Information Processing Letters 109 (2009) 887–896

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Sensitivity and specificity based multiobjective approach for feature selection: Application to cancer diagnosis

J. García-Nieto [a,1], E. Alba [a,*,1], L. Jourdan [b,2], E. Talbi [b,2]

[a] Dept. de Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, Málaga - 29071, Spain
[b] LIFL/INRIA Futurs Parc scientifique de la Haute-Borne, Bât. A, 40 Avenue Halley, 59650 Villeneuve d'Ascq Cedex, France

ARTICLE INFO

Article history:
Available online 15 April 2009
Communicated by A.A. Bertossi

Keywords:
Algorithms
Analysis of algorithms
Combinatorial problems
Databases
Design of algorithms
Performance evaluation
Sensitivity and specificity analysis
Multiobjective genetic algorithm
Microarray gene selection

ABSTRACT

The study of the sensitivity and the specificity of a classification test constitute a powerful kind of analysis since it provides specialists with very detailed information useful for cancer diagnosis. In this work, we propose the use of a multiobjective genetic algorithm for gene selection of Microarray datasets. This algorithm performs gene selection from the point of view of the sensitivity and the specificity, both used as quality indicators of the classification test applied to the previously selected genes. In this algorithm, the classification task is accomplished by Support Vector Machines; in addition a 10-Fold Cross-Validation is applied to the resulting subsets. The emerging behavior of all these techniques used together is noticeable, since this approach is able to offer, in an original and easy way, a wide range of accurate solutions to professionals in this area. The effectiveness of this approach is proved on public cancer datasets by working out new and promising results. A comparative analysis of our approach using two and three objectives, and with other existing algorithms, suggest that our proposal is highly appropriate for solving this problem.

1. Introduction

Microarray technology [1] (DNA Microarray) allows biologists to simultaneously analyze thousands of genes and can thus provide important insights into cell functioning, since changes in the physiology of an organism are generally associated with changes in gene expression patterns. The vast amount of data involved in a typical Microarray experiment naturally leads to using statistical analysis, in addition to classifying the dataset into correct classes. The key issue in this classification procedure is to identify significant and representative gene subsets that may be used to predict class membership for new external samples of genes. The main difficulty in the Microarray classification problem is the availability of a relatively small number of samples in comparison with the large number of genes in each sample. In addition, expression data are highly redundant and noisy, and most genes are believed to be uninformative with respect to classes studied, as only a fraction of genes may present distinct profiles for different classes of samples.

In this context, feature selection is considered a necessary preprocess step to analyze large datasets, as this method can reduce the dimensionality of the datasets and often leads to better analyses [2]. Therefore, in feature selection, the objective is to select subsets, which are as small as possible, of informative features from the initial dataset, in order to obtain high classification accuracy.

Nevertheless, optimal feature selection is a complex problem proved to be NP-hard [3], and hence, only meta-

* Corresponding author.
E-mail addresses: eat@lcc.uma.es (J. García-Nieto), jnieto@lcc.uma.es (E. Alba), laetitia.jourdan@lifl.fr (L. Jourdan), el-ghazali.talbi@lifl.fr (E. Talbi).

[1] Authors acknowledge funds from the Spanish Ministry of Sciences and Innovation FEDER under contract TIN2008-06491-C04-01 (M* project, http://mstar.lcc.uma.es) and CICE, Junta de Andalucía under contract P07-TIC-03044 (DIRICOM project, http://diricom.lcc.uma.es).
[2] Authors acknowledge funds from the INRIA-PERFOM 3 + 3 Méditérrané.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2009.03.029

heuristic approaches are capable of solving it accurately and efficiently. Evolutionary Algorithms (EAs) have been successfully used in the past to tackle the gene selection of Microarray datasets [4–6]. Specifically, Multiobjective Evolutionary Algorithms (MOEAs) [7–9] are actually quite popular for feature selection, since they allow a given solution to be evaluated (selected subset) in a more suitable and straightforward way. From this point of view, two objectives are clearly involved in the evaluation of solutions: minimizing the *number of genes* and maximizing the classification *accuracy*. However, using this approach, the range of good solutions (i.e., $\leqslant$10 genes and $\geqslant$90% accuracy) is normally limited, which is a problem for the subsequent expert decision making process. In this sense, the analysis of the sensitivity and the specificity (ROC analysis) [10] of a diagnostic test constitutes a powerful analysis in supervised classification since it provides the specialists with more detailed information about the validity of a solution. Moreover, the use of the sensitivity and the specificity as objectives in the evaluation task, instead of the accuracy rate, offers a mathematically equivalent method since these two factors are proportionally (weighted to the *prevalence*) included in the overall accuracy calculation. Therefore, despite the intuitive appeal of using only the overall accuracy as a single measure of test validity, its dependence on the *prevalence* renders it inferior to a careful and balanced consideration of sensitivity and specificity [11].

Recent studies have used multiobjective algorithms in order to optimize the sensitivity and the specificity for a given classifier [12,13]. However, the main goal of these approaches consisted of looking for a favorable trade-off between sensitivity and specificity, considering neither subset selection nor dataset reduction purposes. In this paper, we extend these works by using a MultiObjective Genetic Algorithm (MOGA) for gene selection and classification of Microarray datasets, which optimizes three objectives: maximize the sensitivity, maximize the specificity, and minimize the number of genes. Initially, this algorithm selects the subsets of genes encoded in the tentative solutions manipulated by the algorithm. After that, each solution is evaluated using the Support Vector Machines (SVMs) classifier, and 10-fold cross-validation is then applied to assess the percentages of sensitivity and specificity. In addition, each generation, specialized crossover and mutation operators, both adapted to feature selection, are applied to the population. Our contribution is noticeable since this approach is able to offer a number of accurate solutions to professionals in this area. As we will show in the experiments, the effectiveness of this approach is evaluated on three well-known datasets, and new and promising results are obtained.

