



Progressive subspace ensemble learning

Zhiwen Yu^{a,*}, Daxing Wang^{a,*}, Jane You^b, Hau-San Wong^c, Si Wu^a, Jun Zhang^a, Guoqiang Han^a

^a School of Computer Science and Engineering, South China University of Technology, China

^b Department of Computing, Hong Kong Polytechnic University, Hong Kong

^c Department of Computer Science, City University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:

Received 16 July 2015

Received in revised form

15 May 2016

Accepted 19 June 2016

Available online 21 June 2016

Keywords:

Ensemble learning

Classifier ensemble

Random subspace

AdaBoost

Decision tree

ABSTRACT

There are not many classifier ensemble approaches which investigate the data sample space and the feature space at the same time, and this multi-pronged approach will be helpful for constructing more powerful learning models. For example, the AdaBoost approach only investigates the data sample space, while the random subspace technique only focuses on the feature space. To address this limitation, we propose the progressive subspace ensemble learning approach (PSEL) which takes into account the data sample space and the feature space at the same time. Specifically, PSEL first adopts the random subspace technique to generate a set of subspaces. Then, a progressive selection process based on new cost functions that incorporate current and long-term information to select the classifiers sequentially will be introduced. Finally, a weighted voting scheme is used to summarize the predicted labels and obtain the final result. We also adopt a number of non-parametric tests to compare PSEL and its competitors over multiple datasets. The results of the experiments show that PSEL works well on most of the real datasets, and outperforms a number of state-of-the-art classifier ensemble approaches.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays, more and more researchers pay attention to ensemble learning [1–10,62–67], due to its high performance and robustness in various machine learning tasks. A number of ensemble learning algorithms have been proposed in recent years, such as hybrid adaptive ensemble learning [1], ensemble learning based on random linear oracle [2], bagging [3], random subspace [4], classifier ensemble based on neural networks [5,6], boosting [7], fuzzy classifier ensemble [8], rotation forest [9], random forest [10], and heterogeneous ensemble of classifiers [68,69]. Ensemble learning approaches have been successfully applied in the field of data mining [11,12], text categorization [13,14], bioinformatics [15,16] and face image classification [17,18], along with many other tasks. In the field of data mining, Polikar et al. [11] introduced a new classifier ensemble framework named Learn++ MF to address the missing value problem in data mining. It classifies missing values with multiple classifiers. Galar et al. [12] designed a new classifier ensemble algorithm named EUSBoost which incorporates random sub-sampling with boosting in the ensemble to handle the data imbalance problem. In the field of text categorization, Liu et al. [13] designed a novel classifier

ensemble framework named BAM-Vote Box, and introduced a new feature selection algorithm for text categorization task. Saeedian et al. [14] designed a new spam detection algorithm using the ensemble learning framework based on clustering and weighted voting. In the field of bioinformatics, Liu et al. [15] introduced a new ensemble learning algorithm which adopts mutual information to search for important gene subsets, and then use the ensemble method to classify cancer gene expression data. Plumpton et al. [16] designed a new ensemble framework to classify streaming functional magnetic resonance images and measure brain activities. In the field of face image classification, Connolly et al. [17] introduced a face recognition framework which combines multiple neural network classifiers and produces a progressive ensemble learning algorithm for face recognition from video data. Zhang et al. [18] introduced RDA, an extension of LDA, to analyze face image data in a subspace. Ahmadvand et al. [57] adopted the ensemble classifier for MRI brain image segmentation. In summary, both theoretical and empirical analysis show that under certain conditions, ensemble learning achieves better performance and robustness than single classifiers, due to its capability to integrate more information encapsulated in multiple learners.

While there are different kinds of ensemble learning approaches, most of them only consider the distribution of the data in the sample space, or the distributions of the attributes in the

* Corresponding authors.

E-mail address: zhwyu@scut.edu.cn (Z. Yu).

feature space. However, the two sets of distributions should be combined to achieve a better result in most of the cases. To address this limitation, we propose a progressive subspace ensemble learning approach (PSEL) which explores both the data sample space and the feature space at the same time. Specifically, PSEL adopts the random subspace technique to generate a set of subspaces. Each subspace is used to train a classifier in the original ensemble. Then, a progressive selection process is adopted to select the classifiers based on two cost functions which incorporate current and long-term information. Finally, a weighted voting scheme is used to combine the predicted results from individual classifiers of the ensemble, and generate the final result. The properties of PSEL are analyzed theoretically. We also adopt a number of non-parametric tests to compare PSEL and its competitors on multiple datasets. Our experiments show that PSEL works well on the real datasets, and outperforms most of the state-of-the-art classifier ensemble approaches.

The contribution of the paper is twofold. First, the progressive subspace ensemble learning approach is proposed, which not only explores the data sample space, but also takes the feature space into consideration. Second, two cost functions which incorporate current and long-term information are designed to perform the progressive selection process.

The remainder of the paper is organized as follows. Section 2 discusses previous works related to ensemble learning. Section 3 describes the progressive subspace ensemble learning approach and the progressive selection process. Section 4 analyzes the proposed algorithm theoretically. Section 5 experimentally evaluates the performance of our proposed approach. Section 6 presents our conclusion and future work.

2. Related work

As an important branch of ensemble learning, classifier ensemble has drawn the attention of researchers from many fields. In recent years, classifier ensemble approaches based on different viewpoints have been developed. Some of them focus on generating new ensembles [19–24]. For example, Yu et al. [19] introduced a graph based semi-supervised ensemble algorithm which generates the ensemble by performing multiple rounds of dimensionality reduction. Ye et al. [20] designed an extension of the random forest algorithm which ensures that enough good features are selected in each subspace. Guo et al. [21] applied the rough set theory for dimensionality reduction in the ensemble framework. Zhang et al. [22] used the rotation space technique to increase the diversity of the classifiers in the random forest approach. Zhang et al. [23] incorporated label information into canonical correlation analysis and designed a new ensemble algorithm. Hllermeier et al. [24] developed a label ranking based voting framework, and performed theoretical analysis related to weighted voting. Zhang et al. designed a random vector functional link (RVFL) network based classifier [58] and a new oblique decision tree ensemble [59], which achieve good performance on the dataset from UCI machine learning repository. On the other hand, some researchers focus on the ensemble property. For example, Kuncheva [25] studied ensembles with Kappa-error diagrams. Wang et al. [26] analyzed the generalization and fuzziness properties of an ensemble. Other researchers consider the combination of different classifier ensemble approaches. For example, Nanni et al. [68] studied different combinations of ensemble techniques to improve the performance of AdaBoost. They also explored a heterogeneous ensemble which considers the combination of different classification approaches [69]. Zhang et al. [70] investigated the combination of rotation forest and AdaBoost. A survey of the different classifier ensemble techniques have been included in [34].

In addition, ensemble learning is applied in a number of research fields. For example, Rasheed et al. [27] used the classifier ensemble approach for electromyographic signal decomposition. Tian et al. [28] designed a classifier ensemble consisting of Haar-like and shapelet components for pedestrian detection. Guo et al. [29] proposed a two-stage pedestrian detection algorithm using AdaBoost and SVM. There are also applications in breast tumor discrimination [30], protein structural class prediction [31], network intrusion detection [32], and handwritten digit recognition [33] as well. Daliri [54] applied an extreme learning machine-based ensemble classifier for breast tissue classification. In addition, he also integrated multiple feature selection methods for cognitive state prediction in functional magnetic resonance imaging (fMRI) [55], and performed feature selection by applying swarm optimization and support vector machine in medical informatics [56].

While these ensemble learning approaches lead to performance improvement when compared to conventional techniques, most of them focus on the sample space or the feature space separately. Some of these works [64] consider exploring the sample space and the feature space at the same time.

Sometimes not all classifiers need to be selected for the ensemble. Some researchers suggest combining a suitable subset of the classifiers to pursue better performance [35–39]. For example, Hu et al. [35] applied the rough set theory to generate base classifiers, and then used an accuracy based search and prune method to select classifiers. Cruz et al. [36] introduced a dynamic framework which measures the competency of each classifier with a meta-learning approach, and select classifiers from the ensemble. Xiao et al. [37] developed a new dynamic classifier selection algorithm which considers both the accuracy and the diversity. Santos et al. [38] introduced an overproduce-and-choose strategy based ensemble member selection approach. Yang et al. [39] introduced a classifier ensemble selection method which also considers the accuracy and the diversity of the classifiers.

We focus on the random subspace technique [40], which has been successfully used for human detection [41], image retrieval [42], functional magnetic resonance imaging classification [43], traffic flow forecasting [44], and so on. The random subspace technique is one of the most popular ensemble learning approaches, which makes use of a combination of different subspaces to increase the diversity of the ensemble, and improve the performance of the ensemble learning approach. Fig. 1 provides an overview of the random subspace (RS) approach. Specifically, RS first generates B random subspaces using a predefined sampling rate $\kappa \in [0, 1]$. Then, the training samples are transformed from the full space to the corresponding random subspaces. The new training samples in each subspace are used to train a classifier. B trained classifiers corresponding to the B random subspaces will be obtained. Next, given a testing dataset, RS will obtain a set of predicted results for the samples from the individual classifiers in the ensemble. Finally, the majority voting scheme is adopted to

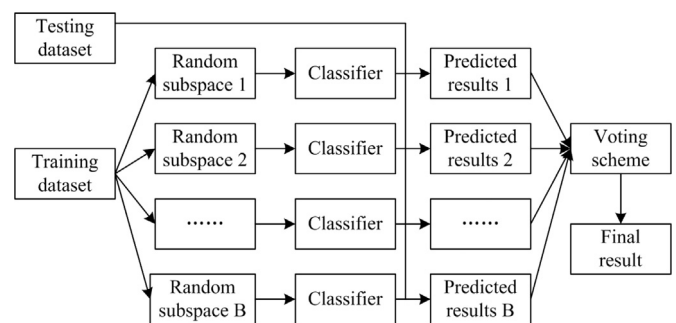


Fig. 1. An overview of the random subspace approach.

obtain the final result. While the random subspace technique is easy to implement and has many successful applications, it has two limitations: (1) not all the subspaces in the ensemble contribute to the final result. The search space for the combination of the subspaces is very large. How to find the optimal combination of the subspaces is an interesting problem which is worth exploring. (2) RS does not explore the data sample space by updating the weights of the training samples progressively. The classifiers in the ensemble generated by RS do not have any relationships among them.

In this paper, a progressive subspace ensemble learning approach (PSEL) is proposed to address the limitation of the random subspace technique. PSEL not only adopts a progressive selection process based on two newly designed cost functions which incorporate current and long-term information to search for the optimal combination of the classifiers, but also updates the weights of training samples in each iteration to establish the relationships of the classifiers in the ensemble. PSEL will select a subset of suitable subspaces from the original ensemble using a progressive selection process to generate a new ensemble with more diversity, while the random subspace approach directly uses all the subspaces in the original ensemble to train the ensemble classifier.

3. Progressive subspace ensemble learning

3.1. An overview of progressive subspace ensemble learning

Fig. 2 provides an overview of the progressive subspace ensemble learning approach, while Algorithm 1 shows a flow chart of the progressive subspace ensemble learning approach. Given a training set T_r with l pairs of training samples $T_r = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$ (where l is the number of training samples, x_i ($i \in \{1, \dots, l\}$) is the training sample, y_i is the label of the training sample, $y_i \in \{1, 2, \dots, k\}$, k is the number of classes), and each sample x_i consists of m attributes $\{x_{i1}, x_{i2}, \dots, x_{im}\}$, the progressive subspace ensemble learning algorithm (PSEL) first generates a set of random subspaces $\hat{S} = \{S_1, S_2, \dots, S_B\}$ (B is the number of subspaces).

