Hybrid Incremental Ensemble Learning for Noisy Real-World Data Classification

Zhiwen Yu, *Senior Member, IEEE*, Daxing Wang, Zhuoxiong Zhao, C. L. Philip Chen, *Fellow, IEEE*, Jane You, Hau-San Wong, and Jun Zhang, *Fellow, IEEE*

***Abstract*—Traditional ensemble learning approaches explore the feature space and the sample space, respectively, which will prevent them to construct more powerful learning models for noisy real-world dataset classification. The random subspace method only search for the selection of features. Meanwhile, the bagging approach only search for the selection of samples. To overcome these limitations, we propose the hybrid incremental ensemble learning (HIEL) approach which takes into consideration the feature space and the sample space simultaneously to handle noisy dataset. Specifically, HIEL first adopts the bagging technique and linear discriminant analysis to remove noisy attributes, and generates a set of bootstraps and the corresponding ensemble members in the subspaces. Then, the classifiers are selected incrementally based on a classifier-specific criterion function and an ensemble criterion function. The corresponding weights for the classifiers are assigned during the same process. Finally, the final label is summarized by a weighted voting scheme, which serves as the final result of the classification. We also explore various classifier-specific criterion functions based on different newly proposed similarity measures, which will alleviate the effect of noisy samples on the distance functions. In addition, the computational cost of HIEL is analyzed theoretically. A set of nonparametric tests are adopted to compare HIEL and other algorithms over several datasets. The experiment results show that HIEL performs well on the noisy datasets. HIEL outperforms most of the compared classifier ensemble methods on 14 out of 24 noisy real-world UCI and KEEL datasets.**

***Index Terms*—Bagging, classification, classifier ensemble, ensemble learning, linear discriminant analysis (LDA).**

Manuscript received August 3, 2017; revised November 3, 2017 and November 12, 2017; accepted November 13, 2017. Date of publication December 4, 2017; date of current version January 15, 2019. This work was supported in part by the NSFC under Grant 61722205, Grant 61572199, Grant 61572540, and Grant U1611461, in part by the Guangdong Natural Science Funds under Grant S2013050014677 and Grant 2017A030312008, in part by the Science and Technology Planning Project of Guangdong Province, China, under Grant 2015A050502011, Grant 2016B090918042, Grant 2016A050503015, and Grant 2016B010127003, in part by the Macau Science and Technology Development under Grant 019/2015/A and Grant 024/2015/AMJ, in part by the Multiyear Research Grants from the University of Macau, in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant CityU 11300715, in part by the Hong Kong General Research Grant under Grant 152202/14E, in part by the PolyU Central Research Grant (G-YBJW), in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project CityU 11300715, and in part by the City University of Hong Kong under Project 7004884. This paper was recommended by Associate Editor D. Tao. *(Corresponding author: Zhiwen Yu.)*

Z. Yu, D. Wang, Z. Zhao, and J. Zhang are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: zhwyu@scut.edu.cn).

C. L. P. Chen is with the Department of Computer and Information Science, University of Macau, Macau 999078, China.

J. You is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong.

H.-S. Wong is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The file contains data and figures relevant to the paper. The material is 4 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2774266

I. Introduction

ENSEMBLE learning [1]–[10], as a very important branch of machine learning, is gaining increasingly more attention. This is in view of its effectiveness in handling different kinds of real-world datasets in the fields of data mining [11], [12], intelligent transportation system [13], [14], bioinformatics [15], [16], as well as pattern recognition [17], [18]. For example, in the field of data mining, Wang *et al.* [11] applied the resampling-based ensemble methods for online class imbalance learning. In the area of video processing, Bashbaghi *et al.* [13] used dynamic support vector machine (SVM) based ensembles for still-to-video face recognition. In the area of bioinformatics, Li *et al.* [15] applied the classifier ensemble approach for self-interacting proteins prediction from amino acids sequences. In the area of pattern recognition, Guan *et al.* [17] applied the classifier ensemble method for human gait recognition. When compared with single learning approaches, the ensemble learning approaches are able to combine multiple classifiers generated with different conditions into a single classifier, and generate more stable, robust, and accurate results.

In recent decades, a number of ensemble learning approaches are proposed [1]–[10], [59]–[68]. However, many of these approaches [1]–[10], [59]–[68] only explore the data sample space, such as bagging, adaboost, ensemble based on neural networks, or the feature space, such as rotation forest and random subspace. However, considering the data the feature space or sample space only is not enough to train powerful classifier ensemble for noisy real-world datasets. For example, patterns in some datasets can be identified in certain subspaces, while patterns in other datasets can be distinguished in the original space by selecting or weighting the training samples. In order to address the above limitations, we designed a hybrid incremental ensemble learning (HIEL) algorithm which explores the feature space and the sample space simultaneously for noisy real-world dataset classification. When compared with conventional ensemble learning algorithms, HIEL adopts the bagging technique to generate

diverse ensemble members, uses linear discriminant analysis (LDA) to explore different feature subspaces, and applies the adaboost technique to identify the importance of data samples and change the weights of the members of ensemble incrementally. Specifically, HIEL first generate a set of bootstraps with bagging, which is useful for reducing the effect of noisy samples. Then, LDA, [53], [54] is applied to perform dimension reduction for each bootstrap, and to reduce the effect of noisy attributes by building a linear classifier in a subspace. Next, the classifiers are selected incrementally, and their corresponding weights are assigned sequentially according to a classifier-specific criterion function and an ensemble criterion function. Finally, the final result is generated with a weighted voting scheme. The time complexity of HIEL are then explained theoretically. We also used the nonparametric tests to compare HIEL with its competitors on multiple datasets. The experiments conducted show that HIEL performs well on noisy datasets, and works better than the state-of-the-art ensemble approaches on 14 out of 24 datasets.

The contribution of this paper is fivefold. First, the HIEL approach is proposed to solve the problem of noisy real-world dataset classification, which explores both the feature space and the sample space. Second, a classifier-specific criterion function and an ensemble criterion function are adopted to perform incremental classifier selection (ICS). Third, various classifier-specific criterion functions based on different similarity measures of bootstraps are introduced. The effect of noisy samples are alleviated by these criterion functions. Fourth, we propose to adopt Kappa analysis to evaluate the effect of the ICS process. Fifth, HIEL is compared with other classifier ensemble algorithms with the nonparametric tests.

The rest of this paper is as follows. Section II discusses ensemble learning and its related works. Section III describes the HIEL algorithm and the incremental selection of classifiers. Section IV presents the similarity measures of bootstraps. The performance of our proposed approach is experimentally measured in Section V. Section VI concludes this paper and presents our future work.

II. Related Work

Classifier ensemble is gaining more and more importance in different application areas, since it is able to combine multiple classifiers into one classifier, and obtain more robust, stable, and accurate result. A lot of new ensemble classification algorithms have been proposed in recent years. Some researchers studied how to generate new classifier ensembles [19]–[23]. For example, Rahman and Verma [19] generated a classifier ensemble by clustering data at multiple layers. Zhang *et al.* [20] designed the ensemble classifier using the multiobjective deep belief networks ensemble. Amasyali and Ersoy [21] proposed to use an extended feature space to generate a classifier ensemble. Nanni and Lumini [22] generated a classifier ensemble by weighting the features in random subspaces. Mao *et al.* [23] proposed the weighted classifier ensemble by balancing the diversity and accuracy in the group of classifiers. Some researchers investigate the voting scheme in the ensemble, such as the weighted consult-and-vote scheme [24]. Some researchers explore the properties of an ensemble of classifiers. For example, Kuncheva [25] analyzed the properties of classifier ensembles using the Kappa-error diagram. Ludwig *et al.* [26] considered how to improve the generalization ability of the ensemble. Some researchers study how to use classifier ensemble approaches in different applications, such as electromyographic signal decomposition [27], pedestrian detection [28], bacterial virulent proteins identification [29], network intrusion detection [31], protein structural prediction [30], vision-based human detection [32], and so on. For example, Liu *et al.* [56]–[58] explored different classification methods to handle the noisy labels, proposed the spectral ensemble clustering technique to improve the accuracy of the ensemble, and studied various k-dimensional coding schemes for dealing with the subspaces. The characteristics of different decision tree ensemble creation techniques are extensively reviewed in [33].

Most of the classifier ensemble approaches can be divided into two main types, namely those which focus on the data sample space, and those which focus on the feature space. The approaches in the first type explore different combinations of sample subsets, which include the bagging approach [1] and the adaboost approach [21]. The bagging approach explores the data sample space by a suitable sampling technique, while the adaboost approach explores the data sample space by reweighting the training samples. The approaches in the second type study the combination of different feature subsets, which include the random subspace approach [9], the random forest approach [9], and the rotation forest approach [9]. The random subspace approach explores the feature space by a suitable sampling technique, while the rotation forest approach explores the feature space by principal component analysis and sampling. In summary, most of the classifier ensemble approaches study how to combine the classifiers in an optimal way from either the viewpoint of the data sample space or the feature space.

How to search for the optimal combination of the classifiers is an important issue in the area of classifier ensemble. There are a number of previous works on classifier ensemble selection [6], [34]–[38]. For example, Diao *et al.* [6] converted the classifier ensemble selection problem to a feature selection problem. Soto *et al.* [34] designed a double pruning scheme to boosting ensembles, and searched for the optimal ensemble classifiers. Qian *et al.* [35] studied how to select monotonic decision trees in the ensemble. Martínez-Muñoz *et al.* [36] analyzed ensemble pruning techniques theoretically. Ko *et al.* [37] considered ensemble selection instead of individual classifier selection. Woloszynski and Kurzynski [38] proposed a probabilistic model for ensemble selection. However, most of these works only consider either classifier selection in the ensemble generated from the data sample space or the feature space. Few of them take into consideration classifier ensemble selection generated from the data sample space and the feature space at the same time.

We pay attention to the bagging approaches [39]–[42], which has been successfully applied to noisy and imbalanced data classification [41]–[43], image retrieval [48]–[50], and so on. Fig. 1 provides an overview of the bagging technique.

