Hybrid k-Nearest Neighbor Classifier

Zhiwen Yu, *Senior Member, IEEE*, Hantao Chen, Jiming Liu, *Fellow, IEEE*,
Jane You, Hareton Leung, and Guoqiang Han

***Abstract*—Conventional *k*-nearest neighbor (KNN) classification approaches have several limitations when dealing with some problems caused by the special datasets, such as the sparse problem, the imbalance problem, and the noise problem. In this paper, we first perform a brief survey on the recent progress of the KNN classification approaches. Then, the hybrid KNN (HBKNN) classification approach, which takes into account the local and global information of the query sample, is designed to address the problems raised from the special datasets. In the following, the random subspace ensemble framework based on HBKNN (RS-HBKNN) classifier is proposed to perform classification on the datasets with noisy attributes in the high-dimensional space. Finally, the nonparametric tests are proposed to be adopted to compare the proposed method with other classification approaches over multiple datasets. The experiments on the real-world datasets from the Knowledge Extraction based on Evolutionary Learning dataset repository demonstrate that RS-HBKNN works well on real datasets, and outperforms most of the state-of-the-art classification approaches.**

***Index Terms*—Classification, ensemble learning, machine learning, nearest neighbor classifier, supervised learning.**

I. Introduction

The k-NEAREST neighbor (KNN) classification methods is one kind of the classical and popular classification approaches [1], [2]. Until now, a lot of researchers pay attention to KNN classification methods due to its useful applications in the areas of pattern recognition (PR) [3]–[7], remote sensing [8], [9], image processing [10], bioinformatics [11], [12], and so on. For example, Shanableh *et al.* [3] adopted the KNN approach for isolated gesture recognition in Arabic sign language. Geng *et al.* [4] combined the KNN approach with the dimensionality reduction technique for pattern classification. Bosch *et al.* [5] adopted KNN classifier for scene classification. Yang *et al.* [6] applied the local mean-based nearest neighbor classifier for discriminant analysis. Xu *et al.* [7] applied the k-local hyperplane distance nearest neighbor classifier for feature extraction and classification. Frigui and Gader [8] adopted a possibilistic KNN classifier to perform land mine detection. Li *et al.* [9] adopted the KNN approach to the hyperspectral image classification. Mensink *et al.* [10] used the KNN method for image classification. Maji [11] used the KNN algorithm for microarray data classification. Raymer *et al.* [12] combined the KNNs classifier with a genetic algorithm for knowledge discovery in medical and biological datasets. In summary, the KNN classification approach has been successfully applied to different areas due to its simpleness and effectiveness.

Manuscript received February 16, 2015; revised April 29, 2015; accepted June 8, 2015. This work was supported in part by the grant from the National Natural Science Foundation of China under Project 61472145, in part by the New Century Excellent Talents in University under Project NCET-11-0165, in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Project S2013050014677, in part by the grant from Hong Kong Baptist University under Project RGC/HKBU211212, in part by Hong Kong General Research under Grant B-Q44D, in part by Hong Kong Polytechnic University under Grant G-YM05 and Grant G-YN39, and in part by the Fundamental Research Funds for the Central Universities under Projects 2014G0007 and 2015PT016. This paper was recommended by Associate Editor D. Tao.

Z. Yu, H. Chen, and G. Han are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: zhwyu@scut.edu.cn).

J. Liu is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: jiming@comp.hkbu.edu.hk).

J. You and H. Leung are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong.

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.2015.2443857

Even though a large number of the KNN approaches are proposed in recent years, few of research works give a brief review on the recent progress of the KNN approaches. In this paper, we will perform a brief survey on the KNN classification methods. Traditional KNN approach is suitable for different kinds of datasets, but its performance will be affected in some special datasets, such as the dataset with the small number of training samples, the dataset with the imbalance size of training classes, and the dataset with noisy samples and noisy attributes. In this paper, we pay attention to how to design the new techniques in the KNN classification framework to tackle the problems caused by the special datasets.

First, we propose the hybrid KNN (HBKNN) classification approach to address the problems raised from the special datasets. As compared with conventional KNN approach, HBKNN are characterized with three properties.

1) HBKNN adopts the fuzzy relative transform technique to tackle the imbalance problem.
2) The fuzzy k-local hyperplane distance nearest neighbor classifier is used by HBKNN to address the sparse problem caused by the small training sample size.
3) HBKNN combined the fuzzy relative transformation-based k-local hyperplane distance nearest neighbor classifier (FRHKNN) with traditional KNN approach to tackle the noise problem.

Conventional KNN approach is a special case of HBKNN. We further design the random subspace ensemble framework based on HBKNN (RS-HBKNN) classifier which is suitable for the datasets with noisy attributes in the high-dimensional space. Finally, the nonparametric tests are adopted to compare HBKNN and other KNN approaches over multiple datasets which are the real-world datasets from the knowledge extraction based on evolutionary learning (KEEL) dataset repository [61]. The experimental results illustrate that RS-HBKNN not only works well on these datasets, but also outperforms most of the state-of-the-art classification approaches.

As a result, the contributions of this paper are as follows.

1) We perform a brief survey on the recent progress of the KNN classification approach.
2) The HBKNN classification approach is designed to tackle the problem caused by the special datasets.
3) The RS-HBKNN classifier is proposed to deal with the high-dimensional dataset with noisy attributes.
4) We propose to adopt the nonparametric tests to compare different KNN approaches over multiple datasets.

The remainder of this paper is organized as follows. Section II gives a brief survey to KNN classification approaches. Section III presents the HBKNN classifier. Section IV introduces the RS-HBKNN classifier. Section V evaluates the performance of HBKNN and RS-HBKNN using real datasets. Section VI draws the conclusion and describes possible future works.

II. Related Works

The KNN classification approach is one of the most classical yet important research topics in data mining and machine learning [88]–[102], which has many advantages for the classification task. For example: 1) the KNN approach is simple, which only need to tune one parameter *k*, and does not make any assumption for the distribution of the training data; 2) it is effective for the dataset with a large number of training data; and 3) KNN has error guarantee, whose asymptotic error rate is at most twice the Bayes error rate. At the same time, it has several limitations related to the number of nearest neighbors (*k*), the similarity measure and the computational cost. As a result, it raises several research issues, such as how to determine the optimal *k* value, how to find the suitable distance metric, and how to reduce the computational cost. There are plenties of the KNN approaches proposed in recent years to address the above issues and improve the performance of traditional KNN approaches. Most of them can be divided into five categories.

The approaches in the first category pay attention to how to design a new KNN approach [13]–[25]. Some researchers consider how to extend the nearest neighbor relationships among data points to the relationships among points to other relationships, such as data points and hyperplane, data points and centers of the classes. For example, Vincent and Bengio [13] proposed the *k*-local hyperplane nearest neighbor algorithms. Li *et al.* [16] proposed a new nearest neighbor classifier based on the local probabilistic centers of each class. Chien and Wu [21] used the nearest feature plane and nearest feature space classifiers for face recognition. Gao and Wang [23] designed the center-based nearest neighbor classifier. Zheng *et al.* [25] proposed two nearest neighbor classifiers for pattern classification, which are named as nearest neighbor line and nearest neighbor plane. Some research works take into account the relationship among the groups instead of the individuals. For example, Veenman and Reinders [14] designed the nearest subclass classifier. Samsudin and Bradley [19] designed the nonparametric group-based classification technique based on the nearest neighbor algorithm. Some prior works studied how to construct the multilevel KNN classifiers. For example, Zhao [15] designed a nearest neighbor classifier tree based on the R4-rule. Gao *et al.* [17] designed a two-level KNN classifier based on an adaptive distance metric. Xu *et al.* [18] designed a coarse to fine KNN classifier. Some researchers focus on how to design a new similarity measure. For example, Zhang and Yang [20] designed a nearest neighbor classification framework based on the linear reconstruction measure. In addition, Bermejo and Cabestany [22] designed a soft KNN classifier. Wen *et al.* [24] designed a new KNN classification approach based on the perceptual relativity.