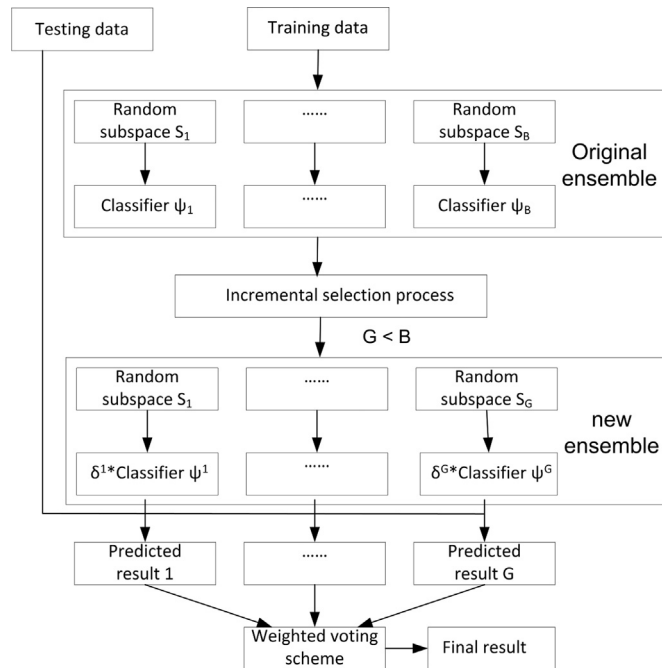


Fig. 2. An overview of the progressive subspace ensemble learning approach.

Algorithm 1. Progressive subspace ensemble learning approach.

Require:

Input: the training set T_r and the testing set T_e ;

Procedure

- 1: Original ensemble generation;
 - 2: Generate B random subspaces $\{S_1, S_2, \dots, S_B\}$;
 - 3: Generate the corresponding classifiers $\psi_1, \psi_2, \dots, \psi_B$;
 - 4: Call Progressive selection process in Algorithm 2;
 - 5: New ensemble generation;
 - 6: Generate G random subspaces $\{S_1, S_2, \dots, S_G\}$ ($G < B$);
 - 7: Generate the corresponding classifiers $\psi^1, \psi^2, \dots, \psi^G$;
 - 8: Label prediction for the samples in T_e ;
 - 9: Weighted voting scheme for the final results;
- Output:** the labels of the samples in T_e .

Then, it trains a set of decision tree classifiers $(\psi_1, \psi_2, \dots, \psi_B)$ using the set of random subspaces $(\{S_1, S_2, \dots, S_B\})$. Specifically, the decision tree classifier ψ_j ($j \in \{1, \dots, B\}$) is constructed based on the information gain χ of the attribute, which is defined as follows:

$$\chi(T_h, a) = H(T_h) - \sum_{v \in a} \frac{|T_{hv}|}{|T_h|} H(T_{hv}, v) \quad (1)$$

$$H(T_h) = -p_h^+ \log_2 p_h^+ - p_h^- \log_2 p_h^- \quad (2)$$

$$H(T_h, v) = -p_{hv}^+ \log_2 p_{hv}^+ - p_{hv}^- \log_2 p_{hv}^- \quad (3)$$

where $\chi(T_h, a)$ denotes the information gain for the h th node with size $|T_h|$ for the attribute a , H is the entropy, p_h^+ and p_h^- denote the proportion of positive and negative samples associated with the node T_h , respectively, and T_{hv} corresponds to the subnode of the node T_h for the attribute value $a = v$. The attribute corresponding to the largest information gain will be selected as the decision node. The decision tree repeats the above process until the decision nodes and leaf nodes are obtained. The classification rules are represented by the paths from root to leaves. The progressive subspace ensemble learning algorithm will associate each classifier with an accuracy value $(AC_j$ ($j \in \{1, \dots, B\}$)), which is calculated by the corresponding decision tree on the training set T_r .

Next, the progressive selection process, as shown in Algorithm 2, is applied to the original ensemble, and G ($G < B$) random subspaces and the corresponding classifiers will be selected based on the two newly designed cost functions. Compared with the classifiers in the original ensemble, the classifiers in the new ensemble are sorted in sequential order based on the progressive selection process.

Algorithm 2. Progressive selection process.

Require

Input: the training sample set $X = (x_1, x_2, \dots, x_l)$;

the corresponding label set $Y = (y_1, y_2, \dots, y_l)$;

the loss function $\theta(\psi(X), y, i) = e^{-y_i \psi(x_i)}$, $i \in \{1, \dots, l\}$;

a set of random subspaces $\hat{S} = \{S_1, S_2, \dots, S_B\}$

the corresponding classifiers $\hat{\Psi} = \{\psi_1, \psi_2, \dots, \psi_B\}$, Ψ :

$X \rightarrow [-1, 1]$;

the empty ensemble $\Psi(X)$;

Procedure

- 1: Initial weights $\omega_1^1 = \dots = \omega_l^1 = \frac{1}{l}$ for all the samples;
- 2: **For** g in $1, \dots, G$:

3: **If** $g=1$,
 4: Find the first classifier $\psi^1 = \operatorname{argmax}_{\psi \in \hat{\mathcal{P}}} \{AC_1, AC_2, \dots, AC_B\}$;
 5: The weighted sum error for misclassified points
 $\epsilon^1 = \sum_i \omega_i^1 \theta(\psi^1(X), y, i)$;
 6: Calculate $\delta^1 = \frac{1}{2} \ln \left(\frac{1-\epsilon^1}{\epsilon^1} \right)$;
 7: Add to new ensemble $\Psi^1(X) = \delta^1 \psi^1$;
 8: **Else**
 9: Choose $\psi^g(X)$;
 10: Consider each classifier $\psi_j^g(X) \in \hat{\mathcal{P}}$ ($j \in \{1, \dots, |\hat{\mathcal{P}}|\}$):
 11: Calculate the current cost function $\Delta_C(\psi_j^g(X))$ in Eqs. (10) and (11);
 12: The weighted sum error for misclassified points
 $\epsilon_j^g = \sum_i \omega_i^g \theta(\psi_j^g(X), y, i)$;
 13: Compute $\delta_j^g = \frac{1}{2} \ln \left(\frac{1-\epsilon_j^g}{\epsilon_j^g} \right)$;
 14: Sort all the classifiers in $\hat{\mathcal{P}}$ according to the values of the current cost function;
 15: $j=0$;
 16: **Repeat**
 17: $j=j+1$;
 18: **Until** $\Delta_L(\Psi^{g-1}(X)) \leq \Delta_L(\Psi^{g-1}(X) + \delta_j^g \psi_j^g)$;
 19: Add to new ensemble: $\Psi^g(X) = \Psi^{g-1}(X) + \delta_j^g \psi_j^g$;
 20: **End**
 21: Update weights:
 22: $\omega_i^{g+1} = \omega_i^g e^{-\gamma_i \delta_j^g \psi_j^g(x_i)}$ for all the samples;
 23: Re-normalize ω_i^{g+1} , such that $\sum_i \omega_i^{g+1} = 1$;
Output: the new ensemble Ψ^G .

Finally, the labels of the samples in the testing set are predicted by the set of classifiers in the new ensemble, and PSEL obtains the final result with the weighted voting scheme.

3.2. Progressive selection process

Algorithm 2 provides an overview of the progressive selection process (PSP). The input of PSP is the original ensemble, while the output is a new ensemble generated in a way similar to AdaBoost. At the beginning, the weights of the samples are initialized as $\frac{1}{l}$ (where l is the number of training samples). The classifier ψ^1 associated with the highest accuracy value will be selected as the first classifier as follows:

$$\psi^1 = \arg \max_{\psi \in \hat{\mathcal{P}}} \{AC_1, AC_2, \dots, AC_B\} \quad (4)$$

where $\hat{\mathcal{P}} = \{\psi_1, \psi_2, \dots, \psi_B\}$ denotes the set of classifiers in the ensemble. The weighted sum error for misclassified points based on the first classifier ψ^1 is calculated as follows:

$$\epsilon^1 = \sum_i \omega_i^1 \theta(\psi^1(X), y, i) \quad (5)$$

where ω_i^1 is the weight value for the sample x_i in the 1th iteration, and $\theta(\psi(X), y, i) = e^{-\gamma_i \psi(x_i)}$ is the loss function. The weight δ^1 of the first classifier ψ^1 is updated as follows:

$$\delta^1 = \frac{1}{2} \ln \left(\frac{1-\epsilon^1}{\epsilon^1} \right) \quad (6)$$

The classifier ψ^1 with its weight δ^1 will be included into the new

ensemble as follows:

$$\Psi^1(X) = \delta^1 \psi^1 \quad (7)$$

The classifier is then removed from $\hat{\mathcal{P}}$. The new weights ω_i^{g+1} for all the samples are updated as follows:

$$\omega_i^{g+1} = \omega_i^g e^{-\gamma_i \delta^1 \psi^1(x_i)} \quad (8)$$

The new weights are normalized so that the following condition is satisfied:

$$\sum_{i=1}^l \omega_i^{g+1} = 1 \quad (9)$$

Then, PSP will select the classifiers one by one, until $G-1$ classifiers are generated (where G is pre-specified by the user). For each classifier ψ_j in $\hat{\mathcal{P}}$, we consider the following cost function $\Delta_C(\psi)$, which we refer to as the current cost function:

$$\Delta_C(\psi_j) = \beta_1 AC_j + \beta_2 \phi(S_j, S_h) \quad (10)$$

where AC_j is the accuracy value of the classifier on the re-weighted sample set, and $\phi(S_j, S_h)$ denotes the similarity between the subspace S_j corresponding to the classifier ψ_j and the subspace S_h corresponding to the classifier ψ^{g-1} , which is the classifier selected in the new ensemble in the previous iteration. The similarity $\phi(S_j, S_h)$ is defined by the Pearson correlation coefficient between two sets of representative centers $U^j = \{\mu_1^j, \mu_2^j, \dots, \mu_{k_j}^j\}$ in the subspace S_j and $U^h = \{\mu_1^h, \mu_2^h, \dots, \mu_{k_h}^h\}$ in the subspace S_h as follows:

$$\phi(S_j, S_h) = \frac{1}{k_j k_h} \sum_{\mu_t^j \in S_j} \sum_{\mu_q^h \in S_h} 1 - \rho_{(\mu_t^j, \mu_q^h)} \quad (11)$$

$$\rho_{(\mu_t^j, \mu_q^h)} = \frac{m \sum_{e=1}^m \mu_{te}^j \mu_{qe}^h - \overline{\mu_t^j} \overline{\mu_q^h}}{\sqrt{m \sum_{e=1}^m (\mu_{te}^j)^2 - (\overline{\mu_t^j})^2} \sqrt{m \sum_{e=1}^m (\mu_{qe}^h)^2 - (\overline{\mu_q^h})^2}} \quad (12)$$

$$\overline{\mu_t^j} = \sum_{e=1}^m \mu_{te}^j, \quad \overline{\mu_q^h} = \sum_{e=1}^m \mu_{qe}^h \quad (13)$$

where k_j and k_h denote the number of representative centers in the subspaces S_j and S_h respectively, $t \in \{1, \dots, k_j\}$, $q \in \{1, \dots, k_h\}$, and μ_{te}^j, μ_{qe}^h denote the t th representative center in the subspace S_j and the q th representative center in the subspace S_h , respectively. The similarity $\phi(S_j, S_h)$ is defined as the average value of Pearson correlation coefficients for the pairs of representatives in the two subspaces S_j and S_h . The objective of the current cost function is to guide the current search process and to discover the most potential candidate classifiers for the selection. If the subspace corresponding to the candidate classifier is different from the subspaces of the classifiers in the new ensemble, the candidate classifier has high probability to be selected according to the current cost function. In other words, the selection process based on the current cost function prefers those classifiers which will increase the diversity of the new ensemble. In general, $\Delta_C(\psi)$ not only takes into account the accuracy of the classifier on the re-weighted sample set, but also considers the diversity of the subspaces.

