ELSEVIER

Computers in Biology and Medicine 36 (2006) 553–573

Computers in Biology and Medicine

www.intl.elsevierhealth.com/journals/cobm

A novel ensemble machine learning for robust microarray data classification

Yonghong Peng*

Department of Computing, University of Bradford, West Yorkshire BD7 1DP, UK

Received 12 November 2004; accepted 11 April 2005

Abstract

Microarray data analysis and classification has demonstrated convincingly that it provides an effective methodology for the effective diagnosis of diseases and cancers. Although much research has been performed on applying machine learning techniques for microarray data classification during the past years, it has been shown that conventional machine learning techniques have intrinsic drawbacks in achieving accurate and robust classifications. This paper presents a novel ensemble machine learning approach for the development of robust microarray data classification. Different from the conventional ensemble learning techniques, the approach presented begins with generating a pool of candidate base classifiers based on the gene sub-sampling and then the selection of a sub-set of appropriate base classifiers to construct the classification committee based on classifier clustering. Experimental results have demonstrated that the classifiers constructed by the proposed method outperforms not only the classifiers generated by the conventional machine learning but also the classifiers generated by two widely used conventional ensemble learning methods (bagging and boosting).

Keywords: Microarray data; Machine learning; Ensemble learning; Classification

1. Introduction

Advances of microarray technology allow recording the expression levels of thousands of genes simultaneously which provide an effective experimental protocol for gaining insight into the cellular mechanism

* Tel.: +44 1274 23 3963; fax: +44 1274 23 3920.
E-mail address: y.h.peng@bradford.ac.uk.

0010-4825/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2005.04.001

and the nature of complex biological process. The microarray data analysis aims at extracting significant and meaningful information from the data, and generally includes two main tasks. One task is to reveal the function of an individual gene or a subset of genes. This is performed by analyzing the change patterns of expression levels of the interesting genes over times or under varied conditions [1,2]. Another task is the identification of conditions of a biological system in terms of the expression levels of the associated genes [3]. The second task is considered as a classification problem from the machine learning point of view and can be divided into two sub-classes. One is to classify the sample into a pre-defined class based on the relevant genes (called microarray data classification), and another is to find the sub-classes of organism conditions that could help further investigation of the biological conditions (called class discovery).

Recently research has demonstrated convincingly that accurate cancer diagnosis can be achieved by performing microarray data classification, i.e. by constructing classifiers to compare the gene expression profile of a tissue of unknown cancer status to a database stored expression profiles from tissues of known cancer status. The process of microarray classification consists of two successive steps. The first step is to select a set of significant and relevant genes and then develop a classification model which can produce accurate prediction for unseen data. Given n training data $T = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, in which $x_i \in R^m$ represents a gene vector involving m genes and $y_i \in \{0, 1, \ldots, k-1\}$ indices the class of the associated instances, the task of microarray classification is to find a optimal subset of genes enabling the construction of an accurate classifier c: $R^{m'} \to \{0, 1, \ldots, k-1\}$ that maximizes the probability that $c(x_i') = y_i$ for $i = 1 \sim n$, where $x_i' \in R^{m'}$ and m' is the number of genes selected ($m' \leqslant m$).

Many machine learning techniques have been developed in the literature [4,5] and many of them have been employed for both steps, including the techniques of feature selection [6,7], clustering methods [1,3,8–10], and classification techniques, e.g. decision tree [11,12], K-NN [13,14], neural networks [15,16], and support vector machines (SVM) [2,17,18]. Most of the existing research attempts to select an optimal subset of genes and then construct an accuracy classification model based on the selected genes. This approach has intrinsic drawbacks for the microarray data classification due to the following issues: (1) the microarray data inherently contains a huge number of genes but a small number of samples (e.g. the leukaemia dataset [3] contains 6817 genes and 72 instances, see Section 3 for the typical cases of microarray data); this problem is known as the curse of dimensionality in machine learning; (2) the microarray data is associated with a variety of uncertainties (the process of microarray data gathering, e.g. fabrication, hybridization, image processing, etc., always adds various sources of noise [19,20]). Due to the influence of various uncertainties together with the limitation of labelled training instances, the conventional machine learning techniques have difficulty to produce reliable classification models. Quite often selecting only one or a few genes can classify a majority of training samples [21]. The robustness of the classification model based on such a few dominant genes and a limited number of labelled training instances remains an open question. There is therefore a great need to develop general approaches and robust methods that are able to overcome the limitation of the small number of training instances and reduce the influence of uncertainties so as to produce reliable classification results. The motivation for this study is to develop general and robust machine learning methods that are less sensitive to the selection of genes and are capable of removing the uncertainties of gene expression, by making as much use as possible of the great number of genes.

The method presented in this paper is based on the idea of ensemble learning through seeking an optimal and robust combination of multiple classifiers. Ensemble learning employs a single or multiple learning algorithms to generate a set of diverse base classifiers, and then combine them together as a committee

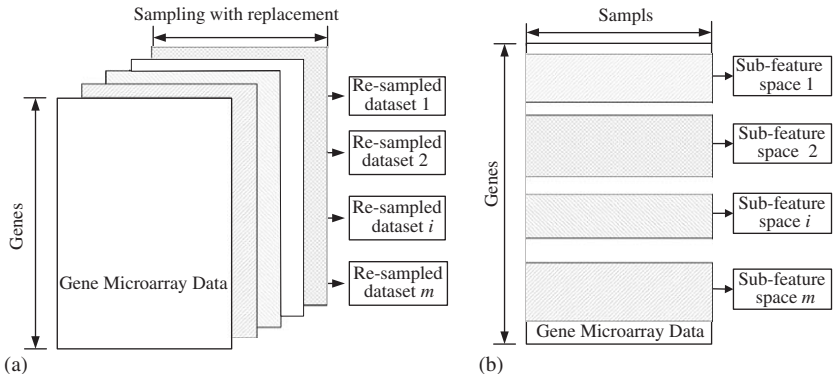

Fig. 1. Two schemes for the generation of base classifiers.

to make more appropriate decisions for classifying new data instances. Much research has shown the promise of ensemble learning for improving the accuracy in classifying data under uncertainties [22]. A necessary and sufficient condition for an ensemble to outperform its individual members is that the base classifiers should be accurate and diverse [23]. An accurate classifier is one that has an error rate of better than randomly guessing classes for new instances, and two classifiers are diverse if they make different (uncorrelated) errors on common data instances [22].

The base classifiers in ensemble learning can be generated by many ways. For example, they can be generated using different learning algorithms, or be generated by means of learning from the re-sampling of the original training dataset. The re-sampling methods have been widely used in ensemble learning [22]. Two basic re-sampling methods are the re-sampling of instances and sub-sampling of features, as shown in Fig. 1. In Fig. 1(a), a set of training sets are generated by means of re-sampling from the original dataset with instance replacement, i.e. the re-sampled dataset have the same number of instance as the original set (some instances are repeated) and the same number of genes. This is called instance re-sampling-based ensemble learning. Most of the existing ensemble learning methods are performed based on the re-sampling of training instances, such as the widely used Bagging [24,25] and Boosting ensemble learning methods [26,27]. In Fig. 1(b), a set of new training sets are generated, which involve different subsets of the original gene set. This is called feature sub-sampling-based ensemble learning. A few researchers [28–30] have investigated the ensemble learning method based on the feature sub-sampling and have demonstrated its promise.

Facing the fact of limited number of training instances and large number of genes available, the approach presented in this paper is developed based on the sub-sampling of the features (genes). The motivation of developing this gene sub-sampling-based ensemble learning approach is to use the functional diversity of large number of genes to generate diverse base classifiers, and then to reduce the influence of uncertainties from genes by involving the overlapping of genes' functions.

