Journal of Informatics and Mathematical Sciences
Vol. 7, No. 1, pp. 21–48, 2015
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications

http://www.rgnpublications.com

An Hybrid Method for Feature Selection based on Multiobjective Optimization and Mutual Information

Research Article

Enguerran Grandchamp[1], Mohamed Abadi[2] and Olivier Alata[3]

[1] *Laboratoire LAMIA, Université des Antilles et de la Guyane, Campus de Fouillole, 97157 Pointe-à-Pitre Guadeloupe, France*
[2] *Institut XLIM-SIC, UMR CNRS 6172, Université de Poitiers, BP 30179, 8962 Futuroscope-Chasseneuil, Cedex France*
[3] *Lab. Hubert Curien, UMR CNRS 5516, Univ. Jean Monnet Saint-Etienne, Univ. Lyon, 42000, Saint-Etienne, France*
***Corresponding author:** egrandch@univ-ag.fr

Abstract. In this paper we propose a hybrid approach using mutual information and multi-objective optimization for feature subset selection problem. The hybrid aspect is due to the sequence of a *filter* method and a *wrapper* method in order to take advantages of both. The *filter* method reduces the exploration space by keeping subsets having good internal properties and the *wrapper* method chooses among the remaining subsets with a classification performances criterion. In the *filter* step, the subsets are evaluated in a multi-objective way to ensure diversity within the subsets. The evaluation is based on the mutual information to estimate the dependency between features and classes and the redundancy between features within the same subset. We kept the non-dominated (Pareto optimal) subsets for the second step. In the *wrapper* step, the selection is made according to the stability of the subsets regarding classification performances during learning stage on a set of classifiers to avoid the specialization of the selected subsets for a given classifiers. The proposed hybrid approach is experimented on a variety of reference data sets and compared to the classical feature selection methods *FSDD* and *mRMR*. The resulting algorithm outperforms these algorithms and the computation complexity remains acceptable even if it increases with regards to these two fast selection methods.

Keywords. Hybrid feature selection; Mutual information; Multiobjective optimization; Classification

MSC. 90C27; 90C29; 90C59

Received: February 2, 2015 **Accepted:** April 27, 2015

1. Introduction

Feature Selection (FS) is an active topic of interest since about 20 years [49]. As a consequence, a large number of algorithms have been proposed by the research community. The basic idea of feature selection is to select a subset from a large set of features. It can be considered as a particular instance of Feature Weighting (assigning weights to features according to their importance for a given application) [57]: weights are set to 0 or 1 instead of a real value in the range $[0,1]$. In addition to this, FS is a branch of the Dimension Reduction problem [20]. The other branch is Feature Extraction [16] whose aim is to produce new features by combining several initial features. FS is an important task in many fields such as text characterization [11], image research [8], bioinformatics [46], color image processing [42], data mining ([37], [38]), etc. The aim is to select relevant features for knowledge interpretation or representation, computation time reduction and overall improvement in performance(such as classification accuracy). Indeed, the size of the subset has an influence on the computation time, and its representativeness of the quality of the post treatments.

The relevancy of the features can have different definitions depending on the application: in knowledge interpretation or representation [34], the size reduction and the semantic and/or the diversity of the selected features are important in order to keep in a lower dimension the topological structure of the information [51]; for classification applications [41], relevancy is directly linked to a good rate in learning or classification; in protein biomarkers identification [20], the reduction of the feature subset size and its stability when applying different learning sets are more important than classification performances. The relevancy is linked to the quality, the complexity, the diversity or the performance of the feature subset.

Depending on the application and constraints (such as time, performances, etc.) different approaches have been developed to select a subset of features. These approaches differ by their research method to explore the subsets, their criterion for comparing and ranking them and their selection process.

In this paper, we design a hybrid method to combine the advantages of both *filter* and *wrapper* approaches: a fast (*filter*) way to select diversified subsets (multi-objective) having good internal properties (*filter*) and a final selection based on performances (*wrapper*). The stability criterion avoid specializing the subsets to a given classifiers.

After a general presentation of the main exploration methods, the fitness functions and selection processes are presented in section 2. Section 3 presents the multi-objective principle. Then in section 4, we present the hybrid method and the criterions. In section 5, some formalism is given concerning the criterion, the non-domination principle and the algorithm. The experiments on benchmarking database, classification and segmentation applications are given in section 6. Finally, section 7 gives conclusions and perspectives of the work.

2. Background: the feature selection problem

During the last years, many papers have been published on the modeling [49] and the description ([20], [46], [53]) of feature selection problem.In this section, we summarize the main ideas implemented in the different feature selection approaches. Categorization is done according to exploration methods, fitness function or selection process.

The main search strategies [54] are:

- Greedy methods based on sequential approaches such as Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS). These methods respectively add or remove one feature at each iteration [55].
- Sequential Forward Floating Selection (SFFS) and Sequential Backward Floating Selection (SBFS). These methods alternate Forward and Backward steps ([24],[35]).
- Genetic Algorithms (GA) ([6], [13]) which are evolutionary methods managing a population of subsets which evolves generation after generation using crossover and mutation operators ([18], [56]).
- Or other strategies like Tabu Search [59] or Ant Colony [26].

The exhaustive search is excluded in most of the cases because the number of feature subsets within a set of size n, is $2n$. This leads to a very time consuming exploration as soon as n is greater than 20. This procedure even becomes impossible when the number of attributes drastically increases to reach hundreds or thousands.

The main fitness function used to evaluate the candidate subsets are based on:

- A quality measure evaluated on each features separately: Dependency [40], entropy [7], relief-f ([25], [30], etc.), distance measures [33], statistical measures [42] and more recently probabilistic measures based on the estimation of Mutual Information ([34], [40]); or directly on the subset: correlation, redundancy [40], Information Criteria [1], for example.
- A performance measure: the good classification rate ([9], [60]) or error rate ([4], [41]) during the learning step of a classifier applied on the candidate's subset.
- A complexity measure: the cardinality of the subset ([17], [21], [41]), the complexity of the classifiers (size of a decision tree [38] or Neural Network [22]).

The selection processes commonly used are:

- A single candidate selection for sequential approaches: maximization (classification rate, relevancy, etc.) or minimization (error rate, correlation, etc.) of the criterion.
- Multiple candidates for evolutionary approaches (GA [4], [60]): multiple selection wheel based on the previous criterion.

At the end of the exploration, one (or more) solution(s) is (are) returned and must be evaluated. The commonly used evaluation is based on their performances in a classification context.

These different approaches lead to the separation of the methods in four families. This separation is mainly based on how to compare and to rank the subsets: *wrapper* [29], *filter* [43] and embedded [54] methods.

- *Wrapper* methods use a machine learning algorithms during the exploration step to evaluate the candidate's subsets and the corresponding classifier during the evaluation of the returned solution (test stage). In this context, the feature selection is often used to improve or accelerate a classifier. In most cases, this approach gives the best performances when applying the final evaluation and is probably the most pertinent. But it is also time consuming because it requires the training of a large number of classifiers, a manual specification of many parameters and an expertise in machine learning.
- *Filter* methods do not use any feedback from the classifiers and use an independent criterion to measure the quality of the feature subsets. These methods are the most popular because they considerably reduce the computation time [25]. Moreover, the risk of over-fitting with *wrapper* methods do not exist with *filter* methods which are better in generalization.
- Embedded methods try to combine the advantages of both approaches [15]. They include the feature selection step in the learning phase of a classifier. Nevertheless, the computation times till remain important and a classifier is used to rank the feature subsets.
- Hybrid methods use a sequence of *Filter* and *Wrapper* methods ([3], [5], [40], [48], [52]).

More details are given in the previous references and particularly in [20] and [51] which are surveys of methods.

3. Mono or Multiobjective approach

Most of the time the exploration methods deal with a single criterion ([3], [4], [12], [20], [21], [22], [25], [23], [36], [40], [41], [50], [55], [60]).

However the use of only one characteristic to rank and select the subsets is insufficient in many cases. Authors then defined combinations of several criterions to integrate quality and performance. In practice, defining a combination of criterions is not an easy task. It depends on the application and often requires parameters to balance the different parts of the criterion. For example, in [40] the authors combine Dependency and Redundancy in a unique criterion called *mRMR* (minimal-redundancy-maximal-relevance). In [27] they use the notion of support for associations in data mining and combine two expressions favoring small and large supports. In most of studies ([21], [41], [56], [60]), the performance criterion (such as classification rate) is modified in order to penalize subsets with high number of selected features.In the same way Information Criteria (IC, [1], [47]) are written as a sum of a maximum likelihood based term, for the quality measure, with a penalty term depending on the complexity of the subset.This leads to different expression of the IC (Akaike IC (AIC), Bayesian IC (BIC) or φ_β). However these criterions generally have opposite behaviors because increasing the performances often requires adding features which increase complexity. The aggregation of these objectives into a single one requires a normalization and weighting of the two objectives and it is possible to investigate alternative ways of aggregation than the classical IC. Moreover,these computations implicitly establish a total order relation between the candidate solutions leading to a unique solution and avoiding the emergence of diversified solutions.

In order to bypass this drawback, a multi-objective approach has been adopted in some studies ([9], [17], [18], [38], [19]) to solve the FS problem. A multi-objective approach try to simultaneously optimize several (mainly two) fitness functions during the exploration. However the criterions often have opposite behaviors leading to non dominated solutions [23] (i.e. a solution is better than another one according to one criterion and worse according to the other one). Indeed, if mono-objective optimization can deal with a unique solution evolving during the iterations, a multi-objective optimization leads to a set of non dominated solutions. This set is called the Pareto set [23].

For the FS problem, the different approaches ([9], [10], [17], [18], [38]), deal with *wrapper* methods using GA as exploration method. GA is the most popular method to explore a wide space without a priori knowledge due to its stochastic convergence [14] and its ability to manage with different kind of solutions. Moreover, the use of a simple binary encoding and standard crossover and mutation seems to be attractive for the FS problem.

Nevertheless, the multi-objective approach is always implemented in a very simple way. One of the objective is the cardinality of the subset and the other one a classification rate or error rate. In ([37], [38]), the approach is nevertheless less trivial as it uses a decision tree to evaluate the feature subsets (the two objectives being the global error rate during the learning step and the size of the resulting tree representing the complexity of the solution), GA and aMulti-Objective Forward Sequential Search (MOFSS) to explore the subsets.

In a two-objective optimization, taking the minimization of the cardinality as first objective is equivalent to keep at most one subset for a given cardinality in the Pareto front. Indeed, for a given cardinality the only retained subset is the one optimizing the second objective, and if a subset with lower number of features has a better second objective this subset is not kept in the Pareto front. In such condition, fixing the number of features directly leads to a unique solution which shows a lack of diversity.

All these approaches try to optimize both the complexity of the model (minimization of the number of features or size of the decision tree) and the performances (classification rate maximization or error rate minimization). The main observations are that (i) only wrappers approaches have been explored in a multi-objective way; (ii) both GA and MOFSS returned a few number of solutions. Let us recall that the main drawbacks of the *wrapper* approach are the computation time and the dependence of the selected features to the classification algorithm used during the exploration. This has been underline as a problem for biomarkers selection [20].