Assume that the selected attributes in the subspace S_j is $A^j = \{a_1, a_2, \dots, a_{m_j}\}$ (where m_j is the number of selected attributes), the representative centers $U^j = \{\mu_1^j, \mu_2^j, \dots, \mu_{k_j}^j\}$ in the subspace S_j are obtained by minimizing the sum of squared error Ξ as follows.

$$\Xi(\mu_1, \mu_2, \dots, \mu_{k_j}) = \sum_{l=1}^{k_j} \sum_{i=1}^{m_j} (a_i - \mu_l)^2 \quad (14)$$

$$(\mu_1^j, \mu_2^j, \dots, \mu_{k_j}^j) = \arg \min_{(\mu_1, \mu_2, \dots, \mu_{k_j}) \in \mathbb{R}^{m_j}} \Xi(\mu_1, \mu_2, \dots, \mu_{k_j}) \quad (15)$$

K-means can be adopted to calculate the representative centers $U^j = \{\mu_1^j, \mu_2^j, \dots, \mu_{k_j}^j\}$ in the subspace S_j . The centers U^h in the subspace S_h can be calculated in a similar way.

The number of representative centers k_j in the subspace S_j is determined by the Silhouette Index (Y) [45], which is defined as follows:

$$Y(k_j) = \frac{1}{k_j} \sum_{l=1}^{k_j} Y_l \quad (16)$$

$$Y_l = \frac{1}{|C_l|} \sum_{i=1}^{|C_l|} \vartheta_i^l \quad (17)$$

$$\vartheta_i^l = \frac{\kappa_i^l - t_i^l}{\max\{t_i^l, \kappa_i^l\}} \quad (18)$$

$$\kappa_i^l = \min_{\substack{t \in \{1, \dots, k_j\} \\ t \neq h}} \left\{ \frac{1}{|C_t|} \sum_{e=1}^{|C_t|} (a_i^l, a_e^t)^2 \right\} \quad (19)$$

$$t_i^l = \frac{1}{|C_l| - 1} \sum_{\substack{e=1 \\ e \neq i}}^{|C_l|} (a_i^l, a_e^l)^2 \quad (20)$$

where $l, t \in \{1, \dots, k_j\}$, and C_l and C_t denote the l th cluster and the t th cluster respectively. a_i^l and a_e^t are the i th and e th attributes in the l th cluster, respectively, $d(\cdot)$ denotes the Euclidean distance, t_i^l is the average distance from the i th attribute to the other attributes in the l th cluster, and κ_i^l is the average distance from the i th attribute to the attributes in another cluster. The range of the Silhouette Index is from -1 to 1 . The number of clusters which corresponds to the largest $SI(k_j)$ is selected as the optimal k_j value. k_h in the subspace S_h can be selected in a similar way.

Next, the weighted sum error for misclassified points for each classifier ψ_j^g is computed by PSP as follows:

$$\epsilon_j^g = \sum_i \omega_i^g \Theta(\psi_j^g(X), y, i) \quad (21)$$

where $j \in \{1, \dots, |\hat{\mathcal{P}}|\}$, and $g \in \{1, \dots, G\}$ is the current iteration. The weight δ_j^g of the classifier ψ_j^g is updated as follows:

$$\delta_j^g = \frac{1}{2} \ln \left(\frac{1 - \epsilon_j^g}{\epsilon_j^g} \right) \quad (22)$$

We further propose a cost function $\Delta_L(\Psi)$ to identify which classifier should be selected in the new ensemble by taking long-term information into consideration. $\Delta_L(\Psi)$ is defined as follows:

$$\Delta_L(\Psi) = \sum_{i=1}^l |y_i - \Psi(x_i)| \quad (23)$$

$$\Psi(x_i) = \arg \max_c \sum_{h=1}^g \delta^h \cdot 1\{\psi^h(x_i) = c\} \quad (24)$$

where $c \in \{-1, 1\}$ is the label, and ψ^h and δ^h denote the h th decision tree classifier and the corresponding weighted value in the new ensemble, respectively. The function $1\{\cdot\}$ is defined such that $1\{true\} = 1$ and $1\{false\} = 0$.

All the classifiers are sorted in descending order according to their current cost function values. If $\Delta_L(\Psi^{g-1}(X)) > \Delta_L(\Psi^g(X)) + \delta_j^g \psi_j^g$, the algorithm will consider the next classifier. Otherwise, the classifier ψ_j^g will be selected and included in the new ensemble as follows:

$$\Psi^g(X) = \Psi^{g-1}(X) + \delta_j^g \psi_j^g \quad (25)$$

The classifier ψ_j^g is removed from $\hat{\mathcal{P}}$.

The new weights ω_i^g for all the samples in the $(g+1)$ th iteration are then updated as follows:

$$\omega_i^{g+1} = \omega_i^g e^{-y_i \delta_j^g \psi_j^g(x_i)} \quad (26)$$

The new weights are re-normalized, such that the following condition is satisfied:

$$\sum_{i=1}^l \omega_i^{g+1} = 1 \quad (27)$$

Finally, the progressive selection process will be stopped when G classifiers are selected, or the value of the long-term cost function becomes stable.

The reasons for adopting the current and long-term cost functions are as follows: (1) The long-term cost function is suitable for global search, while the current cost function is useful for local search. Based on the two cost functions, both global search and local search can be improved by adopting the proposed progressive selection process. (2) The current cost function considers the accuracies of the classifiers and the similarity of the subspaces simultaneously, which will improve the adaptivity of the process.

4. Time complexity analysis

We also perform a theoretical analysis of PSEL in terms of its computational cost. The time complexity T_{PSEL} of PSEL in the training process is estimated as follows:

$$T_{PSEL} = T_{OE} + T_{PSP} \quad (28)$$

where T_{OE} and T_{PSP} denote the computational costs for the original ensemble generation and the progressive selection process, respectively. T_{OE} is related to Algorithm 1, while T_{PSP} is related to Algorithm 2.

T_{OE} is related to the number of training samples l , the number of attributes m , and the number of classifiers B as follows:

$$T_{OE} = O(B \cdot l \cdot m \log m) \quad (29)$$

T_{PSP} will be affected by the number of training samples l , the number of attributes m , the number of classifiers B , the maximum number of representative centers in the subspace k_{max} , and the number of iterations G as follows:

$$T_{PSP} = O(G \cdot (B \cdot (l \cdot m \log m + k_{max} \cdot m)) + B \log B) \quad (30)$$

Since G and B are constants, the computational cost of PSEL is approximately $O(lm \log m)$.

T_{NE} is the time complexity for prediction using the new ensemble in the testing process, which is related to the number of classifiers G in the new ensemble, the number of attributes m , and the number of samples l in the testing set as follows:

Table 1

Summary of the 18 gene datasets (where l denotes the number of data samples, m denotes the number of attributes, and k denotes the number of classes).

Dataset	l	m	k
alizadeh-2000-v1	42	1095	2
armstrong-2002-v1	72	1081	2
armstrong-2002-v2	72	2194	3
bittner-2000	38	2201	2
chowdary-2006	104	182	2
dyrskjot-2003	40	1203	3
golub-1999-v1	72	1868	2
golub-1999-v2	72	1868	3
gordon-2002	181	1626	2
khan-2001	83	1069	4
lapointe-2004-v1	69	1625	3
liang-2005	37	1411	3
nutt-2003-v1	50	1377	4
nutt-2003-v2	28	1070	2
pomeroy-2002-v2	42	1379	5
risinger-2003	42	1771	4
shipp-2002-v1	77	798	2
west-2001	49	1198	2

Table 2

Summary of the four datasets from UCI machine learning repository.

Dataset	l	m	k
robotExecution4	117	90	3
robotExecution5	164	90	5
syntheticcontrol	600	60	6
thoracic-surgery	470	16	2

Table 3

Summary of the parameter settings.

Parameters	Default values	Range
The sampling rate in the random subspace	0.1	0.1, 0.2, 0.3, 0.4, 0.5
The number of subspaces B	180	100, 120, 140, 160, 180, 200
The weight values β_1 in the current cost function	0.3	0.1, 0.3, 0.5, 0.7, 0.9

$$T_{NE} = O(G \cdot l \cdot m \log m) \quad (31)$$

Both the random subspace approach and the progressive subspace ensemble learning approach make use of the random subspace technique to generate a set of subspaces in the ensemble. Compared with the random subspace approach, the progressive subspace ensemble learning possesses the following properties: (1) Not all the classifiers contribute to the final result. PSEL selects the classifiers progressively based on the current and long-term cost functions; (2) For the random subspace approach, the classifiers in the ensemble are independently generated, while the classifiers in the ensemble of PSEL are in a sequential order based on the progressive selection approach. (3) PSEL adopts a weighted voting scheme instead of the majority voting scheme.

Both the AdaBoost and progressive subspace ensemble learning approach generate the classifier in the ensemble one by one. The new classifier in the ensemble is generated from newly weighted training samples based on the predicted results of the previous classifier. Compared with the AdaBoost approach, the main distinguishing characteristic of PSEL is that the new classifier in the ensemble of PSEL is generated based on features of the new subspace instead of the original feature space.

5. Experiment

The performances of PSEL and other classifier ensemble approaches are evaluated using 18 cancer gene expression datasets as shown in Table 1 and 4 datasets from UCI machine learning repository [61] in Table 2 (where l denotes the number of data samples, m denotes the number of attributes, and k denotes the number of classes). The preprocessing step for the cancer gene expression profiles is the same as that in [46,72–75]. Most of them are challenging datasets that are used in a large number of previous works. For example, the Armstrong-2002-v2 dataset in Table 1, which are categorized into 3 classes, is characterized by its high dimensionality (2194 dimensions) and small sample size (72 cancer samples). In this case, the samples are much fewer than the variables, which makes the classification problem more challenging.

We use the classification accuracy (AC) to evaluate the quality of predicted labels, which is defined as follows:

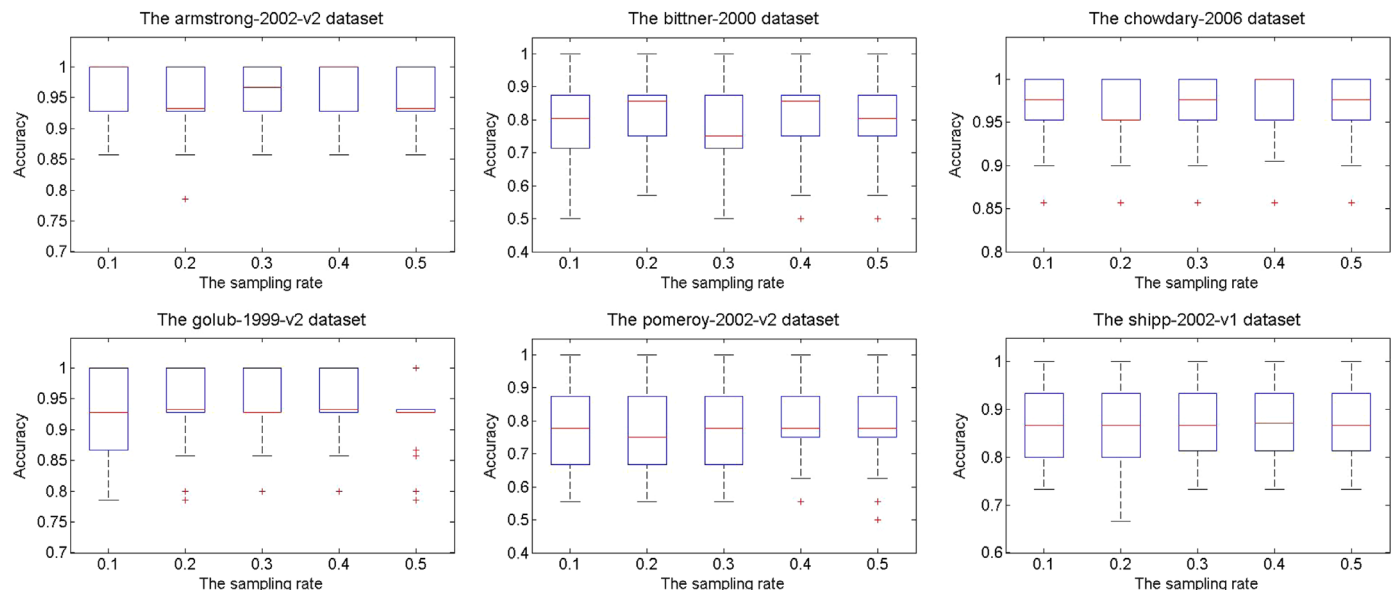


Fig. 3. The effect of the sampling rate in the random subspace technique.