This paper is organized as follows. Section 2 presents in detail the proposed method including review of the techniques related for our proposed approach. In Section 3 a set of experimental results is presented to demonstrate the effectiveness of the proposed method. The discussion of the results is also given in the end of Section 3. The conclusions are given in Section 4.

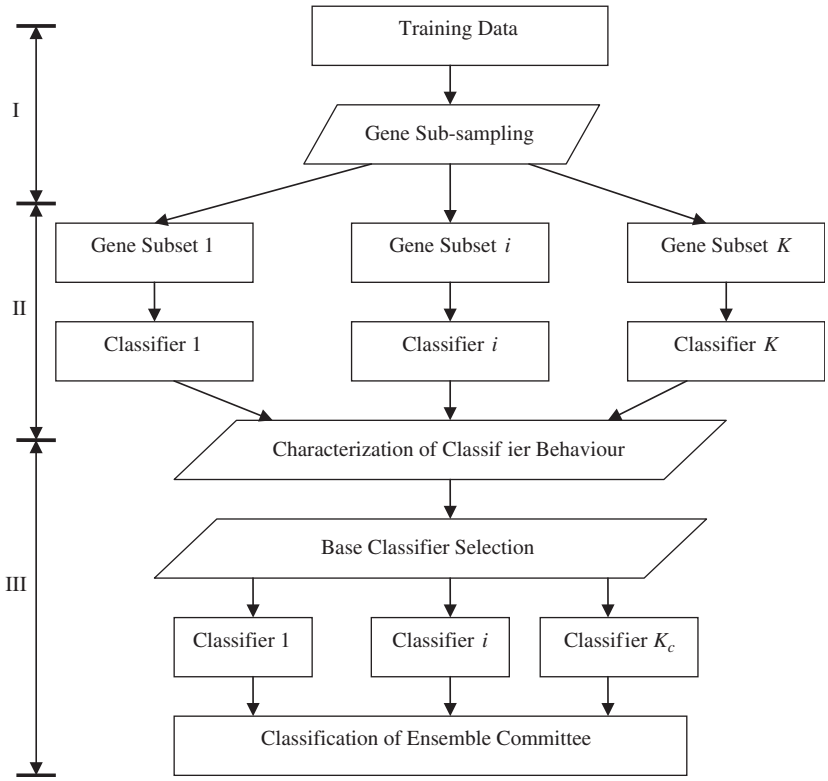

Fig. 2. The proposed ensemble method.

2. Methods

This paper proposes an approach for the construction of accurate and diverse ensemble members by means of learning from sampled sub-sets of genes. The method proposed in this paper is shown in Fig. 2, which consists of three basic procedures.

Step 1: Sub-sampling of genes. The ensemble learning method presented in this paper is developed based on the sub-sampling of genes, as shown in Fig. 1(b). In order to sample genes that can lead to accurate and diverse base classifiers a hybrid sub-sampling method is designed for the gene sub-sampling, as introduced in Sections 2.1 and 2.2.

Step 2: Generation of candidate classifiers. This step applies machine learning methods to generate a pool of candidate classifiers. The inputs for each classifier correspond to the associated gene subset generated in the first step. The particular machine learning algorithm used in this study is the support vector machine, which is discussed in Section 2.3. The consideration of applying SVM is due to the ability of SVM to deal with variable and high dimensionality of training data.

Step 3: Construction of the ensemble committee. This step comprises two sub-steps. One is the characterization of the behaviour of candidate base classifiers generated in the second step, and the other is the selection of a sub-set of base classifiers from the candidate classifiers that construct an robust classification committee. The classifiers having high diversity, i.e. these classifiers that disagree with each

other, and have good classification accuracy will be selected as the committee members. The majority vote mechanism is then employed in this study to make the final decision for the ensemble classification. This step is detailed in Section 2.4.

2.1. Characterization of genes

In the proposed approach, the base classifiers are constructed based on the random selection of genes from the original gene set. This method enables the ensemble classification model to be less sensitive to the gene selection. In this study, a hybrid sub-sampling algorithm is designed to select the genes in order to produce accurate and diverse classifiers. The proposed hybrid sub-sampling method is detailed in Section 2.2. This section discusses the methods of characterizing the significance levels of genes.

A microarray dataset is represented by a $m \times n$ matrix $M = [e_{ij}]_{m\times n}$, where m is the number of genes, and n is the number of instances. A gene (a row in the matrix M) is represented by an expression vector $X(i) = (e_{i1}, e_{i2}, \ldots, e_{in})$ where e_{ij} denotes the expression level of the ith gene in the jth sample. Each sample is labelled with a class $c_j = \{-1, +1\}$. Assume that instances from $j = 1$ to k belong to class $+1$ and sample from $j = k + 1$ to n belong to class -1, the expression vectors of ith gene for class $+1$ and -1 are denoted as $X(i, +) = \{e_{i1}, e_{i2}, \ldots, e_{ik}\}$ and $X(i, -) = \{e_{i(k+1)}, e_{i(k+2)}, \ldots, e_{\mathrm{in}}\}$, respectively.

The gene's significance is measured by its discrimination power that is an important aspect affecting the performance of an individual base classifier. A general approach to characterize the gene's discrimination power is to measure the difference between the expressions levels of samples from different groups [31]

$$d(i) = \frac{\mathrm{dif}(X(i, +), X(i, -))}{s_0 + s}, \tag{1}$$

where $\mathrm{dif}(X(i, +), X(i, -))$ is the difference between group $X(i, +)$ and $X(i, -)$, s is a standard error and s_o is a regularizing constant. Setting $s_o = 0$ Eq. (1) yields a t-statistic.

Another method to measure the gene's discrimination power is the method presented in [3] for the selection of sensible genes for classification. For the ith gene, let $[\mu_+(i), \sigma_+(i)]$ and $[\mu_-(i), \sigma_-(i)]$ denote the means and standard deviation of the expression levels for the samples in class $+1$ and -1, respectively, i.e.

$$\mu_+(i) = \frac{1}{k}\sum_{j=1}^{k} e_{ij}, \tag{2}$$

$$\sigma_+(i) = \sqrt{\frac{\sum_{j=1}^{k}(e_{ij} - \mu_+(i))^2}{k}} \tag{3}$$

and

$$\mu_-(i) = \frac{1}{n-k}\sum_{j=k+1}^{n} e_{ij}, \tag{4}$$

$$\sigma_-(i) = \sqrt{\frac{\sum_{j=k+1}^{n}(e_{ij} - \mu_-(i))^2}{n-k}}. \tag{5}$$

The ability of gene $X(i)$ to discriminate class $+1$ from -1 is then measured by

$$p(i) = \left| \frac{\mu_+(i) - \mu_-(i)}{\sigma_+(i) + \sigma_-(i)} \right|. \tag{6}$$

The $d(i)$ or $p(i)$ measures the capability of the associated gene for distinguishing the classes of samples. Large value of $d(i)$ or $p(i)$ indicates the associated gene's expression levels are more differential between these two classes.

2.2. The proposed hybrid sub-sampling method

Given a set of genes $G = \{g_1, \ldots, g_n\}$, in which each gene has been assigned with an associated significance value, i.e. $S = \{s_1, \ldots, s_n\}$ where $s_i = d(g_i)$ or $p(g_i)$ calculated by Eq. (1) or Eq. (6), the proposed hybrid sub-sampling algorithm involve four steps to generate a subset of genes ($\hat{G}$): (1) gene filtering and parameter calculation; (2) gene ranking; (3) gene partition; (4) gene sub-sampling and re-combinations, as presented in Fig. 3.

