They are important embedded methods that attract increasing attention due to their good performance. These methods introduce additional constraints into the objective function. In effect, the model fits the data by minimizing the coefficients. Hence, features with coefficients that are close to 0 are then eliminated [49]. Some representative methods based on regularization techniques are: (a) the Lasso Regularization [67] based on l_1-norm, (b) Adaptive Lasso [78], which was proposed to improve the performance of the Lasso proposal, (c) Bridge regularization [29] and (d) Elastic net regularization [79] that is a mixture of bridge regularization (see [66] for more details).

Recently, some works focus their efforts on grouping correlated features. This approach produces feature selection results in the form of a set of feature groups, each consisting of features relevant to the class but highly correlated to each other, instead of the traditional form of a single subset of features. The main motivation of this approach lies on the key observation that in high-dimensional data, relevant features are highly correlated so that we can generate groups of correlated features that are resistant to the variations of the sample size. Such set of predictive feature groups, not only generalize well, but also provides additional informative group structure for expert domains to further investigate. Recently, two group-based feature selection frameworks were proposed to improve the robustness by identifying groups of correlated features [47,76]. In [76] the authors introduce the idea of Dense Feature Groups (DFG) based on Kernel Density Estimation (KDE). KDE is a popular non-parametric method for estimating probabilistic density functions and it is applied to estimate the density of the features. Therefore, DFG is composed of features which are close to the same density peak. This framework is motivated by two main observations in the sample space. Firstly, the dense core regions (peak regions), measured by probabilistic density estimation, are stable with respect to sampling of the dimensions (samples). Secondly, the features near the core region are highly correlated to each other, and thus should have similar relevance scores w.r.t. some class labels, assuming that the class labels are locally consistent. Under this framework, an algorithm named DRAGS (Dense Relevant Attribute Group Selector) was proposed, which finds a number of dense feature groups and evaluates the relevance of each group based on the average relevance of features in each group. A novel framework, called CGS (Consensus Group Stable Feature Selection) was proposed in [47]. This proposal, identifies consensus feature groups by subsampling training samples. In order to do this, the proposed approach approximates intrinsic feature groups by a set of consensus feature groups aggregated from multiple sets from ensemble learning so that it uses the idea of DFG to generate groups on each sample.

Another strategy, that is based on the same idea of grouping correlated features, is fast clustering-based feature selection algorithm (FAST). It works in two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature, which is strongly related to target classes, is selected from each cluster to form a subset of features. Features in different clusters are relatively independent. Hence, the clustering-based strategy of FAST has a high probability of producing a subset of useful and independent features.

In this work, we propose to tackle the feature selection problem by reducing the input space. Such reduction is performed by grouping the space search into subsets of correlated features called predominant groups. Each predominant group is composed of one predominant feature [77] X along with all redundant features for which X forms a Markov blanket. To the best of our knowledge, this is the first time in which the Markov blanket is used for grouping features. Then we use the predominant groups to design a Variable Neighborhood Search (VNS) strategy to handle high-dimensional datasets. The computational results show the effectiveness of our proposal versus competitive feature selection algorithms from the literature.

The contributions of this paper are summarized in the following items: (i) introduce the idea of predominant group to reduce the space search, (ii) propose a greedy algorithm (GreedyPGG) to generate the predominant groups, and (iii) develop a new VNS metaheuristic-based strategy under this framework to address the feature selection problem in high-dimensional scenarios.

The rest of the paper is organized as follows. In Section 2 we introduce the feature selection problem and the concepts used to describe the GreedyPGG strategy. Then we describe, in Section 3, the proposed VNS metaheuristic that uses the GreedyPGG algorithm for tackling the feature selection problem on high-dimensional datasets. The main characteristics of the different datasets used in this work are presented in Section 4. Section 5 contains an empirical study of the proposed framework, and finally the conclusions of this work with some possible extensions are presented in Section 6.

2. Feature selection

In this section, we provide the basic concepts of feature selection. Given $\mathcal{E}$ a set of n examples characterized by the pair $(\mathbf{x}_i, y_i)$, where each $\mathbf{x}_i \in \mathcal{X}$ is an instance described by a vector of d features $\mathcal{X} = (X_1, \ldots, X_d)$, and $y_i \in \mathcal{Y}$ is the known class label of $\mathbf{x}_i$, the aim of classification is to learn a function $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$.

In general, not all the features are equally useful for classification purposes. Therefore, removing some of them may improve the predictive model $\mathcal{C}$. In this context, the objective of feature selection is to find the subset of features $S \subseteq \mathcal{X}$ with which $\mathcal{C}$ achieves the lowest error rate. In order to find such subset, we define a quality measure $J(.)$ so that the associated optimization problem consists of finding the subset of features S that optimizes $J(S)$.

In feature selection, algorithms have been traditionally focused on finding a highly discriminating power set of features for minimizing the classification error rate. Several works [3,8,31,36] have made an effort for defining the different feature types according to their contribution to the meaning of the class concept. In this context, feature relevance has arisen as a measure of the amount of relevant information that a feature may contain about the class in classification tasks.

A feature is considered irrelevant if it contains no information about the class and therefore it is not necessary at all for the predictive task. Removing this type of features may improve the predictive model as well as the speed of the learning algorithm. In contrast, relevant features are those that embody information about the class concept. However, for minimizing the error rate it may not be necessary to select all relevant features, but only the subset with the most predictive power. Furthermore, such subset of features may not be unique due to redundancey.

Redundancey is generally defined in terms of feature correlation and it is widely accepted that two features are redundant to each other if their values are correlated. However, linear correlations may not be able to detect non-linear dependencies between features. Therefore, in [77] the authors proposed to use, as non-linear correlation, the Symmetrical Uncertainty (SU) [24] measure which is based on entropy and is defined as follows:

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right], \tag{1}$$

where

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)), \tag{2}$$

represents the entropy and measures the uncertainty about the values of X,

$$H(X|Y) = -\sum_j P(y_j)\sum_i P(x_i|y_j)\log_2(P(x_i|y_j)), \tag{3}$$

is the conditional entropy and measures the uncertainty about the value of X given the value of Y, and

$$IG(X|Y) = H(X) - H(X|Y) \tag{4}$$

is the Information Gain (IG) and measures the reduction in uncertainty about the value of X given the value of Y. IG can be used as a correlation measure since it is a symmetrical measure but it is biased in favor of variables with more values and such values have to be normalized to ensure that the values have the same scale, are comparable and have the same effect. SU overcomes these drawbacks and takes values in [0, 1]. A value of 1 indicates that knowing the values of either feature completely predicts the values of the other; a value of 0 indicates that X and Y are independent. Thus, SU can be used as a correlation measure between features.

Feature redundancy can be determined using the Markov blankets [36] concept. Formally, it is defined as follows:

Definition 1 (Markov blanket)**.** Given a feature X_i, let $M_i \subset X$ $(X_i \notin M_i)$, M_i is said to be a Markov blanket for X_i iff $P(\mathcal{X} - M_i - \{X_i\}, \mathcal{Y}|X_i, M_i) = P(\mathcal{X} - M_i - \{X_i\}, \mathcal{Y}|M_i)$.

The Markov blanket M_i of a given target variable $\mathcal{Y}$ is the theoretically optimal set of features to predict the values of $\mathcal{Y}$. Obtaining the exact Markov blanket for a feature requires an exhaustive enumeration of feature subsets which makes it computationally unacceptable even in not very high dimensional data. Therefore, the authors in [77] define heuristically the approximate Markov blankets by using SU correlation measure between features.

Definition 2 (Approximate Markov blanket)**.** Given two features X_i and X_j $(i \neq j)$ so that $SU(X_j, \mathcal{Y}) \geq SU(X_i, \mathcal{Y})$, then X_j forms an approximate Markov blanket for X_i if $SU(X_i, X_j) \geq SU(X_i, \mathcal{Y})$.

This definition is based on pairwise comparison and assumes that a feature X_i with larger values of SU with the class contains more information about the class than a feature X_j with smaller value. The approximate Markov blanket is computed by comparing the correlation between the features X_i and X_j, and the SU value of X_i and the class. If the correlation between such features is larger, then X_j forms an approximate Markov blanket.

To guarantee that a redundant feature removed in a given step will still find a Markov blanket in any later phase when another redundant feature is removed, they introduce the concept of predominant feature.

Definition 3 (Predominant feature)**.** Given a set of features $S \subset \mathcal{X}$, a feature X_i is a predominant feature of S if it does not have any approximate Markov blanket in S.

Predominant features will not be removed at any stage. We use the Fast Correlation-Based Filter (FCBF) algorithm [77] to find the set of predominant features.

FCBF consists of two stages: obtaining the subset of relevant features and selecting the predominant features from it. In a practical sense, a feature X is relevant if $SU(X, Y)$ exceeds a predefined minimum threshold. Moreover, a relevant feature X_i is predominant if there is no other relevant feature X_j such that X_j is an approximate Markov blanket for X_i.

FCBF proceeds as follows. First, the subset of relevant features is obtained and ordered, in descending order, according to their symmetrical uncertainty with respect to the class. Let $X(1)$ be the first feature in this set. By definition, $X(1)$ is a predominant feature. Next, for each one of the remaining relevant features $X(i)$ $(i > 1)$, it is checked whether $X(1)$ is an approximate Markov blanket for $X(i)$. If this is true, $X(i)$ is removed from the set of relevant features. The above process is repeated while predominant features that can be considered remain on the list.

2.1. Feature grouping

For a given dataset, several subsets of features may generate predictive models with similar performance. Depending on the search strategy of the algorithm or the bias of the sample, some features may be favored. The key observation is that, in general, features from different solutions with similar performance are highly correlated. Detecting all correlated subsets is a combinatorial problem since it requires to take into account 2^d subsets. Because of the intractability of this problem, we relax the condition by assuming that by grouping correlated features we can focus the search on features that do not contain similar information. Therefore, we can split the feature space into groups $\{\mathcal{G}_i\}$ so that each group $\mathcal{G}_i$ is composed of redundant features.