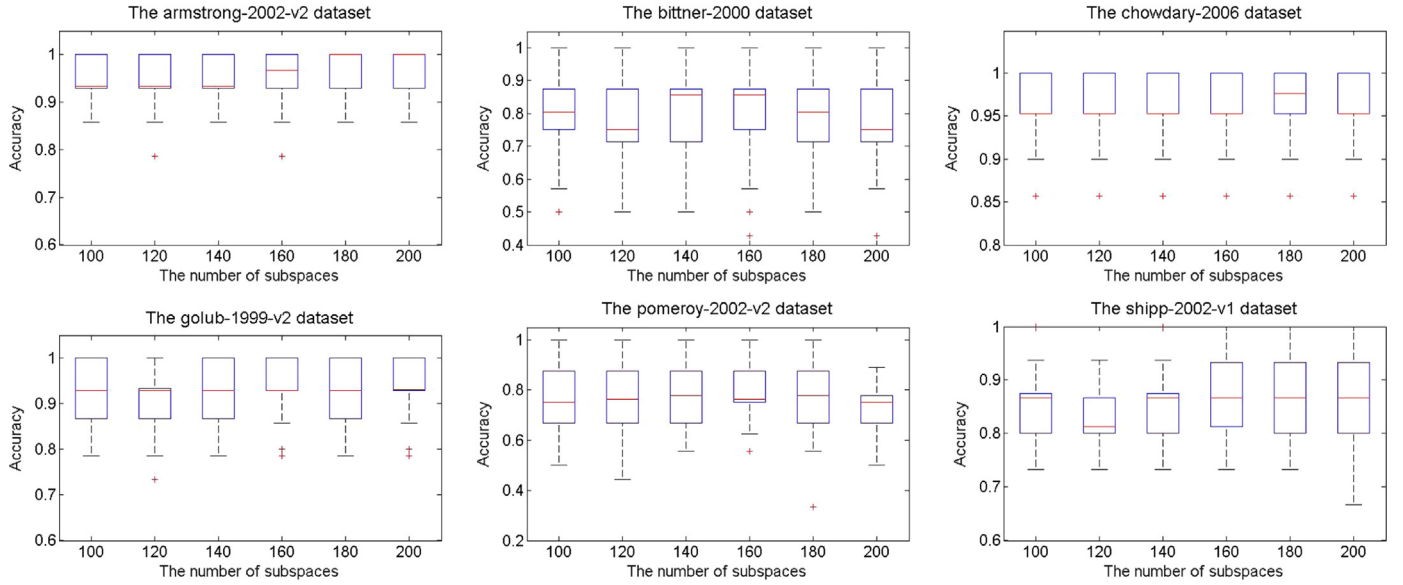


Fig. 4. The effect of the number of subspaces.

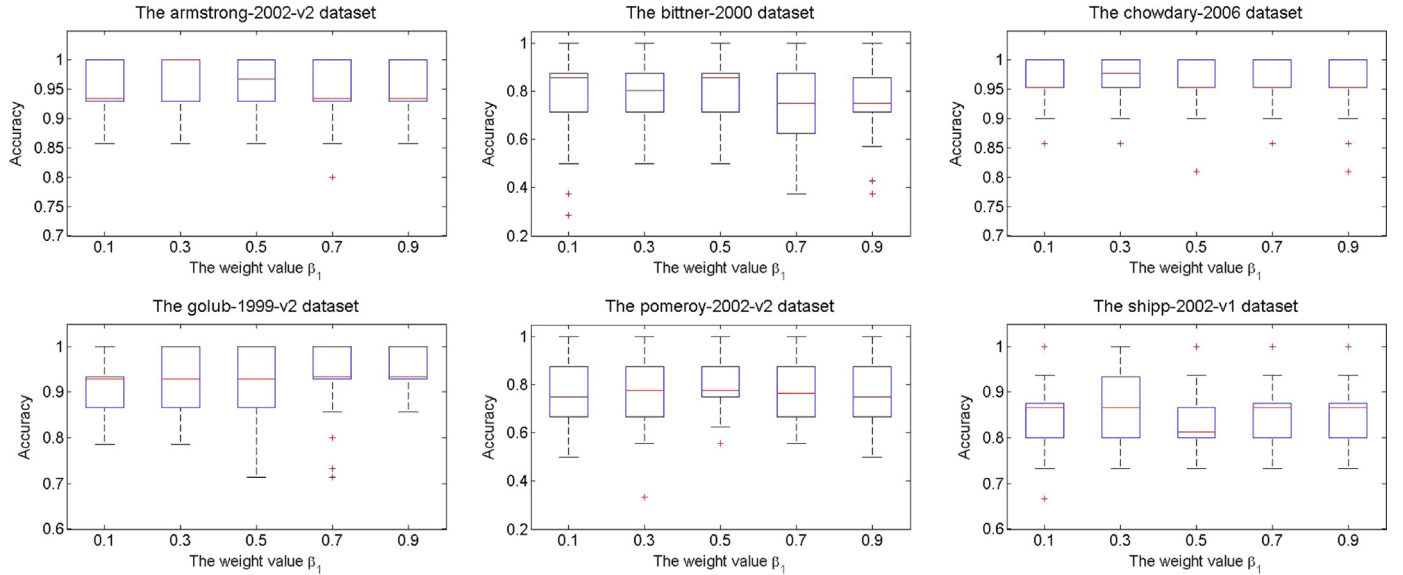


Fig. 5. The effect of the weight value β_1 in the current cost function.

$$AC = \frac{1}{|T_s|} \sum_{x_i \in T_s} 1\{y'_i = y_i^{true}\} \quad (32)$$

where T_s is the set of samples in the testing set, $| \cdot |$ is the cardinality of the set, and y'_i and y_i^{true} denote the predicted label and the true label of the sample x_i , respectively. The performances of the classifier obtained by PSEL and those by other classifier ensemble approaches are measured by the average value and the corresponding standard deviation of the accuracy after 10 runs. 5-fold crossover validation is adopted to reduce the effect of random training/testing set partition. To allow fair comparison, the number of base classifiers in each ensemble approach is set to 50.

In the following experiments, we first investigate the effect of the parameters in PSEL. Then, the effect of the base classifier type is studied. Next, the effect of the progressive selection process is explored. Finally, PSEL is compared with single classification approaches and other classifier ensemble approaches based on all the datasets in Table 1.

5.1. The effect of the parameters

Table 3 provides a summary of parameter settings, which include the default values and the ranges of the parameters. These parameters include the sampling rate in the random subspace technique, the number of random subspaces, and the weight value β_1 in the current cost function. To study the effect of the parameters, they are adjusted one at a time in the following experiments, while the other parameters are set to their default values.

Fig. 3 shows the effect of the sampling rate in the random subspace technique (where the bottom and top of the box correspond to the first and third quartiles of the accuracy values, respectively, and the band inside the box indicates the median). We observe that the accuracy values with respect to different sampling rates are similar for most of the datasets. This indicates that PSEL is robust to changes in the sampling rate. Since a small sampling rate will lead to higher efficiency, the value 0.1 is selected as the default rate.

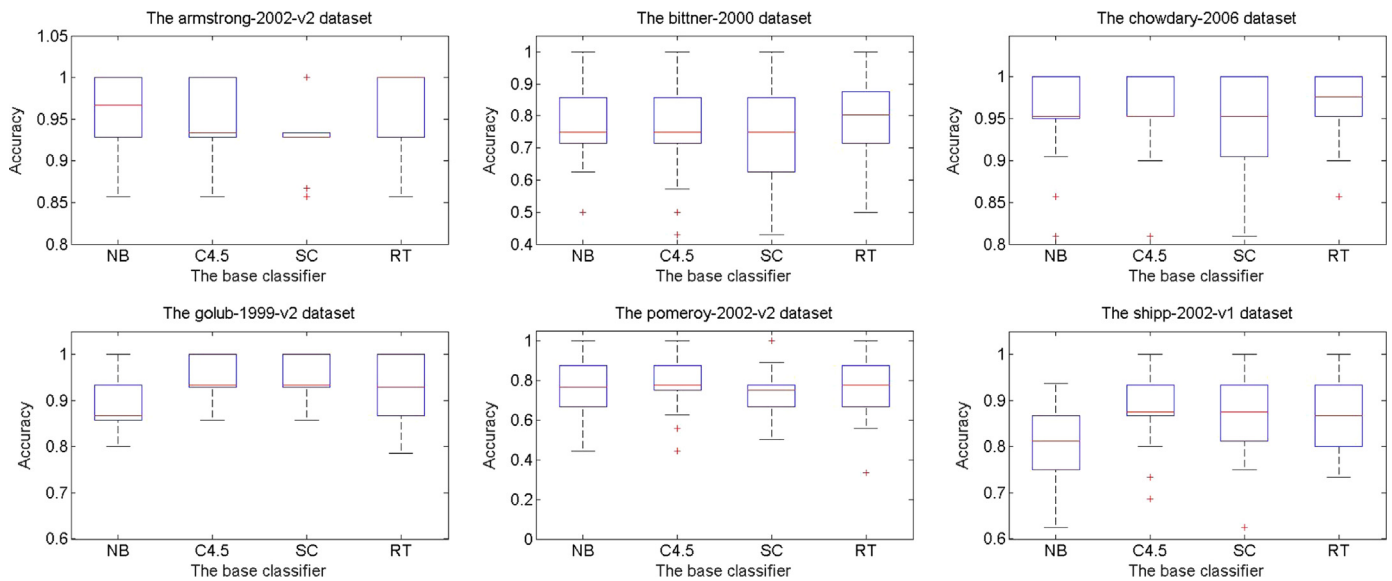


Fig. 6. The effect of the base classifier type (where NB, C4.5, SC and RT denote Naive Bayes, Decision tree, Simple Cart and Random Tree, respectively).

Table 4

The effect of the progressive selection process.

Dataset	PSEL without selection	PSEL with selection
alizadeh-2000-v1	0.863 (0.118)	0.881 (0.106)
armstrong-2002-v1	0.961 (0.058)	0.971 (0.047)
armstrong-2002-v2	0.951 (0.050)	0.961 (0.047)
bittner-2000	0.762 (0.138)	0.804 (0.138)
chowdary-2006	0.965 (0.036)	0.968 (0.037)
dyrskjot-2003	0.823 (0.129)	0.845 (0.115)
golub-1999-v1	0.935 (0.057)	0.962 (0.048)
golub-1999-v2	0.886 (0.071)	0.925 (0.062)
gordon-2002	0.991 (0.013)	0.992 (0.014)
khan-2001	0.969 (0.042)	0.983 (0.034)
lapointe-2004-v1	0.537 (0.082)	0.540 (0.087)
liang-2005	0.895 (0.099)	0.923 (0.092)
nutt-2003-v1	0.754 (0.130)	0.738 (0.129)
nutt-2003-v2	0.777 (0.194)	0.790 (0.204)
pomeroy-2002-v2	0.765 (0.121)	0.760 (0.118)
risinger-2003	0.735 (0.128)	0.732 (0.134)
shipp-2002-v1	0.833 (0.073)	0.856 (0.072)
west-2001	0.837 (0.125)	0.863 (0.111)

Fig. 4 shows the effect of the number of subspaces, which ranges from 100 to 200, with an increment of 20. It is observed that when the number of subspaces is set to 180, PSEL achieves better performance, as can be seen in the Chowdary-2006 dataset and the Pomeroy-2002-v2 dataset. When the number of subspaces increases from 180, the performance of PSEL decreases. The possible reason could be that more and more redundant classifiers

Table 5

The comparison with single classifiers.

