International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 6 (2014) pp. 333 - 343
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Fast Dual Selection using Genetic Algorithms for Large Data Sets

Frédéric Ros[1], Rachid Harba[1] and Marco Pintore[2]

[1] Orleans University,
Laboratoire Prisme Orléans, France
frederic.ros@univ-orleans.fr

[2] Orleans University,
Laboratoire Prisme Orléans, France
rachid.harba@univ-orleans.fr

[3] PILA,
Saint Jean de la Ruelle, France
Marco.pintore@yahoo.fr

Abstract: **This paper is devoted to feature and instance selection managed by genetic algorithms (GA) in the context of supervised classification. We propose a GA encoded by binary chromosomes having the same size as the feature space for selecting features in which each evaluated chromosome delivers a set of instances. The main aim is to optimize the processing time, which is particularly problematic when handling large databases. A key feature of our approach is the variable fitness evaluation based on scalability methodologies. Experimental results indicate that the preliminary version of the proposed algorithm can significantly reduce the computation time and is therefore applicable to high-dimensional data sets.**

Keywords: instance and feature selection, scaling, genetic algorithms**,** k nearest neighbors.

I. Introduction

In a growing number of domains, increasingly large data sets have to be handled and the data captured encapsulate many features. The need is then to extract relevant information in order to allow a decision or exploration support, and more than just delivering an accurate predictive model, to produce interesting knowledge for the expert domain. There is a major drawback in building and using a model in which the user cannot readily comprehend the final rules that define this model. In order to reduce the dimensionality of the feature space, the selection of informative features becomes an essential step towards the classification task as it contributes to understanding the phenomena leading to the information contained in the databases.

In this case, the descriptive aspect is as important as the predictive one, and implies paying particular attention to interpretability of the results.

Dual selection, i.e. selection of relevant features [1] and a family of instances [2] attached to these features, contributes to these two complementary objectives but is difficult to solve as it is a combinatorial optimization problem. This field of research, at the intersection of pattern recognition and soft computing, has been probably one of the most intensively studied in recent decades [3-8].

The problem of feature selection in the context of supervised classification is to determine, from an original set of features, one or more subsets to reduce the number of workspace dimensions in order to improve the quality of data. Unlike feature transform, the fewer dimensions obtained by feature selection facilitate the exploration of results in data analysis and generally lead to higher classification accuracy. Techniques involving linear transformations of the original pattern vectors to new vectors of lower dimensionality are interesting especially for database visualization. They contribute only partially however to a better understanding and after all do not reduce the number of features that must be measured. Feature selection has now been widely applied in many domains during the last twenty years.

So-called irrelevant and superfluous features are discarded, giving subsets of the original features which retain sufficient information to discriminate well among classes.

Feature selection approaches are often segmented into filter and wrapper methodologies [9-10]. Filter methods consist in evaluating the variables individually, ordering them and selecting a subset. These methodologies generally differ in the evaluation itself and in the way the feature subset is selected. The evaluation criterion is not necessarily or directly related to the classification criterion. In wrapper methodologies, the selection of variables is directly related to the classification purpose: the method searches for the combination of a subset of variables that optimizes the classification accuracy.

Filter methods are fast and rather approximate while wrapper methods are generally more accurate but time-consuming. Wrapper approaches require heuristic search techniques when the feature space dimension is non-trivial. Some studies hybridize the two approaches [10-12].

The concept of instance selection extends that of nearest neighbor algorithms [13]. Given a database S with training samples, whose class label is known, instance selection consists in choosing a small region S_z of the available data such that the classification accuracy of C_{1nn} (a nearest classifier) over S is maximal. These condensation methods usually seek to select representative instances. The Condensed Neighbor Rule (CNN) proposed by Hart [14], Reduced nearest neighbor (RNN) introduced by Gates [15], and the Edited Nearest Neighbor (ENN) algorithm developed by Wilson [16] are the pioneering approaches from which many alternative approaches such as the Iterative Case Filtering algorithm (ICF) [17], Ib3[2], and the popular DROP family have evolved [18]. Starting from the original set, these last methods consist in removing patterns step by step in an ordered way to obtain the final set. An item is removed if without it its neighbors can be well classified.

This field of research is still being investigated but is considered as mature. Everybody agrees on the importance of understanding the relationship between features and instances. It is also widely admitted that better models can be determined by removing noise, irrelevant and redundant features and instances, and reducing the overall dimensionality of a data set. We should however mention two points:

(i) While feature and instance selection have been studied independently and exhibited remarkable results, research on dual selection is much less extensive, despite its inherent interest for understandability and interpretability.

(ii) Several efficient methodologies have been developed for relatively small databases. Applications dealing with large data sets and features have emerged in different areas.

New user-friendly approaches are particularly expected to give efficient results within a reasonable time when handling large databases.

For long time, greedy algorithms [19] were the most widely used but other approaches such as evolutionary algorithms have been investigated. Genetic algorithms (GAs) [20] have proved their ability to solve problems where conventional techniques had failed. They are sometimes applied for instance selection, but more often for feature selection, especially for treating wrapper methodologies, which are the most time-consuming. To use GAs, a nonparametric classifier is generally considered (neighborhood approaches, neural networks, etc.), associated with a binary chromosome that represents a solution. The genetic algorithm will evolve a family of chromosomes whose fitness function is computed by the classifier.

The implementation of an evolutionary algorithm for dual selection faces a number of challenges. The space of possible subsets is very large, and the risk of converging onto local, unsatisfactory sub-optima is relevant.

Using GAs is not straightforward: the precise composition of the algorithm cannot be determined at a general level, making its exploitation rather difficult. Parameter tuning and control often have to be determined empirically, and the process can be very time-consuming. Choosing suitable values is still a challenging issue but essential to find a good tradeoff between exploration and exploitation.

Clearly, computation time is the key issue when GAs are applied. Despite the availability of promising mechanisms to make GAs very efficient, expert users in many domains often prefer worse approaches than GAs simply because they are more usable. By reducing the computation time, it is possible to integrate more sophisticated mechanisms that contribute to making GAs more efficient. This is particularly true for large databases.

While we are very concerned by the mechanisms to make GAs more efficient, the innovative feature of the present study lies in the optimization of the resources used by the algorithm to reduce the computation time when dealing with non-trivial databases. When analyzing time sharing in a GA framework, it becomes clear that most of the time is spent on evaluating the fitness function itself. Since the need for accuracy varies during the genetic life, there is substantial room for optimizing this aspect. We propose to greatly reduce the cost of the fitness function by using adaptive scaling methods to achieve the fitness function. This paper describes the preliminaries of our approach and initial results appear to be promising.

