A Memetic Algorithm for Gene Selection and Molecular Classification of Cancer

Béatrice Duval
LERIA, Université d'Angers
2 Boulevard Lavoisier
49045 Angers cedex 01, France
bd@info.univ-angers.fr

Jin-Kao Hao
LERIA, Université d'Angers
2 Boulevard Lavoisier
49045 Angers cedex 01, France
hao@info.univ-angers.fr

Jose Crispin Hernandez Hernandez
Instituto Tecnologico de Apizaco
av Instituto Tecnologico S/N
90300, Apizaco, Mexico
seinez@toro.itapizaco.edu.mx

ABSTRACT

Choosing a small subset of genes that enables a good classification of diseases on the basis of microarray data is a difficult optimization problem. This paper presents a memetic algorithm, called MAGS, to deal with gene selection for supervised classification of microarray data. MAGS is based on an embedded approach for attribute selection where a classifier tightly interacts with the selection process. The strength of MAGS relies on the synergy created by combining a problem specific crossover operator and a dedicated local search procedure, both being guided by relevant information from a SVM classifier. Computational experiments on 8 well-known microarray datasets show that our memetic algorithm is very competitive compared with some recently published studies.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE—*Problem Solving, Control Methods, Search*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation; Feature evaluation and selection*

General Terms

Algorithms

Keywords

Classification, gene selection, local search, memetic algorithm, specialized crossover

1. INTRODUCTION

Microarray technology enables to measure the expression of thousands of genes to identify changes in expression between different biological states. Previous works [9, 3] have shown that this technology can provide new efficient diagnosis tools for the recognition of diseases like cancers or for the discrimination between different kinds of tumors. This discovery has stimulated an increasing interest in the bioinformatics community in order to design more powerful decision-making tools for the molecular diagnosis of cancers.

The basic research problem can be studied from the perspective of supervised classification where the available microarray data serve as training data to obtain new classifiers. However, given the cost of the microarray technology, the available data typically contain a very limited number of samples. Consequently, classification of microarray data is faced with the difficult problem known as ”the curse of dimensionality” because the data are described by a great number of attributes whereas only a few dozen of samples are described. In order to limit the risk of overfitting, it is necessary to reduce the dimensionality of the data by selecting a reduced number of attributes relevant for classification.

For supervised classification, attribute selection methods (see [10] for an introduction to this subject) can be organized into three categories depending on how the selection process is combined with the classification process. *Filter methods* only consider the input data and use the data to rank each attribute according to its correlation with the class label of the given data. The top ranked attributes are then considered as the most relevant ones. This selection occurs before the classification process and is independent of the learned classifier. In *wrapper methods*, selection of relevant attributes is performed in interaction with the classifier. To explore the space of attribute subsets, a search algorithm is ”wrapped” around the classification model. For example, backward selection begins with all the possible attributes and iteratively removes the least relevant attribute. At each step a classifier is trained and tested to evaluate the quality of a potential subset. *Embedded methods* are similar to wrapper methods because they rely on a classifier to evaluate candidate subsets but they are characterized by a deeper interaction between the search of an optimal subset and the classifier construction. Our memetic algorithm belongs to this class of embedded methods.

As the search space of possible subsets grows exponentially with the number of attributes, heuristic methods are good candidates to tackle the difficult problem of finding an optimal subset for classification. Genetic algorithms for gene selection have been previously proposed in many studies [17, 24, 13, 27, 12, 1, 18]. They are often followed by

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

a post-processing step that further reinforces the selection process. For example, in [18] different gene subsets are obtained from different training sets and an analysis of gene frequencies enables to propose a final gene subset. Notice that except some very recent studies like [14], most existing genetic algorithms are based on blind genetic operators such as random crossover and mutation.

In this paper we propose a memetic algorithm that introduces local search into a genetic algorithm. This method is an embedded approach for gene selection where a Support Vector Machine (SVM) tightly interacts with the search process. The SVM is used not only to evaluate the quality of candidate gene subsets but also to provide ranking information about each gene. This enables to design a specialized crossover operator that combines the relevant attributes from two parents to build interesting offspring. In the same way, local search relies on an informed move operator according to information provided by SVM. Experimental results show that our memetic algorithm achieves very competitive results on eight largely studied datasets.

The rest of this paper is organized as follows. In section 2, we recall the main characteristics of SVM and discuss the difficult problem of estimating classification accuracy when the number of samples is very low. We then describe the different components of our memetic algorithm in section 3. In section 4, we describe the experiments and show computational results to assess the effectiveness of our algorithm.