The remaining of this paper is organized as follows. In Section 2, we provide the reader with basic concepts about the feature selection problem, the sensitivity and specificity analysis, the Support Vector Machines classifier and the Microarrays technology. Section 3 gives the details of our specialized MOGA algorithm for gene selection and classification. Experimental results and comparisons are presented in Section 4. Conclusions and further work are finally given in Section 5.

2. Basic concepts

In this section, preliminary concepts of the feature selection problem, the sensitivity and specificity analysis, the SVMs classifier and the Microarrays technology are briefly introduced.

2.1. Feature selection

When applied to Biology, feature selection is also called *gene selection*, targeted to distinguish influential genes from irrelevant ones based on DNA Microarray datasets. This technique is normally coupled with learning algorithms that use the reduced subset of features in order to proceed as a supervised classifier. The formal definition of the feature selection problem is given as follows:

Definition. Let $F = \{f_1, \ldots, f_i, \ldots, f_n\}$ be a set of features; find a subset $F' \subseteq F$ that maximizes a scoring function $\Theta : \Gamma \to G$ such that $F' = \text{argmax}_{G \subset \Gamma}\{\Theta(G)\}$, where Γ is the space of all possible feature subsets of F and G a subset of Γ [14].

In feature selection, two different models may be used depending on whether the learning algorithm is coupled with the selection method or not, respectively *wrapper* model and *filter* model. On the one hand, the *filter model* carries out the selection and classification regardless of the learning algorithm used. Filter methods are based on a performance evaluation metric calculated directly from the data.

On the other hand, the *wrapper model*, which performs feature subset selection and classification in the same single process, internally uses a learning algorithm to measure the accuracy.

2.2. Sensitivity and specificity

The sensitivity is a statistical value measuring how well a binary classification test correctly identifies a condition. The sensitivity is the proportion of *true positives* of all diseased cases in the entire population. For example, for a medical test to determine if a person has a certain disease, the sensitivity to the disease is the probability that if the person has the disease, the test will be positive. However, the sensitivity alone does not tell us how well the test predicts other classes.

The specificity is a statistical measure of how well a binary classification test correctly identifies the negative cases, or those cases that do not meet the condition being studied. For example, given a medical test that determines if a person has a certain disease, the specificity of the test to the disease is the probability that the test indicates "negative" if the person does not have the disease. The specificity is thus the proportion of *true negatives* of all negative cases in the population.[3]

[3] The sensitivity depends on the number of true positives (*#TP*) and true negatives (*#TN*). The specificity depends on the number of true negatives (*#TN*) and false positives (*#FP*). The number of false negatives (*#FN*) is included in the calculation of a third value: accuracy.

$$\text{Sensitivity} = \frac{\#TP}{\#TP + \#TN}, \tag{1}$$

$$\text{Specificity} = \frac{\#TN}{\#TN + \#FP}. \tag{2}$$

Eqs. (1) and (2) calculate the sensitivity and specificity factors of a prediction test, respectively.

A combination of these two measures is the most widely used method to quantify the diagnostic ability of a test, since these measures constitute basic factors included in all well-known statistical analysis such as the *receiver operating characteristic* (ROC curve) and the *F-measure*. A useful measure, the accuracy value of a test, may be determined by combining sensitivity and specificity, with another measure called Prevalence,[4] using the following equation:

$$\text{Accuracy} = \text{Sensitivity} \times \text{Prevalence} + \text{Specificity} \times (1 - \text{Prevalence}) \tag{3}$$

being

$$\text{Prevalence} = \frac{\#\text{classes (2 in this work)}}{\#\text{genes (in the dataset)}}. \tag{4}$$

In addition, the accuracy can also be expressed in terms of the true/false positive/negative factors as formulated in Eq. (5):

$$\text{Accuracy} = \frac{\#TP + \#TN}{\#TP + \#FP + \#FN + \#TN}. \tag{5}$$

2.3. Support Vector Machines

Support Vector Machines, derived from statistical learning theory, are used to classify points by assigning them to one of two disjoint half spaces. The objective is to provide a model which predicts, as efficiently as possible, the class of a given data instance in the testing set where only the values of the features are known and there is no information about the classes. Vapnik and Cortes [15] defined the SVMs method as follows:

Definition. Given a training set of instance-label pairs (x_i, y_i) with $i = 1, 2, \ldots, l$ where $x_i \in R^n$ (training vectors) and $y_i \in \{1, -1\}^l$ (classes), the Support Vector Machines require the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{6}$$

being

$$y_i(w^T \phi(x_i) + b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0. \tag{7}$$

Here, $C > 0$ is the penalty parameter of the error term. For linearly separable data, SVMs obtains the vector which maximizes the distance between the training samples and the class boundary. For non-linearly separable data, samples (x_i) are mapped to a high-dimensional space by means of the function ϕ, where a separating hyperplane can be found. The assignment is carried out by means of the equation $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ called the *kernel* function.

In the classification of informative genes embedded in a given dataset of gene expression levels, SVMs uses the kernel function to find an orthogonal hyperplane to a specific gene dimension. Many works in the literature have successfully used SVMs for gene selection and classification [4,5,16]. In this work, a linear kernel function ($K(x_i, x_j) \equiv x_i^T x_j$) is used by the SVMs classifier.

2.4. Microarrays and gene expression

Microarrays or gene arrays/chip [1] consist of a thin glass substrate containing specific DNA gene samples spotted in an array by a mechanical procedure. These DNA samples are spread with fluorescently labeled m-RNA from an experimental condition. This m-RNA hybridizes strongly with some DNA gene samples and weakly with others, depending on the inherent double helical characteristics. The array is then scanned and the resulting image is processed in order to detect fluorescence levels (using red and green dyes), indicating the strength with which the sample expresses each gene.