The first step filters the genes of which the significance levels ($d(g_i)$ or $p(g_i)$) are less than a given threshold (δ) and calculates the following parameters according to the given number of genes to be sampled (n_G) and two parameters (α and β):

(1) the size of the top gene group (denoted as G_{top}):

$$n_{top} = \alpha \times n_G. \tag{7}$$

(2) the number of genes sampled from G_{top}

$$n_1 = \frac{n_{top}}{\beta}. \tag{8}$$

(3) the number of genes selected from the remaining group of genes

$$n_2 = n_G - n_1. \tag{9}$$

For example, when $n_G = 50$, and the parameters $\alpha = 0.4$ and $\beta = 2$, then the size of G_{top} is $n_{top} = 20$, $n_1 = 10$ and $n_2 = 40$.

In the second step, the genes are ranked in terms of their significance and then are partitioned into two sets. One set contains the top n_{top} genes (the genes with the top significance, denoted as G_{top}) and the other contains the remaining genes (denoted as G_{re}). Two different approaches are employed to perform the sub-sampling of genes, respectively, from these two gene sets. For the genes in G_{top}, they have equal possibility to be sampled as all of them have good discrimination power, while for the genes in G_{re} each of them has different probability to be sampled. These two samplings are performed, respectively, by means of random sampling and Monte Carlo sampling. For sampling from G_{re}, the sampling probabilities of genes are determined by their significance, i.e. a gene with higher significance (indicated by $d(g)$ or $p(g)$) would have higher probability to be sampled. In the end, the genes sampled, respectively, from G_{top} and G_{re} are combined together to form a subset of one sampling. As a result, at the k-run of sub-sampling, a sub-set of genes denoted as $\hat{G}_k$ is thus produced.

$$\hat{G}_k = G'_{top} \cup G'_{re}. \tag{10}$$

Inputs: Gene set G=$\{g_1,\ldots\ g_n\}$ and the associated significant of genes S=$\{s_1,\ldots\ s_n\}$

Parameters: α, β, δ and n_G (number of genes to be sampled in each gene subset),

k, the run of gene subsets to be generated.

Outputs: selected gene set $\hat{G}_k$

Step 1: calculate the n_{top}, n_1 and n_2, and let $s_i = 0$ if $s_i < \delta$.

Step 2: Partition the gene set.

2.1 Rank the genes according to their significances s_i.

2.2 Partition the ranked genes into two sets $G=G_{top} \cup G_{re}$.

Step 3: Sampling from G_{top} and G_{re}.

3.1 Randomly sample n_1 genes from the G_{top}, denoted as G'_{top};

3.2 Apply Monte Carlo method to sample n_2 genes from the G_{re}, denoted as G'_{re}:

3.2.1) Normalize the significance of genes in G_{re}

$$\hat{s}_i = \frac{s_i}{\sum_{G_{re}} s_i}$$

3.2.2) Calculate the cumulative significance distribution function:

$$cd_i = \sum_{j=1}^{i} \hat{s}_j$$

3.2.3) let $\hat{G}_{re} = \Phi$ and repeat the following steps

a) Generate a random number $\xi \in [0,1]$.

b) Retrieve the gene number i such that $cd_{i-1} = \xi$.

c) Check if $i \notin \hat{G}_{re}$ then let $\hat{G}_{re} = \hat{G}_{re} \cup \{i\}$ and go to a), otherwise go to a).

Step 4: $\hat{G}_k = G'_{top} \cup G'_{re}$.

Fig. 3. Hybrid gene sub-sampling algorithm.

By repeating these steps K times (i.e. $k = 1{-}K$), a set of sub-sets of genes can be produced, denoted as $G^S = \{\hat{G}_1, \hat{G}_2, \ldots, \hat{G}_K\}$.

2.3. SVM base classifiers

Many researchers have attempted to apply SVM for the microarray classification and have shown the promise of applying SVMs in microarray data classification. Unlike most of the modelling methods attempting to minimize an objective function (such as the mean square error) for the whole training instances, SVM finds the hyperplanes that produce the largest separation between the decision function values for the instances located at the borderline between two classes. For a given training set, the SVM seeks the optimal hyperplane that maximizes the separating margin between two classes [5].

Given a labelled microarray data $M = \{(x_i, y_i)_j\}$ where, $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$, the target of SVM-based microarray classification is the construction of a decision function $f(x) : R^n \to R$, such that for each x_i, the function yields $f(x_i) > 0$ for $y_i = +1$, and $f(x_i) < 0$ for $y_i = -1$. A SVM employs a linear decision function $f(x) = W^{\mathrm{T}}x + b$ or a nonlinear decision function $f(x) = W^{\mathrm{T}}\phi(x) + b$, where $\phi(x)$ is a nonlinear transform function. The function $f(x)$ is determined by minimizing $J(w, \xi) = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{l} \xi_i$ subject to $y_i(W^{\mathrm{T}}x_i + b) \geqslant 1 - \xi_i$ (linear) or $y_i(W^{\mathrm{T}}\phi(x_i) + b) \geqslant 1 - \xi_i$ (nonlinear), where $C > 0$ is a regularization parameter and $\xi_i \geqslant 0$ $(i = 1, 2, \ldots, l)$ are slack parameters.

For minimization of $J(w, \xi)$, the vector W can be expressed by a linear combination of $\phi(x_i)$, i.e. $w = \sum_{i=1}^{l} a_i y_i \phi(x_i)$. Substituting w into $f(x) = W^{\mathrm{T}}\phi(x) + b$ yields $f(x) = \sum_{i=1}^{l} a_i y_i K(x_i, x) + b$, where the function $K(u, v) = \phi^{\mathrm{T}}(u)\phi(v)$ is called the kernel. The $a_i \geqslant 0$ $(i = 1, 2, \ldots, l)$ are obtained by maximizing $W(a_i) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j K(x_i, x_j)$ subject to $0 \leqslant a_i \leqslant C$, and $\sum_{i=1}^{l} a_i y_i = 0$. Three typical kernel functions used in SVM classification are the Linear $K(u, v) = uv$, polynomial $K(u, v) = (uv + 1)^{\gamma}$ and Gaussian function $K(u, v) = \exp(-\|u - v\|^2/\sigma^2)$.

2.4. Bagging and boosting ensemble learning

Bagging and Boosting are two typical ensemble learning methods based on the instance re-sampling. Bagging is the most straightforward way of generate base classifiers. On each run, Bagging presents the learning algorithm with a training set that consists of a sample of m training examples drawn randomly with replacement from the original training dataset. The Boosting algorithm used in this paper is the AdaBoost developed by Freund and Schapire [26]. In each iteration of AdaBoost, the learning algorithm is invoked to minimize the weighted error on the training set, and generates a base classifier model accordingly. The weights of the training examples are adjusted according to the weighted classification error of the generated classifier model. More weights are placed on the training examples that were misclassified and less weight on examples that were correctly classified. For both Bagging and Boosting, the final classifier is constructed by a weighted vote of the individual classifiers. Refer to [25–27] for more details about Bagging and Boosting.