2. SVM CLASSIFICATION AND ACCURACY ESTIMATION

2.1 SVM

SVMs are state-of-the-art classifiers that solve a binary classification problem by searching a decision boundary that has the maximum margin with the examples. SVMs handle complex decision boundaries by using linear machines in a high dimensional attribute space, implicitly represented by a kernel function. In this work, we only consider linear SVMs because they are known to be well suited to the datasets that we consider [11, 25, 22] and they offer a clear biological interpretation of the results.

For a given training set of labeled samples, a linear SVM determines an optimal hyperplane that divides the positively and the negatively labeled samples with the maximum margin of separation. A noteworthy property of SVM is that the hyperplane only depends on a small number of training examples called the support vectors, they are the closest training examples to the decision boundary and they determine the margin.

Formally, we consider a training set of n samples belonging to two classes; each sample is noted $\{X_i, y_i\}$ where $\{X_i\}$ is the vector of dimension m of attribute values describing the sample and y_i the class label.

A soft-margin linear SVM classifier aims at solving the following optimization problem:

$$\min_{w,b,\xi_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{1}$$

subject to $y_i (w \cdot X_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$, $i = 1, ..., n$.

In this formulation, w is the weight vector that determines the separating hyperplane; C is a given penalty term that controls the cost of misclassification errors. To solve this optimization problem, it is convenient to consider the dual formulation [5]:

$$\min_{\alpha_i} \frac{1}{2} \sum_{i=1}^{n} \sum_{l=1}^{n} \alpha_i \alpha_l y_i y_l X_i \cdot X_l - \sum_{i=1}^{n} \alpha_i \tag{2}$$

subject to $\sum_{i=1}^{n} y_i \alpha_i = 0$ and $0 \leq \alpha_i \leq C$.

The decision function of the linear SVM classifier for an input vector X is given by $f(X) = w \cdot X + b$ with $w = \sum_{i=1}^{n} \alpha_i y_i X_i$ and $b = y_i - w \cdot X_i$. The weight vector w is a linear combination of the training samples. Most weights α_i are zero and the training samples with non-zero weights are the support vectors. Moreover, the maximum margin M is given by

$$M = \frac{2}{\|w\|} \tag{3}$$

2.2 Feature ranking by SVM

As discussed in [11], the weights of a linear discriminant classifier can be used to rank the features for selection purposes. More precisely, in a backward selection method, the idea is to start with all the features and to iterate the removal of the least informative feature. To determine which feature can be removed, one can consider the feature that has the least influence on the cost function of the classification process. For a linear SVM, the cost function is defined by $\frac{1}{2}||w||^2$. So given a SVM with weight vector w, we can define the ranking coefficient vector c given by:

$$c_j = (w_j)^2 \;\; j = 1, \ldots, m \tag{4}$$

Intuitively, in order to select informative genes, one can use the orientation of the separating hyperplane found by a linear SVM. If the plane is orthogonal to a particular feature dimension, then that feature is informative, and *vice versa*. As we show in the next section, the coefficient vector c provides useful ranking information that enables the design of dedicated operators in our hybrid algorithm for gene selection and classification.

2.3 Cross-validation estimation of accuracy

In order to assess the predictive capability of a classifier, it is necessary to estimate the accuracy of the classifier induced by a learning algorithm. When variable selection is required, the accuracies of classifiers built on different subsets of variables are compared to choose the most appropriate subset. Previous works [16, 6] have shown that cross-validation, and more specifically 10-fold cross-validation, provides an accuracy estimate with low variance.

To apply the k-fold cross-validation method, the initial dataset D is split into k subsets of approximately the same size $D_1, \ldots, D_k$. The learning algorithm is applied k times to build k classifiers: in step i, the data subset D_i is left out as a test set, the classifier is induced from the training dataset $D - D_i$ and its accuracy Acc_i is estimated on D_i. The accuracy estimate computed by k-fold cross-validation is then the mean of the Acc_i, for $1 \leq i \leq k$.

In this work, cross-validation is used in the fitness function that evaluates a given subset of genes. Moreover, in the experimental section, we compare different strategies of gene selection and the accuracy estimate obtained by cross-validation is one of the elements that characterize the results. In this case, it is important to notice that our objective is to evaluate both the gene selection process and the classification process. Therefore, as shown in [4], it is necessary to include gene selection into the cross-validation schema. That means that the dataset must be split before gene selection is achieved: each step of cross-validation performs gene selection and classification. Without this careful design of the experimental protocol, the accuracy results may be overestimated.