The remainder of the paper is organized as follows. Section 2 reviews GAs and scaling approaches that are the bricks of our proposal. Section 3 presents our hybrid approach and section 4 details the experimental results. Concluding remarks are given in Section 5

II. Background and related work

A. Genetic Algorithms

Review of the basics

GAs can be seen as powerful techniques miming natural reproduction. To solve a classification problem, a single solution via a fitness function must be presented in a single data structure. GAs will create a population of solutions based on the sample data structure proposed. In fact, they work on the basis of a set of candidate solutions. Each candidate or chromosome represents a trial solution of the problem posed and is a member of the population. For a recent review see the work by Yu *et al*. [21].

GAs are general-purpose search algorithms that use principles inspired by natural genetics to evolve solutions to problems [22]. The basic idea is to maintain a population of chromosomes (representing candidate solutions to the concrete problem being solved) that evolves over time through a process of competition and controlled variation. During successive iterations, called generations, chromosomes in the population are rated for their adaptation as solutions, and on the basis of these evaluations, a new population of chromosomes is formed using a selection mechanism and specific genetic operators such as crossover and mutation. An evaluation or fitness function must be devised for each problem to be solved. Given a particular chromosome, the fitness function returns a single numerical value, which is assumed to be proportional to the utility or adaptation of the solution represented by that chromosome.

The idea is that individuals or chromosomes that fit the environment best should have a better chance to propagate their offspring. Solutions that have the best fitness should receive higher probability to search their neighbors. The number of papers reporting applications of GAs to real problems and the number of scientists and disciplines using them have been growing exponentially. Several tutorials about

GAs have been published in journals devoted to different research fields [23].

Critical points using GA

The set-up of the structure of a GA is a very critical point, and a guide leading to a good architecture is highly beneficial. There are two primary factors in the search carried out by a GA: population diversity and selective pressure [24]. In order to have an effective search there must be a search criterion (the fitness function) and a selection pressure that give individuals with higher fitness a higher chance of being selected for reproduction, mutation, and survival. Elitism or selection pressure favors the partial or full reproduction of the best chromosomes. Without selection pressure, promising regions of the search space would not be favored over regions offering no promise. The search process becomes random and the algorithm cannot converge.

On the other hand, population diversity is crucial to a GA's ability to continue the fruitful exploration of the search space [24-25]. If the lack of population diversity takes place too early, a premature stagnation of the search is caused. Under these circumstances, the search is likely to be trapped in a region not containing the global optimum. This problem, called premature convergence, has long been recognized as a serious failure mode for GAs [26]. Too much selection pressure increases the exploitation and the probability of making the population homogeneous sooner, rather than later. This could diminish the ability of the reproductive operators to produce variation in the population and could decrease the likelihood of converging to a global optimum.

New trends for more operational GAs

An advantage of presenting an approach using GAs is that efficient solutions can be obtained even for complicated optimization problems involving large and complex search spaces. The specialized literature has reported numerous papers relating different problems that have been successfully solved by a GA where conventional techniques had failed [27]. Despite these undeniable successes, applying GAs to a dedicated problem is not straightforward, and objectively their implementation and use face various problems. The extensive use of parameters leaves end-users with a large choice of setting and running parameters to find, which is very often time consuming and even sometimes inefficient. The challenge is not in the search for new concepts but in how to adaptively embed existing mechanisms in the algorithm so as to make the method operational for non-specialist users.

B. *Related work*

Dual selection via GAs

The most natural and straightforward way to combine feature and instance selection is to perform one process after the other. Clearly, this solution cannot be optimal as the different objectives are not independent. There has been a lot of work using genetic and evolutionary algorithms for feature and instance selection. Since the preliminary work by Shalak [28] and Kuncheva [29], several studies [30-32] have addressed this topic using genetic and evolutionary algorithms.

The general idea is to maximize the performance of the 1-nn classifier and minimize both the number of features and instances. The solution to these two problems is designed via the use of a chromosome of length f + p, which is the vector of all features (f) and all instances (p), and then running a GA to solve these problems.

In [33], we investigated a method based on a hybrid genetic algorithm combined with a local optimization procedure. Some concepts were introduced to promote both diversity and elitism in the genetic population. The instance selection aims to remove noisy and superfluous instances and selects among the others only the most critical ones.

The GA is based on self-controlled phases with dedicated objectives combining crowding and elitist strategies. Elitism and pressure preservation are reinforced by a mechanism involving a breaking process and an evolutionary memory. The genetic exploration is driven by an aggregative fitness assignment strategy. The GA is hybridized via forward and backward local procedures. The hybridization is structured in such a way that the classifier tractability and efficiency are optimized. Some neighborhood concepts related to the instance nature are also incorporated in the local procedures. By progressively filtering useless and noisy instances they contribute to facilitating and improving the natural selection of GA. While this encoding is interesting and has proved to give satisfactory results, beyond a certain chromosome size it is no longer manageable and processing responses are too slow. This slowness is particularly due to the time needed to evaluate the fitness function but also to explore all the possible combinations: all k elements among f+p (C_{f+p}^{k}).

Following on from our preliminary research in this area, we were therefore interested in scaling algorithms [34-35] to speed up the processing time for instance selection.

Scalability methodologies

Scalability methodologies are related to data partitioning. They aim at reducing the computation cost without significantly degrading the classifier performances. They involve breaking the data set into regions, learning from one or more of the regions, and possibly combining the results by cumulating the selected instances. The idea is to find instances in small regions instead of all over the training set, which can greatly improve the runtime. Existing methods differ in how they divide the data up and recombine the elementary results. A recent review can be found in [36].

In [37], the authors proposed a new fast instance selection method for large data sets, based on clustering. It selects border instances and some interior instances. They suggest dividing the training set into regions in which instances are selected. The concept of this approach appears promising. However, the authors did not handle the clustering problem itself (cluster number, convergence…). An alternative approach consists in applying the divide-and-conquer principle [38] for scaling up instance selection algorithms. This approach substantially reduces the number of instances, but does not address the tractability issue.

In [39], the work was motivated by the following observations: patterns concerned by the decision borders between categories are generally relatively small compared to the whole population. Condensation algorithms are very costly (complexity of $0(n^2)$), which disqualifies them for non-trivial size problems. Conversely, only calculations involving local patterns around the final instances are critically needed. Therefore, there is considerable room for reducing the level of computations generated by the classical approach. While this proposal follows the classical stratification schematic, a novel feature is that instances are determined by applying condensation algorithms only on useful and small pattern sets. This field of research is relatively recent. Even if different conceptual approaches are available, the following expectations are common to all: (i) better control of the balance between the run time and classification performances, (ii) reducing the input parameters of the algorithms.