The logarithmic ratio between the two intensities of each dye is used as the gene expression data. The relative abundance of the DNA sequences spotted in a pair of DNA or RNA samples is assessed by evaluating the differential hybridization of the two samples compared to the sequences on the Microarray. The gene expression levels can be determined for samples taken either at multiple time instants of a given biological process or under various conditions, such as for tumor samples with different histopathological diagnosis. Each sample corresponds to a high-dimensional row vector of its gene expression profile.

3. Gene selection and classification by MOGA

As explained in the introduction, MOEAs are already being used for solving the gene selection and classification of gene expression datasets [7–9]. In most of these works, the optimized function is computed using two main objectives: the classification accuracy and the number of genes. Our approach here is to use a Multiobjective Genetic Algorithm, in which the evaluation of solutions involves the sensitivity and the specificity of the classification test, as well as the number of genes, but without using the classification accuracy. In this section, we present the main elements of our MOGA focusing on the fitness evaluation, solution encoding and adapted operators. Multiobjective specific aspects such as the preservation of the diversity and the selection of solutions are also described.

3.1. The multiobjective approach

Formally, each gene subset belonging to two classes, $s_i = \{X_i, y_i\}$ where $\{X_i\}$ represents the n training samples and $y_i \in \{-1, +1\}$ their class labels, is associated with

[4] In Biology, the Prevalence of a disease is the proportion of total cases (classes) of the disease in the studied population (set) divided by the number of individuals (genes) in this population (studied subset).

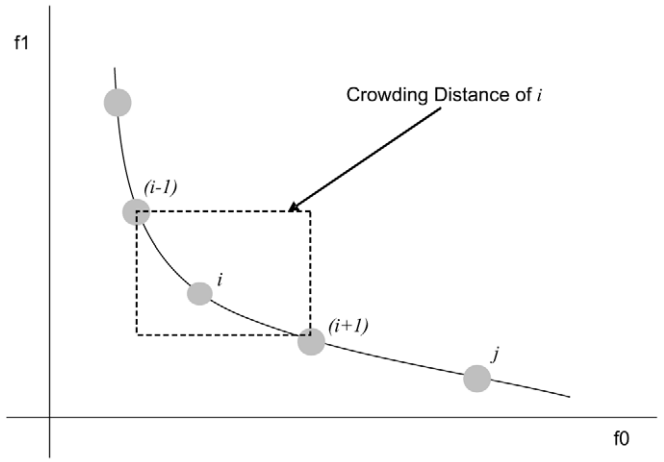

Fig. 1. Crowding distance operation.

a vector evaluation function $F(s_i) = \langle F_1(s_i), F_2(s_i), F_3(s_i)\rangle$, each function corresponding to:

$$F_1(s_i) = \frac{\#(\text{genes in } s_i)}{\#(\text{total genes})} \quad (\text{minimize \#genes}), \tag{8}$$

$$F_2(s_i) = \frac{\#TP}{\#TP + \#TN} \quad (\text{maximize sensitivity}), \tag{9}$$

$$F_3(s_i) = \frac{\#TN}{\#TN + \#FP} \quad (\text{maximize specificity}). \tag{10}$$

Our MOGA evolves by optimizing simultaneously these three fitness functions. In this scenario, a selected gene subset can be more informative (better than), less informative (worse than), equal, but also indifferent to another gene subset with respect to the objective values. Here, one subset s_i is said to "dominate" or be "more informative" than another subset s_j when:

$$\forall k \in \{1, 2, 3\}:\ F_k(s_i) \succeq F_k(s_j) \tag{11}$$

and

$$\exists k:\ F_k(s_i) \succ F_k(s_j). \tag{12}$$

In these equations, given two subsets of genes s_a and s_b, $F_k(s_a) \succeq F_k(s_b)$ means that s_a is not "less informative" than s_b, and $F_k(s_a) \succ F_k(s_b)$ means that the subset s_a is "more informative" than s_b. The "Pareto front" is then defined as the set of nondominated gene subsets which constitutes the best solutions. The MOGA guides the search towards the Pareto front, keeping the nondominated solutions as diverse as possible, and preventing nondominated solutions from being rejected, in order to delay premature convergence.

In our approach, the reinforcement of the diversity in the front is carried out by means of the "Crowding" distance operator (typical in NSGA-II [17]). The Crowding distance operator (Fig. 1) assigns the highest value to the boundary solutions, and the average distance of two solutions $(i-1)$th and $(i+1)$th on either side of the solution i in each of the objectives. The complete front (with new and old nondominated solutions) is sorted and the Crowding distance operator is performed to get a new front with spread solutions. The selection task is then accomplished by a tournament Crowding strategy, in which, given two solutions i and j, the solution i is selected if it dominates the solution j. If neither solution dominates the other, then the one less densely allocated in the search space (i.e., with less Crowding distance) is selected.

3.2. Solution encoding

In our MOGA, each individual encodes a selected subset of genes by using a binary vector where each bit represents a gene in the dataset. If a bit is '1', it means that this gene is selected for the reduced subset and '0' indicates that the gene is not selected. Therefore, the length of the individuals is equal to the number of genes in the initial Microarray dataset.

3.3. Adapted crossover and mutation operators

Specific crossover and mutation operators, adapted to feature selection, are used in our MOGA as reproduction methods. The first one, *Subset Size-Oriented Common Feature* crossover (SSOCF) [18,19], is one of the most commonly used when facing the feature selection. As proved in [18], the SSOCF keeps useful informative patterns and produces offspring which have the same number of features (genes) as the parents. Here (see Fig. 2), the common features (bits '1') are kept by offspring and each non-shared feature is inherited from the ith parent (feature) with a probability $(n_i - n_c/n_u)$. Where n_i is the number of selected features of the ith parent, n_c is the number of commonly selected features from the mating parents, and n_u is the number of non-shared selected features.

