Applied Soft Computing 62 (2018) 203–215

Contents lists available at ScienceDirect

Applied Soft Computing

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

Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification

Indu Jain [a], Vinod Kumar Jain [b,*], Renu Jain [a]

[a] School of Mathematics and Allied Sciences (SOMAAS), Jiwaji University, Gwalior, M.P. 474006, India
[b] PDPM-Indian Institute of Information Technology, Design and Manufacturing, Jabalpur, Dumna Airport Road, P.O. Khamaria, Jabalpur, M.P., India

ARTICLE INFO

Article history:
Received 6 May 2016
Received in revised form 27 August 2017
Accepted 26 September 2017
Available online 12 October 2017

Keywords:
Microarray data analysis
Cancer classification
Improved Binary Particle Swarm Optimization (iBPSO)
Hybrid model
Gene selection
Naive–Bayes

ABSTRACT

DNA microarray technology has emerged as a prospective tool for diagnosis of cancer and its classification. It provides better insights of many genetic mutations occurring within a cell associated with cancer. However, thousands of gene expressions measured for each biological sample using microarray pose a great challenge. Many statistical and machine learning methods have been applied to get most relevant genes prior to cancer classification. A two phase hybrid model for cancer classification is being proposed, integrating Correlation-based Feature Selection (CFS) with improved-Binary Particle Swarm Optimization (iBPSO). This model selects a low dimensional set of prognostic genes to classify biological samples of binary and multi class cancers using Naive–Bayes classifier with stratified 10-fold cross-validation. The proposed iBPSO also controls the problem of early convergence to the local optimum of traditional BPSO. The proposed model has been evaluated on 11 benchmark microarray datasets of different cancer types. Experimental results are compared with seven other well known methods, and our model exhibited better results in terms of classification accuracy and the number of selected genes in most cases. In particular, it achieved up to 100% classification accuracy for seven out of eleven datasets with a very small sized prognostic gene subset (up to <1.5%) for all eleven datasets.

1. Introduction

Parallel measurement of thousands of gene expressions using DNA microarray provides a big picture and better insights of many genetic alterations pertaining to cancer [1]. An early and accurate prognosis of cancer facilitates the proper line of treatment, and DNA microarray technology has shown great potential in diagnosis of cancer and its classification. The cancer datasets produced by microarray technology typically have thousands of gene expressions obtained from each biological sample. Furthermore, the number of samples are very less in comparison to gene expressions. These characteristics of cancer microarray data pose a great difficulty in analysis and its classification. However, only a few genes from these high dimensional datasets are significant for cancer classification [2]. The presence of redundant, irrelevant and noisy genes in the dataset degrades the computing efficiency as well as the classification accuracy of machine learning algorithms, particularly when samples are limited. Therefore, it becomes indispensable to alleviate irrelevant and redundant genes from the dataset using some feature selection methods [3].

Numerous methods are in use for feature (gene) selection which may be grouped into two categories: filters and wrappers. Filter method searches and evaluates either each gene individually (univariate filters) or the subset of genes (multivariate filters) by measuring their intrinsic properties related to class discrimination, independent of a learning method.

The univariate filter methods evaluate each gene separately and give less reliable outcomes due to ignorance of gene interactions. Therefore, a multivariate feature selection method would be a better choice in comparison to a univariate method to identify the most prognostic genes in microarray data. Correlation-based Feature Selection (CFS) is one of the multivariate filter approaches, which selects a good feature subset containing features those have strong correlations with target class, yet uncorrelated mutually. However, CFS may not select a good subset of genes if the values of the gene expressions lie in the small area of search space [4].

Moreover, Wrapper method encapsulates a global search method and the classifier in a single approach. The search method explores the gene space for all possible gene subsets, and the goodness of each subset is evaluated by a specific classifier for sample classification. Wrappers are further classified according to the

* Corresponding author.
E-mail addresses: indujain06@gmail.com (I. Jain), vkjain@iiitdmj.ac.in (V.K. Jain), renujain3@rediffmail.com (R. Jain).

https://doi.org/10.1016/j.asoc.2017.09.038
1568-4946/

search methods used in the approach, as deterministic and randomized wrappers [5,6]. Over the past few years, many wrapper models based on Particle Swarm Optimization (PSO) for gene selection have been applied to determine prognostic genes from microarray data [7–10]. PSO is a popular meta-heuristic based stochastic global search approach. It uses swarm intelligence to find the solution to an optimization problem by generating better candidate feature subsets according to a given fitness criteria and can converge quickly toward a global optimum in a fixed number of iterations [11,12]. In PSO, a swarm of particles (candidate solutions) navigates in the accessible search space with a controlled velocity. The velocity of the particle can be controlled on the basis of its acquired knowledge and the globally acquired knowledge by the whole swarm, to find a near-optimal solution. After each generation, every particle updates its velocity and hence its position. Kennedy and Eberhart [13] proposed a binary version of PSO (BPSO) for discrete problems in which every member (particle) denotes its position in form of a binary string having either the binary digit 0 or 1, and the sigmoid probability function is used for velocity to assign a new status to every particle.

For microarray datasets, filters have the advantages of less computational complexity and faster execution time over wrappers. However, wrappers provide more trustworthy results than filters that maximize the classification accuracy, but they suffer from the limitations of slow computation and greater search complexity due to very high dimensionality of the dataset [14–16].

The main objective of this work is to develop a classification model to predict the class of unknown test samples accurately. The key issue in designing this model is the selection of the optimal set of most relevant and non-redundant genes to maximize the classification accuracy, and to reduce the computational cost. The predictive gene selection is very difficult due to the large gene space and the complex interactions among genes. So, here we propose a classification model CFS-iBPSO ($\uparrow w$)-NB based on hybrid framework of filter and wrapper approaches, in which a pre-filtering step is performed before using the wrapper method to make cutbacks in the computational cost and search complexity. The proposed model combines the faster computational ability of a multivariate filter CFS (Correlation-based Feature Selection) and the efficiency to produce better and reliable classification accuracy of a wrapper model iBPSO-NB (improved Binary Particle Swarm Optimization-Naive–Bayes) in a single approach. The model attempts to select most relevant and non-redundant subset of predictor genes to determine the class of a unknown tissue sample in two phases. First, CFS selects a good and effectively small subset of predictive genes. In the subsequent phase this subset is used as an initial point for the iBPSO-NB wrapper, which further optimizes the predictive gene subset and improves the classification accuracy. The wrapper method uses Naive–Bayes (NB) [17–19] probabilistic learner along with stratified 10-fold cross validation to assess the classification accuracy of each candidate gene subset. However, the problem of moving around a local optimum is very common in BPSO like other Evolutionary Computation (EC) techniques, which hinders the algorithm to obtain an optimal solution globally. We present a solution for the problem of early convergence at a local optimum in an improved Binary Particle Swarm Optimization (iBPSO) by replacing the old global best in a manner, so that all the particles can move to explore the gene search space for getting a global optimum. Some key features of the proposed hybrid framework are highlighted as follows:

- Our proposed CFS-iBPSO ($\uparrow w$)-NB classification model employs a simple and faster yet computationally efficient improved BPSO (iBPSO) to solve the problem of early convergence toward a local optimum. An increasing inertia weight scheme ($\uparrow w$) [20] is applied herein which controls the searching capability of the iBPSO algorithm in such a way that significantly improves the performance of the algorithm.
- The proposed hybrid framework uses more compatible NB learner as wrapped classifier with the multivariate filter CFS which share similar hypothesis about the discriminative features with NB, to improve the classification accuracy [4].
- To analyze the performance of our proposed hybrid model, we use the stratified 10-fold cross validation which is known for the best performance with lower bias and lower variance [21,22] in the estimation of test accuracy. Thus our proposed model provides more reliable and robust results of the classification accuracy and the number of selected genes.

We examined and compared our proposed system with seven other well known methods on 11 benchmark microarray datasets for various cancers. Experimental results show that the model achieved better or comparable performance against seven other tested methods. The rest of the paper is organized as follows. Section 2 presents review of some related work. The CFS-iBPSO-NB algorithm for gene selection and cancer classification is discussed in Section 3. Section 4 describes the microarray datasets, parameter settings and experimental results. Finally we conclude the paper in Section 5.

2. Related work

One of the major objectives of DNA microarray analysis is to provide a generic approach for the cancer classification. In the past decades, several statistical and machine learning models have been implemented to contribute in this area. These models can be divided into unsupervised (class discovery/clustering) and supervised (class prediction/classification) learning models.

In unsupervised models, a hierarchical clustering algorithm is used by Perou et al. [1] to identify different gene expression patterns that described the phenotypic diversity of breast tumors. Similarly a two-way clustering is applied to analyze colon tumor [23]. These prominent results showed that specific cell types exhibits a systematic pattern of gene expression and variations in these patterns reveal important aspects of biological variations in cancer. Golub et al. [2] used self-organizing maps (SOMs) to cluster tumors. They also classify human acute leukemia samples by using weighted voting on the basis of 50 informative genes.

In the field of supervised classification, traditional statistical methods (discriminant analysis, Gaussian and logistic classifiers, etc.) [24–26], and various machine learning techniques like Support Vector Machines (SVMs) [27–29], Neural networks (ANNs) [30,31], *K*-nearest neighbor (*K*-NN)[2,25,32] are the most widely applied class prediction techniques in cancer classification.

The very characteristic of microarray data is high dimensionality (very large number of genes) and small number of tissue samples, make it essential to workout gene selection (feature/attribute selection) for achieving better classification accuracy and making the analysis process fast and cost efficient. A wide range of gene selection methods have been implemented to select meaningful genes and removing irrelevant and redundant features prior to cancer classification.