4. The proposed hybrid approach

4.1 Combination of *Filter* and *Wrapper* approaches

Hybrid methods combining *Filter* and *Wrapper* methods are proposed in the literature ([3], [5], [25], [40], [48], [51]) but the main objective of these works is to reduce the computation time. Indeed the criterion used in the *wrapper* step is the classifier performance in a mono-objective approach. The *filter* method is used to reduce the exploration space in a very high dimensional data set [51] by evaluating the quality of the features in a mono-objective way: Kullback-Leiberdistance between histograms of feature values [3]; *mRMR* criterion [40]; the relief criterion [25] and [30]; the relative certainty gain [48].

The way to select the subsets for the second stage (the *wrapper* step) represents the main differences between the approaches. In [3] they select the features by fixing a threshold on the Kullback-Leiber distance; In [40], they keep visited subsets having a classification error (obtained by applying a classifier on the subset) under a given threshold; In [48] a threshold is also used for the statistical test.

As the number of features is reduced by the *filter* step, the wrapper step manages the retained features by the mean of a classical GA ([3], [25]) or sequential forward and backward search ([40], [48]) with a classification accuracy criterion.

We propose a hybrid method by combining the *Filter* and the *Wrapper* methods in two sequential steps. The approach we adopt improves the lack of diversity of the solutions returned by standard algorithms and reduces the dependency between subsets and classifiers. These improvements are due to a wider exploration of subsets which increases complexity. The computation time remains acceptable thanks to the use of a fast filter approach and a controlled exploration of Pareto solutions during the first step of our method (see section **??** for more details about computation complexity and computation time). These procedures coupled with a multi-objective approach with two quality objectives allow keeping diversity. All the subsets selected using the Pareto front are evaluated during a second step called the *wrapper* step. During this step, we prefer a stability criterion to select the final subsets instead of raw performances regarding one classifier, in order to keep performances and independency between subsets and classifiers. Indeed, we are looking for diversified subsets in the *filter* step in order to have different kinds of solutions to evaluate during the *wrapper* step to increase the probability to reach stable ones. In this way, the building of the Pareto front seems to be the more appropriate choice because the subsets are selected regarding two different quality criterions.

4.2 Criterion and Diversity

The second stage of some previous approaches maintains a kind of diversity by the crossover step and the mutation step of a GA. On the other hand, the selection of the first pool of features by the *filter* step is done using a single criterion which restricts the explored subsets. Indeed, the evaluation of the subsets is done in a single way which leads to reject subsets having good properties according to another criterion. This is particularly the case for single criterions which are composed of multiple parts (*mRMR* for example, composed of Redundancy and Relevance). In this context, solution having very low redundancy or very high relevance could be rejected by the selection process if the resulting aggregation function has a low evaluation.To increase the diversity of the selected subsets our *filter* step explores the space in a multi-objective way with two quality objectives and a complexity objective.

The evaluation of the quality is based on the Mutual Information (MI) to separately measure the Dependency (D) and Redundancy (R) of the subsets. The theoretical interest for Mutual Information has been proved in [40]. These two criterions measure both the individual quality of the selected features and the quality of the subset. The separate evaluation of these two measures (contrary to [40]) is important because a relevant subset is not necessarily a subset containing only significant attributes taken alone. Indeed the relevance of a subset may be due to combinations of features. Fig. 1([3]) illustrates the well known case of two features with no power of discrimination used alone, while combined they separate the two classes optimally.

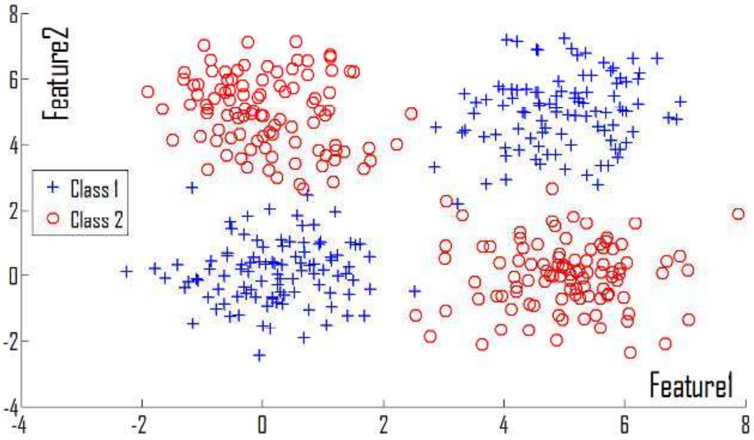


Figure 1. Combination of features

5. Formalism and computation

In this section we give the formulation of the criterions and how to compute them. Then, we present the multi-objective aspect of the method.

5.1 Criterions based on mutual information and complexity

The mutual information is considered to be a good indicator to study the dependency between a feature and the classification and the redundancy between random features.

Mutual Information

Let X and Y be two random variables with discrete probability laws. The Mutual Information (MI) $I(X;Y)$ is defined by $P(X)$, $P(Y)$ and $P(X,Y)$.

$$I(X;Y) = \sum_{y\in\Omega_Y}\sum_{x\in\Omega_X} P(x,y)\cdot\log\frac{P(x,y)}{(P(x)\cdot P(y)}. \tag{5.1}$$

with Ω_X and Ω_Y the sample spaces of X and Y respectively.

When X and Y are dependent, $I(X;Y)$ is high. $I(X;Y)$ is equal to zero when X and Y are independent.

Mutual Information estimation

The estimation of the MI is easy for discrete variables because joint probabilities are estimated by counting the representative samples of each variable [58]. When at least one variable is continuous the MI is not easily computable and hardly depends on the way to estimate the probability density function. Kwak et al. [32] propose to use the Parzenwindow (5.2) to estimate probability density function and to approximate $I(X;Y)$:

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N}\delta(x - x^{(i)}, h) \tag{5.2}$$

where $\delta(\cdot)$ represents the Parzenwindow which is here a Gaussian kernel ([39], [40]). $x^{(i)}$ is the ith sample, N is the number of samples and his the width of the window. Parzen [39] shows that if $\delta(\cdot)$ and hare correctly chosen $\hat{p}(x)$ converges to $p(x)$ when N increases [40].

Selection criterion definition

For each subset of features, we define the relevance expressed by the Dependency (D) which is the average MI between the variables of $S(X_i)$ taken separately and the class of the samples modeled by a discrete random variable called c with sample space equal to the class labels:

$$D_S = \frac{1}{|S|}\sum_{X_i \in S} I(X_i;c) \tag{5.3}$$

$I(X_i;c)$ represents the MI between a variable and the classes. It translates how X_i is useful to describe the classes.

The Dependency has to be maximized. However in order to have a homogenous expression of the objective we prefer to express the opposite of the Dependency $(-D)$ to minimize each criterion.

The feature selection using only D is not optimal because of redundancy between the variables [2]. Different ways exist to measure the redundancy and we use the one expressed in [40]. It is based on the computation of the average MI between two variables $(X_i;X_j)_{i,j=1,\ldots,mi\neq j}$ belonging to the same subset Shaving m variables.

$$R_S = \frac{1}{|s|^2}\sum_{X_i,X_j \in S} I(X_i;X_j) \tag{5.4}$$

The redundancy must be minimized

These criterions are treated separately contrary to [40] and [2]. In these works, the criterions are combined to produce the *mRMR* (minimal-redondance-maximale-pertinence)criterions (ex. $\max_S(D_S - R_S)$ or $\max_S(D_S/R_S)$). These mono-objective criterions didn't ensure the simultaneous convergence of criterions (5.3) and (5.4) to their optimal value but lead to a trade-off between them.

The third criterion is the subset cardinality (L) which must be minimized.

5.2 Multi-objective optimization and Pareto set

The goal of a multi-objective optimization is to improve several criterions. When these criterions have opposite behaviors considering the research of a solution, we necessarily have to degrade at least one criterion to improve another one. This leads to different kind of solutions which are not necessarily comparable. If we don't want to make a choice between too different solutions (for example, between a cheap but slow car and an expensive but fast one) we must keep all solutions being better than any others on at least one criterion. This leads to the notion of domination which is essential to ensure diversity in the final set of subsets.

Without loss of generality, we illustrate this notion in our particular case.

Following the previous section, each subset is evaluated with three values $(f_1, f_2, f_3) = (-D, R, L)$.

- A subset S dominates a subset S_2 according to f_i if $f_i(S) < f_i(S_2)$, $i = 1, 2$, or 3.
- A subset S dominates a subset S_2 if $if_i(S) \leq f_i(S_2)$ and $\exists |\ f_i(S) < f_i(S_2)$.
- A subset S is not dominated if $\nexists\ S_2|S_2$ dominates S $(\nexists S_2|if_i(S_2) \leq f_i(S),\ \exists\ f_i(S_2) < f_i(S))$.
- The set of all non dominated subsets is called the Pareto set.

Since our objective is to minimizeeach criterion the Pareto front follows the template in Fig. 2.

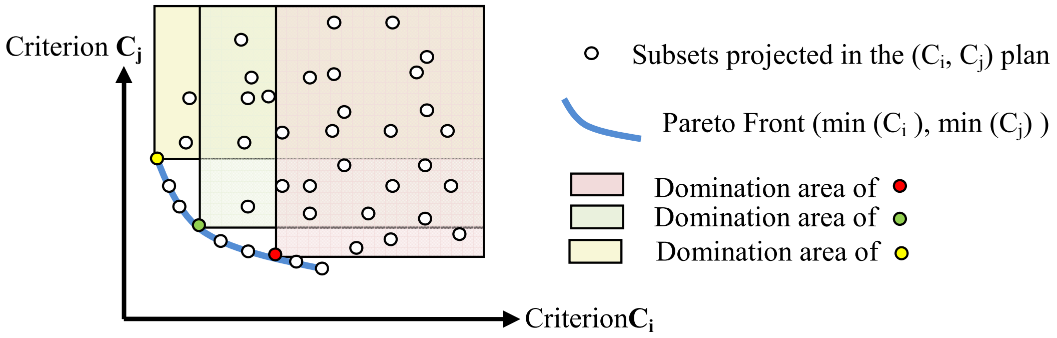


Figure 2. Pareto Front Template

The third criterion, which represents the complexity of the subset through its cardinality, allows keeping subsets with different size (for low number of features the redundancy may be better and for high number of features the dependency may be better). Nevertheless, even if we have one Pareto front for each possible subset size, there is no certainty to obtain at least one subset for each possible size. This could be an inconvenient for some applications. In such condition, the exploration step can deal with only the quality criterions and each intermediate Pareto front (corresponding to a specific size) could be kept. This approach called Multi Pareto Front (MF) is detailed in the next paragraph.