2.5. The proposed ensemble learning approach

In this paper, the majority vote method is employed to perform the final decision based on the outputs of base classifiers. Given k_c classifiers $C_i(x) : R^n \to \{-1, +1\}$, each classifier is assigned with a weight $\omega_{i\in}[0, 1]$ to reflect its significance. Given a new instance x, each classifier predicated class $C_i(x) \in \{-1, +1\}$ $i = 1$–k_c and the majority vote method generates a final classification by

$$c_{\mathrm{en}}(x) = \mathrm{sign}\left(\sum_{i=1}^{K_{\mathrm{c}}} \omega_i c_i(x)\right). \tag{11}$$

In an ensemble, the disagreement of base classifiers that is referred as the diversity of ensemble is crucial for a successful ensemble of classifiers [22]. It has been shown by much research that if the ensemble does not utilize the diversity of the base classifiers, then no benefit arises from the ensemble.

In seeking a robust ensemble classifier, a set of base classifiers should be ideally selected to maximize both the diversity and accuracy. Several researchers have demonstrated the effectiveness of the selection of compact ensemble committee by feature selection method (e.g. [32,33]). The method proposed here

Inputs: A pool of gene subsets generated by the hybrid sub-sampling algorithm.

K, the number of subsets of genes.

K_c, the number of ensemble members expected.

Outputs: A classification committee.

(1) By using K gene subsets and a training dataset having n_0 instances, a performance matrix $V = [\hat{y}_{ij}]_{n_0 \times k}$ is generated by leave-one-out method, where $\hat{y}_{ij} = c_i(x_j)$ is the predicated class for instance x_j and $E_i = (\hat{y}_{i1}, \hat{y}_{i2}, ...\hat{y}_{in_0})$ is the characteristic vector.

(2) By applying the k-means algorithm on the performance matrix, the candidate classifiers are distributed into K_c clusters.

(3) Rank the classifiers in each cluster according to their misclassification rates

$$e(E_i, Y) = \frac{1}{n_0} \sum_{j=1}^{n_0} d(\hat{y}_{ij}, y_j)$$

where $Y = (y_1, y_2, ...y_{n_0})$ are the true classes for the associated instance.

(4) Select classifiers with minimal error rate in different clusters to construct the classification committee for final classification using Eq.(9).

Fig. 4. The proposed ensemble learning algorithm.

first characterizes the behaviour of the candidate base classifiers, and then selects the appropriate base classifiers to construct a high performance classification committee. As detailed in Fig. 4, the proposed method consists of three steps:

(1) Characterize the behaviour of candidate base classifiers.
(2) Cluster the candidate base classifiers in terms of their classification behaviour.
(3) Select the appropriate subset of base classifiers to construct the classification committee.

The leave-one-out performance is applied to characterize the behaviour of the base classifiers (and the associated gene subsets) in this study. Given a training dataset with n_0 instances $V = (x_1, x_2, \ldots, x_{n_0})$, by applying the leave-one-out evaluation method, one classifier is characterized by a vector

$$E_i = (\hat{y}_{i1}, \hat{y}_{i2}, \ldots, \hat{y}_{in_0}), \tag{12}$$

where $\hat{y}_{ij} = c_i(x_j)$ is the prediction of classifier c_i for sample x_j, and the classifier $c_i(x)$ is constructed by the remaining $(n_0 - 1)$ instances of training data (excluding the sample x_j). This vector E_i is called the characteristic vector of classifier $c_i(x)$ and the associated gene sub-set.

In order to select the diverse base classifiers from the candidate classifiers, the clustering method is employed to group the base classifiers sharing similar performance into the same cluster, and distribute the disagreed classifiers into different clusters. Clustering is a process of grouping objects (the associated candidate classifiers) into a set of disjoint classes, called clusters, so that objects within a cluster have high similarity to each other, while objects in separate clusters are more dissimilar.

The k-means algorithm is used particularly in this paper to perform the clustering. The k-means is an iterative clustering algorithm in which data items are moved among clusters until a desired distribution of data items is reached. Given a set of data items $D = \{t_1, t_2, \ldots, t_m\}$, and the pre-specified number of clusters K, the k-means algorithm iteratively moves the data items among clusters until the dataset has been partitioned into K disjoint subsets which optimize a objective function. For the candidate base classifier clusters, a classifier, $c_i(x)$, is represented by the characteristic vector of classifier (E_i). The k-means is to optimize the function of $\sum_{j=1}^{K}\sum_{c_i \in \hat{C}_j}(E_i - \mu_j)^2$, where the $\hat{C}_j$ represents the jth cluster of classifiers, and μ_j is the centroid of cluster $\hat{C}_j$, i.e. mean of E_i for the classifiers within the cluster $\hat{C}_j$. Given a set of classifier represented by $D = \{E_1, E_2, \ldots, E_m\}$, and the number of clusters K_c, the K-means clustering algorithm is performed based on two basic steps:

Step 1: Assign initial vectors of cluster centroids denoted by $\mu_1, \mu_2, \ldots, \mu_K$;

Step 2: Repeat the following sub-steps until convergence criteria is met.

2.1: Assign each classifier characteristic vector (E_i) to the cluster of classifier that has the closest cluster centroid;

2.2: Calculate new cluster centroid of each cluster with updated cluster members, i.e. calculate the mean of the E_i within the same cluster: $\mu_j = 1/\|\hat{C}_j\|\sum_{c_i \in \hat{C}_j} E_i$.

The convergence criteria could be based on the squared error, or when no (or very small) number of samples are assigned to different clusters.

By clustering the candidate classifiers, the classifiers within the same cluster tend to have high homogeneity or alikeness, while the classifiers from different clusters tend to have high heterogeneity. On the other hand, selecting classifiers from different clusters indirectly increases the diversity of the classification committee. In this study, the base classifier with the best accuracy from each cluster is selected in order to ensure the accuracy of the individual committee members.

3. Results

3.1. The datasets

The proposed method has been evaluated by five publicly available microarray datasets, which are breast cancer microarray data [34], central nervous system (CNS) dataset [35], colon tumor data [8], Leukaemia data [3], and prostate cancer data [36]. The following is a brief introduction about these datasets, while more detailed information can be found from, respectively, the data resources as indicated in the following.

3.1.1. Breast cancer dataset

This gene microarray data contains 97 patient samples, among which 46 samples are from patients who had developed distance metastases within 5 years (labelled as 'relapse'), the remaining 51 samples are from patients who remained healthy from the disease after their initial diagnosis for interval of at least 5 years (labelled as 'non-relapse'). The number of genes in this dataset is 24,481. The detailed information regarding this dataset is available from http://www.rii.com/publications/2002/vantveer.html.

3.1.2. CNS dataset

This microarray data was originally provided by Pomeroy et al. [35]. The dataset employed in the experiments is the dataset *C* used to analyse the outcome of the treatment, which contains 60 patient samples, 21 are survivors (patients who were alive after treatment), and 39 are failures, (patients who succumbed to their disease). There are 7129 genes in the datasets. The associated data resource is http://www.broad.mit.edu/mpr/CNS/.

3.1.3. Colon tumor dataset

This gene expression data contains 62 samples collected from colon-cancer patients. Among them, 40 biopsies are from tumors (labelled as 'negative') and 22 normal biopsies are from healthy parts of the colons of the same patients (labelled as 'positive'). Totally 2000 genes were selected for classifying the colon tumour from normal colon tissues. The corresponding data resource is http://microarray.princeton.edu/oncology/affydata/index.html.

3.1.4. Leukaemia dataset

This dataset was originally provided in [3], which contains the expression levels of 6817 genes of 72 patients, among which, 47 patients suffer from the acute lymphoblastic leukaemia (ALL) and 25 patients suffer from the acute myeloid leukaemia (AML). The leukaemia data resource is http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode = view&paper_id = 43.

