Available online at www.sciencedirect.com
ScienceDirect
Computational Biology and Chemistry 32 (2008) 29–38
Computational Biology and Chemistry
www.elsevier.com/locate/compbiolchem

Improved binary PSO for feature selection using gene expression data

Li-Yeh Chuang [a], Hsueh-Wei Chang [b], Chung-Jui Tu [c], Cheng-Hong Yang [c,*]

[a] *Department of Chemical Engineering, I-Shou University, Kaohsiung 840, Taiwan*
[b] *Department of Biomedical Science and Environmental Biology, and Graduate Institute of Natural Products, College of Pharmacy, Kaohsiung Medical University, Kaohsiung, 807, Taiwan*
[c] *Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 807, Taiwan*

Received 24 December 2006; accepted 10 September 2007

Abstract

Gene expression profiles, which represent the state of a cell at a molecular level, have great potential as a medical diagnosis tool. Compared to the number of genes involved, available training data sets generally have a fairly small sample size in cancer type classification. These training data limitations constitute a challenge to certain classification methodologies. A reliable selection method for genes relevant for sample classification is needed in order to speed up the processing rate, decrease the predictive error rate, and to avoid incomprehensibility due to the large number of genes investigated. Improved binary particle swarm optimization (IBPSO) is used in this study to implement feature selection, and the *K*-nearest neighbor (*K*-NN) method serves as an evaluator of the IBPSO for gene expression data classification problems. Experimental results show that this method effectively simplifies feature selection and reduces the total number of features needed. The classification accuracy obtained by the proposed method has the highest classification accuracy in nine of the 11 gene expression data test problems, and is comparative to the classification accuracy of the two other test problems, as compared to the best results previously published.

Keywords: Improved binary particle swarm optimization; Feature selection; Gene expression data

1. Introduction

DNA microarray examples are generated by a hybridization of mRNA from sample tissues or blood and cDNA (in the case of a spotted array), as well as hybridization of oligonuclectides of DNA (in the case of Affymetrix chips, this is done on the surface of the chip-array). DNA microarray technology allows for the simultaneous monitoring and measurement of thousands of gene expression activation levels in a single experiment. Class memberships are characterized by the production of proteins, i.e. gene expressions refer to the production level of proteins specific for a gene. Thus, 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 in screening for potential tumor markers. Differential expressions of genes are analyzed statistically and genes are assigned to various classes, which may (or may not) enhance the understanding of the underlying biological processes.

Microarray gene expression technology has opened the possibility of investigating the activity of thousands of genes simultaneously. Gene expression profiles show the measurement of relative abundance of mRNA corresponding to the genes. Thus, discriminant analysis of microarray data has great potential as a medical diagnosis tool. The goal of microarray data classification is to build an efficient and effective model that identifies the differentially expressed genes and may be used to predict class membership for any unknown samples. The challenges posed in microarray classification are the limited size of samples in comparison to the high dimensionality of the sample, along with experimental variations in measured gene expression levels.

The classification of gene expression data samples involves feature selection and classifier design. Generally, only a small number of genes show a strong correlation with a certain phenotype compared to the total number of genes investigated. Thus, in order to analyze gene expression profiles correctly, feature (gene) selection is most crucial for the classification process. The goal of feature selection is to identify the subset of differentially

* Corresponding author. Tel.: +886 7 370 6752.
E-mail addresses: chuang@isu.edu.tw (L.-Y. Chuang), changhw@kmu.edu.tw (H.-W. Chang), 1093320134@cc.kuas.edu.tw (C.-J. Tu), chyang@cc.kuas.edu.tw (C.-H. Yang).

1476-9271/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2007.09.005

expressed genes that are potentially relevant for distinguishing the sample classes. A good method of selecting genes relevant for sample classification is needed in order to accelerate the processing rate, decrease the predictive error rate, and to avoid incomprehensibility due to spurious data correlations; it should be based on the number of genes investigated. Several methods have been used to perform feature selection on the training and testing data, e.g. genetic algorithms (Raymer et al., 2000; Yang and Honavar, 1998), branch and bound algorithms (Narendra and Fukunage, 1997; Yu and Yuan, 1993), sequential search algorithms (Pudil et al., 1994), mutual information (Roberto, 1994), tabu search (Zhang and Sun, 2002) entropy-based methods (Liu et al., 2005), regularized least squares (Ancona et al., 2005), random forests (Diaz-Uriarte and Alvarez de Andres, 2006), instance-based methods (Berrar et al., 2006), and least squares support vector machines (Tang et al., 2006).

In this paper, improved binary particle swarm optimization (IBPSO) is used to implement the feature selection process. A *K*-nearest neighbor (*K*-NN) serves as an evaluator of the IBPSO for gene expression data classification problems taken from the literature. The results reveal that the proposed classification method achieves superior predictive error rate when applied to 11 data sets from the literature, as compared to methods previously published. Furthermore, the number of genes selected can be significantly decreased.

2. Methods

2.1. Improved Binary Particle Swarm Optimization (IBPSO)

Particle swarm optimization (PSO) is a population-based stochastic optimization technique, which was developed by Kennedy and Eberhart (1995). PSO simulates the social behavior of organisms, such as birds in a flock and fish in a school. This behavior can be described as an automatically and iteratively updated system. In PSO, each single candidate solution can be considered a particle in the search space. Each particle makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. All of the particles have fitness values, which are evaluated by a fitness function to be optimized. During movement, each particle adjusts its position by changing its velocity according to its own experience and according to the experience of a neighboring particle, thus making use of the best position encountered by itself and its neighbor. Particles move through the problem space by following a current of optimum particles. The process is then iterated a fixed number of times or until a predetermined minimum error is achieved (Kennedy et al., 2001).

PSO was originally introduced as an optimization technique for real-number spaces. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other application problems. A comprehensive survey of the PSO algorithms and their applications can be found in Kennedy et al. (2001). However, many optimization problems occur in a space featuring discrete, qualitative distinctions between variables and between levels of variables. Kennedy and Eberhart introduced binary PSO (BPSO), which can be applied to discrete binary variables. In a binary space, a particle may move to near corners of a hypercube by flipping various numbers of bits; thus, the overall particle velocity may be described by the number of bits changed per iteration (Kennedy and Eberhart, 1997).