Predominant features are relevant and not redundant; so that we can generate each group $\mathcal{G}_i$ by considering a predominant feature X_i along with all redundant features X_j, $j \neq i$ for which X_i forms a Markov blanket. We introduce the predominant group as follows.

Definition 4 (Predominant group)**.** A subset of features $\mathcal{G} \subset \mathcal{X}$ is a predominant group if $\mathcal{G} \neq \{\emptyset\}$ and $\exists!\ X_i\ \in \mathcal{G} : X_i$ is a predominant feature.

With this definition, the number of predominant groups coincides with the number of predominant features. Although different strategies can be considered to generate the predominant groups, we can use the greedy approach used by the Fast Correlation based Filter (FCBF) [77]. As we can see in Fig. 1, it first lists the features X_i in descending order of $SU(X_i, \mathcal{Y})$ value. Then all features with SU measure lower than or equal to a threshold δ are removed. Since such threshold is not easy to set, only uncorrelated features $(\delta = 0)$ are usually removed. In the next step, not only predominant features are identified, but also predominant groups are detected. The first feature in the list is a predominant feature since it does not have an approximate Markov blanket. Then, we check if the first predominant feature forms a Markov blanket for any of the features in the list. If so, the redundant feature is added to its predominant group. Once all features have been checked, the algorithm selects as next predominant feature, the next relevant one with the highest SU that do not belong to any predominant group. Moreover, it is checked if the rest of the features in the list are redundant with respect to the predominant feature. This process is repeated until no predominant feature is found. The above procedure is called Greedy Predominant Groups Generator (GreedyPGG.)

Notice that different predominant features may form a Markov blanket for the same redundant and so belong to many predominant groups. However, a predominant feature can only belong to a predominant group.

```
Procedure Greedy Predominant Groups Generator
begin
  $\mathcal{L} \leftarrow \emptyset$.
  $\mathcal{G} \leftarrow \emptyset$.
  foreach $X_i \in \mathcal{X}$
      if $(SU(X_i, \mathcal{Y}) > \delta)$
          $\mathcal{L} \leftarrow X_i$
      end
  end
  $\mathcal{L}_{su} \leftarrow sort(\mathcal{L}) : SU(X_i, \mathcal{Y}) \geq SU(X_j, \mathcal{Y})\ \forall\ i < j$
  $t \leftarrow\ 0$
  foreach $X_i \in \mathcal{L}_{su} : X_i \notin \mathcal{G}$
      $\mathcal{G}_t \leftarrow X_i$
      foreach $X_j \in \mathcal{L}_{su} : X_j \notin \mathcal{G}$
          if $(SU(X_i, X_j) \geq SU(X_j, Y)\ )$
              $\mathcal{G}_t \leftarrow X_j$
          end
      end
      $\mathcal{G} \leftarrow \mathcal{G}_t$
      $k \leftarrow t + 1$
  end
end
```

Fig. 1. Greedy strategy to generate the predominant groups.

3. VNS applied to the feature selection problem

Feature selection can be seen as the process of finding the best subset of features in *X*. Thus, the problem can be formulated as

$$optimize\{J(S) : S \in 2^X\}, \tag{5}$$

where $S \subset X$ is any subset of features, $2^X = \{S : S \subset X\}$ is the set of feasible solutions or search space of the problem (5) and $J(S)$ is the objective function value used to measure the quality of *S*. In this paper, we use the feature subset evaluation function of the Correlation based Feature Selection (*CFS*) algorithm [24] as objective value:

$$J(S) = \frac{m \cdot \overline{SU}(S, Y)}{\sqrt{m + m(m-1) \cdot \overline{SU}(S, S)}},$$

where

$$\overline{SU}(S, Y) = \frac{1}{m} \cdot \sum_{X_i \in S} SU(X_i, Y),$$

and

$$\overline{SU}(S, S) = \frac{2}{m(m-1)} \cdot \sum_{X_i \in S} \sum_{\substack{X_j \in S \\ X_i \neq X_j}} SU(X_i, X_j).$$

The search space consists of 2^d solutions so that exhaustive search is discarded for solving moderate-size instances. Instead, heuristic algorithms that perform an intelligent search in the solution space can be used.

Neighborhood search algorithms are a wide class of heuristic methods which traverse the solution space by moving from one solution to another in its neighborhood. This process is repeated until some stopping criterion is fulfilled. A solution S' belongs to the neighborhood of the solution *S* if S' can be obtained by lightly modifying *S*. Let $\mathcal{N}(S)$ be the neighborhood of solution *S*.

Definition 5. The subset of features *S* is a local optimum of the problem (5), given a neighborhood structure $\mathcal{N}$, if

$$\forall S' \in \mathcal{N}(S) : J(S) \geq J(S').$$

Definition 6. The subset of features S^* is a global optimum of the problem (5) if

$$\forall S' \in 2^X : J(S^*) \geq J(S').$$

Our objective is to find the optimum subset of features with respect to the objective function $J(\ \cdot\ )$.

```
Procedure Variable Neighborhood Search
begin
  Initialize the set of neighborhood structures $\mathcal{N}_k, k = 1, \ldots, k_{max}$;
  Generate the initial solution $S$;
     repeat
        $k \leftarrow 1$;
        repeat
           ShakeMethod ($S$, $S'$);
           ImprovementMethod ($S'$, $S''$);
           if $J(S'') \geq J(S)$ then {
              $S \leftarrow S''$;
              $k \leftarrow 1$;
              }
           else
              $k \leftarrow k + 1$;
        until $(k = k_{max})$
     until $(StoppingCriterion)$
end
```

Fig. 2. Variable Neighborhood Search pseudocode.

The main disadvantage of neighborhood search algorithms is that, in general, they provide locally optimal solutions. Therefore, they usually incorporate some mechanisms that allow them to escape from local optima.

To escape from local optima, Variable Neighborhood Search (VNS) [27] systematically changes the neighborhood structure during the search process. This approach is based on the following observations:

1. A global optimum is a local optimum with respect to any neighborhood structure. Therefore, any neighborhood structure can be used to find the global optimum.
2. A local optimum with respect to one neighborhood structure is not necessarily a local optimum with respect to another neighborhood structure. Thus, a search which changes the neighborhood structure is able to escape from local optima.

VNS was first proposed in [27] for solving combinatorial optimization and global optimization problems by systematically changing the neighborhood of the current solution during the search. Given $\mathcal{N}_k$, $k = 1, \ldots, k_{max}$ a set of finite number of neighborhood structures and $\mathcal{N}_k(S)$ the set of solutions in the kth neighborhood of a solution S, the metaheuristic works as follows. It starts generating an initial solution S. Then it randomly generates, in the shaking step, a new solution S' within the first neighborhood $\mathcal{N}_1(S)$ of S. It continues applying an improvement method to S' obtaining the improved solution S''. If S'' improves the current best solution S, then the search is refocused around $S \leftarrow S''$, and it begins again with the first neighborhood. If S'' does not improve S, the neighborhood is changed to the next one. Finally, if in the last neighborhood $\mathcal{N}_{k_{max}}(S)$ there is no improvement in S, then the search begins from $\mathcal{N}_1(S)$ until a stopping criterion is reached. The pseudocode of the VNS is shown in Fig. 2.

Given $S = \{x_1, \ldots, x_r\}$ and $S' = \{x'_1, \ldots, x'_t\}$ two solutions (sets of features), the distance $d(S, S')$ between them is defined as follows:

$$d(S, S') = |\{\{x_1, \ldots, x_r\} \cup \{x'_1, \ldots, x'_t\}\} \setminus \{\{x_1, \ldots, x_r\} \cap \{x'_1, \ldots, x'_t\}\}|$$

Then the kth neighborhood of a solution S, $\mathcal{N}_k(S)$, is defined as

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

Two variants of VNS for the feature selection problem are described below. In both of them, the greedy strategy Sequential Forward Search (SFS) Fig. 3 is used as local search.

3.1. Random Basic Variable Neighborhood Search (RVNS)

The first VNS variant proposed to solve the feature selection problem, that will be referred to as Random Basic Variable Neighborhood Search (RVNS), consists of using randomness for guiding the search. We will see that despite the simplicity of this proposal, it achieves quite good results. The characteristics of RVNS are explained below.

The initial solution S is generated at random, so that each feature X_i is included in S according to a fixed probability p. Increasing values of p favors solutions with bigger subsets of features. Solution S is shaken to generate a new one S' from the kth neighborhood of S ($S' \in \mathcal{N}(S)$). The kth neighborhood of S consists of those solutions that can be reached from S by exchanging at most k features in S with k features out of S. Then, a local search which uses S' as initial solution is performed.

```
Procedure Sequential Forward Search
begin
  $S \leftarrow \{\emptyset\}$
  repeat
      foreach $X_j \notin S$;
          $J_j \leftarrow J(S \cup \{X_j\})$;
      Let $j' \leftarrow arg\max\{J_j\}$;
      $S' \leftarrow S \cup \{X_{j'}\}$;
      if $J(S') > J(S)$ then
          $S \leftarrow S'$;
          $J(S) \leftarrow J(S')$;
  until $(J(S') \leq J(S) \; || \; |S'| == d)$
end
```

Fig. 3. Sequential Forward Search pseudocode.

```
Procedure Shaking method
begin
 1: $i \leftarrow 1$
 2: repeat
 3:     $\mathcal{F} \leftarrow \emptyset$;
 4:     $j \leftarrow random : j \in [0, d)$;
 5:     if $(j \in [0, s))$
 6:         $S' \leftarrow S \setminus \{X_{(j)}\}$;
 7:         $\mathcal{G} \leftarrow \{\mathcal{G}_t\} : X_{(j)} \in \mathcal{G}_t$;
 8:         $\mathcal{F} \leftarrow \{X_r\} : X_r \in \mathcal{G} \wedge X_r \neq X_{(j)}$
 9:         $X_k \leftarrow random : X_k \in \mathcal{F}$
10:         if $(X_k \notin \mathcal{S})$
11:             $S' \leftarrow S' \cup \{X_k\}$
12:         end
13:     else
14:         $X_k \leftarrow random : X_k \notin S$;
15:         $S' \leftarrow S \cup \{X_i\}$;
16: until $(i = k)$
17: return $S'$;
end
```

Fig. 4. Shaking method pseudocode of the PGVNS strategy.