III. Our method

Our GA consists in studying the evolution of a family of binary chromosomes having the same size as the feature space. Each chromosome presents a solution that is evaluated on its ability to discriminate the classes present. The evaluation function is examined through the neighborhood by a C_{1nn} classifier. This classifier is an instance selection (IS) algorithm that is optimized by integrating scaling methodologies. This means that the output of the fitness function is a set of features and a set of instances.

Using scaling methodologies greatly reduces the exploratory search, which is limited to the feature space. However, it requires more time as an instance selection algorithm has to be performed at each step instead of taking a random selection of patterns. The aim is to reduce the processing time as much as possible by using scaling methodologies. To achieve this, we propose to adapt the level of scaling approaches to the genetic advances. For this, we investigate new hybrid algorithms for instance selection in the context of supervised classification adapted to databases including several thousand patterns.

A. Main elements of the GA

The approach proposed here is not restricted to a particular type of GA, as it addresses the problem of reducing processing time that is common to all approaches. For better interpretability of the results, we propose a relatively simple GA and mention different mechanisms that can be added to improve the overall results.

We can however identify different processing steps in the GA (Fig. 1) where the accuracy of the fitness function is not of equal importance.

At least two levels can be distinguished, which can be divided into several sublevels.

In the first one, which is rather exploratory (exploration phase), we will give the GA every opportunity to identify areas of the feature space that are promising for classification. In this exploratory phase, the genetic advance is based on a RTS (Restricted Tournament Scheme) scheme [40] and the population can progress via a large number of chromosomes. Each chromosome is roughly evaluated since the purpose is only to detect weak signals. The worst features, i.e. those which are selected with a low frequency, are then discarded from the candidate set. This filter contributes to making the GA selection easier and particularly faster.

The second level is a convergence phase and an elitist approach is preferred here to select an accurate solution; the elitist model guarantees that the chromosome with the highest fitness value is always replicated in the next generation of chromosomes. Hence, the function of maximal fitness is a monotonous increasing function.

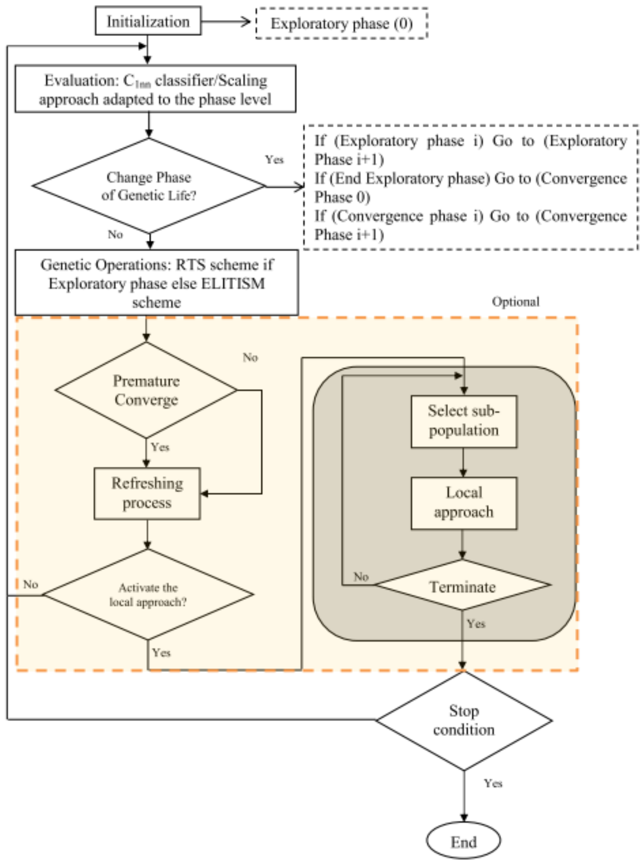

Initialization
Exploratory phase (0)
Evaluation: C1nn classifier/Scaling approach adapted to the phase level
Change Phase of Genetic Life?
Yes
No
If (Exploratory phase i) Go to (Exploratory Phase i+1)
If (End Exploratory phase) Go to (Convergence Phase 0)
If (Convergence phase i) Go to (Convergence Phase i+1)
Genetic Operations: RTS scheme if Exploratory phase else ELITISM scheme
Optional
Premature Converge
No
Yes
Refreshing process
No
Activate the local approach?
Yes
Select sub-population
Local approach
No
Terminate
Yes
Stop condition
Yes
End

Figure 1: Flowchart of the hybrid GA

The idea is to apply for each candidate chromosome encoded for feature selection a condensation algorithm IS that performs the twofold selection within a reasonable space dimension. To speed up the process, IS is applied to subsets of the training set whose dimension is related to the genetic advances. Concerning the genetic aspect, different mechanisms are investigated to favor the search and obtain promising regions for convergence.

Chromosome encoding and validity

Instead of considering a chromosome representing the whole solution f+p, we adopt a chromosome of size f that is strategically better. A chromosome is classically represented by a binary string with a zero or one denoting whether the corresponding feature is to be selected or not.

There is no point in handling a chromosome in which the majority of features is active. Each time a new chromosome is generated, its validity is checked by counting its number of ones. If this number exceeds a given threshold max_f, some of them are randomly removed to reach this threshold.

Initial population

When there is no prior domain knowledge available, it is time consuming or difficult to determine which of the available features are likely to distinguish the present categories. In this case, the only solution is to create the initial population randomly. If prior domain knowledge is available, the idea is to mix the afforded features with random ones.
For both cases, the initial population was randomly generated with chromosomes having less than max_f active features.

Fitness function

The performance to be achieved is multi-objective, and consists in finding the smallest subset of the original feature set such that the classification accuracy of C_{1nn} (a nearest classifier) over the original pattern set Z is maximal.
Then, the fitness function F for a chromosome Y simply takes into account both the classifier accuracy and the size of the feature subset.

$$F = \lambda * S + (1- \lambda) * |Y|/n \qquad (1)$$

where |Y| denotes the number of active features in y, S the classifier score, n the number of all features, and λ controls the relative importance of the criteria (λ = ¾ in our work).

F is based on an aggregative scheme as this is the simplest type. The alternative of a Pareto scheme [41] is worth mentioning, however, for future versions. In a Pareto selection scheme, one chromosome dominates another if and only if its fitness is higher than the other's according to at least one criterion and as good as the other's according to the rest of the criteria. The non-dominated chromosomes of the population are called the Pareto front representing an indistinguishable set of solutions. In an aggregative scheme, the score of each criterion is weighted according to its relative importance. We suggest an aggregative scheme at least for the exploratory search as it is simpler and quicker than a Pareto scheme. A Pareto scheme can be implemented for the convergence phase. The idea is to pick solutions randomly from the Pareto front of features to determine which solution produces the minimal instances. It is more complex to manage. This has not been implemented yet in our current version to focus on the role of scaling methodologies and avoid some confusion.