Finally to improve the visual interpretation of the Pareto solutions for the three Objectives (*3O*) approach, we project the 3D Pareto set $(-D, R, N)$ into the $(-D, R)$ plan and we represent one curve per subset size (Fig. 3). This representation is the same as in the 2D Multi Front (Fig. 2) but is not obtained in the same way. We can build a *3O* front from the 2O Multi-Front by computing the non dominated solutions in a *3O* approach starting with the *2OMF* subsets. The *2OMF* will keep more subsets and all *3O* subsets are included in the *2OMF* subsets.

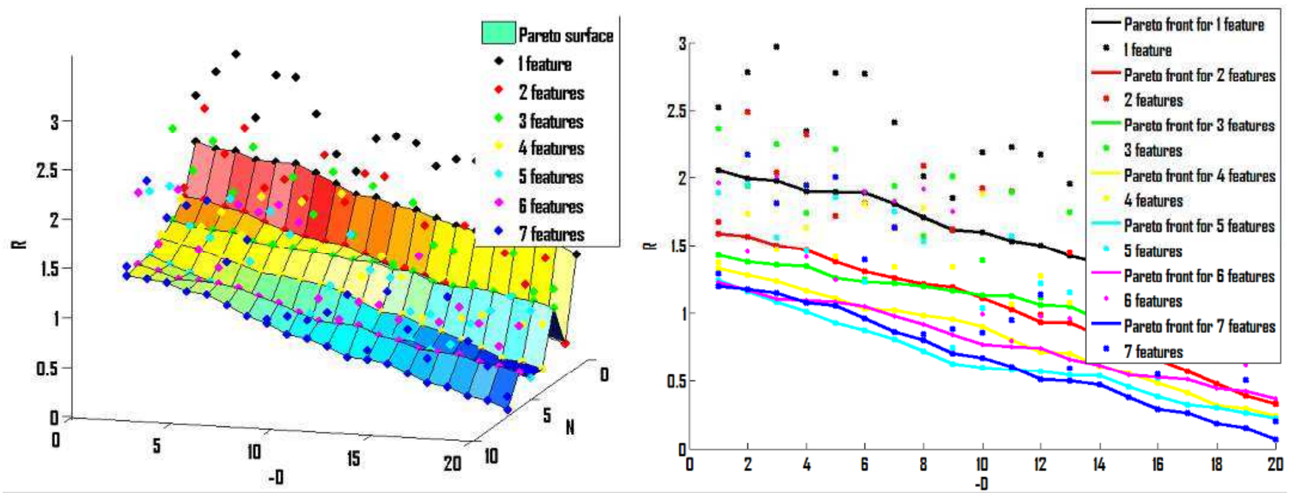


Figure 3. 3D and 2D representation of Pareto front

5.3 *Filter* exploration method

For any optimization problem, a unique Pareto set exists for a given data set and the considered criterions. In a muti-objective context, an exhaustive search, or an algorithm having asymptotic convergence properties such as Genetic algorithm ([18], [56]), is classically required to find this set. Both are time consuming and sometimes too slow to reach the optimal Pareto set in a reasonable time. In practice, people build a sub-optimal Pareto front which is the Pareto front computed over the visited solutions. One of the main qualities of a search method is then its ability to provide solutions close to the ones of the optimal Pareto front. Our *filter* search method joins this way and has been developed to approach the building of the optimal Pareto front.

The *filter* step uses a sequential forward search to explore the subset space adopting the following algorithm:

- Let $F = \{F_i | i \in [1,M]\}$ be the complete set of features.
- We start with all possible pairs of features $V_2 = \{(F_i, F_j) | i \in [1,M], j \in [1,M], i \neq j\}$. Indeed, we don't initialize the method with a unique feature having the better individual quality as in [40] because the redundancy requires at least two features and we want to keep diversity in order to solve the problem illustrated Fig. 1.
- Each subset Sis evaluated with $(-D(S), R(S), N(S) = 2)$ criterions and the non dominated subsets are preserved, and noted ND_2. We denote Fr_2 the Pareto front at iteration 2 $(Fr_2 = ND_2)$.
- At iteration k, ND_k represents the non dominated subsets of size k $(k > 2)$ and Fr_k is the global Pareto Front $(Fr_k = \bigcup_{i=2}^{k} ND_i)$.
- We build V_{k+1} by adding to the subsets of ND_k one new feature taken within the remaining features: $V_{k+1} = \{(S \cup F_i) | S \in ND_k, F_i \in F \setminus S\}$. Each subset S in V_{k+1} is then evaluated with $(-D(S), R(S), N(S) = k+1)$.

- We build ND_{k+1} by retaining the non dominated subsets of size $k+1$ within $E_{k+1} = V_{k+1} \cup Fr_k$. We note that $ND_{k+1} \subseteq V_{k+1}$. This step is required because some subsets of V_{k+1} can be dominated by subsets of Fr_k (the opposite is not possible because each subset in V_{k+1} is greater than each subset in Fr_k).
- The algorithm ends if ND_{k+1} is empty ($Fr_{k+1} = Fr_k$) or $k = M$. We denote m^* the value of k at the end of the algorithm.

There turned subsets are the ones belonging to Fr_{m^*}. This algorithm is called *3O*.

We can note that $Fr_k \subseteq Fr_{k+1}$. Indeed, subsets of V_{k+1} could not dominate subsets of Fr_k because these last ones have a lower size: $S_1 \in Fr_k$, $N(S_1) \leq k$, $S_2 \in V_{k+1}$, $N(S_2) = k+1$.

In this algorithm, subsets are deleted from the Pareto front as soon as they are dominated by a new subset. As mentioned earlier, this could lead to the deletion of all subsets of a given cardinality. If we want to maintain a pool of subsets for each size, we can adapt the algorithm in the following way:

- The subsets S are evaluated with only $(-D(S), R(S))$ at Step 3 and 5
- $E_{k+1} = V_{k+1}$ at Step 6
- The algorithm ends if $k = M$ at Step 7

The new algorithm is called Two Objectives Multi-Front Algorithm (*2OMF*) and the returned set of subsets is $Fr_M = \bigcup_{i=1}^{M} ND_i$.

We can observe that $V_k^{3O} \neq V_k^{2OMF}$ which implies that $ND_k^{3O} \neq ND_k^{2OMF}$ and $Fr_{m^*}^{3O} \neq Fr_M^{2OMF}$ because the domination relation is computed over different subsets (different expression of E_{k+1}).

A final step can be added to the *2OMF* algorithm. This final step consists on computing the Pareto set of Fr_M in a three-objective way $(-D(S), R(S), N(S))$. The corresponding algorithm is called *2OMF-3O*.

5.4 *Wrapper* step and stability criterion

The *wrapper* step is used to rank the selected subsets and to select a subset considering the application. For this step, the exploration space has been sufficiently reduced during the *filter* step to allow an exhaustive evaluation of the remaining subsets ND_F. A large majority of *wrapper* approaches deals with Feature Selection in terms of performances regarding a classifier, but few studies select subsets for their stability. Nevertheless, the stability is a topic of interest in studies dealing with high dimensional data and a small number of samples ([20], [45]). Moreover, *wrapper* methods can lead to good classification accuracy for a specific classifier but with poor generalization properties ([28], [40]) (i.e. over-fitting for one classifier and low performances for another one [44]).

The stability is defined by Somol as being the quality of a subset to have the same performances with different training sets ([28], [53], [51]). Different stability indices can be used [51] such as Hamming distance, correlation coefficients, Tanimoto distance, consistency index (simple, weighted or relative weighted) and Shannon entropy. In [31] and, the stability is measured by running a *wrapper* scheme several times with a unique classifier and different learning sets (no cross-validation). The stability is based on an evaluation of the similarity between subsets returned by different runs. If the index is high the subset is selected. Otherwise the selection is based on the classification rate evaluation.

In this paper, we investigate another kind of stability between different classifiers (each trained and evaluated with a cross-validation process). According [20] to this kind of stability has been neglected in the literature. A subset is stable regarding classifiers if the performances obtained with different classifiers are close. The easiest way to compute the stability of a subset is to compute the amplitude of the classification rates obtained with several classifiers K-Nearest Neighbor (KNN), Linear Discriminant Analysis (LDA), Mahalanobis (Mah), Naive Bayes (NB), Simple Vector Machine (SVM) and Probabilistic Neural Network (PNN):

$$A(S) = \max_{c \in Cl}\{R_c(S)\} - \min_{c \in Cl}\{R_c(S)\} \tag{5.5}$$

Where S is the subset, Cl a set of classifiers and $R_c(S)$ the classification rate obtained with the classifier c applied on the subset S.

Finally, we identify the stable and successful subsets. Therefore, the selection of the interesting subsets is done in a two objectives way by maximizing the mean classification rate ($M(S)$) and by minimizing the amplitude ($A(S)$).

$$M(S) = \operatorname*{mean}_{c \in Cl}\{R_c(S)\} \tag{5.6}$$

6. Experiment on benchmarks

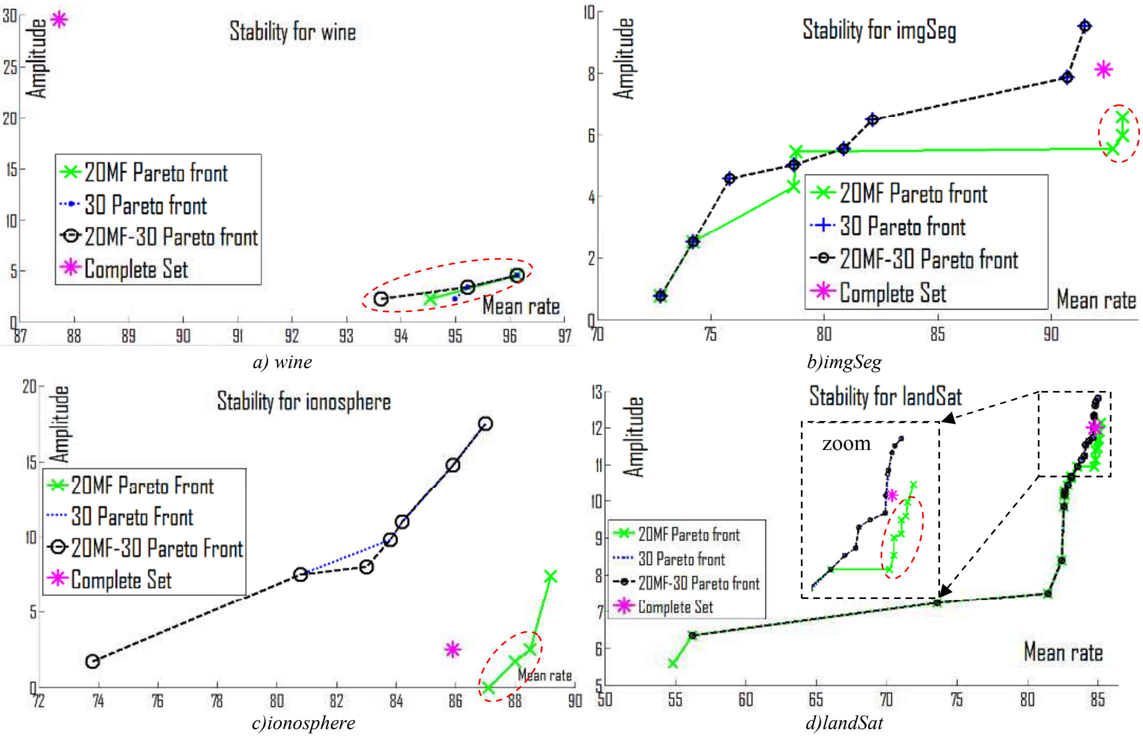
The experimentations are made in three steps. During the first step, our aim is to choose the best method between *2OMF*, *3O* and *2OMF-3O* methods. The comparison is done by means of the diversity of the subsets, the computation time of the *filter* step, and the stability of the subsets after *wrapper* step. The second step presents a more detailed analysis of the best method, the *2OMF*. Then the third step compares the *2OMF* method with two other existing feature selection methods: *mRMR* [40] and *FSDD* [33]. Both are using *filter* criterions to select the features and then they evaluate the unique returned solution using classifiers.We choose *mRMR* method because it uses the same criterions as *2OMF* but in a mono-objective way. We choose *FSDD* because it is a fast algorithm which converges to the optimal solution regarding a distance criterion. In both cases, it is interesting to project the solutions obtained with different *filter* steps in the space (performance, stability) of the *wrapper* step and to compare them with our pool of solutions. The comparison is done by means of the size and the stability of the subsets returned by each method and also the computational time of each method.

