Computers in Biology and Medicine 41 (2011) 228–237

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

A hybrid feature selection method for DNA microarray data

Li-Yeh Chuang [a], Cheng-Huei Yang [b], Kuo-Chuan Wu [c], Cheng-Hong Yang [d,e,*]

[a] Department of Chemical Engineering, I-Shou University, Kaohsiung 80041, Taiwan
[b] Department of Electronic Communication Engineering, National Kaohsiung Marine University, Kaohsiung 81157, Taiwan
[c] Department of Computer Science and Information Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 80708, Taiwan
[d] Department of Network Systems, Toko University, Chiayi 61363, Taiwan
[e] Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 80708, Taiwan

ARTICLE INFO

Article history:
Received 21 August 2010
Accepted 8 February 2011

Keywords:
Feature selection
Taguchi-genetic algorithm
K-nearest neighbor
Leave-one-out cross-validation

ABSTRACT

Gene expression profiles, which represent the state of a cell at a molecular level, have great potential as a medical diagnosis tool. In cancer classification, available training data sets are generally of a fairly small sample size compared to the number of genes involved. Along with training data limitations, this constitutes a challenge to certain classification methods. Feature (gene) selection can be used to successfully extract those genes that directly influence classification accuracy and to eliminate genes which have no influence on it. This significantly improves calculation performance and classification accuracy. In this paper, correlation-based feature selection (CFS) and the Taguchi-genetic algorithm (TGA) method were combined into a hybrid method, and the K-nearest neighbor (KNN) with the leave-one-out cross-validation (LOOCV) method served as a classifier for eleven classification profiles to calculate the classification accuracy. Experimental results show that the proposed method reduced redundant features effectively and achieved superior classification accuracy. The classification accuracy obtained by the proposed method was higher in ten out of the eleven gene expression data set test problems when compared to other classification methods from the literature.

1. Introduction

Microarray data can provide valuable results for a variety of gene expression profile problems and contribute to advances in clinical medicine. The application of microarray data on cancer type classification has recently gained in popularity. Coupled with statistical techniques, gene expression patterns have been used to screen potential tumor markers. Differential expressions of genes are analyzed statistically and each gene expression is assigned to a certain category. The classification of gene expressions can substantially enhance the understanding of the underlying biological processes.

The goal of microarray data classification is to build an efficient and effective model that can differentiate the gene expressions of samples, i.e., determine normal or abnormal states, or classify tissue samples into different classes of diseases. The challenges posed in microarray classification are the limited amount of samples in comparison to the high-dimensionality of the sample, along with experimental variations in measured gene expression levels. In general, only a relatively small number of gene expression data show a strong correlation with a certain phenotype compared to the total number of genes investigated. This means that of the thousands of genes investigated only a small number show significant correlation with the phenotype in question. Thus, in order to analyze gene expression profiles correctly, feature (gene) selection is crucial for the classification process.

Recently, many gene expression data classification and gene selection techniques have been introduced. Kim et al. [1] proposed a novel method based on an evolutionary algorithm (EA) to assemble optimal classifiers and improve feature selection. Tang et al. [2] used an approach that selects multiple highly informative gene subsets. Wang et al. [3] proposed a new tumor classification approach based on an ensemble of probabilistic neural networks (PNN) and neighborhood rough set models based on gene selection. Shen et al. [4] proposed a modified particle swarm optimization that allows for the simultaneous selection of genes and samples. Xie et al. [5] developed a diagnosis model based on support vector machines (SVM) with a novel hybrid feature selection method for diagnosis of erythemato-squamous diseases. Li et al. [6] proposed an algorithm with a locally linear discriminant embedded in it to map the microarray data to a low dimensional space, while Huang et al. [7] proposed an improved decision forest for the classification of gene expression data that incorporates a built-in feature selection mechanism for fine-tuning.

* Corresponding author at: Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan.
Tel.: + 886 7 381 4526x5639; fax: +886 7 383 6844.
E-mail addresses: chuang@isu.edu.tw (L.-Y. Chuang), kuo.chuan.wu@gmail.com (K.-C. Wu), chyang@cc.kuas.edu.tw (C.-H. Yang).

0010-4825/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2011.02.004

In summary, the above feature selection methods can be divided into three common models: filter methods, wrapper methods, and embedded methods. The filter approach separates data before the actual classification process takes place and then calculates feature weight values, and thus features that accurately present the original data set can be identified. However, a filter approach does not account for interactions amongst the features. Methods in the filter approach category include correlation-based feature selection (CFS) [9], *t*-test, information gain [10], mutual information [11], and entropy-based methods [12]. Wrapper models, on the other hand, generally focus on improving classification accuracy of pattern classification problems and typically perform better (i.e., reach higher classification accuracy) than filter models. However, wrapper approaches are more computationally expensive than filter methods [13,14]. Several methods in this category have previously been used to perform feature selection of training and testing data, such as genetic algorithm (GA) [15], branch and bound algorithm [16], sequential search algorithm [17], tabu search [18,19], binary particle swarm optimization [20,21], and hybrid genetic algorithm [22]. Embedded techniques use an inductive algorithm. The inductive algorithm itself represents the feature selector and the classifier. Embedded techniques search for an optimal subset of features that is built into the classifier. Examples of these classification trees are ID3, C4.5 and random forest. The advantage of embedded algorithms is that they take the interaction with the classifier into account. A disadvantage of embedded algorithms is that they are generally based on a greedy mechanism, i.e., they only use top-ranked attributes to perform sample classification [8,23].

Many feature selection methods are combined with a local search process to improve accuracy. One example is presented in Oh et al. [22] who used a local search mechanism in their genetic algorithm. In this paper, we used the Taguchi method as a local search method embedded in the GA. The Taguchi method uses ideas from statistical experimental design to improve and optimize products, processes or equipment. The two main tools of the Taguchi method are: (a) the signal-to-noise ratio (SNR), which measures quality and (b) orthogonal arrays (OAs), which are used to simultaneously study the many design parameters involved. The Taguchi method is a robust design approach [24]. It has been successfully applied in machine learning and data mining, e.g., combined data mining and electrical discharge machining [20]. Sohn and Shin used the Taguchi experimental design for the Monte Carlo simulation of classifier combination methods [25]. Kwak and Choi used the Taguchi method to select features for classification problems [26]. Chen et al. optimized neural network parameters with the Taguchi method [27].

A hybrid feature selection approach consisting of two stages is presented in this study. In the first stage, a filter approach is used to calculate correlation-based feature weights for each feature, thus identifying relevant features. In the second stage, which constitutes a wrapper approach, the previously identified relevant feature subsets are tested by a Taguchi-genetic algorithm (TGA), which tries to determine optimal feature subsets. These optimal feature subsets are then appraised with the K-nearest neighbor method (KNN) [28,29] with leave-one-out cross-validation (LOOCV) [30,31] based on Euclidean distance calculations. Genetic algorithms [32,33] are utilized with randomness for a global search over the entire search space. The genetic operations crossover and mutation are performed to assist the search procedure in escaping from sub-optimal solutions [14]. In each iteration of the proposed nature-inspired method, the Taguchi method [24,34,35] is implemented to help explore better feature subsets (or solutions), which are somewhat different from those in the candidate feature subsets. In other words, the Taguchi algorithm is employed for a local search in the search space. Experimental results show that the proposed method achieved higher classification accuracy rates and outperformed the other methods from the literature it was compared to.