Gene expression data characteristically have a high dimension, so we expect superior classification results in different dimension areas. Each particle adjusts its position according to two fitness value, *pbest* and *gbest*, to avoid being trapped in a local optimum by fine-tuning the inertia weight. *pbest* is a local fitness value, whereas *gbest* constitutes a global fitness value. If the *gbest* value is itself trapped in a local optimum, a search of each particle limit in the same area will occur, thereby preventing superior results of classification. Thus, we propose a method that retires *gbest* under such circumstances and uses an improved binary particle swarm optimization (IBPSO). By resetting *gbest* we can avoid IBPSO getting trapped in a local optimum, and superior classification result can be achieved with a reduced number of selected genes.

Fig. 1a shows that almost all particles converged near *gbest* after a certain period. If the *gbest* value does not change after three iterations, it can be considered stuck at a local optimum. Under such circumstances, the current *gbest* fitness value (classification accuracy and selected features) is reset to zero, i.e. retired (Fig. 1b). This form of IBPSO skips the local optimum, and searches for superior classification results in an area with a lower number of genes. Fig. 1c shows that the individual particles will converge towards the reset *gbest* value and thus leave the local optimum by searching for the new *gbest* value in a region with a lower number of genes (Fig. 1d). This process achieves superior classification and effectively reduces the number of genes that need to be selected.

In this paper, an improved form of binary PSO (IBPSO) was used since the position of each individual particle can be given in binary form (0 or 1), which adequately reflects the straightforward "yes/no" choice of whether a feature needs to be selected

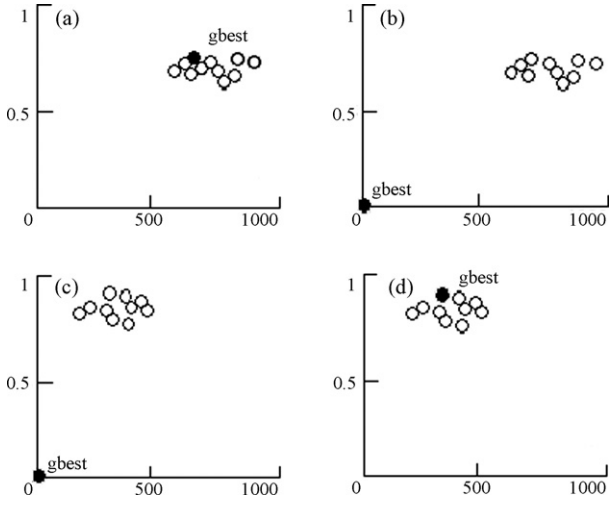

Fig. 1. (a) *gbest* is trapped in a local. (b) *gbest* is rest to zero. (c) Particle movement after resetting of *gbest*. (d) Particles congregated towards the updated *gbest* value, improving the individual position.

or not. The changes in particle velocity can be interpreted as a change in the probability of finding the particle in one state or another. Since this change is a probability it is limited to a range of {0.0–1.0}.

2.2. K-Nearest Neighbor Method

The *K*-nearest neighbor (*K*-NN) method was first introduced by Fix and Hodges in 1951, and is one of the most popular nonparametric methods (Cover and Hart, 1967; Fix and Hodges, 1951). The purpose of the algorithm is to classify a new object based on attributes and training samples. The *K*-nearest neighbor method consists of a supervised learning algorithm where the result of a new instance query is classified based on the majority of the *K*-nearest neighbor categories. Based only on memory, the classifiers do not use any model for fitting. This method works based on a minimum distance from the query instance to the training samples to determine the *K*-nearest neighbors. Any tied results are solved by a random procedure.

The *K*-NN method has been successfully applied in many areas: statistical estimation, pattern recognition, artificial intelligence, categorical problems, and feature selection. One advantage of the *K*-NN method is that it is simple and easy to implement. Invariant to noisy training data (Cover and Hart, 1967), *K*-NN is not negatively affected when the training data is large. Disadvantages of the *K*-NN method are the need to determine parameter *K* (number of nearest neighbors), calculate the distances between the query instance and all the training samples, sort the distances and determine the nearest neighbors based on the *K*th minimum distance, as well as determine the categories of the nearest neighbors. Computation cost is quite high because distances from each query instance to all training samples need to be computed. Some indexing (e.g. *K–D* tree) may reduce this computational cost (Palau and Snapp, 1998).

In this study, the feature subset was measured by the leave-one-out cross-validation (LOOCV) of one nearest neighbor (1-NN). Neighbors are calculated using their Euclidean distance. The fitness value for the 1-NN evolves according to the LOOCV method for all datasets. In the LOOCV method, a single observation from the original sample is selected as the validation data, and the remaining observations are selected as training data. This is repeated so that each observation in the sample is used once as the validation data. This is essentially the same as *K*-fold cross-validation where *K* is equal to the number of observations in the original sample. The obtained classification accuracy is an adaptive functional value.

2.3. IBPSO–NN Procedure

Feature selection was implemented using IBPSO, and a *K*-NN served as an evaluator for the classification obtained by IBPSO. The procedure of the proposed method is the following: initially, the position of each particle is represented in binary string form and is randomly generated; the bit value {0} and {1} represent a non-selected and selected feature, respectively. The predictive accuracy of a 1-NN determined by the LOOCV method is used to measure the fitness of an individual. The best fitness value for each particle is $pbest_p$ (p is number of particles) and the best fitness value within a group of $pbest_p$ is the global fitness value *gbest*. Once *pbest* and *gbest* are obtained, we can keep track of the features of *pbest* and *gbest* particles with regard to their position and velocity. Each particle is updated according to the following equations:

$$v_{\mathrm{pd}}^{\mathrm{new}} = w v_{\mathrm{pd}}^{\mathrm{old}} + c_1 \mathrm{rand}_1 \left(pbest_{\mathrm{pd}} - x_{\mathrm{pd}}^{\mathrm{old}} \right) + c_2 \mathrm{rand}_2 \left(gbest_{\mathrm{d}} - x_{\mathrm{pd}}^{\mathrm{old}} \right) \quad (1)$$

$$\text{if } v_{\mathrm{pd}}^{\mathrm{new}} \notin (V_{\min}, V_{\max}) \quad \text{then } v_{\mathrm{pd}}^{\mathrm{new}} = \max(\min(V_{\max}, v_{\mathrm{pd}}^{\mathrm{new}}), V_{\min}) \quad (2)$$

$$S(v_{\mathrm{pd}}^{\mathrm{new}}) = \frac{1}{1 + \mathrm{e}^{-v_{\mathrm{pd}}^{\mathrm{new}}}} \quad (3)$$

$$\text{if } (\mathrm{rand} < S(v_{\mathrm{pd}}^{\mathrm{new}})) \quad \text{then } x_{\mathrm{pd}}^{\mathrm{new}} = 1; \text{ else } x_{\mathrm{pd}}^{\mathrm{new}} = 0 \quad (4)$$

In these equations, w is the inertia weight, c_1 and c_2 acceleration (learning) factors, and rand, rand_1 and rand_2 are random numbers. Velocities $v_{\mathrm{pd}}^{\mathrm{old}}$ and $v_{\mathrm{pd}}^{\mathrm{new}}$ are those of the old and new particle, respectively, $v_{\mathrm{pd}}^{\mathrm{old}}$ the current particle position (solution), and $v_{\mathrm{pd}}^{\mathrm{new}}$ is the updated particle position (solution).

In Eq. (2), particle velocities of each dimension are tried to a maximum velocity $V_{\max}$. If the sum of accelerations causes the velocity of that dimension to exceed $V_{\max}$, then the velocity of that dimension is limited to $V_{\max}$. $V_{\max}$ and $V_{\min}$ are user-specified parameters ($V_{\max} = 6$, $V_{\min} = -6$).

The new velocity of particles is produced by the product of the inertia weight and the old particle velocity; the cognition-only model of particles and the social-only model of particles can be applied. The cognition-only model of particles belongs to the local optimum, and the social-only model of particles belongs to the global optimum.

The PSO converges rapidly during the initial stages of a search, but then often slows considerably and particles can get trapped in a local optimum (Stacey et al., 2003). In order to avoid particles getting trapped in a local optimum, the *gbest* value has to be evaluated before each particle position is updated. If *gbest* has the same value for a preset number of times (in our case three times), the particle could conceivably be trapped in a local optimum. In such a case, the *gbest* position is reset to zero in the fitness function (classification accuracy), meaning that zero features are selected while *pbest* is kept. In the next iteration, particles in the neighborhood of the local optima will adjust their position by congregating towards the *gbest* position.

The feature after updating is calculated by the function $S(v_{\mathrm{pd}}^{\mathrm{new}})$ (Eq. (3)), in which the velocity value is $v_{\mathrm{pd}}^{\mathrm{new}}$. If $S(v_{\mathrm{pd}}^{\mathrm{new}})$ is larger than a randomly produced disorder number that is within {0.0–1.0}, then its position value F_n, $n = 1, 2, \ldots, m$ is represented as {1} (meaning this feature is selected as a required feature for the next iteration). If $S(v_{\mathrm{pd}}^{\mathrm{new}})$ is smaller than a randomly generated disorder number that is within {0.0–1.0}, then

its position value $F_n, n = 1, 2, \ldots, m$ is represented as $\{0\}$ (meaning this feature is not selected as a required feature for the next iteration).

The whole procedure is repeated until either the fitness of a particle is 1.0 or the number of iterations is 100 (maximum number of iterations). The three factors rand_1, rand_2 and rand are random numbers between (0, 1), whereas c_1 and c_2 are learning factors, $c_1 = c_2 = 2$ (Kennedy et al., 2001). The above values were taken from Shi and Eberhart (1998), as well as Hsu and Lin (2002). The pseudo-code of the proposed method for gene expression classification problems is given below:

Pseudo-code for IBPSO–NN procedure

begin
  Randomly initialize particle swarm
  while (number of iterations, or the stopping criterion is not met)
    Evaluate fitness of particle swarm by **1-Nearest Neighbor** ()
    for $p = 1$ to number of particles
      if fitness of X_p is greater than the fitness of $pbest_p$ **then**
        $pbest_p = X_p$
      end if
      if fitness of any particle of the particle swarm is greater than *gbest* **then**
        gbest = position of particle
      end if
      if fitness of *gbest* is the same Max times **then** give up and reset *gbest*
      end if
      for $d = 1$ to number of dimension of particle
        $v_{\text{pd}}^{\text{new}} = w v_{\text{pd}}^{\text{old}} + c_1 \text{rand}_1 \left(pbest_{\text{pd}} - x_{\text{pd}}^{\text{old}} \right) + c_2 \text{rand}_2 \left(gbest_{\text{d}} - x_{\text{pd}}^{\text{old}} \right)$
        if $v_{\text{pd}}^{\text{new}} \notin (V_{\min}, V_{\max})$ then $v_{\text{pd}}^{\text{new}} = \max(\min(V_{\max}, v_{\text{pd}}^{\text{new}}), V_{\min})$
        $S(v_{\text{pd}}^{\text{new}}) = \frac{1}{1+e^{-v_{\text{pd}}^{\text{new}}}}$
        if $\left(rand < S\left(v_{\text{pd}}^{\text{new}} \right) \right)$ **then** $x_{\text{pd}}^{\text{new}} = 1$ **else** $x_{\text{pd}}^{\text{new}} = 0$
      next ***d***
    next ***p***
  next generation until stopping criterion
end