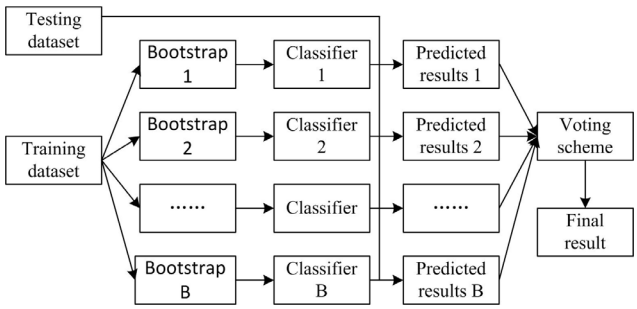


Fig. 1. Structure of the bagging approach.

In this paper, an HIEL approach is designed to address the limitations of the bagging method and handle the noisy real-world data classification problem. HIEL not only adopts an incremental selection process based on a classifier-specific cost function and an ensemble cost function to search for the optimal combination of classifiers associated with the different bootstraps but also establish the relationships of the classifiers in the ensemble by updating the weights of training samples in each iteration.

Exploiting an optimal feature subspace for the subsequent boosting classifier is an important idea of the proposed approach HIEL. For example, Bian and Tao [69] designed the max–min distance analysis approach based on the sequential convex relaxation algorithm for dimension reduction. Xu *et al.* [70] proposed the large margin optimization criteria for dimensionality reduction in a weakly supervised setting. Zhang and Zhou [71] applied dimensionality reduction for enhancing multilabel applications. Xu *et al.* [72] applied feature subspace learning for improving multiview applications.

III. Hybrid Incremental Ensemble Learning

Fig. 2 provides the basic structure of the HIEL approach. In order to perform HIEL, we need a training set P_r. The number of training samples in P_r is l. The samples are represented as $P_r = \{(p_1, y_1), (p_2, y_2), \ldots, (p_l, y_l)\}$, where p_i $(i \in \{1, \ldots, l\})$ represents a training sample, y_i is its label, $y_i \in \{-1, 1\}$ for binary classification. Label y_i can also be extended to suit multiclass problems. Each sample p_i consists of d attributes. First, the HIEL generates some bootstraps $\widehat{O} = \{O^1, O^2, \ldots, O^B\}$ (B is the number of bootstraps). Specifically, HIEL first uses a sampling rate $\varrho \in [\varrho_{\min}, \varrho_{\max}]$ to generate a fraction of training samples in the bootstraps compared among all the samples as follows:

$$\varrho = \varrho_{\min} + \lfloor \tau_1(\varrho_{\max} - \varrho_{\min}) \rfloor \tag{1}$$

where $\tau_1(\tau_1 \in [0, 1])$ is uniformly distributed. Then, the training samples are selected one by one. Their index is selected as follows:

$$j = \lfloor 1 + \tau_2 l \rfloor \tag{2}$$

where j represents the index of the training sample being selected, and τ_2 is also uniformly distributed. The HIEL repeats this process to select ϱl training samples. Each bootstrap is constructed by these selected training samples. At last,

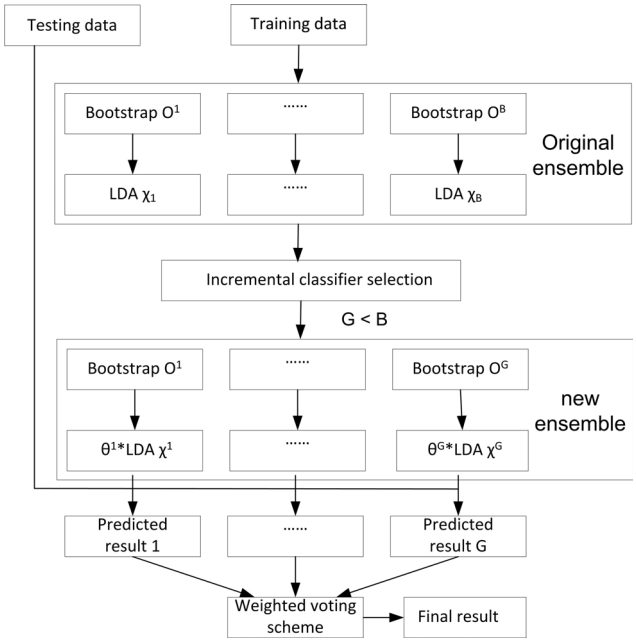


Fig. 2. Structure of the HIEL approach.

HIEL will obtain a set of bootstraps $O^1, O^2, \ldots, O^B$ by repeating the above process B times. The intuition of adopting the bagging technique is as follows: since the proportion of noisy training samples to total training samples is small, most of the bootstraps do not contain the noisy samples or only contain a very small potion of noisy samples. As a result, the bagging technique is robust to the effect of noisy samples.

Then, a group of LDA classifiers $(\chi_1, \chi_2, \ldots, \chi_B)$ is trained using their corresponding bootstraps $(\{O^1, O^2, \ldots, O^B\})$. The main motivation of adopting the LDA classifier is that it will search for a combination of attributes to identify data samples belonging to different classes in the subspace, which is able to reduce noisy and redundant attributes in the dataset. The objective function Ξ^b of the LDA classifier χ_b is defined as follows [53], [54]:

$$\Xi^b = \arg \min_{y^b \in \{1,\ldots,K\}} \sum_{k=1}^{K} \Lambda\left(k|p^b\right) \Upsilon\left(y^b|k\right) \tag{3}$$

where K represents the number of categories, or classes, and $\Lambda(k|p^b)$ is the posterior probability of the class k for the data sample p^b in the bootstrap O^b, and $\Upsilon(y^b|k)$ is the cost of classifying a data sample as y^b when its true class is k. If the data sample y^b is correctly classified, $\Upsilon(y^b|k) = 0$. Otherwise, $\Upsilon(y^b|k) = 1$.

$\Lambda(k|p^b)$ is calculated as follows:

$$\Lambda\left(k|p^b\right) = \frac{\Lambda\left(p^b|k\right)\Lambda(k)}{\Lambda\left(p^b\right)} \tag{4}$$

$$\Lambda\left(p^b|k\right) = \frac{1}{(2\pi\,|\Sigma_k|)^{\frac{1}{2}}} e^{-\frac{1}{2}\left(p^b-\mu_k^b\right)^T \Sigma_k^{-1}\left(p^b-\mu_k^b\right)} \tag{5}$$

where μ_k^b and Σ_k denote the mean matrix and the covariance matrix of the class k in the bootstrap O^b, $|\Sigma_k|$ and Σ_k^{-1} are the determinant and the inverse matrix of Σ_k, and $\Lambda(p^b)$ is a

Algorithm 1 HIEL Algorithm

Require:

Input: the training set P_r and the testing set P_e;

Ensure:

Generating original ensemble;
  Generate B bootstraps $\{O^1, O^2, \ldots, O^B\}$;
  Generate the corresponding classifiers $\chi_1, \chi_2, \ldots, \chi_B$ using LDA with Eq.(1)-(3);
Call Incremental classifier selection in Algorithm 2;
Generating new ensemble;
  Generate G bootstraps $\{O^1, O^2, \ldots, O^G\}$ $(G < B)$;
  Obtain the corresponding classifiers $\chi^1, \chi^2, \ldots, \chi^G$;
Evaluation of sample labels in P_e;
Weighted voting for the final classification results;

Output: the labels of the samples in P_e.

normalization constant. $\Lambda(k)$ is the proportion of samples in the class k in all the samples in O^b.

The HIEL algorithm will calculate each classifier's accuracy value ($\xi_j (j \in \{1, \ldots, B\})$) which is the corresponding LDA on P_r.

Next, the ICS process, as shown in Algorithm 2, is performed on the ensemble, such that G $(G < B)$ bootstraps and their classifiers will be chosen based on a classifier-specific criterion function and an ensemble criterion function. Comparing the classifiers in the new ensemble with those in the original one, the classifiers in the new ensemble have two properties: 1) the classifiers in the new ensemble are ordered sequentially based on the ICS and 2) each classifier is associated with a weight value.

The selected classifiers are used when evaluating the samples in the test set. The results are weighted voted to obtain the final result. The predicted labels of the bth classifier χ^b $(b \in \{1, \ldots, B\})$ is represented as $Y^b = \{y_1^b, y_2^b, \ldots, y_l^b\}$. The labels of the test samples are determined as follows:

$$y^* = \arg\max_c \quad \vartheta_c(f_i) \tag{6}$$

$$\vartheta_c(f_i) = \sum_{b=1}^{B} \theta^b \cdot \left\{ y_i^b = c \right\} \tag{7}$$

where y_i is the label of the ith sample f_i, $c \in \{0, 1, \ldots, k-1\}$, and k is the number of classes.

Finally, the final result is calculated by weighted voting. Algorithm 1 provides a flow chart of the HIEL.

Algorithm 2 provides an overview of ICS. The input of the process is the original classifier ensemble, and the output is the new ensemble which is similar to adaboost. At first, ICS sets the weights of the samples as $(1/l)$, where l is the size of training set. The classifier χ^1 with the greatest accuracy is selected as follows:

$$\chi^1 = \arg\max_{\chi_j \in \widehat{\Gamma}} \quad \xi_j \tag{8}$$

where $\widehat{\Gamma} = \{\chi_1, \chi_2, \ldots, \chi_B\}$ classifiers in the ensemble. The weighted error of the first classifier χ^1 is defined as follows:

$$\epsilon^1 = \sum_i \omega_i^1 \Theta\Big(\chi^1(P), y, i\Big)\Big). \tag{9}$$

Algorithm 2 ICS

Require:

Input: the training sample set $P = (p_1, p_2, \ldots, p_l)$;
the corresponding label set $Y = (y_1, y_2, \ldots, y_l)$;
the error function $\Theta(\chi(P), y, i)) = e^{-y_i \chi(p_i)}$, $i \in \{1, \ldots, l\}$;
a set of bootstraps $\widehat{O} = \{O^1, O^2, \ldots, O^B\}$
the corresponding classifiers $\widehat{\Gamma} = \{\chi_1, \chi_2, \ldots, \chi_B\}$, $\chi: P \to [-1, 1]$;
the empty ensemble $\Gamma(P)$;

Ensure:

Initial weights $\omega_1^1 = ... = \omega_l^1 = \frac{1}{l}$ for all the samples;
For g in $1, \ldots, G$:
  If $g = 1$,
    Find the first classifier $\chi^1 = \arg\max_{\chi_j \in \widehat{\Gamma}} \quad \xi_j$;
      The weighted sum error for points $\epsilon^1 = \sum_i \omega_i^1 \Theta(\chi^1(P), y, i))$;
    Calculate $\theta^1 = \frac{1}{2} \ln(\frac{1-\epsilon^1}{\epsilon^1})$;
    Add to new ensemble $\Gamma^1(P) = \theta^1 \chi^1$;
  Else
    Choose $\chi^g(P)$:
    Consider each classifier $\chi_j^g(P) \in \widehat{\Gamma}$ $(j \in \{1, \ldots, |\widehat{\Gamma}|\})$:
      Compute the classifier-specific criterion function $\Pi_1(\chi_j^g(P))$;
      Compute the weighted sum error for points $\epsilon_j^g = \sum_i \omega_i^g \Theta(\chi_j^g(P), y, i))$;
      Compute $\theta_j^g = \frac{1}{2} \ln(\frac{1-\epsilon_j^g}{\epsilon_j^g})$;
    Order the classifiers in descending order with respect to their classifier-specific criterion function values.
    j=0;
    Repeat
      j= j+1;
    Until the ensemble criterion function $\Pi_2(\Gamma^{g-1}(P)) \leq \Pi_2(\Gamma^{g-1}(P) + \theta_j^g \chi_j^g)$;
    Add to new ensemble: $\Gamma^g(P) = \Gamma^{g-1}(P) + \theta_j^g \chi_j^g$;
  End
  Update weights:
  $\omega_i^{g+1} = \omega_i^g e^{-y_i \theta_j^g \chi_j^g(p_i)}$ for every sample;
  Re-normalize ω_i^{g+1}, so that $\sum_i \omega_i^{g+1} = 1$;

Output: the new ensemble Γ^G.

The weight θ^1 of the first classifier χ^1 is updated as follows:

$$\theta^1 = \frac{1}{2} \ln\left(\frac{1 - \epsilon^1}{\epsilon^1}\right). \tag{10}$$

The classifier χ^1 with its weight θ^1 will be selected as part of the new ensemble as follows:

$$\Gamma^1(P) = \theta^1 \chi^1. \tag{11}$$

The classifier is removed from $\widehat{\Gamma}$. The new weights ω_i^2 of the samples are updated as

$$\omega_i^2 = \omega_i^1 e^{-y_i \theta^1 \chi^1(p_i)} \tag{12}$$

where $\Theta(\chi(P), y, i)) = e^{-y_i \chi(p_i)}$ is the error function. The weights are normalized, and the following condition is satisfied:

$$\sum_{i=1}^{l} \omega_i^2 = 1. \tag{13}$$

Then, ICS will select $G-1$ classifiers one after another (where G is a predefined parameter by the user). For each classifier χ_j in $\widehat{\Gamma}$, a classifier-specific criterion function $\Pi_1(\chi)$ is defined as follows:

$$\Pi_1(\chi_j) = \beta_1 \xi_j + \beta_2 \phi\left(O^j, O^h\right) \tag{14}$$

where ξ_j is the precision of the classifier on the newly weighted data set, and $\phi(O^j, O^h)$ is the similarity between the bootstrap O^j with the classifier χ_j and the bootstrap O^h with the classifier chosen in the previous iteration χ^{g-1}. The advantage of this definition is as follows: the classifier-specific criterion function $\Pi_1(\chi_j)$ considers both the accuracy of the classifier and the similarity between two contiguous bootstraps, through which an ensemble with more diversity can be generated to obtain more accurate and stable results. The weight parameters β_1 and β_2 are used to balance the importance of the two components in the above function.

The weighted error for all the classifiers χ_j^g is computed as follows:

$$\epsilon_j^g = \sum_i \omega_i^g \Theta\left(\chi_j^g(P), y, i\right) \tag{15}$$

where $j \in \{1, \ldots, |\widehat{\Gamma}|\}$, $g \in \{1, \ldots, G\}$ is the current iteration. The weight θ_j^g of the classifier χ_j^g is reweighted as follows:

$$\theta_j^g = \frac{1}{2} \ln\left(\frac{1-\epsilon_j^g}{\epsilon_j^g}\right). \tag{16}$$

All the classifiers are ordered in descending order with respect to their classifier-specific criterion function values $\Pi_1(\chi_j)$. The classifier which will diversify the ensemble will be considered first in the selection process.

Next, an ensemble criterion function $\Pi_2(\Gamma)$ is adopted to select the classifier which is added into the new ensemble in the current iteration. $\Pi_2(\Gamma)$ is defined as follows:

$$\Pi_2(\Gamma) = \sum_{i=1}^{l} |y_i - \Gamma(p_i)| \tag{17}$$

$$\Gamma(p_i) = \arg\max_c \sum_{h=1}^{g} \theta^h \cdot 1\left\{\chi^h(p_i) = c\right\} \tag{18}$$

where $c \in \{-1, 1\}$ is the set of labels, and χ^h denotes the hth LDA classifier in the new ensemble. The weighted voting scheme can derive better classification result as it weight the classifiers according to their importance instead of treat them all equally.

If $\Pi_2(\Gamma^{g-1}(P)) < \Pi_2(\Gamma^{g-1}(P) + \theta^g \chi_j^g)$, the next classifier will be considered. Otherwise, the classifier χ_j^g will be selected from $\widehat{\Gamma}$ as part of the ensemble as

$$\Gamma^g(P) = \Gamma^{g-1}(P) + \theta_j^g \chi_j^g. \tag{19}$$

After that, HIEL updates the weights ω_i^g for the $g+1$th iteration as follows:

$$\omega_i^{g+1} = \omega_i^g e^{-y_i \theta_j^g \chi_j^g(p_i)}. \tag{20}$$

Then, the weights are normalized to one as follows:

$$\sum_{i=1}^{l} \omega_i^{g+1} = 1. \tag{21}$$

At last, the incremental selection will select G classifiers and stop. It may stop earlier if the ensemble criterion function stop to change.

IV. Similarity Measures of Bootstraps

The measure $\phi(O^j, O^h)$ in the classifier-specific criterion function is determined by the similarity between two bootstraps O^j and O^h. Specifically, the bootstraps O^j and O^h can be represented by Gaussian mixture models (GMMs) Ω^j and Ω^h, respectively. The intuition is that the similarity between two GMMs is not affected by noisy samples in the bootstraps. The parameter values of GMMs are initialized by K-means, while the expectation-maximization approach is adopted to determine the optimal parameter values.

Given two GMMs $\Omega^j = \{\Phi_1^j, \Phi_2^j, \ldots, \Phi_{K_1}^j\}$ associated with the weight values $\pi_1^j, \pi_2^j, \ldots, \pi_{K_1}^j$ and $\Omega^h = \{\Phi_1^h, \Phi_2^h, \ldots, \Phi_{K_2}^h\}$ associated with the weight values $\pi_1^h, \pi_2^h, \ldots, \pi_{K_2}^h$ (where K_1 and K_2 denote the number of components in the GMMs Ω^j and Ω^h, respectively), several candidate measures $\phi(O^j, O^h)$ are proposed to evaluate the similarity of two bootstraps.

$\phi_1(O^j, O^h)$, which considers the closest pair of Gaussian distributions, is computed as follows:

$$\phi_1\left(O^j, O^h\right) = \min_{k_1 \in \{1,\ldots,K_1\}, k_2 \in \{1,\ldots,K_2\}} \psi\left(\Phi_{k_1}^j, \Phi_{k_2}^h\right) \tag{22}$$

$$\psi\left(\Phi_{k_1}^j, \Phi_{k_2}^h\right) = \frac{1}{8}\left(\mu_{k_1}^j - \mu_{k_2}^h\right)^T \left(\frac{\Sigma_{k_1}^j + \Sigma_{k_2}^h}{2}\right)^{-1} \left(\mu_{k_1}^j - \mu_{k_2}^h\right) + \frac{1}{2} \ln \frac{\left|\frac{\Sigma_{k_1}^j + \Sigma_{k_2}^h}{2}\right|}{\sqrt{\left|\Sigma_{k_1}^j\right|\left|\Sigma_{k_2}^h\right|}} \tag{23}$$

where $\psi(\Phi_{k_1}^j, \Phi_{k_2}^h)$ denotes the Bhattacharyya distance between two Gaussian distributions $\Phi_{k_1}^j$ and $\Phi_{k_2}^h$, $\mu_{k_1}^j$, $\mu_{k_2}^h$, $\Sigma_{k_1}^j$, and $\Sigma_{k_2}^h$ are the mean vectors and the covariance matrices of the Gaussian distributions $\Phi_{k_1}^j$ and $\Phi_{k_2}^h$, respectively.

$\phi_2(O^j, O^h)$, which takes into account the farthest pair of Gaussian distributions, is calculated as follows:

$$\phi_2\left(O^j, O^h\right) = \max_{k_1 \in \{1,\ldots,K_1\}, k_2 \in \{1,\ldots,K_2\}} \psi\left(\Phi_{k_1}^j, \Phi_{k_2}^h\right). \tag{24}$$

$\phi_3(O^j, O^h)$, which considers the average similarity among the pairs of Gaussian distributions, is defined as follows:

$$\phi_3(O^j, O^h) = \frac{1}{K_1 K_2} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \psi\left(\Phi_{k_1}^j, \Phi_{k_2}^h\right). \tag{25}$$

$\phi_4(O^j, O^h)$, which is based on the average weighted similarity among the pairs of Gaussian distributions is calculated as follows:

$$\phi_4\left(O^j, O^h\right) = \frac{1}{K_1 K_2 \sum_h \sum_l \pi_{k_1}^j \pi_{k_2}^h} \times \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \pi_{k_1}^j \pi_{k_2}^h \psi\left(\Phi_{k_1}^j, \Phi_{k_2}^h\right). \quad (26)$$

In general, the classifier-specific criterion function $\Pi_1(\chi)$ takes into account the classification accuracy on the weighted sample set as well as the bootstraps diversity based on different similarity measures.