2. Material and methods

2.1. Correlation-based feature selection (CFS)

CFS was developed by Hall in 1999 [9]. CFS is a simple filter feature selection method that ranks feature subsets based on a correlation-based heuristic evaluation. This feature selection method is based on the following hypothesis:

> *Good feature subsets contain features highly correlated with* (*i.e., predictive of*) *the class, yet uncorrelated with* (*i.e., not predictive of*) *each other* [9].

This hypothesis is incorporated into the correlation-based heuristic evaluation equation as

$$Merit_S = \frac{k\overline{\gamma}_{cf}}{\sqrt{k+k(k-1)\overline{\gamma}_{ff}}} \quad (1)$$

where $Merit_S$ is the merit of a feature subset S containing k features, $\overline{\gamma}_{cf}$ is the average feature and class correlation, and $\overline{\gamma}_{ff}$ is the average feature-feature intercorrelation ($f \in S$).

General filter methods estimate the significance of a feature individually. CFS is then used to select the best combination of attribute subsets via score values from the original data sets. Heuristic search strategies are employed to identify the best combination. Common strategies include forward selection, backward elimination, and the best-first method. In this study, we used Weka [36] to implement CFS, and used the selected gene subsets to identify different cancer types and various diseases.

2.2. Genetic algorithm

A genetic algorithm (GA) was first developed by Holland in 1970. A GA is a stochastic search algorithm modeled on the process of natural selection underlying biological evolution. GAs has been successfully applied to many search, optimization, and machine learning problems [37]. They represent an intelligent exploitation of a random search within a defined search space to solve a problem. A GA proceeds in an iterative manner by generating new populations of strings from old ones. Every string is the encoded binary, real, etc., version of a candidate solution. An evaluation function connects a fitness measure to every string, indicating its fitness for the problem. Standard GAs apply genetic operators such as selection, crossover, and mutation on an initially random population in order to compute an entire generation of new strings. Further details of GA mechanisms can be found in Holland [37].

For a feature subset selection problem, a possible solution in the solution space is a specific feature subset that can be encoded as a string of n binary digits (or bits). Each feature is represented by binary digits with values 1 or 0, which identify whether the feature is selected or not selected in the corresponding feature subset, respectively. This process is called solution (or chromosome) encoding. For instance, in the 0100100010 string of ten binary digits (i.e., a solution or a chromosome) the features 2, 5, and 9 are selected in the corresponding feature subset.

In the first step of a general GA some solutions are randomly selected from the solution space as the initial set *CS* of candidate solutions. The number of solutions in *CS* is denoted the population size. When two parent solutions p_1 and p_2 are selected from *CS*, the crossover operation is applied to generate a corresponding offspring q. In other words, each feature i of offspring q is the

same as either that of p_1 or that of p_2. Consequently, the mutation operation is utilized to slightly perturb offspring q. Once a perturbed offspring is obtained, an evaluation criterion is applied to analyze the fitness of the parent solutions and the corresponding offspring. For each iteration (generation), solutions that have a high fitness value are retained in the candidate set *CS* (i.e., *CS* is updated). The genetic operations crossover, mutation, fitness evaluation, selection, and replacement are repeated until a predefined number of iterations or a particular termination condition is met. Finally, a set of eight optimal or sub-optimal solutions for a specific problem domain are obtained. In the proposed Taguchi optimization GA-based feature selection method, the GA is utilized with randomness for a global search over the entire search space (i.e., 2^n possible feature subsets). The genetic operations are performed to assist the search procedure in escaping from sub-optimal solutions [14].

2.3. Taguchi method

The Taguchi method was developed by Genichi Taguchi. It is a statistical method of robust design. In a robust experimental design [24,34,35], processes or products can be analyzed and improved by altering relevant design factors. The commonly used Taguchi method [24,34,35] provides two tools, an orthogonal array (OA) and a signal-to-noise ratio (SNR), for analysis and improvement.

Two-level orthogonal arrays are used in this paper. Details of other level OAs can be found in the literature [24,34,35]. An OA can be considered a fractional factorial experimental design matrix that provides a comprehensive analysis of interactions among a balanced set of experimentation runs and systematic comparisons of the different levels of each design factor. The OA is a matrix arranged in rows and columns, with each column indicating a specific design parameter and each row representing an experimental trial with a particular combination of different levels for all design factors. In general, a two-level OA is represented as L_n (2^{n-1}), where $n=2^k$ is the number of experimental trials; k is an integer which is bigger than 1; k is decided by a number of factors. The symbol 2 denotes the number of levels for each factor and $n-1$ is the number of columns in the orthogonal array. Table 1 shows an $L_8(2^7)$ orthogonal array. The number of experimental trials 1 to 8 means that those experimental trials consider A to G factors. The table is generated according to some other rules as well, details of which can be found in the literature [24,34,35].

The SNR is utilized to analyze and optimize the design parameters for a particular target. The larger-the-better SNR type [35] is typically used. Consider that a set of t observations $\{y_1, y_2, \ldots, y_t\}$ is collected. For the larger-the-better characteristic, the SNR is determined by

$$SNR = -10\log\left(\frac{1}{n}\sum_{t=1}^{n}\frac{1}{y_t^2}\right) \quad (2)$$

The SNR is utilized in the Taguchi method to determine the robustness of the levels of each design parameter. Good results for a particular target can be achieved by specifying design parameters at a level with a high SNR.

Table 1
$L_8(2^7)$ orthogonal array.

Number of experimental trial	Design factors						
	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

2.4. K-nearest neighbor method

The KNN method is one of the most popular nonparametric methods [28,29] used for classification of new objects based on attributes and training samples. KNN consists of a supervised learning algorithm, which instantly classifies the results of a query instance based on the majority of the KNN category. Classifiers do not use any model for KNN and are determined solely based on the minimum distance from the query instance to the training samples. Any tied results are solved by a random procedure.

Given the training data $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_n, y_n)\}$ and the test data $\boldsymbol{x}$, where $\boldsymbol{x}$ is the feature vector of the data, y_i is the class of data $\boldsymbol{x}_i$, and n is number of data, the distance measure can be defined as $D(x,x_i)=\sqrt{\sum_{j=1}^{d}(x_j-x_{ij})^2}$, where d is the dimension of the feature vector. The nearest neighbor rule is $\mathrm{nnr}(\boldsymbol{x})=y_k$, where $k=\arg\min_i D(\boldsymbol{x}, \boldsymbol{x}_i)$. A voting strategy is used if $K>1$. For example if $K=3$, three minimal distance measures are calculated; if two points fall into class A and one point falls into class B, class A is chosen. The pseudo-code is shown below.

```
Begin
  For i=1 to number of test set
    For j=1 to number of training set
      Calculate distance of test sample to training set
    Next j
  Next i
    For k=1 to number of parameter K
      Determine class of test set by voting strategy
  Next k
    Determine the classification accuracy
End
```

Major advantages of the KNN method are its simplicity and ease of implementation. KNN is not negatively affected by large training data, and is furthermore indifferent to noisy training data [28]. In this study, the feature subset was measured by the LOOCV of one nearest neighbor (1NN). Neighbors are calculated using their Euclidean distance. The 1NN classifier does not require any user-specified parameters, and the classification results are implementation independent. In the LOOCV method, a single observation from the original sample is selected as the validation data, and the remaining observations are used as the training data. This process is repeated so that each observation in the sample is used once as the validation data. Essentially, the procedure is the same as m-fold cross-validation, where m is equal to the number of observations in the original data set.