3. A MEMETIC ALGORITHM FOR GENE SELECTION AND CLASSIFICATION

In this section, we present our memetic approach (called MAGS) for gene selection and classification of microarray data. We explain the basic ingredients and their underlying rational. As the initial number of attributes can reach several tens of thousands, our method begins by a pre-selection step where we use a filter criterion to obtain a group G_p of p top ranked genes. After experiments with different criteria (t-statistics, wilcoxon test, ...), the BW ratio introduced in [7] was chosen. Then our memetic algorithm is applied in the search space 2^p to select, from G_p a gene subset of smaller size that provides a high classification accuracy. In this work, the number p was fixed to 75. Notice that the algorithm can be applied to explore larger search space with more pre-selected genes (larger p). In this case, more computational efforts will be needed.

3.1 Outline of the MAGS algorithm

Our Memetic Algorithm for Gene Selection (MAGS) first builds an initial population P of gene subsets and then performs a number of generations. There are several ways to generate the individuals of the initial population. In our case, each individual is randomly generated such that the number of genes in each solution varies between $p * 0.9$ and $p * 0.6$.

At each generation, a new population replaces the previous population P. The new population P' is obtained from P in the following way. A certain number, $NbElitism$, of the best individuals of P are copied to P'. This elitism mechanism ensures that the best individuals are conserved along the generations. The rest of P' is completed with individuals obtained by crossover (Section 3.4) and local search (Section 3.5). More precisely, our specialized crossover operator applies to two parents and gives one offspring; the local search operator, based on the Iterated Local Search (ILS) metaheuristic, is applied to improve each offspring and the resulting individual is then added to P'. This process is iterated until a maximum number of generations is reached. The general MAGS procedure is described in Algorithm 1.

3.2 Search space and representation of individuals

An individual $I =< I^g, I^c >$ is composed of two parts I^g and I^c called respectively *gene subset vector* and *ranking coefficient vector* [14]. The first part, I^g, is a binary vector of fixed length p. Each bit I^g_i $(i = 1...p)$ corresponds to a particular gene and indicates whether or not the gene is selected. The second part, I^c, is a positive real vector of fixed length p and corresponds to the ranking coefficient vector c (Equation 4) of a linear SVM classifier trained on I^g. I^c indicates thus for each selected gene the importance of this gene for the SVM classifier.

Algorithm 1: Memetic Algorithm for Gene Selection - MAGS

```
Parameters: $|P|, NbElitism, f_{MA}, maxIter$
begin
    Build the initial population $P$
    Evaluate each individual of $P$ according to the
    fitness function $f_{MA}$
    $nbIter \leftarrow 1$
    while $nbIter < maxIter$ do
        Generate the temporary population $P'$
        Copy the $NbElitism$ best individuals of $P$ into
        $P'$
        Produce $OffSpring$: a set of $|P| - NbElitism$
        individuals produced by CrossOver operator
        Each individual $I$ of $OffSpring$ is improved by
        local search: $I$=ILS($I$)
        Each individual improved $I$ of $OffSpring$ is
        added to $P'$
        The population $P$ is replaced by $P'$
        $nbIter \leftarrow nbIter + 1$
    end
    return The best found gene subset and the
    associated SVM classifier
end
```

Therefore, an individual represents a candidate gene subset with additional information on each selected gene with respect to the SVM classifier. The gene subset vector of an individual will be evaluated by a linear SVM classifier while the ranking coefficients obtained during this evaluation provide useful information for the evolutionary process. Notice that most previous studies use only the classical representation by a gene subset vector.

3.3 Fitness function

The quality of a candidate gene subset is evaluated by its ability to obtain a good classification. To calculate the fitness of an individual, a SVM classifier is trained with this representation and its classification accuracy is estimated by 10-fold cross validation. Due to the mechanism of coregulation between genes, the microarray datasets contain correlated information. We observe that many different subsets of genes can achieve the same performance of classification. Obviously, this phenomenon is reinforced by the very low number of samples available in each dataset.

Therefore in order to further discriminate gene subsets that give the same accuracy, we propose the following evaluation (fitness) function f_{MA} (Equation 5) that also considers the SVM margin to evaluate the quality of a gene subset. Let us recall that the margin of the SVM classifier evaluates the distance between the decision hyperplane and the two classes. So a greater margin indicates a better discrimination.