The approaches in the second category analyze the KNN approach theoretically [26]–[42]. References [1] and [26] have investigated the error bounds of the KNN approach. For example, Cover and Hart [1] analyzed the nearest neighbor classifier theoritically according to the Bayes probability of error. Bax [26] explored the probably approximately correct error bounds for KNN classifiers. Some researchers analyzed the factors which are affected the performance of the KNN approach, such as the shape of the range of KNN, the optimal *k* value, the sample size, the margin of the classifier, and so on [27]–[33]. For example, Cevikalp [27] studied the bounding hyperdisk of each training class for the KNN approach. Ghosh *et al.* [28] considered how to determine the optimal values of *k* in KNN. Parthasarathy and Chatterji [29] explored how to use KNN methods when the sample size is small. Domeniconi *et al.* [30] studied the large margin nearest neighbor classifiers theoritically. Majumdar and Ward [31] studied the combination of the nearest neighbor classifier with the random projection technique. Hu *et al.* [32] investigated the how to enlarge margin of the nearest neighbor classifier by sample weight learning. Sáez *et al.* [33] studied the effect of noise filters for the performance of the nearest neighbor classifier. Some research works investigated the distance metrics of the KNN approach [34], [35]. Argentini and Blanzieri [34] considered the similarity measure for the KNN algorithm, and proposed the neighborhood counting measure. Weinberger and Saul [35] explored how to obtain large margin for KNN classification by distance metric learning. Some researchers introduced other theories to improve the performance of the KNN approach [36], [37]. Derrac *et al.* [36] adopted the cooperative coevolution to improve the performance of the nearest neighbor classifier. Triguero *et al.* [37] addressed the limitations of the nearest neighbor classifier, and adopted the differential evolution to optimize the positioning of prototypes. Researchers have also explored the efficiency property of the KNN approach [38]–[40]. For example, Zhang and Srihari [38] studied the fast KNN algorithms according to the cluster-based trees. Hernández-Rodríguez *et al.* [39] explored the efficiency of KNN classifier, and proposed n approximate fast *k* most similar neighbor classifier based on a tree structure.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Viswanath *et al.* [40] explored the efficiency of the nearest neighbor classifier and designed an efficient nearest neighbor classifier to reduce its computational burden. In the mean time, Lucey and Ashraf [41] studied the generalization property of nearest neighbor classifier by spatially constrained filters, while Ghosh *et al.* [42] studied the visualization and aggregation of nearest neighbor classifiers.

The approaches in the third category studied how to design KNN-based ensemble framework [43]–[45]. For example, Zhou and Yu [43] designed a nearest neighbor classifier-based ensemble framework. Nanni and Lumini [44] proposed the k-local hyperplane-based ensemble framework. Altnay [45] designed a multimodal perturbation-based KNN classifier ensemble. Domeniconi and Yan [81], [82] studied the nearest neighbor ensemble approach and its relationships related to error correlation and accuracy.

The approaches in the fourth category focus on the applications of the KNN approaches [46]–[57]. For example, in the area of medical image, Acha *et al.* [46] used KNN classifier to classify psychophysical experiment image data. In the area of remote sensing, Blanzieri and Melgani [47] applied the KNN classifier to remote sensing image datasets. Yang *et al.* [48] applied the KNN classification for hyperspectral image data. In the area of PR, Munder and Gavrila [49] used KNN classifier for pedestrian classification. Sussman *et al.* [50] applied the KNNs classification rule to vertex classification on the random dot product graphs. Jafari-Khouzani and Soltanian-Zadeh [51] used the KNN classifier for texture analysis. Gutierrez-Osuna and Nagle [52] applied the KNN classifier for odor classification using an array of gas sensors. Mezghani *et al.* [53] applied the KNN classifier for automatic classification of asymptomatic and osteoarthritis knee gait patterns. Shah and Sastry [54] applied the nearest neighbor classifier for fingerprint classification. Lu *et al.* [55] used the nearest neighbor classifier for tensor object recognition. Singh *et al.* [56] applied KNN classifier to natural scene analysis. In the area of biomedical engineering, Preece *et al.* [57] used the nearest neighbor classifier for the classification of dynamic activities from accelerometer data.

The approaches in the fifth category performed a survey for the KNN classifier [58]–[60]. For example, Liu and Nakagawa [58] performed a survey on nearest neighbor classifier and its application to handwritten character recognition. Cunningham and Delany [59] gave a review on KNN classifier. Jiang *et al.* [60] performed a survey of improved KNN classifiers. Bhatia and Vandana [83] analyzed the key ideas, advantages, disadvantages and target data of 17 nearest neighbor methods.

In general, we summarize some possible future directions on KNN classifiers. First, a more detailed survey for the KNN approaches is required because it is lack of the published surveys or reviews for the KNN classifiers in the recent five years. The possible reason is that the KNN approach is the mature and popular technique which is not worth to perform a survey. In fact, due to many applications of the KNN classifiers, it is an urgent requirement to perform a detailed review which is able to spread the recent progress of the KNN approaches to other application areas.

Second, designing a new HBKNN classification approach will be an interesting research topic. There are different kinds of the KNN approaches. And each of them possesses its property, which is useful for tackle one kind of the problems caused by the special datasets. It raises the requirement of the HBKNN classification approaches which take advantage of different kinds of the KNN approaches, have several interesting properties and are able to deal with several kinds of the problems caused by the special datasets, such as the sparse problem, the imbalance problem, and the noise problem.

Third, the distance metric learning-based ensemble framework is still a research topic which is worth to explore. Most of traditional distance metric learning algorithms for the KNN approaches require to construct a distance matrix, and take $O(n^2)$ computational costs, which are more suitable for the medium datasets, not for the large datasets. If the sample size or the attribute size of the dataset becomes very larger, the performance of these algorithms will degrade. One possible solutions is constructing a distance metric learning-based ensemble framework. If the sample size of the dataset is very large, the bagging technique is first adopted to obtain a set of the medium datasets from the large dataset. Then, the distance metric learning algorithm is used to learn the distance matrix. Finally, the voting scheme is applied to generate the final result. If the attribute size of the dataset is large, the random subspace technique is used to generate a set of subspaces in the low-dimensional space. As a whole, an distance metric learning-based ensemble framework will be useful to improve the performance of the KNN classification approaches.

Fourth, improving the efficiency of the KNN approaches in the very large dataset is an important research topic in the future. The tree structure is one possible solution to accelerate the process of finding KNN. Another possible solutions is the multilevel KNN approaches. Different levels of KNN classifiers will correspond to different resolutions of the dataset.

Fifth, how to apply the new KNN classification approaches published in recent years to different areas, such as PR, remote sensing, medical images, bioinformatics, and so on, is still an important research topic in the future. When compared with other classifiers, the KNN classifier is simple, and is easy to understand by the researchers in other areas. The applications of the KNN approaches in different kinds of the datasets will raise a lot of interesting problems which need to be tackled. It is worth to popularize the new KNN classifier in different research areas.

This paper belongs to the first, third, and fifth categories. A brief review is first given for the recent progress of the KNN approaches, which belongs to the fifth category. Then, the HBKNN classification approach is proposed, which belongs to the third category. Finally, the RS-HBKNN classifier is designed, which is included in the third category.

III. Hybrid k-Nearest Neighbor Classifier

Fig. 1 shows an overview of the FRHKNN classifier. FRHKNN first generates the balance training set to tackle the unbalance problem. Then, the relative transformation is

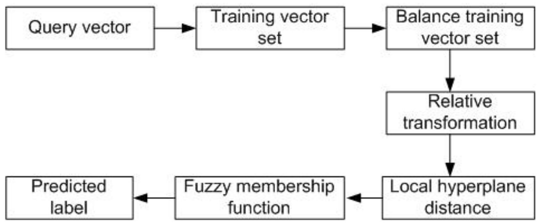


Fig. 1. Overview of the FRHKNN classifier.

adopted by FRHKNN to avoid the effect of noisy data. In the following, FRHKNN constructs the local hyperplane and calculates the local hyperplane distance for each class. Finally, the fuzzy membership function is used to summarize the local hyperplane distance and predicts the label of the query vector.

Specifically, assume that: 1) v is a query feature vector; 2) $F = \{f_1, f_2, \ldots, f_n\}$ is the set of training feature vectors (where n is the number of training feature vectors), and $Y = \{y_1, y_2, \ldots, y_n\}$ is the corresponding set of the labels with respect to F (where $y_i \in \{1, 2, \ldots, k\}$, $i \in \{1, \ldots, n\}$); 3) each class is denoted as C_j ($j \in \{1, \ldots, k\}$), and the number of feature vectors in each class C_j is represented by n_j; 4) ε is the size of the neighborhood in the classification process (where ε is an integer); and 5) β is the penalty parameter for classification.

In the first step, FRHKNN first constructs a local environment for the query feature vector v, which is defined as ε nearest neighbors of the query feature vector v in each class C_j based on the Euclidean distance, and generates a new balance training set F' as follows:

$$F' = \cup_j \, F_j(\varepsilon, v) \tag{1}$$

$$F_j(\varepsilon, v) = \left\{ f_i \in C_j \middle| \phi(f_i, v) \le \phi_j^{\varepsilon} \right\} \tag{2}$$

where ϕ denotes the Euclidean distance function, and ϕ_j^{ε} denotes the Euclidean distance between the query vector v and the ε nearest neighbor in the class C_j. The local environment enable FRHKNN to avoid the effect of the unbalanced problem. And the balance training set treats all the classes equally, which is able to avoid the class boundary biased in favor of the class with more feature vectors.

FRHKNN adopts the relative transformation [24] to construct the relative space in the second step. Given the query feature vector v and a set of feature vectors in F', it first includes v into the training set F' as follows:

$$F'' = F' \cup \{v\}. \tag{3}$$

Then, each feature vector f in the new training set F'' is considered one by one. FRHKNN calculates the distances between f and all the feature vectors in F'', and obtains a set of distance values, which is as follows:

$$\left\{ \phi(f_1, f), \ldots, \phi\left(f_{n'-1}, f\right), \phi(v, f) \right\} \tag{4}$$

where $n' = \varepsilon k + 1$, which is the cardinality of the new training set F''. The set of distance values can be viewed as the new values of the feature vector f in the relative space. The advantage of the relative transformation is to reduce the effect of noisy data. The feature vectors which are close to each other will be closer in the relative space, while the feature vectors which are far from each other will be farther in the relative space. As a result, the training set F'' contains a set of feature vectors and the query vector with new attributes.