V. Complexity Analysis

A complexity analysis of the HIEL approach on its computational cost is delivered. The time complexity T_{HIEL} of HIEL is estimated as follows:

$$T_{\text{HIEL}} = T_{\text{OE}} + T_{\text{ICS}} + T_{\text{NE}} \quad (27)$$

where T_{OE}, T_{ICS}, and T_{NE} denote the computational costs for the generation of ensemble classifiers, the ICS, and the prediction process using new ensemble, respectively. T_{OE} is related to the size of training data set l, the number of attributes m, and the number of classifiers B as follows:

$$T_{\text{OE}} = O\left(B \cdot \left(l \cdot m \cdot t + t^3\right)\right) \quad (28)$$

where $t = \min(l, m)$. T_{ICS} will be affected by the number of training samples l, the number of attributes m, the number of classifiers B, maximum number of representative centers in the subspace $k_{\max}$, and the number of iterations G as follows:

$$T_{\text{ICS}} = O\left(G \cdot \left(B \cdot \left(l \cdot m \cdot m + l \cdot m \cdot t + t^3\right) + B \log B\right)\right. . \quad (29)$$

T_{NE} 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 training set as follows:

$$T_{\text{NE}} = O(G \cdot l \cdot m). \quad (30)$$

Since G and B are much smaller than $l \cdot m \cdot m$, the time complexity of HIEL is $O(lm^2 + lmt + t^3)$. The memory consumption is $O(lm + mt + nt)$.

When compared with the random subspace approach, the HIEL approach is characterized with the following properties.

1) HIEL adopts the LDA method to construct a low dimensional space instead of using a random subspace.
2) HIEL selects the classifiers incrementally based on the classifier-specific and ensemble criterion functions which reduces the effect of the classifiers with less importance.
3) The classifiers generated by random subspace approach are not closely related, while HIEL selects classifiers incrementally.
4) The final labels are determined by weighted voting rather than majority voting.

Both the adaboost approach and the HIEL generate the classifiers in the ensemble sequentially. They iteratively predict the results of the samples, reweight the sample, and generate new classifiers. When compared with the adaboost approach, HIEL has the following distinguishing features: 1) the classifiers are trained on subset of features instead of original features and 2) HIEL applies LDA to allow better discrimination of the samples.

TABLE I
Summary of Real-World Datasets (Where n Denotes the Number of Data Samples, m Denotes the Number of Attributes, and k Denotes the Number of Classes)

Dataset	Rename	Source	n	m	k
BreastTissue	S1	[45]	106	9	6
Climate	S2	[45]	540	18	2
Haberman	S3	[45]	306	3	2
Iris	S4	[45]	150	4	3
Mammographic	S5	[45]	961	4	2
Ozone_eighthr	S6	[45]	2534	73	2
Seeds	S7	[45]	210	7	3
User_knowledge	S8	[45]	403	5	5
Vehicle_sihouettes	S9	[45]	846	18	4
Vertebral_column	S10	[45]	310	6	2
WDBC	S11	[45]	569	30	2
Wine	S12	[45]	178	13	3
Wisconsin	S13	[45]	699	9	2
WPBC	S14	[45]	198	33	2
Balance	S15	[46]	625	4	3
Bupa	S16	[46]	345	6	2
Contraceptive	S17	[46]	1473	9	3
Heart	S18	[46]	270	13	2
Housevotes	S19	[46]	232	16	2
Ring	S20	[46]	7400	20	2
Saheart	S21	[46]	462	9	2
Spectfheart	S22	[46]	267	44	2
Texture	S23	[46]	5500	40	11
Twonorm	S24	[46]	7400	20	2

VI. Experiment

We compare HIEL and other classification ensemble methods using 24 noisy real-world datasets as shown in Table I (where n denotes the number of data samples, m denotes the number of attributes, and k denotes the number of classes). Fourteen of them are from the UCI machine learning repository [45], and 10 of them are from the KEEL repository [46].

The final results are measured by classification accuracy (AC). It is defined as follows:

$$\text{AC} = \frac{1}{|P_s|} \sum_{p_i \in P_s} 1\left\{y_i^{\text{ensemble}} == y_i^{\text{true}}\right\} \quad (31)$$

where P_s is the testing set, $|P_s|$ is the number of samples in P_s, y_i^{ensemble} and y_i^{true} denote the predicted label by the proposed ensemble learning approach HIEL and the true label of the sample p_i, respectively. The experiments are repeated ten times. The mean and the standard deviation of the accuracy are shown in the table. We adopt fivefold crossover validation to reduce the effect of randomness. All the single classifier approaches and classifier ensemble approaches under consideration are run on the Weka platform [47].

In the following experiments, we first explore the effect of the sampling rate and the similarity measure of bootstraps. Then, the effect of the basic classifier in HIEL and that of the ICS process are investigated. Next, we compare HIEL with different single classifier methods and classifier ensemble

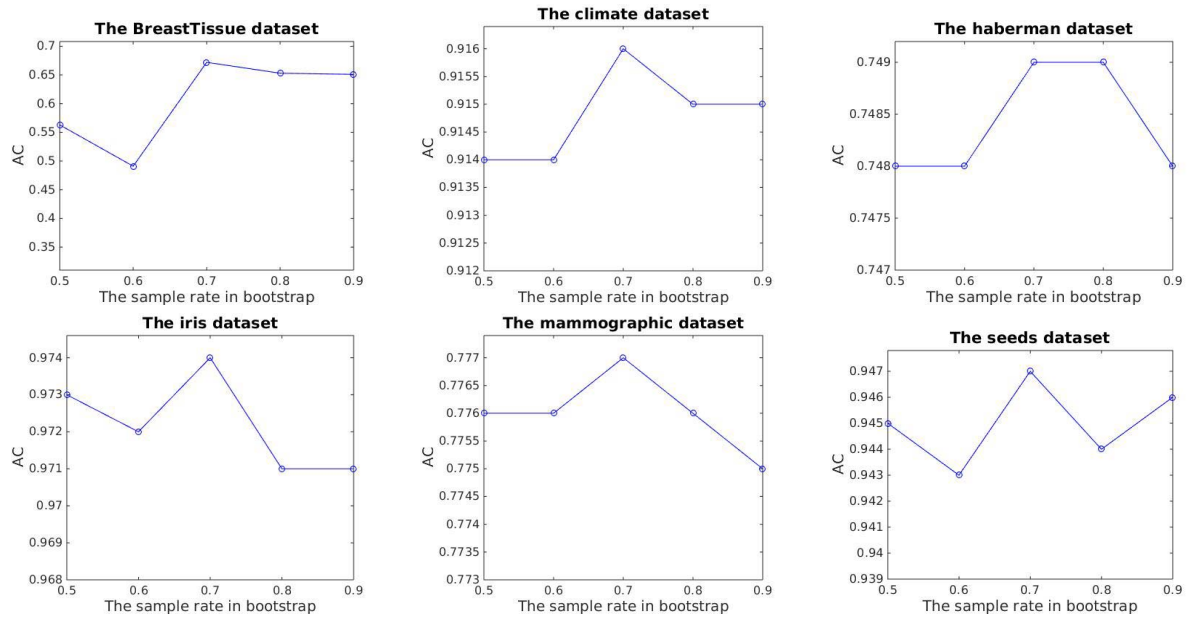


Fig. 3. Effect of the sampling rate.

TABLE II
EFFECT OF THE BOOTSTRAP SIMILARITY MEASURE (WHERE THE BEST VALUES ARE HIGHLIGHTED IN BOLD)

Datasets	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
HIEL-AWS	**0.672** (0.071)	**0.916** (0.009)	**0.749** (0.035)	**0.974** (0.028)	0.777 (0.026)	**0.931** (0.005)	**0.947** (0.031)	**0.912** (0.026)	**0.848** (0.024)	**0.829** (0.049)	**0.957** (0.019)	**0.982** (0.026)
HIEL-C	0.630 (0.091)	0.912 (0.008)	0.747 (0.031)	**0.974** (0.028)	**0.778** (0.029)	0.930 (0.006)	0.946 (0.036)	0.908 (0.024)	0.847 (0.024)	0.822 (0.043)	0.955 (0.018)	0.980 (0.029)
HIEL-F	0.628 (0.084)	0.912 (0.008)	**0.749** (0.037)	**0.974** (0.026)	0.778 (0.029)	0.929 (0.007)	0.945 (0.036)	0.911 (0.030)	**0.848** (0.024)	0.825 (0.041)	0.955 (0.018)	0.981 (0.022)
HIEL-AS	0.631 (0.084)	0.914 (0.005)	0.748 (0.032)	0.972 (0.027)	0.778 (0.029)	0.930 (0.005)	0.946 (0.034)	0.910 (0.022)	0.846 (0.023)	0.824 (0.050)	0.955 (0.014)	0.980 (0.028)
Datasets	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
HIEL-AWS	0.951 (0.017)	**0.763** (0.013)	**0.913** (0.021)	**0.626** (0.062)	**0.508** (0.024)	**0.827** (0.044)	**0.944** (0.041)	**0.979** (0.004)	**0.701** (0.041)	**0.802** (0.031)	**0.999** (0.001)	**0.977** (0.003)
HIEL-C	**0.952** (0.01)	0.761 (0.014)	0.912 (0.026)	0.608 (0.059)	0.507 (0.025)	0.819 (0.050)	0.943 (0.037)	0.975 (0.003)	0.700 (0.041)	0.747 (0.049)	0.997 (0.001)	0.975 (0.000)
HIEL-F	**0.952** (0.018)	0.762 (0.013)	0.911 (0.023)	0.606 (0.059)	0.507 (0.031)	0.818 (0.045)	0.940 (0.040)	0.975 (0.002)	0.699 (0.038)	0.753 (0.018)	0.997 (0.001)	0.976 (0.001)
HIEL-AS	0.951 (0.020)	0.762 (0.012)	**0.913** (0.021)	0.610 (0.058)	0.503 (0.026)	0.826 (0.054)	0.943 (0.036)	0.975 (0.002)	0.699 (0.044)	0.771 (0.034)	0.998 (0.001)	0.974 (0.002)

methods on the real-world datasets. Finally, a set of non-parametric tests is adopted to compare the classifier ensemble methods.