3.2. Predominant Group based Variable Neighborhood Search (PGVNS)

In order to adapt VNS to high-dimensional datasets, this method, called Predominant Group based VNS for Feature Selection (PGVNS), generates the set of predominant groups by means of the Predominant group space generation strategy. The initial solution S is thus composed of the predominant features found and then S is shaken. Fig. 4 shows the pseudocode of the shaking method. As we can see, it shakes the solution as many times as the current neighborhood index, k. For each iteration, it first selects at random a value $j \in [0, d)$. If j is lower than the size s of the solution S (see lines $6 - 11$), this method swaps or removes a feature from S. Otherwise, it adds a feature to S (see lines 14 and 15). If $j < s$, the feature $X_{(j)}$ at such position is removed from S. Then, we obtain the set of predominant groups $\mathcal{G} = \{\mathcal{G}_t\}$ to which $X_{(j)}$ belongs. A new set $\mathcal{F}$ is created with all features X_r from $\mathcal{G}$ except $X_{(j)}$ and a feature X_k is randomly selected from $\mathcal{F}$. If X_k does not belong to S', then it is added to it. Otherwise, it leaves S' as it is. If $j \geq s$ a new feature X_k is randomly added from outside the solution S. Finally, a greedy local search is fed with the shaked solution S'.

4. Data

Two application domains where feature selection has been largely applied in recent years are gene selection in microarray datasets [18,30,46] and text mining [12,33,70]. In the first case, features represent gene expression coefficients corresponding to the aboundance of mRNA in a sample. The sample size is small; usually, fewer than 100 examples are available. However, the number of features is very large usually ranging from 5000 up to 60, 000. In text mining, documents are represented by words containing their frequency counts in the documents. In this field, datasets are large having usually a sampling size from 1000

Table 1
Summary of synthetic datasets.

Dataset	#Features	#Groups	#Rel. features	#Samples
D_{1k}	1000	100 (size 10 ± 5)	10	{100, 200, 500, 1000}
D_{5k}	5000	250 (size 20 ± 5)		

up to $800,000$ documents. These datasets are also high-dimensional since vocabularies of hundreds of thousands of words are common. However, an initial pruning may reduce the dimension considerably.

In this work we have used data from such domains with different complexity to assess various aspects of the strategies under study. The characteristics of the datasets are given below.

4.1. Synthetic datasets

In contrast to real-world data, synthetic datasets provide a controlled environment for analyzing the strengths and limitations of the proposed strategy to generate the predominant groups (GreedyPGG) since relevant, redundant and irrelevant features are known. The datasets considered are D_{1k} and D_{5k} [47], two high-dimensional synthetic datasets that consist of 1000 samples obtained from the same distribution $P(\mathcal{X}, \mathcal{Y})$. A summary of these datasets is provided in Table 1. In D_{1k}, the features set $\mathcal{X}$ consists of 1000 features, including 100 mutually independent features, $X_1, X_2, \ldots, X_{100}$, and a number of (10 ± 5) highly correlated features to each of these 100 features. Within each correlated group, the Pearson correlation of each feature pair is within (0.5, 1), and the average pairwise correlation is below 0.75. The balanced binary class label $\mathcal{Y}$ is decided based on $X_1, X_2, \ldots, X_{10}$ using a linear function of equal weight to these 10 truly relevant features. D_{5k} was created following the same procedure for generating the data, but with higher dimensionality and larger feature groups. In this case, it consists of 5000 features, 250 of which are mutually independent. The number of highly correlated features to each of these 250 features is 20 ± 5. Finally, we study the sample size dependency of the GreedyPGG by considering the sample sizes {100, 200, 500, 1000}.

4.2. High-dimensional data

The analysis of the optimal combination of parameter values of VNS strategies and the comparison of the feature selection approaches were done using high-dimensional datasets from the microarray and text-mining domains.

Microarray datasets are challenging problems to the data mining community since modern medical equipments and diagnostic techniques generate voluminous data that may contain systemic and human errors [40]. Furthermore, these datasets are characterized by high-dimensional data with a small sample size and usually contain many redundant and irrelevant features.

Text mining datasets, however, are not only high-dimensional but also large. Moreover, textual documents usually contain much irrelevant and noisy information that increases the complexity of the classification task. Although the dimensionality of the text representation is very large, the number of words in the different documents may vary widely and the underlying data is sparce.

The summary of the datasets is shown in Table 2. The first column indicates the data domain followed by the dataset id. Next two columns show the number of instances and features, respectively. Then, the distinct labels followed by the number of instances associated to each label. Last column presents the reference to the original work.

5. Computational results

This section presents the experiments done to evaluate the GreedyPGG and VNS algorithms and discusses the obtained results. The aims of this section can be summarized in the following items:

- Evaluate the proposed GreedyPGG strategy on the synthetic datasets.
- Study the best combination of parameter values on some high-dimensional datasets randomly selected.
- Test the VNS strategies on high dimensional datasets. Results are compared with standard feature selection algorithms.

In order to assess model quality on the high-dimensional datasets, we use cross-validation. The number of folds is set to $f = 5$ because cross-validation consumes a great deal of resources. Furthermore, lower values of f produce more pessimistic estimates, while higher values produce more optimistic results. Although the *true* generalization error is not usually known, and so it is not possible to determine whether a given estimate is an overestimate or underestimate, cross-validation is suitable for model comparison purposes.

As performance measures, we use the classification error averaged over the folds. To measure the quality of the used algorithms, we report the average number of features selected by each strategy and the robustness of each algorithm.

The robustness or stability [34,38,39] of feature subset selection strategies is a topic of recent interest that aims to measure the sensitivity to variations of a feature selection algorithm in the dataset. This issue is important specially in high dimensional domains since it enhances the confidence in the analysis of the results.

In order to quantify the robustness of a feature selection method, a measure of similarity between two sets of features is needed. In this work, we use the Jaccard index [56], which is defined as the size of the intersection divided by the size of the

Table 2
Characteristics of the high-dimensional datasets.

Domain	Dataset	Id	#Inst.	#Feat.	Labels	#Inst./label	Ref.
Microarray	Colon	cln	62	2000	Normal/tumor	22/40	[1]
	Breast	bcg	168	2905	Good/poor	111/57	[22]
	CNS	cns	60	7129	Survival/failure	21/39	[55]
	Lymphoma	lym	77	7129	Diffuse/follicular	58/19	[61]
	Lung	lng	181	12,533	MPM/ADCA	31/150	[21]
	Prostate	prt	102	12,600	Tumor/not	52/50	[63]
	Breast	bcc	118	22,215	Positive/negative	75/43	[10]
	Breast/colon	bco	104	22,283	Breast/colon	62/42	[11]
	Crohn	cro	127	22,283	Normal/colitis/crohn	42/26/59	[7]
Text-mining	Alt	alt	4157	2112	Relevant/not	1425/2732	[51]
	Structure	str	3548	2368		927/2621	
	Disease	dis	3237	2376		631/2606	
	Function	fct	3907	2708		818/3089	
	Subcell	scl	7977	4031		1502/6475	
	Acq	acq	12,897	7495	Yes/no	2369/10, 528	[48]
	Money-fx	mfx		7757		717/12, 180	
	Corn	crn		8301		237/12, 660	
	Earn	ern		9499		3964/8933	
	Ship	shp		9930		286/12, 611	
	Grain	grn		12,473		582/12, 315	
	Crude	crd		14,465		578/12, 319	

Table 3
Summary of the results achieved by GreedyPGG on D_{1k} and D_{5k} datasets.

Dataset		#Samples			
		100	200	500	1000
D_{1k}	$\# \mathcal{G}$	9	10	10	10
	$\# \mathcal{G}^*$	9	10	10	10
	$\lvert\mathcal{G}^*\rvert$	10 ± 8.5	9 ± 5.1	9 ± 2.8	9 ± 2.9
D_{5k}	$\# \mathcal{G}$	10	11	13	13
	$\# \mathcal{G}^*$	10	10	10	10
	$\lvert\mathcal{G}^*\rvert$	19.3 ± 15.4	19.3 ± 12.5	19.1 ± 3.0	19.1 ± 3.2

union of the sets. Let A and B be subsets of features such that $A, B \subseteq \mathcal{X}$. The Jaccard index between such subsets $\mathcal{I}_J(A, B)$ is defined as follows:

$$\mathcal{I}_J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{6}$$

Given a set of solutions $\mathcal{S} = \{S_1, \ldots, S_m\}$, the approach for estimating the stability, $\Sigma(S)$, among this set of solutions consists of averaging the pairwise $\mathcal{I}_J(\cdot, \cdot)$ (Σ) similarities

$$\Sigma(\mathcal{S}) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \mathcal{I}_J(S_i, S_j).$$

Higher values correspond to more stable subsets.

Our experiments were performed with naive Bayes and linear Support Vector Machines (SVM) due to their popularity and good results achieved. All experiments have been developed using `Weka` [25], and the source code is available upon request.

In order to support the obtained conclusions, we applied statistical tests following the guidelines proposed by Demšar [14]. For the parameter tuning, we have no control parameter values and so, we applied the Friedman test; a non-parametric test equivalent to the repeated-measures ANOVA. If the null-hypothesis is rejected, we proceed with the Nemenyi post-hoc test. For the comparison between PGVNS and the other studied strategies we have used the Wilcoxon signed-ranks test, which is a non-parametric alternative to the paired t-test.

5.1. Analysis of GreedyPGG

Results of GreedyPGG are summarized in Table 3. First column shows the datasets. Each column, from 3 to 6, corresponds to a different size of the problem. For each dataset, we present, in the first row, the number of predominant groups, $\# \mathcal{G}$. The second row shows the number of non-empty predominant groups, $\# \mathcal{G}^*$, and the last one, the average size of the non-empty predominant groups, $|\mathcal{G}^*|$.