The literature abound in feature selection metrics of different nature for example univariate filter metrics *P*-metric [2,30] and *t*-score [33] are commonly used in microarrays. However they are less efficient than multivariate models, some of them are reported in literature as Correlation-based Feature Selection (CFS) [4], Fast Correlation-Based Filter (FCBF) [34], Minimum Redundancy-Maximum Relevance (mRMR) [35], Uncorrelated Shrunken Centroid (USC) [36] algorithm, ReliefF [37], etc.

Wrapper approaches perform better in terms of feature selection by evaluating the classification accuracy using some induction algorithm. For microarray data, with high dimensionality and small data points these methods suffer from greater computational cost and risk of overfitting. Some influential studies implemented randomized and population based wrappers like Particle Swarm Optimization (PSO) [7–10] and Genetic Algorithms (GA) [32,38,39] for gene selection purpose.

In recent years, a hybrid framework has been reported in many studies for gene selection which combines the superior performance of wrapper models and alleviating the problems of high computational cost and overfitting by pre-treatment with a filter algorithm and providing prominent results in cancer classification. Ruiz et al. [40] presented a hybrid algorithm BIRS (best incremental ranked subset) for gene selection which effectively reduce the number of genes but the classification accuracy were low for all four microarray datasets. Zhu et al. [41] reported significant results for 11 microarray datasets by using Markov blanket-embedded genetic algorithm (MBEGA) for gene selection. Shen et al. [7] incorporated tabu search and PSO in a hybrid framework on microarray datasets but the results were not much satisfactory. Li et al. [8] demonstrated a new hybrid of PSO and Genetic Algorithms (GA) but the resulted accuracy were low with a high number of genes selected. Further Chuang et al. [42] applied gene selection and cancer classification using Taguchi chaotic based binary PSO and reported lower classification error rates for many datasets, however the method is cumbersome and takes more time to execute. For microarray data which have large number of genes and small number of data points, these methods suffer from greater computational cost and more risk of overfitting.

We also propose a hybrid model for gene selection and cancer class determination which attempts to address the drawbacks mentioned above. The key aspect of the model is that it integrates the benefits of fast and efficient dimensionality reduction of multivariate filter (CFS) and simple yet powerful Binary Particle Swarm Optimization approach. Our hybrid model operates in two phases. In the first phase gene selection is performed to select a subset of genes by CFS, and in the following phase gene optimization is done on the gene subset obtained from the first phase by implementing our iBPSO-NB wrapper method. The iBPSO also eliminates the problem of premature local convergence of traditional BPSO. The main objective of the paper is to achieve better classification accuracy using fewer number of highly predictive genes.

3. Proposed cancer classification system

3.1. System model

DNA microarray experiments measure thousands of gene expressions for tissue samples and these data are stored in the form of microarray data matrix. It is well known that variations in the systematic patterns of gene expressions exhibited by a specific cell type is correlated with the biological variations of a particular cancer type [1]. Let M genes form M-dimensional gene expression space E^M corresponding to a sample space U then this association can be represented mathematically as

$$f : E^M \rightarrow U$$

More specifically, there are l target classes for all the given samples then there exist an association between gene expression patterns and the ith class which can be defined as

$$C^{(i)} = f(\mathcal{S}_j^M)$$

where $C^{(i)} \in U$ for $1 \leq i \leq l$ and $\mathcal{S}_j^M \in E^M$. Typically the gene expression vector $\mathcal{S}_j^M$ is an ordered sequence tuple with respect to the M genes for the jth $(1 \leq j \leq N)$ sample

$$\mathcal{S}_j^M = (e_1, e_2, e_3, \ldots, e_M)$$

in which each gene expression value denotes a feature. Suppose the whole microarray dataset $\mathcal{D}$ consists of N observations (samples) which can be represented as

$$\mathcal{D} = \{(\mathcal{S}_j^M, C^{(i)}) : j = 1, 2, \ldots, N, i \in (1, 2, \ldots l)\} \subseteq E^M \times U$$

The vector $\mathcal{S}_j^M (j = 1, 2, \ldots, N)$ consists of gene expression values for M number of genes and $C^{(i)} \in \{1, 2, 3, \ldots, l\}$ is the class label assigned to the gene expression vector.

The main aim of this task is to develop a classification model which can predict the class labels for the given samples on the basis of a low-dimensional set of most relevant and non-redundant genes.

3.2. The detailed CFS-iBPSO-NB method

Suppose there are N biological tissue samples in the microarray dataset and a row vector in the dataset is given as $(\mathcal{S}_j^M, C^{(i)})$ where $\mathcal{S}_j^M \in E^M$ denotes the M observed gene expression values for the jth $(j = 1, 2, \ldots, N)$ sample and $C^{(i)} \in \{1, 2, 3, \ldots, l\}$ denotes the class label associated with $\mathcal{S}_j^M$ gene expression profile, here the M is very large for N sample $(M \gg N)$.

As the dimensionality M of the microarray dataset is very large, and also consists of irrelevant, redundant and noisy genes. Furthermore not all genes except a few important ones participate in the sample classification process. Thus the high dimensionality of the dataset degrades the computing efficiency of many learning algorithms. Therefore, some feature selection techniques are required to reduce the irrelevant and redundant genes from the dataset.

In this study, our key objective is to develop an efficient classification model based on the small dimensional gene subset that determines the class of a given biological sample with higher accuracy. In order to meet the requirement of higher accuracy and lower computational efforts, we propose a hybrid model of filter and wrapper approaches where a multivariate filter CFS (Correlation-based Feature Selection) effectively reduces the large gene space and provides a good initial gene subset to the iBPSO (improved Binary Particle Swarm optimization). The discriminative efficiency of each and every gene subset searched by iBPSO, is then assessed by Naive–Bayes (NB) classifier on the basis of cross-validated misclassification error. The pre-filtering by CFS overcomes the limitations of slow computation and high search complexity of iBPSO in our case.

Fig. 1 and Algorithm 1 show the overall structure and pseudocode of the CFS-iBPSO-NB approach respectively.

The proposed CFS-iBPSO-NB method works in two phases:

(1) Predictive Gene Pre-Filtering (PGPF) phase: In this phase, Correlation-based Feature Selection (CFS) choose a low dimensional subset of genes which have better predictive efficiency toward a class, by excluding redundant and irrelevant genes.
(2) Gene Optimization and Cancer Classification (GOCC) phase: A wrapper model iBPSO-NB is designed in which iBPSO choose optimal set of most relevant and non redundant genes from the gene subset found in PGPF phase, by evaluating the classification efficiency of each gene subset using Naive–Bayes (NB) learner with stratified 10-fold cross-validation.

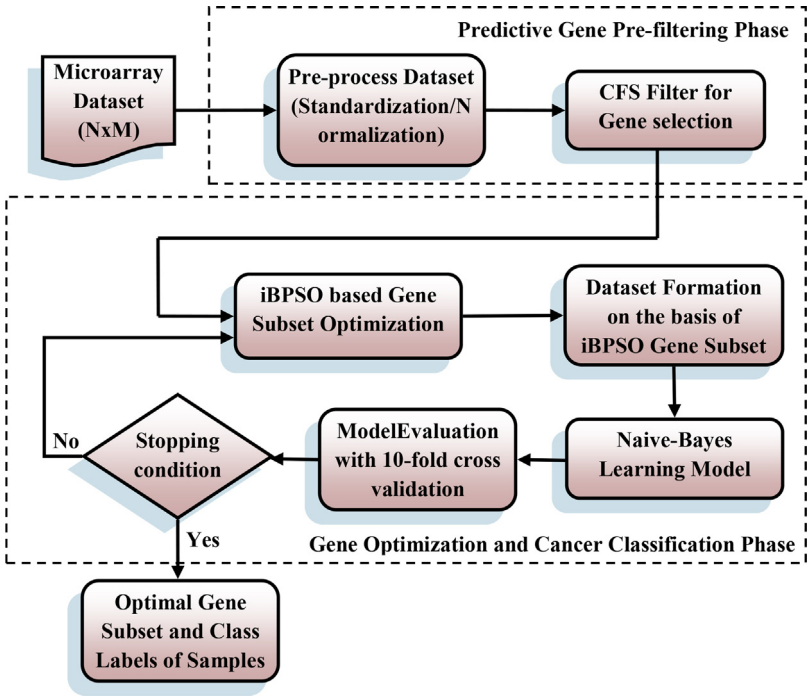


Fig. 1. A unified view of the CFS-iBPSO-NB method.

3.2.1. Predictive Gene Pre-Filtering (PGPF) phase

As the microarray data matrices are characterized by high dimensionality and small number of samples. Thus identifying marker genes, prior to the classification of cancer is one of the essential steps in microarray data analysis. Cancer classification is mainly affected by the fact that how to find out most relevant and small subset of genes needed to detect disease with high accuracy.

In this phase, we start with a data preprocessing step where in each dataset, we replaced the missing values of gene expressions with mean values and the whole dataset is standardized to have a mean value equal to zero and standard deviation equal to one by using following equation:

$$e_{new} = \frac{e - \mu}{\sigma} \tag{1}$$

Here μ is the mean value and σ is the standard deviation for the given vector. e_{new} is the transformed value for a gene expression e.

In order to reduce the high dimensionality of gene space we apply Correlation-based Feature Selection (CFS) [4] algorithm which selects feature subset of reduced dimensionality from a given feature space by including only those features which are mutually uncorrelated but have greater predictive ability toward a class from a given sample space. If the features are mutually uncorrelated then redundancy is eliminated whereas the greater relevancy of the features with class ensures better prognostic ability.