In the third step, FRHKNN computes the k-local hyperplane distances for all the classes. It first constructs a local hyperplane for each class C'_j with ε feature vectors as follows:

$$H_j^{\varepsilon}(v) = \left\{ h \middle| h = \bar{f} + \sum_{t=1}^{\varepsilon} \alpha_t \left(f_t - \bar{f}\right), \alpha_t \in \Re \right\} \tag{5}$$

$$\bar{f} = \frac{1}{\varepsilon} \sum_{t=1}^{\varepsilon} f_t. \tag{6}$$

Then, FRHKNN computes the local hyperplane distance $\phi(H_j^{\varepsilon}(v), v)$ with respect to the query feature vector v and the jth local hyperplane $H_j^{\varepsilon}(v)$ using the following formula:

$$\phi\left(H_j^{\varepsilon}(v), v\right) = \min_{\alpha_t} \left\| v - \bar{f} - \sum_{t=1}^{\varepsilon} \alpha_t \left(f_t - \bar{f}\right) \right\| \tag{7}$$

where α_t can be obtained by solving a linear system as follows:

$$\left(U \cdot U'\right) \cdot A = U' \cdot \left(v - \bar{f}\right) \tag{8}$$

where $A = (\alpha_1, \alpha_2, \ldots, \alpha_{\varepsilon})'$, and U is an $n \times \varepsilon$ matrix composed of column vector $U_t = f_t - \bar{f}$. A penalty item $\beta \sum_{t=1}^{\varepsilon} \alpha_t^2$ (where β is a parameter) is introduced to penalize the large values of α_t. The above formula can be reformulated as follows:

$$\phi\left(H_j^{\varepsilon}(v), v\right) = \min_{\alpha_t} \left\{ \left\| v - \bar{f} - \sum_{t=1}^{\varepsilon} \alpha_t \left(f_t - \bar{f}\right) \right\| + \beta \sum_{t=1}^{\varepsilon} \alpha_t^2. \right. \tag{9}$$

FRHKNN adopts the fuzzy membership function to summarize the local hyperplane distances and assigns a fuzzy value m_j for each class C_j in the fourth step. The fuzzy value m_j is calculated as follows:

$$m_j = \frac{1}{\sum_{l=1}^{k} \left(\frac{\phi\left(H_j^{\varepsilon}(v), v\right)}{\phi\left(H_l^{\varepsilon}(v), v\right)} \right)^{\frac{2}{q-1}}} \tag{10}$$

where $2/(q-1)$ is the fuzziness exponent, q is set to 2, and $\phi(H_j^{\varepsilon}(v), v)$ denotes the Euclidean distance between the local hyperplane $H_j^{\varepsilon}(v)$ and the query vector v. The query feature vector v will be assigned to the class associated with the maximum fuzzy membership values.

We further propose the HBKNN classifier, which is a generalized version of FRHKNN and traditional fuzzy KNN. When compared with FRHKNN, HBKNN not only takes into account the global information of the query vector, but also considers the local information of the query vector.

Given a query feature vector v, HBKNN adopts FRHKNN to generate a probability value p_j^1 for each class C_j according to the fuzzy membership value obtained by the local information of the query vector in the first step. p_j^1 is calculated as follows:

$$p_j^1 = \frac{m_j}{\sum_{l=1}^{k} m_l}. \tag{11}$$

In the second step, it uses traditional fuzzy KNN to directly obtain the neighbor set N which consists of ι nearest neighbors of v in the training set, and obtains the fuzzy membership values of ι nearest neighbors globally.

Assume that: 1) $f_1, f_2, \ldots, f_\iota$ are the set of feature vectors in the neighbor set N; 2) N_j denotes the subset of N in the class C_j; and 3) $|N_j|$ is the cardinality of N_j, the fuzzy membership value $m_i (i \in \{1, \ldots, \iota\})$ for the feature vector f_i is calculated as follows:

$$m_i = \frac{1}{\sum_{l=1}^{\iota} \left(\frac{\phi(f_i, v)}{\phi(f_l, v)} \right)^{\frac{2}{q-1}}} \tag{12}$$

where $\phi(f_i, v)$ represents the Euclidean distance between the feature vector f_i and the query vector v, respectively.

The total fuzzy membership value M is calculated as follows:

$$M = \sum_{i=1}^{\iota} m_i. \tag{13}$$

The sum of the fuzzy membership values M_j for each class C_j is computed as follows:

$$M_j = \sum_{f_i \in N_j} m_i. \tag{14}$$

The probability p_j^2 that the query feature vector v belongs to the class C_j is calculated as follows:

$$p_j^2 = \frac{M_j}{M}. \tag{15}$$

HBKNN uses a confidence factor λ_j to summarize the probability values p_j^1 and p_j^2 of each class C_j as follows:

$$\lambda_j = \omega_1 p_j^1 + \omega_2 p_j^2. \tag{16}$$

If ω_1 and ω_2 are set to 1 and 0, respectively, HBKNN will be FRHKNN. If ω_1 and ω_2 are set to 0 and 1, respectively, HBKNN will change to traditional KNN. Suitable values for ω_1 and ω_2 will keep a balance between FRHKNN and traditional KNN. The label y_v of the query feature vector v is predicted by HBKNN according to the values of the confidence factors as follows:

$$y_v = \arg \max_{j \in \{1, \ldots, k\}} \lambda_j. \tag{17}$$

The time complexity T_{HBKNN} of HBKNN consists of two parts: 1) T_{FRHKNN} and 2) T_{FKNN}. T_{FRHKNN} is the computational cost for the FRHKNN algorithm and T_{FKNN} is the computational cost for the fuzzy KNN algorithm. T_{FRHKNN} is related to the computational cost of generating the balance training set $O(n \log n)$, performing the relative transformation $O(n'^2)$, calculating the local hyperplane distance $O(k^2)$, and computing the fuzzy member values $O(k^2)$ (where n is the number of feature vectors and k is the number of classes). Since $k << n$ and $n' << n$, the time complexity T_{FRHKNN} is $O(n \log n)$. The time complexity T_{FKNN} is $O(n \log n)$. As a result, the complexity T_{HBKNN} of HBKNN is $O(n \log n)$.

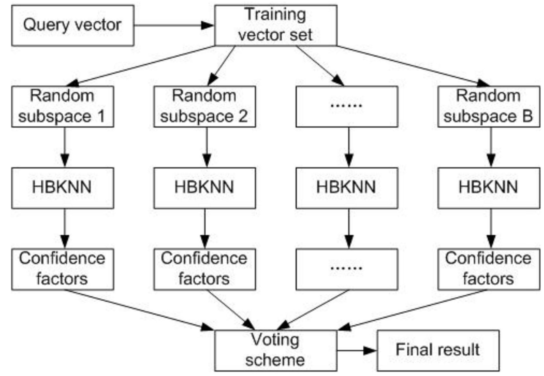


Fig. 2. Overview of the RS-HBKNN classifier.

IV. Random Subspace Ensemble Framework Based on Hybrid k-Nearest Neighbor Classifier

The random subspace technique is often used to generate multiple random subspaces or views, which is able to increase the diversity of the ensemble and improve the final result. Recently, Liu and Tao [78] and Liu *et al.* [79] proposed the multiview Hessian regularization approach and the multiview Hessian discriminative sparse coding approach for image annotation. They also designed the Hessian-regularized co-training method for multiview learning [80], which exploited the local structure of the underlying data manifold by Hessian. Wang *et al.* [86], [87] proposed the subspace indexing model on Grassmann manifold for nearest neighbor search [84], [85], which partitions the global space into local patches with a hierarchical structure, and obtains good results. Motivated by the advantages of the random subspace technique, we design the RS-HBKNN classifier as illustrated in Fig. 2.

In the first step, RS-HBKNN adopts the random subspace technique to generate a set of random subspaces $D^1, D^2, \ldots, D^B$ (where B is the number of subspaces). Specifically, a constant ratio θ ($\theta_{\min} \le \theta \le \theta_{\max}$) is first generated as follows:

$$\theta = \theta_{\min} + \lfloor \tau_1 (\theta_{\max} - \theta_{\min}) \rfloor \tag{18}$$

where θ is a constant ratio of the number of features in the subspace over the total number of features in the original space, $\tau_1 (\tau_1 \in [0, 1])$ is a uniform random variable, and $\theta_{\max}, \theta_{\min} \in [0, 1]$ are the parameters prespecified by the user. Then, RS-HBKNN chooses the feature one by one until θd attributes (where d is the total number of features) are obtained. The index of each randomly selected feature is determined as follows:

$$s = \lfloor 1 + \tau_2 \theta d \rfloor \tag{19}$$

where s is the sth feature in the training data, and $\tau_2 (\tau_2 \in [0, 1])$ is a uniform random variable. Finally, RS-HBKNN adopts the randomly selected θd features to construct a subspace. It repeats the above process B times, and generates a set of random subspaces, D^1, $D^2, \ldots, D^B$.