The second operator, consisting of a *weighted* mutation, is applied with a probability of $p_{mut} = 0.1$ to the population. When an individual is mutated, its bits are flipped with different probabilities (p_{flip}) in order to adjust the number of flips from '1' to '0', and vice versa. We carried out tuning experiments with three options of flipping a simple bit as indicated in Table 1.

In this table, if we choose *one reduction* bit-flip mutation, if a given bit is '1' then it is flipped to '0' with

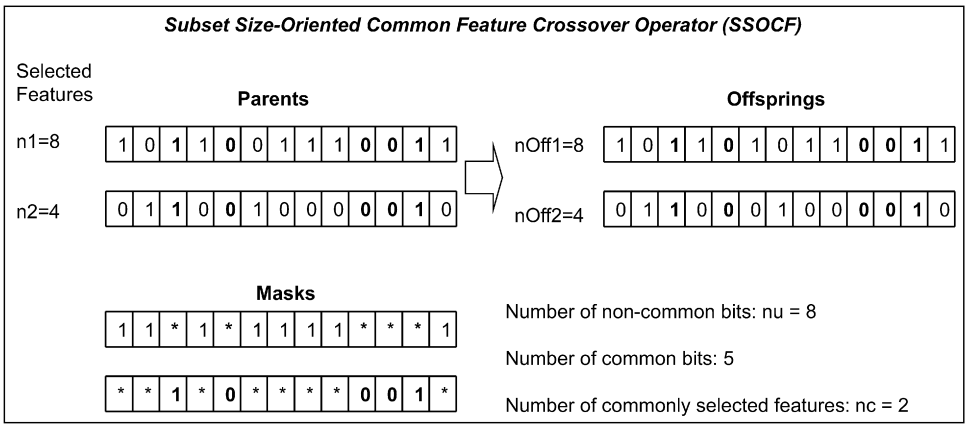

Fig. 2. Operation scheme of the SSOCF crossover. The mask promotes common shared features from parents to offspring.

Table 1
Three different kinds of bit-flip mutations. The values indicate the probability of bit-flip (p_{flip}).

Bit-flip option	1 to 0	0 to 1
uniform	0.3	0.3
zero reduction	0.3	0.6
one reduction	0.6	0.3

a probability of $p_{flip} = 0.6$, and if this bit is '0' then it is flipped to '1' with $p_{flip} = 0.3$. For the tuning experiments, we first used a *uniform* bit-flip (always $p_{flip} = 0.3$). This kind of mutation led the algorithm to obtain solutions with low percentages of sensitivity and specificity (hence accuracy) close to 65% and 60%, respectively. Secondly, when we used the *zero reduction* mutation, the feature selection procedure never obtained subsets with less than 30 genes which provokes the stagnation of the reduction process prematurely. Finally, we incorporated the *one reduction* mutation to our algorithm which obtained subsets always with less than 10 genes and percentages of sensitivity and specificity higher than 85% and 81%, respectively. Therefore, for the subsequent experimentation we used the *one reduction* mutation since it showed the best performance.

3.4. The general MOGA

The MOGA employed here generates an initial population P of individuals randomly (uniformly) initialized. Each individual, codifying a gene subset, is evaluated by means of the SVMs classifier and then 10-fold cross-validation is applied to assess the percentage of sensibility and specificity.

The population is sorted by using the dominance Crowding criteria described in Section 3.1. From this, a new elitist population E is generated selecting 10% of the best individuals. This selection is accomplished using the crowding selection operator of Section 3.1. The adapted SSOCF crossover and mutation are applied to the elitist population (E) to generate the offspring P'. Both, offspring and parent populations are then combined ($P' \cup P$). Finally, the best members replace the worst parents. When it is not possible to accommodate all the members of a particular front, that front is sorted according to the crowding distance. The individuals are selected on the basis of higher crowding distance. This selection is repeated to completely fill the new population with one of the same size the old one. The MOGA evolves for a fixed number of generations.

4. Experiments

We have implemented the proposed MOGA for gene selection in C++ using the ParadisEO [20] framework. As SVMs classifier, we have used a set of object classes provided by the LIBSVM [21] library consisting of training, testing, and validation tools. This classes were coupled with the MOGA algorithm in the evaluation phase. In this section, the experiments are described concerning the datasets, the experimentation setup, the analysis of results, and discussions.

4.1. Datasets

The used instances are classified into three well-known datasets obtained from real-word Microarray experiments. All of them were taken from the public UPITT Cancer Gene Expression Data Set Link Database in URL http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html.

- The *ALL-AML Leukemia* dataset consists of 72 tissue samples with 7129 gene expression levels. Two classes exist in this dataset: *Acute Myeloid Leukemia* (AML) and *Acute Lymphoblastic Leukemia* (ALL). The complete dataset contains 25 AML and 47 ALL samples. The original data are divided into a training set of 38 samples and a test set of 34 samples.
- The *Colon tumor* dataset consists of 62 tissue samples collected from colon-cancer patients with 2000 genes. Among them, 40 tumor biopsies are from *tumors* and 22 biopsies are from healthy parts of the colons of the same patients.
- *Types of Diffuse Large B-cell Lymphoma* dataset consists of 47 tissue samples, 24 of them are from the *germinal centre B-like group* while the remaining 23 are *activated B-like group*. Each sample is described by 4026 genes.

The expression levels were normalized in order to scale the intensities, enabling thus a comparison of the different datasets previously introduced. Each attribute was scaled to $[-1, 1]$ (as LIBSVM recommends) by means of:

$$a'_j(x_i) = 2 \times \frac{a_j(x_i) - min_j}{max_j - min_i} - 1, \tag{13}$$

where max_j and min_j correspond to the maximum and minimum gene expression values for attribute a_j in all samples.