Let the dataset has N gene expression vectors $\vec{S}_j (j = 1, 2, \ldots, N)$ of gene expressions, each of dimensionality M. And let $\mathcal{G}$ is a gene set consists of M genes for which, expression values are provided in every sample. By using best-first search heuristic, CFS explores the given gene set $\mathcal{G}$ for a gene subset $\mathcal{G}_1$ of reduced dimensionality $m\,(m \leq M)$. The subset $\mathcal{G}_1$ contains highly relevant and non-redundant predictor genes only.

In this search heuristic, initially the set contains no genes, then by taking single gene at a time search moves in forward direction. The basic concept of CFS filter is to find such subsets of genes in which each member shows high correlation with one of the known classes $C^{(i)} \in \{1, 2, 3, \ldots, l\}$ (i.e. better predictive capability) but less correlated with other members of the subset (i.e. lower redundancy of features). To assess the worth of subsets, algorithm uses following correlation-based merit criteria for the iterative search process:

$$\mathbb{R}_{\mathcal{G}_1} = \frac{m\bar{\rho_{c,e}}}{\sqrt{m + (m-1)\bar{\rho_{e,e}}}} \tag{2}$$

Here $\mathbb{R}_{\mathcal{G}_1}$ is the rank value of the gene subset $\mathcal{G}_1$ of dimensionality m

$\bar{\rho_{c,e}}$ = mean gene-class correlation
$\bar{\rho_{e,e}}$ = mean gene-gene inter-correlation

where $e \in \mathcal{G}_1$ and $c \in C^{(i)}$, $i = 1, 2, \ldots, l$. A higher rank value indicates better prognostic efficiency of the subset of genes for the class (numerator) and lower redundancy among the genes in the subset (denominator). The search is finished after a limited number of iterations when no improvement is shown by the five consecutive gene subsets over the current one [4]. Symmetric uncertainty ($\mathcal{SU}$) measure is used to calculate the gene-class correlations and gene-gene inter-correlations as:

$$\mathcal{SU}(X_1, X_2) = 2 \times \left[\frac{\mathcal{IG}(X_1|X_2)}{\mathcal{H}(X_1) + \mathcal{H}(X_2)}\right]$$

where $\mathcal{IG}(X_1|X_2) = \mathcal{H}(X_1) + \mathcal{H}(X_2) - \mathcal{H}(X_1, X_2)$ is the Information Gain which measures the information gained about X_1 after observing X_2. Symmetric uncertainty ($\mathcal{SU}$) balances the influential behavior of information gain toward multi-valued attributes and normalizes its values to the range [0,1], where $\mathcal{H}(X_1)$ and $\mathcal{H}(X_1, X_2)$ are the entropy measures given as:

$$\mathcal{H}(X_1) = -\sum_{x_1 \in X_1} p(x_1)\log_2(p(x_1))$$

If X_1 attribute in the training data is partitioned according to some another attribute X_2 then the entropy measure is given as:

$$\mathcal{H}(X_1|X_2) = -\sum_{x_2 \in X_2} p(x_2) \sum_{x_1 \in X_1} p(x_1|x_2)\log_2(p(x_1|x_2))$$

Algorithm 1. Pseudo-Code of CFS-iBPSO-NB

Preprocess *dataset* $\mathcal{D} = \{(\vec{\mathcal{S}}_j^M, C^{(i)}) : j = 1, 2, \ldots, N, i \in (1, 2, \ldots l)\}$ by Eq. (1)
Apply CFS algorithm to produce reduced dataset D which consists of N gene expression vectors $\vec{S}_j (j = 1, 2, \ldots, N)$, each of reduced dimensionality m
Set iBPSO parameters and initialize a swarm (Ω) of W particles randomly, where each particle have an initial position $P_k^0 = (p_{k1}^0, p_{k2}^0, \ldots, p_{km}^0)$ and velocity $V_k^0 = (v_{k1}^0, v_{k2}^0, \ldots, v_{km}^0) \quad k = 1, 2, \ldots, W$
repeat
  Adjust w^t inertia weight component with Eq. (6) or Eq. (7)
  for all particle k in the Swarm **do**
Construct dataset D' for particle k, which consists of N gene expression vectors $\vec{s}_j (j = 1, 2, \ldots, N)$, each of reduced dimensionality m'
Divide dataset D' into 10 disjoint sets $D'_v, v = 1, 2, \ldots, 10$ of similar size
for all set D'_v of D' **do**
  Learn the classifier $NB_v : s \rightarrow C$ on the basis of $(D' - D'_v)$ by using Eq. (11)
  Calculate $\mathcal{E}_v(D'_v, \mathfrak{g}_i)$ by using Eq. (13)
end for
Compute cross-validated $\mathcal{E}(D', \mathfrak{g}_i)$ by using Eq. (12)
Revise the $p_{best,k}^{t+1}$ of particle k by using Eq. (8)
  end for
  Update g_{best}^{t+1} according to Eq. (9)
  if g_{best}^{t+1} does not change for three iterations **then**
Replace current g_{best}^{t+1} according to Eq. (10)
  end if
  for all particle k in the Swarm **do**
Update the velocity vector of kth particle according to Eq. (4)
Update the position vector of kth particle according to Eq. (5)
  end for
until Total (T) iterations are not done
return Optimal gene subset $\hat{\mathfrak{g}}_i$ according to Eq. (3)
return Minimum misclassification error $\mathcal{E}(D', \hat{\mathfrak{g}}_i)$ achieved by algorithm

3.2.2. Gene Optimization and Cancer Classification (GOCC) phase

The multivariate CFS filter effectively reduces the dimensionality of the gene space and provides the set $\mathcal{G}_1 (\subseteq \mathcal{G})$ of important genes which better represents the dataset. The reduced dataset D contains N gene expression vector $\vec{S}_j (\subseteq \vec{\mathcal{S}}_j)$ each of dimensionality $m\,(m \leq M)$.

Now we design a cancer classification model to further optimize the dimensionality of the gene subset $\mathcal{G}_1$, using a randomized wrapper approach improved-Binary Particle Swarm Optimization-Naive–Bayes (iBPSO-NB), for which the model achieves minimum misclassification error. The Naive–Bayes classifier(NB) is used to evaluate the predictive ability of the optimal subset of genes and, stratified 10-fold cross-validation is performed to reduce the risk of overfitting.

Specifically the objective function of this optimization problem can be modeled as:

$$\textbf{minimize}_{\mathfrak{g}_i \subseteq \mathcal{G}_1} \quad \mathcal{E}(D, \mathfrak{g}_i), \; where \quad \mathcal{G}_1 \subseteq \mathcal{G} \tag{3}$$

where $\mathcal{E}(D, \mathfrak{g}_i)$ is the misclassification error for Naive–Bayes classifier with stratified 10-fold cross-validation defined over a subset $\mathfrak{g}_i$ of most important genes and $\mathcal{G}_1$ is gene subset of m dimensions filtered by CFS. The objective is to find an optimal gene subset $\hat{\mathfrak{g}}_i$ such that $\mathcal{E}(D, \hat{\mathfrak{g}}_i) \leq \mathcal{E}(D, \mathfrak{g}_i), \quad \textbf{for}\text{all}\mathfrak{g}_i \subseteq \mathcal{G}_1$.

Particle Swarm optimization (PSO) is a randomized wrapper approach based on swarm-intelligence, introduced by Kennedy and Eberhart [11–13]. Binary PSO (BPSO) is a discrete version of original PSO. A swarm of particles move through a feature space to obtain an optimal solution for a given objective function in a limited number of iterations. Each particles's state in an iteration is defined by two values first its position in the search space and second the velocity value to modify the position. In each iteration, every particle (or a candidate solution) updates its current position and velocity based on the best positions obtained previously by the particle (p_{best}) and by the whole swarm (g_{best}).

Let the swarm (Ω) have W particles, where each particle represents a gene subset of m dimensions and T is the total number of iterations. The particle in the swarm is defined as a string of m binary values where a bit value 1 denotes the presence of that gene whereas a bit value 0 denotes absence of the gene in the subset. The position and velocity values of kth particle (i.e., gene subset) at tth iteration is defined as

$$P_k^t = (p_{k1}^t, p_{k2}^t, \ldots, p_{km}^t)\, k = 1, 2, \ldots, W; \quad t = 1, 2, \ldots, T$$

$$V_k^t = (v_{k1}^t, v_{k2}^t, \ldots, v_{km}^t)\, k = 1, 2, \ldots, W; \quad t = 1, 2, \ldots, T$$