2.5. CFS–TGA procedure

This paper introduces a hybrid CFS method to implement a gene ranking process, and combines it with a TGA. The KNN with the LOOCV method serves as a classifier to calculate the classification accuracy. The flowchart of CFS–TGA in Fig. 1 details the execution of the individual steps.

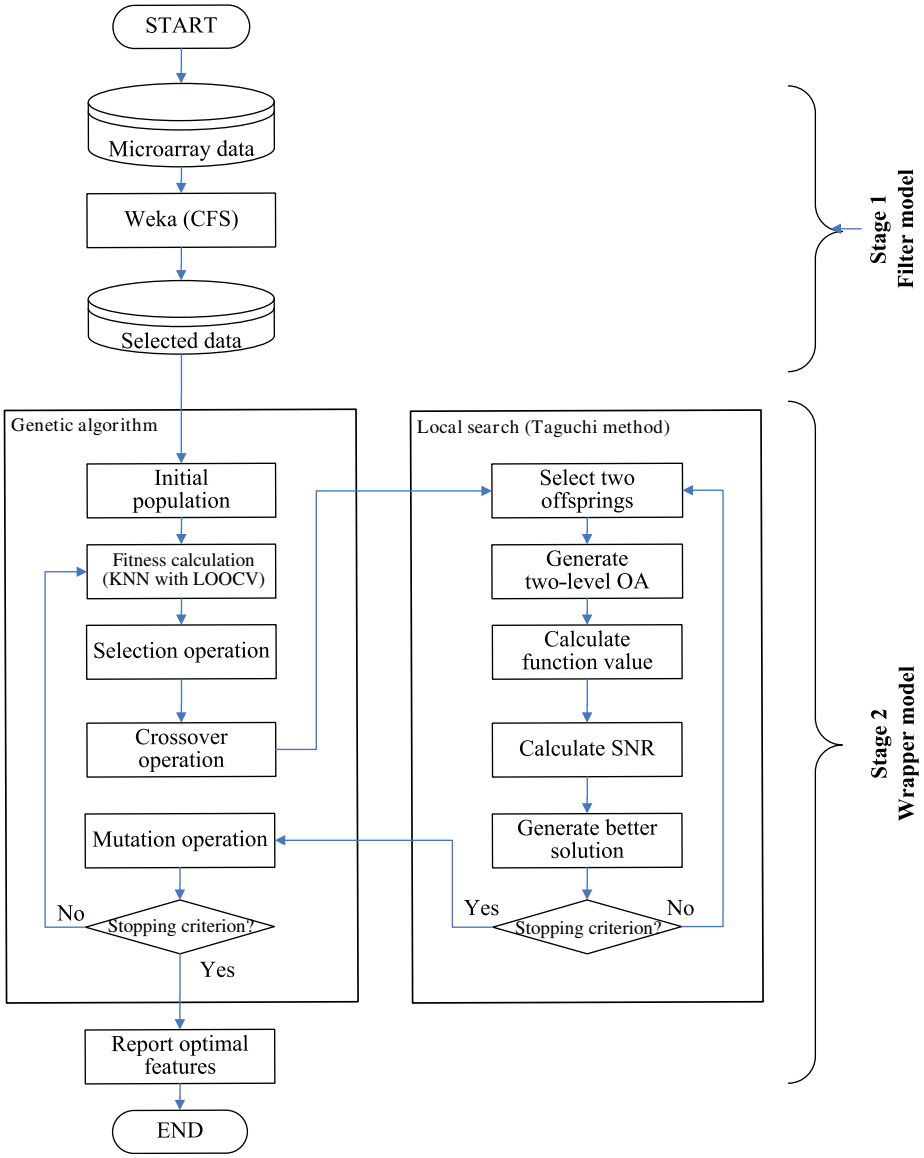

Fig. 1. CFS–TGA flowchart.

Step 1) A feature selection set is generated by CFS in Weka [36]. Weka calculates the CFS merit of each correlation-based feature, and then the subset with the highest merit is selected. This specific subset is used in the next selection stage (i.e., TGA, steps 2–13).

Step 2) A set (sub-population) CS of candidate feature subsets ($CS \subseteq S$) is initially used. For each feature set CS_j, the average classification accuracy for training set T (denoted ACC(T, CS_j)) is determined using the KNN classification rule with the LOOCV technique.

Step 3) $ps \times cr$ parent solutions are randomly selected from CS with the roulette wheel selection method. Consequently, $ps \times cr/2$ pairs of parent solutions, denoted VS, are obtained.

Step 4) $ps \times cr$ offsprings are generated by $ps \times cr/2$ crossover operations performed on the corresponding $ps \times cr/2$ pairs of the parent solutions in VS. A set of $ps \times cr$ offspring solutions, denoted BS_1, is obtained.

Step 5) Two solutions, denoted b_1 and b_2, are randomly selected from BS_1. Consider that b_1 and b_2 have w different bits ($w \geq n$, where n is number of data).

Step 6) An extended two-level OA with respect to the above particular w bits (i.e., features or factors) of b_1 and b_2 is generated. The level of feature i in the OA is replaced by the corresponding bit of b_1 if the original level is 1. Conversely, the level of feature i in the OA is replaced by the corresponding bit of b_2 if the original level is 2. Notably, the factors of the remaining $(d-w)$ bits in the two-level OA are the same as the corresponding bits of b_1 and b_2, where d is the number of features. In each experimental trial j, 1 or 0 in each column i of the extended two-level OA indicate whether feature i is selected or not selected for pattern classification in the corresponding feature set CS_j.

Step 7) ACC(T, CS_j) is considered an observation or objective function of experimental trial j in the extended two-level OA. This process, called fitness evaluation, is used to measure the viability of each feature set or solution CS_j.

Step 8) The corresponding SNR for each level (i.e., levels 1 and 2) of the particular w bits is calculated according to observations from all experimental trials in the extended two-level OA.

Step 9) A better solution *t_best* is generated based on the results in the extended two-level OA. For all w bits in *t_best*, each bit is determined by value 1 if the corresponding SNR for level 1 is greater than that for level 0, and vice versa. Notably, the remaining $(d-w)$ bits of *t_best* are the same as those of b_1 and b_2.

Step 10) Repeat Steps 5–9 until a set of $\lfloor ps*pc/4 \rfloor$ better solutions (i.e., $\lfloor ps*pc/4 \rfloor$ different *t_bests*), denoted BS_2, is obtained.

Step 11) Each offspring solution in BS_2 is perturbed by a mutation operation. The mutation probability is determined by *mr*. Similarly, a set of $ps \times cr/4$ perturbed offspring solutions, denoted BS_3, is obtained.