4.2. Experimental setting

As explained in the introduction, the individuals of the population, representing gene subsets, are evaluated by means of the SVMs classifier and 10-fold cross-validation. At each iteration, the dataset is divided into ten subsets, nine of them constitute the training set and the remaining one is the test set. The SVMs is trained using the training set and then the sensitivity and the specificity of the classifier (once trained) are evaluated on the test set. This evaluation is repeated ten times, alternating the test set used each time. This method reinforces the validation process, so that the sensitivity and specificity values are the average of the ten validation folds. Moreover, such a strong validation is necessary when the number of samples is low in relation to the number of features, which is the case in this work.

Therefore, an optimal configuration of the SVMs classifier is crucial since it influences the training effectiveness. In these experiments, the main kernel parameters, γ and C coefficient (explained in Section 2), were systematically optimized in a preprocess phase for each dataset (by means of the Grid Tool of LIBSVM [21]) as follows:

- Leukemia: $C = 8$ and $\gamma = 1.220703125 \times 10^{-4}$.
- Colon: $C = 128$ and $\gamma = 1.220703125 \times 10^{-4}$.
- Lymphoma: $C = 8$ and $\gamma = 3.05175578125 \times 10^{-5}$.

These parameters were set using the SVMs classifier independently of the MOGA, in order to obtain accuracy as close to 100% as possible.

For the MOGA algorithm, the population size was fixed to 100 individuals; and 30 independent runs were performed with 2000 generations each one. The crossover and mutation operators were applied as explained in Section 3.3.

4.3. Results

In this section, we first report the results obtained by MOGA operating with three objectives. Following the standard methodology when comparing classification rates, the average and standard deviation (obtained after 30 runs) of the sensitivity, the specificity and the number of genes are shown in Table 2. The number of solutions that constitute the final Pareto front is reported in the last column. Since the number of final solutions is always higher than one, we report the best solution (in each final front) to discuss internal details like the number of genes, to assess the average and standard deviation.

Table 2
Results obtained by our MOGA in 3 objective optimization. The columns indicate the average and standard deviation of the number of genes (*NG*), the sensitivity rate (*Sen.*), the specificity rate (*Spe.*), and the number of solutions in the final Pareto front (*NSf*).

Dataset	*NG*	*Sen.* (%)	*Spe.* (%)	*NSf*
Leukemia	7.44 ± 2.14	87.63 ± 5.60	81.56 ± 10.34	9.11 ± 2.31
Colon	2.25 ± 0.20	85.93 ± 6.44	83.89 ± 4.08	12.25 ± 3.63
Lymphoma	5.00 ± 1.94	91.55 ± 7.56	86.36 ± 3.79	5.60 ± 3.43

As we can observe in Table 2, our algorithm obtains subsets of between 9 (Leukemia) and 2 (Colon) genes with rates of sensitivity and specificity higher than 85% and 81%, respectively. These results lead us to state that our MOGA performs very efficiently, since considering a *Prevalence* with a constant value, it can obtain over 85% accuracy, as explained in Section 2.2. Moreover, the number of solutions provided (between 2 and 16) reinforces this claimed effectiveness. In this sense, as well as in the number of solutions, we must consider the diversity in the final Pareto front, since this informs how different the final solutions are. Fig. 3 illustrates three representative fronts of solutions obtained in our experiments. We can observe that solutions are sufficiently scattered between 98% and 78% of sensitivity, and between 90% and 70% of specificity, despite the low number of genes in subsets.

4.4. 3-objective versus 2-objective approaches

In order to analyze the performance of our MOGA, an additional experimentation was carried out to compare the effectiveness when operating with 3 and 2 objectives.

The 3-objectives MOGA optimizes the sensitivity, the specificity and the number of genes. However, the 2-objectives MOGA optimizes the accuracy and the number of genes.

In these experiments, the subsets of genes resulting from both approaches after 30 runs were evaluated using the same cross validation method external to the selection process. Initially, the datasets were divided into two subsets, a training set and a test set. The selection algorithm was applied to the training set, and when an optimized subset of genes was obtained (either by the 3-objectives or the 2-objectives MOGA), its accuracy was evaluated on the external test set. This way, we were able to accurately compare the two different approaches.

Table 3 shows the average and standard deviation (in 30 runs) of the results reported by our MOGA in both 3-objective and 2-objective mode. We have carried out a set of statistical tests in order to find significant differences between both approaches. In each case the procedure for generating the statistical information was the following.

First a Kolmogorov–Smirnov test was performed in order to check whether the variables were normal or not and the Levene test to check the homocedasticity of samples (equality of variances). If they were (normal with equal variances), an ANOVA I test was performed, otherwise we performed a Kruskal–Wallis test. A level of significance of

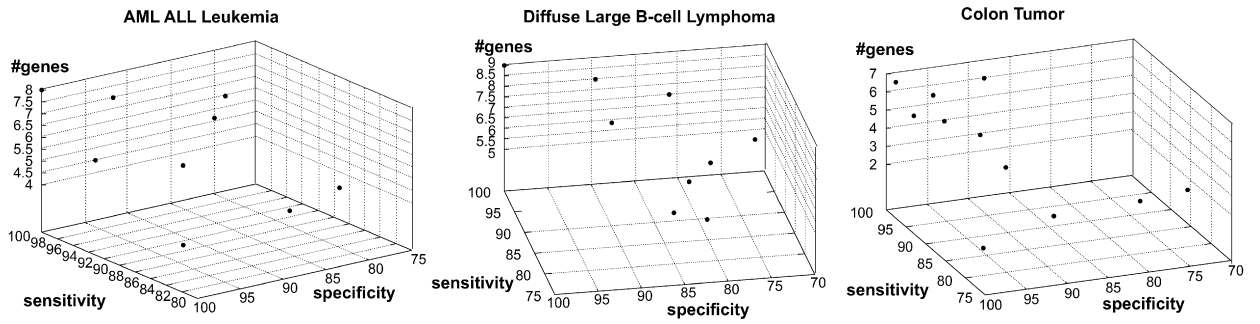

Fig. 3. Pareto fronts obtained by our MOGA in the classification of Leukemia, Lymphoma, and Colon datasets. Three objectives are optimized: the number of genes (#genes), the sensitivity, and the specificity.