In the second step, a set of the HBKNN classifiers are trained in the subspace, and generate a set of confidence factors $\{\lambda_1^1, \lambda_2^1, \ldots, \lambda_k^B\}$ that will be used to predict the label

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
SUMMARY OF REAL DATASETS

Dataset	source	K	n	m
Chess	[61]	2	3196	36
Dermatology	[61]	6	358	34
Ionosphere	[61]	2	351	33
Movement-libras	[61]	15	360	90
Optdigits	[61]	10	5620	64
Segment	[61]	7	2310	19
Texture	[61]	11	5500	40
Twonorm	[61]	2	7400	20
Vowel	[61]	11	990	13
Wdbc	[61]	2	569	30
Wine	[61]	3	178	13
Wisconsin	[61]	2	683	9

TABLE II
PARAMETER SETTINGS

Parameters	Default values	Range
The number of subspace B in RS-HKBNN	20	5, 10, 15, 20, 25
ι nearest neighbors in traditional fuzzy KNN	15	5, 10, 15, 20, 25
ε nearest neighbors in FRHKNN	8	3, 5, 8, 10, 12
The proportion between ω_2 and ω_1 in HKBNN	2:1	1:1, 2:1, 3:1, 4:1, 5:1

of the query vector in the testing set (where λ_j^b denotes the value of the confidence factors obtained by HBKNN in the bth subspace, $j \in \{1, \ldots, k\}$, $b \in \{1, \ldots, B\}$).

RS-HBKNN adopts the voting scheme to summarize the set of confidence factors and predict the label y_v of the query vector v, which is as follows:

$$y_v = \arg \max_{j \in \{1,\ldots,k\}} \overline{\lambda}_j \tag{20}$$

$$\overline{\lambda}_j = \frac{1}{B} \sum_{b=1}^{B} \lambda_j^b. \tag{21}$$

The computational cost $T_{\text{RS-HBKNN}}$ of RS-HBKNN is as follows:

$$T_{\text{RS-HBKNN}} = B \cdot T_{\text{HBKNN}}. \tag{22}$$

In general, the time complexity of RS-HBKNN is $O(n \log n)$.

V. EXPERIMENT

The performances of HBKNN, RS-HBKNN, and other classification approaches are evaluated using 12 real-world datasets from the KEEL machine learning repository in Table I, which provides a summary of 12 real-world datasets from the KEEL dataset repository [61] (where K denotes the number of classes, n denotes the number of data samples, and m denotes the number of attributes). The used datasets include some challenging datasets. For example, the movement-libras dataset only has 360 90-dimensional samples, which should be distributed into 15 classes. Since the number of classes is relatively large, while the sample size is small and the given number of attributes is large, which poses one type of challenge to conventional classification approaches.

The classification accuracy (AC) is adopted to measure the quality of predicted labels, which is defined as follows:

$$\text{AC} = \frac{1}{|T_s|} \sum_{f_i \in T_s} 1\{y_i == y_i^{\text{true}}\} \tag{23}$$

where T_s is the set of samples in the testing set, $|\cdot|$ is the cardinality of the set, y_i and y_i^{true} denote the predicted label and the true label of the sample f_i, respectively. The performances of the classifier obtained by HBKNN, RS-HBKNN, and other classification approaches obtained by other classifier ensemble approaches are measured by the average value and the corresponding standard deviation of the accuracy, respectively, after ten runs. We adopt tenfold crossover validation to avoid the effect of the randomness.

In the following experiments, we first explore the effect of different parameter settings. Then, the effect of the components in RS-HBKNN is investigated. Finally, RS-HBKNN will compare with the classical classification approaches, which include single classifiers and ensemble classifiers, on the real datasets in Table I.

A. Effect of Parameter Values

Table II summarizes the parameters of HKBNN and RS-HKBNN along with their default values and ranges, which include the number of subspace B in RS-HKBNN, ι nearest neighbors in traditional fuzzy KNN, ε nearest neighbors in FRHKNN, and the proportion between ω_1 and ω_2 in HKBNN. We vary the values of the parameters one at a time in the following experiments on the chess dataset, the optdigits dataset, and the texture dataset, while setting the values of the other parameters to their default values.

Fig. 3 shows the effect of the number of subspace B in RS-HKBNN. The accuracy value increases gradually as the number of subspace increases in Fig. 3(a), while the accuracy values maintain stable when the number of subspace increases in Fig. 3(b) and (c). The possible reason is as follows: when the number of subspace becomes larger, the diversity of the individual classifiers in the ensemble will increases on some datasets, which will lead to the improvement of the accuracy. As a whole, $B = 20$ is a suitable choice when considering both the accuracy and the computational cost.

Fig. 4 illustrates the effect of ι nearest neighbors in traditional fuzzy KNN. RS-HKBNN achieves the best performance when ι is set to 15. When ι deviates from 15, the performance of RS-HKBNN becomes less satisfactory as shown in Fig. 4(a) and (b). The possible reason is as follows: when ι is small, RS-HBKNN is difficult to capture the local distribution of the samples, which will prevent the proposed approach to obtain the better result. When ι becomes larger than 15, more redundant information will be included, which will decrease the effectiveness of RS-HBKNN. In summary, $\iota = 15$ is a better choice to improve the performance of RS-HBKNN.

Fig. 5 demonstrates the effect of ε nearest neighbors in FRHKNN. The results become better on some datasets, such as the texture dataset in Fig. 5(c), when ε increases, while RS-HBKNN do not achieve the best performance on

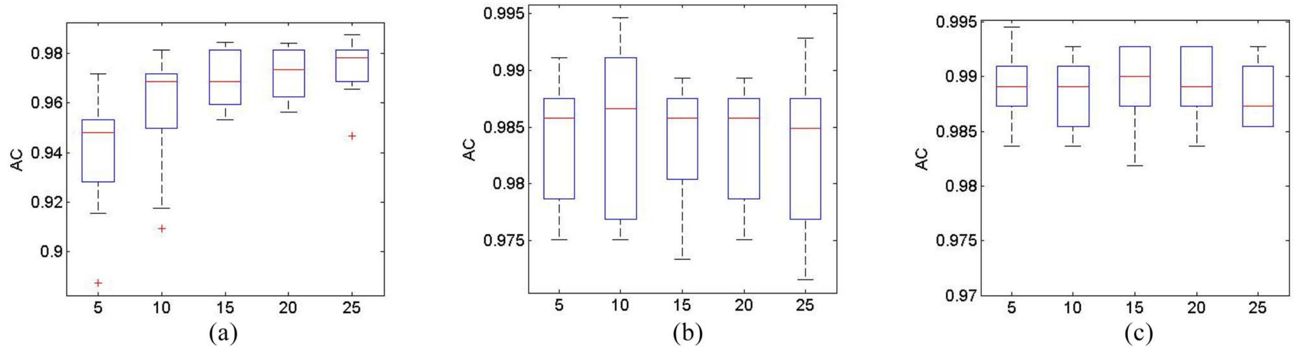


Fig. 3. Effect of the number of subspace B in RS-HKBNN. (a) Chess dataset. (b) Optdigits dataset. (c) Texture dataset.

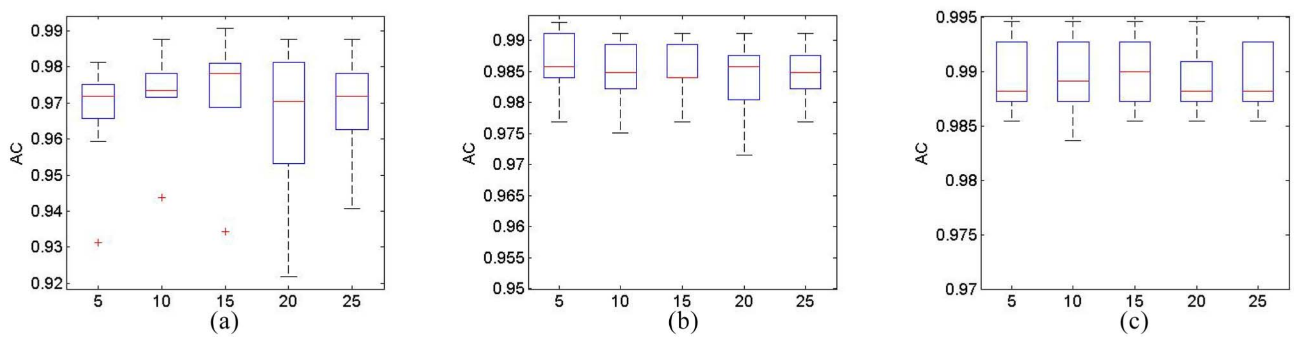


Fig. 4. Effect of ι nearest neighbors in traditional fuzzy KNN. (a) Chess dataset. (b) Optdigits dataset. (c) Texture dataset.