Population Evolution

Our GA adopts a restricted tournament selection RTS during the exploratory phase and a classical proportionate selection of the convergence phase. The global mechanism is however similar: A set of (n/2) parents is randomly selected from the current population at generation t (P_t). In our process, one chromosome can be parent several times.
Each selected chromosome A et B is submitted to the crossover (A=>A', B=>B') and mutation (A'=>A'', B'=>B'') operators with their respective probabilities. A" and B" are then candidates for the new population. By this process, the n chromosomes from generation t produce n_1 (n_1<n) children (Pc_t).
A copy of the parent and children population is kept. Parents (P_t) and children (Pc_t) are then considered to form the new population (P_{t+1}).

Concerning genetic operators, single-point crossover and mutation are used. Fitness selection is implemented by assigning a probability value to each individual, based on its fitness value, and by making individuals with higher probability values more likely to be selected to produce offspring. The operators are described as follows:

Crossover operator: this operator is based on the crossover probability (P_c). One point is selected randomly to produce the combination of two parent chromosomes that give two children by swapping their components.

Mutation operator: this operator is based on the mutation probability (P_m). Each bit is flipped to produce a new chromosome whose validity is checked. We adopt a uniform mutation, meaning that each chromosome undergoes the mutation process with the same probability of mutation.

Selection operator: during the exploration phase, a RTS scheme is applied; this is then switched to a classic proportional selection for the convergence phase, which is more elitist. RTS belongs to the family of crowding methodologies in which the basic idea is to encourage the insertion of new chromosomes in the population by replacing the most similar ones. RTS initially selects two chromosomes A and B to produce A'' and B" as presented before. For each A" and B", the members of the current population are scanned, and the closest among the group to A" and B" is saved for further processing (say A_{1nn} and B_{1nn}). A" competes against A_{1nn} and B" competes against B_{1nn} and the winners are inserted in the new population.
In our selective scheme, a set of chromosomes is randomly chosen (probability P_{keep}) from the current population representing a level of population tournament from one generation to another one. This subset is placed in the next population without undergoing any other genetic operations. The other chromosomes (probability 1 - P_{keep}) compete with chromosomes of (Pc_t). The best are kept and the worse discarded. The size of the tournament controls the amount of selection pressure and hence the convergence speed. A new population (P_{t+1}) of size n is then obtained.

Optional additional mechanisms or adaptation

We propose two mechanisms experimented in selection problems [33], i.e.: an archive population, and a breaking mechanism in order to auto-balance diversity and elitism. The archive population is used as a repository of solutions; it provides an extra source of results and favors more elitism.
Each time a sign of premature convergence is detected in the current population, the breaking mechanism, which is integrated with the main objective of preventing premature convergence, encourages diversification by re-seeding selected chromosomes.
In addition to these mechanisms, a local search procedure can be incorporated but only during the convergence phase. The incorporation of local search heuristics serves as an extra operator and it works together with crossover and mutation as

part of the GA's loop in order to accelerate convergence towards a better solution space.

B. Optimization of the fitness function

Most of GA time is generally spent on chromosome evaluation. Although intelligent mechanisms are necessary to guide the GA advances, reducing the evaluation time while maintaining an acceptable classification accuracy is essential, especially for managing non-trivial databases. We propose to link the resources of the scaling algorithm with the genetic advances. This means that a very light (approximate) version of the scaling algorithm is used during the preliminary stages of the GA, which are specifically devoted to the exploratory part. A more complete (accurate) version is used in the elitist part.

Integration of scaling approaches

Scaling approaches have proved to be effective. However, the integration of scaling methods in a GA requires special precautions as chromosome evaluation is needed at every moment without the possibility of parameter tuning or user intervention. This imposes a given flexibility for the setting of initial parameters and for the integration of scaling methodologies, which are inherently likely to affect the classification performances. We propose two alternatives: a very rough but fast approach and another much more accurate, but slower, approach. The scaling methods to be integrated into the GA are based on some pioneering methods [31] and also on some of our own research. It should be specified that the problem of minority classes [42] has been taken into account.

Approximate approach

The approximate approach we propose is based on a simple partitioning of the population into primary strata that are randomly generated. Instances extracted from each stratum are simply aggregated to form the final set (Fig.2). Two parameters guide the approximate approach: the first is the size of each stratum and the second the number of strata used.

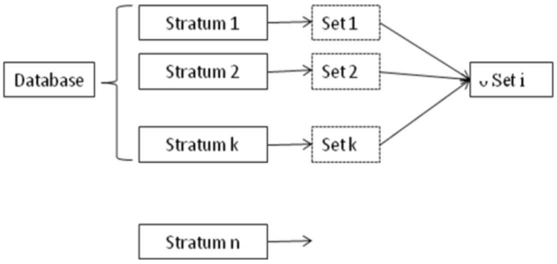


Figure 2: Instances of k strata are used for the final set

The best configuration depends on the boundary complexity. For well-separated data and simple boundaries, a limited number of small strata are sufficient. For overlapping data and complex boundaries, instances are likely to be not consistent with small strata. Obviously, extracting instances using large strata is more costly but naturally leads to greater accuracy as the data degradation is less. Limiting the number of strata reduces the instance set size but is likely to affect accuracy. The first parameter is difficult to estimate as it depends on the boundary complexity. We can nevertheless identify elementary statistics to determine a stratum size corresponding to a minimum of representativeness.
The probability P[X=k] of reaching a pattern from a region R representing a probability p_r within k random extraction is
$P[X=k] = p_r * (1 - p_r)^k$.

Based on this element, it is then possible to fix n_s (the stratum size per class). For the second parameter, the data complexity can be assessed iteratively.
Let Q be the set of instances generated by n_1 strata already accumulated. If the set of instances I_c generated by a new candidate stratum does not afford any additional information, this means that Q is sufficient to characterize the data complexity. The set Q is used to classify a random subset of training patterns extracted from each class using the 1_{nn} rule. If classification accuracy is high or does not progress between two iterations the algorithm is stopped. Note that the process is optimized: the nearest distances calculated for each training pattern are stored. Then adding a new stratum only requires the calculation of distances between the training patterns with the new instances. The objective is to obtain a weak signal on the power of discrimination.
The only parameter required to run the GA is the maximum number of strata Max_s.