The process starts with an initial population of particles say Ω_0 chosen randomly, then initial position and velocity of a particle in the population is given by

$$P_k^0 = (p_{k1}^0, p_{k2}^0, \ldots, p_{km}^0)\, k = 1, 2, \ldots, W$$

$$V_k^0 = (v_{k1}^0, v_{k2}^0, \ldots, v_{km}^0)\, k = 1, 2, \ldots, W$$

A new population Ω_{t+1} is produced from a present population Ω_t by flipping the bits of the positions of particles in the present population by using Eqs. (4) and (5). This modification is regulated by velocity value (i.e., probability measure) $v_{kd} (d \in m)$ which, in turn, is used to calculate the possible value of a dimension in position vector p_{kd} as a 0 or 1, using following formula:

$$v_{kd}^{t+1} = w^t \times v_{kd}^t + c_1 \times \psi_1^t \times (p_{best,k}^t - p_{kd}^t) + c_2 \times \psi_2^t \times (g_{best}^t - p_{kd}^t), \tag{4}$$

finally a logistic (sigmoid) function is used to define the change in the bit position of each particle in swarm as

$$p_{kd}^{t+1} = \begin{cases} 1 & \text{if } (Sig(v_{kd}^{t+1}) > \psi_3^t) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $Sig(v_{kd}^{t+1}) = 1/(1 + e^{-v_{kd}^{t+1}})$. w^t is the inertia weight component of the velocity value which controls the exploration of the search space in the current iteration. Out of many proposed inertia weight schemes, the decreasing inertia weight is commonly used in Continuous PSO. Recently, Liu et al. [20] analyzed the effect of increasing inertia weight scheme in BPSO. So, this work also analyzes the proposed iBPSO with both inertia weight schemes (increasing ($\uparrow w$) and decreasing ($\downarrow w$)). In linearly increasing scheme ($\uparrow w$) the inertia weight component w^t is calculated in each iteration t using following equation:

$$w^t = \begin{cases} w_{\min} + \dfrac{t.(w_{\max} - w_{\min})}{\xi \cdot T}, & \text{if } t \leq \xi \cdot T \\ w_{\max}, & \text{if } \xi \cdot T < t \leq T \end{cases} \tag{6}$$

Here the weight component w^t is increased linearly in each iteration and regulates the velocity of all particles. An initially small value of w^t, provides better exploration of the search space to all particles in swarm and gradually increasing value facilitates faster convergence (better exploitation) to the global solution. Here, $w_{\max}$ and $w_{\min}$ are the initial and final values of the inertia weight component respectively, t is the current iteration and T is total number of iterations. $0 \leq \xi \leq 1$ is the parameter to control the fraction of iterations where w^t increases linearly from $w_{\min}$ to $w_{\max}$. Here $\xi = 0.9$ is used to achieve stronger exploration in initial iterations and high exploitation in later iterations.

Similarly, in linearly decreasing scheme ($\downarrow w$) the inertia weight component w^t is calculated in each iteration t using following equation:

$$w^t = \begin{cases} w_{\max} - \dfrac{t.(w_{\max} - w_{\min})}{\xi \cdot T}, & \text{if } t \le \xi \cdot T \\ w_{\min}, & \text{if } \xi \cdot T < t \le T \end{cases} \quad (7)$$

where again $\xi = 0.9$ is used in the equation.

v_{kd}^t, v_{kd}^{t+1} and p_{kd}^t,p_{kd}^{t+1} are the velocity and position vectors of dth dimension for particle k in tth and $(t+1)$th iterations respectively, where v_{kd} are limited in the range $[v_{\min}, v_{\max}]$ in our case it is [−6,6].

c_1 and c_2 are positive acceleration constants which regulate the effect of particle's best and global best positions respectively; ψ_1^t, ψ_2^t and ψ_3^t are positive random values generated from a uniform distribution $U(0.0, 1.0)$ in tth iteration. $p_{best,k}^t$ is the personal best position of particle k refers to the position (i.e., gene subset) in the gene search space, where particle had the smallest misclassification error ($\mathcal{E}$) as determined by Eq. (12), found in process till iteration t. Whereas $g_{best,d}^t$ is the global best position refers to the position (gene subset) having minimum misclassification error ($\mathcal{E}$) amongst all the $p_{best,k}^t$ found by the swarm till iteration t.

The personal best position $p_{best,k}^t$ of kth particle can be updated for $t+1$ iteration as follows:

$$p_{best,k}^{t+1} = \begin{cases} p_{best,k}^t & \text{if } (\mathcal{E}(D, p_{kd}^{t+1}) \ge p_{best,k}^t) \\ p_{kd}^{t+1} & \text{otherwise} \end{cases} \quad (8)$$

and the global best position g_{best}^{t+1} for $(t+1)$th iteration is calculated as follows:

Let $p_{best,\min_k}^t$ is the position vector of the particle which has lowest $\mathcal{E}$ in tth iteration and $\mathcal{E}_{\min}^t$ is the minimum value of $\mathcal{E}$ obtained in tth iteration. Then

$$g_{best}^{t+1} = \begin{cases} g_{best}^t & \text{if } (\mathcal{E}_{\min}^t \ge \mathcal{E}(D, g_{best}^t)) \\ p_{best,\min_k}^t & \text{otherwise} \end{cases} \quad (9)$$

Note that when two particles are found having same misclassification error $\mathcal{E}$ in tth iteration, the one with smaller number of genes is selected by the algorithm.

Like other Evolutionary Computation (EC) techniques, standard BPSO also suffers from a common limitation that it converges to a local optimum prematurely which prevents the swarm from getting a global solution. Thus the g_{best}^t position shows no improvement over several generations of population. To overcome this problem, in iBPSO we propose a computationally efficient yet effective rule to update the g_{best}^t position in a manner so that all particles get a new direction to move toward the global best position. After a fixed number of generations if current g_{best}^t position retains the same value, then we update every dimension of it using following rule:

$$g_{best,d}^t = \begin{cases} 1 & \text{if } (P(p_{best,kd}^t = 1) \ge (P(p_{best,kd}^t = 0) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where $k = 1, 2, \ldots, W$ and $d = 1, 2, \ldots, m$.

$$P(p_{best,kd}^t = 1) = \frac{No.\ of\ p_{best}^t\ having\ 1's\ at\ d\text{th}\ bit.}{W}$$

The previous value of g_{best}^t is updated by this new g_{best}^t, such that all particles come out of a local stagnation, and explore the gene search space for a low dimensional gene subset and achieves a minimum misclassification error ($\mathcal{E}$). The proposed improved BPSO attains minimum misclassification error for an effectively reduced gene subset $\hat{g}_i$.

The performance of each particle (gene subset) of every newly generated population is evaluated by Naive-Bayes classifier. Naive–Bayes (NB) [17–19] is a probabilistic learning approach, which works on an underlying assumption that all the feature values are conditionally independent given a corresponding target class. In this study, the Naive–Bayes classifier is chosen since CFS is based on the similar hypothesis as of Naive–Bayes [4]. So we combined both approaches in a single model to achieve superior predictive performance on microarray datasets. First of all, genes present in the kth particle, i.e. where $p_{kd} = 1$, are used to find a gene subset g_i which reduces the gene dimensions to m' in the final dataset D'. The dataset D' contains N gene expression vectors $\vec{s}_j (\subseteq \vec{S}_j, j = 1, 2, \ldots, N)$ each of dimensionality m'. Then following equation is used by the NB classifier to predict the target class of the given gene expression values $(e_1, e_2, \ldots, e_{m'})$ for the selected gene subset(g_i) that defines the sample vector $\vec{s}_j$:

$$c_{\text{NB}}^{(i)} = \underset{c^{(i)} \in C^{(i)}}{\textbf{argmax}} \; P(c^{(i)}) \prod_{j=1}^{m'} P(e_j | c^{(i)}) \quad (11)$$

where $c_{\text{NB}}^{(i)}$ is the predicted class value for test sample which have maximum probability assigned by NB classifier. The probabilities $P(c^{(i)})$ and $P(e_j|c^{(i)})$ are simply estimated by counting the frequencies over the training data.

We have partitioned the dataset D' into 10 disjoint sets $(D'_1, D'_2, \ldots, D'_{10})$ of similar size using stratified 10-fold cross-validation. The NB classifier learned on the basis of $(D' - D'_v)$ training data and then this learned hypothesis is used to classify the samples in the set D'_v by applying the rule in Eq. (11). This process is repeated ten times so that each set D'_v is used once as the test set. Finally a misclassification error for 10-fold cross validation ($\mathcal{E}$) is calculated as follows:

$$\mathcal{E}(D', g_i) = \frac{1}{10} \sum_{v=1}^{10} \mathcal{E}_v(D'_v, g_i) \quad (12)$$

where

$$\mathcal{E}_v(D'_v, g_i) = \frac{|\{(s_j^{m'}, c^{(i)}) \in D'_v : c_{\text{NB}}^{(i)} \ne c^{(i)}\}|}{|D'_v|} \quad (13)$$

4. Experimental setup and results

In this section, we describe experimental setup used to evaluate the performance of our method. First subsection deliberates the gene expression datasets and parameter settings used in the experiments which is followed by the results and discussion subsection.

4.1. Gene expression datasets and parameter settings

We have chosen eleven benchmark cancer datasets of microarray gene expression data, which are obtained from http://csse.szu.edu.cn/staff/zhuzx/Datasets.html [41], to evaluate our proposed CFS-iBPSO-NB approach. Table 1 shows the descriptive summary of each dataset, includes number of observed samples, number of genes per sample and number of classes. The datasets used in the experiments have large number of dimensions (thousands of genes) and consist of two and more than two classes, which are appropriate to show the effectiveness of our approach. Thus these datasets represent binary and multi-class cancer classification problems. The cancer datasets are of Colon Tumor, Leukemia 2-class (ALL-AML), Leukemia 3-class (ALL-AML-3), Leukemia 4-class (ALL-AML-4), Central Nervous System (CNS), Breast, Lung, Ovarian, Lymphoma, mixed-lineage leukemia (MLL), small round blue cell tumors (SRBCT). The PGPF phase of the proposed method

Table 1
Descriptive summary of microarray datasets.

Datasets	No. of total genes	No. of samples	No. of classes
Colon	2000	62	2
Central nervous system	7129	60	2
ALL-AML	7129	72	2
Breast	24,481	97	2
Lung	12,533	181	2
Ovarian	15,154	253	2
ALL-AML-3	7129	72	3
ALL-AML-4	7129	72	4
Lymphoma	4026	62	3
MLL	12,582	72	3
SRBCT	2308	83	4

Table 2
Parameters for improved Binary PSO algorithm.