Step 12) *CS* is updated by using the superior candidate feature subsets (solutions) in BS_3.

Step 13) Steps 3–12 are repeated until a certain number of iterations have been completed. Consequently, the best feature subset, denoted *g_best* in *CS*, is utilized as the final feature subset for pattern classification.

Table 2
Genes of chromosomes b_1 and b_2.

Factors	–	A	B	C	D	E	F	G	–	Accuracy (%)
Level 1 (chromosome b_1)	0	0	0	0	0	1	1	1	0	60.28
Level 2 (chromosome b_2)	0	1	1	1	1	0	0	0	0	60.75

Table 3
Extended $L_8(2^7)$ orthogonal array.

Number of experimental trial	Design factors								
	–	A	B	C	D	E	F	G	–
	Column number								
	–	1	2	3	4	5	6	7	–
1	0	0	0	0	0	1	1	1	0
2	0	0	0	0	1	0	0	0	0
3	0	0	1	1	0	1	0	0	0
4	0	0	1	1	1	0	1	1	0
5	0	1	0	1	0	0	1	0	0
6	0	1	0	1	1	1	0	1	0
7	0	1	1	0	0	0	0	1	0
8	0	1	1	0	1	1	1	0	0

2.6. Illustrative example

This section provides an example that illustrates the details, in particular the steps regarding the Taguchi method (Steps 5–9) of the proposed CFS–TGA feature selection method. In the UCI glass data set pattern classification problem [38] with 214 samples, each sample x_e has a set of 9 attributes. Each specific feature subset is encoded as a string of nine binary digits (or bits). Each feature can be described by a binary digit with the value 1 or 0, which indicates whether the feature is selected or not selected in the corresponding feature subset.

Two candidate feature subsets, b_1 and b_2 (011100000 and 000011110, respectively) are randomly selected from the population in Step 5 (as shown in Table 2). The candidate accuracies are $b_1=60.28$ and $b_2=60.75$. These two candidate feature subsets are comprised of seven different bits, i.e., features A, B, C, D, E, F, and G. Therefore, we selected an $L_8(2^7)$ orthogonal array (Table 1).

In Step 6, the level of feature i in the *OA* is replaced by the corresponding bit of b_1 if the original level is 1. Conversely, the level of feature i in the *OA* is replaced by the corresponding bit of b_2 if the original level is 2. For example, the second experimental trial in Table 1 shows the string 1112222; in it each factor is given by $b_1b_1b_1b_2b_2b_2b_2$. According to Table 2, the second experimental trial it yields the string 0001000, as shown in Table 3. Consequently, a new, extended two-level *OA* (Table 3) with respect to the above *seven* bits of b_1 and b_2 is obtained. The levels of the remaining *two* features in the two-level *OA* are the same as the corresponding bits of b_1 and b_2.

In each experimental trial j in the new, extended two-level *OA*, 1 or 0 in each column i indicate whether feature i is selected or not selected in the corresponding feature set CS_j. For each feature set CS_j, $ACC(T, CS_j)$ can be determined using the KNN classification rule with the LOOCV technique. $ACC(T, CS_j)$ is considered an observation or objective function of experimental trial j in the new, extended two-level *OA*. For example, the average classification accuracy of feature subset {E, F, G} (i.e., experimental trial 1 in Table 4) is 54.21%. This process, the fitness evaluation, is used to measure the quality of each feature set or solution CS_j. The experimental layout and signal-to-noise data of the Glass pattern classification problem is summarized in Table 4. The larger-the-better characteristic (Eq. (2)) is selected to calculate the SNR, as maximum classification accuracy is preferred in pattern classification. Next, as shown in Table 4, the corresponding SNR for each level of the particular *seven* features can be calculated according to observations from all experimental trials in the new, extended

Table 4
Generation of better genes from two chromosomes using the Taguchi method.

Number of experimental trial (j)	Factors (Features)									Function value
	–	A	B	C	D	E	F	G	–	
	Column number (i)									
	–	1	2	3	4	5	6	7	–	
1	0	0	0	0	0	1	1	1	0	54.21
2	0	0	0	0	1	0	0	0	0	28.97
3	0	0	1	1	0	1	0	0	0	63.08
4	0	0	1	1	1	0	1	1	0	67.66
5	0	1	0	1	0	0	1	0	0	64.95
6	0	1	0	1	1	1	0	1	0	67.76
7	0	1	1	0	0	0	0	1	0	52.80
8	0	1	1	0	1	1	1	0	0	69.16
E_{F1}	–	33.01	33.04	32.80	35.27	35.94	36.00	35.47	–	
E_{F2}	–	35.92	35.86	36.36	33.38	33.00	32.97	33.26	–	
Optimal level	–	b_2	b_2	b_2	b_1	b_1	b_1	b_1	–	
Optimal position	0	1	1	1	0	1	1	1	0	69.63

two-level *OA*. Here we take the factors A as an example

$$E_{A1} = -10\log\left(\frac{1}{4}\sum_{j=1,2,3,4}\frac{1}{FV_j^2}\right) = = -10\log\left[\frac{1}{4}\left(\frac{1}{54.21^2} + \frac{1}{28.97^2} + \frac{1}{63.08^2} + \frac{1}{67.66^2}\right)\right] = 33.01$$

$$E_{A2} = -10\log\left(\frac{1}{4}\sum_{j=5,6,7,8}\frac{1}{FV_j^2}\right) = = -10\log\left[\frac{1}{4}\left(\frac{1}{64.95^2} + \frac{1}{67.76^2} + \frac{1}{52.8^2} + \frac{1}{69.16^2}\right)\right] = 35.92$$

where *FV* is the function value and *j* is assigned the level value of Table 1. The optimal level of each factor is decided by the bigger value of either E_{F1} or E_{F2}, where $F=A$. In the example, the optimal level is level 2 (b_2) because E_{A2} is larger than E_{A1}. As a result, a better solution *t_best*, encoded 011101110, can be obtained based on Table 4. For all *seven* bits in *t_best*, each bit is determined by value 1 if its corresponding SNR for level 1 is greater than that for level 0, and vice versa. The average classification accuracy of the better solution *t_best* is 69.63%, a value that is significantly higher than that of the feature subset in each experimental trial in the new, extended two-level *OA*.

3. Results

3.1. Data description

Due to the large number of genes and the small sample size of gene expression data, many researchers are currently studying how to select genes effectively before using a classification method to decrease the predictive error rate. In general, gene selection is based on two aspects: one is to obtain a set of genes that have similar functions and a close relationship, the other is to find the smallest set of genes that can provide meaningful diagnostic information for disease prediction without diminishing accuracy. In feature selection, relatively few features are used since only selective features need to be considered. This does not affect the predictive error rate in a negative way; on the contrary, the predictive error rate can even be improved.

