Neurocomputing 97 (2012) 155–163

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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

Perceptual relativity-based local hyperplane classification

Guihua Wen [a,b,*], Lijun Jiang [a], Jun Wen [c], Jia Wei [a], Zhiwen Yu [a]

[a] South China University of Technology, Guangzhou 510641, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[c] Hubei Institute for Nationalities, Ensi 445000, China

ARTICLE INFO

Article history:
Received 19 July 2011
Received in revised form
4 November 2011
Accepted 29 March 2012
Communicated by W. Lu
Available online 31 May 2012

Keywords:
Local hyperplane
Nearest neighbors
Classification
Perceptual relativity

ABSTRACT

The *k*-local hyperplane distance nearest neighbors classification (HKNN) builds a non-linear decision surface with maximal local margin in the input space, with invariance inferred from the local neighborhood rather than the prior knowledge, so that it performs very well in many applications. However, it still cannot be comparable with human being in classification on the noisy, the sparse, and the imbalance data. This paper proposes a new approach,called relative local hyperplane classifier(RLHC),to overcome this problem by utilizing the perceptual relativity to HKNN. It finds *k* nearest neighbors for the query sample from each class and then performs the relative transformation over all these nearest neighbors to build the relative space. Subsequently, each local hyperplane is constructed in the relative space, which is then applied to perform the classification. Experimental results on both real and simulated data suggest that the proposed approach often gives the better results in classification and robustness.

1. Introduction

The *k* nearest neighbor (KNN) classification is one of the most popular and simple methods for classification [1–4]. It is well suited for multiclass problems with large amounts of training data. Despite its simplicity, KNN as well as its variants is competitive with the state of the art on various vision benchmarks [5] and is considered one of the top 10 methods in data mining. Theoretically, the one nearest neighbor (1NN) has asymptotic error rate that is at most twice the Bayes error rate, independent of the distance metric used [6], whereas the asymptotic performance of KNN is even better than that of the 1NN [7]. KNN requires the tuning of only one free parameter and does not assume any particular statistical distribution of the training data [8]. However, it and its variants are still confronted with some problems [9].

(1) *The sparse problem*: In many practical classification tasks, such as face recognition, it is difficult and expensive to collect a large number of training samples.When only few training samples compared to the intrinsic dimensionality of the feature space are available, KNN is not guaranteed to obtain the optimal results [9–11]. The performance of KNN relies on if an appropriate distance metric is used to faithfully model the underlying relationships such as the similarity between the input data points [12,5]. Currently some well-designed distance metrics can be applied or learned [13,12], including locally adaptive distance [1,14–16], kernel trick [17], neighbor counting [18], Mahalanobis distance metric [12], and data gravitation [19]. As a matter of fact, the sparse problem can also exist in the low dimensional space if the samples are not representative and the clear geometrical shape of data are not formed. Besides defining the appropriate distance metric to solve the sparse problem, another way is to virtually enlarge the training samples by using local manifold approaches, such as nearest feature line (NFL) [20], *k*-local hyperplane distance nearest neighbor (HKNN) [21], nearest neighbor line (NNL) [24], and center-based nearest neighbor (CNN) [25]. These ideas have also been validated in the feature extraction [26–28]. These approaches try to approximate the local data by some kind of low-dimensional manifolds, aiming at virtually enlarging the training set to mitigate the limitation arising from the small number of training samples.

(2) *The imbalance problem*: This problem will be met when the data in one class heavily outnumbers the data in another class, the class boundary can be skewed towards the class with few data samples [29]. To solve the problem, besides preprocessing the training data such as by resampling and fuzzifying training data [30], some more efficient classification rules can be designed. For example, local mean classifier (LMC) uses the categorical *k* nearest neighbors of the query sample to compute the local centers per class and to classify the unseen query sample in terms of the distance between the query and these centers [31].

* Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.
E-mail address: crghwen@scut.edu.cn (G. Wen).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.03.018

In this way, the same number of samples from each class is utilized in classification. This similar idea has been extended by utilizing the probability center (LPC) [2], HKNN [21], and ALH [37,22,23].

(3) *The noise problem*: KNN and its variants are also very sensitive to outliers or noises as they equally treat the noisy samples and normal samples. The main reason is that we have difficulties to distinguish the noisy samples from normal ones. One way to overcome this problem is to apply fuzzy mechanism and weighting techniques to assess conflicts among the nearest neighbors [39]. Another complicated method is to integrate fuzzy set with evidence theory [32]. Since that we cannot distinguish and then remove the noisy samples from the training data, an effective way is to reduce their negative influences.