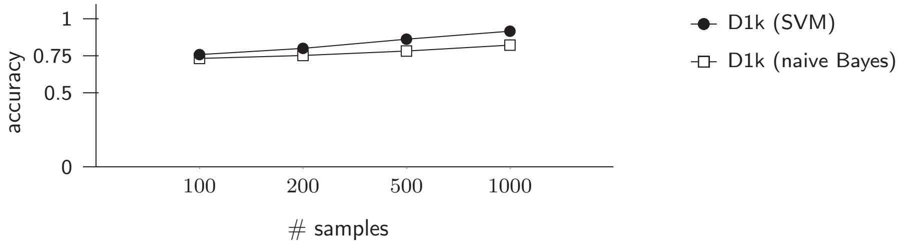

Fig. 5. Comparison of naive Bayes and SVM on D_{1k} dataset.

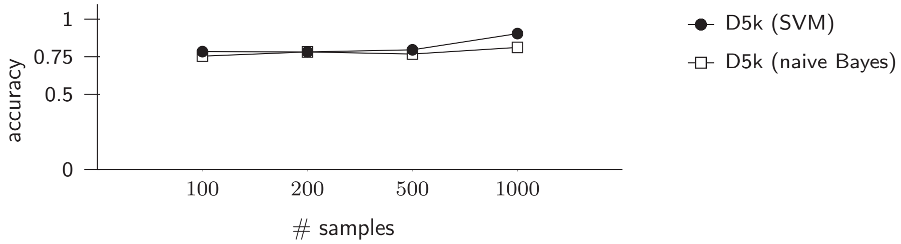

Fig. 6. Comparison of naive Bayes and SVM on D_{5k} dataset.

Table 4
Parameter values considered for parameter tuning on proposed VNS strategies.

VNS	$k_p(\%)$	p_i
RVNS	{1, 5, 10, 25}	{0.25, 0.5, 0.75}
PGVNS		–

GreedyPGG finds, in all cases except in D_{1k} with 100 samples, the relevant features and assigns them to a predominant group. In D_{1k}, it finds as many predominant groups as relevant features except for the smaller case that fails to find a relevant one. The average size of each predominant group $\mathcal{G}$ is 9 features. With D_{5k}, the number of predominant groups increases when the sample becomes larger. However, if we remove those $\mathcal{G}$ that are empty, GreedyPGG finds the 10 relevant features. In this case, the average size of the predominant groups is 19 features approximately.

We also tested the accuracy of the subset of predominant features on naive Bayes and SVM classifiers. For such purpose, we used and independent test set of 500 samples randomly generated from the same distribution as the training set. Fig. 5 presents the performance of naive Bayes and SVM on D_{1k} dataset. In both cases, the larger the sample is, it can get the higher accuracy with each predictive model. Moreover, SVM achieves better accuracy in all samples. It is worth mentioning that, in small samples, the accuracy achieves values higher than 70% with both classifiers.

The performance on D_{5k} is shown in Fig. 6. The accuracy is somewhat lower, but as in the previous case, SVM achieves a higher predictive model than naive Bayes. In contrast, accuracy is quite similar in samples of size 200 and 500, while it improves considerably with a sample size of 1000.

Therefore, GreedyPGG is a heuristic strategy that is able to find the set of relevant features and assign it to a predominant group. Moreover, it also associates a set of features correlated with the predominant feature. However, it may also detect false predominant features, but in this case, the size of the predominant group is 0 or very small when compared to other predominant groups.

5.2. Parameter tuning

In this section, we study the combination of parameters that maximizes the classification accuracy on six datasets from microarray and text mining domains. In case of no statistical significant differences, we analyze the size of the solutions found. The different values of the parameters considered for both VNS strategies are shown in Table 4. For RVNS, we changed the probability p_i of randomly selecting a feature in the initial solution. Lower values of p_i generate smaller solutions. For both strategies, we change the k_{max} value. Due to the variety of the datasets sizes, we set k_{max} values according to percentages of the size. We denote this parameter as k_p. Bigger values of k_p increase the number of iterations and so the size of the search space.

The accuracy with both classifiers and the size of the solutions are shown in Table 5 for each VNS approach. For a given parameter combination, the first row presents the average results and the following one the standard deviation. The VNS approach and its parameters are given in the first three columns. The next fourteen columns show the accuracy with naive Bayes and SVM, respectively. Finally, we can see the size of the solutions in the last seven columns. For each VNS approach and parameter

Table 5
Mean accuracy values with their respective standard deviation obtained with naive Bayes and SVM classifiers after applying the algorithms RVNS and PGVNS with different combinations of the parameter values.

			Naive Bayes							SVM							#Features						
$\mathcal{A}$	p_i	k_p	lym	pro	col	alt	fct	mfx	Mean	lym	pro	col	alt	fct	mfx	mean	lym	pro	col	alt	fct	mfx	Mean
RVNS	0.25	1	84.42	84.33	75.77	75.99	77.53	93.72	81.96	87.00	84.43	66.15	75.99	79.11	94.91	81.27	27.60	17.80	10.60	1.00	16.20	9.60	13.80
			(7.17)	(7.99)	(11.83)	(0.79)	(1.37)	(0.55)		(4.74)	(5.97)	(2.81)	(0.79)	(0.42)	(0.22)		(5.32)	(2.49)	(1.82)	(0.00)	(5.40)	(1.82)	
		5	84.42	83.33	75.77	75.99	78.32	93.72	81.93	84.33	83.38	66.15	75.99	79.42	94.91	80.70	27.60	14.40	10.60	1.00	10.00	9.60	12.20
			(7.17)	(9.60)	(11.83)	(0.79)	(2.06)	(0.55)		(3.97)	(7.33)	(2.81)	(0.79)	(0.66)	(0.22)		(5.32)	(7.89)	(1.82)	(0.00)	(8.69)	(1.82)	
		10	84.42	83.33	75.77	75.99	78.32	93.72	81.93	85.67	85.29	66.15	75.99	79.37	94.91	81.23	27.60	14.40	10.60	1.00	10.00	9.60	12.20
			(7.17)	(9.60)	(11.83)	(0.79)	(2.06)	(0.55)		(3.20)	(7.00)	(2.81)	(0.79)	(0.73)	(0.22)		(5.32)	(7.89)	(1.82)	(0.00)	(8.69)	(1.82)	
		25	84.42	80.48	75.77	75.99	79.29	93.72	81.61	85.67	84.33	66.15	75.99	79.68	94.91	81.12	27.60	11.80	10.60	1.00	4.80	9.60	10.90
			(7.17)	(9.05)	(11.83)	(0.79)	(1.14)	(0.55)		(3.20)	(6.42)	(2.81)	(0.79)	(0.65)	(0.22)		(5.32)	(9.26)	(1.82)	(0.00)	(8.50)	(1.82)	
	0.50	1	83.25	91.29	75.77	86.38	75.43	93.71	84.31	83.00	87.24	69.49	86.36	78.78	95.56	83.41	44.60	25.80	13.80	2.00	47.20	15.80	24.87
			(8.11)	(8.64)	(8.09)	(1.37)	(1.36)	(0.51)		(7.66)	(5.50)	(9.60)	(1.38)	(0.97)	(0.34)		(2.30)	(5.36)	(4.38)	(0.00)	(3.90)	(2.17)	
		5	83.25	91.29	75.77	86.38	75.43	93.71	84.31	80.33	86.24	69.62	86.36	78.81	95.56	82.82	44.60	25.80	13.80	2.00	47.20	15.80	24.87
			(8.11)	(8.64)	(8.09)	(1.37)	(1.36)	(0.51)		(8.35)	(6.39)	(14.40)	(1.38)	(1.01)	(0.34)		(2.30)	(5.36)	(4.38)	(0.00)	(3.90)	(2.17)	
		10	83.25	91.29	75.77	86.38	75.43	93.71	84.31	83.00	86.24	71.28	86.36	78.86	95.59	83.55	44.60	25.80	13.80	2.00	47.20	15.80	24.87
			(8.11)	(8.64)	(8.09)	(1.37)	(1.36)	(0.51)		(7.66)	(6.39)	(14.46)	(1.38)	(0.96)	(0.38)		(2.30)	(5.36)	(4.38)	(0.00)	(3.90)	(2.17)	
		25	83.25	91.29	75.77	86.38	75.43	93.71	84.31	84.33	86.24	67.95	86.36	78.68	95.56	83.19	44.60	25.80	13.80	2.00	47.20	15.80	24.87
			(8.11)	(8.64)	(8.09)	(1.37)	(1.36)	(0.51)		(5.94)	(6.39)	(12.81)	(1.38)	(0.95)	(0.34)		(2.30)	(5.36)	(4.38)	(0.00)	(3.90)	(2.17)	
	0.75	1	88.33	92.24	72.44	86.38	76.30	93.99	84.95	85.58	88.24	70.90	86.36	78.58	96.53	84.36	52.40	20.60	19.60	2.00	60.60	26.00	30.20
			(5.25)	(8.79)	(7.61)	(1.37)	(1.06)	(0.34)		(5.79)	(4.42)	(4.79)	(1.38)	(1.16)	(0.13)		(4.77)	(3.58)	(8.32)	(0.00)	(9.18)	(2.55)	
		5	88.33	92.24	72.44	86.38	76.30	93.99	84.95	85.58	88.24	70.90	86.36	78.48	96.58	84.36	52.40	20.60	19.60	2.00	60.60	26.00	30.20
			(5.25)	(8.79)	(7.61)	(1.37)	(1.06)	(0.34)		(5.79)	(4.42)	(4.79)	(1.38)	(1.29)	(0.16)		(4.77)	(3.58)	(8.32)	(0.00)	(9.18)	(2.55)	
		10	88.33	92.24	72.44	86.38	76.30	93.99	84.95	86.92	88.29	70.90	86.36	78.55	96.53	84.59	44.60	25.80	13.80	2.00	47.20	15.80	24.87
			(5.25)	(8.79)	(7.61)	(1.37)	(1.06)	(0.34)		(6.78)	(7.37)	(4.79)	(1.38)	(0.99)	(0.13)		(4.77)	(3.58)	(8.32)	(0.00)	(9.18)	(2.55)	
		25	88.33	92.24	72.44	86.38	76.30	93.99	84.95	84.25	89.24	70.90	86.36	78.55	96.53	84.30	52.40	20.60	19.60	2.00	60.60	26.00	30.20
			(5.25)	(8.79)	(7.61)	(1.37)	(1.06)	(0.34)		(7.81)	(4.05)	(4.79)	(1.38)	(1.18)	(0.13)		(4.77)	(3.58)	(8.32)	(0.00)	(9.18)	(2.55)	
PGVNS	–	1	89.58	92.19	75.51	87.68	77.73	94.20	86.15	94.83	94.10	75.51	87.66	79.70	96.39	88.03	33.40	14.60	11.80	2.00	23.60	16.40	16.97
			(3.63)	(8.97)	(10.74)	(0.97)	(1.88)	(0.39)		(2.90)	(8.89)	(12.25)	(0.95)	(0.68)	(0.27)		(9.24)	(4.22)	(4.82)	(0.00)	(8.91)	(3.78)	
		5	90.92	90.24	75.90	88.93	76.73	93.77	86.08	90.83	91.19	77.18	88.55	79.40	96.70	87.31	47.60	20.60	20.60	3.00	68.60	27.80	31.37
			(5.88)	(11.55)	(5.06)	(1.21)	(0.92)	(0.22)		(3.81)	(5.33)	(11.27)	(1.15)	(0.72)	(0.26)		(4.88)	(4.51)	(8.02)	(0.00)	(17.78)	(3.63)	
		10	90.92	90.24	75.90	88.93	76.76	93.77	86.09	85.67	87.19	80.51	88.57	79.17	96.67	86.30	48.00	21.40	20.80	3.00	71.40	27.80	32.07
			(5.88)	(11.55)	(5.06)	(1.21)	(0.94)	(0.22)		(5.45)	(5.73)	(9.48)	(1.18)	(0.93)	(0.27)		(4.90)	(4.04)	(8.32)	(0.00)	(18.92)	(3.63)	
		25	90.92	90.24	75.90	88.93	76.71	93.77	86.08	86.83	92.19	82.18	88.55	79.01	96.67	87.57	48.00	22.40	21.60	3.00	71.20	27.80	32.33
			(5.88)	(11.55)	(5.06)	(1.21)	(0.91)	(0.22)		(6.88)	(5.52)	(6.79)	(1.15)	(1.12)	(0.27)		(4.90)	(5.50)	(7.80)	(0.00)	(18.75)	(3.63)	