3.1.5. Prostate cancer dataset

This microarray dataset is originally provided by Singh et al. [36]. The dataset *A* is used in the experiments in this paper. The data contains 102 samples of which 52 are prostate tumor samples and 50 normal samples. There are 12,600 genes in this dataset. The data resource is http://www-genome.wi.mit.edu/mpr/prostate.

3.2. Performance evaluation method

In this paper, the SVM is employed as the base machine learning and the leave-one-out validation is employed to evaluate the performance of classification. Two typical cross-validation methods (namely *k*-fold cross-validation and leave-one-out validation) have been widely used in machine learning classification evaluation. For cross validation, the original dataset is partitioned into a set of sub-datasets, and the performance of the classifier is evaluated by averaging the accuracy of classifiers trained by different partitions of the dataset and tested by the remaining partitions of the dataset. In the *k*-fold cross-validation, a dataset with m available data samples is partitioned into k sub-sets of size m/k. Each of the subsets is used as the validation set and the combination of other subsets is used as the training set. In leave-one-out cross-validation, each of the m data samples is repeatedly used as the validation set (size 1), and the remaining is used as training set (size $m - 1$).

Comparing to the *k*-fold cross-validation method, the leave-one-out validation method is more applicable for the cases having limited number of data samples. In the experiments, the transformed microarray data in c4.5 format were obtained from the Kent Ridge Bio-medical Data Set Repository.[1] In the cases of

[1] http://sdmc.lit.org.sg/GEDatasets/Datasets.html.

Table 1
Misclassification rate of using different classification methods (%)

	Breast cancer	CNS	Colon	Leukaemia	Prostate
SVM (all genes)	32.9	31.7	19.3	6.94	8.8
SVM (top 50)	24.7	41.7	19.3	6.94	7.8
Bagging (all genes)	20.6	23.3	12.9	4.17	8.8
Bagging (top 50)	21.6	33.3	17.7	5.56	9.8
Boosting (all genes)	25.8	30.0	17.7	6.94	10.8
Boosting (top 50)	21.6	33.3	19.3	6.94	10.8
enSVM (50)	**18.6**	**20.0**	**11.3**	**2.78**	**4.9**

the original microarray datasets containing two separate training and testing datasets, they are combined together to perform the leave-one-out cross-validation as detailed below.

Given a dataset containing N data instances, $(N-1)$ data instances are used to construct a classifier and then apply the remaining one data instance to test this classifier. By repeating this process of successively using each data instances (x_i) as the testing data instance, totally N prediction $e_i = c(x_i)$ $(i = 1\text{–}N)$ are obtained. The performance of the classifier is then measured by the average misclassification rate:

$$\mathrm{Er} = \frac{1}{N}\sum_{i=1}^{N} \delta(e_i, y_i), \tag{13}$$

where y_i is the true class label for instance x_i, and

$$\delta(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y. \end{cases}$$

3.3. Experimental results

Four sets of experiments have been performed to evaluate the effectiveness of the proposed method. The machine learning algorithm for base classifier generation used in this paper is the support vector machine. Particularly, the OSU SVM Classifier Matlab Toolbox[2] is employed in the experiments.

The first experiment is intended to compare the performances of the proposed ensemble learning method (denoted as enSVM) to that of single SVM classifier (denoted as SVM), the bagging ensemble learning (denoted as Bagging) and boosting ensemble learning (denoted as Boosting). In this experiment, the number of genes used is set to be 50, i.e., the top 50 genes (the 50 most significant genes) are used in single SVM, bagging and boosting while the randomly sampled gene subsets (with 50 genes) are used in enSVM. Furthermore, the number of candidate classifiers is set to be $K = 200$ and the number of classification committee members is $K_c = 25$.

Table 1 and Fig. 5(a) report the misclassification rate of these methods for the associated dataset. To reduce the influence of gene selection on the classification performance, another comparison has been performed in which all genes are used to train the single SVM, Bagging and Boosting, while the randomly sampled 50 genes are still used to train the enSVM. The results are shown in Table 1 and Fig. 5(b).

[2] Which is available from http://www.ece.osu.edu/~maj/osu_svm/.

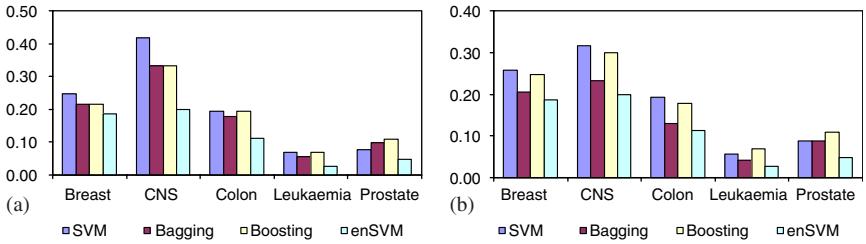

Fig. 5. Classification performances of using different classification methods.

These results clearly show that the enSVM method not only performs better than signal SVM classifier but also outperform these two widely used ensemble methods (Bagging and Boosting). Furthermore, it has been shown that the bagging learning method performs better than boosting method. This observation has also been recognized by other researches (e.g. [12]). One explanation for the observations, based on the experiments performed in this study, is due to the limited number of training samples existing in microarray data, and the nature of boosting learning itself. The boosting ensemble learning induces base classifiers sequentially, and terminates the induction when training data has been perfectly classified. Given the fact of limited training instances in the dataset, the boosting method always terminates before the given number of classifier has been obtained and the base classifiers generated in the latter stages of boosting focuses on learning from only a few instances.

The second experiment is performed to evaluate the effectiveness of the proposed method under different numbers of the genes. In this experiment, the numbers of genes are set to be $n_G = 30$, 50, 75, 100, 150, 200 and 300. Given $\alpha = 0.4$, $\beta = 2$, the sizes of G_{top} are $n_{top} = 12$, 20, 40, 60, 80, 120, and the corresponding number of genes $n_1 = 6$, 10, 15, 20, 30, 40, 60, $n_2 = 24$, 40, 60, 80, 120, 160, 240 respectively. Table 2 and Fig. 6 present the results for the associated methods under the corresponding number of genes. These results show that the enSVM method produce the best classification accuracy comparing to the use of single SVM, bagging and boosting methods under the varying numbers of genes.

The third experiment is to evaluate the effectiveness of the proposed method under variable numbers of base classifiers. In this experiment, only the bagging method is compared. The boosting method is not comparable as it sometimes terminates the induction of base classifiers before the given numbers of base classifiers have been obtained when all the training data have been perfectly classified.

Table 3 and Fig. 7 show the results of the proposed ensemble learning method based on 15, 25, 35, 45, 55, 65, 75 base classifiers. In this experiment, the number of genes is set to be 50. It is clearly shown that the proposed method performs much better than the Bagging ensemble learning. This observation indicates that, under the context of microarray data, sampling in gene space (enSVM) produces more appropriate base classifiers than sampling in instance space (Bagging).

