Journal of Signal Processing 50, 293–304, 2008

DOI: 10.1007/s11265-007-0120-3

Effective Gene Selection Method Using Bayesian Discriminant Based Criterion and Genetic Algorithms

ZHAOHUI GAN, TOMMY W. S. CHOW AND D. HUANG
Department of Electronic Engineering, City University of Hong Kong, Hong Kong, People's Republic of China

Received: 15 October 2006; Revised: 20 February 2007; Accepted: 13 June 2007

Abstract. Microarray gene expression data usually consist of a large amount of genes. Among these genes, only a small fraction is informative for performing cancer diagnostic tests. This paper focuses on effective identification of informative genes. A newly developed gene selection criterion using the concept of Bayesian discriminant is used. The criterion measures the classification ability of a feature set. Excellent gene selection results are then made possible. Apart from the cost function, this paper addresses the drawback of conventional sequential forward search (SFS) method. New genetic algorithms based Bayesian discriminant criterion is designed. The proposed strategies have been thoroughly evaluated on three kinds of cancer diagnoses based on the classification results of three typical classifiers which are a multilayer perception model (MLP), a support vector machine model (SVM), and a 3-nearest neighbor rule classifier (3-NN). The obtained results show that the proposed strategies can improve the performance of gene selection substantially. The experimental results also indicate that the proposed methods are very robust under all the investigated cases.

Keywords: gene selection, genetic algorithms, feature selection, sequential forward search

1. Introduction

Microarray techniques, such as DNA chip and high-density oligonucleotide chip, are powerful biotechnological means because they are able to record the expression levels of thousands of genes simultaneously [12]. Systematic and computational analysis on microarray data enables us to understand phenological and pathologic issues in a genomic level [11, 12]. Microarray data, however, always contains a huge gene set of up to thousands and a small sample set that down to tens. Moreover, only a very small fraction of the genes are informative for a certain task [13]. Different diseases are related to different gene sets. Intuitively, research on the identification of cancer-causing genes has become very challenging. Effectively tackling this problem has many merits. Using a small gene set, we can conduct computational data analysis in a relatively low-dimensional data domain. This is very useful to deliver precise, reliable and interpretable results. Also, with the gene selection results, biology researchers can focus only on the marker genes, and confidently ignore the irrelevant genes. The cost of biological experiments and decisions can thus be greatly reduced.

Various machine learning and statistical feature selection models have been directly applied or adapted to gene selection/reduction [13, 15, 17, 19, 21, 27, 29, 31, 32, 35]. A (gene) feature selection framework basically consists of two parts: a search engine to determine the promising feature subset candidates and a criterion to determine which candidate is the best [22, 20]. There are several

search engines including ranking, optimal search, heuristic search and stochastic search. Feature selection models are categorized as filter model, wrapper model and embedded model according to the type of evaluation criterion.

The filter selection model that uses heuristic/stochastic search engines [21, 27, 29, 31], wrapper model [10, 15] and the embedded selection model [15, 19, 32] are the three widely used feature selection frameworks for conducting gene selection. In this paper, we focus on the former one. A typical filter model is selected and implemented to demonstrate the capabilities of our proposed strategies. In this model, the employed search engine is the genetic algorithms (GA). The evaluation criterion is based on Bayesian discriminant because it can evaluate the classification ability of a feature subset in a straightforward fashion [16].

In a gene (feature) selection scheme, the evaluation criterion and search engine play equally important roles. Evaluation criteria have been heavily investigated in many studies [13, 15, 19, 21, 27, 29, 31, 32, 35]. In contrast, study on search engines has drawn little attentions. The heuristic search engines, especially the sequential forward/backward search (SFS/SBS), and the stochastic search engines (e.g., Genetic algorithm) are pervasively employed for gene selection. The sequential forward selection (SFS) [9] methods start from an empty set and gradually add features selected by some evaluation function while the sequential backward selection (SBS) [9] schemes start with the complete feature set and discard features one by one till an optimum feature subset is retained. However, in SFS once a feature is selected, it cannot be rejected later and reverse is true for SBS.

Evolutionary algorithms have also been used for Forward Search (FS) [3, 8, 23, 26, 28]. Siedleki and Sklansky [28] used a branch-and-bound technique in their GA for FS. Casillas et al. [3] devised a genetic FS scheme for fuzzy rule based classification systems. Richeldi and Lanzi [26] proposed a genetic algorithm based FS scheme with C4.5 induction learning. In [23], Pal et al. proposed a new genetic operator called self-crossover for FS. Despite all these work, there have been very few attempts on using genetic algorithms (GA) for gene selection [3, 27, 28].

The genetic algorithms consist of several operators. Each operator affects the performance of the GA in different ways. In the case of GA based gene selection, the fitness function is one of the main elements affecting the overall result significantly. Different fitness functions may deliver very different results on the same dataset. In [8], a GA based gene selection method was proposed. Its fitness function consists of two parts. The first part is the deviation of one gene between other genes within the same group. Another part is the deviation of a gene group between other gene groups. The objective is to get the gene with the smallest deviation within the group and the biggest deviation between the groups. In [4], another gene selection method using GA and support vector machines (SVM) was proposed. SVM was used as discriminant to evaluate the effectiveness of a subset of expressed genes. GA was applied to identify the best subsets in the combinational space of feature subsets. In [7], an evolutionary algorithm, which utilized a score function as fitness function, was proposed for gene selection. In this approach, higher scores were given to certain genes when more data points were correctly classified. Simulated annealing was applied to evolutionary algorithm for speeding up the convergence. It is clear that GA has been used as search engine in different approaches, but they all deliver different results when working with different fitness functions.