HKNN solves the imbalance problem and the sparse problem at the same time, so that it performs very well in a variety of applications [33,22,23]. However, it does not solve the noise problem. Particularly, it cannot be comparable with human being when performing classification on the noisy, the sparse, and the high dimensional data [34]. As in such cases, the performance of used distance functions easily becomes invalidate [18]. This paper proposes a new approach to overcome this problem that utilizes the perceptual relativity to HKNN [52]. It finds k nearest neighbors for the query sample from each class and then performs the relative transformation over all these nearest neighbors to build the relative space. Subsequently, each local hyperplane is constructed in the relative space, where the classification is performed. The major contributions are as follows.

1. The perceptual relativity has been applied to improve the classification performance on the sparse, the noisy or the imbalanced data, indicating that the other perceptual laws in cognitive psychology can be considered for classification.
2. A new classifier has been designed from HKNN with the relative transformation. This classifier is efficient, simple, general and easy to implement, without any additional parameter.

The rest of this paper is organized as follows: Section 2 presents some elementary concepts. A new classifier is designed in Section 3. The proposed classifier is evaluated through experiments in Section 4. Section 5 presents a review of related work. The paper is concluded with a summary and a discussion of possible future work.

2. Elementary concepts

In this section, we present two concepts that our approach is based upon and which will serve as building blocks.

2.1. HKNN

In the case of a finite number of samples, "missing" samples would appear as "holes" introducing artifacts in the decision surface produced. To fix this problem, the idea of HKNN is to somehow fantasize the missing points, based on a local linear approximation of the manifold of each class. It first selects k nearest neighbors from each class as the class prototypes, and then constructs a local hyperplane to approximate the local manifold of each class based on these class prototypes. Hence, the class label of the query is assigned according to the distance between the query and the local hyperplane of each class. Let $x \in w_j$ be a training sample belong to the class ω_j, n_c be the number of the classes, q be an arbitrary query sample and $d(x, y)$ be a distance metric, we obtain a local categorical set consisting of k nearest ω_j samples, denoted as $X_{\omega_j}(q,k)$, where $d(q,x) < d(q,p)$ if p is not in $X_{\omega_j}(q,k)$ but $p \in \omega_j$. Similarly, the k nearest neighbors of the query sample q from all classes is denoted as $X(q,k)$.

Algorithm HKNN (q,X,λ,k). /* q be the query sample, X be the training samples, k be the neighborhood size for classification, and λ be the penalty parameter*/

Step 1: Select k nearest neighbors for the query sample q from each class ω_j using Euclidean distance, denoted as $X_{\omega_j}(q,k)$.

Step 2: For each $X_{\omega_j}(q,k) = \{x_1, \ldots, x_i, \ldots, x_k\}$, we define the local hyperplane as

$$H^k_{\omega_j}(q) = \left\{ p \,|\, p = \overline{x} + \sum_{i=1}^{k} a_i V_i, a_i \in \Re \right\} \tag{1}$$

where $\overline{x} = \sum_{i=1}^{k} x_i/k$, and $V_i = x_i - \overline{x}$.

Step 3: Compute the k-local hyperplane distance by

$$d(q, H^k_{\omega_j}(q)) = \min_{p \in H^k_{\omega_j}(q)} \|q-p\| = \min_{a_i \in \Re} \left\| q - \overline{x} - \sum_{i=1}^{k} a_i V_i \right\| \tag{2}$$

where a_i can be solved through solving a linear system that can be easily expressed in matrix form as

$$(V' \cdot V) \cdot a = V' \cdot (q - \overline{x}) \tag{3}$$

where q and $\overline{x}$ be n dimensional column vectors, $a = (a_1, \ldots, a_k)'$ and V is an $n \times k$ matrix composed of column vectors V_i.

To penalize the large values of a_i, a penalty term λ is brought in. Then the k-local hyperplane distance can be redefined as

$$d(q, H^k_{\omega_j}(q)) = \min_{a_i \in \Re} \left\{ \left\| q - \overline{x} - \sum_{i=1}^{k} a_i V_i \right\|^2 + \lambda \sum_{i=1}^{k} a_i^2 \right\} \tag{4}$$

Step 4: Classify the query sample q to the class ω_j in terms of

$$\omega_j = \arg \min_{j \in \{1,2,\ldots,n_c\}} d(q, H^k_{\omega_j}(q))) \tag{5}$$

For any query point q, HKNN can find the closest neighbors, not among the training set, but among an abstract and virtually enriched training set that would contain all the fantasized "missing" points of the manifold of each class, locally approximated by an affine subspace. Noticeably, the performance of HKNN significantly depends on the value of the penalty term.