Parameters	Values
Swarm size (W)	60
Total number of iterations (T)	100
w_{max}	0.9
w_{min}	0.4
ξ	0.9
c_1	2
c_2	2
v_{min}	−6
v_{max}	6

is implemented using Weka 3.6 data mining software [43]. Here we replaced the missing values of gene expressions with mean values and all the datasets are standardized to have mean equal to zero and standard deviation equal to one by Eq. (1). Then the filter algorithm Correlation-based Feature Selection is applied to each dataset to generate corresponding reduced datasets. The parameters used in the proposed iBPSO of GOCC phase are listed in Table 2. Here the parameter settings was adopted by conducting many trials to get best objective value and inline with many related work which utilizes Binary PSO [13,44].

4.2. Results and discussion

In this section, the performance of our proposed hybrid framework CFS-iBPSO-NB is examined by comparing the experimental results obtained on 11 microarray datasets with seven other techniques on the basis of classification accuracy and number of genes selected. In particular, we compared with the popular classification algorithms Support Vector Machine [27,45] and Random Forest [46] (without any feature selection method), FCBF [34], which is a Fast Correlation-Based Filter feature selection method, conventional Binary Particle Swarm Optimization (BPSO), PSO based Decision-Tree classifier [9] and hybrid model based methods like Markov blanket-embedded genetic algorithm (MBEGA) [41], and Taguchi chaotic binary particle swarm optimization (CFS-TCBPSO-1NN) [42].

In earlier studies, performance evaluation is done by randomly partitioning the original dataset into training and test sets and performing gene selection on training data and checking fitness of selected genes on test sets, but this approach is less reliable in the microarray datasets having small sample size. It is reported in the literature that a stratified 10-fold cross-validation is the best method for performance evaluation of machine learning algorithms [21,22]. So, we have applied stratified 10-fold cross-validation to evaluate the performance of selected gene subsets by Naive–Bayes (NB) learning algorithm. Stratification is used herein to reduce variance in accuracy calculation especially for multiclass datasets and it also preserved the class distribution while partitioning the dataset into training and test sets [4]. For the comparison purpose FCBF and BPSO are implemented using the same learning approach i.e. Naive–Bayes with stratified 10-fold cross-validation and other methods are also implemented using their learning approach with stratified 10 fold cross-validation. These methods are implemented using Weka environment and MATLAB 2012 environment.

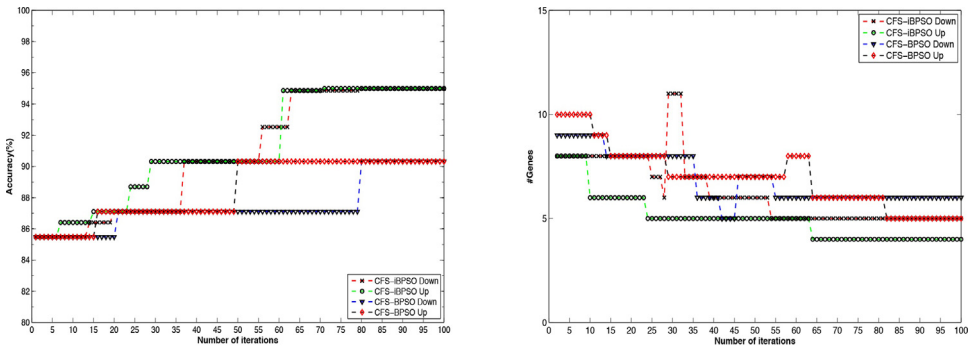
We have analyzed the effect of increasing ($\uparrow w$) and decreasing ($\downarrow w$) inertia weight schemes with our proposed iBPSO. To show the efficacy of the proposed iBPSO, we have compared the results of CFS-iBPSO ($\downarrow w$) and CFS-iBPSO ($\uparrow w$) with CFS-BPSO ($\downarrow w$) and CFS-BPSO ($\uparrow w$).

Figs. 2–4 compare the results obtained using CFS-BPSO ($\downarrow w$), CFS-BPSO ($\uparrow w$), CFS-iBPSO ($\downarrow w$) and CFS-iBPSO ($\uparrow w$) for 11 microarray datasets, part (a) of the figure represents classification accuracy (%) for 100 iterations whereas, the part (b) represents number of genes selected (#genes) over 100 iterations. These curves show that increasing ($\uparrow w$) scheme performed significantly better than decreasing ($\downarrow w$) scheme for both BPSO and iBPSO. Figs. 2–4 also validate that iBPSO effectively solves the local stagnation problem of BPSO and provides better results than conventional BPSO in terms of both higher classification accuracy and lower number of selected genes for all 11 microarray datasets.

In Table 3 we have shown the best value, average and standard deviation of classification accuracy (accu(%))and number of genes (#genes) for the 11 microarray datasets obtained by CFS-iBPSO ($\downarrow w$) and CFS-iBPSO ($\uparrow w$) for 10 independent executions (100 iterations each) of the algorithm. In terms of both mean classification accuracy and number of selected genes, CFS-iBPSO ($\uparrow w$) provides better or statistically comparable results than CFS-iBPSO ($\downarrow w$). It clearly shows that the increasing ($\uparrow w$) inertia weight scheme is more likely to perform better than the decreasing ($\downarrow w$) scheme. So, for further investigations we have used the increasing ($\uparrow w$) inertia weight scheme in our method. Table 4 shows the comparative evaluation of our method with other seven methods on the basis of the mean classification accuracy obtained for 11 microarray datasets. The CFS-iBPSO ($\uparrow w$)-NB has achieved highest average classification accuracy 98.28% among all eight approaches. The average accuracy of the algorithms are calculated over 10 independent executions (100 iterations each) for each dataset individually. The bold typeface in the table represents the highest mean classification accuracy among the eight algorithms on each dataset. The CFS-iBPSO ($\uparrow w$)-NB achieved 100% classification accuracy on seven out of eleven datasets (ALL-AML, Lung, Ovarian, ALL-AML-3, Lymphoma, MLL, SRBCT) and have shown better performance on all datasets except for the Colon tumor. The SVM classifier, in which complete gene dimensions are used, has the lowest classification accuracy among the eight methods for all datasets. The average number of genes selected by the eight approaches are shown in Table 5. Again CFS-iBPSO ($\uparrow w$)-NB has selected less than 1% genes from total genes for all the datasets except SRBCT dataset where about 1.5% genes are selected. The gene dimensionality in Ovarian and ALL-AML-3 (Leukemia-3) datasets are reduced to 3.3 and 6 genes respectively.

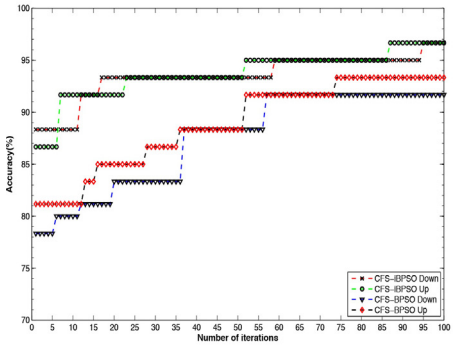
In Table 6, we have reported the average execution time (in seconds) for 100 iterations each of the iBPSO ($\uparrow w$)-NB and the TCBPSO-1NN algorithms for 11 microarray datasets. It can be summarized that our proposed iBPSO ($\uparrow w$)-NB takes less execution time (approximately 40–60%) than the TCBPSO-1NN algorithm in Gene Optimization and Cancer Classification (GOCC) phase. Thus our proposed iBPSO ($\uparrow w$) method ensures faster and more reliable gene selection for classification process without increasing complexity.

The results in Tables 4 and 5 show that CFS-iBPSO ($\uparrow w$)-NB attains superior or comparable classification accuracy with effectively small dimensional gene subset on most of the microarray datasets. Experimental results demonstrate that methods with gene selection achieve better classification performance than without gene selection. But in case of microarray datasets, using only

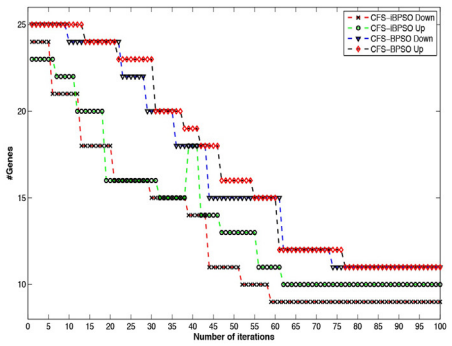


Colon (a)

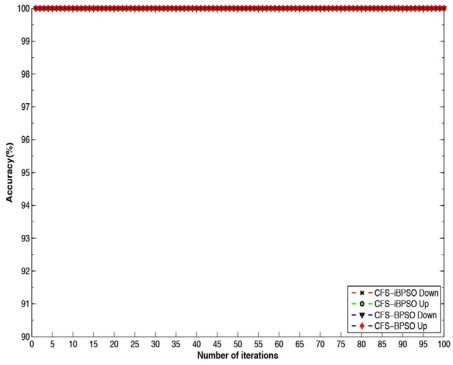
Colon (b)



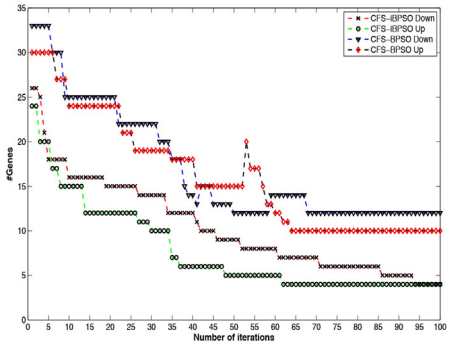
Central Nervous System (a)