More formally, the fitness function f_{MA} of our memetic algorithm can be written as follows:

$$f_{MA}(I) =< Acc(I^g), M_{SVM}(I^g) > \tag{5}$$

where

- $Acc(I^g)$ is the classification accuracy of the SVM classifier using the set of genes of I^g,
- $M_{SVM}(I^g)$ is the maximum margin of the SVM classifier, given by Equation 3.

Now given two candidate solutions I and J, it is possible to compare them: $f_{MA}(I)$ is better than $f_{MA}(J)$, denoted by $f_{MA}(I) > f_{MA}(J)$, if the following condition is satisfied: $f_{MA}(I) > f_{MA}(J) \Leftrightarrow Acc(I^g) > Acc(J^g)$ or $Acc(I^g) = Acc(J^g) \wedge M_{SVM}(I^g) > M_{SVM}(J^g)$.

3.4 Crossover operator

Crossover is one of the key evolution operators for any effective GA and needs a particularly careful design. As our goal is to obtain small subsets of selected genes with a high classification accuracy, we have designed a highly specialized crossover operator following two fundamental principles [14]: 1) to conserve the genes shared by both parents and 2) to preserve "high quality" genes from each parent even if they are not shared by both parents. The notion of "quality" of a gene here is defined by the corresponding ranking coefficient stored in I^c. Notice that applying the first principle will have as main effect of getting smaller and smaller gene subsets while applying the second principle allows us to keep up good genes along the search process.

Let $I =< I^g, I^c >$ and $J =< J^g, J^c >$ be two selected individuals (parents), we combine I and J to obtain a single child $K =< K^g, K^c >$ using the following steps:

1. Extract the subset of genes shared by both parents by boolean logic AND operator ($\otimes$) and arrange them in an intermediary gene subset vector F.

   $$F = I^g \otimes J^g$$

2. For the subset of genes obtained from the first step, extract the maximum coefficients max_I and max_J accordingly from their original ranking vectors I^c and J^c.

   $$max_I = max\,\{I^c_i \mid i \; such \; that \; F_i = 1\}$$

   and

   $$max_J = max\,\{J^c_i \mid i \; such \; that \; F_i = 1\}$$

3. This step aims to transmit high quality genes from each parent I and J which are not retained by the logic AND operator in the first step. These are genes with a ranking coefficient greater than max_I and max_J. The genes selected from I and J are stored in two intermediary vectors AI and AJ

   $$AI_i = \begin{cases} 1 & if \; I^g_i = 1 \; and \; F_i = 0 \; and \; I^c_i > max_I \\ 0 & otherwise \end{cases}$$

   and

   $$AJ_i = \begin{cases} 1 & if \; J^g_i = 1 \; and \; F_i = 0 \; and \; J^c_i > max_J \\ 0 & otherwise \end{cases}$$

4. The gene subset vector K^g of the offspring K is then obtained by grouping all the genes of F, AI and AJ using the logical "OR" operator ($\oplus$).

   $$K^g = F \oplus AI \oplus AJ$$

The ranking coefficient vector K^c will be filled up when the individual K is evaluated by the SVM based fitness function.

From the search perspective, this crossover operator essentially plays a guided diversification role. The intensification of the search will mainly be driven by an iterated local search algorithm.

3.5 Iterated local search

An individual I produced by crossover will be improved by local search before being inserted into the population. Our local search operator is ensured by an Iterated Local Search (ILS) procedure [21]. ILS alternates between a local search (e.g. descent) procedure and a perturbation operator. Starting from an initial solution, the local search procedure is used to reach a local optimum. Then the perturbation operator is employed to displace the solution to a new region, whereupon a new round of local search starts.

We have seen that the crossover operator combines relevant genes from two parents to form a new offspring. The number of genes in the new individual can be quite different from the number of genes of its parents. To complete this process that ensures diversity of the population, our ILS operator focuses its search on gene subsets of fixed size, ensuring thus an intensified exploitation within a limited search space [15].

So given an offspring I with g genes, created by crossover, the neighbors explored by the ILS operator are gene subsets that contain exactly g genes. The ILS process tries to improve the accuracy of classification obtained from these subsets. The move operator that defines the neighborhood is therefore a "drop/add" operator informed by semantic knowledge about the relevance of genes for classification.