The fourth experiment is to verify the effectiveness of the proposed base classifier selection for enhancing the performance of ensemble learning. In the experiment, the number of genes is varied, and the number of base classifiers is set to be 25, and the number of candidate classifiers is 200, i.e. to select 25 base classifiers from the 200 candidate classifiers. The performance of the proposed method (based on clustering candidate classifiers, denoted as clustering) is compared with the results of: (1) randomly selecting base classifiers (denoted as random), (2) selecting the best accurate classifiers (denoted as top). The results for these three methods are shown in Table 4 and Fig. 8. From these experimental results,

Table 2
Misclassification rates under varying number of genes (%)

Dataset	Method	Number of genes							
		30	50	75	100	150	200	300	500
Breast	SVM	24.7	21.6	22.7	23.7	22.7	23.7	21.6	23.7
	Bagging	22.7	24.7	22.7	21.6	19.6	25.8	21.6	21.6
	Boosting	23.7	23.7	27.8	26.8	25.8	24.7	28.9	26.8
	enSVM	**16.5**	**18.6**	**17.5**	**16.5**	**17.5**	**17.5**	**19.6**	**20.6**
CNS	SVM	30.6	30.0	20.0	26.7	21.6	23.3	25.0	28.3
	Bagging	26.7	23.3	25.0	21.7	21.6	26.7	21.6	30.0
	Boosting	26.7	33.3	28.3	23.3	33.3	25.0	33.3	33.3
	enSVM	**20.0**	**20.0**	**18.3**	**20.0**	**20.0**	**21.7**	**18.3**	**21.7**
Colon	SVM	19.4	25.8	20.9	14.5	14.5	17.7	20.9	14.5
	Bagging	17.7	22.6	14.5	12.9	14.5	17.7	16.1	12.9
	Boosting	19.4	22.6	24.2	17.4	22.6	22.6	20.9	12.9
	enSVM	**11.3**	**11.3**	**11.3**	**12.9**	**12.9**	**12.9**	**11.3**	**12.9**
Leukaemia	SVM	6.94	6.94	5.56	5.56	8.33	6.94	8.33	6.94
	Bagging	6.94	5.56	5.56	4.17	6.94	4.17	6.94	5.55
	Boosting	8.33	6.94	6.94	6.94	6.94	5.55	6.94	6.94
	enSVM	**2.78**	**2.78**	**1.39**	**2.78**	**1.39**	**2.78**	**1.39**	**2.78**
Prostate	SVM	12.7	7.8	5.8	7.8	8.8	10.8	7.8	5.8
	Bagging	7.8	6.9	5.8	5.8	6.9	6.9	5.8	5.8
	Boosting	9.8	10.8	7.8	10.8	8.8	4.9	7.8	5.8
	enSVM	**5.8**	**4.9**	**5.8**	**5.8**	**5.8**	**2.9**	**5.8**	**5.8**

it is clearly shown that the proposed method has successfully reduced the misclassification rates for all these four datasets. Moreover, the experimental results show that selecting the top base classifiers produce worse classification accuracy than randomly selecting the base classifiers. This observation illustrates that selecting the best group base classifiers does not necessary produce good classification accuracy, which demonstrates the importance of selecting diverse base classifier in the ensemble learning.

3.4. Discussion

The principal objective of this experimental study is to verify the effectiveness of the ensemble learning based on the gene sub-sampling and classifier clustering, by comparing with the performance of single classifier and two widely used instance re-sampling-based ensemble learning methods (bagging and boosting). The comparison has been performed based on the common experimental conditions, and no additional data pre-processing has been performed before the classifier construction. The experimental results have illustrated that, compared to the conventional machine learning techniques (single classifier,

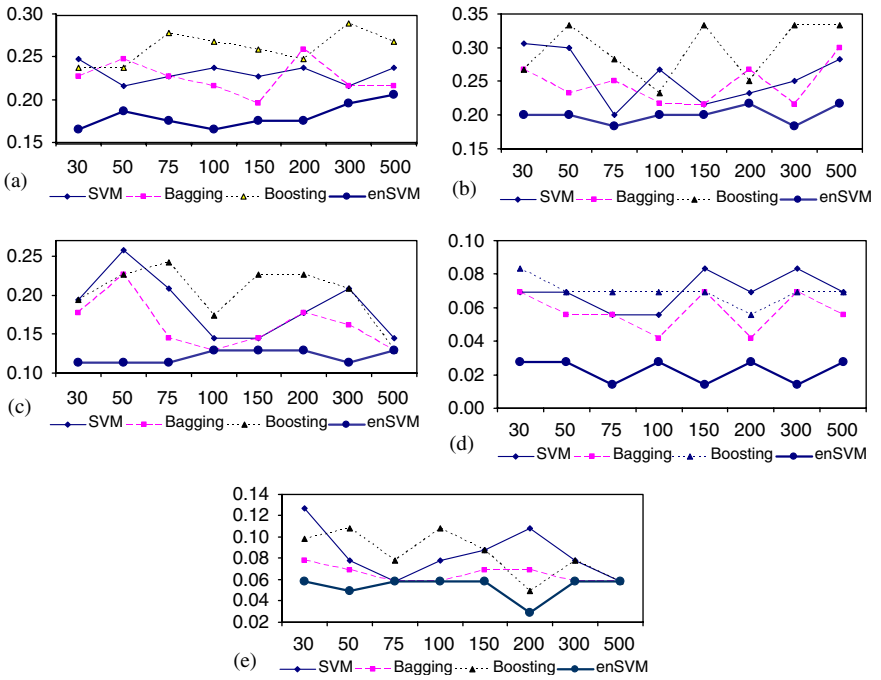

Fig. 6. Classification performances under varying number of genes. (a) Breast, (b) CNC, (c) Colon, (d) Leukaemia, (e) Prostate.

Table 3
Misclassification rates of using different number of base classifiers (%)

Dataset	Method	Number of base classifiers						
		15	25	35	45	55	65	75
Breast	Bagging	22.7	20.6	24.7	23.7	23.7	23.7	22.6
	enSVM	**17.5**	**15.5**	**20.6**	**19.6**	**19.6**	**18.6**	**19.6**
CNS	Bagging	21.7	23.3	26.7	21.7	23.3	23.3	20.0
	enSVM	**11.7**	**16.7**	**18.3**	**18.3**	**16.7**	**20.0**	**21.7**
Colon	Bagging	20.9	19.4	24.2	22.6	20.9	22.6	22.6
	enSVM	**9.6**	**12.9**	**12.9**	**11.3**	**12.9**	**12.9**	**11.3**
Leukaemia	Bagging	4.17	5.56	5.56	4.17	5.56	4.17	5.56
	enSVM	**2.78**	**2.78**	**1.39**	**2.78**	**2.78**	**1.39**	**2.78**
Prostate	Bagging	6.9	6.9	5.9	6.9	6.9	5.9	6.9
	enSVM	**3.9**	**3.9**	**3.9**	**5.9**	**5.9**	**2.9**	**3.9**

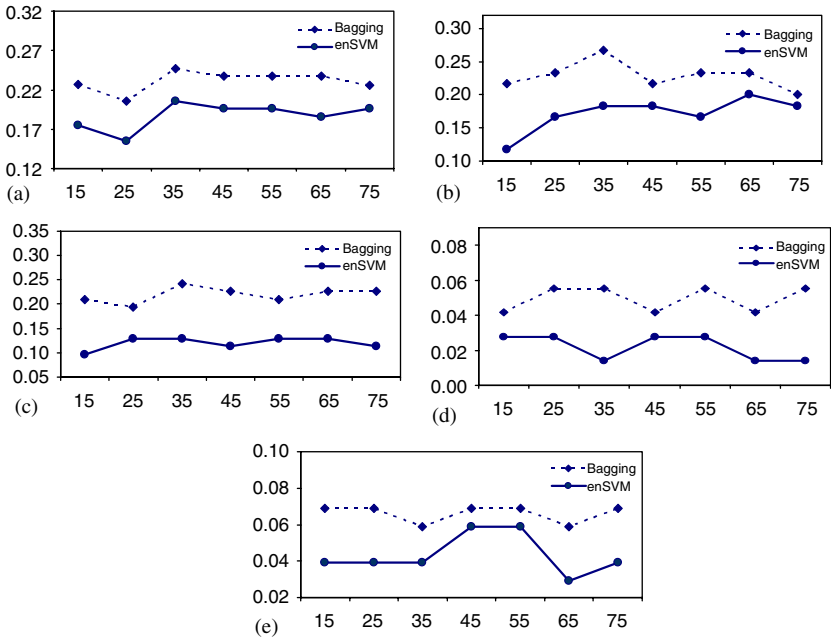

Fig. 7. Classification performances under varying number of base classifiers. (a) Breast, (b) CNC, (c) Colon, (d) Leukaemia, (e) Prostate.

bagging and boosting), the consistent and significant improvements of accuracy have been produced by the method proposed under varying experimental conditions.