In this study, we propose the use of genetic algorithm (GA) together with Bayesian discriminant cost function to address the aforesaid problems. The main aim of this study is to enhance the effectiveness of searching. We analyze search engines from the perspective of global optimization theory [2]. The analysis of this type, which has been overlooked in most previous studies, reveals a major drawback of sequential search methods. It is found that conventional search engines cannot perform optimization in a maximal way because their searching mechanisms do not completely incorporate with global optimization theory. To address this drawback, we employ genetic algorithms to formulate a new searching strategy, in which gene selection is conducted along the possible global optimization direction.

The presentation of this paper is organized as follows. In Section 2, the probability based sequential forward search (SFS) gene selection model is briefed. Our proposed search strategies are detailed in Section 3. In Section 4, simulation examples are presented and discussed. We make a conclusion in Section 5.

2. Sequential Forward Gene Selection Process

Assume that we have a classification dataset $D = \{X\ C\} = \{(x_1, c_1), (x_2, c_2),..., (x_N, c_N)\}$, (x_i, c_i) represents a set of data sample, in which x_i is the input vector, and c_i records the class label of x_i. x_i is an M-dimensional vector, that is, a sample is described with the expression levels of M genes. We represent these genes as $F = \{f_1, f_2,...,f_M\}$. Moreover, the samples in D are grouped into L classes denoted as $\{\omega_1,...,\omega_L\}$. For a data sample (say, x_i), we have $c_i = \omega_k$, where $1 \le k \le L$. A gene subset evaluation criterion is represented by $\Phi(S)$, where S is a gene subset. Furthermore, without loss of generality, we suppose that a large value of $\Phi(S)$ means a good S. Thus, the goal of a gene selection process is to maximize $\Phi(S)$ through adjusting S. In general, $\Phi(S)$ is optimized in the following ways. After a pool of gene subsets is suggested according to certain rules, $\Phi(S)$ of each suggested subset is calculated. The one with the optimal $\Phi(S)$ is selected. The schemes for determining the gene subset pools require trading the quality of optimization results with computational complexity. Among these schemes, the SFS strategy is the most popular one. The steps are summarized as followings.

Step 1: Set the selected gene subset S to empty.
Step 2: Repeat the following until certain stopping conditions are met. Identify the most useful gene (say, g_u) from the unselected genes, and place it into S. g_u satisfies $g_u = \arg\max_g \Phi(S + g)$.

One of the main shortcoming of SFS is that it is a greedy search scheme and can only deliver local optimal results. To alleviate this shortcoming, there are several modification approaches. In the stepwise strategy [25], that is, the floating (compound) search, selecting k features (genes) is followed by eliminating j "worst" selected ones, where j is less than k. Al-Ani and Deriche [1] used only the "elite" selected features to identify the important items from unselected features. Although these methods enhance the performance of SFS to a certain extent, they cannot change the basic working rationale of SFS. The modified SFS are still a type of greedy search strategy and cannot deliver globally optimal results.

For gene evaluation, many criteria can be used. In this study, Bayesian discriminant based criterion (BD) [16] is employed. With the given dataset D, BD is defined as

$$BD(S) = \frac{1}{N}\sum_{i=1}^{N} \log \frac{p_S(c_i|x_i)}{p_S(\overline{c_i}|x_i)} = \frac{1}{N}\sum_{i=1}^{N} \log \frac{p_S(c_i|x_i)}{1 - p_S(c_i|x_i)}, \quad (1)$$

where $\overline{c_i}$ means all the classes but class c_i, and $p_S(.)$ represents a probability density function estimated based on the gene set S. In order to estimate the posterior probabilities $p(c|x)$ in Eq. (1), the margin probability $p(x)$ and the joint probability $p(x,c)$ should be firstly obtained. We use Parzen window [24] to build $p(x)$ and $p(x,c)$. Given the aforementioned dataset $D = \{X,C\}$, Parzen window estimators are modeled as

$$p(x, c) = \sum_{x_i \in \text{class } c} p(x_i)p(x|x_i) = \sum_{x_i \in \text{class } c} p(x_i)\kappa(x - x_i, h_i), \quad (2)$$

$$p(x) = \sum_{\text{all class } c} p(x, c) = \sum_{\text{all } x_i \in X} p(x_i)\kappa(x - x_i, h_i), \quad (3)$$

where κ is the kernel function and h_i is the width of κ. With a proper selection of $\kappa(\cdot)$ and h, a Parzen window estimator can converge to the real probability density. We choose the Gaussian function as κ, that is,

$$\kappa(x - x_i, h_i) = G(x - x_i, h_i) = \frac{1}{(2\pi h_i^2)^{M/2}} \exp\left(-\frac{1}{2h_i^2}(x - x_i)^T(x - x_i)\right),$$

where M is the dimension of x. The width h_i is set with $h_i = 2d(x_i,x_j)$, where $d(x_i,x_j)$ is the Euclidean distance between x_i and x_j, and x_j is the third nearest neighbor of x_i. Following the general rule, we have $P(x_i) = 1/N$. Thus, according to the Bayes formula we can model $p(c|x)$ as

$$p(c|x) = \frac{p(x|c)P(c)}{p(x)} = \frac{p(x,c)}{p(x)} = \frac{\sum_{x_i \in \text{class } c} p(x_i)\kappa(x - x_i, h_i)}{\sum_{\text{all } x_i} p(x_i)\kappa(x - x_i, h_i)}.$$

3. Genetic Algorithms for Gene Selection Process

In order to effectively overcome the shortcoming of conventional SFS, we use GA as a search engine. The GA is biologically inspired and has many mechanisms mimicking natural evolution. It has been widely applied to numerous scientific and engineering optimization problems or search problems. The following additional GA procedures are incorporated into our proposed searching scheme.