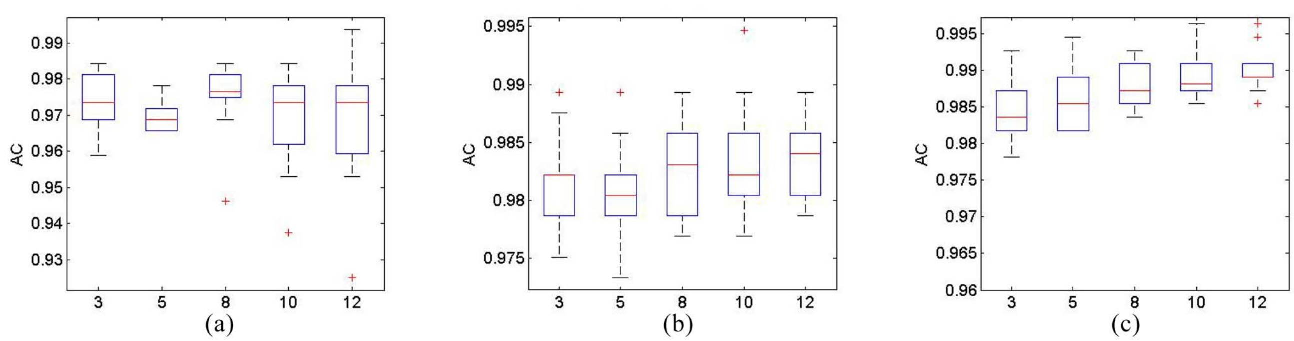


Fig. 5. Effect of ε nearest neighbors in FRHKNN. (a) Chess dataset. (b) Optdigits dataset. (c) Texture dataset.

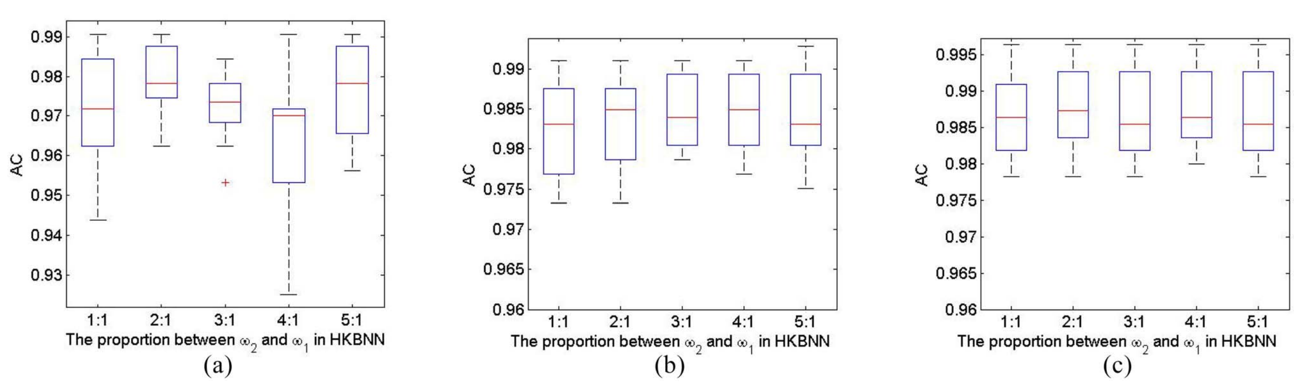


Fig. 6. Effect of the proportion between ω_2 and ω_1 in HKBNN. (a) Chess dataset. (b) Optdigits dataset. (c) Texture dataset.

some datasets, such as the chess dataset in Fig. 5(a), when ε becomes larger. In general, $\varepsilon = 8$ is a good choice to balance the accuracy and the computational cost.

Fig. 6 shows the effect of the proportion between ω_1 and ω_2 in HKBNN. RS-HKBNN is not sensitive to this parameter.

TABLE III
PERFORMANCE OF RSHBKNN, RSHBKNN1, RSHBKNN2, AND RSHBKNN3 WITH RESPECT TO THE AVERAGE VALUE AND THE CORRESPONDING STANDARD DEVIATION OF THE ACCURACY ON ALL THE DATASETS (WHERE THE VALUES IN BOLDFACE INDICATE THE BEST RESULTS)

Dataset	RSHBKNN	RSHBKNN1	RSHBKNN2	RSHBKNN3
Chess	**0.9700** (0.0159)	0.9649 (0.0123)	0.9562 (0.0166)	0.9543 (0.0155)
Dermatology	0.9748 (0.0205)	0.9721 (0.0346)	**0.9778** (0.0255)	0.9722 (0.0262)
Ionosphere	0.9176 (0.0541)	0.9139 (0.0381)	0.8824 (0.0568)	**0.9278** (0.0375)
Movement-libras	0.8744 (0.0766)	**0.8856** (0.0705)	0.8600 (0.0635)	0.8444 (0.0775)
Optdigits	0.9854 (0.0056)	0.9854 (0.0051)	0.9835 (0.0037)	**0.9870** (0.0054)
Segment	**0.9719** (0.0140)	0.9628 (0.0170)	0.9593 (0.0098)	0.9675 (0.0146)
Texture	**0.9891** (0.0032)	0.9824 (0.0080)	0.9864 (0.0033)	0.9875 (0.0056)
Twonorm	0.9735 (0.0056)	0.9724 (0.0054)	**0.9753** (0.0035)	0.9708 (0.0065)
Vowel	**0.9909** (0.0075)	0.9848 (0.0167)	0.9404 (0.0121)	0.9333 (0.0313)
Wdbc	**0.9771** (0.0263)	0.9737 (0.0237)	0.9719 (0.0206)	0.9754 (0.0189)
Wine	**0.9944** (0.0176)	0.9882 (0.0249)	0.9771 (0.0297)	0.9833 (0.0268)
Wisconsin	**0.9825** (0.0134)	0.9737 (0.0089)	0.9737 (0.0275)	0.9752 (0.0204)

B. Effect of Components

In order to investigate the effect of component in RS-HBKNN, we remove each component one at a time in the following experiments, while maintaining the other components. RSHBKNN, RSHBKNN1, RSHBKNN2, and RSHBKNN3 denote the RS-HBKNN approach, the RS-HBKNN approach without considering traditional fuzzy KNN, the RS-HBKNN approach without considering the relative transformation operation and the RS-HBKNN approach without considering both the traditional fuzzy KNN and the relative transformation operation, respectively.

Table III provides a comparison of the performance of RSHBKNN, RSHBKNN1, RSHBKNN2, and RSHBKNN3 in terms of the average value and the corresponding standard deviation of the accuracy on all the datasets in Table I. We have several interesting observations from the table. First, RSHBKNN outperforms its competitors on seven out of 12 datasets. The possible reason is that RSHBKNN keeps a balance between FRHKNN and traditional fuzzy KNN, which provides an opportunity for RSHBKNN to capture the global and local information to improve the final result. Second, both the relative transformation operation and the traditional fuzzy KNN play an important role in the RS-HBKNN approach. If one of them is missing, the capability of RS-HBKNN to obtain the best result will be compromised. In general, all the components in RS-HBKNN should work together to allow RS-HBKNN to obtain better results.

C. Comparison With the Classical Classification Approaches

The proposed approach RS-HBKNN is compared with the classical classification approaches, such as the naive Bayesian (NB) approach [63], the KNN approach [64], the J48 approach [65], the 1R (oneR) classifier [66], and the random tree (RT) approach [62] with respect to the accuracy on all the datasets. All the classical classification approaches are run on the Weka platform [62] with the default setting.

Fig. 1 in the supplementary file shows the performance of RS-HBKNN, NB, KNN, J48, oneR, and RT on all the datasets. It has been seen from the following.

1) RS-HBKNN outperforms the classical classification approaches on most of the datasets, especially on the ionosphere dataset, the movement-libras dataset, the vowel dataset, and the Wisconsin dataset. The possible reason is as follows: first, RS-HBKNN adopts the HBKNN classifier as the basic classifier, which is able to capture the global and local information of the samples. Second, it adopts the random subspace technique to take into account the effect of different attribute combinations, which will be useful to improve the final result. Third, RS-HBKNN is able to integrate the predicted labels obtained by different basic classifiers to generate a more accuracy final result.
2) The results obtained by the oneR approach are not satisfactory. Since the oneR approach only uses a very simple classification rule, it is not suitable for the classification task in the complex real-world datasets.
3) KNN, J48, and RT achieve good performance on most of datasets, while NB obtain less satisfactory results on some datasets. For example, the performance of NB on the segment dataset, the texture dataset, and the vowel dataset. The possible reason is that NB is not suitable for the datasets with the large number of clusters. Though the results obtained by RS-HBKNN are better than the results obtained by the classical classification approaches, RS-HBKNN spend more computational cost to obtain the better results. In summary, RS-HBKNN is a better choice for the classification task which belongs to the offline task.

D. Comparison With Other Classifier Ensemble Approaches

We also compare RS-HBKNN (P1) with several the state-of-the-art classifier ensemble approaches, which include the AdaBoostM1 approach (P2, [67]), the bagging approach (P3, [68]), the dagging approach (P4, [71]), the filtered classifier approach (P5, [62]), the LogitBoost approach (P6, [72]), the random committee approach (P7, [62]), the ordinal class classifier approach (P8, [62]), the random subspace approach (P9, [73]), and the rotation forest approach (P10, [74]). All the approaches are run on the Weka platform with the default setting.