Pseudo-code for 1-nearest neighbor procedure

begin
  for $i = 1$ to sample number of classification problem
    for $j = 1$ to sample number of classification problem
      for $k = 1$ to dimension number of classification problem
        $\text{dist}_i = \text{dist}_i + (\text{data}_{ik} - \text{data}_{jk})^2$
      next ***k***
      if $\text{dist}_i <$ nearest **then**
        $\text{class}_i = \text{class}_j$
        nearest = dist_i
      end if
    next ***j***
  next ***I***
  for $i = 1$ to sample number of classification problem
    if class = real class of testing data **then** correct = correct + 1
    end if
  next ***i***
  Fitness value = correct/number of testing data
end

3. Results and Discussion

Selecting relevant genes for gene expression classification is a common challenge in bioinformatics. Classification and prediction of gene expression data is a prerequisite for current genetic research in biomedicine and molecular biology, since a correct analysis of results can help biologists solve complex biological problems. Gene expression data can effectively be used for gene identification, cell differentiation, pharmaceutical development, cancer classification, and disease diagnosis and prediction. However, due to the fact that gene expression data is of a high dimensionality and has a small sample size, classification of gene expression data is time-consuming. Choosing feature selection as a pretreatment method prior to the actual classification of gene expression data can effectively reduce the calculation time without negatively affecting predictive error rate.

Due to the peculiar characteristics of gene expression data (high number of genes and small sample size) 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. Feature selection uses relatively fewer features since only selective features need to be used. This does not affect the predictive error rate in a negative way; on the contrary, predictive error rate can even be improved.

In this study, the datasets consist of 11 gene expression profiles, which were downloaded from http://www.gems-system.org. They include tumor, brain tumor, leukemia, lung cancer, and prostate tumor samples. The dataset formats are shown in Table 1, which contains the dataset name and a detailed description.

Gene subset ranking and selection bias were not used in this paper. The initial position of each particle is randomly generated. Two loops, inner loop and outer loop, were used for cross-validation in Statnikov et al. (2004). The inner loop and outer loop are used to determine the best parameter of a classifier and estimate the performance of the classifier, respectively. In this paper, only one cross-validation cycle was used (the feature subset was measured by the leave-one-out cross-validation of one nearest neighbor.), not two nested ones, because the parameter used was previously determined to be optimized in the literature (Shi and Eberhart, 1998).

The data format shown in Table 2 includes the dataset name, number of samples, categories, samples, genes, selected genes, and percentage of gene selected percentage. The average percentage of genes selected is 0.17. The highest and lowest percentage of genes selected is reduced to 0.24 and 0.12 for the 11_Tumors, Brain_Tumor2, Leukemia2, and Prostate_Tumor datasets, respectively. Fig. 2 shows that the number of genes can be reduced, which are graphically illustrated. For the

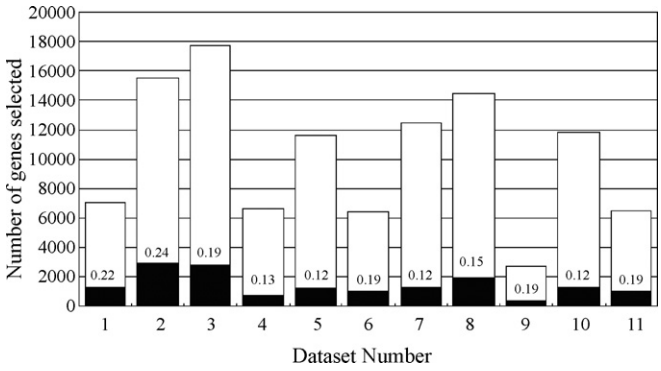

Fig. 2. Number of genes (features) selected for each of the 11 datasets (numbers on the bars indicate the percentage of selected genes).

Leukemia1, SRBCT, and DLBCL datasets, the proposed method reached 100% classification accuracy even though the percentage of genes selected is reduced to 0.19, 0.12, 0.19, and 0.12. This means that not all features are necessary to achieve total classification accuracy.

Table 3 compares experimental results obtained by other methods from the literature and the proposed method. Non-SVM and MC-SVM results were taken from Statnikov et al. for comparison (Statnikov et al., 2004). Various methods were used compared our proposed method. They include: support vector machines: (1) one-versus-rest and one-versus-one (Kreßel, 1999), (2) DAGSVM (Platt et al., 2000), (3) the method by Weston and Watkins (Hsu and Lin, 2002; Weston and Watkins, 1999), and (4) the method by Crammer and Singer (Hsu and Lin, 2002; Crammer and Singer, 2000). The non-SVM methods include: the *K*-nearest neighbor

Table 1
Cancer-related human gene expression datasets used in this study

Dataset name	Description
9_Tumors	Oligonucleotide microarray gene expression profiles for the chemosensitivity profiles of 232 chemical compounds
11_Tumors	Transcript profiles of 11 common human tumors for carcinomas of the prostate, breast, colorectum, lung, liver, gastroescophagus, pancreas, ovary, kidney, and bladder/ureter
14_Tumors	Oligonucleotide microarray gene expression profiles of 14 human tumors, including breast adenocarcinoma, prostate adenocarcinoma, lung adenocarcinoma, colorectal adenocarcinoma, lymphoma, bladder transitional cell carcinoma, melanoma, uterine adenocarcinoma, leukemia, renal cell carcinoma, pancreatic adenocarcinoma, ovarian adenocarcinoma, pleural mesothelioma, and central nervous system
Brain_Tumor1	DNA microarray gene expression profiles derived from 99 patient samples. The medulloblastomas included primitive neuroectodermal tumors (PNETs), atypical teratoid/rhabdoid tumors (AT/RTs), malignant gliomas and the medulloblastomas activated by the Sonic Hedgehog (SHH) pathway
Brain_Tumor2	Transcript profiles of four malignant gliomas, including classic glioblastoma, nonclassic glioblastoma, classic oligodendroglioma, and nonclassic oligodendroglioma
Leukemia1	DNA microarray gene expression profiles of acute myelogenous leukemia (AML), acute lymphoblastic leukemia (ALL) B-cell and T-cell
Leukemia2	Gene expression profiles of a chromosomal translocation to distinguish mixed-lineage leukemia (MLL), acute lymphoblastic leukemia (ALL), and acute myelogenous leukemia (AML)
Lung_Cancer	Oligonucleotide microarray transcript profiles of 203 specimens, including lung adenocarcinomas, squamous cell lung carcinomas, pulmonary carcinomas, small-cell lung carcinomas, ans normal lung tissues
SRBCT	cDNA microarray gene expression profiles of small, round blue cell tumors, which include neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS)
Prostate_Tumor	cDNA microarray gene expression profiles of prostate tumors. Based on MUC1 and AZGP1 gene expression, the prostate cancer can be distinguished as a subtype associated with an elevated risk of recurrence or with a decreased risk of recurrence
DLBCL	DNA microarray gene expression profiles of diffuse large B-cell lymphoma (DLBCL), in which the DLBCL can be identified as cured versus fatal or refractory disease