Each step uses UCI databases for validation and more particularly iris, *TAE*, *abalone*, *PimaIndiansDiabetes*, *wineRed*, *wineWhite*, *wine*, *imgSeg*, *ionosphere* and *landSat* databases containing 4, 5, 7, 8, 11, 11, 13, 18, 34 and 36 features respectively. Figure 4 to Figure 11 present some of the obtained results.

The stability is computed after applying *KNN*, *LDA*, *Mah*, *NB*, and *PNN* classifiers.

6.1 Comparison between *2OMF*, *3O* and *2OMF-3O*

We compare the performances of the 3 algorithms. Figure 4 illustrates the Pareto fronts of each method in the (performance, stability) space for the *wine*, *imgSeg*, *ionosphere* and *landsatdatabases*. Our analysis focuses on these databases because they have the largest number of features. Therefore, they are more representative of the studied problem. However, the same observations can be done on the other databases. We also compare the subsets of the different Pareto fronts to the complete set regarding the stability of the classification.



a) wine

b)imgSeg

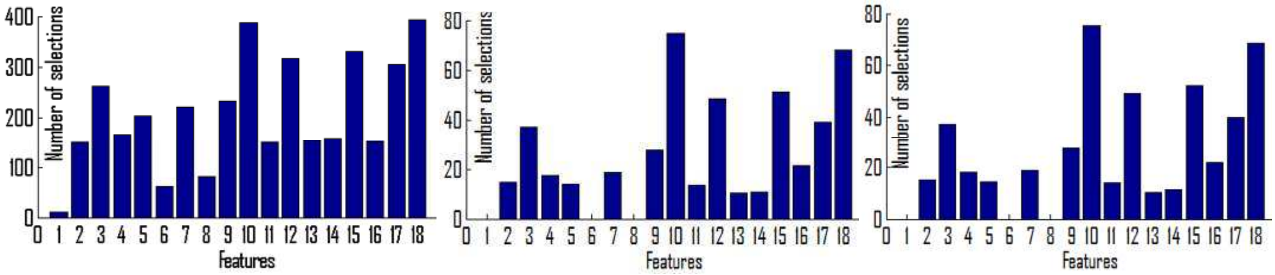
c)ionosphere

d)landSat

Figure 4. *Wrapper* Stability front comparison between *2OMF*, *3O* and *2OMF-3O*

Figure 4 shows that the *2OMF* Pareto front (green solid line) is at least equivalent and often better than the *3O* (blue dotted line) and *2OMF-3O* (black dashed line) Pareto fronts. Moreover, Figure 4 shows that the complete set of features (purple star in the figure) never dominates a subset of the *2OMF* Pareto front. It is also often the case for the *3O* (blue dotted line) and *2OMF-3O* (black dashed line) fronts: curves often overlap or are very close for these two methods. In Figure 4(a) all subsets in the fronts dominate the complete set. In Figure 4(b), 4(c) and 4(d) the subsets within the red dashed ellipse dominate the complete set. It means that besides being not dominated, solutions from *2OMF* front have better stability properties than the complete set regarding both objectives (mean rate and amplitude). Moreover, the complete set is always dominated by at least one subset of the *2OMF* Pareto front. This first result is of interest because it shows subsets having better performances than the complete set although having a lower number of features. We can also conclude from this first result that *2OMF* is a better search method than *3O* and *2OMF-3O*.

In order to explain the differences between the three methods we analyze the composition of the subsets contained in the Pareto fronts returned by the *filter* step. Figure 5(a), 5(b) and 5(c) show the histogram of the selected features in the subsets of the Pareto front returned by the three approaches. We note that the global shape of the histograms is the same for the three methods. However we observe a more homogenous histogram in the case of *2OMF* with all features selected in the different subsets. This is not the case for the two other methods where some features totally disappear (feature 6 and 8 for example) and where the gap between the most represented and least represented features is proportionally larger (features number 10 and 11 for example). Moreover, all subset sizes are not retained with *3O* and *2OMF-3O* (size < 10) (Figure 6)instead of *2OMF* which includes all subset sizes.



a) 2OMF *b) 3O* *c) 2OMF – 3O*

Figure 5. Comparison of the selected features after *Filter* step with *2OMF*, *3O*, *2OMF-3O* for *imgSeg* database

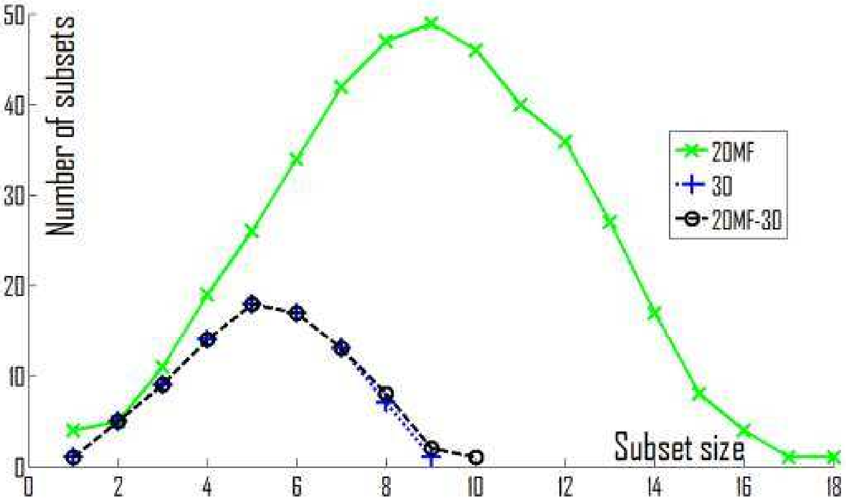
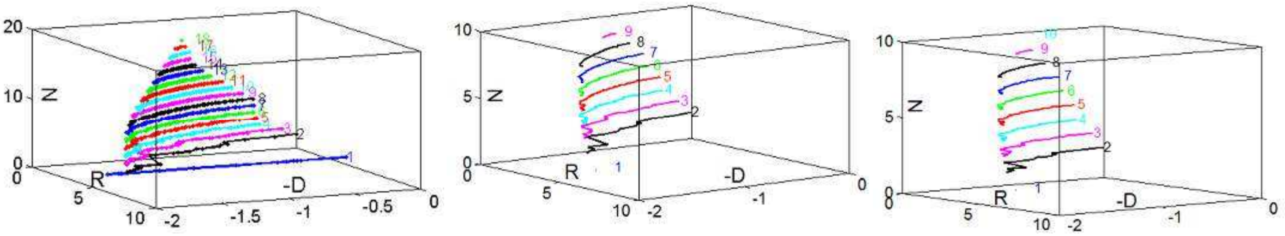


Figure 6. Subset sizes distribution after *Filter* step with *2OMF*, *3O*, *2OMF-3O* for *imgSeg* database

Figure 6 shows that *2OMF* returns more subsets than the other methods. In order to give an information about their composition, Figure 7 displays the Pareto front in three dimension $(-D,R,N)$ for each algorithm. Let us notice that the Pareto fronts have the same shape for the three algorithms. Nevertheless the one of *2OMF* contains more subsets and more subset sizes.



a) 2OMF *b) 3O* *c) 2OMF-3O*

Figure 7. 3D Pareto fronts representation of *filter* step for *imgSeg* database

Figure 8 shows the evolution of the Pareto front during the *filter* step ($(-D,R)$ space) for *wineWhite* database for subset sizes from 4 to 10. Each line represents the Pareto front for a given subset size and each column for a given algorithm. We can see that *2OMF-3O* and *3O* subsets are very close and by construction *2OMF-3O* subsets are a subpart of the *2OMF* subsets. Moreover, when the size of the subsets increases the number of subsets exclusively explored by *2OMF* increases too. In the same time the number of subsets within *3O* and *2OMF-3O* Pareto fronts decreases until 0 for a subset size 10. The same observation can be done for the other databases. A further analysis of the selected subsets by the three methods shows that the Pareto subsets obtained with *3O* and *2OMF-3O* are statistically dominated by at least one *2OMF* subset: 98.7% of *3O* and 100% of *2OMF-3O* subsets are dominated and only 0.1% of *3O* subsets dominate one *2OMF* subset.

Finally the computation time, is linked to the number of explored subsets during the search procedure and *2OMF-3O* is the more time consuming. Indeed, the *2OMF-3O* is not efficient because it requires running the *2OMF* algorithm and after applying a *3O* reduction. *2OMF* is around three times longer than *3O* method. Nevertheless, it is not a limiting factor for our application (see section **??**).

The *2OMF* approach gives better results because the *filter* step returns more subsets and more diversified ones (size and composition). The reason is that the systematic exploration of every subset size with the Multi-Front technique avoids falling into a local optimum leading to exclude some features. Indeed, during the *filter* step the *3O* algorithm stops whenever no new subset (build from the current Pareto front by adding one feature) improves one of the objectives (D or R). For example, in the case of *imgSeg* database, the *3O* algorithm ends with a maximum subset size of 9 and the *2OMF-3O* ends for subset size equal to 10 for *imgSeg* database (Figure 6). Let us recall that the gain on one objective is obtained when adding several features in some cases (Figure 1).

Therefore, the *3O* method can be preferred if computation time reduction is necessary or can be used to underline useful features. Elsewhere, *2OMF* can be recommended. Indeed, the lack of some kinds of subsets (composition and size) can be a drawback in the second step of the algorithm. For these reasons, we choose to retain the *2OMF* algorithm in the rest of the paper.