3.1. Chromosome Encoding

In the gene selection problem, we encode a chromosome as a variable length integer string. An integer represents a feature with value presenting the feature number. For example, chromosome 3267, 654, 2109 means that the 3267th, 654th, and 2109th features are selected, while all other features are removed. The length of a chromosome is the number of features selected, which is determined by users. Once the length is defined by the user, it becomes a constant in the algorithm. Given by the fact that most genes originally given in a microarray dataset are irrelevant to certain tasks, a widely used gene pre-filtering strategy is used to eliminate the irrelevant or the largely insignificant genes before the commencement of gene selection. This is an effective way to relieve the computational burden. In our study, we firstly use the gene pre-filtering strategy to remove irrelevant genes, implying more informative genes are kept. We set the number of remained genes to 200, which is very sufficient for including all the possible informative genes. As a result, the maximal length of chromosome encoding in our proposed scheme is 200. The initial population is randomly generated.

3.2. Fitness Evaluation

The objective of gene selection is to optimize an evaluation criterion. In this study, we use a Bayesian discriminant based criterion (BD) [16] as the evaluation criterion. The fitness of a chromosome C is defined as $fitness\ (C) = BD(X_C)$, where X_C is the corresponding feature subset of C.

3.3. Selection, Replacement, and Stop

The chromosome selection for the next generation is conducted on the basis of fitness. The selection mechanism should ensure that fitter chromosomes have a higher probability of survival. Our design uses the rank-based selection scheme. The individuals in the population are sorted in terms of their fitness in descending order, and the ith individual is assigned a probability of selection by a linear function. A larger value of probability enforces a stronger selection pressure. We selected two parent chromosomes using the above method. The crossover operation generates a new chromosome (offspring) out of the two parents. The mutation operation slightly perturbs the offspring. If the offspring chromosome is superior to both parents, it replaces the parent. If it is in between the two parents, it replaces the inferior parent; otherwise, the most inferior chromosome in the population is replaced. When the number of running generations reaches the preset value, the genetic algorithm stops.

3.4. Crossover and Mutation

The standard single point crossover and mutation operators is used. It chooses one cutting points at random and alternately copies each segment out of the two parents. The operations are exemplified in Fig. 1.

The parents' chromosome P1 and P2 consist of five genes. At the third gene two parents crossover each other, which means that the fourth and the fifth genes of the two parents exchange with each other. After the crossover of two parents, the mutation is applied to the offspring. As only the best 200 features are selected at the pre-filter procedure, the numbers of 200 selected best features are stored in memory units encoded by number 1 to 200. The mutation procedures are as follows:

Step 1. Randomly generate N numbers within [0, 1] to be stored in the first column of matrix M. N numbers within [1,L] are stored in the second column of matrix M. N numbers within [1,200] are stored in the third column of matrix M, where N is the number of population, and L is the length of the

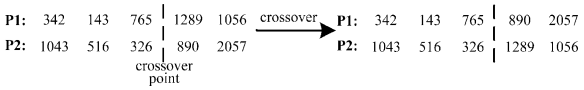

Figure 1. An example of one point crossover.

chromosome encoding. The second column of matrix M is the position of chromosome of an individual gene being mutated, while the third column is the mutation results, which correspond to the gene position, with values ranged between 1 to 200. Figure 2a shows a typical content of matrix M.

Step 2. Compare the N numbers located at the first column of matrix M with mutation probability P_m. If a number is less than P_m, return 1, otherwise, return 0. Those returned values are stored in the first column of matrix M as shown in Fig. 2b. As a result, this will form a vector including only 1 and 0. If an element of this vector is 1, then the corresponding individual of population will mutate.

For instance, from Fig. 2, we know that the second individual will mutate. The 15th gene will mutate. The result of mutation is stored at the 100th memory unit. Suppose the value being stored at the 100th memory unit is 451, the result of mutation is the 451th gene.

3.5. Parameters

There is no general systematic parameter optimization approach for GA. The parameters of GA depend on the given tasks, but tuning the GA parameters to appropriate values for a specific data set may give rise to an improved performance. In our study, the following parameter set was chosen.

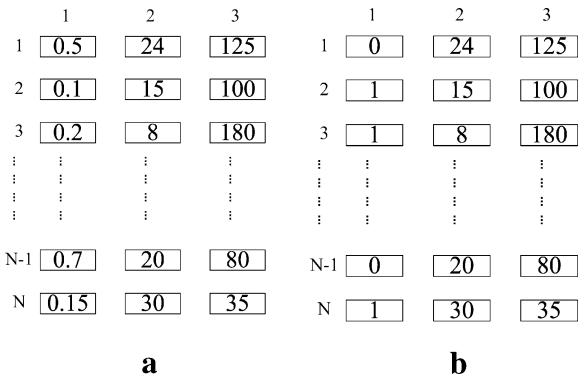

Figure 2. The content of matrix M. **a** Typical content of matrix M. **b** Returned values stored in the first column of matrix M.

Population size	50
Pc (crossover probability)	0.8
Pm (mutation rate)	0.3
Ger (maximum generation)	300

3.6. Flowchart of Genetic Algorithms

The computational complexity of the SFS is $O(M^2)$, where M is the number of genes. A microarray gene expression dataset generally contains information of thousands or ten thousands genes. Clearly, directly handling a huge gene set may cost an unbearable computational burden. Given by the fact that most genes originally given in a microarray dataset are irrelevant to certain tasks, a widely used pre-filtering-gene strategy is used to eliminate the irrelevant or insignificantly relevant genes before the commencement of gene selection. This is an effective way to relieve the computational burden. In details, for each given gene g, $BD(g)$ is calculated based on Eq. (1). The genes with small values of BD are considered irrelevant and eliminated. In such as way, a huge gene set can be safely reduced.

With the simple pre-filter operation, our proposed GA based gene selection procedures are as follows.