Table 2
Format of gene expression classification data

Dataset number	Dataset name	Number of				Percentage of genes selected
		Samples	Categories	Genes	Genes selected	
1	9_Tumors	60	9	5726	1280	0.22
2	11_Tumors	174	11	12533	2948	0.24
3	14_Tumors	308	26	15009	2777	0.19
4	Brain_Tumor1	90	5	5920	754	0.13
5	Brain_Tumor2	50	4	10367	1197	0.12
6	Leukemia1	72	3	5327	1034	0.19
7	Leukemia2	72	3	11225	1292	0.12
8	Lung_Cancer	203	5	12600	1897	0.15
9	SRBCT	83	4	2308	431	0.19
10	Prostate_Tumor	102	2	10509	1294	0.12
11	DLBCL	77	2	5469	1042	0.19
Average						0.17

Table 3
Classification accuracies of gene expression data obtained via different classification methods

Datasets	Methods								IBPSO
	Non-SVM			MC-SVM					KNN
	KNN	NN	PNN	OVR	OVO	DAG	WW	CS	
9_Tumors	43.90	19.38	34.00	65.10	58.57	60.24	62.24	65.33	**78.33**
11_Tumors	78.51	54.14	77.21	94.68	90.36	90.36	94.68	**95.30**	93.10
14_Tumors	50.40	11.12	49.09	74.98	47.07	47.35	69.07	**76.60**	66.56
Brain_Tumor1	87.94	84.72	79.61	91.67	90.56	90.56	90.56	90.56	**94.44**
Brain_Tumor2	68.67	60.33	62.83	77.00	77.83	77.83	73.33	72.83	**94.00**
Leukemia1	83.57	76.61	85.00	97.50	91.32	96.07	97.50	97.50	**100.0**
Leukemia2	87.14	91.03	83.21	97.32	95.89	95.89	95.89	95.89	**100.0**
Lung_Cancer	89.64	87.80	85.66	96.05	95.59	95.59	95.55	**96.55**	**96.55**
SRBCT	86.90	91.03	79.50	100.0	100.0	100.0	100.0	100.0	**100.0**
Prostate_Tumor	85.09	79.18	79.18	92.00	92.00	92.00	92.00	92.00	**92.16**
DLBCL	86.96	89.64	80.89	97.50	97.50	97.50	97.50	97.50	**100.0**
Average	77.16	67.73	72.38	89.44	85.15	85.76	88.03	89.10	**92.29**

Legends: (1) Non-SVM: traditional classification method. (2) MC-SVM: multi-class support vector machines. (3) KNN: *K*-nearest neighbors. (4) NN: backpropagation neural networks. (5) PNN: probabilistic neural networks. (6) OVR: one-versus-rest. (7) OVO: one-versus-one. (8) DAG: DAGSVM. (9) WW: method by Weston and Watkins. (10) CS: method by Crammer and Singer. (11) IBPSO: improved binary particle swarm optimization.

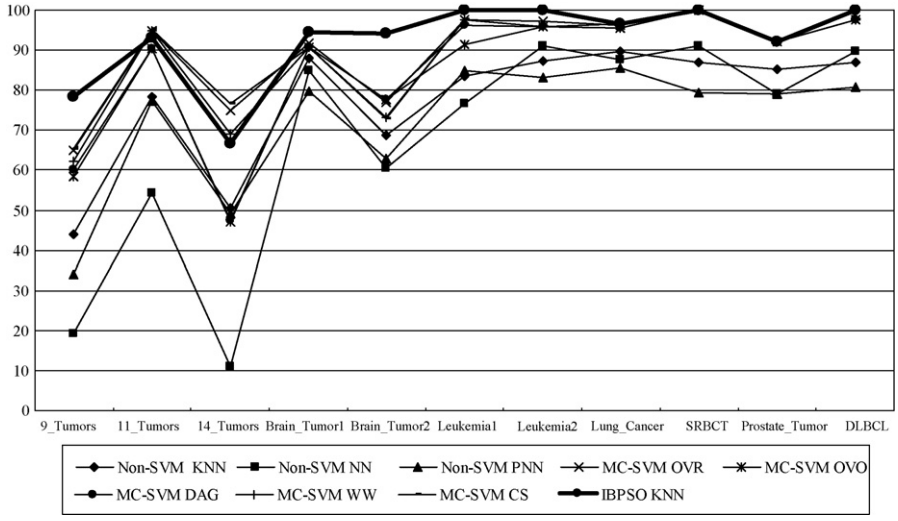

Fig. 3. Graphical comparison of classification accuracies obtained via different methods.