The experimental data sets of this study were downloaded from http://www.gems-system.org [39]. They consist of the 9_Tumors, 11_Tumors, 14_Tumors, Brain_Tumor1, Brain_Tumor2, DLBCL, Leukemia1, Leukemia2, Lung_Cancer, SRBCT, and Prostate_Tumor data sets. The data set formats are summarized in Table 5 and include the data set name, the number of samples, the number of classes, the number of genes, and the diagnostic task. Since feature value scaling can enhance pattern recognition accuracy, the values were normalized to [0,1]. The normalization is given by formula (3), where f'_{value} is a scaled value of a feature, f_{value} is the original value of a feature, $value_{MAX}$ is the upper boundary of the feature value, and $value_{MIN}$ is the lower boundary of the feature value

$$f'_{value} = \frac{f_{value} - value_{MIN}}{value_{MAX} - value_{MIN}} \quad (3)$$

3.2. Parameter settings

Many researchers base their choices for tuning the control parameters on experiments conducted with different values and select the ones that give the best results [40]. Arabas et al. [41] mentioned that if the population size is too small, the genetic algorithm may converge too quickly; if it is too large, the genetic algorithm may waste computational resources and the waiting time for an improvement might be too long. Four parameters have to be set for the GA, namely the number of iterations (generations), the population size, the probability of crossover, and the probability of mutation. In this paper, the respective values are 30, 60, 1.0, and 0.01. These parameters were suggested by Elbeltagi et al. [42].

In the KNN parameter *K*, the best choice of the number of neighbors depends upon the data. Generally, larger values of *K* reduce the effect of noise on the classification, but boundaries between classes become less distinct. Ghosh [43] indicated that the optimum value *K* depends on the specific data set and is to be estimated using the available training sample observations. On the other hand, since the time complexity of KNN is $O(kn \log n)$, the parameter *K* directly influences the performance. For the purposes of this study we used $K=1$.

3.3. Experimental results

Table 6 compares experimental results obtained by other methods in the literature and the proposed method. Non-SVM and MC-SVM results were taken from Statnikov et al. [39] for comparison. The non-SVM methods include: backpropagation neural networks (NN), probabilistic neural networks (PNN), and the KNN method. The SVM methods compared to the proposed method include: support vector machines: (1) one-versus-one (OVO) and one-versus-rest (OVR), (2) DAG, (3) Weston and Watkins (WW), and (4) Crammer and Singer (CS). The highest average classification accuracy of non-SVM, MC-SVM, and the proposed method (CFS–TGA) is 77.16%, 89.44%, and 96.54%, respectively. Tests for each data set were executed ten times. The accuracies are given in the form "average ± standard deviation" in Table 6. Generally speaking, it is difficult to design a

Table 5
Format of gene expression classification data.

Data set name	Number of			Diagnostic task
	Samples	Classes	Genes	
9_Tumors	60	26	5726	9 various human tumor types
11_Tumors	174	11	12,533	11 various human tumor types
14_Tumors	308	9	15,009	14 various human tumor types and 12 normal tissue types
Brain_Tumor1	90	5	5920	5 human brain tumor types
Brain_Tumor2	50	4	10,367	4 malignant glioma types
DLBCL	77	2	5469	Diffuse large b-cell lymphomas (DLBCL) and follicular lymphomas
Leukemia1	72	3	5327	Acute myelogenous leukemia (AML), acute lympboblastic leukemia (ALL) B-cell, and ALL T-cell
Leukemia2	72	3	11,225	AML, ALL, and mixed-lineage leukemia (MLL)
Lung_Cancer	203	5	12,600	4 lung cancer types and normal tissues
SRBCT	83	4	2308	Small, round blue cell tumors (SRBCT) of childhood
Prostate_Tumor	102	2	10,509	Prostate tumor and normal tissues

Table 6
Classification accuracy for gene expression data.

Data set	Non-SVM			MC-SVM					CFS	IBPSO	CFS–TGA
	NN	PNN	KNN	OVO	DAG	WW	CS	OVR	KNN	KNN	KNN
9_Tumors	19.38	34.00	43.90	58.57	60.24	62.24	65.33	65.10	70.00	78.33	90.50 ± 0.81 (**91.67**)
11_Tumors	54.14	77.24	78.51	90.36	90.36	94.68	95.30	94.68	90.23	93.10	100.0 ± 0.00 (**100.0**)
14_Tumors	11.12	49.09	50.40	47.07	47.35	69.07	**76.60**	74.98	64.61	66.56	74.39 ± 0.67 (75.65)
Brain_Tumor1	84.72	79.61	87.94	90.56	90.56	90.56	90.56	91.67	92.22	94.44	99.45 ± 0.59 (**100.0**)
Brain_Tumor2	60.33	62.83	68.67	77.83	77.83	73.33	72.83	77.00	90.00	94.00	100.0 ± 0.00 (**100.0**)
Leukemia1	76.61	85.00	83.57	97.32	96.07	97.50	97.50	97.50	98.61	**100.0**	100.0 ± 0.00 (**100.0**)
Leukemia2	91.03	83.21	87.14	95.89	95.89	95.89	95.89	97.32	**100.0**	**100.0**	100.0 ± 0.00 (**100.0**)
Lung_Cancer	87.80	85.66	89.64	95.59	95.59	95.55	96.55	96.05	95.07	96.55	98.42 ± 0.31 (**99.01**)
SRBCT	91.03	79.50	86.90	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	100.0 ± 0.00 (**100.0**)
Prostate_Tumor	79.18	79.18	85.09	92.00	92.00	92.00	92.00	92.00	94.12	92.16	99.22 ± 0.41 (**100.0**)
DLBCL	89.64	80.89	86.96	97.50	97.50	97.50	97.50	97.50	96.10	**100.0**	100.0 ± 0.00 (**100.0**)
Average	67.73	72.38	77.16	85.70	85.76	88.03	89.10	89.44	90.09	92.29	***96.54***

(1) Non-SVM: traditional classification methods. (2) MC-SVM: multi-class support vector machines. (3) NN: backpropagation neural networks. (4) PNN: probabilistic neural networks. (5) KNN: K-nearest neighbors. (6) OVO: one-versus-one. (7) DAG: DAGSVM. (8) WW: method by Weston and Watkins. (9) CS: method by Crammer and Singer. (10) OVR: one-versus-rest. (11) CFS: correlation-based feature selection. (12) IBPSO: improved binary particle swarm optimization. (13) CFS–TGA: correlation-based feature selection–Taguchi genetic algorithm.Highest values are indicated in bold type. The standard deviation is given where available. For CFS–TGA, the highest value over ten runs is given in parentheses.

Table 7
Number of selected genes for each feature selection method.