A. Effect of the Sampling Rate

The sampling rate in bootstrap is one of the important parameters that affects the accuracy of HIEL. Fig. 3 shows the effect of the sampling rates on the average values of accuracy on the several data sets, including the BreastTissue, the Climate, the Haberman, the Iris, the Mammographic, and the Seeds. It is observed that HIEL achieves better performance on all the datasets when the sampling rate is set to 0.7. With a sampling rate smaller than 0.7, the number of samples in the bootstraps could be too small to reflect the underlying structure of the original dataset. When the sampling rate increases from 0.7, more redundant and noisy samples might be included in the bootstraps, which will make the result unsatisfactory. In general, the sampling rate 0.7 is suitable for HIEL on most of the real-world datasets.

B. Effect of the Bootstrap Similarity Measure

The bootstrap similarity measure is another important factor which affects the quality of the final result through its role in the classifier-specific criterion function. We consider HIEL based on four types of similarity measures, which are HIEL based on the average weighted similarity among the pairs of Gaussian distributions (HIEL-AWS), HIEL based on the closest pair of Gaussian distributions (HIEL-C), HIEL based on the farthest pair of Gaussian distributions (HIEL-F), and HIEL based on the average similarity among the pairs of Gaussian distributions (HIEL-AS). Table II shows the results obtained by HIEL-AWS, HIEL-C, HIEL-F, and HIEL-AS. It can be seen that HIEL-AWS outperforms its competitors on 22 out of 24 datasets. The possible reason is that HIEL-AWS not

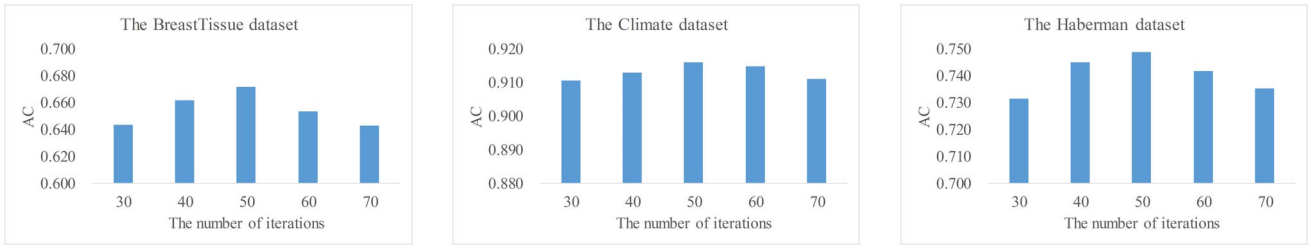


Fig. 4. Effect of the Number of Iterations

TABLE III
OVERALL PARAMETER SETTINGS

Parameters	Default values	Range
The bootstrap similarity measures	HIEL-AWS	HIEL-AWS, HIEL-C, HIEL-F, HIEL-AS
The sampling rate for the bootstrap technique	0.7	0.5, 0.6, 0.7, 0.8, 0.9
The number of iterations G	50	30, 40, 50, 60, 70
The number of basic classifiers B	300	100, 200, 300, 400, 500

only makes use of all the information of the pairs of Gaussian distributions generated by the bootstraps, but also takes into account the size of the bootstraps. HIEL-C and HIEL-F obtain unsatisfactory results on the BreastTissue dataset. The possible reason is that HIEL-C and HIEL-F only consider single pairs of Gaussian distributions, which will lead to information loss and decreases the performance of HIEL. In general, HIEL-AWS is a better choice, and the summary of the parameter settings is shown in Table III.

We vary the value of each of the parameters one at a time to investigate the effect of the parameters, and set the values of the other parameters to their default values. Fig. 4 shows the effect of the number of iterations (G) on the performance, while Fig. 5 illustrates the effect of the number of basic classifiers (B). It can be seen that when the number of iterations is set to 50, and the number of basic classifiers is set to 300, the proposed approach HIEL achieves better performance.

In order to explore the important role of the LDA, we compare the proposed approach HIEL with the approach (LDA+Adaboost), which first performs dimension reduction using LDA and then applies adaboost on the BreastTissue dataset, the Climate dataset, and the Haberman dataset. Table IV illustrates the comparison results obtained by the two approaches with respect to AC. It is observed that HIEL outperforms its competitor on these datasets, especially on the BreastTissue dataset.

TABLE IV
EFFECT OF THE LDA (WHERE THE BEST VALUES ARE HIGHLIGHTED IN BOLD)

Datasets	HIEL	LDA+ADABOOST
BreastTissue	**0.672**±0.071	0.379 ± 0.078
Climate	**0.916** ± 0.009	0.908 ± 0.013
Haberman	**0.749** ± 0.035	0.733 ± 0.044

C. Effect of the Basic Classifier

In order to explore the effect of the basic classifier, we replace LDA with the k-nearest neighbor (KNN) method or the decision tree. HIEL based on LDA (HIEL-LDA) is compared with HIEL based on the KNN method (HIEL-KNN), and HIEL based on decision tree (HIEL-DT) with respect to the average accuracy value on all the datasets in Table I. Fig. 6 shows the results obtained by HIEL, HIEL-KNN, and HIEL-DT. It can be seen that HIEL-LDA outperforms its competitors on 19 out of 24 datasets. For example, HIEL-LDA obtains the best results on the Balance dataset (S15) and the Ring dataset (S20) with average accuracy values 0.913 and 0.979, which are 0.100 and 0.095 larger than the second best results, respectively. The possible reason could be that LDA is capable of finding a suitable linear combination of features in a subspace which alleviates the problem of noisy attributes. HIEL-KNN obtains good results on the wisconsin dataset (S13), while HIEL-DT achieves good performance on the vertebral_column dataset (S10), the bupa dataset (S16) and the housevotes dataset (S19). In summary, HIEL-LDA based on LDA is preferred when dealing with different kinds of real-world datasets.

D. Effect of the Incremental Classifier Selection Process

In order to explore the effect of the ICS, we compare HIEL based on this selection process (HIEL-ICS) with HIEL without ICS (HIEL-WICS) with respect to the average values and the corresponding standard deviations of the accuracy on all the real-world datasets. Table V shows the results obtained by HIEL-ICS and HIEL-WICS. We observe that HIEL-ICS achieves the best performance on all datasets. In general, the ICS process allows HIEL to obtain better results. If this selection process in HIEL is missing, its effectiveness will be reduced.

We further perform Kappa analysis [55] to evaluate the diversity of the new ensemble generated by the ICS process as follows:

$$\kappa = \frac{\zeta_1 - \zeta_2}{1 - \zeta_2} \tag{32}$$

$$\zeta_1 = \frac{\sum_{c=1}^{k} S_{cc}}{l} \tag{33}$$

$$\zeta_2 = \sum_{c=1}^{k} \left(\sum_{h=1}^{k} \frac{S_{ch}}{l} \cdot \sum_{h=1}^{k} \frac{S_{hc}}{l} \right) \tag{34}$$

where S_{ch} denotes the number of samples p for which $\chi(p) = c$ and $\chi'(p) = h$, and χ, χ' denote two classifiers. If κ is equal to 1, the predicted results obtained by χ and χ' are the same. A small κ value corresponds to a greater difference between the predicted results obtained by the two

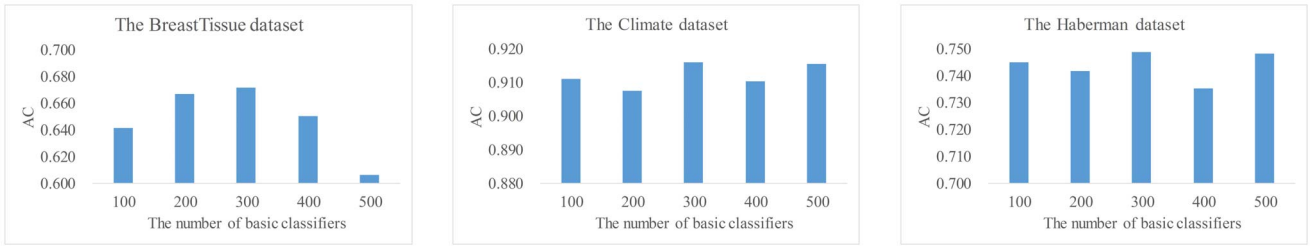


Fig. 5. Effect of the number of the basic classifiers.

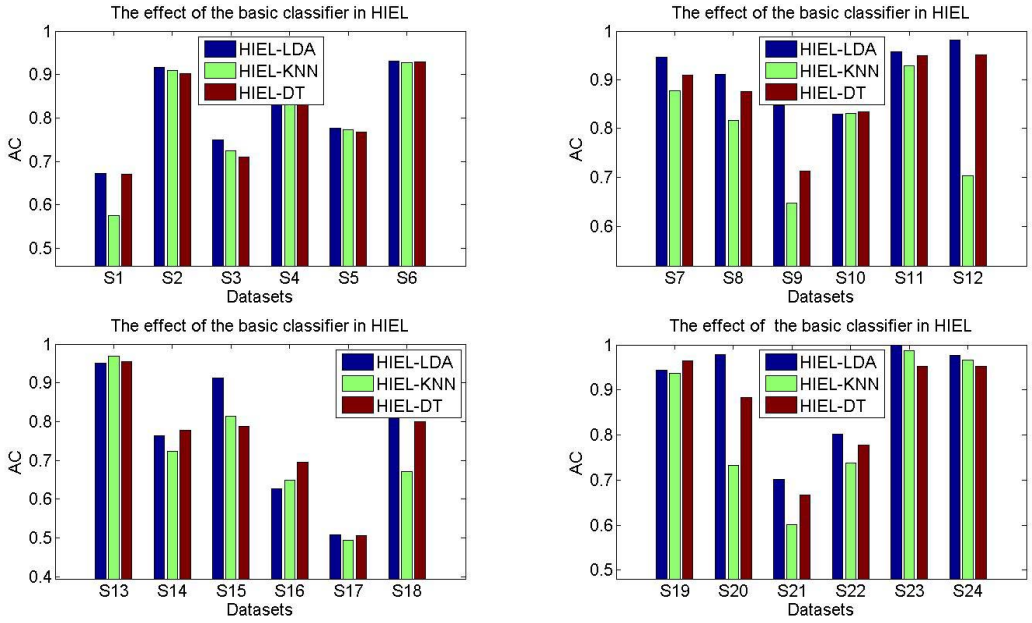


Fig. 6. Effect of the basic classifier in HIEL.

classifiers. The reason that we adopt Kappa-error analysis is as follows: Kappa-error diagrams are used to evaluate the diversity of ensemble classifiers, which provide insight on why the proposed HIEL method combined with classifier selection is better than another approach on a given data set.