More precisely, for an individual $I =< I^g, I^c >$ represented by $I^g = (g_1, g_2...g_p)$ and $I^c = (c_1, c_2...c_p)$, our move operator drops the least informative gene g_i (i.e. having the smallest non-null coefficient c_i). Consequently, g_i=1 becomes g_i=0. One then adds to the current solution another gene g_j ($j \neq i$) (g_j=0 becomes g_j=1). Such a move is denoted by $mv_{i,j}$. Applying a move $mv_{i,j}$ to a solution I leads to a new solution I'. Such an application is denoted by: $I' = mv_{i,j} \oplus I$.

From an individual, ILS iteratively applies this move operator until a local optimum is reached. Then a perturbation operator is applied to explore alternative solutions to this local optimum. Perturbation changes the previous best solution by some random moves (random drop/add of genes). The altered solution constitutes a new starting point for a new round of local search. During all these steps a tabu list is used to prevent the method from cycling. More precisely, each time a gene g_i is dropped from the current individual, g_i is added to the tabu list and cannot be re-added to the solution during a certain number of moves (see Algorithm 2).

Algorithm 2: Iterated Local Search Algorithm with a tabu list

```
Data: $I$: an individual obtained by crossover
Result: $I'$: an improvement of $I$
Parameters: $tl$: size of the tabu list
while not Stop-Condition do
    Choose the best authorized move $mv_{i,j}$ and apply
    the move to $I$: $I' = mv_{i,j} \oplus I$
    Add gene $i$ in the tabu list for $tl$ iterations
    if a local optimum is reached then
        Perturbation: $I = perturbation(I')$
    end
end
```

3.6 Parameters of the memetic algorithm

Now that we have defined the different components of our method described in Algorithm 1, we can specify the different parameters used in our experiments.

The population size was fixed to 30. $NbElitism$, the number of the best individuals copied to the next population was fixed to 5, and $maxIter$ was fixed to 30. All these parameters were determined experimentally while trying to limit the computational cost and to obtain the best results for gene selection and classification. Note these parameter values are used for all the experimentations presented in Section 4.

4. EXPERIMENTATIONS AND RESULTS

4.1 Datasets

Since the first publications about molecular classification of cancer [3, 9], several datasets have been studied and are publicly available for example on the Kent Ridge Biomedical repository (http://datam.i2r.a-star.edu.sg/datasets/krbd/). Table 1 gives a brief description of the datasets used in our experiments. It is important to test a method on several datasets because special characteristics can be observed in some data and the problems may be of different difficulty. For example, it is now well recognized that the two kinds of Leukemia in the dataset can be easily discriminated even with a very small number of genes while the Colon cancer dataset is more difficult, perhaps because it contains some mis-classified samples [8].

To report the computational results (Tables 2 and 3), each dataset is independently solved 10 times and statistics based on the results of these runs are reported.

Table 1: Summary of datasets used for experimentation

Dataset	#Genes	#Samples
Colon Cancer	2000	62
Leukemia	7129	72
Breast Cancer	24481	97
Lung Cancer	12533	181
Prostate Cancer	12600	109
Ovarian Cancer	15154	253
CNS Cancer	7129	60
Lymphoma	4026	47

4.2 Comparison of MAGS with a genetic algorithm and a local search procedure

In order to evaluate the importance of the different components of our memetic algorithm MAGS, we first compare its results with the results obtained by a genetic algorithm and a local search algorithm. Such a comparison allows us to highlight the importance of combining genetic and local search with a single process. The experiments are conducted on the 8 datasets described in Table 1 and the results are presented in Table 2.

The local search algorithm is the ILS algorithm used in MAGS. The initial candidate solution can be randomly created with a risk of being of bad quality. For this reason, we devise a simple way to obtain a set of ”not-too-bad” initial individuals as follows. To generate each individual (Section 3.2), we first generate randomly $l = 10$ solutions such that the number of genes in each solution varies between $p * 0.9$ and $p * 0.6$ (p being the number of pre-selected genes with a filter, see Section 3.1), from which the best solution according to the evaluation function f_{MA} given by Equation 5 (Section 3.3).

Recall that ILS considers neighbors that all contain the same number of genes. To reduce the number of selected genes, we combine this exploration with a reduction phase that withdraws the least relevant gene from the current solution, whereupon a new ILS is applied. This two-stage process stops when removing the least interesting gene worsens the classification accuracy on the training data. This method, named ILS+Reduction, is summarized in algorithm 3.