Data set name	Original number of genes	IBPSO/KNN		CFS/KNN		CFS–TGA/KNN*	
		Number of selected genes	Percentage of original genes (%)	Number of selected genes	Percentage of original genes (%)	Number of selected genes	Percentage of original genes (%)
9_Tumors	5726	1280	22.4	47	0.8	24.6 ± 2.55	0.4
11_Tumors	12,533	2948	23.5	379	3.0	137.5 ± 4.35	1.1
14_Tumors	15,009	2777	18.5	89	0.6	53.7 ± 4.03	0.4
Brain_Tumor1	5920	754	12.7	141	2.4	44.7 ± 3.68	0.8
Brain_Tumor2	10,367	1197	11.6	117	1.1	33.1 ± 2.64	0.3
Leukemia1	5327	1034	19.4	93	1.8	22.4 ± 1.78	0.4
Leukemia2	11,225	1292	11.5	138	1.2	35.9 ± 2.92	0.3
Lung_Cancer	12,600	1897	15.1	550	4.4	195.2 ± 5.41	0.6
SRBCT	2308	431	18.7	112	4.9	29 ± 2.40	0.3
Prostate_Tumor	10,509	1294	12.3	87	0.8	24.7 ± 1.42	0.2
DLBCL	5469	1042	19.1	84	1.5	17.1 ± 2.33	0.3
Average			16.79		2.1		0.6

* Mean ± standard deviation (ten runs).

classification method that obtains high classification accuracy with relatively few features selected. The proposed method obtained the highest classification accuracies in ten out of the 11 data sets tested, with the sole exception being the 14_Tumors data set. An average classification accuracy of 100% was reached in six of the data sets, i.e., the 11_Tumors, Brain_Tumor2, Leukemia1, Leukemia2, SRBCT, and DLBCL data sets. For three more data sets, Brain_Tumors1 (99.45%), Lung Cancer (99.01%), and Prostate Tumor (99.22%). These classification accuracies were also the highest of the methods tested, although they were slightly lower than 100%. For the 9_Tumors data sample, the classification accuracy was again the highest, although in this case it was significantly lower than 100% (i.e., 91.67%). Only for the 14_Tumors data set did the presented method not achieve the highest accuracy. At 75.65%, it was about 1% lower than the highest achieved accuracy (76.60%). The average classification accuracy of CFS and CSF–TGA was 90.09% and 96.54%, respectively. This shows that TGA can increase classification accuracy by 6.45% compared to the exclusive use of CFS. Furthermore, the classification accuracy of the proposed method (96.54%) was better than the one of IBPSO (92.29%) [20]. The classification accuracies for the 9_Tumors and Brain_Tumor2 data sets obtained by the proposed method were 78.33% and 94.00%, respectively, an increase of (34.43% and 13.00%) and (25.33% and 16.17%) classification accuracy compared to the Non-SVM and MC-SVM methods. For the 11_Tumors and 14_Tumors data sets, the classification accuracy obtained by the proposed method was better than the classification accuracy of Non-SVMs and comparable to the MC-SVM methods.

4. Discussion

The performances of various classifiers for microarray data have been discussed. Each classifier has its advantages and disadvantages, so no single one can be considered ideal. As a classifier, KNN performs well for cancer classification when compared to the more sophisticated classifiers. KNN is an easily implemented method that has a simple parameter (the number of nearest neighbors) that needs to be predefined, given that the distance metric is Euclidean [44]. In order to enhance the accuracy and efficiency of classifiers, feasible feature reduction techniques are needed. A general overall feature selection approach can be found in Ref. [8].

Filter and wrapper approaches are used in this paper. Wang et al. [45] indicate that filter approaches can select important and

speedier feature subsets than wrapper approaches. However, wrapper approaches generally obtain better classification accuracies. Inza et al. [46] and Xiong et al. [47] used a wrapper approach to implement feature selection and selected better feature subsets to boost classification accuracy. Nevertheless, if only a wrapper approach is used, optimal solutions are difficult to find due to the vastness of the search space. We combined the advantages of filter and wrapper models in our proposed approach. In the CFS filter approach, merit is estimated from feature–feature and feature–class correlation by a heuristic search. The heuristic search measures the viability of feature subsets by taking into account the usefulness of features or filtering out features

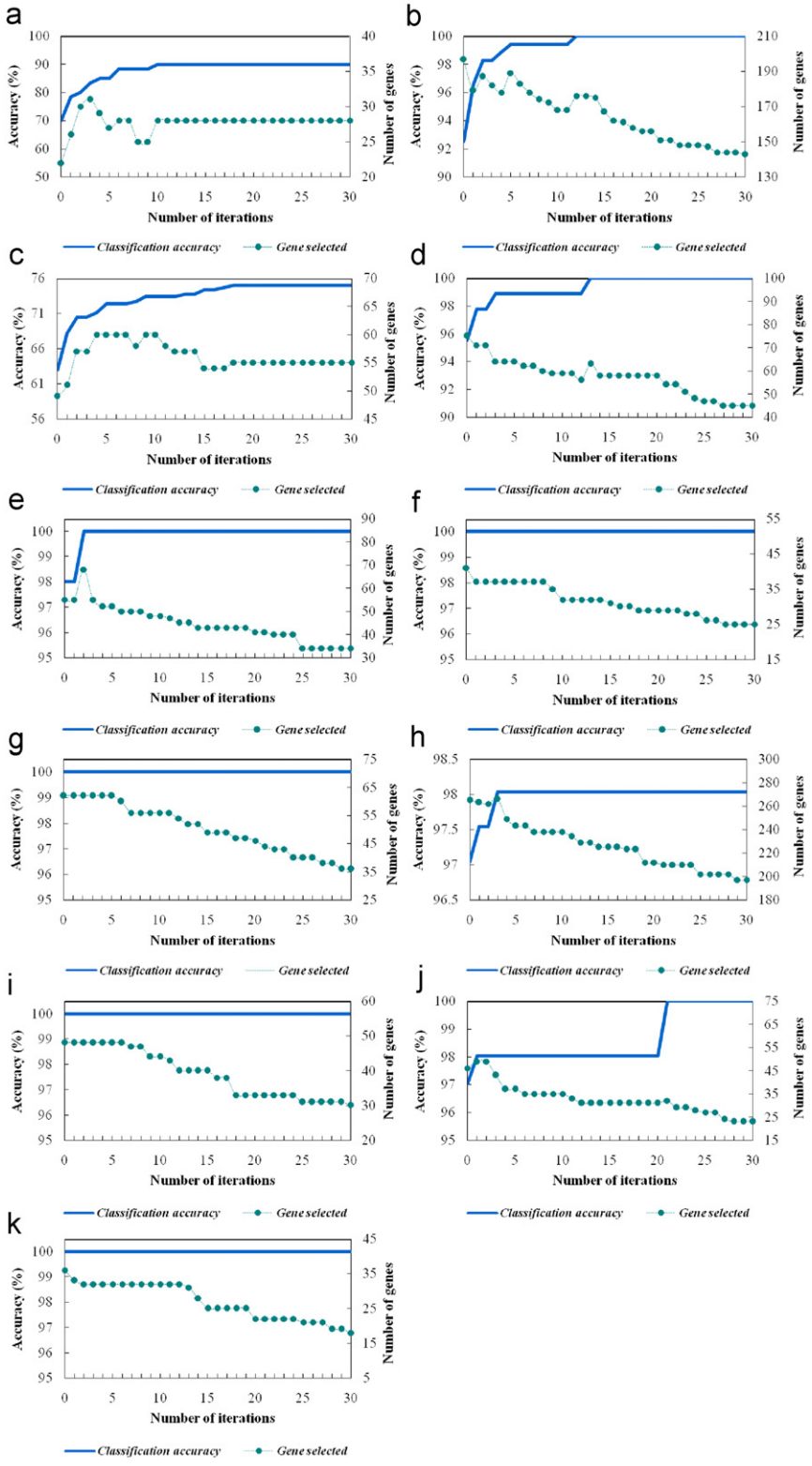

Fig. 2. Number of iterations vs. accuracy and number of genes: (a) 9_Tumors; (b) 11_Tumors; (c) 14_Tumors; (d) Brain_Tumor1; (e) Brain_Tumor2; (f) Leukemia1; (g) Leukemia2; (h) Lung_Cancer; (i) SRBCT; (j) Prostate_Tumor and (k) DLBCL.