Fig. 7 illustrates the performance of different classifier ensemble approaches with respect to AC on all the datasets. It is observed that RS-HBKNN outperforms its competitors on most of datasets, and achieves the best performance on nine out of 12 datasets. For example, the movement-libras dataset is a challenge dataset characterized with the high dimension and the large number of classes. RS-HBKNN obtains the best result with the average accuracy

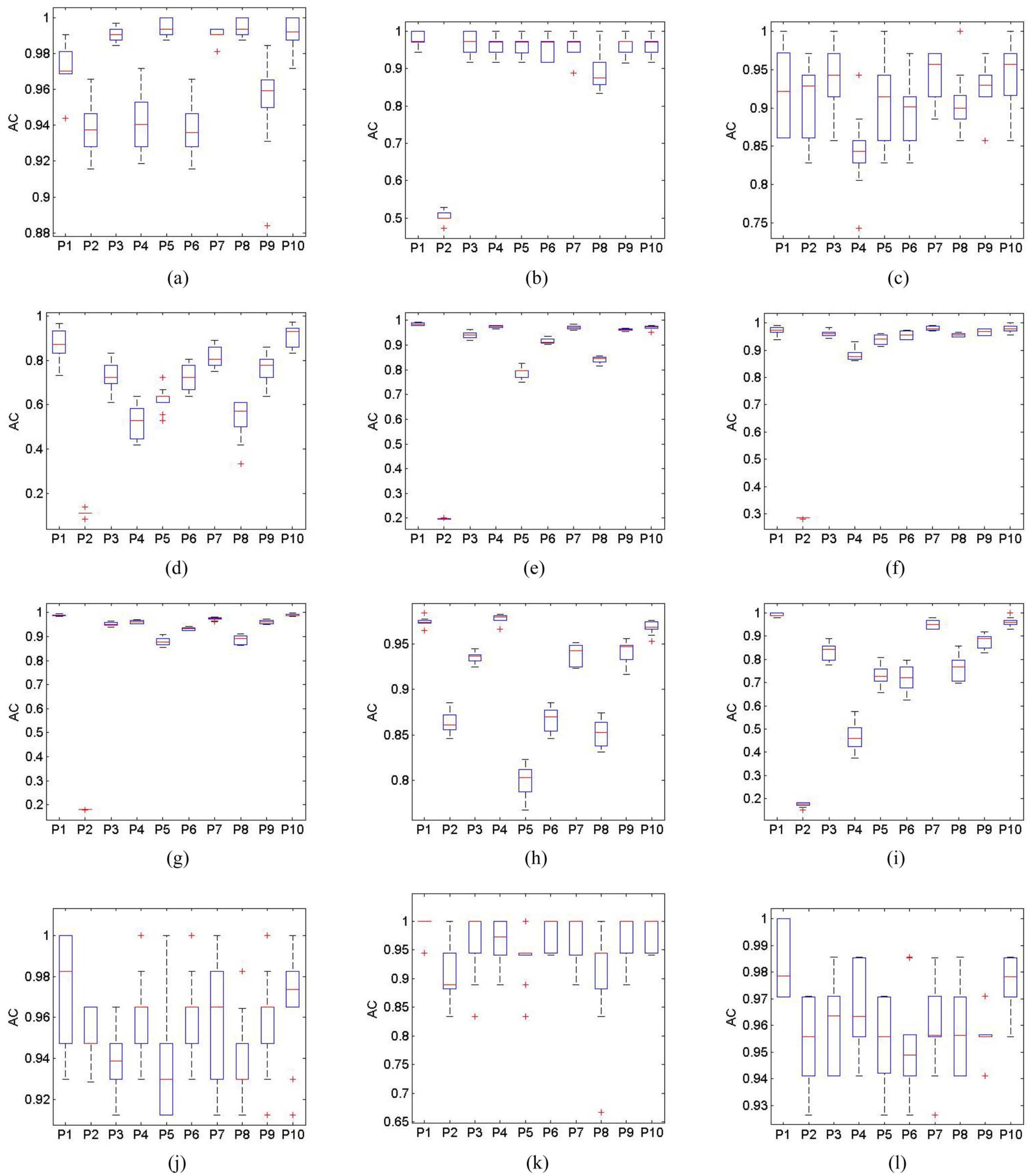


Fig. 7. Performances of different classifier ensemble approaches with respect to AC on all the datasets (where P1–P10 denote the RS-HBKNN approach, the AdaBoostM1 approach, the bagging approach, the dagging approach, the filtered classifier approach, the LogitBoost approach, the random committee approach, the ordinal class classifier approach, the random subspace approach, and the rotation forest approach, respectively). (a) Chess, (b) dermatology, (c) ionosphere, (d) movement-libras, (e) optdigits, (f) segment, (g) texture, (h) twonorm, (i) vowel, (j) wdbc, (k) wine, and (l) Wisconsin datasets.

value 0.8744 on the movement-libras dataset, and significantly outperforms other approaches. The possible reasons are as follows: 1) RS-HBKNN makes use of the random subspace technique to explore the structure of the dataset in the attribute dimension, which will reduce the effect of the noisy attributes and 2) it adopts HBKNN as the basic classifier, which is able to capture the local and the global information of the query sample. In summary, RS-HBKNN is a better choice when compared with other classifier ensemble approaches.

The nonparametric tests [75], [76] is used to compare the classifier ensemble approaches, such as the RS-HBKNN approach, the AdaBoostM1 approach, the bagging approach, the dagging approach, the filtered classifier approach, the LogitBoost approach, the random committee approach, the ordinal class classifier approach, the random subspace approach, and the rotation forest approach, over multiple datasets in Table I. The Friedman test is a nonparametric procedure for multiple comparison in hypothesis testing situations, which is designed by Friedman [77]. It first computes the ranks of the learning algorithms for each dataset separately from the average accuracy values obtained by multiple classifier ensemble algorithms on multiple datasets, and obtains a new matrix $\{R_{ij}\}_{I\times J}$ (where the entry R_{ij} denotes the rank of the ith algorithm on the dataset F_j), $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$, I denotes the number of the learning algorithms, and J denotes the number of datasets). Then, the test statistics Θ is calculated as follows:

$$\Theta = \frac{12J}{I(I+1)}\left[\sum_i \overline{R}_{i.}^2 - \frac{I(I+1)^2}{4}\right] \tag{24}$$

$$\overline{R}_{i.} = \frac{1}{J}\sum_{j=1}^{J} R_{ij}. \tag{25}$$

In the following, the value of Θ can be approximately by that of a chi-squared χ^2 distribution when I or J is large. The corresponding p-value is obtained by the following equation:

$$p = P\left(\chi^2_{I-1} \geq \Theta\right). \tag{26}$$

If I or J is small, the approximation of χ^2 distribution becomes unsatisfactory, and the p-value should be obtained from the table of Θ which is prepared for the Friedman test. Finally, the suitable *post hoc* multiple comparisons tests, such as the Bonferroni–Dunn test, the Holm test, the Hochberg test, and the Hommel test, will be performed, once the p-value is significant.

Tables I–III in the supplementary file show the results of multiple comparison of the classifier ensemble algorithms using different nonparametric statistical procedures, such as the Bonferroni–Dunn test, the Holm test, the Hochberg test, and the Hommel test. It is observe that the average ranking of RS-HBKNN is smaller than those of other classifier ensemble approaches as shown in Table I, and the results in Tables II and III in the supplementary file shows the results of multiple comparison of the classifier ensemble algorithms using different nonparametric statistical procedures, such as the Bonferr indicate the significance between RS-HBKNN and other nine classifier ensemble approaches, which jointly illustrate the distinct improvement of RS-HBKNN over other classifier ensemble approaches.

We also compare the computational cost of the different classification or classifier ensemble approaches and tabulated the results in Table IV in the supplementary file. It has been seen that the filtered classifier approach obtains the minimum running time on most of the datasets due to its simplified process. More computational costs are spent by the proposed approach RS-HBKNN on most of the datasets. The possible reason is that RS-HBKNN uses the relative transformation operator and the calculation of local hyperplane distances to improve the performance, which increases the inevitable computational cost. In general, RS-HBKNN is good choice for the special datasets when the accuracy of the algorithms become more important.

TABLE IV
SUMMARY OF SYNTHETIC DATASETS

Dataset	K	m	m'	n/c
Synthetic1	5	500	200	2000, 2000, 2000, 2000, 2000
Synthetic2	5	500	200	1000, 1500, 2000, 2500, 3000
Synthetic3	5	500	200	200, 1100, 2000, 2900, 3800

In order to investigate the performance of the proposed approach on the imbalance datasets with noisy data, we generate three synthetic datasets in the space $[0, 1]^m$ (where m is the number of attributes) using a set of Gaussian distributions with randomly selected centers, and with the covariance matrices of all distributions set to 0.95 I (where I denotes the identity matrix). Table IV illustrates a summary of all the synthetic datasets (where m' denotes the number of noisy attributes, and n/c denotes the number of samples for each class). These datasets contains 10 000 500-dimensional data sample. Each sample contains 200 noisy attributes. Fig. 2 in the supplementary file shows the results obtained by different approaches with respect to AC on the synthetic datasets. It is observed that RS-HBKNN achieves good performance on all the synthetic datasets. In summary, RS-HBKNN is suitable for the noisy datasets and the imbalance datasets.