Step 1 (Pre-filter) Calculate the BD of each given gene, and rank these genes in a descending order of BD. Keep the top two hundred genes for the following selection process.

Step 2 (Initialization) randomly generate N individuals as the initial population.

Step 3 (Gene selection) Repeat the following until the maximal generation reaches.

(a) (Selection) the fitter chromosomes are selected from the N individuals in the population for evolution. The fitness of a chromosome C is defined as fitness (C) = BD(X_C),the selection of the fitter chromosomes is based on the fitness value.

(b) (Crossover) choose one cutting points at random and alternately copies each segment out of the two parents.

(c) (Mutation) determines which bit of character string of a individual in population will mutate, and change it as another gene.

4. Experimental Results

Our proposed GA based gene selection method is evaluated through comparing with several related methods, namely, the conventional SFS, support machine learning recursive feature elimination scheme (SVM RFE) [15], and the another GA based gene selection method. SVM RFE, a typical embedded feature selection model, begins with the training of an SVM (of linear kernel) with all the given features. According to the parameters of the trained SVM, features are ranked in terms of their importance. This enables half of the features to be eliminated. The training-SVM-eliminating-half-of-features process repeats until no feature is left. In order to further validate the effect of GA, we also implement another kind of GA, which uses the same genetic operators but different selection operator (roulette wheel selection operator). In this paper, we call it GA1 in the following discussions.

Prior to gene selection, each gene variable is preprocessed having zero mean and unit variance. Our focus is placed on comparing the SFS with our proposed GA. But detailed comparative performance between the BD based gene (feature) selection models and other related methods, for instance, mutual information based ones, SVM based approach, and distribution-based method, can be found in [16].

Different gene selection methods are compared on several cancer diagnosis datasets. In these real datasets, no a priori knowledge is available. We rely on experimental classification results to assess the quality of gene selection results. Using a selected gene subset, certain classifiers are constructed on training data that are also used for gene selection. Then, we evaluate the built classifiers on the testing dataset. Good classification results must indicate a respectable gene subset. We use four typical classifiers including a multilayer perception model (MLP), a support vector machine models (SVM), and a 3-nearest neighbor rule classifier (3-NN). The MLP used in our study is available at http://www.ncrg.aston.ac.uk/netlab/. For convenience, we set the number of hidden neurons to 6 for all the investigated examples. It is worth noting that slightly different number of hidden neurons does not have an effect on the overall performance. The number of training cycles is set to 100 to ease the problem of overfitting. The other learning parameters are set with default values. The SVM models used in this study are available at http://www.isis.ecs.soton.ac.uk/resources/ svminfo. In this study we employed only the SVM model with "Linear" kernel (SVM-L).

The following gene expression datasets are included in our study.

(a) Colon tumor classification The original raw data are published at http://www.research.i2r.a-star.edu.sg/rp/. It is noted that there are two methods used to pre-process the Colon cancer dataset. One method is to take the logarithm of all values to diminish the data value. It ignores the data when it contains zeros [8, 10, 15]. Another method only normalizes the feature vectors without taking the logarithm of all values [16, 34]. In our experiment, the latter method is used. This dataset contains 62 samples collected from colon-cancer patients. Among these samples, 40 samples are tumor, and 22 are labeled "normal". There are 2,000 genes selected based on the confidence in the measured expression levels. We split the 62 samples into ten disjoint groups. In each evaluation, one group was used for testing while the other nine groups were used for training. The investigations were repeated on tenfold of training and testing data to deliver reliable evaluations.

(b) Prostate cancer classification The objective of this task is to distinguish prostate cancer cases from non-cancer cases. The original raw data are published at http://www.genome.wi.mit.edu/mpr/prostate. This dataset consists of 102 samples from the same experimental conditions. Each sample is described using 12,600 genes. We split the 102 samples into ten disjoint groups—one group was used for testing while the other nine groups were used for training. Similar to the last example, the studies on this dataset were repeated on tenfold of training and testing data. The results are summarized and presented in this paper.

(c) Leukemia subtype classification This dataset, which are available at http://www.broad.mit.edu/cgibin/cancer/datasets.cgi, are used for performing leukemia subtype classification. The given samples are labeled with ALL, MLL or AML. Training data contains 57 samples—20 labeled with ALL, 17 with MLL and 20 with AML. In the test data, there are 15 samples—4 ALL samples,

3 MLL ones and 8 AML ones. There are no SVM-related results in this example because the SVM models we employed are designed to deal with two-class data only.

(d) Effect of GA parameters In the gene selection method which GA serves as a searching engine, many parameters, for example, population size, maximum generation number, crossover probability *Pc* and mutation rate *Pm* etc, exhibit different effect on the performance of gene selection. In this section, we evaluate their effect on gene selection using colon cancer dataset. First, we keep *Pc* and *Pm*as constants. The population size is ranged from 20 to 200, and the maximum generation number varies from 300 to 600 in order to maintain stable gene selection results. Figure 3 shows that GA all reaches convergence after 300 generations when number of selected genes is 10, 20 and 30 respectively. To let the maximum generation number as 300 is found to be appropriate. The variation of population size and maximum generation number posed unnoticeable effect on gene selection results. Secondly, we maintain the population size and maximum generation number to 50 and 300 respectively. Crossover probability, *Pc*, and mutation rate, *Pm*, vary from 0.1 to 0.8 respectively. Figure 4 shows the classification accuracy of 3NN, SVM-L and MLP when the gene number is 10. From Fig. 4, we can conclude that both the *Pc* and *Pm* exhibit significant effect on gene selection results. In the above presented results, the best GA parameters set were used to conduct the simulations. The parameters set were as follows:

Population size	50
Pc (crossover probability)	0.8
Pm (mutation rate)	0.3
Ger (maximum generation)	300