irrelevant for the prediction of class labels. The CFS based heuristic thus evaluates the worth of a feature. It can be easily and quickly executed and extended to continuous class problems by applying suitable correlation measures. CFS can be of use to machines learning algorithms in terms of improving accuracy and comprehensibility of the induced models [9]. Hall and Holmes [48] indicate that CFS is much faster than other methods, and that it can improve classification performance. A GA is a stochastic search algorithm modeled on the process of natural selection in biological evolution. GAs has been successfully applied to many search, optimization, and machine learning problems [37]. In this paper, we combined these particular two methods because they complement each other well. Since CFS is a filter method, it searches through the feature space efficiently. TGA is a wrapper method that uses an induction algorithm to evaluate the feature subsets directly. As stated above, wrapper methods generally outperform filter methods in terms of prediction accuracy.

In the first stage of our hybrid feature selection approach, a correction-based feature weight of each feature is calculated and relevant features are identified. In the second stage, the previously identified relevant feature subsets are tested by a genetic algorithm, which tries to determine optimal feature subsets. These optimal feature subsets are then appraised by KNN. The Taguchi method is inserted between the crossover and mutation operations of the genetic algorithm. It is used to select genes more ideally suited for the crossover operation, thus generating representative chromosomes which can be used as the new offspring in the subsequent generation. The Taguchi principle improves the quality of a product by minimizing the effect of the causes of variation without eliminating these causes [24]. The two-level OA and the SNR of the Taguchi method are used for exploitation. The optimum chromosome can easily be found by using both experimental runs and SNRs instead of executing combinations of factor levels.

Table 7, which shows the reduction of gene numbers, gives proof that the CFS–TGA method reduces the large number of features effectively. CFS–TGA reduced the number of features by 99% on average. The retained features are only those that have a positive influence on classification. The results in Table 7 indicate that the combination of a filter approach (CFS) and a wrapper approach (TGA) is very effective in reducing the number of genes selected and can improve classification accuracy considerably over other methods. Fig. 2(a)–(k) shows the accuracy and number of genes vs. the number of iterations. The charts reveal that TGA converges fast and obtains the highest classification accuracy. The figures also show that the number of features selected is not necessarily associated with a high classification accuracy. It can be seen that as the number of selected features is reduced, the classification accuracy hits a certain maximum after which it does not further improve. Fig. 3 shows the range of accuracies over 10 trial runs for the eleven test data sets.

The Taguchi method implemented under the GA procedure is responsible for the local search. The Taguchi method is a robust design approach, which uses many ideas from statistical experimental design to improve the products, processes and equipment [24]. The Taguchi principle is used to improve the quality of a product by minimizing the effect of the causes of variation without eliminating these causes [24]. The two-level orthogonal array and the SNR of the Taguchi method are used for exploitation. The optimum particles can easily be identified by using both experimental runs and SNRs instead of executing combinations of all of factor levels. Consequently, a superior candidate feature subset with high classification performance for the classification task at hand can be obtained in a subsequent iteration. An illustrative example was given in the example section. Since feature subsets b_1 and b_2 have *seven* different features, $2^7=128$ possible experimental trials have to be considered in a full factorial experimental design. The Taguchi method decreases the number of experimental trials associated with these *seven* different features to eight trials (see Table 3). Prior to the classification process, feature subset evaluation efforts can thus be significantly reduced based on the two-dimensional, fractional factorial experimental design matrix. Features important and relevant for pattern classification can easily singled out.

5. Conclusions

Classification problems associated with microarray data analysis constitute a very important research area in bioinformatics.

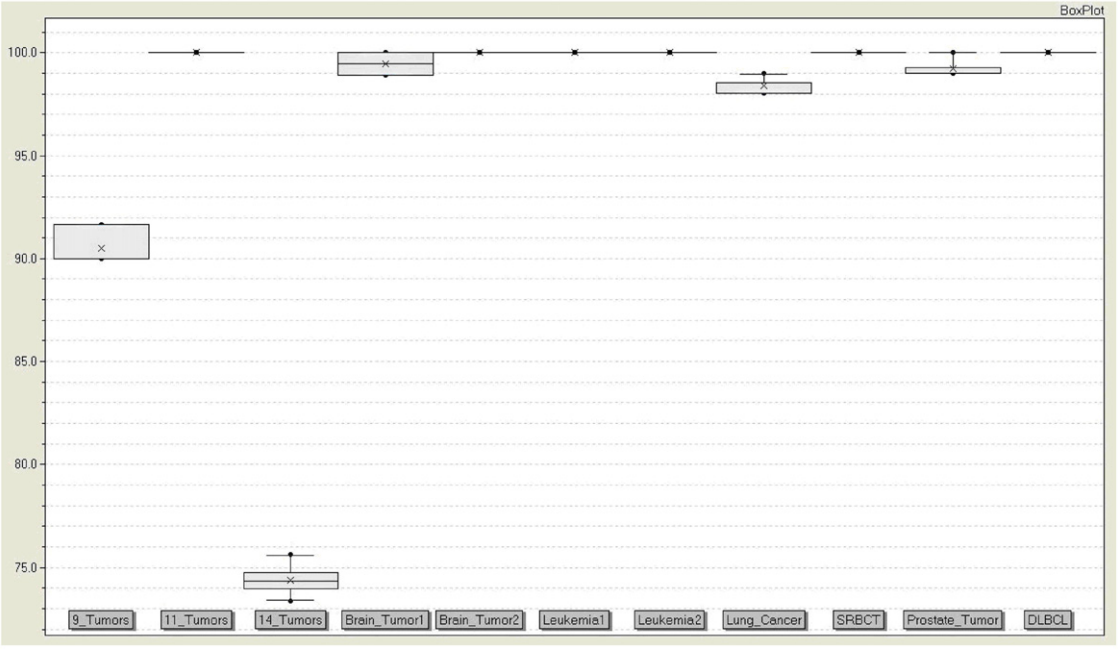

Fig. 3. Boxplot of the range of accuracies over 10 trial runs for the eleven test data sets.

In this paper, a filter (CFS) and wrapper (TGA) feature selection method were merged into a new hybrid method, and KNN with LOOCV method served as a classifier for 11 classification profiles. Experimental results show that this method effectively simplifies feature selection by reducing the total number of features needed. The classification accuracy obtained by the proposed method had the highest classification accuracy in ten out of the eleven gene expression data sets it was tested on. The proposed method could conceivably be used in other research projects that implement feature selection.

Conflict of interest statement

None declared.