Table 6
Mean accuracy values with their respective standard deviation obtained on microarray data with naive Bayes and SVM classifers after applying the algorithms PGVNS, RVNS, FAST, FCBF and CVNS.

	Naive Bayes					SVM					#Fatures				
Id	PGVNS	RVNS	FAST	FCBF	CVNS	PGVNS	RVNS	FAST	FCBF	CVNS	PGVNS	RVNS	FAST	FCBF	CVNS
cln	80.51	74.23	74.36	83.97	82.18	77.82	72.69	77.56	79.23	71.15	14.20	8.40	5.4	14.40	21.20
	(11.16)	(13.11)	14.88	(5.25)	(13.02)	(16.93)	(13.36)	(13.22)	(11.96)	(8.36)	(2.77)	(2.30)	(2.19)	(2.61)	(5.26)
bcg	73.87	72.05	70.82	70.29	74.44	72.05	73.19	73.80	75.61	75.03	58.40	34.00	15.8	32.40	73.20
	(12.46)	(8.40)	(6.78)	(10.27)	(10.63)	(9.82)	(4.87)	(3.32)	(7.28)	(5.64)	(5.32)	(7.68)	(2.17)	(5.13)	(4.02)
cns	71.67	61.67	55.00	63.33	65.00	60.00	60.00	58.33	61.67	60.00	33.60	22.40	9.00	33.00	40.80
	(12.64)	(13.94)	(7.45)	(12.64)	(14.91)	(10.87)	(9.13)	(10.21)	(9.50)	(12.36)	(6.88)	(6.35)	(3.61)	(9.41)	(7.46)
lym	92.42	88.42	89.83	88.42	91.00	86.00	90.83	87.17	91.08	84.58	34.40	25.60	16.6	40.80	47.60
	(10.26)	(8.06)	(12.12)	(9.34)	(7.21)	(11.94)	(7.50)	(10.03)	(7.06)	(9.28)	(10.53)	(7.23)	(3.65)	(9.04)	(7.30)
lng	97.22	97.79	98.89	99.44	97.78	98.33	97.22	98.89	99.44	98.33	9.00	13.60	142.8	100.40	10.40
	(2.78)	(2.32)	(1.52)	(1.24)	(3.04)	(1.52)	(1.96)	(1.52)	(1.24)	(1.52)	(5.52)	(6.35)	(11.63)	(24.34)	(8.05)
prt	99.42	99.16	99.48	99.42	99.42	98.12	97.41	99.48	99.35	78.77	1.20	1.40	304.00	201.60	2.00
	(0.53)	(0.49)	(0.18)	(0.58)	(0.87)	(1.03)	(2.06)	(0.29)	(0.56)	(43.53)	(0.45)	(0.55)	(7.65)	(40.38)	(1.41)
bcc	86.34	86.30	87.17	88.01	87.17	85.47	82.97	81.34	85.54	83.80	103.20	81.80	64.80	136.00	105.80
	(6.56)	(7.91)	(6.99)	(8.98)	(6.99)	(7.97)	(4.62)	(3.82)	(4.08)	(7.19)	(3.42)	(6.38)	(17.6)	(5.00)	(4.55)
Mean	87.32	84.35	82.22	85.90	85.28	83.87	82.17	82.37	85.69	79.99	33.85	26.92	79.77	75.82	43.00
bco	97.10	95.19	92.33	94.29	*na*	90.24	91.38	96.14	95.24	*na*	16.80	28.20	94.40	48.00	*na*
	(2.65)	(3.37)	(4.22)	(6.21)	–	(9.39)	(3.93)	(2.16)	(5.83)	–	(13.52)	(5.89)	(11.46)	(6.28)	–
cro	86.62	82.65	77.94	85.78	*na*	90.52	84.22	86.49	88.89	*na*	128.20	88.60	54.20	125.40	*na*
	(5.94)	(8.51)	(10.83)	(6.11)	–	(7.13)	(7.47)	(8.39)	(5.32)	–	(8.87)	(8.62)	(3.90)	(9.79)	–
Mean	87.24	84.16	82.87	85.88	–	84.61	82.40	84.36	86.04	–	44.33	33.78	78.56	81.33	–
p-Val	–	0.02	0.05	0.29	1.00	–	0.53	0.91	0.04	0.28	–	0.13	0.91	0.16	0.02

combination, the mean accuracy values on each classifier as well as the mean size of the solutions are presented in columns under the heading mean.

RVNS converges quickly with $p_i \geq 0.5$ reaching the same solutions in almost all datasets. If $p_i = 0.25$ the fast convergence is reached only in some datasets. For the other ones, results obtained for a given dataset with different values of k_p are very similar one to each other. The differences found in RVNS are not statistically significant neither with naive Bayes nor SVM. For PGVNS, results vary slightly when increasing k_p. As in previous case, these differences are not statistically significant.

When comparing the size of the solutions with RVNS, higher values of p_i yield to larger subsets of features. As it was explained with accuracy values, due to the convergence to a local optimum for $p_i \geq 0.5$, the solutions do not change in almost all cases. However, for $p_i = 0.25$, larger values of k_p give smaller subsets and these differences were found to be statistically significant by the Friedman test with a p-Value of $6 \cdot 10^{-9}$. Results with the parameter combination of $p_i = 0.25$ and $k_p = 5, 10, 25$ were found statistically significant, by the Nemeyi test, with respect to those obtained with $p_i = 0.75$. The significance level was of 90%, 90% and 95% with p-Values of 0.08, 0.08 and 0.04 respectively. For PGVNS, the Friedman tests found significant differences to a level of 99% (p-Value of $2 \cdot 10^{-4}$). According to the Nemenyi post-hoc test, parameter $k_p = 1$ differs significantly to $k_p = \{10, 25\}$ to a level of 90% with p-Values of 0.09 and 0.06 respectively.

For RVNS, all results with $p_i = 0.25$ are very similar. Therefore, we select the parameter $k_p = 5$ since it requires fewer number of iterations. For PGVNS, we choose $k_p = 1\%$.

5.3. Experiments on high-dimensional datasets

This section compares the performance of the proposed PGVNS algorithm with the following strategies: RVNS, FCBF, FAST and CVNS, a VNS that uses CGS instead of GreedyPGG to find the predominant groups. As explained in Section 3.1, RVNS is our first attempt to adapt VNS to large scale feature selection. The performance of FCBF is a good reference since the GreedyPGG strategy is based on it. Therefore, it is interesting to see if our proposed method outperforms it or not. FAST has been included in this study because it uses similar concepts to those introduced in this work. Finally, we include CVNS since it will allow us to compare how different the results achieved by VNS are when changing the method to generate predominant groups. In the discussion about the results, we will highlight differences in the performance of the classifier higher than several percentage points.

5.3.1. Microarray data

Results on microarray datasets are presented in Table 6. First column correspond to naive Bayes accuracy followed by SVM. In both cases, models were obtained with the feature subsets found by the feature selection algorithm. Finally, we can see the number of features found by each strategy. For CVNS, we have no results on bco and cro datasets due to lack of memory resulting in missing values, that are represented by the symbol *na* (not available). The next to last row shows the mean values achieved over all datasets for all algorithms except CVNS. To compare the mean values of all algorithms, we also show the average over the first seven datasets. The last row contains the *p*-Values of the statistical test when comparing the output of each algorithm with PGVNS.