(e) Comparisons of SFS and GA To demonstrate the merits of our developed strategies, the GA are compared with other methods, especially with the SFS, in terms of classification accuracy. In the colon cancer dataset and prostate cancer dataset, we divide all the dataset into tenfolds. Among them, ninefolds are used for selecting

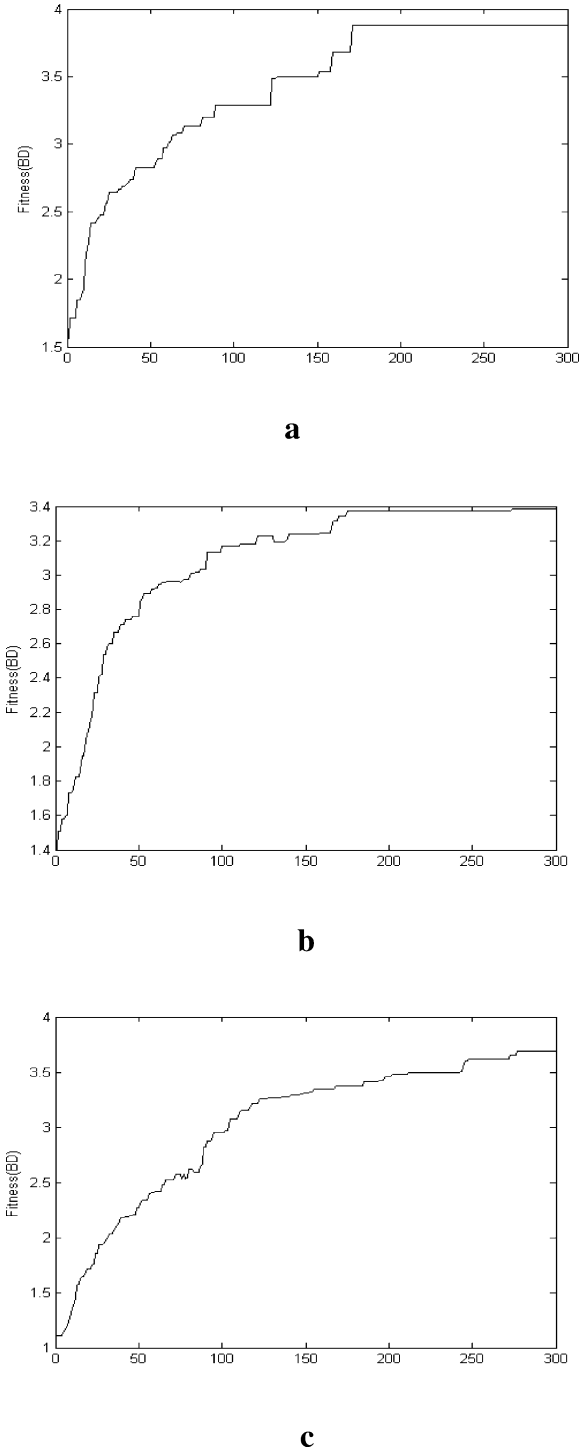

Figure 3. **a** Fitness value over generation number when gene number is 10. **b** Fitness value over generation number when gene number is 20. **c** Fitness value over generation number when gene number is 30.

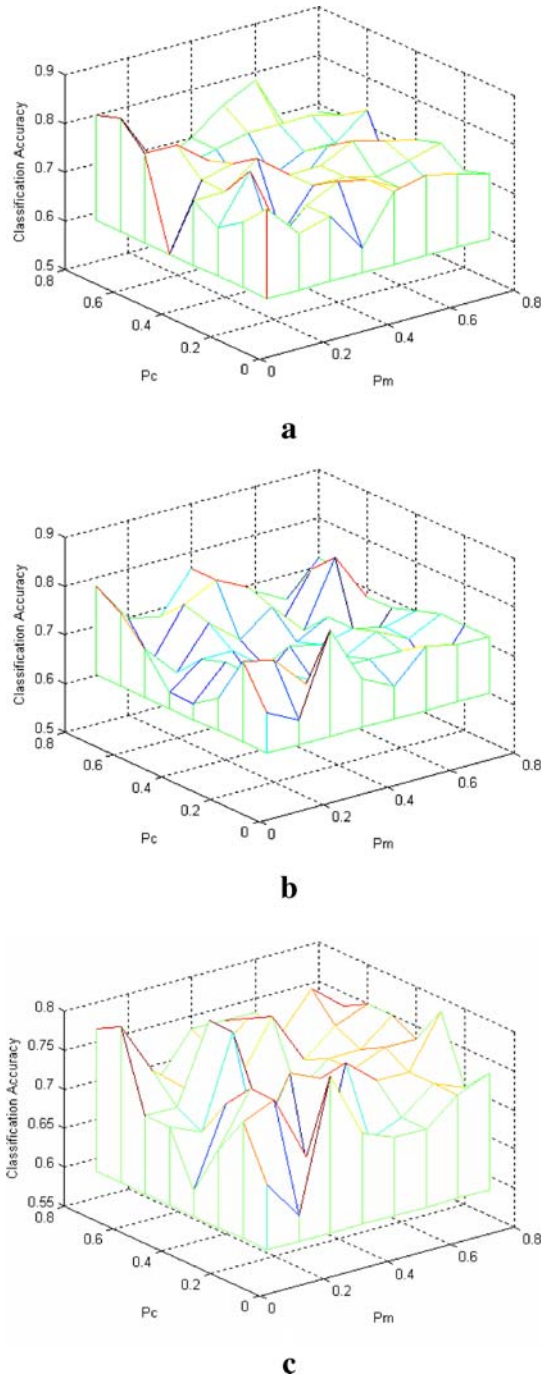

Figure 4. **a** Classification accuracy of 3NN on colon cancer dataset, with our proposed method when *Pc* and *Pm* ranging from 0.1 to 0.8 and gene number is 10. **b** Classification accuracy of SVM on colon cancer dataset, with our proposed method when *Pc* and *Pm* ranging from 0.1 to 0.8 and gene number is 10. **c** Classification accuracy of MLP on colon cancer dataset, with our proposed method when *Pc* and *Pm* ranging from 0.1 to 0.8 and gene number is 10.