Dataset	PSEL	KNN	SVM	C4.5	RT	RVFL
alizadeh-2000-v1	0.881 (0.106)	0.786 (0.111)	0.920 (0.081)	0.687 (0.153)	0.610 (0.156)	0.683 (0.133)
armstrong-2002-v1	0.971 (0.047)	0.919 (0.072)	0.667 (0.026)	0.914 (0.057)	0.856 (0.101)	0.947 (0.049)
armstrong-2002-v2	0.961 (0.047)	0.897 (0.074)	0.389 (0.028)	0.792 (0.072)	0.709 (0.103)	0.904 (0.061)
bittner-2000	0.804 (0.138)	0.657 (0.136)	0.821 (0.113)	0.594 (0.129)	0.566 (0.201)	0.715 (0.143)
chowdary-2006	0.968 (0.037)	0.975 (0.026)	0.596 (0.022)	0.931 (0.057)	0.912 (0.061)	0.804 (0.086)
dyrskjot-2003	0.845 (0.115)	0.800 (0.098)	0.500 (0.000)	0.700 (0.166)	0.635 (0.175)	0.738 (0.114)
golub-1999-v1	0.962 (0.048)	0.936 (0.063)	0.652 (0.012)	0.862 (0.068)	0.762 (0.123)	0.893 (0.082)
golub-1999-v2	0.925 (0.062)	0.912 (0.078)	0.528 (0.027)	0.941 (0.073)	0.686 (0.131)	0.876 (0.078)
gordon-2002	0.992 (0.014)	0.978 (0.023)	0.829 (0.009)	0.933 (0.038)	0.932 (0.044)	0.925 (0.031)
khan-2001	0.983 (0.034)	0.994 (0.025)	0.999 (0.008)	0.872 (0.087)	0.712 (0.121)	0.999 (0.008)
lapointe-2004-v1	0.540 (0.087)	0.417 (0.101)	0.550 (0.058)	0.432 (0.140)	0.426 (0.120)	0.540 (0.083)
liang-2005	0.923 (0.092)	0.971 (0.056)	0.888 (0.102)	0.785 (0.135)	0.833 (0.144)	0.818 (0.125)
nutt-2003-v1	0.738 (0.129)	0.596 (0.125)	0.300 (0.000)	0.552 (0.130)	0.450 (0.127)	0.636 (0.119)
nutt-2003-v2	0.790 (0.204)	0.681 (0.147)	0.460 (0.050)	0.624 (0.239)	0.699 (0.184)	0.833 (0.163)
pomeroy-2002-v2	0.760 (0.118)	0.737 (0.142)	0.239 (0.014)	0.559 (0.133)	0.468 (0.142)	0.732 (0.106)
risinger-2003	0.732 (0.134)	0.654 (0.118)	0.701 (0.115)	0.514 (0.170)	0.451 (0.163)	0.662 (0.141)
shipp-2002-v1	0.856 (0.072)	0.780 (0.100)	0.753 (0.025)	0.782 (0.099)	0.733 (0.103)	0.760 (0.036)
west-2001	0.863 (0.111)	0.710 (0.133)	0.489 (0.022)	0.845 (0.103)	0.688 (0.173)	0.794 (0.114)

would then be included in the ensemble, which will affect the performance.

Fig. 5 shows the effect of the weight value β_1 in the current cost function of the progressive selection process (β_2 in the same cost function is set to $1 - \beta_1$). β_1 controls the relative importance of training accuracy and similarity between subspaces in the progressive selection process. It can be seen that when β_1 is set to 0.3, PSEL obtains the best performance on most of the datasets, such as the Armstrong-2002-v2 dataset, the Chowdary-2006 dataset and the Shipp-2002-v1 dataset.

5.2. The effect of the base classifier type

To investigate the effect of the base classifier type, we compare PSEL based on the random tree (RT) with PSEL based on Naive Bayes (NB), Decision tree (C4.5, [50] [71]) and Simple Cart (SC). Fig. 6 shows the results obtained by PSEL based on different base classifiers. It can be seen that PSEL based on the random tree achieves the best performance on the Armstrong-2002-v2 dataset, the Bittner-2000 dataset and the Chowdary-2006 dataset. For example, it obtains the best result with the highest average accuracy value 0.9682 on the Chowdary-2006 dataset, when compared with the results obtained by PSEL based on other classifiers. Using random tree as the base classifier also provides the advantage of efficiency, since only a small number of features are considered at each node of the tree. PSEL based on Naive Bayes, C4.5 and Simple Cart also attain satisfactory results.

5.3. The effect of the progressive selection process

To explore the effect of the progressive selection process, PSEL with selection is compared with PSEL without selection. Table 4

Table 6
Comparison of the UCI datasets with single classifiers.

Dataset	PSEL	KNN	SVM	C4.5	RT	RVFL
robotExecution4	0.921 (0.046)	0.875 (0.056)	0.616 (0.023)	0.795 (0.073)	0.769 (0.090)	0.708 (0.048)
robotExecution5	0.727 (0.060)	0.674 (0.065)	0.282 (0.016)	0.600 (0.093)	0.572 (0.083)	0.387 (0.023)
syntheticcontrol	0.986 (0.011)	0.985 (0.007)	0.450 (0.041)	0.917 (0.027)	0.877 (0.040)	0.943 (0.019)
thoracic-surgery	0.851 (0.000)	0.773 (0.029)	0.851 (0.000)	0.841 (0.017)	0.769 (0.040)	0.851 (0.000)

Table 7
Comparison with ensemble classifiers.

Dataset	PSEL	Random-Forest	Random-Subspace	AdaBoostM1	MultiboostAB	Bagging	CEREP	CECMP	RTboost
alizadeh-2000-v1	0.8808 (0.1064)	0.8836 (0.1219)	0.8325 (0.1255)	0.8631 (0.1115)	0.8786 (0.0983)	0.8367 (0.1075)	0.6758 (0.1769)	0.6772 (0.1284)	0.6217 (0.1362)
armstrong-2002-v1	0.9705 (0.0474)	0.9666 (0.0455)	0.9679 (0.0511)	0.9694 (0.0596)	0.9667 (0.0597)	0.9599 (0.0622)	0.9003 (0.0999)	0.9073 (0.0755)	0.8293 (0.1004)
armstrong-2002-v2	0.9611 (0.0466)	0.9430 (0.0543)	0.9346 (0.0517)	0.9045 (0.0879)	0.9085 (0.0722)	0.9457 (0.0478)	0.8170 (0.1057)	0.8072 (0.1147)	0.7182 (0.1129)
bittner-2000	0.8039 (0.1384)	0.7514 (0.1447)	0.7654 (0.1391)	0.7793 (0.1253)	0.7696 (0.1171)	0.7993 (0.1398)	0.6189 (0.1370)	0.6264 (0.1928)	0.5575 (0.1824)
chowdary-2006	0.9682 (0.0372)	0.9634 (0.0383)	0.9558 (0.0420)	0.9616 (0.0422)	0.9673 (0.0393)	0.9576 (0.0464)	0.9184 (0.0581)	0.9174 (0.0606)	0.9156 (0.0646)
dyrskjot-2003	0.8450 (0.1145)	0.8275 (0.1098)	0.8025 (0.1239)	0.7425 (0.1681)	0.7325 (0.1370)	0.7325 (0.1263)	0.6650 (0.1524)	0.6225 (0.1777)	0.6475 (0.2031)
golub-1999-v1	0.9623 (0.0475)	0.9165 (0.0706)	0.9425 (0.0634)	0.9317 (0.0595)	0.9233 (0.0649)	0.9175 (0.0660)	0.8877 (0.0954)	0.9079 (0.0795)	0.8267 (0.1212)
golub-1999-v2	0.9251 (0.0624)	0.8874 (0.0630)	0.9392 (0.0601)	0.9361 (0.0520)	0.9407 (0.0477)	0.8986 (0.0824)	0.9120 (0.0706)	0.8907 (0.0855)	0.5276 (0.0268)
gordon-2002	0.9917 (0.0139)	0.9917 (0.0139)	0.9823 (0.0214)	0.9901 (0.0174)	0.9873 (0.0178)	0.9846 (0.0230)	0.9470 (0.0313)	0.9564 (0.0329)	0.9193 (0.0474)
khan-2001	0.9832 (0.0344)	0.9797 (0.0355)	0.9712 (0.0423)	0.5400 (0.0673)	0.5400 (0.0673)	0.9373 (0.0665)	0.7940 (0.1073)	0.8168 (0.1195)	0.7579 (0.1053)
lapointe-2004-v1	0.5398 (0.0871)	0.5546 (0.0798)	0.5629 (0.0396)	0.4932 (0.1125)	0.4915 (0.1151)	0.5425 (0.0599)	0.4530 (0.1164)	0.4964 (0.1172)	0.4166 (0.1421)
liang-2005	0.9232 (0.0918)	0.8921 (0.0961)	0.8118 (0.1376)	0.8214 (0.1498)	0.8750 (0.1218)	0.8000 (0.1281)	0.7707 (0.1409)	0.7818 (0.1312)	0.8254 (0.1671)
nutt-2003-v1	0.7380 (0.1292)	0.7300 (0.1199)	0.6160 (0.1346)	0.5180 (0.1024)	0.5180 (0.1024)	0.6180 (0.1207)	0.4680 (0.1708)	0.4800 (0.1807)	0.4880 (0.1154)
nutt-2003-v2	0.7900 (0.2037)	0.7793 (0.1837)	0.6433 (0.1619)	0.6693 (0.1953)	0.6853 (0.2056)	0.7007 (0.1804)	0.5740 (0.2179)	0.6073 (0.1992)	0.6427 (0.1878)
pomeroy-2002-v2	0.7603 (0.1176)	0.7358 (0.1257)	0.6911 (0.0984)	0.3842 (0.1017)	0.3842 (0.1017)	0.6669 (0.1196)	0.5239 (0.1657)	0.5528 (0.1655)	0.4264 (0.1528)
risinger-2003	0.7319 (0.1339)	0.7158 (0.1484)	0.6147 (0.1219)	0.5456 (0.1237)	0.5456 (0.1237)	0.6689 (0.1368)	0.5514 (0.1523)	0.6014 (0.1618)	0.7008 (0.1152)
shipp-2002-v1	0.8561 (0.0717)	0.8227 (0.0667)	0.7624 (0.0479)	0.8780 (0.0665)	0.8834 (0.0734)	0.8418 (0.0757)	0.7768 (0.0813)	0.7924 (0.0966)	0.7346 (0.1004)
west-2001	0.8631 (0.1113)	0.8629 (0.1041)	0.8638 (0.0920)	0.8338 (0.0955)	0.8316 (0.0841)	0.8593 (0.0912)	0.7949 (0.1297)	0.6596 (0.1287)	0.6596 (0.1315)

shows the results obtained by the two approaches on 18 datasets. It is observed that PSEL with selection achieves better performances on 15 out of 18 datasets. For example, the average accuracy values obtained by PSEL with selection are 0.8039 and 0.9251 on the bittner-2000 dataset and the golub-1999-v2 dataset respectively, which are 0.0418 and 0.0391 higher than those obtained by PSEL without selection. The reason could be that the long-term cost function and the current cost function serve complementary roles in enhancing the capability of our approach to select an optimal subset of classifiers in the ensemble, which will in turn improve the quality of the final result.

5.4. Comparison with single classifiers

We compare the performance of PSEL with state-of-the-art single classifiers [47] with respect to the average accuracy and the corresponding standard deviations on all the datasets in Tables 1 and 2. These single classifiers include the k-nearest neighbor classifier (KNN, [48]), the support vector machine (SVM, [49]), the decision tree classifier (C4.5, [65]), the random tree classifier (RT) and the random vector functional link network based classifier (RVFL, [58]). Tables 5 and 6 show the performances of different classification approaches. It can be observed that PSEL outperforms its competitors on 10 out of 18 gene expression datasets, and on all UCI datasets. For example, PSEL obtains better results on the Armstrong-2002-v1 dataset and the shipp-2002-v1 dataset, attaining average accuracy values of 0.9705 and 0.8561 respectively, which are 0.0513 and 0.0766 larger than the second best results. When compared with the RVFL classifier, PSEL achieves better performance on 16 out of 18 datasets. The possible reasons are as follows: (1) PSEL is able to integrate multiple classifiers into an ensemble framework to generate more accurate, stable and robust results. (2) The random subspace approach in PSEL is useful for exploring different feature subspaces, which

Table 8
Comparison with ensemble classifiers on UCI datasets.