We first calculate the average kappa value for all the pairs of classifiers in the original ensemble using HIEL-WICS. Then, the average kappa value for all the classifier pairs in the new ensemble generated by HIEL-ICS is computed. Table VI shows the results of Kappa analysis obtained by HIEL-ICS and HIEL-WICS on all the real-world datasets in Table I. It is observed that the average kappa values on all the datasets obtained by HIEL-ICS are smaller than those by HIEL-WICS. For example, the average kappa value of HIEL-ICS on the BreastTissue dataset (S1) is 0.432, which is 0.206 smaller than that by HIEL-WICS. This indicates that the diversity of the new ensemble generated by HIEL-ICS is higher than that of HIEL-WICS, and leads to improved performance of the ensemble classifier.

E. Comparison With Single Classifier Approaches

HIEL is compared with traditional classifier approaches, which include the KNN approach, the random tree (RT) approach, the SVM, the J48 decision tree (J48), and the decision stump (DS) approach corresponding to the average values and the corresponding standard deviations of the accuracy. Table VII shows the results of different single classification algorithm. We can see that HIEL achieves the highest accuracy on 17 out of 24 datasets. This mainly because: 1) HIEL integrates multiple classifiers in the ensemble to a unified one and improve the performance of HIEL and 2) it assigns different weights to different classifiers, which reflect the relative importance of these classifiers. On the other hand, while HIEL leads to better accuracy, it requires a higher computational cost due to the need to train all the classifiers. In general, we need to consider the tradeoff between accuracy and computational costs for HIEL when handling different classification problems.

F. Comparison With Classifier Ensemble Approaches

We compare HIEL with conventional classifier ensemble approaches on the real-world datasets in Table I with respect to the accuracy, which include the random forest approach (RaForest, [3]), the AdaboostM1 approach (AdaboostM1, [2]), the bagging approach (Bagging, [1]), the rotation forest approach (Ro-Forest, [5]), the classifier ensemble method

TABLE V
EFFECT OF THE ICS APPROACH (WHERE THE BEST VALUES ARE INDICATED IN BOLD)

Datasets	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
HIEL-ICS	**0.672** (0.071)	**0.916** (0.009)	**0.749** (0.035)	**0.974** (0.028)	**0.777** (0.026)	**0.931** (0.005)	**0.947** (0.031)	**0.912** (0.026)	**0.848** (0.024)	**0.829** (0.049)	**0.957** (0.019)	**0.982** (0.026)
HIEL-WICS	0.642 (0.068)	0.907 (0.010)	0.738 (0.031)	0.967 (0.002)	0.770 (0.038)	0.927 (0.009)	0.933 (0.020)	0.891 (0.022)	0.837 (0.022)	0.816 (0.037)	0.946 (0.036)	0.966 (0.031)
Datasets	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
HIEL-ICS	**0.951** (0.017)	**0.763** (0.013)	**0.913** (0.021)	**0.626** (0.062)	**0.508** (0.024)	**0.827** (0.044)	**0.944** (0.041)	**0.979** (0.004)	**0.701** (0.041)	**0.802** (0.031)	**0.999** (0.001)	**0.977** (0.003)
HIEL-WICS	0.947 (0.013)	0.753 (0.021)	0.902 (0.030)	0.583 (0.076)	0.494 (0.027)	0.796 (0.035)	0.932 (0.041)	0.973 (0.003)	0.684 (0.023)	0.787 (0.014)	0.997 (0.001)	0.973 (0.003)

TABLE VI
KAPPA ANALYSIS ON THE EFFECT OF THE ICS PROCESS (WHERE THE HIGHEST DIVERSITY VALUES ARE HIGHLIGHTED IN BOLD)

Datasets	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
HIEL-ICS	**0.432**	**0.329**	**0.780**	**0.958**	**0.958**	**0.372**	**0.945**	**0.926**	**0.841**	**0.831**	**0.936**	**0.930**
HIEL-WICS	0.638	0.493	0.917	0.991	0.983	0.425	0.979	0.962	0.905	0.926	0.975	0.978
Datasets	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
HIEL-ICS	**0.961**	**0.429**	**0.943**	**0.625**	**0.808**	**0.774**	**0.859**	**0.989**	**0.638**	**0.182**	**0.829**	**0.990**
HIEL-WICS	0.992	0.579	0.979	0.687	0.827	0.859	0.936	0.996	0.818	0.287	0.998	0.994

TABLE VII
COMPARISON WITH SINGLE CLASSIFIER APPROACHES (WHERE THE BEST VALUES ARE HIGHLIGHTED IN BOLD)

Datasets	HIEL	KNN	RT	SVM	J48	DS
BreastTissue	**0.672**±0.071	0.662 ± 0.104	0.666 ± 0.087	0.218 ± 0.027	0.661±0.087	0.406±0.028
Climate	**0.916** ± 0.009	0.909 ± 0.017	0.885 ± 0.020	0.915 ± 0.004	0.906 ± 0.021	0.915 ± 0.004
Haberman	**0.749** ± 0.035	0.694 ± 0.041	0.651 ± 0.055	0.727 ± 0.022	0.715 ± 0.034	0.723 ± 0.026
Iris	**0.974** ± 0.028	0.950 ± 0.036	0.943 ± 0.034	0.973 ± 0.023	0.945 ± 0.037	0.667 ± 0.000
Mammographic	0.777 ±0.026	0.778 ±0.027	0.742 ±0.028	0.791 ± 0.026	**0.802** ± 0.023	0.776 ± 0.023
Ozone_eighthr	0.931 ±0.005	0.895 ± 0.115	0.913 ±0.012	**0.937**± 0.000	0.918 ± 0.010	**0.937** ± 0.000
Seeds	**0.947** ± 0.031	0.928 ± 0.037	0.907 ± 0.040	0.900 ± 0.045	0.910 ± 0.041	0.652 ± 0.014
User_knowledge	**0.912** ± 0.026	0.818 ± 0.036	0.867±0.042	0.809 ± 0.026	0.888 ± 0.031	0.569 ± 0.014
Vehicle_sihouettes	**0.848** ± 0.024	0.693 ±0.027	0.695 ± 0.029	0.298 ± 0.024	0.720 ± 0.029	0.400 ± 0.012
Vertebral_column	**0.829** ±0.049	0.773 ± 0.047	0.812 ± 0.045	0.677 ± 0.000	0.816 ± 0.038	0.772 ± 0.046
WDBC	0.957 ± 0.019	**0.968** ± 0.014	0.932 ± 0.027	0.627 ± 0.004	0.935 ± 0.026	0.892 ± 0.022
Wine	**0.982** ± 0.026	0.955 ± 0.030	0.916 ± 0.046	0.447 ± 0.038	0.924 ± 0.047	0.579 ± 0.048
Wisconsin	0.951 ± 0.017	**0.963** ± 0.015	0.945 ± 0.021	0.959 ± 0.018	0.946 ± 0.019	0.922 ± 0.025
WPBC	**0.763** ± 0.013	0.738 ± 0.047	0.679 ± 0.064	**0.763** ± 0.011	0.725 ± 0.056	**0.763** ± 0.011
Balance	**0.913** ±0.021	0.852 ± 0.020	0.776 ± 0.029	0.896 ± 0.011	0.777 ± 0.025	0.596 ± 0.024
Bupa	0.626 ± 0.062	0.612 ± 0.045	0.624 ± 0.055	0.592 ± 0.014	**0.655** ± 0.051	0.592 ± 0.054
Contraceptive	0.508 ± 0.024	0.457 ± 0.018	0.477 ± 0.025	**0.527** ± 0.024	0.501 ± 0.298	0.427 ± 0.001
Heart	**0.827** ± 0.044	0.778 ± 0.047	0.730 ± 0.051	0.556 ± 0.000	0.774 ± 0.053	0.716 ± 0.052
Housevotes	0.944 ± 0.041	0.924 ± 0.037	0.933 ± 0.040	0.968 ± 0.026	0.966 ± 0.026	**0.970** ± 0.025
Ring	**0.979** ± 0.004	0.747 ± 0.007	0.885 ± 0.009	0.505 ± 0.000	0.904 ± 0.009	0.601 ± 0.008
Saheart	**0.701** ± 0.041	0.679 ± 0.034	0.624 ± 0.047	0.654 ± 0.002	0.692 ± 0.046	0.658 ± 0.022
Spectfheart	**0.802** ± 0.031	0.699 ± 0.051	0.744 ± 0.053	0.794 ± 0.002	0.749 ± 0.057	0.794 ± 0.002
Texture	**0.999** ± 0.001	0.990 ± 0.003	0.913 ± 0.011	0.960 ± 0.007	0.929 ± 0.008	0.182 ± 0.000
Twonorm	**0.977** ± 0.003	0.947 ± 0.005	0.849 ± 0.008	**0.977** ± 0.003	0.847 ± 0.009	0.668 ± 0.009

with Reduce-Error Pruning (CE-REP, [35]), the classifier ensemble method with Complementarity-Measure Pruning (CE-CMP, [35]), the classifier ensemble approach with Margin-Distance Minimization Pruning (CE-MDMP, [35]), and the classifier ensemble approach with Statistical Instance-Based Ensemble Pruning (CE-SIEP, [36]).

Figs. 1 and 2 in the supplementary file compare the accuracy values obtained by using different classifier ensemble approaches on all the datasets in Table I. It is observed that HIEL outperforms its competitors on 14 out of 24 datasets. As an example, HIEL's mean accuracy on the vehicle_silhouettes dataset is 0.848, which defeats the second best one by 0.073. The possible reasons are as follows: 1) HIEL adopts the ICS process to remove the redundant ensemble members, which contribute to higher performance and 2) LDA is used to perform dimension reduction and remove the noisy attributes. The rotation forest approach also achieves good performance on the climate, the seeds, and the wpbc dataset. Their could be some explanations, such as a better representation is found which allows more effective discrimination of data points from different classes.