VI. CONCLUSION

In this paper, we address the limitations of the conventional KNN approaches, and propose the HBKNN classification approach. When compared with traditional KNN approaches, HBKNN tackles the imbalance problem, the sparse problem and the noise problem using the combination of the FRHKNN classifier and traditional KNN approach. In order to improve the performance of HBKNN, the RS-HBKNN classifier is designed. We have conducted experiments on 12 real-world datasets from the KEEL dataset repository for the classification task and draw several conclusions as follows: 1) suitable parameter values in RS-HBKNN are able to improve the accuracy of the classifiers; 2) different stages of RS-HBKNN contribute to the final results; and 3) RS-HBKNN outperforms most of traditional approaches on the real-world KEEL datasets. In the future, we will analyze RS-HBKNN theoretically, and apply it to different areas.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for the constructive advice on the revision of this paper.

REFERENCES

[1] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[2] T. Denoeux, "A k-nearest neighbor classification rule based on Dempster–Shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 804–813, May 1995.

[3] T. Shanableh, K. Assaleh, and M. Al-Rousan, "Spatio-temporal feature-extraction techniques for isolated gesture recognition in Arabic sign language," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 641–650, Jun. 2007.

[4] X. Geng, D.-C. Zhan, and Z.-H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1098–1107, Dec. 2005.

[5] A. Bosch, A. Zisserman, and X. Muoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.

[6] J. Yang, L. Zhang, J. Yang, and D. Zhang, "From classifiers to discriminators: A nearest neighbor rule induced discriminant analysis," *Pattern Recognit.*, vol. 44, no. 7, pp. 1387–1402, 2011.

[7] J. Xu, J. Yang, and Z. Lai, "K-local hyperplane distance nearest neighbor classifier oriented local discriminant analysis," *Inf. Sci.*, vol. 232, pp. 11–26, May 2013.

[8] H. Frigui and P. Gader, "Detection and discrimination of land mines in ground-penetrating radar based on edge histogram descriptors and a possibilistic *K*-nearest neighbor classifier," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 1, pp. 185–199, Feb. 2009.

[9] M. Li, M. M. Crawford, and J. Tian, "Local manifold learning-based *k*-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[10] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Distance-based image classification: Generalizing to new classes at near-zero cost," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2624–2637, Nov. 2013.

[11] P. Maji, "Fuzzy–rough supervised attribute clustering algorithm and classification of microarray data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 222–233, Feb. 2011.

[12] M. L. Raymer, T. E. Doom, L. A. Kuhn, and W. F. Punch, "Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 5, pp. 802–813, Oct. 2003.

[13] P. Vincent and Y. Bengio, "K-local hyperplane and convex distance nearest neighbour algorithms," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 14. Vancouver, BC, Canada, 2002, pp. 985–992.

[14] C. J. Veenman and M. J. T. Reinders, "The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1417–1429, Sep. 2005.

[15] Q. Zhao, "Inducing NNC-trees with the R4-rule," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 3, pp. 520–533, Jun. 2005.

[16] B. Li, Y. W. Chen, and Y.-Q. Chen, "The nearest neighbor algorithm of local probability centers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 1, pp. 141–154, Feb. 2008.

[17] Y. Gao, J. Pan, G. Ji, and Z. Yang, "A novel two-level nearest neighbor classification algorithm using an adaptive distance metric," *Knowl-Based Syst.*, vol. 26, pp. 103–110, Feb. 2012.

[18] Y. Xu *et al.*, "Coarse to fine K nearest neighbor classifier," *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 980–986, 2013.

[19] N. A. Samsudin and A. P. Bradley, "Nearest neighbour group-based classification," *Pattern Recognit.*, vol. 43, no. 10, pp. 3458–3467, 2010.

[20] J. Zhang and J. Yang, "Linear reconstruction measure steered nearest neighbor classification framework," *Pattern Recognit.*, vol. 47, no. 4, pp. 1709–1720, 2014.

[21] J. Chien and C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1644–1649, Dec. 2002.

[22] S. Bermejo and J. Cabestany, "Adaptive soft *k*-nearest-neighbour classifiers," *Pattern Recognit.*, vol. 33, no. 12, pp. 1999–2005, 2000.

[23] Q. Gao and Z. Wang, "Center-based nearest neighbor classifier," *Pattern Recognit.*, vol. 40, no. 1, pp. 346–349, 2007.

[24] G. Wen, J. Wen, J. Wei, L. Jiang, and Z. Yu, "Perceptual relativity-based local hyperplane," *Neurocomputing*, vol. 97, pp. 155–163, Nov. 2012.

[25] W. Zheng, L. Zhao, and C. Zou, "Locally nearest neighbor classifiers for pattern classification," *Pattern Recognit.*, vol. 37, no. 6, pp. 1307–1309, 2004.

[26] E. Bax, "Validation of *k*-nearest neighbor classifiers," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3225–3234, May 2012.

[27] H. Cevikalp, B. Triggs, and R. Polikar, "Nearest hyperdisk methods for high-dimensional classification," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, 2008, pp. 120–127.

[28] A. K. Ghosh, P. Chaudhuri, and C. A. Murthy, "Multiscale classification using nearest neighbor density estimates," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 5, pp. 1139–1148, Oct. 2006.

[29] G. Parthasarathy and B. N. Chatterji, "A class of new KNN methods for low sample problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 3, pp. 715–718, May/Jun. 1990.

[30] C. Domeniconi, D. Gunopulos, and J. Peng, "Large margin nearest neighbor classifiers," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 899–909, Jul. 2005.

[31] A. Majumdar and R. K. Ward, "Robust classifiers for data reduced via random projections," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 5, pp. 1359–1371, Oct. 2010.

[32] Q. Hu, P. Zhu, Y. Yang, and D. Yu, "Letters: Large-margin nearest neighbor classifiers via sample weight learning," *Neurocomputing*, vol. 74, no. 4, pp. 656–660, 2011.

[33] J. A. Sáez, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognit.*, vol. 46, no. 1, pp. 355–364, 2013.

[34] A. Argentini and E. Blanzieri, "About neighborhood counting measure metric and minimum risk metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 763–765, Apr. 2010.

[35] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[36] J. Derrac, I. Triguero, S. Garcia, and F. Herrera, "Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 5, pp. 1383–1397, Oct. 2012.

[37] I. Triguero, S. García, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognit.*, vol. 44, no. 4, pp. 901–916, 2011.

[38] B. Zhang and S. N. Srihari, "Fast *k*-nearest neighbor classification using cluster-based trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 525–528, Apr. 2004.

[39] S. Hernández-Rodríguez, J. F. Martínez-Trinidad, and J. A. Carrasco-Ochoa, "Fast *k* most similar neighbor classifier for mixed data (tree *k*-MSN)," *Pattern Recognit.*, vol. 43, no. 3, pp. 873–886, 2010.

[40] P. Viswanath, N. Murty, and S. Bhatnagar, "Overlap pattern synthesis with an efficient nearest neighbor classifier," *Pattern Recognit.*, vol. 38, no. 8, pp. 1187–1195, 2005.

[41] S. Lucey and A. B. Ashraf, "Nearest neighbor classifier generalization through spatially constrained filters," *Pattern Recognit.*, vol. 46, no. 1, pp. 325–331, 2013.

[42] A. K. Ghosh, P. Chaudhuri, and C. A. Murthy, "On visualization and aggregation of nearest neighbor classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1592–1602, Oct. 2005.

[43] Z.-H. Zhou and Y. Yu, "Ensembling local learners through multimodal perturbation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 4, pp. 725–735, Aug. 2005.

[44] L. Nanni and A. Lumini, "An ensemble of K-local hyperplanes for predicting protein-protein interactions," *Bioinformatics*, vol. 22, no. 10, pp. 1207–1210, 2006.

[45] H. Altnay, "Ensembling evidential *k*-nearest neighbor classifiers through multi-modal perturbation," *Appl. Soft Comput.*, vol. 7, no. 3, pp. 1072–1083, 2007.

[46] B. Acha, C. Serrano, I. Fondon, and T. Gomez-Cia, "Burn depth analysis using multidimensional scaling applied to psychophysical experiment data," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 1111–1120, Jun. 2013.

[47] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[48] J. Yang, P. Yu, and B. Kuo, "A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 3, pp. 1279–1293, Mar. 2010.

[49] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1863–1868, Nov. 2006.

[50] D. L. Sussman, M. Tang, and C. E. Priebe, "Consistent latent position estimation and vertex classification for random dot product graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 48–57, Jan. 2014.

[51] K. Jafari-Khouzani and H. Soltanian-Zadeh, "Rotation-invariant multiresolution texture analysis using Radon and wavelet transforms," *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 783–795, Jun. 2005.

[52] R. Gutierrez-Osuna and H. T. Nagle, "A method for evaluating data-preprocessing techniques for odour classification with an array of gas sensors," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 5, pp. 626–632, Oct. 1999.