2.2. Relative transformation

To nicely perform the classification on the sparse or noisy data, it is naturally for machine classification to learn from human being. The existing classification approaches, such as those to recognizing digits and faces, require hundreds if not thousands of samples for training, while human visual recognition can be trained with very few samples [35]. This is because humans routinely classify objects according to both their individual attributes and membership in higher order groups, where individual attributes may be influenced and regulated by their group [36]. This can be illustrated in Fig. 1. When we observe the circle x, it looks bigger than its original size as it is surrounded by smaller circles. In contrast, when the circle y is observed, it appears smaller than its original size as it is surrounded by bigger circles. Consequently, when we observe x and y simultaneously, x is perceived to be bigger than y, although they are of the same sizes. This cognitive characteristic is very important for us to distinguish an object from its surrounding objects and can be then formalized using geometry

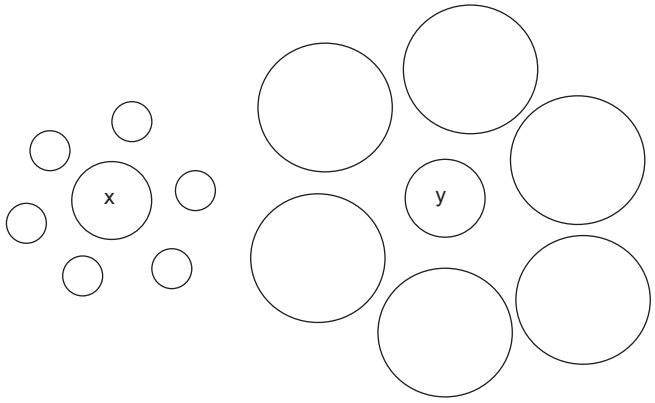


Fig. 1. Human visual perception is relative.

model to process the data more efficiently. One way is to define a transformation on the original space to build a new space whose dimensions are composed of all points in the original space. The newly created space is called the relative space and can be generated through relative transformation:

$$f^r : X \to Y \subset R^n$$

$y_i = f^r(x_i) = (d_{i1}, \ldots, d_{ij}, \ldots, d_{in}) \in Y, d_{ij} = |x_i - x_j|$, where n is the number of elements in data set X, the point x_i in the original space is mapped to the point $y_i \in R^n$ in the relative space, and $|\cdot|$ is the distance norm.

Algorithm $Y = f^r(X)$. /*X be the original space as input and Y be the relative space as output */

Step 1: Select the sample $x \in X$, we calculate the distances between it and any element $x_i \in X$ where distance can be Euclidean distance:

$$d(x,x_i) = |x - x_i|, \quad i = 1 \ldots |X|$$

Step 2: Map the sample $x \in X$ to the point $y \in Y$ in the relative space by the following way:

$$y = (y_1, \ldots, y_i, \ldots, y_{|X|}) \in R^{|X|}, \quad y_i = d(x,x_i),\ i = 1 \ldots |X|$$

Step 3: Repeat the above two steps until all samples in X are mapped to the relative space Y.

Relative transformation can make the data more distinguishable [52]. Some data can be distinguishable in the relative space while they cannot be identified in the original space. The relative transformation is also simple and efficient in dealing with noisy data or outliers, as shown in Fig. 2. In the original space the point $x4$ can be regarded as a noisy point or an outlier since it is far away from the other three points. However, we have $d(x_3,x_1) = d(x_3,x_4)$ in the original space. This means that the point $x4$ has the same possibility with the point $x1$ to be taken as a nearest neighbor of the point $x3$. This is not in line with our intuition. In the relative space, $d(y_3,y_1) < d(y_3,y_4)$, the outlier or noisy point becomes further away from the normal points. In such case, it can be recognized easily. Furthermore, it may also make points which originally lie on the same surface of the manifold closer to each other and points that are from the different surfaces further away from each other. This is especially useful to the sparse data. Finally, this approach has a simple mathematical basis and it allows a compact mathematical description of arbitrarily shaped neighborhood in the original space.

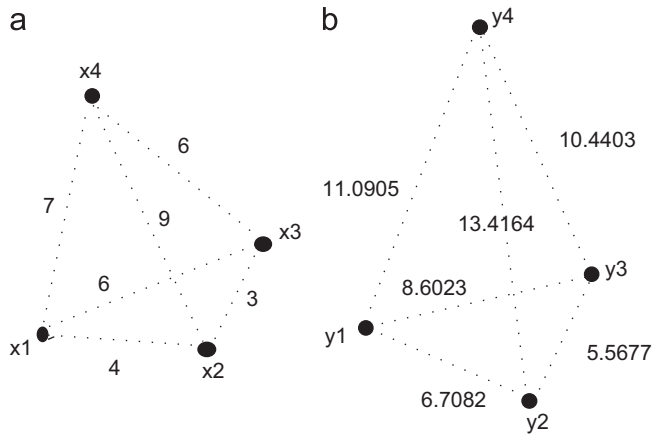


Fig. 2. Function of the relative transformation on noisy data, where (a) is the original space and (b) is the relative space.