Accurate approach

This so-called accurate approach is based on the notion of metastrata, a metastratum being simply a set of elementary strata. The process for obtaining a metastratum is similar to the one presented for the approximate approach. Instead of using an instance selection algorithm, a more approximate but faster approach is proposed to extract instances in each elementary stratum since the role of the generated instances is different.
To avoid confusion, we will call these "interesting patterns". Once a first reference metastratum M_r and its "interesting patterns" I_p are available, the database can be divided into metastrata.

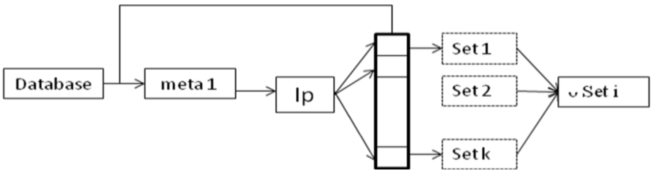


Figure 3: Instances are extracted from k clusters generated from metastratum 1

A clustering approach is applied to I_p and each cluster C_i will represent a "target pattern" for recruiting influential patterns. k sets of influential patterns are recruited, one for each cluster. The influential patterns are recruited by classifying the patterns contained in M_r by the set of C_i using the 1_{nn} rule (Fig. 3). Finally, each C_i characterizes a region, meaning that instance selection algorithms can be applied on small sets of patterns. A "crisp" 1_{nn} approach may be not sufficient. Some patterns close to the decision boundaries can be assigned to a region just near the decision boundaries. It will then be ignored by the IS, which can affect the classification accuracy.

For this reason, the recruitment of influential patterns from the active cluster C_a is done by completing its members by taking its two nearest neighbors C_{n1} and C_{n2} (Fig. 4). C_{n1} and C_{n2} are selected on the basis of the distance between two clusters denoted as d_c (C_i,C_j):

$$d_c(C_i,C_j)=d(G_i,G_j) \qquad (2)$$

where $d(z_m, z_n)$ is the Euclidean distance between z_m and z_n, anf G_i is the center of gravity of cluster i.

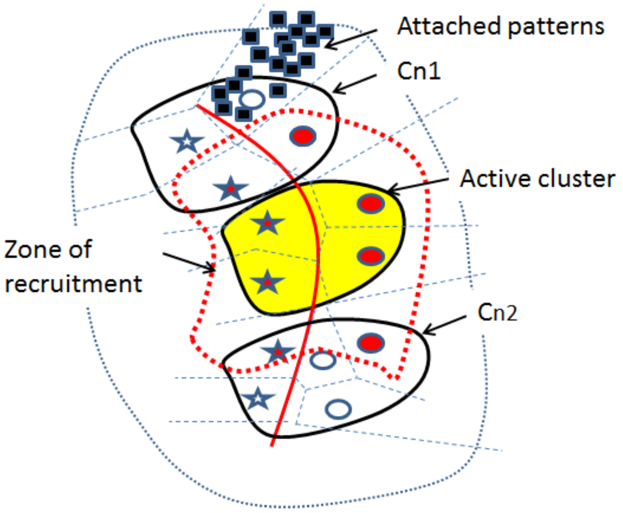


Figure 4: Recruitment of the influential patterns

Finally, a set of preliminary influential patterns $I(C_a)$ for the cluster C_a is built by gathering the patterns attached to C_a and to some members of C_{n1} and C_{n2}. The proportion of patterns from C_{n1} and C_{n2} is however limited to β% (β from 0 to1%). They are selected on the basis of their distances with C_a.

$$I(C_a)=I(C_a)+I'(C_{n1})+I'(C_{n2}) \quad (3)$$

where I(x) denotes the influential patterns attached to x, and I'(x) the selected ones.

Two cases can occur: the sets can be either too small or too large. For the former, the related patterns are associated with another "target pattern". For the latter, the population is cut up in order to obtain an acceptable size (S_z) for the instance method. The cutting is done through a fast clustering method. Working from one metastratum makes it possible to achieve a certain level of accuracy. The procedure can be more accurate if duplicated with the other metastrata on the basis of the same I_p. In the GA, we will play on this setting in order to achieve greater or less accuracy depending on the genetic advances. The level of results can be controlled during the procedure which can be stopped in the event of satisfactory results. As for the exploratory search, the maximum number of metastrata Max_{ms} serves to drive the GA.

Combination with the GA

The range of fitness scores is divided into levels. Each level is associated to a specific scaling approach. Different alternatives are then possible to assess the advances of the GA: the fitness score of the best chromosome evaluation, the average over the whole population, etc. However, once the algorithm has passed through a given level, it can only remain at that level or move up a level. The average may indeed vary negatively, for example if diversification mechanisms are implemented. Let there be 4 levels of average: α_1, α_2, α_3, α_4. The first two are used for the approximate approach and the others for the accurate one. In each interval, we define a value for Max_s or Max_{ms} depending on the approach.

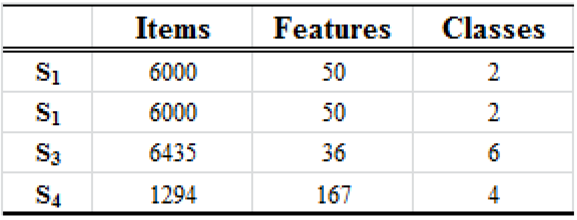
IV. Experimental results

The objective of this section is to demonstrate that our approach is interesting for managing the dual selection problem with non-trivial databases. $DROP_4$ [19] was selected as the C_{1nn} with the most standard parameters.

A. Data sets used

The proposed selection method was tested on four data sets with known complexity. Their sizes are not trivial but nevertheless allow the application of non-scaling approaches. The first one, an academic example, is the artificial Bullseye problem [43]. The Bullseye problem consists of a set of 6000 dimensional samples generated by 50 features and divided into two categories. With only two features, denoted 0 and 1, the Bullseye categories can be classified with a minimal error, as the population distribution associated to the remaining features is modified by exchanging samples randomly between the categories. The second data set is similar: it is composed of 3 well-separated eyes of 2000 patterns giving more complex boundaries. The others are real and classes are not well separated. The third data set is based on data derived from the image field (satimage) [44], and the last one from a chemometric database [45] composed of compounds coming from various medicinal and drug data reports, published during the last twenty years. We used a subset of this database that is relevant for 4 anti-cancerous properties.

The data set characteristics are reported in Table I. We split the data sets into training (80%) and test (20%) sets by applying a random partitioning.



	Items	Features	Classes
S_1	6000	50	2
S_1	6000	50	2
S_3	6435	36	6
S_4	1294	167	4

Table 1. Data set characteristics.