Table 3
Comparison of 3 versus 2 objective MOGA. The columns indicate the average and standard deviation of the number of selected genes (*NG*), and the accuracy rate (*Ac.*). T_S indicates the percentage of samples of the external test set.

Dataset	**3 obj.**			**2 obj.**			T_S (%)	Statistical test
	NG	*Ac.* (%)	*NSf*	*NG*	*Ac.* (%)	*NSf*		
Leukemia	6 ± 2.64	$\mathbf{98.03 \pm 1.61}$	9.11 ± 2.31	7.33 ± 1.52	95.27 ± 2.21	6.66 ± 0.57	47%	+
Colon	3.33 ± 1.52	$\mathbf{89.58 \pm 1.80}$	12.25 ± 3.63	3.66 ± 0.57	86.45 ± 4.77	5.66 ± 0.57	50%	+
Lymphoma	3.75 ± 1.50	$\mathbf{96.05 \pm 3.04}$	5.60 ± 3.43	3.33 ± 1.52	92.97 ± 3.03	3 ± 1	50%	+

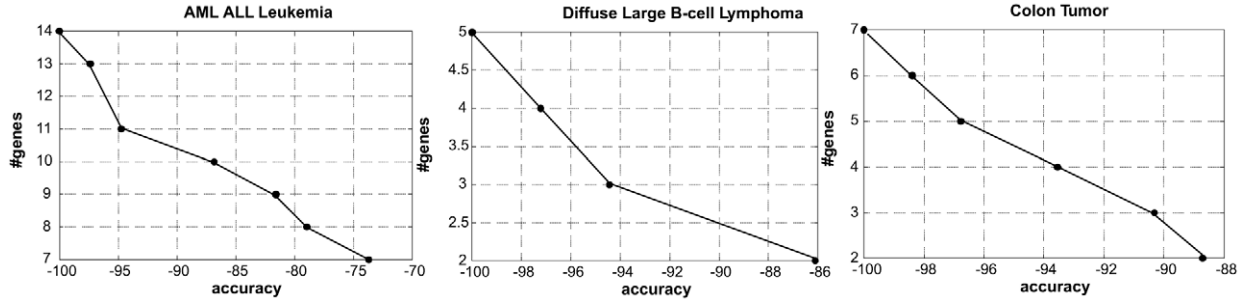

Fig. 4. Pareto fronts obtained by our MOGA in the classification of Leukemia, Lymphoma and Colon datasets. Two objectives are optimized: the number of genes (#genes), and the accuracy rate.

95% ($\alpha = 0.05$) is always applied in order to check if statistically significant differences exist. After that, we did a multiple comparison test whose results are presented in the last column of Table 3 where a plus sign means that the difference is significant (minus sign would mean that it was not).

This way, we can observe that the accuracy rate obtained by the 3-objective approach is better than the one obtained by a 2-objective approach for the three datasets. Furthermore, for all the instances, the results are statistically different (+), which leads us to ensure the final higher performance of the 3-objective approach.

As a secondary observation, the 3-objective approach obtains a larger number of solutions in the final fronts, which is an important issue when facing the decision making process. This property is clearly observable in Fig. 4, where several 2-objective fronts obtained in these experiments are shown in contrast with Fig. 3. Nevertheless, the diversity and quality of solutions shown in Fig. 4 are also suitable for gene selection, leading us recommend our MOGA with 2 objectives for other future scenarios.

4.5. Comparison with other approaches

In this section we first compare the performance of our MOGA (in 3-objective mode) with a base-line method for the gene selection. This method runs a K-Mean procedure for clustering, in which we have used the same number of genes of the final subsets selected by MOGA as representative centroids ($K = NG$). Then, each resulting subset (gene centroids) is used to train the SVM classifier (configured as explained in Section 4.2), and cross-validated with an external test set.

For this purpose, we used the K-Means procedure available in Weka tools for data mining [23]. A number of 30 independent runs were performed with 2000 iterations of each one in order to obtain as accurate a result as possible. Table 4 shows the results obtained by K-Means and MOGA in terms of mean and standard deviation of the accuracy percentage.

We can observe in this table that MOGA clearly outperforms the K-Means procedure in all the datasets. Specifically, the difference regarding the accuracy percentage in the lymphoma dataset (57.89 ± 1.11 of K-Means in con-

Table 4
Comparison with base-line method: *K*-Means clustering. The columns indicate the average and standard deviation of the number of selected genes (*NG*), and accuracy rate (*Ac.*). T_S indicates the percentage of samples of the external test set.

Dataset	*NG*	*K*-Means *Ac.* (%)	MOGA 3 obj. *Ac.* (%)	T_S (%)
Leukemia	6 ± 2.64	85.29 ± 1.02	$\mathbf{98.03 \pm 1.61}$	47%
Colon	3.33 ± 1.52	78.12 ± 2.70	$\mathbf{89.58 \pm 1.80}$	50%
Lymphoma	3.75 ± 1.50	57.89 ± 1.11	$\mathbf{96.05 \pm 3.04}$	50%

trast with 96.05 ± 3.04 of MOGA 3 obj.) gives us some insights into the power of our proposal. We would expect these differences in results, since *K*-Means is a naive method without any information about the problem in its procedure. For this reason, we have carried out further comparisons with two related metaheuristic approaches found in the literature.

The first work, Liu and Iba (2002) [8], consists of a multiobjective evolutionary algorithm which optimizes simultaneously 3-objectives: the misclassification rate, the difference in the error rate among the classes, and the number of selected genes. In the second approach, Hernandez et al. (2007) [22], a genetic algorithm embedded with a pre-filtering criteria is used, which optimizes an aggregative function using the accuracy rate and the number of genes.