Dataset	PSEL	Random-Forest	Random-Subspace	AdaBoostM1	MultiBoostAB	Bagging	CEREP	CECMP	RTboost
robotExecution4	0.9206 (0.0463)	0.9012 (0.0460)	0.8908 (0.0719)	0.6333 (0.0754)	0.6391 (0.0734)	0.8653 (0.0762)	0.7801 (0.0841)	0.8157 (0.0910)	0.7791 (0.0852)
robotExecution5	0.7272 (0.0597)	0.7162 (0.0642)	0.6949 (0.0515)	0.3719 (0.0312)	0.3719 (0.0312)	0.6870 (0.0555)	0.6621 (0.0684)	0.6605 (0.0813)	0.6237 (0.0900)
syntheticcontrol	0.9855 (0.0114)	0.9818 (0.0102)	0.9690 (0.0147)	0.3313 (0.0052)	0.3313 (0.0052)	0.9530 (0.0194)	0.9323 (0.0282)	0.9458 (0.0229)	0.8858 (0.0394)
thoracic-surgery	0.8511 (0.0000)	0.8451 (0.0101)	0.8511 (0.0000)	0.8462 (0.0081)	0.8481 (0.0061)	0.8500 (0.0039)	0.8270 (0.0274)	0.8251 (0.0262)	0.7764 (0.0316)

Table 9
Average rankings of the ensemble classifiers.

Algorithm	Ranking
PSEL	1.5000
Random-Forest	3.1591
Random-Subspace	3.8409
AdaBoostM1	5.4091
MultiBoostAB	5.0909
Bagging	4.1818
CEREP	7.2273
CECMP	6.7727
RTboost	7.8182

allows effective learning of the data characteristics from multiple perspectives. We also observe that some of the single classifiers are effective when applied to specific datasets but not others. For example, SVM achieves the best performance on the alizadeh-2000-v1 dataset, but its result on the golub-1999-v1 dataset is not satisfactory. In general, PSEL is the best choice when compared with single classifiers.

5.5. Comparison with other ensemble classifiers

We also compare PSEL with five state-of-the-art ensemble classifiers in terms of the average accuracy values and the corresponding standard deviations on all the datasets in Table 1, which includes Random Forest [9], Random Subspace [4], AdaBoostM1 [51], MultiBoostAB [52] and Bagging [3]. Table 7 shows the results obtained by different ensemble classifiers. It is observed that PSEL achieves the best performance on 13 out of 18 datasets, and outperforms other classifier ensemble approaches. The possible reasons are as follows: (1) PSEL adopts the progressive selection process based on the long-term and current cost functions to remove the redundant classifiers and generate an optimal subset of classifiers. (2) PSEL considers both the feature space and the sample space, and assigns different weight values to the various classifiers in the ensemble, which will be useful to improve the performance of the final result. On the other hand, the random forest approach obtains better results on the alizadeh-2000-v1 dataset and the gordon-2002 dataset, while the MultiBoostAB

approach achieves better performance on the golub-1999-v2 dataset and the shipp-2002-v1 dataset. In summary, the ensemble classifiers obtain satisfactory results on most of the datasets, and PSEL is more effective when compared with traditional ensemble classifiers.

We also compare PSEL with the classifier ensemble approach based on Reduce-Error Pruning (CEREP, [60]) and the classifier ensemble approach based on Complementarity-Measure Pruning (CECMP, [60]). It can be seen that PSEL obtains better results on all the real-world datasets when compared with CEREP and CECMP. The possible reason is that PSEL adopts the random subspace technique to explore the underlying subspace structure, which is useful for improving its discriminative capability.

We further compare PSEL with a simple boosting model using random trees as weak classifiers, which is denoted as RTboost. It is observed that PSEL outperforms RTboost on all the high dimensional datasets. For example, the average accuracy value 0.9232 obtained by PSEL in the Liang-2005 dataset is 0.0978 higher than that obtained by RTboost. The possible reason is that PSEL not only adopts the random subspace technique to take into account the effect of the high dimensionality, but also uses a progressive selection process to remove the redundant classifiers and select the best candidate classifiers.

In addition, the performance of PSEL is further evaluated on four UCI datasets in Table 2. Table 8 shows the results obtained by the different classifier ensemble approaches. PSEL is observed to also outperform its competitors on these four datasets. This indicates that PSEL is not only suitable for the datasets with very high dimensionality, but also suitable for a broad class of datasets.

5.6. Significance test

We further apply a number of nonparametric tests [53] to determine the degree to which the performance difference among the various approaches is significant. Tables 9–11 summarize the results of multiple comparisons of the different classifier ensemble approaches using various nonparametric statistical procedures, including the Bonferroni-Dunn test, the Holm test, the Hochberg test and the Hommel test. Table 9 shows the average ranking of the classifier ensemble approaches. It can be observed that the average ranking of PSEL is higher than those of other approaches.

Table 10
Adjusted p -values in nonparametric tests (Bonferroni-Dunn, Holm, Hochberg and Hommel tests; PSEL is the control and bold results denote $p > 0.0500$).

Algorithm	Unadjusted p	p_{Bonf}	p_{Holm}	p_{Hoch}	p_{Hommel}
RTboost	1.9834E-14	1.5867E-13	1.5867E-13	1.5867E-13	1.5867E-13
CEREP	4.0315E-12	3.2252E-11	2.8221E-11	2.8221E-11	2.8221E-11
CECMP	1.7074E-10	1.3659E-9	1.0244E-9	1.0244E-9	1.0244E-9
AdaBoostM1	2.1998E-6	1.7599E-5	1.0999E-5	1.0999E-5	1.0999E-5
MultiBoostAB	1.3688E-5	1.0950E-4	5.4752E-5	5.4752E-5	5.4752E-5
Bagging	0.0012	0.0093	0.0035	0.0035	0.0035
Random-Subspace	0.0046	0.0367	0.0092	0.0092	0.0092
Random-Forest	0.0445	0.3561	0.0445	0.0445	0.0445

Table 11Adjusted p -values in multiple tests (Nemenyi, Holm, Shaffer and Bergmann tests, where the bold values denote that the results are not significant).

Hypothesis	Unadjusted p	p_{Neme}	p_{Holm}	p_{Shaf}	p_{Berg}
PSEL vs .AdaBoostM1	2.1998E−6	7.9193E−5	6.5994E−5	6.1595E−5	3.9597E−5
PSEL vs .Bagging	0.0012	0.0419	0.0256	0.0256	0.0153
PSEL vs .CEREP	4.0315E−12	1.4514E−10	1.4110E−10	1.1288E−10	1.1288E−10
PSEL vs .CECMP	1.7074E−10	6.1466E−9	5.8051E−9	4.7807E−9	3.7562E−9
PSEL vs .MultiboostAB	1.3688E−5	4.9277E−4	3.6958E−4	3.3794E−4	2.1901E−4
PSEL vs .Random-Forest	0.0445	1.0000	0.5834	0.5786	0.4451
PSEL vs .Random-Subspace	0.0046	0.1650	0.0871	0.0825	0.0458
PSEL vs .RTboost	1.9834E−14	7.1403E−13	7.1403E−13	7.1403E−13	7.1403E−13
Random-Forest vs .AdaBoostM1	0.0064	0.2316	0.1158	0.1158	0.0836
Random-Forest vs .Bagging	0.2155	1.0000	1.0000	1.0000	1.0000
Random-Forest vs .CEREP	8.3582E−7	3.0089E−5	2.6746E−5	2.3403E−5	1.7552E−5
Random-Forest vs .CECMP	1.2069E−5	4.3450E−4	3.3794E−4	3.3794E−4	1.9311E−4
Random-Forest vs .MultiboostAB	0.0193	0.6951	0.3089	0.3089	0.1738
Random-Forest vs .Random-Subspace	0.4090	1.0000	1.0000	1.0000	1.0000
Random-Forest vs .RTboost	1.6766E−8	6.0357E−7	5.5327E−7	4.6944E−7	4.6944E−7
Random-Subspace vs .AdaBoostM1	0.0575	1.0000	0.6905	0.6905	0.5179
Random-Subspace vs .Bagging	0.6797	1.0000	1.0000	1.0000	1.0000
Random-Subspace vs .CEREP	4.1121E−5	0.0015	0.0011	9.0466E−4	6.5793E−4
Random-Subspace vs .CECMP	3.8434E−4	0.0138	0.0092	0.0085	0.0046
Random-Subspace vs .MultiboostAB	0.1301	1.0000	1.0000	1.0000	0.7804
Random-Subspace vs .RTboost	1.4594E−6	5.2538E−5	4.5241E−5	4.0863E−5	3.2107E−5
AdaBoostM1 vs .Bagging	0.1372	1.0000	1.0000	1.0000	0.9604
AdaBoostM1 vs .CEREP	0.0277	0.9961	0.4151	0.4151	0.1937
AdaBoostM1 vs .CECMP	0.0986	1.0000	1.0000	1.0000	0.5179
AdaBoostM1 vs .MultiboostAB	0.7000	1.0000	1.0000	1.0000	1.0000
AdaBoostM1 vs .RTboost	0.0035	0.1270	0.0706	0.0635	0.0423
MultiboostAB vs .Bagging	0.2709	1.0000	1.0000	1.0000	1.0000
MultiboostAB vs .CEREP	0.0097	0.3483	0.1645	0.1548	0.0871
MultiboostAB vs .CECMP	0.0417	1.0000	0.5834	0.5417	0.2917
MultiboostAB vs .RTboost	9.5693E−4	0.0344	0.0220	0.0211	0.0153
Bagging vs .CEREP	2.2582E−4	0.0081	0.0056	0.0050	0.0029
Bagging vs .CECMP	0.0017	0.0613	0.0358	0.0358	0.0170
Bagging vs .RTboost	1.0634E−5	3.8284E−4	3.0840E−4	2.9776E−4	1.9142E−4
CEREP vs .CECMP	0.5820	1.0000	1.0000	1.0000	1.0000
CEREP vs .RTboost	0.4742	1.0000	1.0000	1.0000	1.0000
CECMP vs .RTboost	0.2055	1.0000	1.0000	1.0000	1.0000

Table 12

Summary of the twenty datasets from UCI machine learning repository.

Dataset	l	m	k
Australian	690	14	2
Balance	625	4	3
Cnae-9	1080	856	9
CTG	2126	21	3
Dermatology	358	34	6
Ecoli	336	7	8
Glass	214	9	6
Haberman	306	3	2
Heart	270	13	2
Ionosphere	351	34	2
Iris	150	4	3
Mfeat	2000	649	10
Movement-libras	360	90	15
Parkinsons	195	22	2
Segment	2310	19	7
Sonar	208	60	2
Spectf	267	44	2
Syntheticcontrol	600	60	6
Wdbc	569	30	2
Wine	178	13	3

Tables 10 and 11 list the adjusted p -values associated with the different non-parametric tests, which further confirm the significant performance improvement of PSEL over other classifier ensemble approaches.

5.7. Comparison with other ensemble classifiers on UCI datasets

We also compare PSEL with other ensemble classifiers on twenty real-world datasets from UCI machine learning repository, which is shown in Table 12. The comparison results obtained by different ensemble classifier approaches are illustrated in Table 13. It is observed that PSEL outperforms its competitors on 12 out of 20 datasets, which attests that PSEL has a good discriminative ability for different kinds of datasets.

6. Conclusion and future work

In this paper, we investigate the problem of how to combine multiple classifiers, and how to select a suitable classifier subset in the ensemble. Our major contribution is a progressive subspace ensemble learning approach (PSEL) which takes into account both the feature space and the sample space at the same time. PSEL integrates the random subspace technique, the proposed progressive ensemble member selection process, and the weighted voting scheme to obtain a more accurate, stable and robust final result. The long-term and current cost functions in the progressive selection process are designed to eliminate the redundant classifiers and generate a suitable subset of classifiers in the ensemble. In addition, a number of non-parametric tests are adopted to compare PSEL with other classifier ensemble approaches over multiple datasets. We perform a thorough exploration of PSEL on

Table 13

Comparison with different ensemble classifiers on UCI datasets.