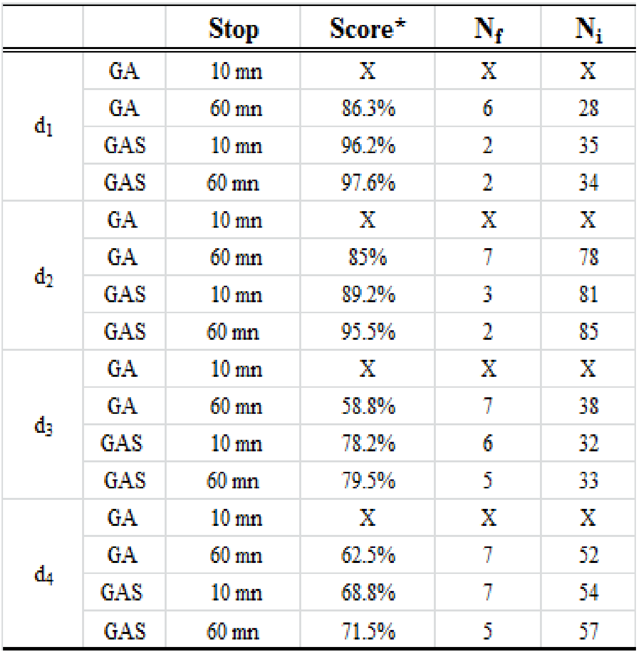
B. *The GA implemented*

The feature selection problem is essentially multimodal and a standard GA may have some difficulty in obtaining adequate results without being trapped in a partial good solution. For the preliminary experiment, we selected the most basic GA scheme so as to focus on scalability and avoid possible interactions with additional mechanisms. In a second series of experiments we introduce the breaking process mentioned above.

The chromosomes are submitted to genetic operations with the most standard probability levels that remain the same during the genetic life:
Probability of crossover P_c=0.5
Probability of mutation P_m=0.1
Probability of tournament $P_{keep=}$=0.5

The initial population is randomly generated with chromosomes having less than max_f = 20 active features.
Preliminary experiments

In order to show the interest of the approach we compare the results with non-scaling approaches running with 30 chromosomes. In our algorithm (GAS), 30 chromosomes were used during the two phases. The following parameters were adopted: the α parameters were respectively 0.1, 0.5, 0.7 and 0.9 with the corresponding values for the maximum of strata (1, 10) and metastrata (1, 2, 3). The size of elementary strata n_s was fixed at 100 per class and S_z at 300.
The value β for the scaling approach was fixed at 0.5%. Table II summarizes the results obtained by the criteria of efficiency, the number of features and instances. Only the score of the best chromosome is given in the Table. Two results are provided: after the first 10 minutes of processing and after 60 minutes. For this preliminary test, we launched ten runs to validate the concept. The results are the average of the runs.



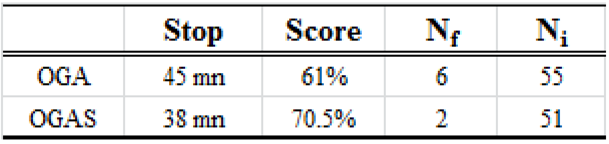
		Stop	Score*	N_f	N_i
d_1	GA	10 mn	X	X	X
	GA	60 mn	86.3%	6	28
	GAS	10 mn	96.2%	2	35
	GAS	60 mn	97.6%	2	34
d_2	GA	10 mn	X	X	X
	GA	60 mn	85%	7	78
	GAS	10 mn	89.2%	3	81
	GAS	60 mn	95.5%	2	85
d_3	GA	10 mn	X	X	X
	GA	60 mn	58.8%	7	38
	GAS	10 mn	78.2%	6	32
	GAS	60 mn	79.5%	5	33
d_4	GA	10 mn	X	X	X
	GA	60 mn	62.5%	7	52
	GAS	10 mn	68.8%	7	54
	GAS	60 mn	71.5%	5	57

Table 2. Results for different configurations
(*X= no result obtained)

Two results clearly stand out and highlight the contribution of integrating the scaling effect. First, it can be seen that for the 4 data sets, interesting results are already obtained after only 10 min of processing. Second, more than 100 iterations can be processed in less than 1 hour with the scaling approach. When scaling is not used, the first iteration is not finished after 10 min of processing and less than 10 iterations are possible in 60 min. 100 iterations requires a very long processing time (several hours). This makes the algorithm intractable for large databases. If we compare the two results of classification and number of selected features after 100 iterations, no clear differences between non-scaling and scaling approaches can be observed in our benchmark. There is no difference for the synthetic base while the results vary a little for other data sets. For example, within the same number of genetic generations (100), 73.5% of correct classification was obtained on data set 4 with non-scaling approaches versus 71.5% with scaling ones. This is the cost of using scaling methodologies. It should be noted that scaling approaches generally generate a few more instances (about 10% in our benchmark) than non-scaling approaches. A further specific reduction is always possible via a dedicated GA by considering a set of promising chromosomes at the end of the genetic process. These preliminary results show the importance of the exploratory phase and its ability to reduce the search very quickly. This can be explained by the presence of far more bad solutions than good ones. Then, the weak signals delivered via the rough fitness function are sufficient to push the genetic advance. The convergence phase is computationally more expensive especially when several metastrata are used. Comparatively, the progression in genetic advances is better during the exploratory phase.

A. *Additional experiments with a more efficient GA*

In order to accelerate the evolutionary process and reach an efficient solution with a reasonable execution time, several mechanisms from the literature [33, 46] were incorporated in the standard GA. They are briefly presented here. Firstly, genetic operators, and particularly the mutation operator were modified so as to be fully adapted to the level of the genetic process and therefore limit the simultaneous tuning of several static parameters. Secondly, a ranking selection scheme was introduced in order to limit the promotion of extraordinary chromosomes, thus preventing premature convergence. Thirdly, the new GA called OGA (OGAS when scaling methodologies are incorporated) was reinforced by a mechanism involving a breaking process to reseed the population when necessary. This makes it possible to continuously manage the trade-off between elitism and pressure preservation.
Table III details the results for data set 4. For each run, the genetic process was stopped when comparable results with the standard GA were obtained.



	Stop	Score	N_f	N_i
OGA	45 mn	61%	6	55
OGAS	38 mn	70.5%	2	51

Table 3. Results for OGA algorithm.

The observed difference is that the GA produces similar results to the standard GA within a shorter CPU time (25% to 30% less). With this algorithm, each time a sign of premature convergence is detected, the population is reseeded, which affects the fitness average without losing the best chromosome. This speeds up the genetic advances and improves the final convergence. To go further and evaluate the contribution of the scaling methodologies, we let the OGA work until its performance was close to that of OGAS: on the basis of three runs, more than 100 mn were necessary to reach this objective (Score: 70.7%, $N_f = 2$ and $N_i = 52$). Without generalizing, it can then be said that incorporating scaling methodologies in GAs can speed up the genetic advances. The improvement acts directly on the fitness function and the effect is complementary to the one that may be obtained by optimizing the genetic process itself.