Furthermore, the results obtained in this study are competitive with the existing methods in literature. In [37], five different ensemble machine learning methods had been compared, and the best accuracy obtained in [37] for the colon tumor dataset is 14.52%. The enSVM reduced misclassification rate to a range between 11.3% and 12.9% which involve less than 300 genes.

In addition, the training/testing evaluation method has also been employed to examine the effectiveness of the proposed approach. Two separate datasets are, respectively, used for training and testing. While the Prostate dataset originally contains 102 samples as a whole dataset, the Breast, CNS, Colon and Leukaemia datasets originally consist of separate training and testing datasets, which, respectively, contain 78-training (19-testing), 35(25), 37(25), 38(34) samples. For the Prostate data, it is randomly partitioned into two datasets (the training dataset has 70 samples while the testing datasets contains 32 samples). The experiment results are shown in Table 5 and Fig. 9. In the experiment, the parameters are set to be $N_G = 50$, $K_c = 25$, and $K = 200$. These results clearly show the enSVM outperforms the SVM, Bagging and Boosting methods.

The accuracy of enSVM in classifying the Leukaemia testing dataset can be directly compared to the results obtained originally in [3], in which 29 instances were correctly classified, and the improved results obtained in [18] applying SVM classifier, in which there are between 30 and 32 correct classification, while the enSVM correctly classify 33 of the 34 testing instances. This result is competitive with the results obtained in [37].

Table 4
Misclassification rates of using difference base classifier selection methods (%)

Dataset	Methods	Number of genes							
		30	50	75	100	150	200	300	500
Breast	Random	22.7	20.6	21.7	20.6	24.7	21.7	24.7	24.7
	Top	31.96	30.9	31.96	32.99	34.0	29.9	28.9	27.8
	Clustering	**17.5**	**18.6**	**17.5**	**17.5**	**16.5**	**19.6**	**20.6**	**21.7**
CNS	Random	21.7	20.0	23.3	21.7	20.0	20.0	21.7	26.7
	Top	38.3	28.3	31.7	30.0	31.7	31.7	26.7	35.0
	Clustering	**15.0**	**13.3**	**16.7**	**15.0**	**16.7**	**16.7**	**20.0**	**21.7**
Colon	Random	14.5	12.9	14.5	14.5	12.9	14.5	12.9	19.4
	Top	25.8	25.8	19.4	20.97	19.4	17.7	25.8	24.2
	Clustering	**11.3**	**11.3**	**11.3**	**11.3**	**11.3**	**12.9**	**11.3**	**14.5**
Leukaemia	Random	5.55	4.17	5.55	4.17	4.17	4.17	4.17	5.55
	Top	6.94	6.94	6.94	5.55	5.55	5.55	6.94	6.94
	Clustering	**2.78**	**2.78**	**1.39**	**2.78**	**1.39**	**2.78**	**1.39**	**2.78**
Prostate	Random	6.9	5.9	6.9	5.9	5.9	6.9	7.8	6.9
	Top	14.7	8.8	7.8	8.8	7.8	9.8	10.8	9.8
	Clustering	**4.90**	**2.94**	**4.90**	**3.92**	**4.90**	**3.92**	**4.90**	**4.90**

4. Conclusions

Microarray data analysis is challenging the traditional machine learning techniques due to the availability of a limited number of training instances and the existence of large number of genes, together with the inherent various uncertainties. Conventional machine learning techniques rely too much on the gene selection, which may cause irrecoverable misclassification error when the gene selection criterion used is not suitable to the data under analysis, especially when there are limited labelled instances and the dataset is associated with high uncertainties.

The objective of this study is to develop a generic machine learning approach that can address the issues existing in microarray data analysis and reduce its degree of relying on the gene selection, and produce a robust classification for microarray data. The principal idea behind this study is to involve multiple classifiers constructed by using different subsets of genes so as to fuse the information from diverse gene subsets. The approach presented in this paper consists of three basic steps: gene sub-sampling, generation of candidate base classifiers, and construction of an effective ensemble committee.

A set of experiments has been performed to evaluate the effectiveness of the proposed method and related aspects including gene sub-sampling, and base classifier selection. Based on the experimental results, we can draw the following important conclusions:

(1) Under the context of microarray data analysis, the gene sub-sampling-based ensemble learning methods provide more appropriate approach to implement robust classification. The experimental

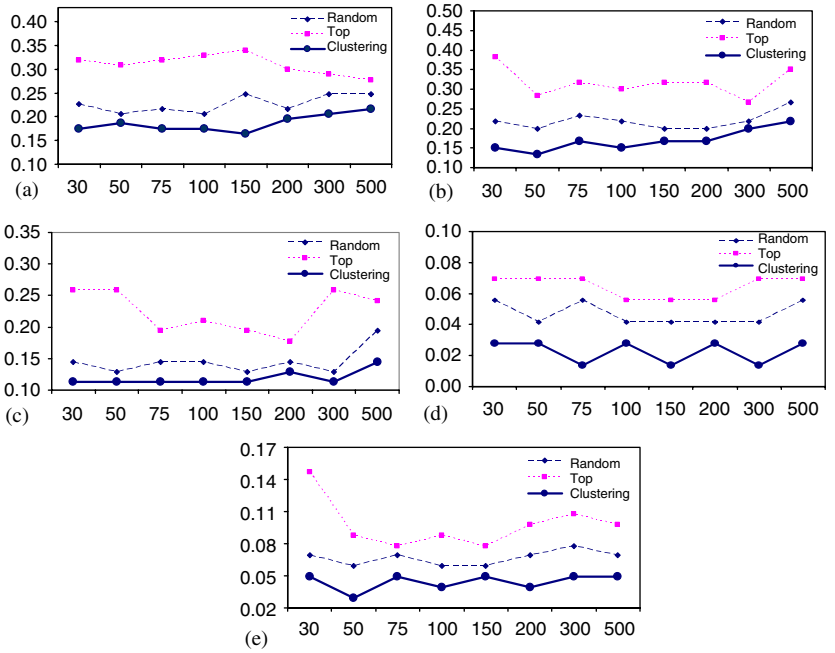

Fig. 8. Classification performances of using different base classifier selection methods. (a) Breast, (b) CNC, (c) Colon, (d) Leukaemia, (e) Prostate.

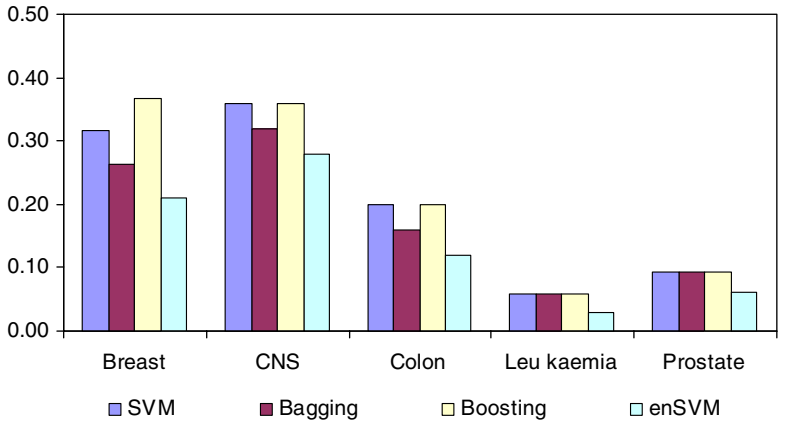

Fig. 9. Misclassification rates for testing dataset.

results show that the gene sub-sampling-based ensemble learning always outperforms the instance re-sampling ensemble learning method such as bagging and boosting methods.