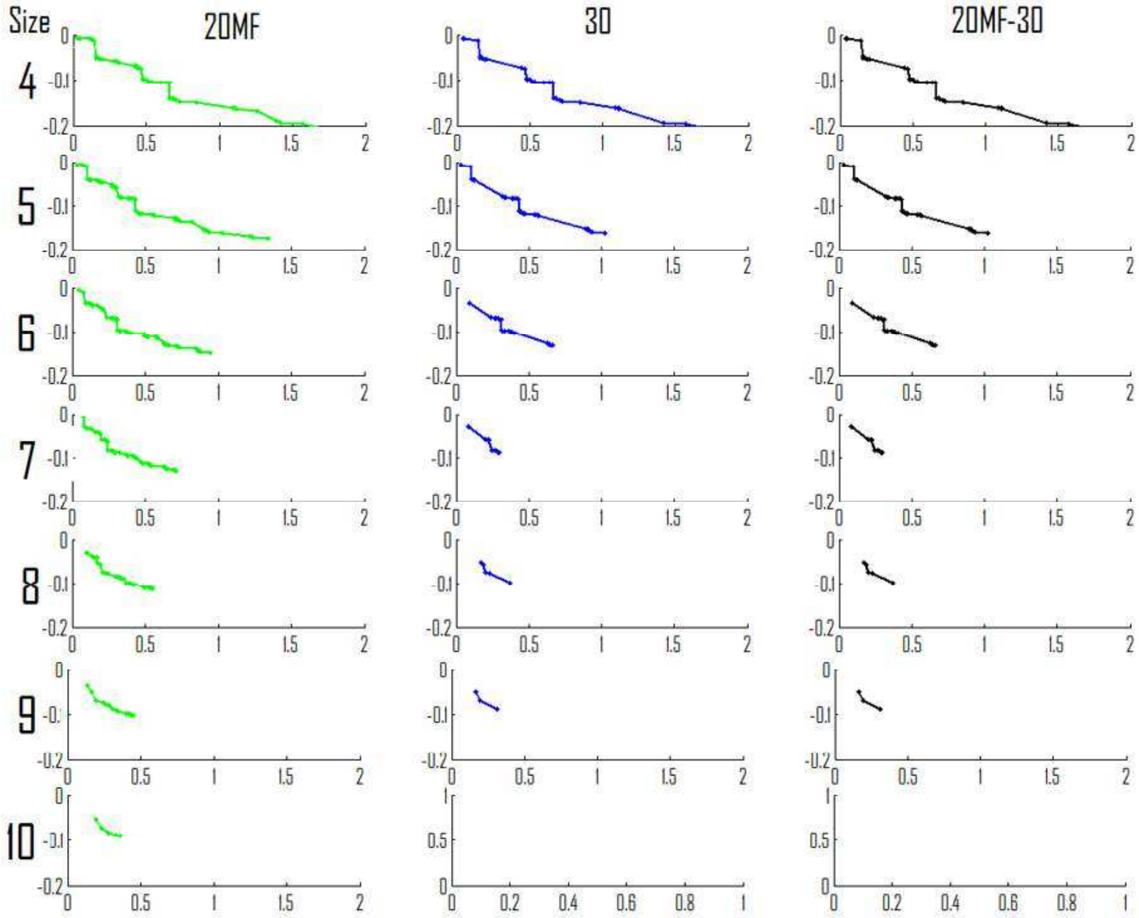
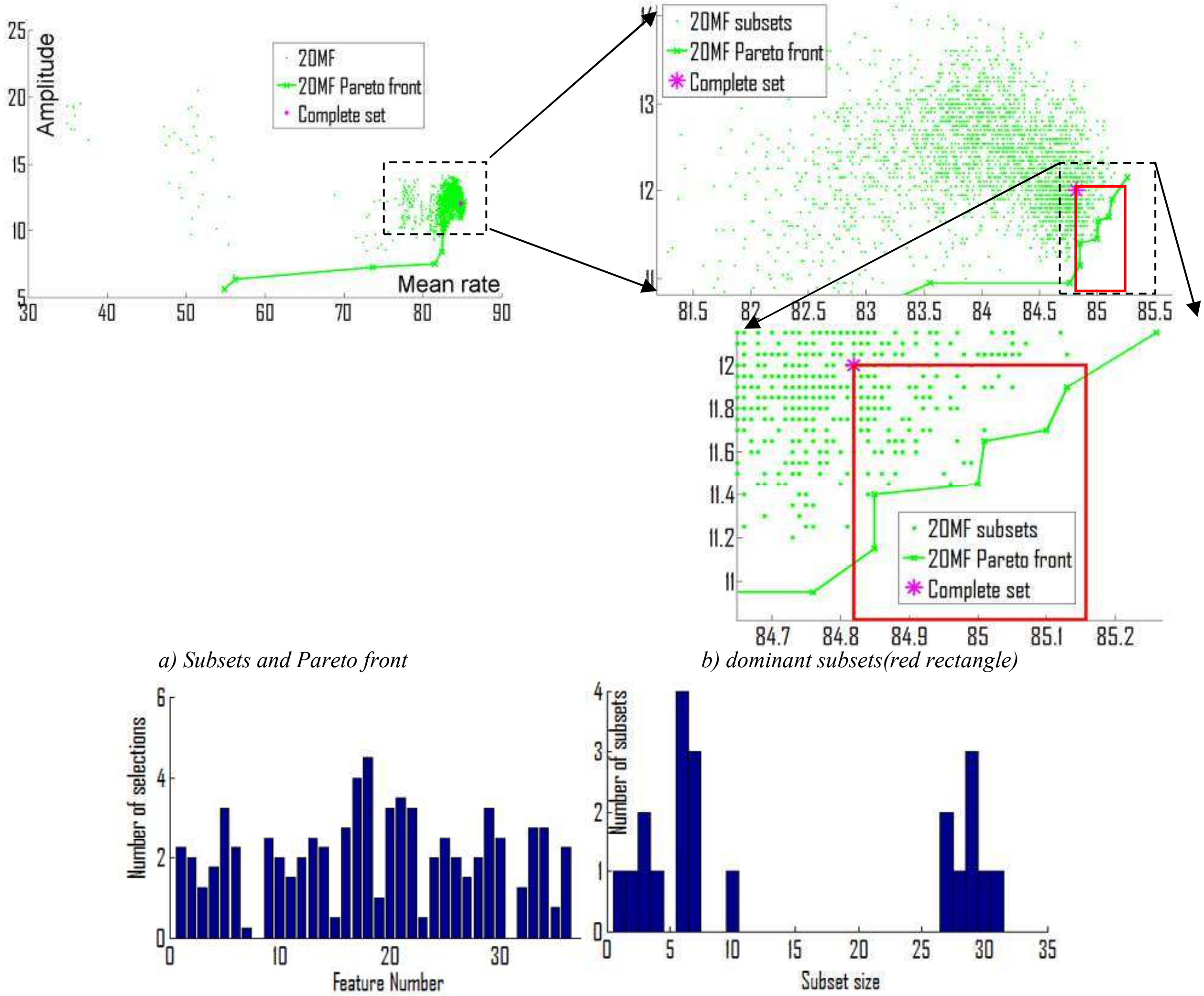


Figure 8. *filter* Pareto fronts in $(-D, R)$ space from size 4 to 10 for *wineWhite* (11 features) database during *2OMF*, *3O* and *2OMF-3O*

6.2 *2OMF* method analysis

We now analyze more precisely the results of the *2OMF* algorithm after the second step of the algorithm. This step is based on the *wrapper* approach which sorts and then selects among the retained subsets during the *filter* step. We recall that the used criterion is the stability(in a two objectives way) when different classifiers are applied. In this space (mean rate, amplitude) we compute a new Pareto front composed of several solutions and we focus on them to select the most interesting ones.

Figure 9 displays information about the stability of the selected subsets after *filter* step (green points) for *landSat* database. As showed in Figure 9(b) (which is a zoomed part of the Figure 9(a)), a lot of subsets dominates the complete set (purple star in the figure) even if they are not in the Pareto front: these subsets are within the red rectangle. All of these subsets have higher mean classification rate and lower amplitude than the complete set. They can also be interesting because some of them h ave lower number of features than the one in the Pareto Front and a quite good classification stability as it is better than the complete set stability. For the studied database, there are 21 subsets in the front (6 dominating complete set) and 73 subsets that dominate the complete set.



a) Subsets and Pareto front

b) dominant subsets(red rectangle)

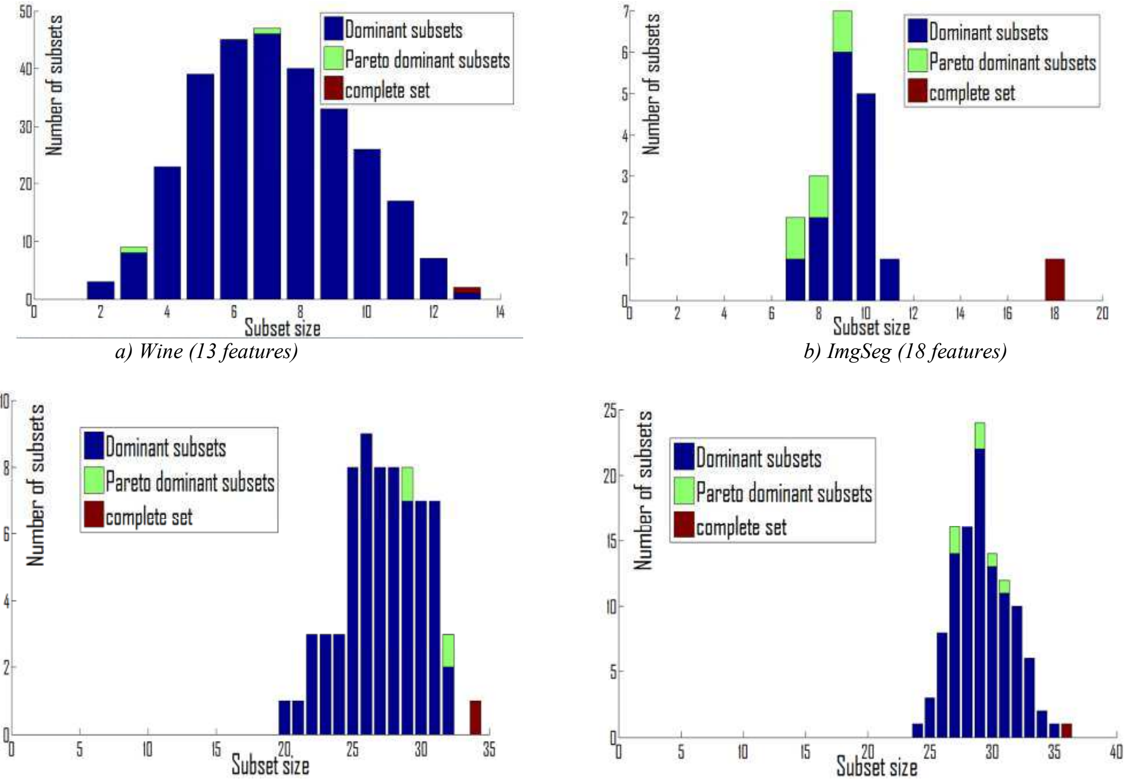
c) Histogram of selected Features in the wrapperPareto front

d)Histogram of subset size in the wrapper Pareto front

Figure 9. Stability analysis for *landSat* database

Figure 9(c) shows the histogram of the selected features computed using all subsets of the *2OMF wrapper* Pareto Front. We note that quite every features are represented. In the same way Figure 9(d) gives the repartition of the size of the Pareto subsets in order to illustrate the diversity of the solutions. Landsat database is composed of 36 features and some of the subsets in the Pareto front are composed of less than 10 features. A further analysis of the subset sizes is given in the next section.