V. Conclusion

In this study, an algorithm for feature and instance selection in the context of supervised classification has been investigated. The novelty of our approach resides in the integration of scaling methods in a GA devoted to feature and instance selection. The key features are the following: the type of scaling method is adapted to the genetic advances and the computational time of the scaling methods themselves is optimized.

The results obtained with the use of a basic GA are encouraging and in line with the objective of improving tractability. While our benchmark is limited to medium-sized databases, the results have nonetheless revealed the effectiveness and applicability of the approach. Under this process, very large databases can be managed on the basis of instance selection algorithms, which is impossible with non-scaling approaches. Additional tests are needed on larger databases and significant noisy domains have to be investigated.

We think that the combination of GA and scaling procedures can be improved by including more flexibility. The experiments carried out show that preliminary good results can be obtained very quickly through the exploration phase while the advance is slower during the convergence phase. The question is how to adapt the use of scaling methodologies. How can the run time be reduced without degrading the classification performances too much?

We plan to deal with some of these issues in future work.

Another future plan is to extend the algorithm to an additional phase (optimization phase) for handling a chromosome encoded for feature and instance selection simultaneously. This represents a major difference with previous work where this encoding is impossible because of the chromosome dimension. In this case, both the feature and instance spaces are highly reduced as only subsets of features and instances from the original database are considered. The subset of promising features is known by the essence of the feature selection algorithm. The subset of promising instances can be established by integrating a dynamic archive memory during the feature process selection. This memory stores and sorts the instances that have been selected by the application of the scaling methodologies during the feature selection process. We can therefore imagine obtaining a chromosome vector of less than one thousand dimensions even for huge databases. For this dimension, a dual selection algorithm can be applied and may lead to more efficient results.

References

[1] Guyon I., Elisseeff A. “An Introduction to Variable and Descriptor Selection”, *Machine Learning Research*, 3, pp. 1157–1182, 2003.

[2] Aha D., Kibler D., Albert M.K. “Instance-based learning algorithms”, *Machine Learning*, 6, pp. 37–66, 1991.

[3] Cano J.R., García S., Herrera F. “Subgroup Discovery in Large Size Data Sets Preprocessed Using Stratified Instance Selection for Increasing the Presence of Minority Classes”, *Pattern Recognition Letters*, 29, 2008.

[4] Souza, J. T., Carmo R. A. F., Campos G.A. “A novel approach for integrating feature and instance selection”, *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 374-379, 2008.

[5] Zongker D., Jain A.K. “Algorithms for feature selection: an evaluation”, *IEEE Trans Pattern Anal Mach Intelligence*, 26(9), pp. 1105–1113, 2004.

[6] Ros F., Harba R. **“**Fast instance selection hybrid algorithms adapted to large data sets”, *Proceedings of the 2011 International Conference of Soft Computing and Pattern Recognition, IEEE Conference Publishing Service*, 2011.

[7] Jaekyung Y., Sigurdur O. “Optimization-based feature selection with adaptive instance sampling”, *Computers & Operations Research*, 33, pp. 3088–3106, 2006.

[8] Fen T. “Improving Feature Selection Technique for Machine Learning”, *technical report*, Georgia State University, 2007.

[9] Kohavi R., John G.H. “Wrappers for feature subset selection”, *Artificial Intelligence journal, special issue on relevance*, 97, pp. 273–324 , 1997.

[10] Yang C.G., Chuang L.Y., Yang CH.. “IG-GA: A Hybrid Filter/Wrapper Method for feature Selection of Microarray Data”, *Journal of Medical and Biological Engineering*, 30(1), pp. 23-28, 2009.

[11] Kumari B., Swarmkar T. “Filter versus wrapper feature subset selection in large dimensionality micro array”, *Computer Science. Information.Technology*, 2, pp. 1048-1053, 2011.

[12] Jain A.K., Zongker D. “Feature selection: evaluation, application, and small sample performance”, *IEEE Trans Pattern Anal Mach Intelligence*, 19(2), pp. 153–158, 1997.

[13] Nitin B., Vandana S. “Survey of Nearest Neighbor Techniques”, *International Journal of Computer Science and Information Security*, 8(2), 2010.

[14] Hart P.E. “The condensed nearest neighbor rule”, *IEEE Trans Inf Theory*, 16, pp. 515–516, 1968.

[15] Gates G.W. “The reduced nearest neighbor rule”, *IEEE Trans Inf Theory*, 18(3), pp. 431–433, 1972.

[16] Alejo R., Sotoca J.M., Valdovinos R.M., Toribio P. “Edited Nearest Neighbor Rule for Improving Neural Networks Classifications”, *Advances in Neural Networks Lecture Notes in Computer Science*, 6063, pp. 303-310, 2010.

[17] Brighton H., Mellish C. “On the consistency of information filters for lazy learning algorithms. Principles of

Data Mining and Knowledge Discovery", *3^{rd} European Conference*, LNAI 1704, pp. 283–288, 1999.

[18] Wilson D.R., Martinez T.R. "Reduction techniques for instance-based learning algorithms", *Machine Learning*, 38(3), pp. 257–286, 2000.

[19] Zhang T. "On the Consistency of Feature Selection using Greedy Least Squares Regression", *Machine Learning Research*, 10, pp. 555-568, 2009.

[20] Lobo F.G., Goldberg D.E. "Decision making in a hybrid genetic algorithm", *IEEE International Conference on evolutionary Computation*, pp. 122-125, 1997.

[21] Yu T., Davis L., Baydar C.,Roy. R. "Evolutionary Computation in Practice", *Studies in Computational Intelligence*, 88, 2008.

[22] Goldberg D.E. "Genetic algorithms in search, optimization and machine learning", Addison-Wesley, Boston, 1989.

[23] Krasnogo N., Smith J. "A tutorial for competent memetic algorithms: model, taxonomy and design issues", *IEEE Transaction Evolutionary Computation*, 9(5), pp. 474-488, 2005.

[24] Deepti G., Shabina G. "an overview of methods maintaining diversity in Genetics Algorithms", *International Journal of Emerging Technology and Advanced Engineering*, 2: 5, 2012.

[25] Colman, A.M., Browning L. "Evolution of cooperative turn-taking", *Evolutionary Ecology Research*, 11, pp. 949–963, 2009.