(2) This research demonstrates the importance of selecting appropriate base classifiers from the candidate classifiers for constructing an effective classification committee. This paper presents a method to characterize the behaviour of a base classifier, based on which the candidate classifiers can be clustered

Table 5
Misclassification rates for testing dataset (%)

	Breast	CNS	Colon	Leukaemia	Prostate
SVM	31.6	36	20	5.88	9.38
Bagging	26.3	32	16	5.88	9.38
Boosting	36.8	36	20	5.88	9.38
enSVM	21.1	28	12	2.94	6.25

and selected so as to increase the diversity of ensemble members. Experimental results also illustrate the effectiveness of this proposed method for the base classifier selection.

(3) It has been demonstrated by the experimental results, that the proposed method outperforms the single classifiers and the conventional ensemble learning methods (bagging and boosting) under various experimental setup (varying number of genes, and different number of base classifiers).

The ensemble learning approach proposed in this paper is generic. It is robust to the selection of genes and can be applied on different base classifiers. These characteristic of the proposed method suggest the proposed method has great potential for the developments of generic platform for microarray data classification.

References

[1] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA 95 (25) (1998) 14863–14868.
[2] M.P. Brown, W.N. Grundy, D. Lin, et al., Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. USA 97 (1) (2000) 262–267.
[3] T.R. Golub, D.K. Slonim, P. Tamayo, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[4] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.
[5] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, 2000.
[6] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1/3) (2002) 389–422.
[7] R. Blanco, P. Larrañaga, I. Inza, B. Sierra, Gene selection for cancer classification using wrapper approaches, Int. J. Pattern Recognition Artif. Intell. 18(8) (2004) 1373–1390.
[8] U. Alon, N. Barkai, D.A. Notterman, K. Gish, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 96 (12) (1999) 6745–6750.
[9] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, et al., Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc. Natl. Acad. Sci. USA 96 (6) (1999) 2907–2912.
[10] D. Jiang, C. Tang, A. Zhang, Cluster analysis for gene expression data: a survey, IEEE Trans. Knowledge Data Eng. 16 (11) (2004) 1370–1386.
[11] D. Berrar, B. Sturgeon, I. Bradbury, W. Dubitzky, Microarray data integration and machine learning techniques for lung cancer survival prediction, Proceedings of the CAMDA-2003, 2003.
[12] A.C. Tan, D. Gilbert, Ensemble machine learning on gene expression data for cancer classification, Appl. Bioinform. 2 (Suppl. 3) (2003) 75–83.
[13] L. Li, C.R. Weinberg, et al., Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, Bioinformatics 17 (2001) 1131–1142.

[14] S.B. Cho, H.H. Won, Machine learning in DNA microarray analysis for cancer classification, Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics, 2003.
[15] S.B. Cho, H.H. Won, Neural network classifiers and gene selection methods for microarray data on human lung adenocarcinoma, CAMDA 2003 Conference, 2003.
[16] N. Friedman, M. Linial, et al., Using bayesian networks to analyze expression data, J. Comput. Biol. 7 (3/4) (2000) 601–620.
[17] S. Mukherjee, P. Tamayo, J.P. Mesirov, et al., Support vector machine classification of microarray data, Technical Report 182, AI Memo 1676, CBCL, 1999.
[18] T.S. Furey, N. Cristianini, et al., Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (10) (2000) 906–914.
[19] K.R. Coombes, W.E. Highsmith, et al., Identifying and quantifying sources of variation in microarray data using high-density cDNA membrane arrays, J. Comput. Biol. 9 (4) (2002) 655–669.
[20] X. Wang, M.J. Hessner, Y. Wu, et al., Quantitative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction, Bioinformatics 19 (11) (2003) 1341–1347.
[21] W. Li, Y. Yang, How many genes are needed for a discriminant microarray data analysis?, CAMDA 2000 Conference, 2000.
[22] T.G. Dietterich, Ensemble methods in machine learning, Proceedings of the First International Workshop on Multiple Classifier Systems, 2000, pp. 1–15.
[23] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001.
[24] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, Mach. Learn. 36 (1–2) (1999) 105–139.
[25] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[26] Y. Freund, R.E. Schapire, A Decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[27] R.E. Schapire, A brief introduction to boosting, Proceedings of the 16th International Joint Conference on Artificial Intelligence, 1999, pp. 1401–1406.
[28] T.K. Ho, The random subspace method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 832–844.
[29] T.K. Ho, C4.5 Decision forests, Proceedings of the 14th International Conference on Pattern Recognition, Brisbane, Australia, 1998, pp. 545–549.
[30] J. Cao, M. Ahmadi, M. Shridhar, Recognition of handwritten numerals with multiple feature and multistage classifier, Pattern Recognition 28 (2) (1995) 153–160.
[31] V. Tusher, R. Tibshirani, G. Chu, Significance analysis of microarrays applied to the ionizing radiation response, Proc. Natl. Acad. Sci. USA 98 (9) (2001) 5116–5121.
[32] E. Xing, M. Jordan, and R. Karp, Feature selection for high-dimensional genomic microarray data, Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 601–608.
[33] L. Yu, H. Liu, Redundancy based feature selection for microarray data, Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2004, pp. 737–742.
[34] L.J. van't Veer, H. Dai, et al., Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–536.
[35] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, Prediction of central nervous system embryonal tumour outcome based on gene expression, Letters to Nature, Nature 415 (2002) 436–442.
[36] D. Singh, P.G. Febbo, et al., Gene expression correlates of clinical prostate cancer behaviour, Cancer Cell 1 (2002) 203–209.
[37] M. Dettling, P. Bühlmann, Boosting for tumor classification with gene expression data, Bioinformatics 19 (9) (2003) 1061–1069.

Yonghong Peng received the B.Sc, M.Sc. and Ph.D. degrees in 1989, 1992, and 1995, respectively, from the East China Jiaotong University, the Southeast University, and the South China University of Technology. The subject of his Ph.D. dissertation is about neural network and fuzzy logic-based intelligent control technologies and systems. He has been working as a lecturer and then associate professor at the South China University of Technology, China since 1995 and 2000, respectively. He worked at

the City University of Hong Kong from 1998 to 1999 for a project on wavelets and hybrid neural networks, and was working for a Europe project on machine learning/data mining and meta-learning (www.metal-kdd.org) in the Department of Computer Science of University of Bristol from 2001 to 2002.

He is currently a lecturer in computer science at the University of Bradford, UK. His research areas include Machine Learning and Data Mining, and Bioinformatics. He is a member of the IEEE, member of Computer Society, and member of SMC. He has published over 50 research papers in the related areas. He has been a member of the programme committee of several international conferences and workshops. He referees papers for several journals such as the IEEE Transactions on Systems, Man and Cybernetics (part C), IEEE Transactions on Evolutionary Computation, Journal of Fuzzy Sets and Systems, Journal of Bioinformatics, and Journal of Data Mining and Knowledge Discovery, and has refereed papers for several conferences.