The histograms in Figure 10show the size repartition of the subsets that dominate the complete set for *wine*, *imgSeg*, *ionosphere* and *landSat* databases. We remark that a lot of subsets have a size between 40% and 70% of the complete set size. This is an interesting result which proves that a controlled reduction of the number of features allows an improvement of the classification rate independently of the classifiers (which is in connection with the expected stability of the selected subsets). We also observe that, the subsets with the lowest size are obtained for non Pareto subsets (see Figure 10(a), 10(c) and 10(d)). This leads to a proposition of modification of the second step of the algorithm. The final set of subsets will be composed of all subsets which dominate the complete set regarding stability and not only the Pareto subsets. The choice among these subsets is then made by the user according to his own criterion (lowest number of features, best mean rate, etc.).



a) Wine (13 features)

b) ImgSeg (18 features)

c) Ionosphere (34 features)

d) landsat (36 features)

Figure 10. Histogram of the dominant subsets size for *2OMF*

6.3 Comparison between *2OMF*, *mRMR* and *FSDD*

In order to evaluate the performances of our algorithm we compare it with two well known feature selection methods: *mRMR* and *FSDD*. Figure 11 displays the visited subsets using *mRMR* algorithm (blue) with their corresponding Pareto front (blue line) and using *FSDD* algorithm (black) with their corresponding Pareto front (black line). We observe that these subsets are not Pareto optimal when compared to the *2OMF* subsets (green points). Moreover, few of them dominate the complete subset.

The same observation can be done for most of the databases. Indeed, in few databases we observe an *mRMR* subset that dominates the complete subset. A subset of *mRMR* and a subset of *FSDD* fall into the Pareto set only for TAE database and for iris. Figure 11 shows a comparison between the Pareto Fronts obtained with *mRMR* (blue), *FSDD* (black) and *2OMF* (green) algorithms for different databases. The size of the corresponding subsets is also displayed near the subset as well as the complete set (purple star). We can observe that the subset size follows high variations: between 2 and 33 for the *2OMF* Pareto front for ionosphere database and between 4 and 29 for the *2OMF* Pareto front of the *landSat* database for example. The mean classification rate is also varying in a wide range: for ionosphere database the mean rate is about 75.5% for a two features subset, 86% for a 5 features subset and 84% for a 10 features subset; for the *landSat* database the mean rate is about 82.5% for a 4 features subset and 85.5% for a 28 features subset.

We now focus on the good classification rate obtained for some interesting subsets. The Table 1 shows subsets obtained with *2OMF*, *mRMR* and *FSDD* methods. Information about the Pareto optimality is also given ($PO = Y$ in the table). In addition to this, we present whether a subset dominates the complete set ($Do = Y$). The *mRMR* and *FSDD* subsets are chosen among the visited ones according to their mean rate value.

Table 1. Good classification rate and stability of interesting subsets (Bold faces indicate the best result(s))

						Classifiers rate						
Db	Method	Subset	Size	Do.	PO	Mean	Var.	KNN	LDA	Mah	NB	PNN
Wine	Complete set	[1-13]	13		N	87.7	29.5	70.4	**100**	92.0	96.6	79.5
	Best *mRMR*	[1 7 10 11 1213]	6	**Y**	N	88.63	25.0	73.86	98.86	**97.72**	97.7	75
	Best *FSDD*	[1 6 7 10 12 13]	6	**Y**	N	88.18	25.0	73.86	98.86	97.72	95.4	75.0
	2OMF	[1 7 11]	3	**Y**	**Y**	96.2	4.5	97.7	93.2	96.6	97.7	95.4
		[1, 2, 6, 7, 8, 11, 12]	**7**	**Y**	**Y**	**94.5**	**2.2**	**95.5**	93.2	93.2	**95.5**	**95.5**
		[1 7]	**2**	**Y**	N	**94.3**	**5.6**	**96.6**	90.9	94.3	**95.5**	**94.3**
imgSeg	Complete set	[1-18]	18		N	92.3	8.2	95.5	91.1	NA	87.3	95.3
	Best *mRMR*	[. . .]	16	N	N	91.57	7.9	94.9	90.4	NA	87	94
	Best *FSDD*	[. . .]	14	**Y**	N	92.46	7.87	95.32	**91.60**	NA	87.5	95.41
	2OMF	[2 10 12 13 15 17 18]	**7**	**Y**	**Y**	**93.2**	**6.5**	**96.1**	90.9	NA	**89.5**	**96.0**
		[2 9 10 12 13 15 17 18]	**8**	**Y**	**Y**	**93.1**	**5.9**	**96.0**	90.7	NA	**90.0**	**95.6**
		[2 5 10 11 12 13 15 17 18]	**9**	**Y**	**Y**	**92.7**	**5.5**	95.2	90.3	NA	**89.8**	**95.4**
ionoSphere	Complete Set	[1-34]	34		N	76.23	29.7	87.1	NA	NA	84.2	57.4
	Best *mRMR*	[1 3 4 5 14]	5	**Y**	N	83.80	30	92.85	NA	NA	94.3	64.28
	BestFSDD	[1 3 5 7 9]	5	**Y**	N	82.38	**25.71**	91.42	NA	NA	90	**65.71**
	2OMF	[1 3 4 5 6 7 8 14 15 28]	10	**Y**	**Y**	83.80	28.57	94.28	NA	NA	91.4	**65.71**
		[1 3 4 5 7]	5	**Y**	**Y**	**85.71**	32.85	**97.14**	NA	NA	**97.7**	64.28
landSat	Complete Set	[1-36]	36		N	84.8	12	89.4	83.9	81.6	78.5	90.5
	Best *mRMR*	[. . .]	27	N	N	84.58	11.5	89.85	83.00	80.15	**79.2**	90.7
	Best *FSDD*	[. . .]	28	N	N	84.73	11.35	89.7	83.4	80.8	79.2	**90.55**
	2OMF	[. . .]	**29**	**Y**	**Y**	85.0	11.4	**89.8**	83.7	82.1	78.9	90.3
		[. . .]	**31**	**Y**	**Y**	**85.1**	11.9	89.6	83.7	83.0	78.7	**90.6**
		[. . .]	**27**	**Y**	N	84.85	11.15	89.9	83.4	81.4	**79.2**	90.35
		[. . .]	**25**	**Y**	N	84.9	11.9	89.3	83.2	82.7	78.7	**90.6**
		[. . .]	**26**	**Y**	N	84.8	11.7	**89.6**	83.2	82.3	78.6	**90.6**
		[5 13 18 20 21 29 34]	**7**	N	**Y**	83.13	**10.7**	**84.95**	82.45	**83.15**	72.2	87.9

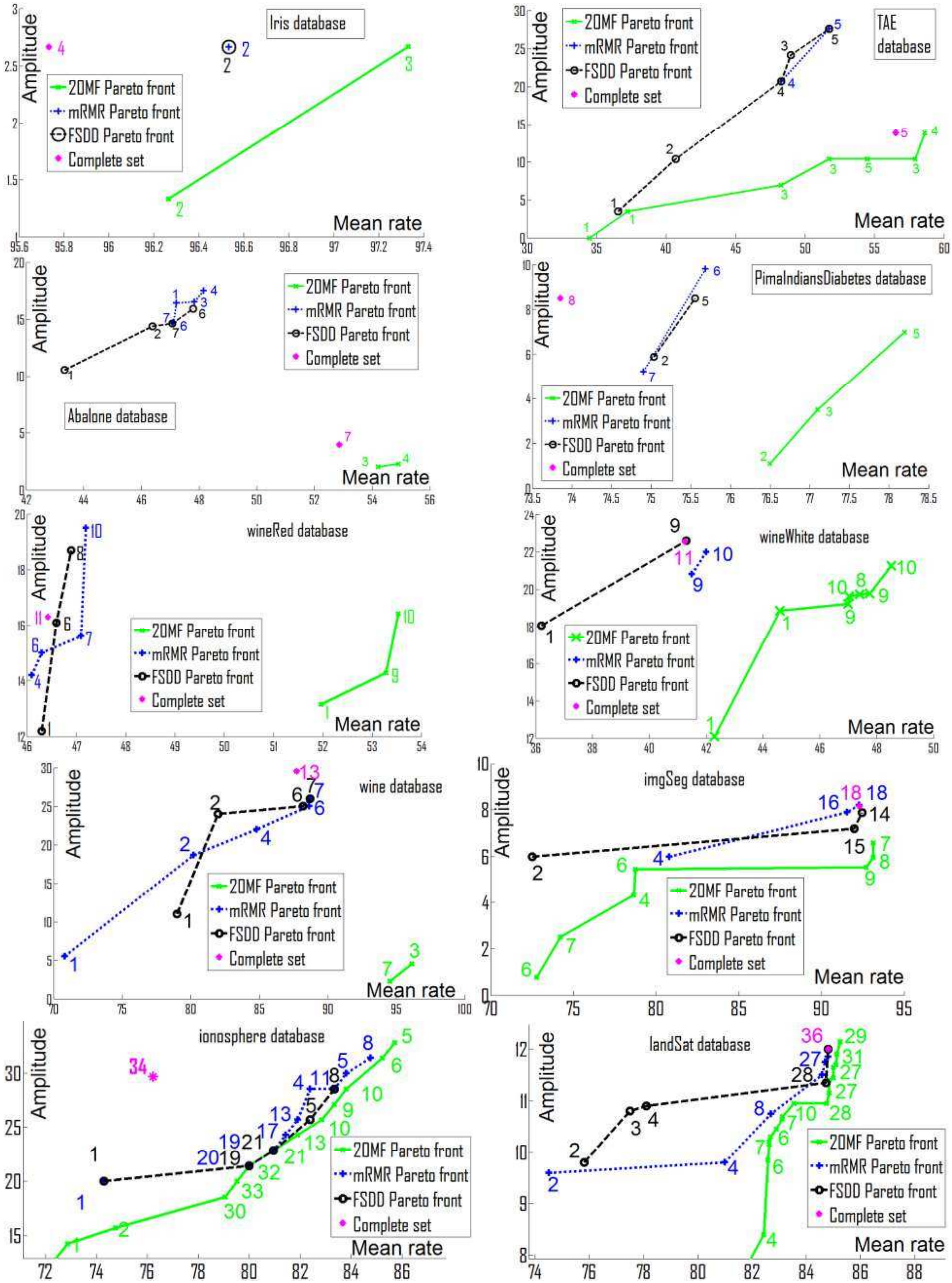


Figure 11. Comparison of the stability Pareto Fronts for *2OMF* (green), *mRMR* (blue) and *FSDD* (black) algorithms

All the displayed subsets obtained with the *2OMF* method are interesting because they have a low number of features and a better stability than the complete set. Nevertheless, some subsets have lowest number of features and others highest classification rates. For example, for the wine database, a subset with two features (features number 2 and 7) have a higher mean rate and a lower amplitude than the complete set having 13 features. Moreover, it has a higher classification rates for 4 classifiers over 5. In the same way, for *imgSeg* database the number of features is divided by 2 with the *2OMF* method.