[26] Kordon A., Castillo F., Smits G., and Kotanchek M. "Application Issues of Genetic Programming in Industry", *In Genetic Programming Theory and Practice III*, T. Yu, R. Riolo and B. Worzel (Eds), pp. 241-258, 2006.

[27] Steven H. "On the practical usage of genetic algorithms in ecology and evolution", *Methods in Ecology and Evolution*, 4: 2, pp. 184–194, 2013.

[28] Shalak D.B. "Prototype and feature selection by sampling and random mutation hill climbing algorithms", *In: Proceedings of the 11th international conference on machine learning,* New Brunswick. Morgan Kaufman, New Jersey, pp. 293–301, 1994.

[29] Kuncheva L., Jain L.C. "Nearest neighbor classifier: Simultaneous editing and feature selection", *Pattern Recognition Letters*, 20, pp. 1149-1156, 1999.

[30] Ros F, Guillaume S. Pintore M., Chretien J.R. "Hybrid Genetic Algorithm for Dual Selection", *Journal of Pattern Analysis and application*, 1, pp. 179-198, 2008.

[31] García S., Cano J.R., Herrera F. "A Memetic Algorithm for Evolutionary Prototype Selection. A Scaling Up Approach", *Pattern Recognition,* 41:8, pp. 2693-2709, 2008.

[32] Ros F., Pintore M., Harba R.. "Fast Dual Selection Using Genetic Algorithms for large data sets", *Proceedings of the 2012 International Conference of Soft Computing and Pattern Recognition*, IEEE Conference Publishing Service, Brunei, 2012.

[33] Ros F., Guillaume S. "An efficient nearest classifier, Book Chapter of Hybrid Evolutionary Systems", Studies in Computational Intelligence, 75, Springer Verlag, pp. 131-147, 2007.

[34] Zhu X, Wu X. "Scalable representative instance selection and ranking", *Proceedings of the 18th international conference on pattern recognition (ICPR'06)*, 3, pp. 352–355, 2006.

[35] Ros F., Harba R., Pintore M., Piclin N. "Neighborhood and stratification approaches to speed up instance selection algorithm ", *International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)*, 4, pp. 537-545, 2012.

[36] Pedrajas N.G., Haro-Garcia A., "Scaling up data mining algorithms: review and taxonomy", *Artificial Intelligence*, 1, pp. 71-87, 2012

[37] Olvera-Lopez J.A., Corrasco-Ochoa J.A., and Martinez Trinilad J.F. "A new fast prototype selection method based on clustering", *Pattern Analysis and aplication*,13(2), pp.131-141, 2010.

[38] Haro-Garcia A., Garcia-Pedrajas N. "A divide-and-conquer recursive approach for scaling up instance selection algorithms", *Data Mining and Knowledge Discovery*, 18(3), pp. 392-418, 2009.

[39] Ros F., Harba R, Piclin N., Pintore M. "Fast instance selection algorithm adapted to large data sets", *International Conference on Soft Computing and Pattern Recognition*, pp. 320-325, 2011.

[40] Lozano M., Herrera F., Cano J.R. "Replacement strategies to preserve useful diversity in steady-state genetic algorithms", *Information Sciences*, 23, pp.4421-4433, 2007.

[41] Smits G., Kotanchek M. "Pareto Front Exploitation in Symbolic Regression", In Genetic Programming Theory and Practice II , U.M. O'Reilly, T. Yu, R. Riolo and B. Worzel (Eds), pp. 283-300, Springer, 2004.

[42] Barandela R., Sanchez J.S., Garcia V., Rangel R. "Strategies for learning in class imbalance problems", *Pattern Recognition*, 36, pp.849-851, 2003.

[43] Ripley B.D.. "Statistical aspects of neural networks. Networks and Chaos: Statistical and Probabilistic Aspects", Chapman & Hall, London, O.E. Barndorff-Nielsen, J.L. Jensen, W.S. Kendall (Eds.), pp. 40–123, 1993.

[44] Blake C., Keogh E., Merz C.J.: UCI repository of machine learning databases: http://www.ics.uci.edi/ mlearn /MLRepository.html, 1998.

[45] Piclin N., Pintore M., Wechman C., Chretien J.R. "Classification of a large anticancer data set by Adaptive Fuzzy Partition", *Journal Computer-Aided Molecular Design*, 18, pp. 577-586, 2004.

[46] Boudjelaba K., Ros F. "Evolutionary techniques for the synthesis of 2 D FIR Filters", *in Statistical Signal Processing Workshop (SSP)*, IEEE, 2011.

Author Biographies

Frederic Ros was born in 1968. He has an engineering degree in Microelectronics and Automatic, a Master in Robotics from Montpellier University and a Ph.D. degree from ENGREF (Ecole Nationale du Genie Rural des Eaux et Forets) Paris. He began his career in 1991 as a research scientist working on the field of image analysis for robotics and artificial systems from CEMAGREF (Centre National d'Ingénieurie en Agriculture) where pioneer applications combining neural networks, statistics and vision were developed. He managed the vision activity in GEMALTO during 14 years which is the world leader in the smart card industry. He was particularly involved in applied developments (related to data analysis, fuzzy logic and neural networks) with the aim of providing adaptive and self-tuning systems corresponding to the growing complexity of industrial processes and especially multi-disciplinary interactions. He has been an associate researcher at PRISME laboratory and head an innovation park for 4 years. He has co-authored over 70 conference and journal papers and made several reviews in this field.

Rachid Harba was born in 1960. He received the Agregation in electrical engineering from ENS Cachan, Cachan, France in 1983. He received the PhD degree in electrical engineering from INPG Grenoble, France in 1985. Since 1987, he has joined the Laboratory of Electronics, Signals, Images (LESI), Orléans, France, as an Associate Professor and teached at Polytech'Orleans engineering school. In 1997, he became a full Professor and took the head of the LESI. He is now a first class Professor at PRISME laboratory, University of Orleans. He is interested in signal and image processing applied to biomedical domains, material science and industrial applications. He is the author or coauthor of about 100 papers in Journals and conferences.

Dr. Marco Pintore was born in 1970. He is at present CEO of EtnaLead srl, a SME focused on the design of alternatives solutions for the pharmaceutical and cosmetic domains, after achieving a long experience in managing BioChemics Consulting SAS, a SME providing bioactivity prediction and virtual screening engineering services (pharmacy, cosmetics and environment). Marco has a large expertise, besides management, in the prediction of molecular bioactivity by data mining and 3D molecular modeling techniques, and he has been involved in about 40 publications in peer reviewed scientific journals. Marco has also participated as scientific leader in several projects funded by the European Commission, within the Fifth and Sixth Framework Programme, and was coordinator of several national funded R&D projects.