[53] N. Mezghani *et al.*, "Automatic classification of asymptomatic and osteoarthritis knee gait patterns using kinematic data features and the nearest neighbor classifier," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 3, pp. 1230–1232, Mar. 2008.

[54] S. Shah and P. S. Sastry, "Fingerprint classification using a feedback-based line detector," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 1, pp. 85–94, Feb. 2004.

[55] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 103–123, Jan. 2009.

[56] S. Singh, J. Haddon, and M. Markou, "Nearest-neighbour classifiers in natural scene analysis," *Pattern Recognit.*, vol. 34, no. 8, pp. 1601–1612, 2001.

[57] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 3, pp. 871–879, Mar. 2009.

[58] C. Liu and M. Nakagawa, "Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition," *Pattern Recognit.*, vol. 34, no. 3, pp. 601–615, 2001.

[59] P. Cunningham and S. J. Delany, "*k*-nearest neighbour classifiers," School Comput. Sci. Informat., Univ. Coll. Dublin, Dublin, Ireland, Tech. Rep. UCD-CSI-2007-4, 2007.

[60] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving *k*-nearest-neighbor for classification," in *Proc. 4th Int. Conf. Fuzzy Syst. Knowl. Disc. (FSKD)*, vol. 1. Haikou, China, 2007, pp. 679–683.

[61] 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.

[62] M. Hall *et al.*, "The WEKA data mining software: An update," *SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[63] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proc. 11th Conf. Uncertainty Artif. Intell.*, San Mateo, CA, USA, 1995, pp. 338–345.

[64] D. Aha and D. Kibler, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.

[65] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[66] R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learn.*, vol. 11, no. 1, pp. 63–91, 1993.

[67] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1996, pp. 148–156.

[68] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[69] M. I. Lopez, C. Romero, S. Ventura, and J. M. Luna, "Classification via clustering for predicting final marks starting from the student participation in forums," in *Proc. 5th Int. Conf. Educ. Data Min.*, Chania, Greece, 2012, pp. 148–151.

[70] R. Kohavi, "Wrappers for performance enhancement and oblivious decision graphs," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 1995.

[71] K. M. Ting and I. H. Witten, "Stacking bagged and dagged models," in *Proc. 14th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 1997, pp. 367–375.

[72] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

[73] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[74] L. I. Kuncheva and J. J. Rodriguez, "Classifier ensembles with a random linear oracle," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 500–508, Apr. 2007.

[75] S. García and F. Herrera, "An extension on 'statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, Dec. 2008.

[76] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.

[77] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc. (Amer. Statist. Assoc.)*, vol. 32, no. 200, pp. 675–701, 1937.

[78] W. Liu and D. Tao, "Multiview Hessian regularization for image annotation," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2676–2687, Jul. 2013.

[79] W. Liu, D. Tao, J. Cheng, and Y. Tang, "Multiview Hessian discriminative sparse coding for image annotation," *Comput. Vis. Image Understand.*, vol. 118, pp. 50–60, Jan. 2014.

[80] W. Liu, Y. Li, X. Lin, D. Tao, and Y. Wang, "Hessian-regularized co-training for social activity recognition," *PLoS ONE*, vol. 9, no. 9, 2014, Art. ID e108474.

[81] C. Domeniconi and B. Yan, "Nearest neighbor ensemble," in *Proc. ICPR*, vol. 1. Cambridge, U.K., 2004, pp. 228–231.

[82] C. Domeniconi and B. Yan, "On error correlation and accuracy of nearest neighbor ensemble classifiers," in *Proc. SDM*, Newport Beach, CA, USA, 2005, pp. 217–226.

[83] N. Bhatia and Vandana, "Survey of nearest neighbor techniques," *Int. J. Comput. Sci. Inf. Security*, vol. 8, no. 2, pp. 302–305, 2010.

[84] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases (VLDB)*, Edinburgh, U.K., 1999, pp. 518–529.

[85] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.

[86] X. Wang, Z. Li, L. Zhang, and J. Yuan, "Grassmann hashing for approximate nearest neighbor search in high dimensional space," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Barcelona, Spain, 2011, pp. 1–6.

[87] X. Wang, Z. Li, and D. Tao, "Subspaces indexing model on Grassmann manifold for image search," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2627–2635, Sep. 2011.

[88] J. Yu, D. Tao, 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.

[89] 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.

[90] 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.

[91] Z. Yu, L. Li, J. Liu, and G. Han, "Hybrid adaptive classifier ensemble," *IEEE Trans. Cybern.*, vol. 42, no. 2, pp. 177–190, Feb. 2015.

[92] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[93] 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.

[94] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.

[95] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[96] C. Xu, D. Tao, and C. Xu, "Multi-view intact space learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015. DOI: 10.1109/TPAMI.2015.2417578.

[97] D. Tao, L. Jin, W. Liu, and X. Li, "Hessian regularized support vector machines for mobile image annotation on the cloud," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 833–844, Jun. 2013.

[98] D. Tao, X. Lin, L. Jin, and X. Li, "Principal component 2-D long short-term memory for font recognition on single Chinese characters," *IEEE Trans. Cybern.*, 2015. DOI: 10.1109/TCYB.2015.2414920.

[99] P. Yang *et al.*, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 445–455, Mar. 2014.

[100] V. Soto, S. Garcia-Moratilla, G. Martinez-Munoz, D. Hernandez-Lobato, and A. Suarez, "A double pruning scheme for boosting ensembles," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2682–2695, Dec. 2014.

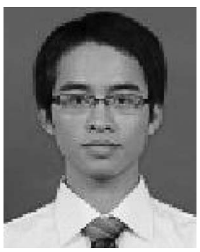
[101] R. Diao, F. Chao, T. Peng, N. Snooke, and Q. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1259–1268, Aug. 2014.

[102] X.-Z. Wang, Y.-L. He, and D. D. Wang, "Non-naive Bayesian classifiers for classification problems with continuous attributes," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 21–39, Jan. 2014.

Zhiwen Yu (S'06–M'08–SM'14) received the Ph.D. degree in computer science 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 current research interests include data mining, machine learning, bioinformatics, and pattern recognition. He has published over 90 referred journal papers and international conference papers, including the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Cybernetics, the IEEE Transactions on Multimedia, the IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Computational Biology and Bioinformatics, the IEEE Transactions on NanoBioscience, pattern recognition, information sciences, and *Bioinformatics*.

Dr. Yu is a Senior Member of the China Computer Federation.



Hantao Chen received the B.Sc. and the M.Phil degrees in computer science from the South China University of Technology, Guangzhou, China, in 2012 and 2015, respectively.

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

Jiming Liu (F'11) received the M.Eng. and Ph.D. degrees from McGill University, Montreal, QC, Canada, in 1990 and 1994, respectively.

He is a Chair Professor of Computer Science and an Associate Dean with the Faculty of Science (Research and Graduate Studies), Hong Kong Baptist University, Hong Kong. His current research interests include data analytics, complex network analysis, digital health, data-driven computational epidemiology, and complex systems modeling.

Prof. Liu has served as the Founding Editor-in-Chief of the *Web Intelligence Journal* (IOS), an Associate Editor for *Big Data and Information Analytics* (AIMS), the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, and *Computational Intelligence* (Wiley), and an Editorial Board Member for several other international journals.

Jane You received the B.Eng. degree in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 1986, and the Ph.D. degree in computer science from La Trobe University, Melbourne, VIC, Australia, in 1992.

She is currently a Professor with the Department of Computing and the Chair of the Department Research Committee with Hong Kong Polytechnic University, Hong Kong, where she has been a Principal Investigator for one Innovation & Technology Fund project, three General Research Fund projects, and several other joint grants since 1998. She is a Team Member for two successful patents, such as one HK patent and one U.S. patent. She has researched extensively in the fields of image processing, medical imaging, computer-aided diagnosis, and pattern recognition. Her research output on retinal imaging has successfully led to technology transfer with clinical applications. She has over 190 research papers published with over 1000 nonself citations.

Prof. You was a recipient of the Special Prize and the Gold Medal with Jury's Commendation at the 39th International Exhibition of Inventions of Geneva in 2011 for her current work on retinal imaging, the Second Place in SPIE Medical Imaging'09 Retinopathy Online Challenge, and three awards, including Hong Kong Government Industrial Award. She is an Associate Editor of *Pattern Recognition* and other journals.

Hareton Leung received the Ph.D. degree in computer science from the University of Alberta, Edmonton, AB, Canada, in 1990.

He is an Associate Professor and the Director of the Laboratory for Software Development and Management, Department of Computing, Hong Kong Polytechnic University, Hong Kong. His current research interests include software testing, project management, risk management, quality and process improvement, software metrics, and e-health.

Guoqiang Han received the B.Sc. degree in applied mathematics from Zhejiang University, Hangzhou, China, in 1982, and the master's and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 1985 and 1988, respectively.

He is a Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, where he is the Head of the School of Computer Science and Engineering. His current research interests include multimedia, computational intelligence, machine learning, and computer graphics. He has published over 100 research papers.