Let us consider now the methods from the literature. For *landSat* database none of the visited subsets dominate the complete set for both *mRMR* and *FSDD*. Moreover, stable and successful subsets obtained with *FSDD* have a higher number of features than the ones obtained with *2OMF*. Only one stable subset having low number of features is obtained with *mRMR* (8 features). However, it is dominated by the subset returned by *2OMF* which has seven features (last line in the table). We always found a subset among *2OMF* subsets having a lower number of features, a higher classification mean rate and a lower classification amplitude than the best subsets returned by *mRMR* and FSD. An exception exists for TAE (Figure 11): two different subsets of size three are returned by *2OMF* and *mRMR* without domination relation between them (higher mean rate for *2OMF* and lower amplitude for *mRMR*).

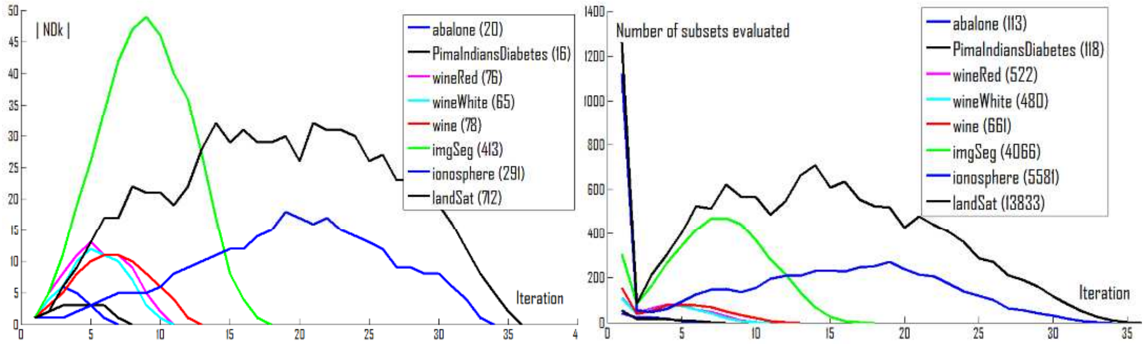
6.4 Computation complexity and computation time

In order to complete the previous comparison between *2OMF*, *mRMR* and FSDDwe study the computation complexity of the proposed method and we give some computation time measurements. For both *filter* and *wrapper* steps, we evaluate the computation complexity by counting the number of subsets.

For *filter* step the number of explored subsets is linked to the number of non dominated subsets at each iteration (ND_k). This value depends on the data and on the iteration. However, it can be estimated using simulations and verified experimentally using UCI databases.

In order to estimate the mean complexity of the *filter* step we define T the maximum number of subsets in ND_k. The computation complexity is then proportional to

$$\begin{aligned}
M(M-1)+\sum_{k=3}^{M}|ND_{k-1}|(M-k) \le M(M-1)+\sum_{k=3}^{M}T(M-k) &= M(M-1)+T\sum_{k=3}^{M}(M-k) \\
&= M(M-1)+T\sum_{k=1}^{M-3}k \\
&= M(M-1)+T\frac{(M-3)(M-2)}{2} \\
&= \left(1+\frac{T}{2}\right)M^2-\left(1+\frac{5T}{2}\right)M+\frac{5}{2}T \\
&= O\left(\frac{T}{2}M^2\right) \qquad (6.1)
\end{aligned}$$



a) $|ND_k|$ per iteration

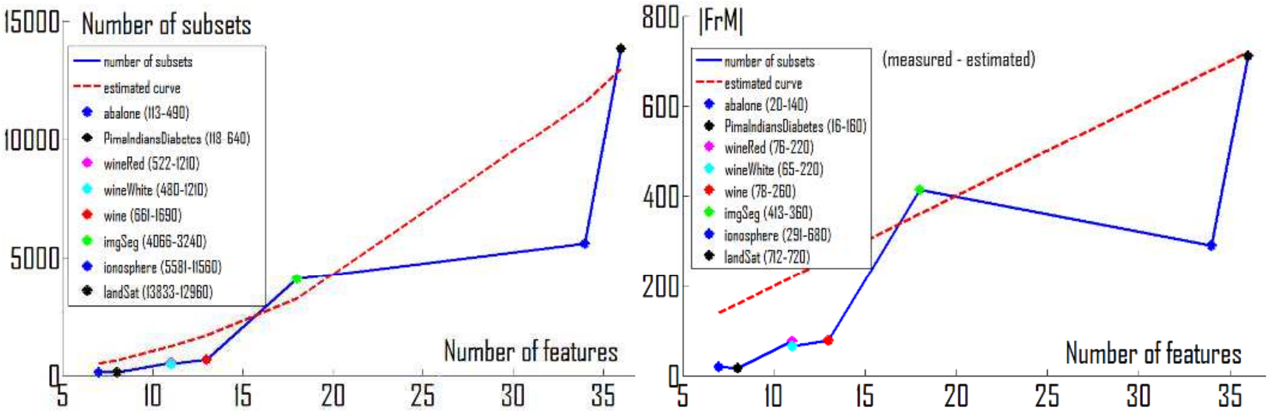
b) Number of evaluations per iteration

Figure 12. Pareto Front size and total number of evaluations

After tests on UCI databases (Figure 12 we observe that the Pareto front reaches a maximum size around $T = 50$. For large number of features we can limit the number of Non Dominated subsets at each iteration with Niching techniques [9] when the computation time must be reduced. Moreover, the computation complexity of the algorithm can also be decreased to $O\left(\frac{T}{2}KM\right)$ by limiting the cardinality of subsets at a given value $K < M$.

We run the *2OMF* algorithm on simulated data to experimentally and statistically verify the computation complexity. Figure 13(a) shows the evolution of the total number of subsets explored during the *filter* step. This value depends on the size of the input data, and also the projection of $f(M) = \frac{T}{2}M^2$ function. We also display the number of explored subsets for real databases. Figure 13(a) shows that the model proposed in equation (6.1) is close to the complexity computed on simulated and real case data.

The *wrapper* complexity is directly linked to the size of FrM : $Fr_M = \sum_{k=1}^{M} D_k \approx MT = O(TM)$. The complexity decreases to $O(TK)$ when we stop the algorithm for a given number of features K. Figure 13(b) shows the complexity corresponding to simulated and real case data. The curves provide a visual validation of the proposed evaluation of the complexity.



a) Filter

b) Wrapper

Figure 13. *Filter* and Wrapper computation complexity

The *filter* step requires the exploration of many more subsets than the *wrapper* step but the evaluation of one subset during the *filter* step is less time consuming than during *wrapper* step. Indeed, the computation of the criterions during the *wrapper* step requires running 5 classifiers. Let us notice that every MI (equation (5.1)) useful for Redundancy and Dependency computation can be computed just one time: MI between each pair of features ($I(X_i;X_j)$) and MI between each feature and the classes ($I(X_i;c)$). Then D and R computation (equation (5.3) and (5.4)) is reduced to the sum of pre-calculated values and is very fast.

Table 2 shows the computation time and complexity of the different algorithms used in the experiments. *FSDD* and *mRMR* are faster than *2OMF*. However, *2OMF* returns more stable subsets while maintaining an acceptable computation time for for non real time applications which cover a large part of the applications (storage, pre-treatment, etc.).

iris, *TAE*, *abalone*, *PimaIndiansDiabetes*, *wineRed*, wineWhite, *wine*, *imgSeg*, *ionosphere* and *landSat* databases containing 4, 5, 7, 8, 11, 11, 13, 18, 34 and 36

Table 2. Complexity and Computation time comparison (on core2 Duo 2.4 GHz)

					2OMF			
	FSDD		*mRMR*		*Filter*		*Wrapper*	
Database	*Nb subsets*	*Times*	*Nb subsets*	*Times*	*Nb subsets*	*Times*	*Nb subsets*	*Times*
Abalone	7	< 1s	21	< 1s	113	<1s	20	~5mn
PimaIndiansDiabetes	8	< 1s	28	< 1s	118	<1s	16	~5mn
wineRed	11	< 1s	55	< 1s	522	<1s	76	~25 mn
wineWhite	11	< 1s	55	< 1s	480	<1s	65	~20 mn
Wine	13	< 1s	78	< 1s	660	<1s	78	~25 mn
imgSeg	18	< 1s	153	~1s	4066	<1s	413	~2H10mn
Ionosphere	34	< 1s	561	~4s	5581	<1s	291	~1H25mn
Landsat	36	< 1s	630	~5s	13833	~1s	712	~3H30mn

7. Conclusion

This paper presents a two steps algorithm for feature selection. The algorithm begins with a *filter* step based on a forward approach. The forward approach is applied in order to quickly select a first pool of subsets in a Multi-Objectives and Multi-Fronts way (*2OMF*). The subsets are evaluated using the Dependency (D) and the Redundancy (R) of the features (*2O* for two objectives). One front is kept for each subset size which produces Multi-Fronts (MF). Then a second step is applied to measure the performances of the subsets. This step is based on a *wrapper* approach with the use of several classifiers (KNN, LDA, Mah, NB, PNN). Then the selection of the interesting subsets is performed using the stability of the subsets. The stability is evaluated with the mean and amplitude of the classification rates. From our experimentations, it is observed that the interesting subsets dominate the complete set regarding both objectives. The use of the stability to select the subsets leads to robust results which are very interesting for some applications such as in biology where the stability of the subsets is more important than its raw classification rate. The *wrapper* step is required because some subsets of the *filter* Pareto front could have a higher classification rate than the complete set for a given classifiers but not for another one, as shown in the paper. A selection of features only based on a *filter* method does not ensure that the selected subset will improve classification rates for a large set of classifiers.

The results are very convincing for all tested databases. The subsets obtained after applying our algorithm have lower number of features and better classification performances compare to the complete set of features. Moreover, the diversity of the final pool of subsets allows selecting a subset adapted to a specific application (good classification expected or reduction of a high number of features). We also compared the proposed algorithm with two feature-selection methods (*mRMR* and *FSDD*). It is observed that our method outperforms the other tested methods in almost all cases.