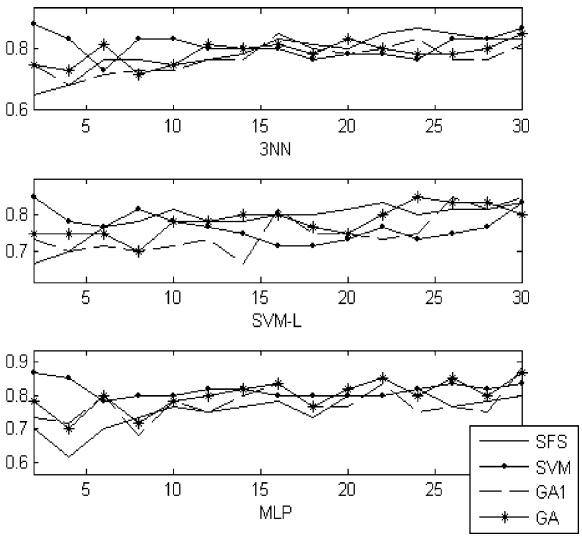

Figure 5. Comparison between SFS, SVM-RFE, GA and GA1 in terms of classification accuracy on the colon cancer classification data. In these figures, the y-axes are the classification accuracy, and the x-axes are the number of the selected genes.

features and training classifier. The other fold is used for performing classification test. The whole feature selection, training classifiers and classification runs on a given dataset with tenfold Cross Validation method. As leukemia subtype dataset has fixed training and classification dataset, the whole feature selection, training classifiers and classification are only run once on the

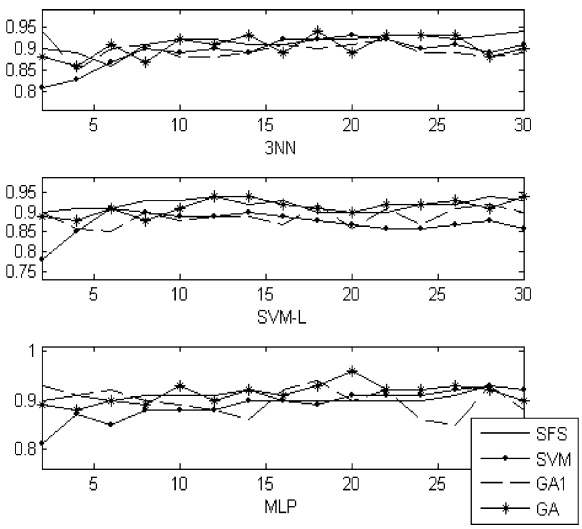

Figure 6. Comparison between SFS, SVM-RFE, GA and GA1 in terms of classification accuracy on the prostate cancer classification data. In these figures, the y-axes are the classification accuracy, and the x-axes are the number of the selected genes.

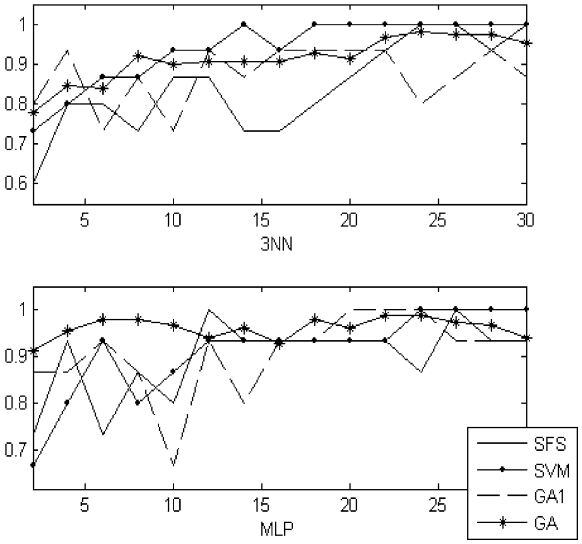

Figure 7. Comparison between SFS, SVM-RFE, GA and GA1 in terms of classification accuracy on for leukemia subtype classification data. In these figures, the y-axes are the classification accuracy, and the x-axes are the number of the selected genes.

dataset. The comparative results are presented in Fig. 5 for colon cancer classification, in Fig. 6 for prostate cancer classification and in Fig. 7 for leukemia subtype classification.

In most examples, such as the ones about colon cancer and prostate cancer, the proposed GA scheme outperforms the conventional SFS and shows comparative performance with the SVM-RFE method. Clearly, this is attributed to the proposed GA searching strategies. In addition, GA1 delivers similar results with our proposed GA method.

(f) Detailed results on prostate cancer Apart from the above machine-learning-based evaluations, we check the obtained results from the biological point of views. In Table 1, one gene result of the GA is listed. The functions of these genes range from cell adhesion (VCL, NELL2) to immune response (DF, C7), from cellular transport (MRC2, RBP1) to regulation of transcription (LMO3), from protein kinase activity (ILK) to hormone activity (IGF1).

Table 1. The genes that are identified WMSFS to be related with the prostate cancer.