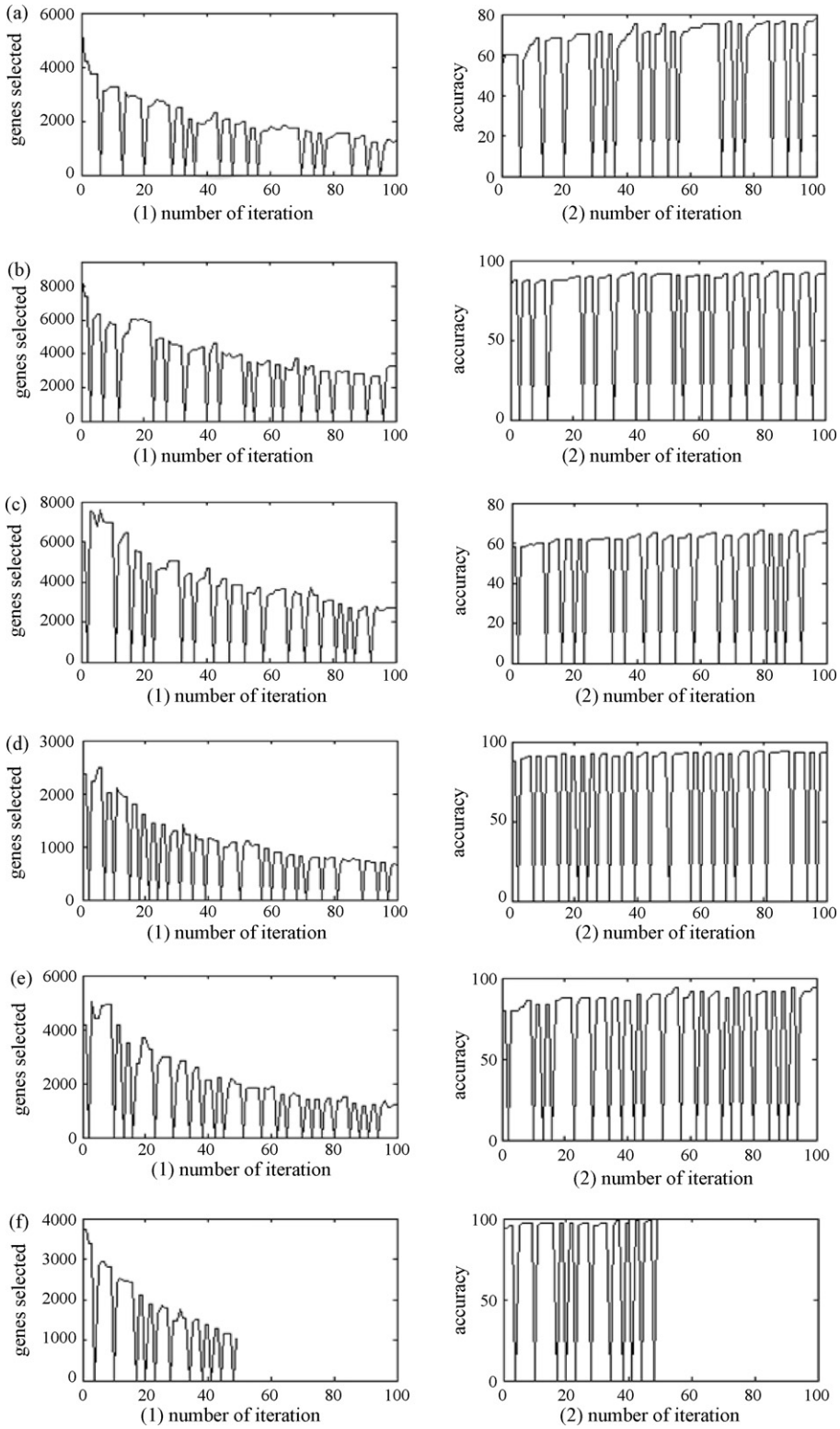

Fig. 4. (a) 9_Tumors data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (b) 11_Tumors data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (c) 14_Tumors data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (d) Brain_Tumor1 data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (e) Brain_Tumor2 data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (f) Leukemia1 data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (g) Leukemia2 data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (h) Lung_Cancer data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (i) SRBCT data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (j) Prostate_Tumor data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2), (k) DLBCL data set—number of iterations vs. genes selected (1) and number of iterations vs. accuracy (2).