We have also compared the performance of different classifier ensemble approaches on datasets including the BreastTissue, the Climate and the Haberman with respect to the training time and the testing time. Tables VIII and IX show the running time of those classification approaches.

TABLE VIII
COMPARISON AMONG DIFFERENT CLASSIFIER ENSEMBLE APPROACHES IN TERMS OF THE TRAINING TIME (UNIT: SECOND)

Dataset	HIEL	Random Forest	Adaboost	Bagging	Rotation Forest
BreastTissue	182.20	108.03	6.97	8.97	12.04
Climate	47.16	136.42	8.32	10.06	11.17
Haberman	24.32	120.81	7.45	9.28	10.07

TABLE IX
COMPARISON AMONG DIFFERENT CLASSIFIER ENSEMBLE APPROACHES IN TERMS OF THE TESTING TIME (UNIT: SECOND)

Dataset	HIEL	Random Forest	Adaboost	Bagging	Rotation Forest
BreastTissue	2.67	2.27	0.90	1.38	1.78
Climate	0.16	2.92	0.96	1.32	1.49
Haberman	0.54	2.62	0.91	1.34	1.39

The training time of HIEL is comparable with the training time of the Ra-Forest approach, but not as good as those of other approaches. The testing time of HIEL is comparable with those of other approaches. Since the training process is a one-time effort, HIEL is a good choice when considering its effectiveness.

G. Nonparametric Tests

In order to identify the significant difference among the results of different classifier ensemble algorithms in Figs. 1 and 2 in the supplementary file, we adopt a set of nonparametric tests [51], [52] to compare multiple classifier ensemble approaches, which include the HIEL approach, the Ra-Forest approach, the AdaboostM1 approach, the Bagging approach, the Ro-Forest approach, the classification ensemble approach based on Reduce-Error Pruning (CE-REP), the classification ensemble approach based on Complementarity-Measure Pruning (CE-CMP), the classification ensemble approach based on Margin-Distance Minimization Pruning (CE-MDMP), and the classification ensemble approach based on Statistical Instance-Based Ensemble Pruning (CE-SIEP) over multiple datasets. The Friedman test [51], [52] show that the results obtained by HIEL are significantly better than the compared algorithms.

We need to evaluate the significance of the experiment results of the algorithms on the accuracy. Let $V = \{v_{ijh}\}$ $(i = 1, \ldots, I, j = 1, \ldots, J, h = 1, \ldots, H)$ be the accuracy values, where i stands for the algorithms, j stands for the different datasets, and h stands for different runs, or repetitions. The Friedman test obtains the average accuracy $\overline{v}_{ij} = (1/H)\sum_{h=1}^{H} v_{ijh}$ of those algorithms on those dataset. Then ranks are assigned and the rank matrix $\{R_{ij}\}_{I\times J}$ is obtained, where R_{ij} is the rank of the ith algorithm on the jth dataset. Then, a statistic Ω is obtained as follows:

$$\Omega = \frac{\Psi_1}{\Psi_2} \tag{35}$$

$$\Psi_1 = J\sum_{i=1}^{I}\left(\overline{R}_{i.} - \overline{R}\right)^2 \tag{36}$$

$$\Psi_2 = \frac{1}{(I-1)J}\sum_{i=1}^{I}\sum_{i=1}^{J}\left(R_{ij} - \overline{R}\right)^2 \tag{37}$$

TABLE X
SUMMARY OF THE ADDITIONAL REAL-WORLD DATASETS USED

Dataset	Source	n	m	k
Glass	[46]	214	10	6
Ionosphere	[46]	351	34	2
Sonar	[46]	208	60	2

TABLE XI
COMPARISON OF THE PROPOSED APPROACH HIEL AND CER WHICH SEARCHES FOR THE OPTIMAL COMBINATION OF CLASSIFIERS

Dataset	HIEL	CER
Glass	0.7727	0.7446
Ionosphere	0.9143	0.9130
Sonar	0.7600	0.7531

$$\overline{R}_{i.} = \frac{1}{J}\sum_{j=1}^{J} R_{ij} \tag{38}$$

$$\overline{R} = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{i=1}^{J} R_{ij}. \tag{39}$$

The value of Ω approaches a chi-square χ^2 distribution when I and J approaches infinity. The p-value is calculated by

$$p = P\left(\chi^2_{I-1} \geq \Omega\right). \tag{40}$$

If I or J is small, the approximation will not be accurate. Calculating the p-value from a table of Ω values are preferred.

Tables I–III in the supplementary file show the results of multiple comparisons of the classifier ensemble approaches using different nonparametric statistical procedures, which include the Bonferroni–Dunn test, the Holm test, the Hochberg test, and the Hommel test. Table I in the supplementary file is the average ranking of the classification ensemble approaches. HIEL is has higher ranking compared to other classification ensemble approaches. In general, the results in Tables I–III in the supplementary file indicate the improvement of HIEL over other classifier ensemble approaches.

We have also compared HIEL with the more recently proposed classifier ensemble reduction (CER) approach in [34] which searches for the optimal combination of classifiers on the datasets in Table X. Table XI shows the results obtained by HIEL and CER. HIEL obtains a better result on the Glass dataset, and comparable results on the Ionosphere and Sonar datasets.

VII. Conclusion

In this paper, we propose a new classifier ensemble approach, which is referred to as the HIEL approach, for noisy data classification. When compared with conventional classifier ensemble approaches, HIEL is characterized with the following properties: 1) HIEL explores the feature space and the sample space simultaneously and 2) an incremental selection method based on a classifier-specific criterion function and an ensemble cost function is used to select a subset of classifiers, and different weights are assigned to the individual classifiers in the ensemble. We conduct experiments on 24 noisy real datasets from the UCI machine learning repository and the KEEL repository, and draw several conclusions: 1) LDA largely improves the performance of HIEL and 2) HIEL outperforms conventional single classifier approaches and classification ensemble algorithms on most of the datasets in the experiments. In the future, we shall consider applying the incremental selection process to a broader range of classifier ensemble approaches.

References

[1] Z. Yu *et al.*, "Progressive semisupervised learning of multiple classifiers," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2017.2651114.

[2] R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, and T. I. Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognit.*, vol. 48, no. 5, pp. 1925–1935, 2015.

[3] Z. Yu *et al.*, "Hybrid *k*-nearest neighbor classifier," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1263–1275, Jun. 2016.

[4] N. Garcia-Pedrajas, "Constructing ensembles of classifiers by means of weighted instance selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 258–277, Feb. 2009.

[5] C. Tekin, J. Yoon, and M. van der Schaar, "Adaptive ensemble learning with confidence bounds," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 888–903, Feb. 2017.

[6] R. Diao, F. Chao, T. Peng, N. Snooke, and Q. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1259–1268, Aug. 2014.

[7] P.-B. Zhang and Z.-X. Yang, "A novel adaboost framework with robust threshold and structural optimization," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2623900.

[8] Z. Yu, L. Li, J. Liu, and G. Han, "Hybrid adaptive classifier ensemble," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 177–190, Feb. 2015.

[9] Z. Yu *et al.*, "A new kind of nonparametric test for statistical comparison of multiple classifiers over multiple datasets," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4418–4431, Dec. 2017, doi: 10.1109/TCYB.2016.2611020.

[10] Y. Mu, W. Ding, and D. Tao, "Local discriminative distance metrics ensemble learning," *Pattern Recognit.*, vol. 46, no. 8, pp. 2337–2349, 2013.

[11] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, May 2015.

[12] J. H. Abawajy, A. Kelarev, and M. Chowdhury, "Large iterative multitier ensemble classifiers for security of big data," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 352–363, Sep. 2014.

[13] S. Bashbaghi, E. Granger, R. Sabourin, and G.-A. Bilodeau, "Dynamic ensembles of exemplar-SVMs for still-to-video face recognition," *Pattern Recognit.*, vol. 69, pp. 61–81, Sep. 2017.

[14] M. Osadchy, D. Keren, and D. Raviv, "Recognition using hybrid classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 759–771, Apr. 2016.

[15] J.-Q. Li, Z.-H. You, X. Li, M. Zhong, and X. Chen, "PSPEL: In silico prediction of self-interacting proteins from amino acids sequences using ensemble learning," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 14, no. 5, pp. 1165–1172, Sep./Oct. 2017, doi: 10.1109/TCBB.2017.2649529.

[16] J. Hu *et al.*, "Predicting protein-DNA binding residues by weightedly combining sequence-based features and boosting multiple SVMs," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, to be published, doi: 10.1109/TCBB.2016.2616469.

[17] Y. Guan, C.-T. Li, and F. Roli, "On reducing the effect of covariate factors in gait recognition: A classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1521–1528, Jul. 2015, doi: 10.1109/TPAMI.2014.2366766.

[18] Y. Su, S. Shan, X. Chen, and W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1885–1896, Aug. 2009.

[19] A. Rahman and B. Verma, "Novel layered clustering-based approach for generating ensemble of classifiers," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 781–792, May 2011.

[20] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017, doi: 10.1109/TNNLS.2016.2582798.

[21] M. F. Amasyali and O. K. Ersoy, "Classifier ensembles with the extended space forest," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 549–562, Mar. 2014.

[22] L. Nanni and A. Lumini, "Evolved feature weighting for random subspace classifier," *IEEE Trans. Neural Netw.*, vol. 19, no. 2, pp. 363–366, Feb. 2008.

[23] S. Mao *et al.*, "Weighted classifier ensemble based on quadratic form," *Pattern Recognit.*, vol. 48, no. 5, pp. 1688–1706, 2015.

[24] M. D. Muhlbaier, A. Topalis, and R. Polikar, "*Learn*$^{++}$.NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 152–168, Jan. 2009.

[25] L. I. Kuncheva, "A bound on Kappa-error diagrams for analysis of classifier ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 494–501, Mar. 2013.

[26] O. Ludwig, U. Nunes, B. Ribeiro, and C. Premebida, "Improving the generalization capacity of cascade classifiers," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2135–2146, Dec. 2013.

[27] S. Rasheed, D. W. Stashuk, and M. S. Kamel, "Integrating heterogeneous classifier ensembles for EMG signal decomposition based on classifier agreement," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 866–882, May 2010.

[28] L. Oliveira, U. Nunes, and P. Peixoto, "On exploration of classifier ensemble synergism in pedestrian detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 16–27, Mar. 2010.