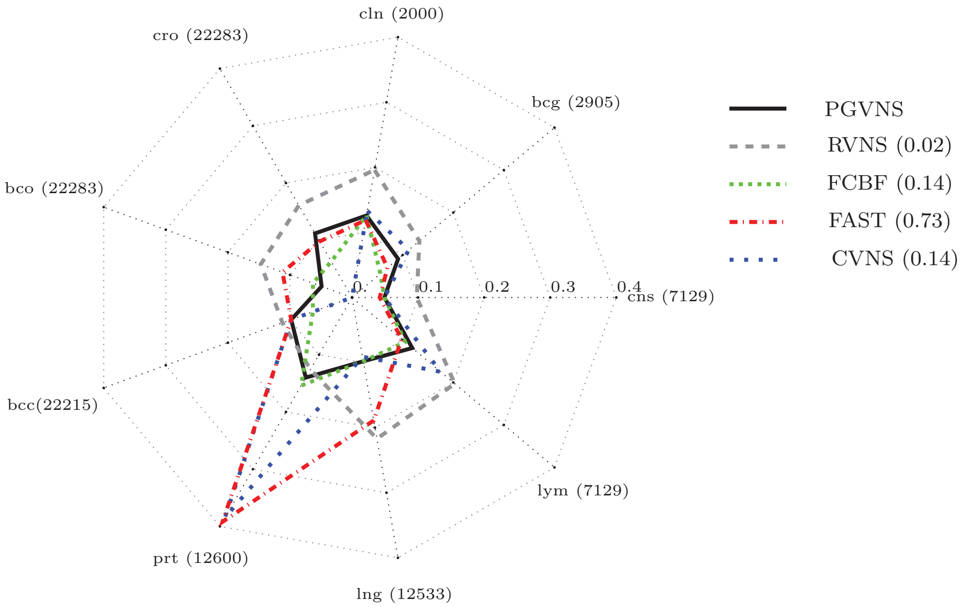

Fig. 7. Spider web diagram showing the stability index values obtained on microarray data with the strategies PGVNS, RVNS, FCBF, FAST and CVNS. We show, in brackets, the *p*-Values.

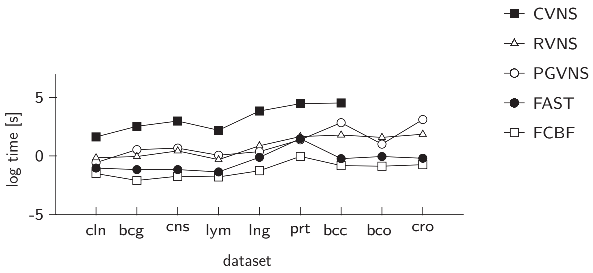

Fig. 8. Running time values obtained on microarray data with the strategies PGVNS, RVNS, FAST, FCBF and CVNS.

The results presented in Table 6 show that for naive Bayes PGVNS is the approach that obtains, on average, the best results in terms of accuracy (see row mean). We found statistically significant differences between PGVNS and RVNS and FAST to a level of 95% and 90%, respectively. RVNS degrades the classifier performance more than three percentage points on cln, cns, lym and cro, while FAST does so on cln, bcg, cns, bco and cro. When comparing with FCBF, we find the differences in the classifier performance on cln, bcg, cns, bco and cro. In all cases, except for cln, results are in favor of PGVNS. With CVNS, results are similar to PGVNS except for cns. With SVM, FCBF is the strategy that obtains, on average, the best accuracy followed by PGVNS, FAST, RVNS and CVNS. Results obtained with FCBF are slightly higher than PGVNS and achieve a performance difference higher than three percentage points on bcg, lym and bco in favor of FCBF. Differences between both algorithms are statistically significant at level of 0.05 (95%). Although results with the other strategies are not statistically different, for some datasets, we find variations higher than three percentage points in favor of PGVNS with RVNS on cln and cro, with FAST on bcc and cro and with CVNS on cln and prt. Finally, against PGVNS we can see that with RVNS, only on lym and with FAST on bco.

When comparing the reduction, there is no one strategy that reduces most in all problems. Only differences with CVNS are statistically significant to a level of 95%. However, on an average, RVNS is the strategy that reduces most followed by PGVNS, CVNS (taking into account only the first seven datasets), FAST and FCBF. In summary, it could be argued that PGVNS presents a good dimensionality reduction while maintaining a good performance of the classifier.

The stability index is shown in Fig. 7 with the *p*-Value in brackets. The stability value for bco and cro datasets with CVNS is set to 0 since such results are not available. RVNS is the algorithm that obtains more stable solution in all datasets except with prt (the *p*-Value of these differences are significant at level 0.05). CVNS obtains solutions slightly more stable than PGVNS in some datasets. CVNS and FAST obtain quite stable solutions while FCBF is the less robust strategy.

Fig. 8 shows the running time. CVNS is, by far, the slowest algorithm. In this sense, it differs several orders of magnitude with the other strategies presented in the study. FCBF is the fastest solution followed by FAST. RVNS and PGVNS present similar results for all datasets.

Table 7
Mean accuracy values with their respective standard deviation obtained on text mining data with naive Bayes and SVM classifers after applying the algorithms PGVNS, RVNS, FAST, FCBF and CVNS.

	Naive Bayes					SVM					#Features				
Id	PGVNS	RVNS	FAST	FCBF	CVNS	PGVNS	RVNS	FAST	FCBF	CVNS	PGVNS	RVNS	FAST	FCBF	CVNS
alt	87.68	86.38	84.96	84.07	88.91	87.66	86.36	88.12	88.09	88.50	2.00	2.00	14.60	29.00	3.00
	(0.60)	(0.74)	(0.75)	(0.83)	(0.42)	(0.57)	(0.71)	(0.74)	(0.73)	(0.49)	(0.00)	(0.00)	(0.55)	(2.65)	(0.00)
str	78.95	76.33	78.64	77.76	78.95	78.69	76.33	79.82	80.21	79.31	6.20	2.40	15.80	32.80	16.20
	(0.75)	(0.95)	(1.05)	(1.64)	(1.20)	(0.88)	(1.08)	(0.60)	(0.28)	(1.40)	(5.85)	(3.13)	(0.84)	(2.17)	(8.04)
dis	79.80	79.21	79.24	78.07	79.30	80.07	80.32	80.29	79.58	80.23	9.40	13.80	9.00	27.40	29.00
	(1.05)	(0.84)	(2.10)	(1.63)	(2.61)	(0.97)	(0.64)	(0.57)	(1.06)	(0.78)	(5.81)	(7.22)	(1.00)	(4.04)	(24.93)
fct	77.89	78.19	77.99	77.22	76.96	79.32	79.17	79.19	79.06	78.91	20.80	3.60	12.20	34.60	70.20
	(1.76)	(1.50)	(1.76)	(1.26)	(1.64)	(1.03)	(0.55)	(1.33)	(0.82)	(0.84)	(16.12)	(2.41)	(1.64)	(2.07)	(14.96)
scl	83.60	80.21	83.94	83.53	83.29	83.99	81.28	84.14	84.37	84.07	6.40	3.40	19.80	41.60	8.20
	(0.48)	(0.99)	(0.80)	(0.89)	(0.43)	(0.73)	(0.27)	(0.61)	(0.67)	(0.83)	(0.55)	(1.67)	(1.10)	(1.82)	(0.45)
acq	89.68	85.30	89.26	89.71	89.84	92.07	88.62	94.15	92.64	92.39	20.80	13.20	140.60	58.80	27.60
	(1.48)	(0.92)	(1.15)	(0.74)	(0.63)	(0.54)	(0.70)	(1.00)	(0.34)	(0.51)	(1.30)	(1.79)	(4.77)	(2.39)	(5.22)
mfx	93.90	93.56	88.68	94.01	93.66	96.18	95.23	96.20	95.99	96.53	17.00	8.00	88.60	54.00	27.40
	(0.25)	(0.47)	(0.48)	(0.72)	(0.33)	(0.16)	(0.43)	(0.29)	(0.30)	(0.14)	(3.67)	(0.00)	(2.61)	(2.00)	(5.55)
crn	99.65	98.45	94.46	98.63	99.65	99.65	98.76	99.43	99.70	99.65	2.00	3.00	66.80	22.60	2.00
	(0.09)	(0.66)	(0.93)	(0.47)	(0.09)	(0.09)	(0.36)	(0.32)	(0.10)	(0.09)	(0.00)	(1.87)	(7.82)	(1.67)	(0.00)
ern	92.43	85.33	92.74	94.74	92.43	92.29	85.33	94.39	95.08	92.29	3.00	1.00	163.20	77.80	3.00
	(0.36)	(0.76)	(0.50)	(0.25)	(0.36)	(0.38)	(0.76)	(0.78)	(0.30)	(0.38)	(0.00)	(0.00)	(4.92)	(4.97)	(0.00)
shp	98.89	97.46	95.23	98.77	98.85	98.95	97.98	98.79	99.12	98.93	9.00	3.40	79.00	40.00	10.20
	(0.14)	(0.46)	(0.56)	(0.35)	(0.14)	(0.24)	(0.15)	(0.07)	(0.30)	(0.25)	(0.71)	(1.34)	(6.04)	(1.22)	(1.10)
Mean	88.25	86.04	86.52	87.65	88.18	88.89	86.94	89.45	89.39	89.08	9.66	5.38	60.96	41.86	19.68
grn	98.71	96.83	91.46	98.43	*na*	98.86	96.99	98.13	99.22	*na*	4.40	2.40	110	34.00	*na*
	(0.14)	(0.66)	(0.54)	(0.05)	–	(0.27)	(0.60)	(0.27)	(0.20)	–	(0.89)	(0.89)	(9.03)	(2.65)	–
crd	97.40	95.32	88.25	97.68	*na*	97.49	96.66	97.53	97.84	*na*	3.80	2.00	137.2	64.20	*na*
	(0.21)	(0.50)	(0.59)	(0.48)	–	(0.27)	(0.20)	(0.26)	(0.36)	–	(0.45)	(0.00)	(3.49)	(7.56)	–
Mean	89.88	87.71	87.07	89.38	–	90.43	88.58	90.85	90.91	–	8.73	4.85	71.4	43.07	–
p-Val	–	$1e^{-3}$	$9e^{-3}$	0.18	0.45	–	$1e^{-3}$	0.23	0.08	0.14	–	0.03	$2e^{-3}$	$5e^{-4}$	0.01