Algorithm 3: ILS+Reduction Procedure

```
begin
    Generate an initial solution $I^g$;
    repeat
        Evaluate $I^g$ using the SVM classifier and fill $I^c$ ;
        $I =< I^g, I^c >$ /* $I$ is the current solution*/ ;
        $I$ = ILS($I$) /* ILS phase: apply ILS to improve
        solution $I$;
        $I^g = I^g - \{g_i\}$ /* Gene reduction phase: remove
        the least informative gene from the best solution
        found by ILS phase */;
    until stop condition is verified;
end
```

We also provide the results obtained when the sole Genetic Algorithm (GA) is applied. Our GA is the MAGS algorithm where its ILS operator is disabled. More precisely the GA uses the specialized crossover operator presented in section 3.4, but replaces the ILS operator by a standard random mutation operator. In order to select gene sets of small sizes, the GA uses a fitness function f_{GA} that combines the two objectives: maximizing the accuracy and minimizing the number of genes:

$$f_{GA}(I) = \frac{Acc(I^g) + \left(1 - \frac{|I^g|}{p}\right)}{2} \qquad (6)$$

The first term $Acc(I^g)$ is the same as in f_{MA} (formula 5): it is the classification accuracy obtained with the SVM classifier and evaluated via 10-fold cross-validation. The second term ensures that for two gene subsets having an equal classification accuracy, the smaller one is preferred. For a given individual I, this fitness function leads to a positive real fitness value $f_{GA}(I)$ (higher values are better).

Table 2 contrasts the results of MAGS with the results of ILS+Reduction and GA on the 8 datasets. Each cell shows two pieces of information: the average accuracy estimate (over 10 runs) with the standard deviation (see Section 2.3) and the average number of selected genes with the standard deviation. Due to the fact that the fitness function used by GA (formula 6) is different from that employed by MAGS and ILS (formula 5), it would be difficult to compare the number of selected genes obtained by different algorithms. So the most important information provided in Table 2 concerns the accuracy estimates since they are calculated exactly in the same way for the three cases (MAGS, ILS+Reduction and GA).

From Table 2, we see that the results of MAGS dominate clearly those of its two competitors. The results of MAGS are better than those of GA on the eight datasets and a Wilcoxon test for paired samples confirmed that this difference is significant with a confidence level of 95% (p-value under 0.05). Since the test deals with only eight datasets, the approximation of the statistics by normal distribution is not valid and the result of the test ($R^- = 0$) must be compared with tabulated values. Concerning the comparison between MAGS and ILS, we observe that they give the same perfect accuracy on two datasets, ILS slightly outperforms MAGS on the Prostate dataset but MAGS dominates on the other datasets. A Wilcoxon test confirms the dominance of MAGS with a p-value of 0.05.

These results tend to show that removing from MAGS its specialized crossover or its ILS procedure weakens its search power and consequently highlight the importance of the synergy between genetic and local search.

Note that MAGS achieved a perfect classification for 4 out of the 8 datasets and the classification accuracy of at least 95% for the remaining cases.

Table 2: Results of our three heuristic methods: GA, ILS and MAGS. Each cell indicates the average classification estimate with the standard deviation and the number of selected genes with the standard deviation. Globally, MAGS dominates the two other methods.

Data	GA	ILS+Reduction	MAGS
Colon Cancer	84.6±6.6% 7.05±1.07	87.00±7.36% 8.20±2.09	98.33±5.27% 7.70±1.95
Leukemia	91.50±5.90% 3.17±1.16	91.94±4.06% 3.14±1.08	100% 6.20±3.19
Breast	85.68±4.64% 10.70±8.34	89.58±2.49% 7.90±6.27	95.78±5.46% 16.10±4.93
Lung	99.79±0.32% 6.64±3.57	99.93±0.20% 4.66±2.25	100% 7.30±3.02
Prostate	97.00±2.49% 6.76±7.12	98.41±1.48% 5.08±4.10	97.03±3.83% 25.20±7.27
Ovarian Cancer	99.98±0.12% 3.10±0.91	100% 2.52±0.54	100% 3.00±0.00
CNS	79.33±4.71% 13.16±5.83	84.00±1.65% 9.06±4.24	95.00±8.05% 20.70±4.55
Lymphoma^{47}	96.78±3.14% 7.72±6.18	100% 7.34±4.16	100% 5.70±2.16

4.3 Comparison with other works

In this section, we give a comparison of the results obtained by MAGS and some recent works. The results can only be compared if the experimental protocols are the same or equivalent. As stated in section 2, the process of selection combined with classification must be evaluated by cross-validation and the selection of a subset of genes must be realized at each fold of the cross-validation. The results obtained by such a protocol are the mean and standard deviation of the classification accuracies and the mean and standard deviation of the number of selected genes.