[29] L. Nanni, A. Lumini, D. Gupta, and A. Garg, "Identifying bacterial virulent proteins by fusing a set of classifiers based on variants of Chou's pseudo amino acid composition and on evolutionary information," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 9, no. 2, pp. 467–475, Mar./Apr. 2012.

[30] A. Dehzangi, K. Paliwal, A. Sharma, O. Dehzangi, and A. Sattar, "A combination of feature extraction methods with an ensemble of different classifiers for protein structural class prediction problem," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 10, no. 3, pp. 564–575, May 2013.

[31] W. Hu, W. Hu, and S. Maybank, "AdaBoost-based algorithm for network intrusion detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 577–583, Apr. 2008.

[32] J. Marín, D. Vázquez, A. M. López, J. Amores, and L. I. Kuncheva, "Occlusion handling via random subspace classifiers for human detection," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 342–354, Mar. 2014.

[33] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "A comparison of decision tree ensemble creation techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 173–180, Jan. 2007.

[34] V. Soto, S. García-Moratilla, G. Martínez-Muñoz, D. Hernandez-Lobato, and A. Suárez, "A double pruning scheme for boosting ensembles," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2682–2695, Dec. 2014.

[35] Y. Qian, H. Xu, J. Liang, B. Liu, and J. Wang, "Fusing monotonic decision trees," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2717–2728, Oct. 2015.

[36] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 245–259, Feb. 2009.

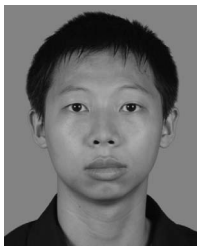
[37] A. H. R. Ko, R. Sabourin, and A. S. Britto, Jr., "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognit.*, vol. 41, no. 5, pp. 1718–1731, 2008.

[38] T. Woloszynski and M. Kurzynski, "A probabilistic model of classifier competence for dynamic ensemble selection," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2656–2668, 2011.
[39] Y. Zhang and W. N. Street, "Bagging with adaptive costs," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 577–588, May 2008.
[40] M. M. Islam, X. Yao, S. M. S. Nirjon, M. A. Islam, and K. Murase, "Bagging and boosting negatively correlated neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 3, pp. 771–784, Jun. 2008.
[41] B. Wang and J. Pineau, "Online bagging and boosting for imbalanced data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3353–3366, Dec. 2016.
[42] G. Martínez-Muñoz and A. Suárez, "Out-of-bag estimation of the optimal sample size in bagging," *Pattern Recognit.*, vol. 43, no. 1, pp. 143–152, 2010.
[43] X. Zhu and Y. Yang, "A lazy bagging approach to classification," *Pattern Recognit.*, vol. 41, no. 10, pp. 2980–2992, 2008.
[44] G. Meng, C. Pan, N. Zheng, and C. Sun, "Skew estimation of document images using bagging," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1837–1846, Jul. 2010.
[45] A. Asuncion and D. J. Newman, *UCI Machine Learning Repository*, School Inf. Comput. Sci., Univ. California at Irvine, Irvine, CA, USA, 2007. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html
[46] J. Alcalá-Fdez *et al.*, "KEEL data-mining software tool: Dataset repository, integration of algorithms and experimental analysis framework," *J. Multiple Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.
[47] M. Hall *et al.*, "The WEKA data mining software: An update," *SIGKDD Explor.*, vol. 11, no. 1, pp. 10–18, 2009.
[48] J. Yu, D. Tao, M. Wang, and Y. Rui, "Learning to rank using user clicks and visual features for image retrieval," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 767–779, Apr. 2015.
[49] J. Yu, Y. Rui, and D. Tao, "Click prediction for Web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2019–2032, May 2014.
[50] J. Yu, Y. Rui, Y. Y. Tang, and D. Tao, "High-order distance-based multiview stochastic learning in image classification," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2431–2442, Dec. 2014.
[51] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937.
[52] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Stat.*, vol. 11, no. 1, pp. 86–92, 1940.
[53] G. E. P. Box, "A general distribution theory for a class of likelihood criteria," *Biometrika*, vol. 36, no. 3, pp. 317–346, 1949.
[54] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[55] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *Proc. 14th Conf. Mach. Learn.*, 1997, pp. 211–218.
[56] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.
[57] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, Sydney, NSW, Australia, 2015, pp. 715–724.
[58] T. Liu, D. Tao, and D. Xu, "Dimensionality-dependent generalization bounds for k-dimensional coding schemes," *Neural Comput.*, vol. 28, no. 10, pp. 2213–2249, Oct. 2016.
[59] Z. Yu *et al.*, "Adaptive semi-supervised classifier ensemble for high dimensional data classification," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2017.2761908.
[60] Z. Yu *et al.*, "Progressive subspace ensemble learning," *Pattern Recognit.*, vol. 60, pp. 692–705, Dec. 2016.
[61] Z. Yu, H.-S. Wong, J. You, G. Yu, and G. Han, "Hybrid cluster ensemble framework based on the random combination of data transformation operators," *Pattern Recognit.*, vol. 45, no. 5, pp. 1826–1837, 2012.
[62] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, and J. You, "Semi-supervised classification based on random subspace dimensionality reduction," *Pattern Recognit.*, vol. 45, no. 3, pp. 1119–1135, 2012.
[63] Z. Yu *et al.*, "Adaptive ensembling of semi-supervised clustering solutions," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1577–1590, Aug. 2017.
[64] Z. Yu *et al.*, "Probabilistic cluster structure ensemble," *Inf. Sci.*, vol. 267, pp. 16–34, May 2014.
[65] Z. Yu *et al.*, "Double selection based semi-supervised clustering ensemble for tumor clustering from gene expression profiles," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 11, no. 4, pp. 727–740, Jul./Aug. 2014.
[66] Z. Yu, H. Chen, J. You, G. Han, and L. Li, "Hybrid fuzzy cluster ensemble framework for tumor clustering from biomolecular data," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 10, no. 3, pp. 657–670, May 2013.
[67] Z. Yu *et al.*, "Distribution-based cluster structure selection," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3554–3567, Nov. 2017.
[68] Z. Yu, L. Li, J. You, H.-S. Wong, and G. Han, "SC3: Triple spectral clustering based consensus clustering framework for class discovery from cancer gene expression profiles," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 9, no. 6, pp. 1751–1765, Nov./Dec. 2012.
[69] W. Bian and D. Tao, "Max–min distance analysis by using sequential SDP relaxation for dimension reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 1037–1050, May 2011.
[70] C. Xu, D. Tao, C. Xu, and Y. Rui, "Large-margin weakly supervised dimensionality reduction," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, 2014, pp. 865–873.
[71] Y. Zhang and Z.-H. Zhou, "Multilabel dimensionality reduction via dependence maximization," in *Proc. 23rd Nat. Conf. Artif. Intell. (AAAI)*, vol. 3. 2008, pp. 1503–1505.
[72] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 12, pp. 2531–2544, Dec. 2015.

Zhiwen Yu (S'06–M'08–SM'14) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2008.

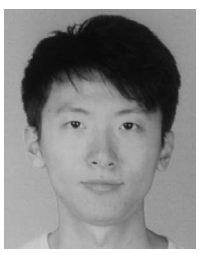
He is a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China, and an Adjunct Professor with Sun Yat-sen University, Guangzhou. His research interests include data mining, machine learning, and pattern recognition. He has been published over 100 referred journal papers and international conference papers, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON MULTIMEDIA, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Dr. Yu is a Council Member of China Computer Federation (CCF), a Distinguished Member of CCF, and a Senior Member of ACM and CAAI.



Daxing Wang received the B.Sc. degree from the South China University of Technology, Guangzhou, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering.

His current research interests include pattern recognition, machine learning, and data mining.



Zhuoxiong Zhao received the B.Sc. and M.Phil. degrees from the South China University of Technology, Guangzhou, China, in 2014 and 2017, respectively.

His current research interests include pattern recognition, machine learning, and data mining.

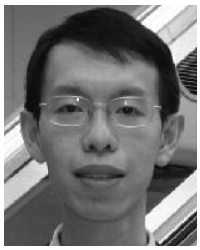
C. L. Philip Chen (S'88–M'88–SM'94–F'07) received the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 1985 and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1988, both in electrical engineering.

He is currently a Chair Professor with the Department of Computer and Information Science and the Dean of the Faculty of Science and Technology, University of Macau, Macau, China. His current research interests include computational intelligence, systems, and cybernetics.

Jane You received the Ph.D. degree from La Trobe University, Melbourne, VIC, Australia, in 1992.

She is currently a Professor with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, and the Chair of Department Research Committee. She has researched extensively in the fields of image processing, medical imaging, computer-aided diagnosis, and pattern recognition. She has over 190 research papers published with over 1000 nonself citations. She has been a Principal Investigator for one ITF project, three GRF projects, and many other joint grants since she joined PolyU in 1998. She is a Team Member for two successful patents (one HK patent and one U.S. patent).

Dr. You was a recipient of three awards including Hong Kong Government Industrial Awards, the Special Prize and Gold Medal with Jury's Commendation at the 39th International Exhibition of Inventions of Geneva in 2011 for her current work on retinal imaging, and the Second Place in an International Competition [SPIE Medical Imaging'2009 Retinopathy Online Challenge in (ROC'2009)]. Her research output on retinal imaging has been successfully led to technology transfer with clinical applications. She is an Associate Editor of *Pattern Recognition* and other journals.



Hau-San Wong received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, and the Ph.D. degree in electrical and information engineering from the University of Sydney, Sydney, NSW, Australia.

He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He held research positions with the University of Sydney and Hong Kong Baptist University, Hong Kong. His current research interests include multimedia information processing, multimodal human–computer interaction, and machine learning.

Jun Zhang (M'02–SM'08–F'16) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

Since 2016, he has been with the South China University of Technology, Guangzhou, China, where he is currently a Cheung Kong Professor. His current research interests include computational intelligence, cloud computing, big data, and wireless sensor networks. He has authored seven research books and book chapters, and over 100 technical papers in the above areas.

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 the Current Chair of the IEEE Guangzhou Subsection and ACM Guangzhou Chapter.