Dataset	PSEL	Random-Forest	Random-Subspace	AdaboostM1	MultiboostAB	Bagging	CEREP	CECMP	RTboost
Australian	0.8619 (0.0299)	0.8591 (0.0310)	0.8651 (0.0255)	0.8612 (0.0297)	0.8619 (0.0322)	0.8646 (0.0289)	0.8549 (0.0325)	0.8578 (0.0295)	0.7881 (0.0328)
Balance	0.7494 (0.0386)	0.8186 (0.0243)	0.8403 (0.0337)	0.7290 (0.0351)	0.7290 (0.0351)	0.8571 (0.0226)	0.8309 (0.0269)	0.8397 (0.0234)	0.7827 (0.0278)
Cnae-9	0.9375 (0.0169)	0.9252 (0.0164)	0.8922 (0.0193)	0.2034 (0.0088)	0.2034 (0.0088)	0.8527 (0.0185)	0.9043 (0.0230)	0.9071 (0.0195)	0.7727 (0.0395)
CTG	0.9467 (0.0088)	0.9438 (0.0096)	0.9341 (0.0102)	0.7788 (0.0028)	0.7788 (0.0028)	0.9372 (0.0102)	0.9344 (0.0096)	0.9343 (0.0101)	0.9279 (0.0125)
Dermatology	0.9751 (0.0178)	0.9737 (0.0171)	0.9698 (0.0168)	0.5025 (0.0126)	0.5025 (0.0126)	0.9609 (0.0299)	0.9506 (0.0218)	0.9481 (0.0241)	0.9255 (0.0347)
Ecoli	0.8342 (0.0352)	0.8530 (0.0325)	0.8438 (0.0317)	0.6459 (0.0118)	0.6459 (0.0118)	0.8306 (0.0363)	0.8242 (0.0456)	0.8282 (0.0351)	0.7711 (0.0386)
Glass	0.7762 (0.0563)	0.7744 (0.0638)	0.7262 (0.0518)	0.4467 (0.0201)	0.4467 (0.0201)	0.7268 (0.0572)	0.6950 (0.0618)	0.7188 (0.0657)	0.6959 (0.0735)
Haberman	0.7118 (0.0371)	0.6830 (0.0460)	0.7334 (0.0073)	0.7271 (0.0456)	0.7353 (0.0449)	0.7268 (0.0376)	0.7163 (0.0460)	0.7107 (0.0468)	0.6687 (0.0600)
Heart	0.8252 (0.0517)	0.8281 (0.0447)	0.8215 (0.0538)	0.8163 (0.0452)	0.8319 (0.0473)	0.8178 (0.0536)	0.7907 (0.0517)	0.8015 (0.0547)	0.7311 (0.0570)
Ionosphere	0.9362 (0.0187)	0.9353 (0.0191)	0.9336 (0.0222)	0.9237 (0.0231)	0.9120 (0.0265)	0.9220 (0.0261)	0.8994 (0.0322)	0.9145 (0.0291)	0.8838 (0.0368)
Iris	0.9487 (0.0358)	0.9513 (0.0345)	0.9467 (0.0404)	0.9533 (0.0356)	0.9520 (0.0376)	0.9460 (0.0386)	0.9367 (0.0411)	0.9373 (0.0419)	0.9420 (0.0397)
Mfeat	0.9835 (0.0063)	0.9815 (0.0068)	0.9767 (0.0081)	0.1991 (0.0016)	0.1991 (0.0016)	0.9698 (0.0099)	0.9642 (0.0124)	0.9677 (0.0117)	0.8432 (0.0247)
Movement-libras	0.8108 (0.0439)	0.8033 (0.0425)	0.7306 (0.0515)	0.1219 (0.0102)	0.1219 (0.0102)	0.7344 (0.0597)	0.7136 (0.0449)	0.7317 (0.0511)	0.6306 (0.0515)
Parkinsons	0.9072 (0.0430)	0.9036 (0.0429)	0.8703 (0.0527)	0.8913 (0.0476)	0.8605 (0.0539)	0.8744 (0.0596)	0.8667 (0.0595)	0.8662 (0.0639)	0.8503 (0.0654)
Segment	0.9800 (0.0059)	0.9783 (0.0064)	0.9648 (0.0087)	0.2855 (0.0007)	0.2855 (0.0007)	0.9640 (0.0081)	0.9670 (0.0090)	0.9686 (0.0079)	0.9503 (0.0124)
Sonar	0.8390 (0.0556)	0.8221 (0.0584)	0.7677 (0.0709)	0.8197 (0.0600)	0.8024 (0.0516)	0.7807 (0.0674)	0.7317 (0.0751)	0.7717 (0.0606)	0.7063 (0.0710)
Spectf	0.8106 (0.0338)	0.8112 (0.0344)	0.7940 (0.0019)	0.8015 (0.0380)	0.8083 (0.0437)	0.8072 (0.0262)	0.7876 (0.0430)	0.7985 (0.0427)	0.7374 (0.0542)
Syntheticcontrol	0.9863 (0.0102)	0.9818 (0.0102)	0.9690 (0.0147)	0.3313 (0.0052)	0.3313 (0.0052)	0.9530 (0.0194)	0.9357 (0.0283)	0.9523 (0.0247)	0.8858 (0.0394)
Wdbc	0.9650 (0.0162)	0.9635 (0.0168)	0.9524 (0.0203)	0.9620 (0.0164)	0.9580 (0.0168)	0.9483 (0.0202)	0.9344 (0.0254)	0.9425 (0.0236)	0.9294 (0.0286)
Wine	0.9696 (0.0315)	0.9769 (0.0287)	0.9633 (0.0333)	0.9135 (0.0454)	0.9191 (0.0503)	0.9505 (0.0445)	0.9146 (0.0435)	0.9140 (0.0525)	0.9286 (0.0465)

high dimensional real datasets in our experiments, and obtain several conclusions: (1) The progressive selection process plays an important role in PSEL. (2) PSEL outperforms state-of-the-art classifier ensemble approaches on most of the datasets. In the future, we shall further optimize the performance of PSEL by considering alternative cost functions, and apply the framework to other research areas. We shall also consider how to generalize the current framework by using a probabilistic formulation to measure the similarity of two subspaces, and how to apply the proposed approach to a heterogeneous ensemble of classifiers.

Acknowledgment

The authors are grateful for the constructive advice received from the anonymous reviewers of this paper. The work described in this paper was partially funded by the grant from the National High-Technology Research and Development Program (863 Program) of China No. 2013AA01A212, the grant from the NSFC for Distinguished Young Scholars 61125205, and the grants from the NSFC Nos. 61332002, 61300044, 61472145, 61572199, 61502174, and 61502173, the grant from the Guangdong Natural Science Funds for Distinguished Young Scholars (No. S2013050014677), the Fundamental Research Funds for the Central Universities (Nos. D2153950, 2014G0007 and 2015PT016), the grant from the Key Lab of Cloud Computing and Big Data in Guangzhou (No. SITGZ [2013]268–6), the grant from Science and Technology Planning Project of Guangdong Province, China (No. 2015A050502011, No. 2013B090500015, No. 2016A050503015), the grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CityU 11300715), the grant from City University

of Hong Kong (No. 7004220), the grant from Hong Kong General Research Grant (No. B-Q44D), and the grants from the Hong Kong Polytechnic University (Nos. G-YM05 and G-YN39).

References

- [1] Z. Yu, L. Li, J. Liu, G. Han, Hybrid adaptive classifier ensemble, *IEEE Trans. Cybern.* 42 (2) (2015) 177–190.
- [2] L.I. Kuncheva, J.J. Rodriguez, Classifier ensembles with a random linear oracle, *IEEE Trans. Knowl. Data Eng.* 19 (4) (2007) 500–508.
- [3] L. Breiman, Bagging predictors, *Mach. Learn.* 24 (2) (1996) 123–140.
- [4] T.K. Ho, The random subspace method for constructing decision forests, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (8) (1998) 832–844.
- [5] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, *Artif. Intell.* 137 (1–2) (2002) 239–263.
- [6] Z.-H. Zhou, Y. Jiang, NeC4.5: neural ensemble based C4.5, *IEEE Trans. Knowl. Data Eng.* 16 (6) (2004) 770–773.
- [7] 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.
- [8] L.I. Kuncheva, Fuzzy vs non-fuzzy in combining classifiers designed by boosting, *IEEE Trans. Fuzzy Syst.* 11 (6) (2003) 729–741.
- [9] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (10) (2006) 1619–1630.
- [10] L. Breiman, Random forests, *Mach. Learn.* 45 (1) (2001) 5–32.
- [11] R. Polikar, J. DePasquale, H.S. Mohammed, G. Brown, L.I. Kuncheva, Learn++ MF: a random subspace approach for the missing feature problem, *Pattern Recognit.* 43 (11) (2010) 3817–3832.
- [12] M. Galar, A. Fern, E. Barrenechea, F. Herrera, EUSBoost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, *Pattern Recognit.* 46 (12) (2013) 3460–3471.
- [13] R. Liu, M. Jiang, Chinese text classification based on the BVB Model, in: Fourth International Conference on Semantics, Knowledge and Grid, 2008, 2008, pp. 376–379.
- [14] M.F. Saedian, H. Beigy, Spam detection using dynamic weighted voting based on clustering, in: Second International Symposium on Intelligent Information Technology Application, 2008, vol. 2, 2008, pp. 122–126.