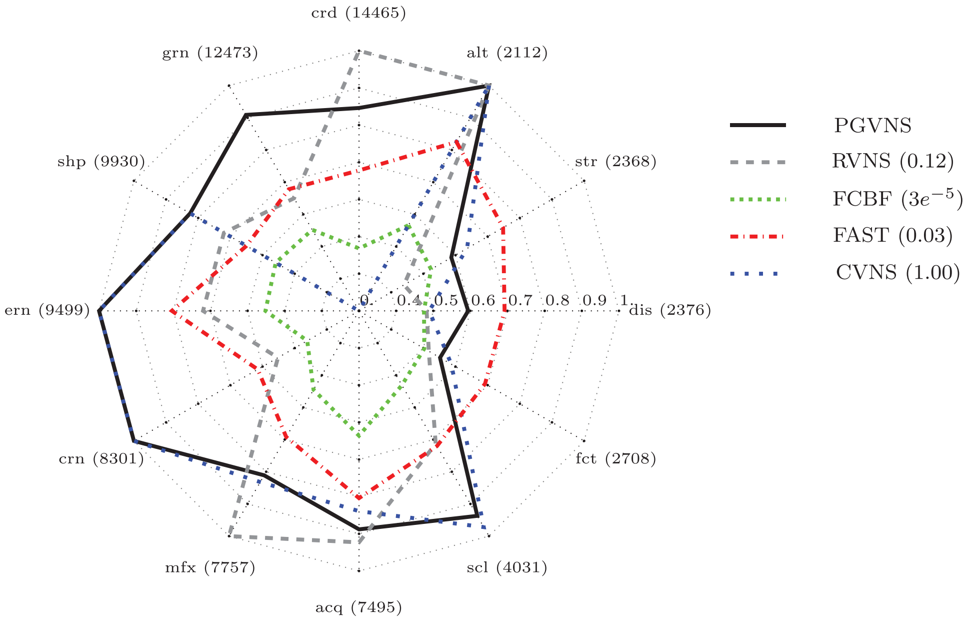

Fig. 9. Spider web diagram showing the stability index values obtained on text mining data with the strategies PGVNS, RVNS, FAST, FCBF and CVNS.

In conclusion, these results show that PGVNS is able to obtain, in a reasonable time, small and stable solutions without degrading the classifier performance. Therefore, PGVNS has very competitive performance in terms of accuracy, number of features, stability and running time.

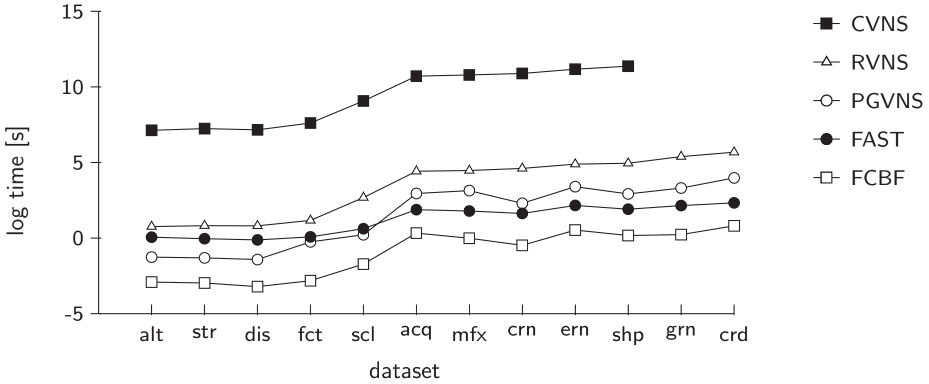

Fig. 10. Running time values obtained on text mining data with the strategies PGVNS, RVNS, FAST, FCBF and CVNS.

5.3.2. *Text mining data*

Results on text mining datasets are shown in Table 7. As it happened with microarray data, some results for grn and crd datasets are missing due to lack of memory. The first mean row gives the average values over the first 10 datasets while the second one includes the values of grn and crd. The last row shows the *p*-Values.

With naive Bayes, PGVNS is the strategy that obtains, on an average, the best performance followed by CVNS, FCBF, RVNS and FAST. PGVNS, compared to RVNS and FAST, achieves higher accuracy in most datasets and these results are statistically significant to a level of 99%. RVNS degrades the classifier on scl, acq and ern while FAST on mfx, crn, shp, grn and crd. In general, FCBF and FAST reach similar results to PGVNS in all datasets. Only in the case of alt with FCBF, the performance of the classifier degrades. With SVM, the average accuracy is similar in all methods except for RVNS that is slightly lower. Differences between PGVNS and RVNS and FCBF were found significant at a level of 0.01(99%) and 0.1(90%), respectively. In the case of RVNS, it degrades the performance of the classifier on acq and ern. In any other case, all strategies achieve very similar results.

The strategy that reduces most, on an average is RVNS followed by PGVNS, CVNS, FCBF and FAST. The differences in the reduction are statistically significant in all cases. With FAST and FCBF, the significance level is of 99% while with RVNS and CVNS of 95%.

Fig. 9 presents the comparison of the stability index. No available results are set to 0 as it is the case for grn and crd datasets with CVNS. The results show that PGVNS is, on average, the most stable strategy. Differences between PGVNS and FCBF and FAST are significant at 99% and 95% respectively. It is worth mentioning that FCBF is by far the least stable algorithm.

Finally, comparison in running time is shown in Fig. 10. FCBF is the fastest strategy. Then PGVNS is faster in the first five datasets and FAST in the rest ones. Finally, RVNS and CVNS are the slowest strategies. In general, CVNS is the strategy that consumes more resources in memory and in time.

These results are consistent with the results described in the previous section, where PGVNS obtained very competitive results in terms of dimensionality reduction, stability and running time without degrading the accuracy of both classifiers.

6. Conclusions and future work

In this work, we tackle the feature selection problem on high-dimensional data by grouping the input space. We develop a Variable Neighborhood Search that is capable of handling high-dimensional datasets (PGVNS). This strategy utilizes feature grouping to find a good solution. Based on the concepts of approximate Markov blanket and predominant feature, we introduce the idea of predominant group, and propose the heuristic strategy GreedyPGG for grouping the input space. For comparison purpose, we develop, as a first attempt to adapt the VNS to high-dimensional problems, a naive stochastic strategy (RVNS) that uses randomness to guide the search. Finally, we have conducted several experiments on synthetic and real datasets from microarray and text mining domains for testing the quality of the proposed strategy PGVNS. We have also compared the method with RVNS and three popular feature selection algorithms: FCBF, FAST and CVNS.

Experiments on synthetic datasets show that GreedyPGG is an efficient way of grouping the input space since it is able to find, for each predominant feature, the set of features correlated with it. However, more research has to be done to remove irrelevant features. This task is still challenging in the feature selection community since the relevance assessment depends on the dataset.

The results presented in this paper on real datasets show that RVNS is a naive strategy that achieves interesting results in stability and dimensionality reduction, but the classifier looses discriminatory capability since it degrades its performance in many cases. Therefore, this method is not suitable for the feature selection problem.

In contrast, PGVNS, compared to popular methods like FCBF, FAST and CVNS, has shown to be a new robust and competitive strategy. In this sense, it has obtained very good accuracy results in all datasets with both classifiers and also, very competitive performance in terms of the number of features, stability and running time for both performed experiments (specially on text mining datasets). Moreover, an interesting characteristic of this strategy is that, since it uses GreedyPGG to group features, it not only provides the subset of features found, but also the input space grouped.

Future work should focus on investigating a general framework to discard irrelevant features and improve the robustness of the proposed method, specially in microarray domains.

Acknowledgments

Belén Melián-Batista and J. Marcos Moreno-Vega acknowledge the financial support of the project TIN2012-32608 of the Spanish MINECO. Part of the computer time was provided by the Centro Informático Científico de Andalucía (CIC). We would like to thank to Dr. Guangtao Wang for providing the software to run the FastSearch algorithm.

References

[1] U. Alon, N. Barkai, D.A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 96 (12) (1999) 6745–6750.

[2] R. Bekkerman, R. El-Yaniv, N. Tishby, Y. Winter, Distributional word clusters vs. words for text categorization, J. Mach. Learn Res. 3 (2003) 1183–1208.

[3] D. Bell, H. Wang, A formalism for relevance and its application in feature subset selection, Mach Learn 41 (2) (2000) 175–195.

[4] A. Ben-Dor, R. Shamir, Z. Yakhini, Clustering gene expression patterns, J. Comput Biol 6 (1999) 281–297.

[5] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1–2) (1997) 245–271.

[6] A.L. Blum, R.L. Rivest, Training a 3-node neural networks is NP-complete, Neural Netw. 5 (1992) 117–127.

[7] M.E. Burczynski, R.L. Peterson, N.C. Twine, K.A. Zuberek, B.J. Brodeur, L. Casciotti, V. Maganti, P.S. Reddy, A. Strahs, F. Immermann, W. Spinelli, U. Schwertschlag, A.M. Slager, M.M. Cotreau, A.J. Dorner, Molecular classification of crohn's disease and ulcerative colitis patients using transcriptional profiles in peripheral blood mononuclear cells, J. Mol. Diagn. 8 (1) (2006) 51–61.

[8] R. Caruana, D. Freitag, How useful is relevance? in: Proceedings of Working Notes of the AAAI Fall Symposium on Relevance, 1994, pp. 25–29.

[9] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.

[10] K. Chin, S. DeVries, J. Fridlyand, P.T. Spellman, R. Roydasgupta, W.-L. Kuo, A. Lapuk, R.M. Neve, Z. Qian, T. Ryder, F. Chen, H. Feiler, T. Tokuyasu, C. Kingsley, S. Dairkee, Z. Meng, K. Chew, D. Pinkel, A. Jain, B.M. Ljung, L. Esserman, D.G. Albertson, F.M. Waldman, J.W. Gray, Genomic and transcriptional aberrations linked to breast cancer pathophysiologies, Cancer Cell 10 (6) (2006) 529–541.

[11] D. Chowdary, J. Lathrop, J. Skelton, K. Curtin, T. Briggs, Y. Zhang, J. Yu, Y. Wang, A. Mazumder, Prognostic gene expression signatures can be measured in tissues collected in RNAlater preservative, J. Mol. Diagn. 8 (1) (2006) 31–39.