The results are presented in Table 3 for the 8 datasets. Each cell here has the same meaning as that of Table 2. We must notice that two different datasets concerning Lymphoma are studied in the literature. We give in the table the results concerning the dataset that contains 47 samples (24 GC B-like samples and 23 activated B-like samples) [2]. Studies dealing with the other Lymphoma dataset is not compared here.

First, let us compare MAGS and the method of [28] which shares 6 out of the 8 datasets. It is also one of the scarce studies found in the literature using a hybrid genetic algorithm for gene selection. Their GA is a classical wrapper method where individuals are represented by binary strings. Standard random crossover and mutation operators are used to evolve the population and a SVM classifier is trained to calculate the fitness of each individual. In each generation, the best individual undergoes a local search improvement. The local search operator ranks the unselected genes according to a correlation measure and adds the most relevant one to the subset. Similarly the correlation measure identifies the most relevant gene already selected, computes an approximate Markov blanket for this gene and eliminates all the genes which are in this Markov blanket. The local search is therefore a kind of filter method and the GA a wrapper method that uses the SVM as a black box, whereas in our method the interaction with the classifier is a central element.

From Table 3, one observes that MAGS achieves better classification accuracies than the method of [28] for all the 6 datasets. This observation is confirmed by a wilcoxon test with a p-value of 0.05 ($R^- = 0$)[1]. This tends to demonstrate that the interaction between the classifier and search operators boosts the performance of the MAGS method.

Moreover, to achieve these results, MAGS selects smaller gene subsets except for one dataset (breast). This is probably not very surprising because the fitness function of [28] concerns the sole objective of classification accuracy. It seems that removing genes by the Markov blanket process does not suffice to obtain small gene subsets.

Now concerning the results of the three other methods presented in Table 3 (columns 4-6), one observes that MAGS competes always very favorably in terms of classification accuracy and the number of selected genes. The number of common experiments with these three cases are not sufficient to allow a statistical validation.

Finally let us mention that a perfect classification estimate for the Colon data is reported in [1]. However, given that their selection process is realized before cross-validation, the results may suffer from the selection bias and consequently can be optimistic with overestimated accuracies. As explained in Section 2.3, it is impossible to have a fair comparison of the results of [1] with those presented in Table 3.

5. CONCLUSION

In this paper, we have presented MAGS, a memetic algorithm for gene selection and classification of microarray data. This algorithm combines both specialized genetic search and local search to establish a good balance between exploration and exploitation of the search space. MAGS is based on

[1] We must point out that accuracy in [28] is evaluated by the .632 bootstrap estimator whereas 10-fold cross-validation is used in our case. It would be better to have results with the same estimator to perform a rigorous comparison of the two methods.

Table 3: Comparaison with other recent works

Data		References			
	MAGS	[28]	[23]	[26]	[19]
Colon	98.33±05.27 7.70±1.95	85.66±5.46 24.5±7.0	83.81±10.26 23.40±5.03	83.87 -	88.70±01.60 16.83±1.15
Leuk.	100 6.2±3.19	95.89±2.46 12.8±4.9	- -	100 -	95.08±1.27 20.76±1.49
Breast	95.78±5.46 16.10±4.93	80.74±3.45 14.5±4.2	- -	95.88 -	- -
Lung	100 7.30±3.02	98.96±0.88 14.10±7.00	- -	99.45 -	- -
Prost.	97.03±3.83 25.20±7.27	- -	97.06 -	- -	- -
Ovar.	100.00 3.00±0.00	99.71±0.53 9.0±2.06	98.80±1.10 25.60±5.90	- -	- -
CNS	95.00±8.05 20.70±4.54	72.21±5.91 20.50±6.9	65.00±16.02 46.20±5.50	95.00 -	- -
Lymp.[47]	100.00 5.7±2.16	- -	- -	- -	- -

the embedded approach where a SVM classifier is used not only to evaluate gene subsets but also to provide valuable information about gene relevancy. MAGS takes advantage of this information to design specialized crossover and local search operators.