Table 5 summarizes our results together with those reported in [8] and [22] on the same three datasets. As shown in bold face, the accuracy rate reported by our MOGA is the best in all the datasets, although it is clear that the number of genes selected by [22] in the Leukemia dataset is smaller. In addition, our results are competitive with respect to those reported in a third work [7]. For this reason, we can claim that our approach shows an efficient and better performance in comparison with existing state of the art algorithms.

4.6. Biological analysis

Finally, in this section we provide a biological analysis of the computed gene subsets. Although this biological study is very important, only a few papers considered it in the past [24,25]. Most articles addressed just the machine learning problem regardless of the actual meaning of the datasets. We will show here the much broader impact of our technique, capable of computing real biological ensembles of genes that have been suggested in the domain only (e.g., Science [26]).

In Fig. 5, a graphical distribution of the most frequently obtained genes in 30 independent executions of the MOGA (in 3-objective mode) are reported. We have used the Leukemia dataset, since it is the one commonly studied by other related works in the literature. In this figure, we highlight in bold face the genes also reported in the list of the 30 most important genes (selected from 7129 in Leukemia) suggested in Golub et al. [26]. In Table 6, we arrange these genes by means of the rank assigned in the Golub et al. list (column 1 in the referenced table).

The genes reported in Table 6 were also selected as the most informative genes in recent specialized works. Specifically, Dramiński et al. [24] used a Monte Carlo method for feature selection and supervised classification on Leukemia and Lymphoma datasets. Wang and Zhu [25] proposed a Nearest Shrunken Centroid (NSC) classifier on the Leukemia dataset. Both works assign the gene M31166_at great importance, which matches with our main results. In addition, in Table 7 we present a list of

Table 5
Comparison with other authors. The columns indicate the average and standard deviation of the number of selected genes (*NG*), the accuracy rate (*Ac.*), and the number of solutions in the final Pareto front (*NSf*). T_S indicates the percentage of samples of the external test set.

Dataset	Liu and Iba (2002) [8]			Hernandez et al. (2007) [22]			MOGA 3 obj.		
	NG	*Ac.* (%)	T_S (%)	*NG*	*Ac.* (%)	T_S (%)	*NG*	*Ac.* (%)	T_S (%)
Leukemia	15.2 ± 4.54	90.00 ± 7.00	30%	3.17 ± 1.16	91.5 ± 5.9	47%	6 ± 2.64	$\mathbf{98.03 \pm 1.61}$	47%
Colon	11.4 ± 4.27	80.00 ± 8.3	30%	7.05 ± 1.07	84.6 ± 6.6	50%	3.33 ± 1.52	$\mathbf{89.58 \pm 1.80}$	50%
Lymphoma	12.9 ± 4.40	90.00 ± 3.4	30%	5.29 ± 1.31	93.3 ± 3.1	50%	3.75 ± 1.50	$\mathbf{96.05 \pm 3.04}$	50%

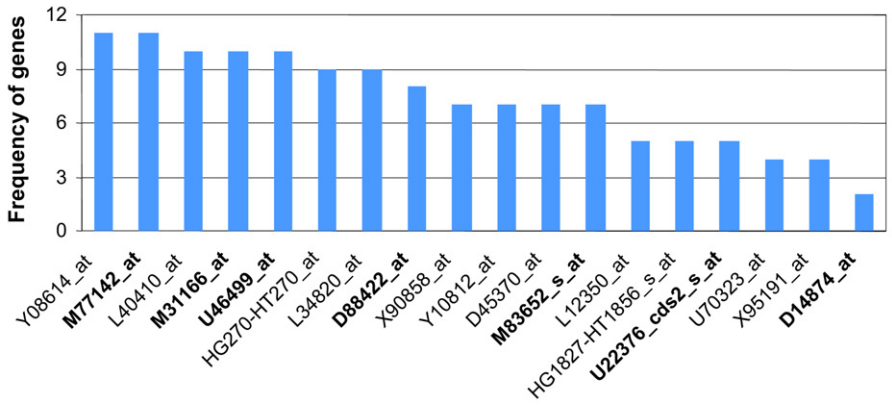

Fig. 5. Distribution of the most frequently obtained genes (in 30 independent executions) by our MOGA in Leukemia dataset.

Table 6
Top 7 genes ranked with MOGA (3-objectives) also reported in the list of the 30 most important genes suggested by Golub et al. on the Leukemia dataset.

Rank	Index	Accession	Gene description
2	1926	M31166_at	"PTX3 Pentaxin-related gene, rapidly induced by IL-1 beta"
5	760	D88422_at	CYSTATIN A
13	2233	M77142_at	NUCLEOLYSIN TIA-1
17	3256	U46499_at	"GLUTATHIONE S-TRANSFERASE, MICROSOMAL"
18	6379	M83652_s_at	"PFC Properdin P factor, complement"
22	249	D14874_at	ADM Adrenomedullin
29	5772	U22376_cds2_s_at	"C-myb gene extracted from Human (c-myb) gene"

Table 7
New genes suggested in this work for the Leukemia dataset.

Index	Accession	Gene description
5975	J03778_s_at	MICROTUBULE-ASSOCIATED PROTEIN TAU
1584	L40410_at	Thyroid receptor interactor (TRIP8) mRNA, 3′ end of cds
865	HG270-HT270_at	Lymphocyte Chemoattractant Factor
1494	L34820_at	NAD+-dependent succinate-semialdehyde dehydrogenase (SSADH) mRNA, 3′ end
4780	X90858_at	Uridine phosphorylase
5021	Y10812_at	GB DEF = Fructose-1,6-bisphosphatase
442	D45370 at	ApM2 mRNA for GS2374 (unknown product specific to adipose tissue)
1295	L12350_at	THBS2 Thrombospondin 2
5731	HG1827-HT1856_s_at	Cytochrome P450, Subfamily Iic, Alt. Splice Form 2
3635	U70323_at	SCA2 Spinocerebellar ataxia 2
4835	X95191_at	GB DEF = Delta-sarcoglycan

new genes that we consider to be important, since they were the ones that overlapped most in the resulting subsets (together with the ones shown in Table 6) in our experiments.