- [15] H. Liu, L. Liu, H. Zhang, Ensemble gene selection for cancer classification, *Pattern Recognit.* 43 (8) (2010) 2763–2772.
- [16] C.O. Plumptre, L.I. Kuncheva, N.N. Oosterhof, S.J. Johnston, Naive random subspace ensemble with linear classifiers for real-time classification of fMRI data, *Pattern Recognit.* 45 (6) (2012) 2101–2108.
- [17] J.-F. Connolly, E. Granger, R. Sabourin, An adaptive ensemble of fuzzy ARTMAP neural networks for video-based face classification, in: 2010 IEEE Congress on Evolutionary Computation (CEC), 2010, pp.1–8.
- [18] X. Zhang, Y. Jia, A linear discriminant analysis framework based on random subspace for face recognition, *Pattern Recognit.* 40 (9) (2007) 2585–2591.
- [19] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, J. You, Semi-supervised classification based on random subspace dimensionality reduction, *Pattern Recognit.* 45 (3) (2012) 1119–1135.
- [20] Y. Ye, Q. Wu, J.Z. Huang, M.K. Ng, X. Li, Stratified sampling for feature subspace selection in random forests for high dimensional data, *Pattern Recognit.* 46 (3) (2013) 769–787.
- [21] Y. Guo, L. Jiao, S. Wang, S. Wang, F. Liu, K. Rong, T. Xiong, A novel dynamic rough subspace based selective ensemble, *Pattern Recognit.* 48 (5) (2015) 1638–1652.
- [22] L. Zhang, P.N. Suganthan, Random forests with ensemble of feature spaces, *Pattern Recognit.* 47 (10) (2014) 3429–3437.
- [23] J. Zhang, D. Zhang, A novel ensemble construction method for multi-view data using random cross-view correlation between within-class examples, *Pattern Recognit.* 44 (6) (2011) 1162–1171.
- [24] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, *Pattern Recognit.* 43 (1) (2010) 128–142.
- [25] L.I. Kuncheva, A bound on kappa-error diagrams for analysis of classifier ensembles, *IEEE Trans. Knowl. Data Eng.* 25 (3) (2013) 494–501.
- [26] X.-Z. Wang, H.-J. Xing, Y. Li, Q. Hua, C.-R. Dong, W. Pedrycz, A study on relationship between generalization abilities and fuzziness of base classifiers in ensemble learning, *IEEE Trans. Fuzzy Syst. PP* (99) (2015) 1, <http://dx.doi.org/10.1109/TFUZZ.2014.2371479>.
- [27] S. Rasheed, D.W. Stashuk, M.S. Kamel, Integrating heterogeneous classifier ensembles for emg signal decomposition based on classifier agreement, *IEEE Trans. Inf. Technol. Biomed.* 14 (3) (2010) 866–882.
- [28] H. Tian, Z. Duan, A. Abraham, H. Liu, A novel multiplex cascade classifier for pedestrian detection, *Pattern Recognit. Lett.* 34 (14) (2013) 1687–1693.
- [29] L. Guo, P.-S. Ge, M.-H. Zhang, L.-H. Li, Y.-B. Zhao, Pedestrian detection for intelligent transportation systems combining adaboost algorithm and support vector machine, *Expert Syst. Appl.* 39 (4) (2012) 4274–4286.
- [30] S. Ali, A. Majid, Can-Evo-Ens: classifier stacking based evolutionary ensemble system for prediction of human breast cancer using amino acid sequences, *J. Biomed. Inf.* 54 (2015) 256–269.
- [31] Q. Gu, Y.-S. Ding, T.-L. Zhang, An ensemble classifier based prediction of G-protein-coupled receptor classes in low homology, *Neurocomputing* 154 (2015) 110–118.
- [32] G. Giacinto, R. Perdisci, M.D. Rio, F. Roli, Intrusion detection in computer networks by a modular ensemble of one-class classifiers, *Inf. Fus.* 9 (1) (2008) 69–82.
- [33] S. Günter, H. Bunke, Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition, *Pattern Recognit. Lett.* 25 (11) (2004) 1323–1336.
- [34] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, *IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev.* 42 (4) (2012) 463–484.
- [35] Q. Hu, D. Yu, Z. Xie, X. Li, EROS: ensemble rough subspaces, *Pattern Recognit.* 40 (12) (2007) 3728–3739.
- [36] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, T.I. Ren, META-DES: a dynamic ensemble selection framework using meta-learning, *Pattern Recognit.* 48 (5) (2015) 1925–1935.
- [37] J. Xiao, C. He, X. Jiang, D. Liu, A dynamic classifier ensemble selection approach for noise data, *Inf. Sci.* 180 (18) (2010) 3402–3421.
- [38] E.M.D. Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, *Pattern Recognit.* 41 (10) (2008) 2993–3009.
- [39] L. Yang, Classifiers selection for ensemble learning based on accuracy and diversity, *Proc. Eng.* 15 (2011) 4266–4270.
- [40] R. Bryll, Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, *Pattern Recognit.* 20 (6) (2003) 1291–1302.
- [41] J. Marin, D. Vazquez, A.M. Lopez, J. Amores, L.I. Kuncheva, Occlusion handling via random subspace classifiers for human detection, *IEEE Trans. Cybern.* 44 (3) (2014) 342–354.
- [42] D. Tao, X. Tang, X. Li, X. Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (7) (2006) 1088–1099.
- [43] L.I. Kuncheva, J.J. Rodriguez, C.O. Plumptre, D.E.J. Linden, S.J. Johnston, Random subspace ensembles for fMRI classification, *IEEE Trans. Med. Imaging* 29 (2) (2010) 531–542.
- [44] S. Sun, C. Zhang, The selective random subspace predictor for traffic flow forecasting, *IEEE Trans. Intell. Transp. Syst.* 8 (2) (2007) 367–373.
- [45] L. Konopnicki, P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, New York, 1990.
- [46] M.C. de Souto, I.G. Costa, D.S. de Araujo, T.B. Ludermir, A. Schliep, Clustering cancer gene expression data: a comparative study, *BMC Bioinform.* 9 (2008) 497, <http://dx.doi.org/10.1186/1471-2105-9-497>.
- [47] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, *SIGKDD Explor.* 11 (1) (2009).
- [48] D. Aha, D. Kibler, Instance-based learning algorithms, *Mach. Learn.* 6 (1) (1991) 37–66.
- [49] C.-C. Chang, C.-J. Lin, LIBSVM—a library for support vector machines, *ACM Trans. Intell. Syst. Technol. Arch.* 2 (3) (2001).
- [50] J. Ross, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [51] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Thirteenth International Conference on Machine Learning, 1996.
- [52] G.I. Webb, MultiBoosting: a technique for combining boosting and wagging, *Mach. Learn.* 40 (2) (2000) 159–196.
- [53] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, *J. Mach. Learn. Res.* 9 (2008) 2677–2694.
- [54] M.R. Daliri, Combining extreme learning machines using support vector machines for breast tissue classification, *Comput. Methods Biomech. Biomed. Eng.* 18 (2) (2015) 185–191.
- [55] M.R. Daliri, Predicting the cognitive states of the subjects in functional magnetic resonance imaging signals using the combination of feature selection strategies, *Brain Topogr.* 25 (2) (2012) 129–135.
- [56] M.R. Daliri, Feature selection using binary particle swarm optimization and support vector machines for medical diagnosis, *Biomed. Eng.* 57 (5) (2012) 395–402.
- [57] A. Ahmadvand, M. Sharififar, M.R. Daliri, Supervised segmentation of MRI brain images using combination of multiple classifiers, *Australas. Phys. Eng. Sci. Med.* 38 (2) (2015) 241–253.
- [58] L. Zhang, P.N. Suganthan, A comprehensive evaluation of random vector functional link networks, *Inf. Sci. this issue*, (2016), <http://dx.doi.org/10.1016/j.ins.2015.09.025>.
- [59] L. Zhang, P.N. Suganthan, Oblique decision tree ensemble via multisurface proximal support vector machine, *IEEE Trans. Cybern.* 45 (10) (2015) 2165–2176.
- [60] G. Martinez-Muoz, D. Hernandez-Lobato, A. Suarez, An analysis of ensemble pruning techniques based on ordered aggregation, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2) (2009) 245–259.
- [61] A. Asuncion, and D.J. Newman, “UCI Machine Learning Repository (<http://www.ics.uci.edu/ml/MLRepository.html>)”, University of California, School of Information and Computer Science, Irvine, CA, 2007.
- [62] Z. Yu, H. Chen, J. You, H. Leung, J. Liu, G. Han, Hybrid K nearest neighbor classifier, *IEEE Trans. Cybern.* 46 (6) (2016) 1263–1275.
- [63] Z. Yu, P. Luo, J. You, H.-S. Wong, H. Leung, S. Wu, J. Zhang, G. Han, Incremental semi-supervised clustering ensemble for high dimensional data clustering, *IEEE Trans. Knowl. Data Eng.* 28 (3) (2016) 701–714.
- [64] Z. Yu, L. Li, J. Liu, J. Zhang, G. Han, Adaptive noise immune cluster ensemble using affinity propagation, *IEEE Trans. Knowl. Data Eng.* 27 (12) (2015) 3176–3189.
- [65] Z. Yu, H.-S. Wong, H. Wang, Graph-based consensus clustering for class discovery from gene expression data, *Bioinformatics* 23 (2007) 2888–2896.
- [66] Z. Yu, J. You, H.-S. Wong, G. Han, From cluster ensemble to structure ensemble, *Inf. Sci.* 168 (2012) 81–99.
- [67] Z. Yu, L. Li, J. You, G. Han, SC3: triple spectral clustering based consensus clustering framework for class discovery from cancer gene expression profiles, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9 (6) (2012) 1751–1765.
- [68] L. Nanni, S. Brahnam, A. Lumini, Double committee adaboost, *J. King Saud Univ.-Sci.* 25 (1) (2013) 29–37.
- [69] L. Nanni, S. Brahnam, S. Ghidoni, A. Lumini, Toward a general-purpose heterogeneous ensemble for pattern classification?, *Comput. Intell. Neurosci.* Article ID 90912 (2015) 1–10.
- [70] C.-X. Zhang, J.-S. Zhang, RotBoost: a technique for combining rotation forest and adaboost, *Pattern Recognit. Lett.* 29 (10) (2008) 1524–1536.
- [71] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [72] Z. Yu, H. Chen, J. You, G. Han, L. Li, hybrid fuzzy cluster ensemble framework for tumor clustering from bio-molecular data, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 10 (3) (2013) 657–670.
- [73] Z. Yu, H.-S. Wong, J. You, Q. Yang, H. Liao, Knowledge based cluster ensemble for cancer discovery from biomolecular data, *IEEE Trans. NanoBioSci.* 10 (2) (2011) 76–85.
- [74] Z. Yu, H.-S. Wong, Class discovery from gene expression data based on perturbation and cluster ensemble, *IEEE Trans. NanoBioSci.* 8 (2) (2009) 147–160.
- [75] Z. Yu, H. Chen, J. You, H.-S. Wong, J. Liu, L. Li, G. Han, Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 11 (4) (2014) 727–740.

Zhiwen Yu (S'06-M'08-SM'14) is a Professor in School of Computer Science and Engineering, South China University of Technology, and an adjunct professor in Sun Yat-Sen University. He is a senior member of IEEE, ACM, CCF (China Computer Federation) and CAAI (Chinese Association for Artificial Intelligence). Until now, Dr. Yu has published more than 90 referred journal papers and international conference papers, including TKDE, TEVC, TCYB, TMM, TSMC-B, TCVST, TCBB, TNB, PR, IS, Bioinformatics, SIGKDD, and so on. Refer to the homepage for more details: <http://www.hgml.cn/yuzhiwen.htm>.

Daxing Wang is a Ph.D. candidate in the School of Computer Science and Engineering in South China University of Technology. He received the B.Sc. degree from South China University of Technology in China. His research interests include bioinformatics, machine learning and data mining.

Jane You is currently a Professor in the Department of Computing at the Hong Kong Polytechnic University and the Chair of Department Research Committee. Prof. You obtained her B.Eng. in Electronic Engineering from Xi'an Jiaotong University in 1986 and Ph.D. in Computer Science from La Trobe University, Australia in 1992. She was a Lecturer at the University of South Australia and Senior Lecturer (tenured) at Griffith University from 1993 till 2002. Prof. You was awarded French Foreign Ministry International Postdoctoral Fellowship in 1993 and worked on the project on real-time object recognition and tracking at Université Paris XI. She also obtained the Academic Certificate issued by French Education Ministry in 1994. Prof. Jane You has worked extensively in the fields of image processing, medical imaging, computer-aided diagnosis, pattern recognition. So far, she has more than 190 research papers published with more than 1000 non-self citations. She has been a principal investigator for one ITF project, three GRF projects and many other joint grants since she joined PolyU in late 1998. Prof. You is also a team member for two successful patents (one HK patent, one US patent) and three awards including Hong Kong Government Industrial Awards. Her current work on retinal imaging has won a Special Prize and Gold Medal with Jury's Commendation at the 39th International Exhibition of Inventions of Geneva (April 2011) and the second place in an international competition (SPIE Medical Imaging'2009 Retinopathy Online Challenge (ROC'2009)). Her research output on retinal imaging has been successfully led to technology transfer with clinical applications. Prof. You is also an Associate Editor of Pattern Recognition and other journals.

Hau-San Wong is currently an Associate Professor in the Department of Computer Science, City University of Hong Kong. He received the B.Sc. and M.Phil. degrees in Electronic Engineering from the Chinese University of Hong Kong, and the Ph.D. degree in Electrical and Information Engineering from the University of Sydney. He has also held research positions in the University of Sydney and Hong Kong Baptist University. His research interests include multimedia information processing, multimodal human-computer interaction and machine learning.

Si Wu received the Ph.D. degree in computer science from City University of Hong Kong, Kowloon, Hong Kong, in 2013. He is an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include computer vision and pattern recognition.

Jun Zhang (M'02-SM'08) received the Ph.D. degree in Electrical Engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2002. Since 2004, he has been with Sun Yat-Sen University, Guangzhou, China, where he is currently a Cheung Kong Professor. He has authored seven research books and book chapters, and over 100 technical papers in his research areas. His current research interests include computational intelligence, cloud computing, big data, and wireless sensor networks. Dr. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011 and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Industrial Electronics, and the IEEE Transactions on Cybernetics. He is the Founding and Current Chair of the IEEE Guangzhou Subsection, the Founding and Current Chair of ACM Guangzhou Chapter.

Guoqiang Han is a Professor at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is the head of the School of Computer Science and Engineering in SCUT. He received his B.Sc. degree from the Zhejiang University in 1982, and the Master and Ph.D. degree from the Sun Yat-Sen University in 1985 and 1988 respectively. His research interests include multimedia, computational intelligence, machine learning and computer graphics. He has published over 100 research papers.