The performance of MAGS is assessed on 8 well-known datasets. We show that removing from MAGS its genetic part (crossover) or its local search procedure inevitably weakens its performance. This demonstrates the importance of the memetic approach which combines tightly the genetic search and local search with a single search process. More importantly, computational comparisons with some most recent methods for gene selection show clearly that MAGS is able to produce higher classification accuracy with a small number of genes.

6. ACKNOWLEDGMENTS

We acknowledge that the work is partially supported by the French Ouest Genopole® and the Region "Pays de La Loire" via the BIL Project. We would like to thank the reviewers of the paper for their helpful comments.

7. REFERENCES

[1] E. Alba, J. Garcia-Nieto, L. Jourdan, and E.G. Talbi. Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms. In *Proceedings of the IEEE CEC'07*, pages 284–290, 2007.

[2] A. Alizadeh, M.B. Eisen, R.E. Davis, C.Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J.J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, and L.M. Staudt. Distinct types of diffuse large (b)–cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, February 2000.

[3] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Nat. Acad. Sci. USA.*, 96:6745–6750, 1999.

[4] C. Ambroise and G. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Nat. Acad. Sci. USA*, 99(10):6562–6566, 2002.

[5] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *COLT*, pages 144–152, 1992.

[6] U. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[7] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97:77–87, 2002.

[8] T. Furey, N. Cristianini, N. Duffy, D. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[9] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

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

[11] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[12] E. Bonilla Huerta, B. Duval, and J.K. Hao. A hybrid GA/SVM approach for gene selection and classification of microarray data. In *Lecture Notes in Computer Science*, 3907: 34–44. Springer, 2006.

[13] T. Jirapech-Umpai and J. Stuart Aitken. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6:148, 2005.

[14] J.C. Hernandez Hernandez, B. Duval and J.K. Hao. A genetic embedded approach for gene selection and classification of microarray data. *Lecture Notes in Computer Science*, 4447: 90–101, Springer, 2007.

[15] J.C. Hernandez Hernandez, B. Duval and J.K. Hao. SVM-based local search for gene selection and classification of Microarray data. *Communications in Computer and Information Science*, 13: 599–598, Springer, 2008.

[16] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 1137–1143, Morgan Kauffman, 1995.

[17] L. Li, C. Weinberg, T. Darden, and L. Pedersen. Gene selection for sample classification based on gene

expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, 17(12):1131–1142, 2001.

[18] S. Li, X. Wu, and X. Hu. Gene selection using genetic algorithm and support vectors machines. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 12(7):693–698, 2008.

[19] S. Li, X. Wu, and M. Tan. Gene selection using hybrid particle swarm optimization and genetic algorithm. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 12(11):1039–1048, 2008.

[20] B. Liu, Q. Cui, T. Jiang, and S. Ma. A combinational feature selection and ensemble neural network method for classification of gene expression data. *BMC Bioinformatics*, 5(138):1–12, 2004.

[21] H. R. Lourenco, O. Martin and T. Stutzle. Iterated local search. *Handbook of Metaheuristics*. F. Glover and G. Kochenberger (Eds.), Springer-Verlag, 321-353, 2003.

[22] E. Marchiori and M. Sebag. Bayesian learning with local support vector machines for cancer classification with gene expression data. *Lecture Notes in Computer Science*, 3449: 74–83, Springer, 2005.

[23] S. Pang, I. Havukkala, Y. Hu, and N. Kasabov. Classification consistency analysis for bootstrapping gene selection. *Neural Computing and Applications*, 16:527,539, 2007.

[24] S. Peng, Q. Xu, X.B. Ling, X. Peng, W. Du, and L.Chen. Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. *FEBS Letters*, 555(2):358–362, 2003.

[25] A. Rakotomamonjy. Variable selection using SVM-based criteria. *Machine Learning Research*, 3:1357–1370, 2003.

[26] C.W. Wang. New ensemble machine learning method for classification and prediction on gene expression data. In *IEEE EMBS Annual International Conference. Institute of Electrical and Electronics Engineers*, pages 3478–3481, 2006.

[27] W. Xiong, C. Zhang, C. Zhou, and Y. Liang. Selection for feature gene subset in microarray expression profiles based on a hybrid algorithm using svm and ga. *ISPA 2006 - LNCS*, 4331:637–647, 2006.

[28] Z. Zhu, Y.S. Ong, and M. Dash. Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, 40:3236–3248, 2007.