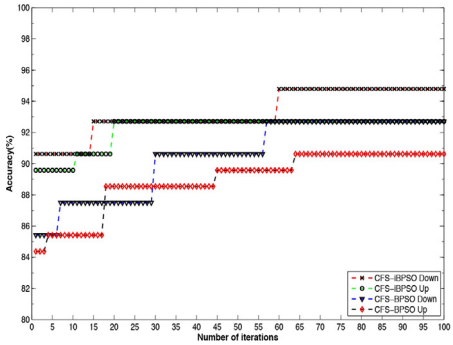
Central Nervous System (b)



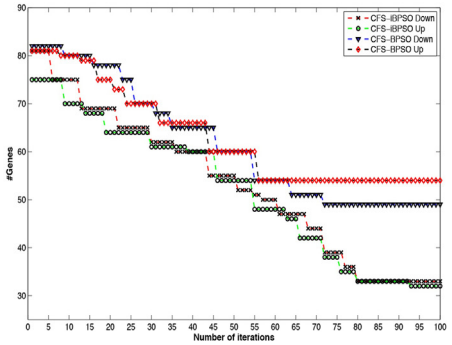
ALL-AML (a)



ALL-AML (b)

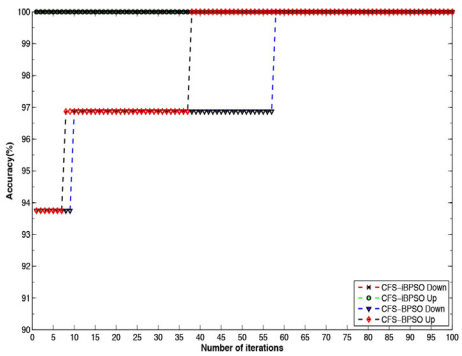


Breast (a)

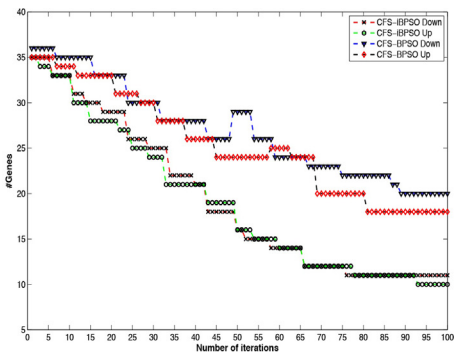


Breast (b)

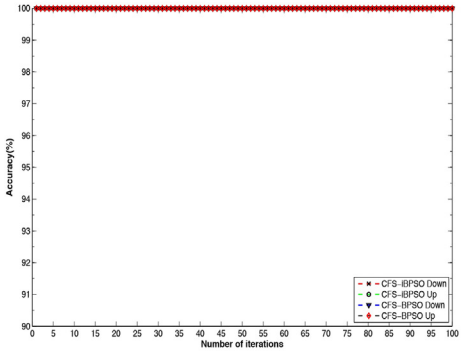
Fig. 2. Comparative evaluation curves of CFS-BPSO ($\downarrow w$), CFS-BPSO ($\uparrow w$), CFS-iBPSO ($\downarrow w$) and CFS-iBPSO ($\uparrow w$) on classification accuracy (a) and number of genes (b) in Colon, Central Nervous System, ALL-AML, and Breast microarray datasets.



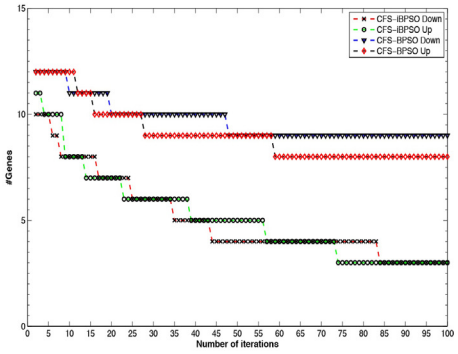
Lung (a)



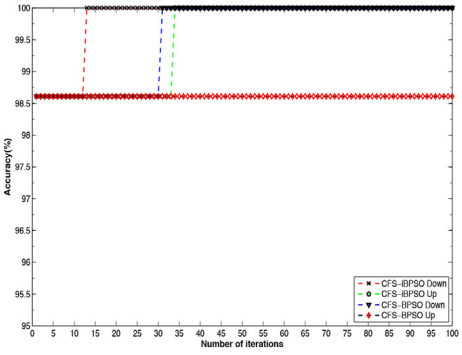
Lung (b)



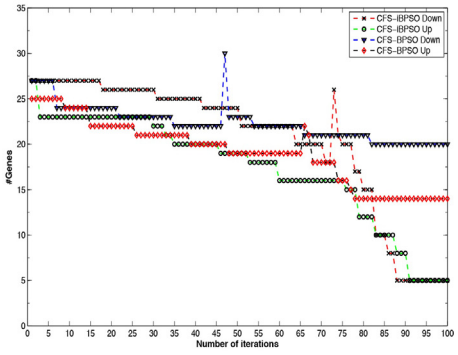
Ovarian (a)



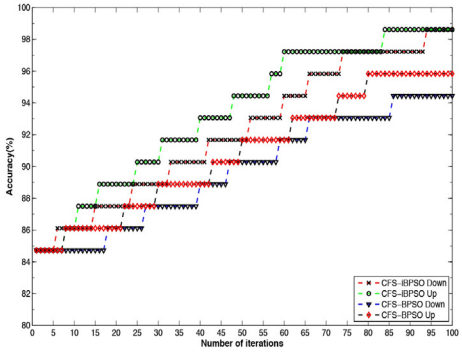
Ovarian (b)



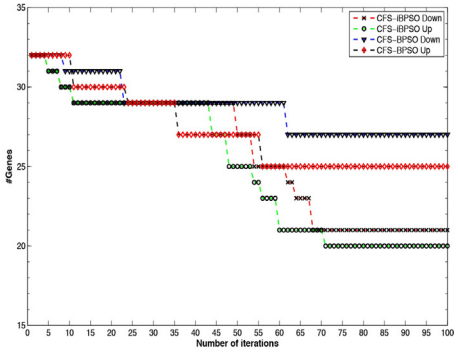
ALL-AML-3 (a)



ALL-AML-3 (b)



ALL-AML-4 (a)



ALL-AML-4 (b)

Fig. 3. Comparative evaluation curves of CFS-BPSO ($\downarrow w$), CFS-BPSO ($\uparrow w$), CFS-iBPSO ($\downarrow w$) and CFS-iBPSO ($\uparrow w$) on classification accuracy (a) and number of genes (b) in Lung, Ovarian, ALL-AML-3, and ALL-AML-4 microarray datasets.

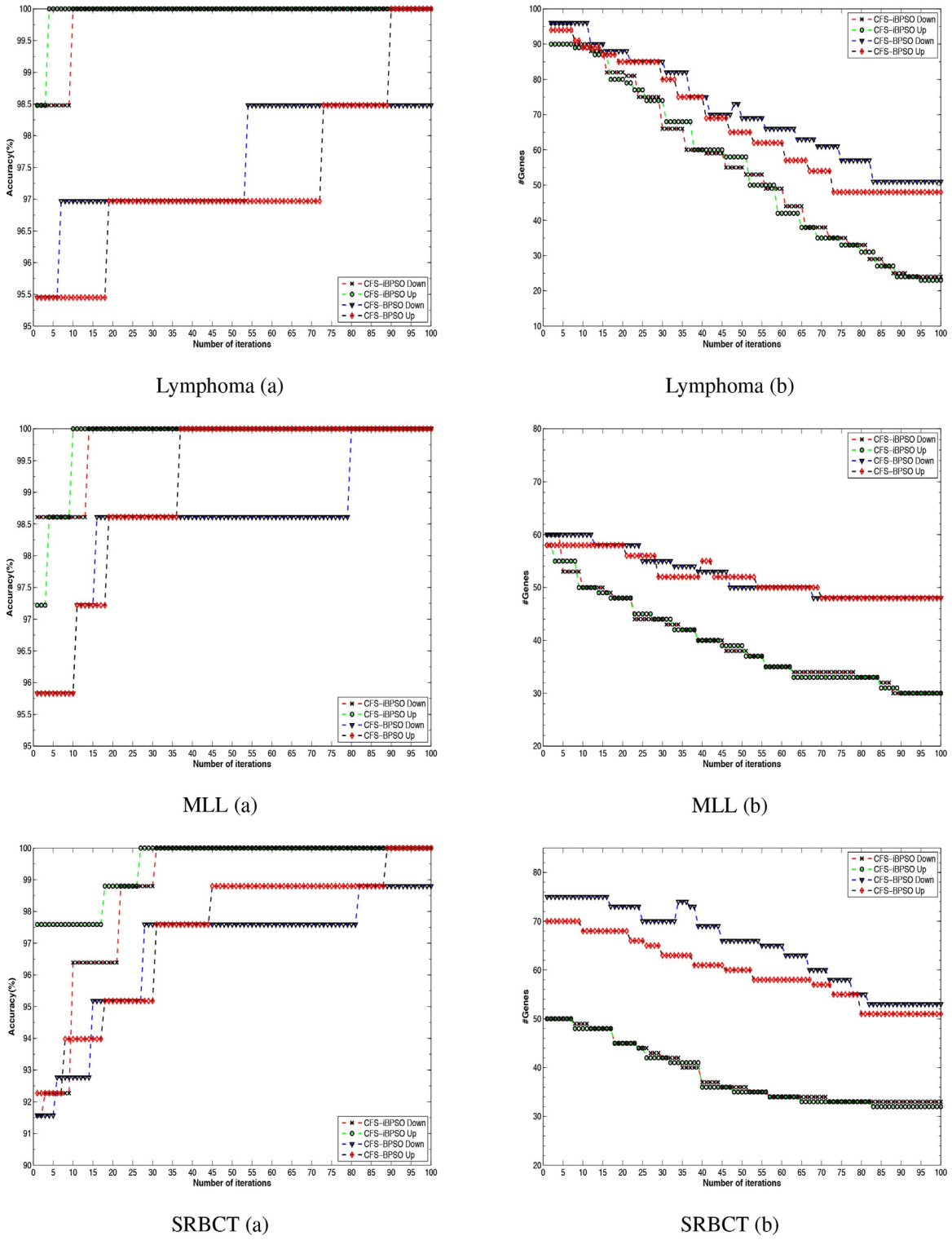


Fig. 4. Comparative evaluation curves of CFS-BPSO (↓ w), CFS-BPSO (↑ w), CFS-iBPSO (↓ w) and CFS-iBPSO (↑ w) on classification accuracy (a) and number of genes (b) in Lymphoma, MLL, and SRBCT microarray datasets.