Acknowledgement

This work is partly supported by the National Science Council in Taiwan under Grants NSC96-2622-E-151-019-CC3, NSC96-2622-E214-004-CC3, NSC95-2221-E-151-004-MY3, NSC95-2221-E-214-087, NSC95-2622-E-214-004, NSC94-2622-E-151-025-CC3, and NSC94-2622-E-151-025-CC3.

References

[1] K.-J. Kim, S.-B. Cho, An evolutionary algorithm approach to optimal ensemble classifiers for DNA microarray data analysis, IEEE Transactions on Evolutionary Computation 12 (2008) 377–388.
[2] Y. Tang, Y.-Q. Zhang, Z. Huang, X. Hu, Y. Zhao, Recursive fuzzy granulation for gene subsets extraction and cancer classification, IEEE Transactions on Information Technology in Biomedicine 12 (2008) 723–730.
[3] S.-L. Wang, X. Li, S. Zhang, J. Gui, D.-S. Huang, Tumor classification by combining PNN classifier ensemble with neighborhood rough set based gene reduction, Computers in Biology and Medicine 40 (2010) 179–189.
[4] Q. Shen, Z. Mei, B.-X. Ye, Simultaneous genes and training samples selection by modified particle swarm optimization for gene expression data classification, Computers in Biology and Medicine 39 (2009) 646–649.
[5] J. Xie, W. Xie, C. Wang, X. Gao, A. Novel, Hybrid feature selection method based on IFSFFS and SVM for the diagnosis of erythemato-squamous diseases, in: Proceedings of the JMLR: Workshop and Conference Proceedings, Cumberland Lodge, Windsor, UK, 2010, pp. 142–151.
[6] B. Li, C.-H. Zheng, D.-S. Huang, L. Zhang, K. Han, Gene expression data classification using locally linear discriminant embedding, Computers in Biology and Medicine 40 (2010) 802–810.
[7] J. Huang, H. Fang, X. Fan, Decision forest for classification of gene expression data, Computers in Biology and Medicine 40 (2010) 698–704.
[8] Y. Saeys, I. Inza, P. Larranaga, A review of feature selection techniques in bioinformatics, Bioinfomatics 23 (2007) 2507–2517.
[9] M.A. Hall, Correlation-based feature subset selection for machine learning, PhD thesis, Department of Computer Science, University of Waikato, Hamilton, New Zealand, 1999.
[10] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[11] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (1994) 537–550.
[12] X. Liu, A. Krishnan, A. Mondry, An entropy-based gene selection method for cancer classification using microarray data, BMC Bioinformatics 6 (2005) 76.
[13] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.
[14] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (2005) 491–502.
[15] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, A.K. Jain, Dimensionality reduction using genetic algorithms, IEEE Transactions on Evolutionary Computation 4 (2000) 164–171.
[16] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, IEEE Transactions on Computers C-26 (1977) 917–922.
[17] P. Pudil, J. Novovicov, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (1994) 1119–1125.
[18] L.-Y. Chuang, C.-H. Yang, C.-H. Yang, Tabu search and binary particle swarm optimization for feature selection using microarray data, Journal of Computational Biology 16 (2009) 1689–1703.
[19] H. Zhang, G. Sun, Feature selection using tabu search method, Pattern Recognition 35 (2002) 701–711.
[20] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, C.-H. Yang, Improved binary PSO for feature selection using gene expression data, Computational Biology and Chemistry 32 (2008) 29–38.
[21] L.-Y. Chuang, C.-S. Yang, K.-C. Wu, C.-H. Yang, Correlation-based gene selection and classification using Taguchi-BPSO, Methods of Information in Medicine 49 (2010) 254–268.
[22] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1424–1437.
[23] P. Yang, B. Zhou, Z. Zhang, A. Zomaya, A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data, BMC Bioinformatics 11 (2010) S5.
[24] J.-T. Tsai, T.-K. Liu, J.-H. Chou, Hybrid Taguchi-genetic algorithm for global numerical optimization, IEEE Transactions on Evolutionary Computation 8 (2004) 365–377.
[25] S.Y. Sohn, H.W. Shin, Experimental study for the comparison of classifier combination methods, Pattern Recognition 40 (2007) 33–40.
[26] N. Kwak, C.-H. Choi, Input feature selection for classification problems, IEEE Transactions on Neural Networks 13 (2002) 143–159.
[27] W.-C. Chen, P.-H. Tai, M.-W. Wang, W.-J. Deng, C.-T. Chen, A neural network-based approach for dynamic quality prediction in a plastic injection molding process, Expert Systems with Applications 35 (2008) 843–849.
[28] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.
[29] E. Fix, J.L. Hodges Jr, Discriminatory analysis. Nonparametric discrimination: consistency properties, Technical Report. USAF School of Aviation Medicine, Randolph Field, TX., 1951, pp. 238–247.
[30] G.C. Cawley, N.L.C. Talbot, Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers, Pattern Recognition 36 (2003) 2585–2592.
[31] M. Stone, Cross-validatory choice and assessment of statistical predictions, Journal of the Royal Statistical Society. Series B (Methodological) 36 (1974) 111–147.
[32] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[33] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[34] G. Taguchi, S. Chowdhury, S. Taguchi, Robust Engineering, McGraw-Hill, New York, NY, 2000.
[35] Y. Wu, A. Wu, Taguchi Methods for Robust Design, ASME Press, New York, NY, 2000.
[36] E. Frank, M. Hall, L. Trigg, G. Holmes, I.H. Witten, Data mining in bioinformatics using Weka, Bioinfomatics 20 (2004) 2479–2481.
[37] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[38] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, 1998.
[39] A. Statnikov, C.F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy, A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, Bioinformatics 21 (2005) 631–643.
[40] Á.E. Eiben, R. Hinterding, A.E.E.R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Transactions on Evolutionary Computation 3 (2000) 124–141.
[41] J. Arabas, Z. Michalewicz, J. Mulawka, GAVaPS—a genetic algorithm with varying population size, in: Proceedings of the Conference on Evolutionary Computation, Orlando, FL, 1994, pp. 73–78.
[42] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics 19 (2005) 43–53.
[43] A.K. Ghosh, On optimum choice of k in nearest neighbor classification, Computational Statistics & Data Analysis 50 (2006) 3113–3123.
[44] K. Deb, A. Raji Reddy, Reliable classification of two-class cancer data using evolutionary algorithms, Biosystems 72 (2003) 111–129.
[45] Y. Wang, I.V. Tetko, M.A. Hall, E. Frank, A. Facius, K.F.X. Mayer, H.W. Mewes, Gene selection from microarray data for cancer classification—a machine learning approach, Computational Biology and Chemistry 29 (2005) 37–46.
[46] I. Inza, P. Larranaga, R. Blanco, A.J. Cerrolaza, Filter versus wrapper gene selection approaches in DNA microarray domains, Artificial Intelligence in Medicine 31 (2004) 91–103.
[47] M. Xiong, X. Fang, J. Zhao, Biomarker identification by feature wrappers, Genome Research 11 (2001) 1878–1887.
[48] M.A. Hall, G. Holmes, Benchmarking attribute selection techniques for discrete class data mining, IEEE Transactions on Knowledge and Data Engineering 15 (2003) 1437–1447.