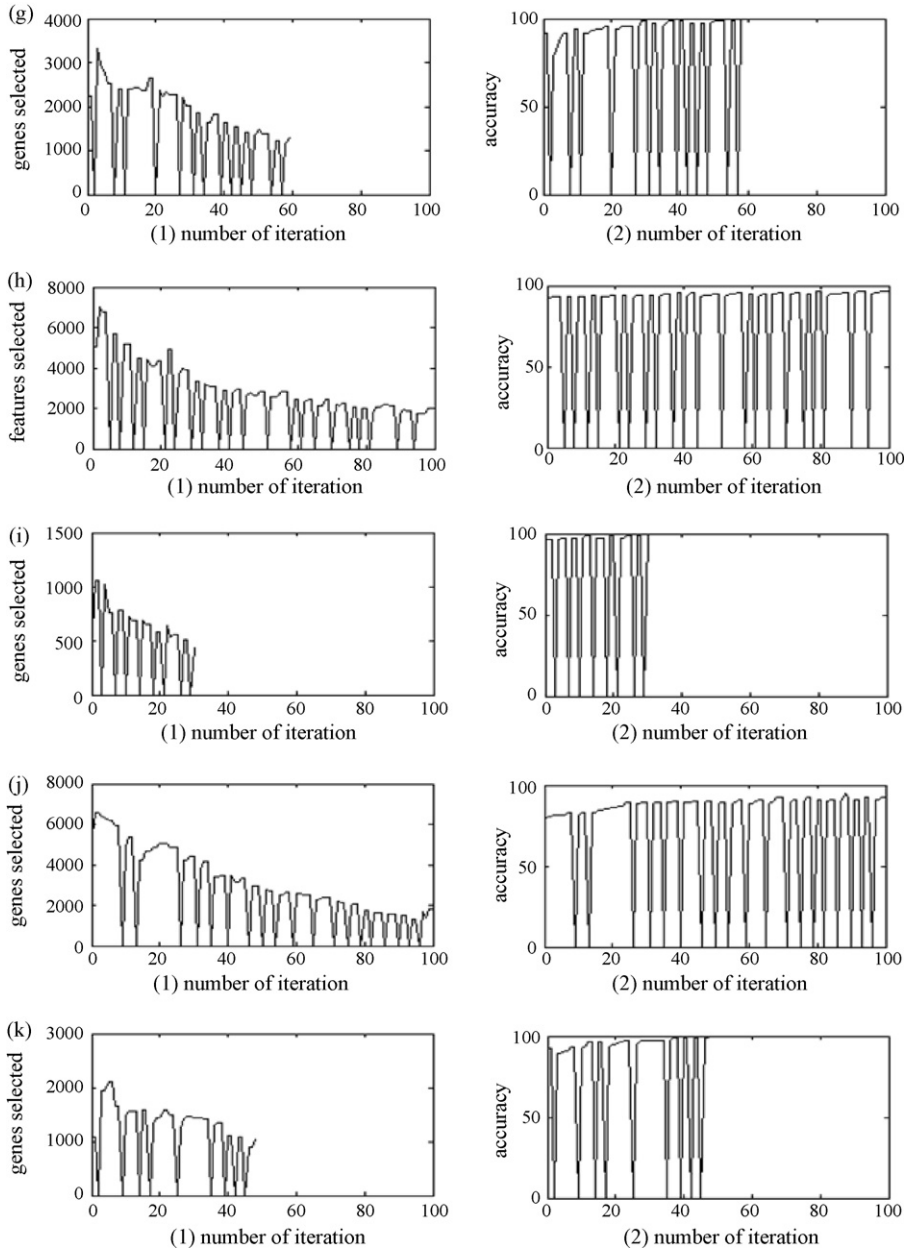

Fig. 4. (*Continued*).

method (Dasarathy, 1991; Cover and Hart, 1967), backpropagation neural networks (Mitchell, 1997), and probabilistic neural networks (Specht, 1990). The average highest classification accuracy of non-SVM, MC-SVM, and the proposed method is 77.16, 89.44, and 92.29, respectively. The proposed method obtained nine of the highest classification accuracies for the 11 test datasets, i.e. for the 9_Tumors, Brain_Tumor1, Brain_Tumor2, Leukemia1, Leukemia2, Lung_Cancer, SRBCT, Prostate_Tumor, and DLBCL data sets. The classification accuracy of the 9_Tumors and Brain_Tumor2 data sets obtained by the proposed method are 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-SVMs and MC-SVMs method. For the 11_Tumors and 14_Tumors datasets, the classification accuracy obtained by the proposed method is better than the classification accuracy of Non-SVMs and is comparable to the MC-SVM method. Fig. 3 gives a graphical comparison of classification accuracies obtained via different methods.

Fig. 4a–k shows the number of iterations versus genes (features) selected and the number of iterations versus classification accuracy for the tested datasets. The grey line represents the actually measured values. The grey line zigzag shape shows exactly at which iterations the *gbest* value had to be reset to zero, i.e. had reached a local optimum. For the Leukemia1 (Fig. 4f), Leukemia2 (Fig. 4g), SRBCT (Fig. 4i), DLBCL (Fig. 4k) dataset, the proposed method obtained 100% classification accuracy before reaching the maximum number of iterations. Fig. 4f and g shows that the number of features selected converges at an early stage. However the classification accuracy keeps improving and finally reaches 100%. This

illustrates that in a case where the number of genes selected is identical, the classification accuracy can still be improved since the actual genes selected can be different even if the total number selected is the same. For this reason, good gene selection should not only decrease the number of genes, but also identify the ones that help improve classification accuracy. The number of genes selected by SRBCT (Fig. 4i) keeps decreasing while the number of iterations keeps increasing, and then the proposed method obtained 100% classification accuracy. This demonstrates that although the number of genes selected is lower, classification accuracy does not have to be negatively affected; as long as the chosen genes contain enough feature classification information, better classification accuracy can be achieved.

The relatively low classification accuracy obtained for the tumor_14 sample (66.56%) obtained by our method can be explained by the very large number of classification categories involved (26 categories). This number is more than twice as high as for any other sample, the second highest number of classification categories being 11. The large number of categories involved proved to be detrimental to the classification accuracy. Still, most mother methods obtained and even lower accuracy for this sample; only two methods: the CS method and the WW method achieved a higher accuracy (76.60% and 69.07%, respectively).

GAs had been shown to outperform SFS (sequential forward search), PTA (plus and take away) and SFFS (sequential forward floating search) in Oh et al. (Oh et al., 2004). PSO shares many similarities with evolutionary computation techniques like GAs. PSO is based on the idea of collaborative behavior and swarming in biological populations. Both PSO and GAs are population-based search approaches that depend on information sharing among their population members to enhance the search processes by using a combination of deterministic and probabilistic rules. However, PSO does not include genetic operators such as crossover and mutation. The recognition and social model of interaction between particles is similar to crossover. For example, the random parameters rand_1 and rand_2 (in Eq. (1)) will affect the speed of a particle, similar to the mutation in a GA. In fact, the only difference between both is that the crossover and mutation in a GA is probabilistic (crossover rate and mutation rate), but the renewed particle in PSO should be processed at each iteration without any probability. Compared with GAs, the information sharing mechanism in PSO is considerably different. In GAs, the evolution is generated by using crossover and mutation in the same population. Chromosomes share information with each other, so the whole population moves like one group towards an optimal area. In the problem space, this model is similar to searching for only one area. Therefore, the drawback of this model is that it can become easily trapped in a local optimum. Although mutation is used, the probability usually is lower. Therefore, the improved performance is limited. In PSO, each particle is uniformly distributed in the problem space. But only *gbest* provides information to other particles. It is a one-way information sharing mechanism. Evolution only looks for the best solution. In most cases all the particles tend to converge towards the best solution quickly even in the local version.

Compared to GAs, the PSO has a much more profound intelligent background and can be performed more easily (Shi et al., 2005). The performance of PSO is affected by the parameter settings, inertia weight w, and the acceleration factors c_1 and c_2. However, if the proper parameter values are set, the results can easily be optimized. Proper adjustment of the inertia weight w and the acceleration factors c_1 and c_2 is very important. If the parameter adjustment is too small, the particle movement is too small. This scenario will also result in useful data, but is a lot more time-consuming. If the adjustment is excessive, particle movement will also be excessive. This will cause the algorithm to weaken early, so that a useful feature set cannot be obtained. Hence, suitable parameter adjustment enables particle swarm optimization to increase the efficiency of feature selection.