a wrapper approach for gene selection has the limitations of slow computation speed and greater search complexity due to its very high dimensionality. So, here we designed a hybrid model in which a pre-filtering is done before using the wrapper model to reduce the computational cost and search complexity of the model and providing upto 100% classification accuracy on the basis of significantly small gene dimensions.

In Fig. 5, a graph is plotted to compare classification accuracy obtained by methods on each dataset. Fig. 6 shows the classification accuracy averaged over 11 microarray datasets for all the eight methods, and it exhibits that CFS-iBPSO (↑ w)-NB has outperformed.

Table 3
Statistical results obtained by CFS-iBPSO ($\downarrow w$) and CFS-iBPSO ($\uparrow w$) on 11 microarray datasets.

Datsets	Performance measures	CFS-iBPSO ($\downarrow w$)			CFS-iBPSO ($\uparrow w$)		
		Best	Mean	SD	Best	Mean	SD
Colon(2000)	accu(%)	95	94.12	0.02	95	94.89	0.06
	#genes	5	5.6	1.78	4	4.2	0.43
CNS(7129)	accu(%)	95.6	95.00	0.06	96.67	95.84	0.08
	#genes	9	11.1	1.15	10	10.5	0.52
ALL-AML(7129)	accu(%)	100.00	100.00	0	100.00	100.00	0
	#genes	4	4.5	0.60	4	4.3	0.48
Breast(24481)	accu(%)	94	92.75	0.05	92.75	92.75	0.02
	#genes	33	35.6	0.40	32	32.7	0.48
Lung(12533)	accu(%)	100.00	100.00	0	100.00	100.00	0
	#genes	11	12.1	1.02	10	10.6	0.51
Ovarian(15154)	accu(%)	100.00	100.00	0	100.00	100.00	0
	#genes	3	3.5	0.51	3	3.3	0.48
ALL-AML-3(7129)	accu(%)	100.00	100.00	0	100.00	100.00	0
	#genes	5	6.0	0.99	5	6.0	0.99
ALL-AML-4(7129)	accu(%)	98.02	97.22	0.04	98.61	97.63	0.03
	#genes	21	22.5	1.60	20	20.7	0.48
Lymphoma(4026)	accu(%)	100.00	100.00	0.002	100.00	100.00	0.002
	#genes	24	25.5	0.53	23	24.0	0.52
MLL(12582)	accu(%)	100.00	100.00	0	100.00	100.00	0
	#genes	30	31.1	0.93	30	30.8	0.91
SRBCT(2308)	accu(%)	100.00	100.00	0	100.00	100.00	0
	#genes	33	35.4	0.78	32	34.1	0.42

Table 4
Comparisons of mean classification accuracy obtained for 11 microarray datasets.

Datasets	SVM	Random Forest	FCBF	BPSO	PSO-DT	MBEGA	CFS-TCBPSO-1NN	CFS-iBPSO-NB
Colon	64.52	80.65	85.48	74.19	90.32	86.66	**98.43**	94.89
Central nervous system	65.40	58.33	76.67	53.33	58.33	72.21	93.33	**95.84**
ALL-AML	65.28	94.44	**100.00**	**100.00**	95.83	95.89	**100.00**	**100.00**
Breast	52.58	63.92	57.73	54.63	67.01	80.74	91.70	**92.75**
Lung	89.93	91.27	98.79	99.30	**100.00**	98.96	99.33	**100.00**
Ovarian	86.95	94.07	99.91	94.07	97.23	99.71	**100.00**	**100.00**
ALL-AML-3	52.89	83.33	95.83	97.22	95.83	96.64	99.67	**100.00**
ALL-AML-4	52.78	76.39	95.00	91.66	94.44	91.93	95.00	**97.63**
Lymphoma	69.70	96.97	98.48	95.45	98.50	97.68	**100.00**	**100.00**
MLL	65.28	94.44	98.61	95.8	94.04	94.33	97.32	**100.00**
SRBCT	64.94	**100.00**	98.94	99.00	92.49	99.23	**100.00**	**100.00**
Average	66.39	84.89	91.40	86.79	89.46	92.18	97.71	**98.28**

Table 5
Comparisons of average number of genes selected for 11 microarray datasets.

Datasets	SVM	Random Forest	FCBF	BPSO	PSO-DT	MBEGA	CFS-TCBPSO-1NN	CFS-iBPSO-NB
Colon	2000	2000	14	478.1	643.3	24.5	9	**4.2**
Central nervous system	7129	7129	28	2233	1486	20.5	25.2	**10.5**
ALL-AML	7129	7129	51	572.4	1468	15.8	**3.1**	4.3
Breast	24481	24481	92	1930	10465	**14.5**	30.2	32.7
Lung	12533	12533	119	2778	1657	14.1	15	**10.6**
Ovarian	15154	15154	30	5074.7	3594.2	9.0	10	**3.3**
ALL-AML-3	7129	7129	53	3379	1294.1	20.1	10.2	**6.0**
ALL-AML-4	7129	7129	71	2802	1845	26.2	25.6	**20.7**
Lymphoma	4026	4026	105	948.8	1346	34.3	30.5	**24.0**
MLL	12582	12582	97	4721	4847	32.1	32	**30.8**
SRBCT	2308	2308	82	794	874	60.7	39.1	**34.1**

The bold values indicate the minimum value of the average number of genes selected for the microarray dataset.

5. Conclusion

In this paper, we proposed a two phase hybrid model for gene selection and cancer classification using DNA microarray. The model combines a multivariate filter CFS and a wrapper approach iBPSO ($\uparrow w$)-NB with stratified 10-fold cross-validation to implement the classification model. In this proposed model, CFS reduces the dimensionality of the data by eliminating the irrelevant and redundant genes in the Predictive Gene Pre-Filtering phase. Then in Gene Optimization and Cancer Classification phase, iBPSO ($\uparrow w$) makes use of this low dimensional gene subset to select an optimal subset of important genes with the help of Naive–Bayes classifier and 10-fold cross-validation which provides highest classification accuracy. We compared our model with seven popular methods from each category like Support Vector Machines, Random Forest, a filter (FCBF), a wrapper (standard BPSO and PSO-DT) and hybrid models (MBEGA and CFS-TCBPSO-1NN) on 11 benchmark cancer microarray datsets. Experimental results show that CFS-iBPSO ($\uparrow w$)-NB outperforms in terms of classification accuracy and number of selected genes in most cases. It attains 100% classification accuracy for seven datasets. CFS-iBPSO ($\uparrow w$)-NB effectively reduces the dimensionality by eliminating irrelevant and redun-

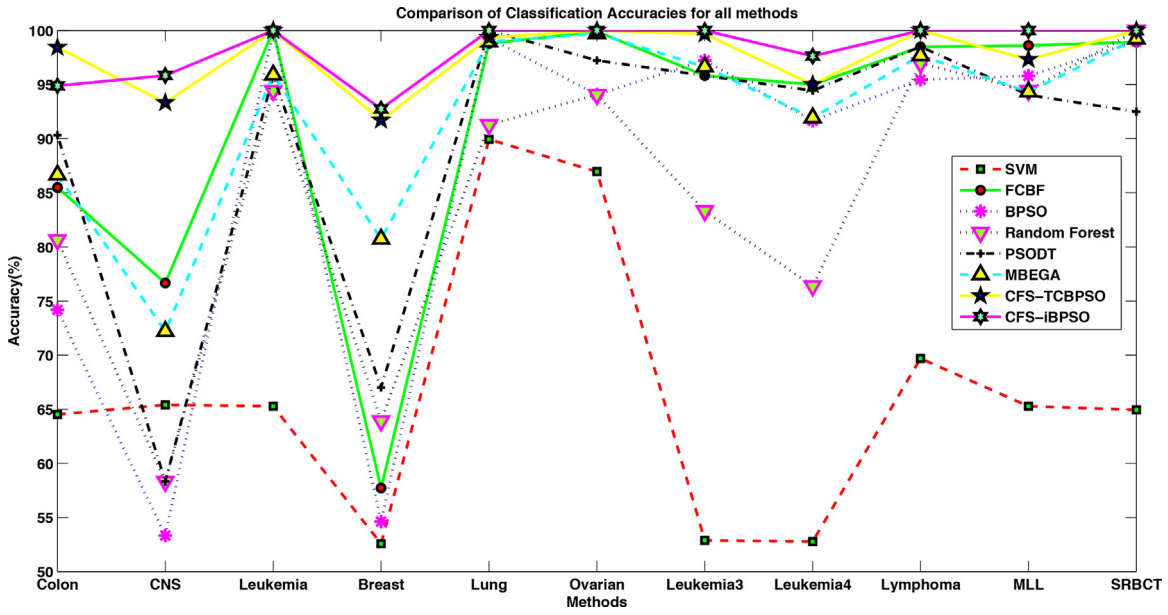


Fig. 5. Comparison of mean classification accuracy for all methods.