Order of selection	Gene symbol	Gene title	Relation with prostate cancer
1	VCL	Vinculin	Vinculin, a cytoskeletal protein, can regulate the ability of cancer cell to move away from tumors. It may contribute to metastatic process of prostate cancer.
2	DF	D component of complement (adipsin)	Adipsin, a member of the trypsin family of peptidases, is a component of the alternative complement pathway playing an important part in humoral suppression of infectious agents.
			Uzma et al. [30] find out this gene up-regulates in the samples with prostate diseases, such as prostate cancer. Also, [6] suggest it a good cancer marker.
3	MRC2	Mannose receptor, C type 2	
4	NELL2	NEL-like 2 (chicken)	The close correlation of this gene to prostate cancer is also suggested in other studies ([30, 33]).
5	RBP1	Retinol binding protein 1, cellular	Retinoids are involved in cell growth, differentiation, and carcinogenesis. This gene has been found to overexpress in prostate carcinoma [18].
6	C7	Complement component 7	This gene takes part in androgen-regulated processes that play important roles in malignant transformation of prostate gland [30].
7	IGF1	Homeodomain interacting protein kinase 3	The role that this gene plays in prostate development and carcinogenesis has been well-recognized and widely examimed [5].
8	ILK	Integrin-linked kinase	This gene overexpression can suppress anoikis, promote anchorage-independent cell cycle progression, and induce tumorigenesis and invasion [14].
9	GAGEC1	G antigen, family C, 1	The protein encoded in this gene is PAGE4, which is a Cytoplasmic protein and is prostate associated.
10	LMO3	LIM domain only 3 (rhombotin-like 2)	The protein encoded in this gene is a LIM-only protein (LMO), which is involved in cell fate determination. This gene has been noted to upregulate in the prostate cancer samples [30].

We note that almost all of these selected genes have been associated with development and diagnosis of prostate cancer—some of them are well-known prostate-cancer-associated genes, such as IGF1, GAGEC1, RBP1, DF, NELL2, ILK, etc., and others have been suggested to overexpress in prostate cancer samples, for example, C7, LMO3.

5. Conclusion

A genetic algorithms based gene selection method working together with Bayesian discriminant cost function is proposed for performing gene selection. As GA is a kind of random search method, which can find the optimal or near optimal solution, the overall performance of gene selection is substantially enhanced compared with using conventional SFS search engine. The results obtained on real data demonstrate that the proposed strategies deliver very promising improvement on gene selection.

It is worth noting that the proposed strategy is only applicable to one representative gene selection model—BD based genetic algorithms search. In future work, we will extend these strategies to other gene selection models and further evaluate their merits and limitations.

Acknowledgment

The work described in this paper was fully supported by a grant (SRG-7001998-570) from the City University of Hong Kong.

References

1. A. Al-Ani and M. Deriche, "Optimal feature selection using information maximisation: case of biomedical data," in *Proc. of the 2000 IEEE Signal Processing Society Workshop*, vol. 2, 2000, pp. 841–850.
2. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, New York, 1995.
3. J. Casillas, O. Cordon, M. J. Del Jesus, and F. Herrera, "Genetic Feature Selection in a Fuzzy Rule-based Classification System Learning Process for High-dimensional Problems," *Inf. Sci.*, vol. 136, 2001, pp. 135–157.
4. X. W. Chen, "Gene Selection for Cancer Classification Using Bootstrapped Genetic Algorithms and Support Vector Machines," Proc. Bioinformatics Conference, 2003.
5. I. Cheng, D. O. Stram, K. L. Penney, M. Pike et al., "Common Genetic Variation in IGF1 and Prostate Cancer Risk in the Multiethnic Cohort," *J. Natl. Cancer Inst.*, vol. 98, no. 2, 2006, pp. 123–124.
6. M. L. Chow, E. J. Moler, and I. S. Mian, "Identifying Marker Genes in Transcription Profiling Data Using a Mixture of Feature Relevance Experts," *Physiol. Genomics*, vol. 5, 2001, pp. 99–111.
7. J. Deutsch, "Evolutionary Algorithms for Finding Optimal Gene Sets in Microarray Prediction," *Bioinformatics*, vol. 19, 2003, pp. 45–52.
8. S. Ding et al., "A Genetic Algorithm Applied to Optimal Gene Subset Selection," Evolutionary Computation, Congress on, CEC2004. vol. 2, 2004, pp. 1654–1660, Jun.
9. P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
10. Kai-bo Duan et al., "Multiple SVM-RFE for Gene Selection in Cancer Classification with Expression Data," *IEEE Trans. Nanobioscience*, vol. 4, no. 3, 2005, pp. 228–234, Sep.
11. S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of Discrimination Methods for the Classification of Tumours Using Gene Express Data," *J. Am. Stat. Assoc.*, vol. 97, no. 457, 2002, pp. 77–87.
12. R. Ekins and F. W. Chu, "Microarrays: Their Origins and Applications," *Trends Biotech.*, vol. 17, 1999, pp. 217–218.
13. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard et al., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, 1999, pp. 531–537.
14. J. R. Graff, J. A. Deddens, B. W. Knoicek, B. M. Colligan et al., "Integrin-linked Kinase Expression Increases with Prostate Tumor Grade," *Clin. Cancer Res.*, vol. 7, 2002, pp. 1987–1991.
15. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification Using Support Vector Machines," *Mach. Learn.*, vol. 46, 2002, pp. 389–422.
16. D. Huang and T. W. S. Chow, "Efficiently Searching the Important Input Variables Using Bayesian Discriminant," *IEEE Trans. Circuits Syst.*, vol. 52, no. 4, 2005, pp. 785–793.
17. D. Huang, T. W. S. Chow, E. W. M. Ma, and J. Li, "Efficient Selection of Salient Features from Microarray Gene Expression Data for Cancer Diagnosis," *IEEE Trans. Circuits Syst. Part I*, vol. 52, no. 9, 2005, pp. 1909–1918.
18. C. Jerónimo, R. Henrique, J. Oliveira, F. Lobo et al., "Aberrant Cellular Retinol Binding Protein 1 (CRBP1) Gene Expression and Promoter Methylation in Prostate Cancer," *J. Clin. Pathol.*, vol. 57, 2004, pp. 872–876.