5. Conclusions

In this paper, we propose the use of the sensitivity and the specificity rates, in addition to the number of selected genes, as three main objectives to optimize when facing the gene selection and classification of DNA Microarrays. We have used a NSGA-II based MOGA algorithm which evolves optimizing simultaneously these three objectives. In this algorithm, the classification task is accomplished by SVMs, and 10-fold cross-validation is applied to the resulting subsets to evaluate the solutions. The effectiveness of this approach is proved on public Microarray datasets (Leukemia, Lymphoma, and Colon).

The first statistical analysis confirms that breaking up the accuracy factor among the sensitivity and the specificity factors can increase, in terms of quality and diversity, the number of good solutions. The comparisons presented in Table 2 show the difference in the accuracy percentage and number of solutions of both strategies: 2 and 3 objectives.

The accuracy percentages of 98.03 ± 1.61 obtained by MOGA 3 obj., in contrast with the ones of 95.27 ± 2.21 obtained by MOGA 2 obj. for Leukemia guarantee the usefulness of our proposal. In a second analysis, we compare our approach with a naive method based on K-Means clustering, and with two related multiobjective approaches. Our results suggest that the MOGA 3 obj. is highly appropriate for solving the gene selection, outperforming the compared techniques for all the datasets.

A final biological analysis reports a list of the most representative genes selected by our approach. We can notice that seven of these genes were also reported as the most relevant ones in the original work of Golub et al. concerning the Leukemia dataset. Specifically, the M31166_at gene was also considered very important in related works, which is consistent with our results.

As to future work, we are interested in evaluating our algorithm in new Microarray datasets, and plan to test and compare different MOGA approaches (IBEA, SPEA, etc.) in order to offer fresh points of view to this problem.

References

[1] A. Pease, D. Solas, E. Sullivan, M. Cronin, C.P. Holmes, S. Fodor, Light-generated oligonucleotide arrays for rapid DNA sequence analysis, Proc. Natl. Acad. Sci. USA 96 (1994) 5022–5026.
[2] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46 (1–3) (2002) 389–422.
[3] M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Trans. Comput. 26 (1977) 917–922.
[4] E. Alba, J. García-Nieto, L. Jourdan, E.-G. Talbi, Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms, in: IEEE Congress on Evolutionary Computation CEC-07, Singapore, 2007, pp. 284–290.
[5] E.B. Huerta, B. Duval, J.-K. Hao, A hybrid GASVM approach for gene selection and classification of microarray data, in: F. Rothlauf, et al. (Eds.), EvoWorkshops, in: LNCS, vol. 3907, Springer, 2006, pp. 34–44.
[6] T. Juliusdottir, D. Corne, E. Keedwell, A. Narayanan, Two-phase EA/K-NN for feature selection and classification in cancer microarray datasets, in: CIBCB, 2005, pp. 1–8.
[7] K. Deb, A. Raji, Reliable classification of two-class cancer data using evolutionary algorithms, BioSystems 72 (2003) 111–129.
[8] J. Liu, H. Iba, Selecting informative genes using a multiobjective evolutionary algorithm, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC '02, vol. 1, 2002, pp. 297–302.
[9] M. Banerjee, S. Mitra, H. Banka, Evolutionary rough feature selection in gene expression data, IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 37 (4) (2007) 622–632.
[10] C. Metz, Basic principles of ROC analysis, Seminars in Nuclear Medicine 8 (4) (1978) 283–298.
[11] A.J. Alberg, J.W. Park, B.W. Hager, M.V. Brock, M. Diener-West, The use of "overall accuracy" to evaluate the validity of screening or

diagnostic tests, J. of General Internal Medicine 19 (5) (2004) 460–465.
[12] M. Kupinski, M. Anastasio, Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves, IEEE Trans. on Medical Imaging 18 (8) (1999) 675–685.
[13] R.M. Everson, J.E. Fieldsend, Multi-class ROC analysis from a multi-objective optimisation perspective, Pattern Recogn. Lett. 27 (8) (2006) 918–927.
[14] H. Liu, H. Motoda, Feature Extraction, Construction and Selection: A Data Mining Perspective, Kluwer Academic Publishers, Norwell, MA, USA, 1998.
[15] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[16] T. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Support vector machines classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.
[17] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. on EC 6 (2) (2002) 182–197.
[18] C. Emmanouilidis, A. Hunter, J. MacIntyre, A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator, in: IEEE Congress on Evolutionary Computation, California, USA, 2000, pp. 309–316.
[19] L. Jourdan, C. Dhaenens, E.G. Talbi, S. Gallina, A data mining approach to discover genetic and environmental factors involved in multifactorial diseases, Knowledge-Based Systems 15 (4) (2002) 235–242.
[20] A. Liefooghe, M. Basseur, L. Jourdan, E.-G. Talbi, ParadisEO-MOEO: A framework for evolutionary multi-objective optimization, in: EMO, in: LNCS, 2007, pp. 386–400.
[21] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, Software available at URL http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2002.
[22] J. Hernandez, B. Duval, J.-K. Hao, A genetic embedded approach for gene selection and classification of microarray data, in: E. Marchiori, et al. (Eds.), EvoBIO, in: LNCS, 2007, pp. 90–101.
[23] I. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, M. Kaufmann, 2005.
[24] M. Dramiński, A. Rada-Iglesias, S. Enroth, C. Wadelius, J. Koronacki, J. Komorowski, Monte Carlo feature selection for supervised classification, Bioinformatics 24 (1) (2008) 110–117.
[25] S. Wang, J. Zhu, Improved centroids estimation for the nearest shrunken centroid classifier, Bioinformatics 32 (2) (2007) 972–979.
[26] R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.