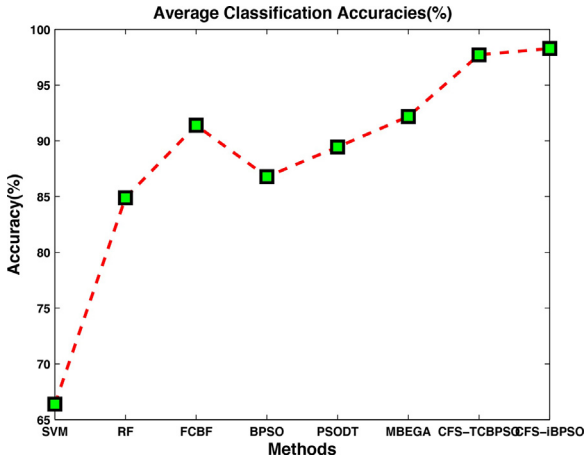


Fig. 6. Average classification accuracy of all methods.

Table 6
Comparison of average execution time (in seconds) for iBPSO ($\uparrow w$)-NB and TCBPSO-1NN algorithms for 11 microarray datasets.

Datasets	iBPSO ($\uparrow w$)-NB	TCBPSO-1NN
Colon	39.2723	85.0923
Central nervous system	78.4314	194.1163
ALL-AML	141.4822	236.7786
Breast	231.8018	637.7027
Lung	311.2191	553.2579
Ovarian	92.9048	220.4648
ALL-AML-3	204.1492	355.1865
ALL-AML-4	321.3342	578.7346
Lymphoma	366.2350	746.4172
MLL	245.7108	483.9951
SRBCT	302.8177	633.9447

dant genes and hence provides a low dimensional set of most important genes which are capable to achieve higher classification accuracy with much less complexity. Thus it could be an efficient tool for DNA microarray analysis.

References

[1] C.M. Perou, S.S. Jeffrey, M. Van De Rijn, C.A. Rees, M.B. Eisen, D.T. Ross, A. Pergamenschikov, C.F. Williams, S.X. Zhu, J.C. Lee, et al., Distinctive gene expression patterns in human mammary epithelial cells and breast cancers, Proc. Natl. Acad. Sci. U. S. A. 96 (16) (1999) 9212–9217.
[2] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[3] M.-Y. Wu, D.-Q. Dai, Y. Shi, H. Yan, X.-F. Zhang, Biomarker identification and cancer classification based on microarray data using Laplace Naive Bayes model with mean shrinkage, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (6) (2012) 1649–1662, http://dx.doi.org/10.1109/TCBB.2012.105.
[4] M.A. Hall, Correlation-based feature selection for machine learning (Ph.D. thesis), The University of Waikato, 1999.
[5] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1) (1997) 273–324.
[6] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artif. Intell. 97 (1) (1997) 245–271.
[7] Q. Shen, W.-M. Shi, W. Kong, Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, Comput. Biol. Chem. 32 (1) (2008) 53–60.
[8] S. Li, X. Wu, M. Tan, Gene selection using hybrid particle swarm optimization and genetic algorithm, Soft Comput. 12 (11) (2008) 1039–1048.

[9] K.-H. Chen, K.-J. Wang, K.-M. Wang, M.-A. Angelia, Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data, Appl. Soft Comput. 24 (2014) 773–780.
[10] R. Xu, G.C. Anagnostopoulos, D.C. Wunsch, Multiclass cancer classification using semisupervised ellipsoid artmap and particle swarm optimization with gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (1) (2007) 65–77, http://dx.doi.org/10.1109/TCBB.2007.1009.
[11] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, vol. 1, New York, NY, 1995, pp. 39–43.
[12] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, vol. 4, IEEE, December, 1995, pp. 1942–1948.
[13] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm algorithm, in: 1997 IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, vol. 5, IEEE, 1997, pp. 4104–4108.
[14] I. Inza, P. Larra naga, R. Blanco, A.J. Cerrolaza, Filter versus wrapper gene selection approaches in DNA microarray domains, Artif. Intell. Med. 31 (2) (2004) 91–103.
[15] Y. Saeys, I. Inza, P. Larra naga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.
[16] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. De Schaetzen, R. Duque, H. Bersini, A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) 9 (4) (2012) 1106–1119.
[17] A. Kelemen, H. Zhou, P. Lawhead, Y. Liang, Naive Bayesian classifier for microarray data, in: Proceedings of the International Joint Conference on Neural Networks, 2003, vol. 3, IEEE, 2003, pp. 1769–1773.
[18] T.M. Mitchell, Machine Learning, 1997.
[19] I. Rish, An empirical study of the Naive Bayes classifier, in: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, vol. 3, IBM, New York, 2001, pp. 41–46.
[20] J. Liu, Y. Mei, X. Li, An analysis of the inertia weight parameter for binary particle swarm optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 666–681, http://dx.doi.org/10.1109/TEVC.2015.2503422.
[21] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, Ijcai, vol. 14 (1995) 1137–1145.
[22] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, vol. 112, Springer, 2013.
[23] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. U. S. A. 96 (12) (1999) 6745–6750.
[24] Y.-J. Li, L. Zhang, M.C. Speer, E.R. Martin, Evaluation of current methods of testing differential gene expression and beyond, in: Methods of Microarray Data Analysis II, Springer, 2002, pp. 185–194.
[25] E.P. Xing, M.I. Jordan, R.M. Karp, et al., Feature selection for high-dimensional genomic microarray data, in: ICML, vol. 1, Citeseer, 2001, pp. 601–608.
[26] M. Xiong, X. Fang, J. Zhao, Biomarker identification by feature wrappers, Genome Res. 11 (11) (2001) 1878–1887.
[27] T.S. Furey, N. Cristianini, N. Duffy, D.W. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.
[28] Y. Tang, Y.Q. Zhang, Z. Huang, Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (3) (2007) 365–381, http://dx.doi.org/10.1109/TCBB.2007.70224.
[29] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, et al., Multiclass cancer diagnosis using tumor gene expression signatures, Proc. Natl. Acad. Sci. U. S. A. 98 (26) (2001) 15149–15154.
[30] K.-B. Hwang, D.-Y. Cho, S.-W. Park, S.-D. Kim, B.-T. Zhang, Applying machine learning techniques to analysis of gene expression data: cancer diagnosis, in: Methods of Microarray Data Analysis, Springer, 2002, pp. 167–182.
[31] F. Fernández-Navarro, C. Hervás-Martínez, R. Ruiz, J.C. Riquelme, Evolutionary generalized radial basis function neural networks for improving prediction accuracy in gene classification using feature selection, Appl. Soft Comput. 12 (6) (2012) 1787–1800.
[32] L. Li, L.G. Pedersen, T.A. Darden, C.R. Weinberg, Computational analysis of leukemia microarray expression data using the GA/KNN method, in: Methods of Microarray Data Analysis, Springer, 2002, pp. 81–95.
[33] W. Li, Y. Yang, How many genes are needed for a discriminant microarray data analysis, in: Methods of Microarray Data Analysis, Springer, 2002, pp. 137–149.
[34] L. Yu, H. Liu, Feature selection for high-dimensional data: a fast correlation-based filter solution, ICML, vol. 3 (2003) 856–863.
[35] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinform. Comput. Biol. 3 (02) (2005) 185–205.
[36] R. Tibshirani, T. Hastie, B. Narasimhan, G. Chu, Diagnosis of multiple cancer types by shrunken centroids of gene expression, Proc. Natl. Acad. Sci. U. S. A. 99 (10) (2002) 6567–6572.
[37] Y. Wang, F. Makedon, Application of relief-F feature filtering algorithm to selecting informative genes for cancer classification using microarray data, in: 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings, IEEE, 2004, pp. 497–498.
[38] E.B. Huerta, B. Duval, J.-K. Hao, A hybrid GA/SVM approach for gene selection and classification of microarray data, in: Applications of Evolutionary Computing, Springer, 2006, pp. 34–44.
[39] C.-P. Lee, Y. Leu, A novel hybrid feature selection method for microarray data analysis, Appl. Soft Comput. 11 (1) (2011) 208–213.
[40] R. Ruiz, J.C. Riquelme, J.S. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Pattern Recognit. 39 (12) (2006) 2383–2392.
[41] Z. Zhu, Y.-S. Ong, M. Dash, Markov blanket-embedded genetic algorithm for gene selection, Pattern Recognit. 40 (11) (2007) 3236–3248.
[42] L.-Y. Chuang, C.-S. Yang, K.-C. Wu, C.-H. Yang, Gene selection and classification using Taguchi chaotic binary particle swarm optimization, Expert Syst. Appl. 38 (10) (2011) 13367–13377.
[43] E. Frank, M. Hall, L. Trigg, G. Holmes, I.H. Witten, Data mining in bioinformatics using Weka, Bioinformatics 20 (15) (2004) 2479–2481.
[44] Y. Del Valle, G.K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, R.G. Harley, Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput. 12 (2) (2008) 171–195.
[45] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[46] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.