One of the major goals of this research is to reduce the computation time of classification task. This goal is achieved by selecting a set of feature subsets returned by the *filter* step. The selection is made according to the composition of the subsets in order to keep a diversified pool of subsets. In future works, we will use techniques inspired from the one used in Multi Objectives Genetic Algorithms to select diversified non dominated offspring among the resulting population after the crossover operator. This technique is called NPGA (Niched Pareto Genetic Algorithm [9]). We will also compare the Redundancy and Relevance criterions with other ones in the *filter* step and add more classifiers in the *wrapper* step to select more robust subsets.

Acknowledgment

The authors want to thanks the European institutions and more precisely the INTERREG IV program within the CESAR Part II project leads by the University of Antilles and Guyana and the Poitou-Charentes region.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed equally and significantly in writing this article. All the authors read and approved the final manuscript.

References

[1] M. Abadi, E. Grandchamp, O. Alata, O. Olivier and M. Khoudeir, Information criteria performance for feature selection, in *Proceedings of the 4th International Congress on Image and Signal Processing*, Vol. 2, Shangay: Chine, October 2011, pp. 919–923.

[2] A. Al-Ani, M. Deriche and J. Chebil, A new mutual information based measure for feature selection, *Intelligent Data Analysis* **7** (1), 43–57, 2003.

[3] E. Cantu-Paz, Feature subset selection, class separability, and genetic algorithms, in *Genetic and Evolutionary Computation*, 2004, pp. 959–970.

[4] H. Chouaib, O. Ramos-Terrades, S. Tabbone, F. Cloppet and N. Vincent, Feature selection combining genetic algorithm and Adaboost classifiers, in *19th International Conference on Pattern Recognition - ICPR*, Tampa, USA, 2008.

[5] S. Das, Filters, wrappers and a boosting-based hybrid for feature selection, in *Proceedings of the Eighteenth International Conference on Machine Learning*, San Francisco, CA, USA, 2001, pp. 74–81.

[6] L. Davis, *Handbook of Genetic Algorithms*, L. Davis (Ed.), New York: Van Nostrand Reinhold, 1991.

[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley and Sons, Chichester, 2001.

[8] J. Dy, C. Brodley, A. Kak, L.S. Broderick and A.M. Aisen, Unsupervised feature selection applied to content-based retrieval of lung images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25** (3), 373–378, 2003.

[9] C. Emmanouilidis, A. Hunter and J. MacIntyre, A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator, in *Proceedings of the Congress on Evolutionary Computation*, California, July 2000, pp. 309-316.

[10] C. Emmanouilidis, A. Hunter and J. MacIntyre, A multi-objective genetic algorithm approach to feature selection in neural and fuzzy modeling, *Evolutionary Optimization* **3** (1), 1–26, 2001.

[11] G. Forman, An extensive empirical study of feature selection metrics for text classification, *Journal of Machine Learning Research* **3** (1), 1289–1305 2003.

[12] J.Q. Gan, S.H. Bashar Awwad and C.S.L. Tsui, A hybrid approach to feature subset selection for brain-computer interface design, in *Intelligent Data Engineering and Automated Learning - IDEAL 2011*, Vol. 6936, 2011, pp. 279–286.

[13] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st edition, London: Addison-Wesley Longman, 1989.

[14] R. Günter, Convergence analysis of canonical genetic algorithms, *IEEE Transactions on Neural Networks* **5**, 96–101, 1994.

[15] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *Journal of Machine Learning Research* **3** (1), 1157–1182, 2003.

[16] I. Guyon et al. (Eds.), *Feature Extraction: Foundations, and Applications*, 2006.

[17] J. Handl and J. Knowles, Feature subset selection in unsupervised learning via multiobjective optimization, *International Journal of Computational Intelligence Research* **2** (3), 217–238, 2006.

[18] B.A.S. Hasan and J.Q. Gan, A Multi-objective particle swarm optimization for channel selection in brain-computer interfaces, in *The UK Workshop on Computational Intelligence - UKCI*, 2009.

[19] B.A.S. Hasan, J.Q. Gan and Z. Qingfu, Multi-objective evolutionary methods for channel selection in brain-computer interfaces: some preliminary experimental results, in *IEEE Congress Evolutionary Computation*, Barcelona, Spain, July 2010, pp. 1-6.

[20] M. Hilario and A. Kalousis, Approaches to dimensionality reduction in proteomic biomarker studies, *Briefings in Bioinformatics* **9** (2), pp. 102–118, 2008.

[21] F. Hussein, R. Ward and N. Kharma, Genetic algorithms for feature selection and weighting, a review and study, in *International Conference on Document Analysis and Recognition*, 2001, pp. 1240–1244.

[22] L.B. Jack, Feature selection for ANNs using genetic algorithms in condition monitoring, in *European Symposium on Artificial Neural Networks - ESANN*, Bruges, Belgium, 1999, pp. 313–318.

[23] O.A. Jadaan, L. Rajamani and C.R. Rao, Non-dominated ranked genetic algorithm for solving multi-objective optimization problems: NRGA, *Journal of Theoretical and Applied Information Technology*, 60–67, 2008.

[24] A.K. Jain, R.P.W. Duin and J. Mao, Statistical pattern recognition: a review, *IEEE Trans. Pattern Analysis and Machine Intelligence* **22** (1), 4–37, 2000.

[25] J. Jarmulak and S. Craw, Genetic algorithms for feature selection and weighting, in *Proceedings of the workshop on Automating the Construction of Case Based Reasoners*, 1999, pp. 28–33.

[26] R. Jensen, Performing Feature Selection with ACO, *Studies in Computational Intelligence* **34**, pp. 45–73, 2006.

[27] L. Jourdan, C. Dhaenens and E-G. Talbi, A genetic algorithm for feature selection in data-mining for genetics, in *Metaheuristic International Conference* 2001, Porto, Portugal, July 2001, pp. 29–34.

[28] A. Kalousis, J. Prados and M. Hilario, Stability of feature selection algorithms: a study on high-dimensional spaces, *Knowledge and Information Systems* **12** (1), 95–116, 2007.

[29] R. Kohavi and G. John, Wrapper for feature subset selection, *Artificial Intelligence* **97** (1-2), 273–324, 1997.

[30] I. Kononenko, Estimating attributes: analysis and extensions of relieF, in *Proceedings of ECML-94*, 1994, pp. 171–182.

[31] L.I. Kuncheva, A stability index for feature selection, in *Proceedings of the 25th Conference on Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, Innsbruck, Austria, February 2007, pp. 390–395.

[32] N. Kwak and C.H. Choi, Input feature selection by mutual information based on parzen window, *IEEE Trans. Pattern Anal. Mach. Intell* **24** (12), 1667–1671, 2002.

[33] J. Liang, S. Yang and A-C. Winstanley, Invariant optimal feature selection: A distance discriminant and feature ranking based solution, *Pattern Recognition* **41** (5), 1429–1439, 2008.

[34] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[35] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering, *IEEE Transactions on Knowledge and Data Engineering* **17** (4), 491–502, 2005.

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

[37] G.L. Pappa, A.A. Freitas and C.A.A. Kaestner, Attribute selection with a multi-objective genetic algorithm, in *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, London, UK, 2002, pp. 280-290.

[38] G.L. Pappa et al., A multiobjective genetic algorithm for attribute selection, in *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*, Nottingham Trent University, December 2002, pp. 116–121.

[39] E. Parzen, On estimation of a probability density function and mode, *The Annals of Mathematical Statistics* **33** (3), 1065–1076, 1962.

[40] H. Peng, F. Long and C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27** (8), 1226–1238, 2005.

[41] F. Pernkopf and P. O'Leary, Feature selection for classification using genetic algorithms with a novel encoding, in *Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*, London, UK, 2001, pp. 161–168.

[42] A. Porebski, N. Vandenbroucke and L. Macaire, Comparison of feature selection schemes for color texture classification, in *Proceedings of the International Conference on Image Processing Theory, Tools and Applications*, 2010, pp. 32-37.

[43] P. Pudil and J. Novovicova, Novel methods for subset selection with respect to problem knowledge, IEEE Intelligent Systems **13** (2), 66–74, 1998.

[44] S. J Raudys, Feature over-selection, Structural, Syntactic, and Statistical Pattern Recognition **LNCS 4109**, pp. 622–631, 2006.

[45] Y. Saeys, T. Abeel and Y. Peer, Robust Feature Selection Using Ensemble Feature Selection Techniques, in Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II, Antwerp, Belgium, 2008, pp. 313-325.

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

[47] Y. Sakamoto and H. Akaike, Analysis of cross classified data by AIC, *Annals of the Institute of Statistical Mathematics* **30** (1), 185–197, 1978.

[48] M. Sebban and R. Nock, A hybrid filter wrapper approach of feature selection using information theory, *Pattern Recognition* **35** (4), 835–846, 2002.

[49] J. Sheinvald, B. Dom and W. Niblack, A modeling approach to feature selection, in *Proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, NJ, Jun 1990, pp. 535–539.

[50] A. Solanas et al., Feature selection and outliers detection with genetic algorithms and neural networks, in *Proceedings of the Conference on Artificial Intelligence Research and Development*, Amsterdam, The Netherlands, 2005, pp. 41–48.

[51] P. Somol, J. Novovicova and P. Pudil, Efficient feature subset selection and subset size optimization, in *InTech-Open Access Publisher* **56**, 2010, pp. 1–24.

[52] P. Somol, J. Novovicova and P. Pudil, Flexible Hybrid sequential floating search in statistical feature selection, in *Proceedings of the International Conference on Structural, Syntactic, and Statistical Pattern Recognition*, Joint IAPR, Hong Kong, China, 2006, pp. 632–639.

[53] P. Somol, J. Novovicova and P. Pudil, On the over fitting problem of complex feature selection methods, in *Proceedings of the 5th International Computer Engineering Conference*, Cairo University, December 2009.

[54] Y. Sun, S. Todorovic and S. Goodison, Local-learning-based feature selection for high-dimensional data analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32** (9), 1610–1626, 2010.

[55] H. Vafaie and I.F. Imam, Feature selection methods: genetic algorithms vs greedy-like search, in *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems*, 1994.

[56] P. Villar, A. Fern and F. Herrera, A genetic algorithm for feature selection and granularity learning in fuzzy rule-based classification systems for highly imbalanced data-sets, in *IPMU* (1), 2010, pp. 741–750.

[57] D. Wettschereck and D.W. Aha, Weighting features, in *Proceedings of the First International Conference on Case-Based Reasoning*, 1995, pp. 347–358.

[58] Y.Y. Yao, Information-theoretic measures for knowledge discovery and data mining, in *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, 2003, pp. 115–136.

[59] H. Zhang and G. Sun, Feature selection using tabu search method, *Pattern Recognition* **35**, 701–711, 2002.

[60] L. Zhuo, J.and Wang, F. Zheng, X. Li, B. Ai and J. Qian, A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine, in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* Part B7 Vol. **XXXVII**, Beijing, China, 2008, pp. 397–402.