4. Conclusions

In this paper, improved binary particle swarm optimization (IBPSO) is used to implement a feature selection, and a *K*-nearest neighbor (*K*-NN) serves as an evaluator of IBPSO for gene expression data classification problems. Experimental results show that our method effectively simplified gene (feature) selection and reduced the total number of genes (features) needed. The classification accuracy obtained by the proposed method was the highest nine out of the 11 gene expression data test problems, and is comparative to the classification accuracy of the other two test problems. The average classification accuracy for the proposed method was increased by 2.85% compared to the best results of the previously published. The proposed method can serve as an ideal pre-processing tool to help optimize the feature selection process, since it increases the classification accuracy and, at the same time, keeps computational resources needed to a minimum. It could also be applied to problems in other areas in the future.

Acknowledgements

This work is partly supported by the National Science Council in Taiwan under grants NSC94-2622-E-151-025-CC3, NSC94-2311-B037-001, NSC93-2213-E-214-037, NSC92-2213-E-214-036, NSC92-2320-B-242-004, NSC92-2320-B-242-013, and by the CGMH fund CMRPG1006.

References

Ancona, N., Maglietta, R., D'Addabbo, A., Liuni, S., Pesole, G., 2005. Regularized least squares cancer classifiers from DNA microarray data. Bioinformatics 6, S2.

Berrar, D., Bradbury, I., Dubitzky, W., 2006. Instance-based concept learning from multiclass DNA microarray data. Bioinformatics 7, 73.

Cover, T., Hart, P., 1967. Nearest neighbor pattern classification. In: Proceedings of the IEEE Transactions Information Theory, pp. 21–27.

Crammer, K., Singer, Y., 2000. On the learnability and design of output codes for multiclass problems. In: Proceedings of the Thirteen Annual Conference on Computational Learning Theory (COLT 2000), Stanford University, Palo Alto, CA, June 28–July 1.

Dasarathy, B.V., 1991. In: Dasarathy, B.V. (Ed.), NN Concepts and Techniques, Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, pp. 1–30.

Diaz-Uriarte, R., Alvarez de Andres, S., 2006. Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7, 3.

Fix, E., Hodges, J.L., 1951. Discriminatory Analysis—Nonparametric Discrimination: Consistency Properties. Technical Report 21-49-004, Report no. 4, US Air Force School of Aviation Medicine, Randolph Field, pp. 261–279.

Hsu, C.-W., Lin, C.-J., 2002. A comparison of methods for multi-class support vector machines. IEEE Trans. Neural Netw. 12, 415–425.

Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, Perth, Australia, pp. 1942–1948.

Kennedy, J., Eberhart, R.C., 1997. A discrete binary version of the particle swarm algorithm. Systems, Man, and Cybernetics, 1997. In: Proceedings of the IEEE International Conference on Computational Cybernetics and Simulation, vol. 5, October 12–15, pp. 4104–4108.

Kennedy, J., Eberhart, R.C., Shi, Y., 2001. Swarm Intelligence. Morgan Kaufman, San Mateo, CA.

Kreßel, U., 1999. Pairwise classification and support vector machines. In: Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge, MA, pp. 255–268.

Liu, X., Krishnan, A., Mondry, A., 2005. An entropy-based gene selection method for cancer classification using microarray data. BMC Bioinformatics 6, 76.

Mitchell, T.M., 1997. Machine Learning. McGraw-Hill, New York, NY, USA.

Narendra, P.M., Fukunage, K., 1997. A branch and bound algorithm for feature subset selection. IEEE Trans. Comput. 6 (9), 917–922.

Oh, et al., 2004. Hybrid genetic algorithm for feature selection. IEEE Trans. Pattern Anal. Mach. Intell. 26 (11), 2004, Nov.

Palau, A.M., Snapp, R., 1998. The labeled cell classifier: a fast approximation to k nearest neighbors. In: Proceedings of the Fourteenth International Conference on Pattern Recognition, 1, pp. 823–827.

Platt, J.C., Cristianini, N., Shawe-Taylor, J., 2000. Large margin dags for multiclass classification. In: Advances in Neural Information Processing Systems 12. MIT Press, pp. 547–553.

Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognit. Lett. 15, 1119–1125.

Raymer, M.L., Punch, W.F., Goodman, E.D., Kuhn, L.A., Jain, A.K., 2000. Dimensionality reduction using genetic algorithms. IEEE Trans. Evol. Comput. 4 (2), 164–171.

Roberto, B., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Trans. Neural Netw. 5 (4), 537–550.

Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, L.M., 2005. An improved ga and a novel pso-ga-based hybrid algorithm. Inf. Process. Lett. 93 (5), 255–261.

Shi, Y., Eberhart, R.C., 1998. A Modified Particle Swarm Optimizer. IEEE International Conference on Evolutionary Computation. Anchorage, Alaska, pp. 69–73.

Specht, D.F., 1990. Probabilistic neural network. Neural Netw. 3, 109–118.

Stacey, A., Jancic, M., Grundy, I., 2003. Particle swarm optimization with mutation. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2003), Canbella, Australia, pp. 1425–1430.

Statnikov, A., Aligeris, C.F., Tsamardinos, L., Hardin, D., Levy, S., 2004. A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21 (5), 631–643, Sep.

Tang, E.K., Suganthan, P., Yao, X., 2006. Gene selection algorithms for microarray data based on least squares support vector machine. Bioinformatics 7, 95.

Weston, J., Watkins, C., 1999. Support vector machines for multi-class pattern recognition. In: Proceedings of the Seventh European Symposium on Artificial Neural Networks (ESANN 99), Bruges, pp. 21–23.

Yang, J.H., Honavar, V., 1998. Feature subset selection using a genetic algorithm. IEEE Intell. Syst. 13 (2), 44–49.

Yu, B., Yuan, B., 1993. A more efficient branch and bound algorithm for feature selection. Pattern Recognit. 26 (6), 883–889.

Zhang, H., Sun, G., 2002. Feature selection using tabu search method. Pattern Recognit. 35, 701–711.