[12] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, M.W. Mahoney, Feature selection methods for text classification, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 2007, pp. 230–239.

[13] M. Dash, H. Liu, Feature selection for classification, Intell. Data Anal. 1 (1997) 131–156.

[14] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[15] M. Dettling, P. Buhlmann, Supervised clustering of genes, Genome Biol. 3 (12) (2002) 0069.1–0069.15.

[16] M. Dettling, P. Buhlmann, Finding predictive gene groups from microarray data, J. Multivar. Anal. 90 (1) (2004) 106–131.

[17] I.S. Dhillon, S. Mallela, R. Kumar, A divisive information theoretic feature clustering algorithm for text classification, J. Mach. Learn. Res. 3 (2003) 1265–1287.

[18] C.H.Q. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinform. Comput. Biol. 3 (2) (2005) 185–206.

[19] J. Fan, J. Lv, A selective overview of variable selection in high dimensional feature space, Stat. Sin. 20 (2010) 101–148.

[20] M. García-Torres, R. Armañanzas, C. Bielza, P. Larrañaga, Comparison of metaheuristic strategies for peakbin selection in proteomic mass spectrometry data, Inf. Sci. 222 (2013) 229–246.

[21] G.J. Gordon, R.V. Jensen, L.L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, R. Bueno, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Res. 62 (17) (2002) 4963–4967.

[22] Gravier, Eleonore, G. Pierron, A. Vincent-Salomon, N. gruel, V. Raynal, A. Savignoni, Y. De Rycke, J.-Y. Pierga, C. Lucchesi, F. Reyal, A. Fourquet, S. Roman-Roman, F. Radvanyi, X. Sastre-Garau, B. Asselain, O. Delattre, A prognostic DNA signature for T1T2 node-negative breast cancer patients., Genes Chromosomes Cancer 49 (12) (2010) 1125.

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

[24] M.A. Hall, Correlation-based Feature Subset Selection for Machine Learning, University of Waikato, Hamilton, New Zealand, 1999 (Ph.D. thesis).

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

[26] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, Y. Liu, Feature subset selection by gravitational search algorithm optimization, Inf. Sci. 281 (2014) 128–146.

[27] P. Hansen, N. Mladenović, Variable neighborhood search, Comput. Oper. Res. 24 (1997) 1097–1100.

[28] H.H. Hsu, C.W. Hsieh, M.D. Lu, Hybrid feature selection by combining filters and wrappers, Expert Syst. Appl. 38 (7) (2011) 8144–8150.

[29] J. Huang, J.L. Horowitz, S. Ma, Asymptotic properties of bridge estimators in sparse high-dimensional regression models, Ann. Stat. 36 (2) (2008) 587–613.

[30] T. Jirapech-Umpai, S. Aitken, Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes, BMC Bioinform. 6 (148) (2005) 1–11.

[31] G.H. John, R. Kohavi, K. Pfleger, Irrelevant feature and the subset selection problem, in: Proceedings of the Eleventh International Conference on Machine Learning, 1994, pp. 121–129.

[32] R. Jörnsten, B. Yu, Simultaneous gene clustering and subset selection for sample classification via MDL, Bioinformatics 19 (9) (2003) 1100–1109.

[33] W. Junyun, Study and analyze on feature selection in text categorization for engineering domain, Adv. Mater. Res. 487 (2012) 383–386.

[34] A. Kalousis, J. Prados, M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, Knowl. Inf. Syst. 12 (1) (2007) 95–116.

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

[36] D. Koller, M. Sahami, Toward optimal feature selection, in: Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 284–292.

[37] C. Krier, D. Francois, F. Rossi, M. Verleysen, Feature clustering and mutual information for the selection of variables in spectral data, in: Proceedings of the European Symposium on Artificial Neural Networks, 2007, pp. 157–162.

[38] P. Krízek, J. Kittler, V. Hlavác, Improving stability of feature selection methods, Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, 4673, Springer, 2007, pp. 929–936.

[39] L.I. Kuncheva, A stability index for feature selection, in: Proceedings of the 25th IASTED International Multi-Conference, 2007, pp. 390–395.

[40] N. Lavrac, Selected techniques for data mining in medicine, Artif. Intell. Med. 16 (1999) 3–23.

[41] J. Lee, D. Kim, Memetic feature selection algorithm for multi-label classification, Inf. Sci. 293 (2015) 80–96.

[42] P.M. Lewis, The characteristic selection problem in recognition systems., IRE Trans. Inf. Theory 8 (2) (1962) 171–178.

[43] J. Li, H. Zha, Simultaneous classification and feature clustering using discriminant vector quantization with applications to microarray data analysis, in: 1st IEEE Computer Society Bioinformatics Conference, CSB, 14–16 August 2002, IEEE Computer Society, 2002, pp. 246–255, doi:10.1109/CSB.2002.1039347.

[44] H. Liu, H. Motada, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 1998.

[45] H. Liu, H. Motoda, On issues of instance selection, Data Min. Knowl. Discov. 6 (2) (2002) 115–130.

[46] H.C. Liu, P.C. Peng, T.C. Hsieh, T.C. Yeh, C.J. Lin, C.Y. Chen, J.Y. Hou, L.Y. Shih, D.C. Liang, Comparison of feature selection methods for cross-laboratory microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 10 (3) (2013) 593–604.

[47] S. Loscalzo, L. Yu, C. Ding, Consensus group stable feature selection, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, International Conference on Knowledge Discovery and Data Mining, 2009, pp. 567–576.

[48] R. Ltd., I. Carnegie Group, Reuters-21578, 1995.

[49] S. Ma, J. Huang, Penalized feature selection and classification in bioinformatics., Brief. Bioinform. 9 (5) (2008) 392–403.

[50] A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, in: Provceedings of AAA Workshop on Learning for Text Categorization, AAAI Press, 1998, pp. 41–48.

[51] A. Mitchell, A. Divoli, J. Kim, M. Hilario, I. Selimas, T.K. Attwood, METIS: multiple extraction techniques for informative sentences., Bioinformatics 21 (22) (2005) 4196–4197.

[52] C.H. Ooi, M. Chetty, S.W. Teng, Relevance, redundancy and differential priorization in feature selection for multiclass gene expression data, in: Proceedings of the 6th International Symposium on Biological and Medical Data Analysis, in: Lecture Notes in Computer Science/Lecture Notes in Bioinformatics, 3745, Springer, 2005, pp. 367–378.

[53] H.L. Peng, C.F. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.

[54] F. Pereira, N. Tishby, L. Lee, Distributional clustering of English words, in: Proceedings of Association for Computational Linguistics, ACL, 1993, pp. 183–190.

[55] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M.A.M.E. McLaughlin, J.Y.H. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, T.R. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (6870) (2002) 436–442.

[56] Y. Saeys, T. Abeel, Y. van de Peer, Robust feature selection using ensemble feature selection techniques, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, in: Lecture Notes In Artificial Intelligence, 5212, 2008, pp. 313–325.

[57] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics., Bioinformatics 23 (19) (2007) 2507–2517.

[58] M. Sebban, R. Nock, A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recogn. 35 (2002) 835–846.

[59] G.S. Sebestyen, Decision-Making Processes in Pattern Recognition, ACM monograph series, Macmillan, 1962.

[60] X. Shen, H.-C. Huang, Grouping pursuit through a regularization solution surface, J. Am. Stat. Assoc. 105 (490) (2010) 727–739.

[61] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus, T.S. Ray, M.A. Koval, K.W. Last, A. Norton, T.A. Lister, J. Mesirov, D.S. Neuberg, E.S. Lander, J.C. Aster, T.R. Golub, Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning., Nat. Med. 8 (1) (2002) 68–74.

[62] B. Silva, N. Marques, Feature clustering with self-organizing maps and an application to financial time-series for portfolio selection, in: Proceeding of the Sixth International Conference on Neural Computation, 2010, pp. 301–309.

[63] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2) (2002) 203–209.

[64] N. Slonim, N. Tishby, The power of word clusters for text classification, in: Proceedings of the23rd European Colloquium on Information Retrieval Research, 2001.

[65] Q. Song, J. Ni, G. Wang, A fast clustering-based feature subset selection algorithm for high-dimensional data, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 1–14.

[66] J. Tang, S. Alelyani, H. Liu, Data Classification: Algorithms and Applications, Data Mining and Knowledge Discovery Series, CRC Press, pp. 37–64.

[67] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B 58 (1994) 267–288.

[68] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, K. Knight, Sparsity and smoothness via the fused lasso, J. R. Stat. Soc. Ser. B 67 (1) (2005) 91–108.

[69] A. Unler, A. Murat, R.B. Chinnam, mr2pso: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Inf. Sci. 181 (20) (2011) 4625–4641.

[70] H. Uğuz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, Knowl.-Based Syst. 24 (7) (2011) 1024–1032.

[71] J.R. Vergara, P.A. Estv€z, A review of feature selection methods based on mutual information., Neural Comput. Appl. 24 (1) (2013) 175–186.

[72] C.M.M. Wahid, A.B.M.S. Ali, K.S. Tickle, A novel hybrid approach of feature selection through feature clustering using microarray gene expression data, in: Proceedings of Hybrid Intelligent Systems, HIS, IEEE, 2011, pp. 121–126.

[73] H. Wang, D. Bell, F. Murtagh, Axiomatic approach to feature subset selection based on relevance, IEEE Trans. Pattern Anal. Mach. Intell. 21 (3) (1999) 271–277.

[74] L. Wang, F. Chu, W. Xie, Accurate cancer classification using expressions of very few genes, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (1) (2007) 40–53.

[75] C.H. Yang, L.Y. Chuang, C.H. Yang, Ig-ga: a hybrid filter/wrapper method for feature selection of microarray data, J. Med. Biol. Eng. 30 (1) (2010) 23–28.

[76] L. Yu, C. Ding, S. Loscalzo, Stable feature selection via dense feature groups, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 803–811.

[77] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.

[78] H. Zou, The Adaptive Lasso and Its Oracle Properties, J. Am. Stat. Assoc. 101 (476) (2006) 1418–1429.

[79] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B 67 (2005) 301–320.