19. K. E. Lee, N. Sha, E. R. Dougherty et al., "Gene Selection: A Bayesian Variable Selection Approach," *Bioinformatics*, vol. 19, no. 1, 2003, pp. 90–97.
20. H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer, London, UK, 1998.
21. X. Liu, A. Krishnan, and A. Mondry, "An Entropy-based Gene Selection Method for Cancer Classification Using Microarray Data," *BMC Bioinformatics*, vol. 6, no. 76, 2005.
22. L. C. Molina, L. Belanche, and A. Nebot, "*Feature Selection Algorithms: A Survey and Experimental Evaluation*," available at: http://www.lsi.upc.es/dept/techreps/html/R02-62.html, Technical Report, 2002.
23. N. R. Pal, S. Nandi, and M. K. Kundu, "Self-crossover: A New Genetic Operator and Its Application to Feature Selection," *Int. J. Syst. Sci.*, vol. 29, no. 2, 1998, pp. 207–212.
24. E. Parzen, "On the Estimation of a Probability Density Function and Mode," *Ann. Math. Stat.*, vol. 33, 1962, pp. 1064–1076.
25. P. Pudil, J. Novovicova, and J. Kittler, "Floating Search Methods in Feature Selection," *Pattern Recogn. Lett.*, vol. 15, 1994, pp. 1119–1125.
26. M. Richeldi and P. Lanzi, "Performing Effective Feature Selection by Investigating the Deep Structure of the Data," in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*. Menlo Park, CA, 1996, pp. 379–383.
27. S. C. Shah and A. Kusiak, "Data Mining and Genetic Algorithm Based Gene/SNP Selection," *Artif. Intell. Med.*, vol. 31, no. 3, 2004, pp. 183–196.
28. W. Siedlecki and J. Sklansky, "A Note on Genetic Algorithms for Large Scale Feature Selection," *Pattern Recogn. Lett.*, vol. 10, 1989, pp. 335–347.
29. T. J. Umpai and S. Aitken, "Feature Selection and Classification for Microarray Data Analysis: Evolutionary Methods for Identifying Predictive Genes," *BMC Bioinformatics*, vol. 6, no. 148, 2005.
30. S. S. Uzma and H. G. Robert, "Fingerprinting the Diseased Prostate: Associations between BPH and Prostate Cancer," *J. Cell. Biochem.*, vol. 91, 2004, pp. 161–169.
31. E. P. Xing, M. I. Jordan, and M. Karp, "Feature Selection for High-dimensional Genomic Microarray Data," in *Proc. 18th Intl. Conf. On Machine Learning*, 2001.
32. K. Yeung, R. E. Bumgarner, and A. E. Raftery, "Bayesian Model Averaging: Development of An Improved Multi-class, Gene Selection and Classification Tool for Microarray Data," *Bioinformatics*, vol. 21, no. 10, 2005, pp. 2394–2402.
33. C. Zhang, Hai-Ri Li, Jian-Bing Fan, J. Wang-Rodriguez et al., "Profiling Alternatively Spliced mRNA Isoforms for Prostate Cancer Classification," *BMC Bioinformatics*, vol. 7, 2006, pp. 202–236.
34. Chaolin Zhang et al., "Significance of Gene Ranking for Classification of Microarray Samples," *IEEE/ACM Trans. Computational Biology and Bioinformatics*, vol. 3, no. 3, 2006, pp. 312–320.
35. X. Zhou, X. Wang, and E. Dougherty, "Nonlinear Probit Gene Classification Using Mutual Information and Wavelet-based Feature Selection," *J. Biol. Syst.*, vol. 12, no. 3, 2004, pp. 371–386.

Zhaohui Gan is with College of Information Science and Engineering, Wuhan University of Science & Technology. He received his B.Eng. degree in Electrical Engineering from Wuhan University of Science and Technology, China in 1990 and the M.Eng. degree in Electrical Engineering from the same University in 1995. He is currently working toward the Ph.D. degree in Dept. of Electronic Engineering, City University of Hong Kong, HongKong. His research interests include pattern recognition, machine learning and evolutionary computation.

Tommy W. S. Chow received the B.Sc. (1st Hons.) and Ph.D. degrees from the Department of Electrical and Electronic Engineering, University of Sunderland, U.K. in 1984 and 1988, respectively. He undertook his Trainee with Reyrolle Technology at Tyne and Wear, U.K. His Ph.D. research worked on a collaborative project between The International Research and Development, Newcastle Upon Tyne, U.K, and the Ministry of Defense (Navy) U.K. He is a Professor in the Department of Electronic Engineering at the City University of Hong Kong, Hong Kong. He has been

working on different consultancy projects with the Mass Transit Railway, Kowloon-Canton Railway Corporation, Hong Kong. He has also conducted other collaborative projects with the Kong Electric Co. Ltd, and Royal Observatory Hong Kong, and the MTR Hong Kong on the application of neural networks for machine fault detection and forecasting. He is the author and co-author of numerous published works, including book, book chapters, and over 110 journal articles related to his research. His main research has been in the area of machine learning, optimizations, bioinformatics, and fault diagnostics. Prof. Chow received the Best Paper Award in 2002 IEEE Industrial Electronics Society Annual meeting in Seville, Spain. He was the Chairman of Hong Kong Institute of Engineers, Control Automation and Instrumentation Division from 1997 to 1998.

D. Huang received the Ph.D. degree in 2005 with the Department of Electronic Engineering of City University of Hong Kong. Her research focuses on machine learning and bioinformatics.