3. Proposed new classifier

This section presents a new classifier that applies the relative transformation to HKNN, called RLHC. It finds k nearest neighbors for the query sample from each class and then performs the relative transformation over all these nearest neighbors to build the relative space. Subsequently, each local hyperplane is constructed in the relative space. The class label of the query is assigned according to the distance between the query and the local hyperplane of each class.

Algorithm RLHC (q,X,λ,k). /* q be a query sample, X be the training samples, k be the neighborhood size for classification, and λ be the penalty parameter*/

Step 1: Select k nearest neighbors for the query sample q from each class ω_j using Euclidean distance, denoted as $X_{\omega_j}(q,k)$

Step 2: Build the local region of the query sample by

$$X(q,k) = \bigcup_{1 \le j \le n_c} X_{\omega_j}(q,k)$$

Step 3: Build the relative space by

$$X^r(q,k) = f^r(X(q,k) \cup \{q\})$$

Step 4: For each $X^r_{\omega_j}(q,k) = \{y_1, \ldots, y_i, \ldots, y_k\} \subseteq X^r(q,k)$, we define the local hyperplane as

$$H^k_{\omega_j}(q) = \left\{ p \,|\, p = \overline{y} + \sum_{i=1}^{k} a_i V_i, a_i \in \Re \right\} \tag{6}$$

where $\overline{y} = \sum_{i=1}^{k} y_i/k$, and $V_i = y_i - \overline{y}$.

Step 5: Compute the k-local hyperplane distance by

$$d(q,H^k_{\omega_j}(q)) = \min_{p \in H^k_{\omega_j}(q)} \|q-p\| = \min_{a_i \in \Re} \left\| q - \overline{y} - \sum_{i=1}^{k} a_i V_i \right\| \tag{7}$$

where a_i can be solved through solving a linear system that can be easily expressed in matrix form as

$$(V' \cdot V) \cdot a = V' \cdot (q - \overline{y}) \tag{8}$$

where q and $\overline{y}$ be n dimensional column vectors, $a = (a_1, \ldots, a_k)'$ and V is an $n \times k$ matrix composed of column vectors V_i. To penalize the large values of a_i, a penalty term λ is brought in. Then

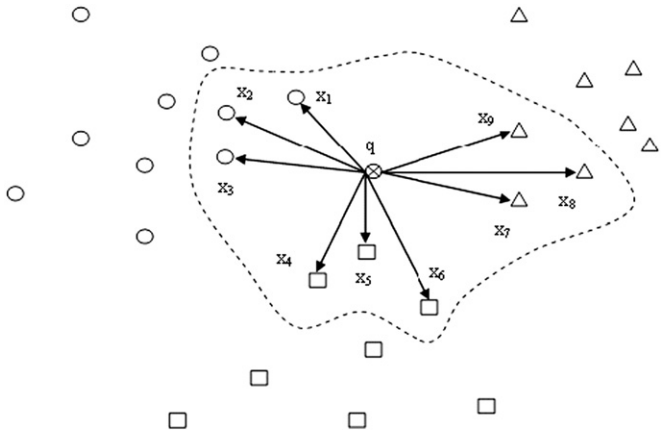


Fig. 3. Relative transformation used in RLHC.

the k local hyperplane distance can be redefined as

$$d(q,H^k_{\omega_j}(q)) = \min_{a_i \in \Re}\left\{\left\|q-\bar{y}-\sum_{i=1}^{k} a_i V_i\right\|^2 + \lambda\sum_{i=1}^{k} a_i^2\right\} \tag{9}$$

Step 6: Classify the query q to the class ω_j in terms of

$$\omega_j = \arg\min_{j\in\{1,2,\ldots,n_c\}} d(q,H^k_{\omega_j}(q))) \tag{10}$$

RLHC differs from HKNN in that it applies the relative transformation to build the relative local hyperplane for the query sample. The other steps keep unchanged. The idea of RLHC can be illustrated by Fig. 3.

Firstly, k nearest neighbors, where $k=3$, from each class are selected using Euclidean distance as follows: $X_{cir}(q,3)=\{x_1,x_2,x_3\}$, $X_{rec}(q,3)=\{x_4,x_5,x_6\}$, $X_{tri}(q,3)=\{x_7,x_8,x_9\}$. From these neighborhoods, the local region and the relative space are constructed as follows:

$$\begin{aligned} X(q,3) &= \bigcup_{\omega\in\{cir,tri,rec\}} X_\omega(q,3) \\ &= \{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9\} \end{aligned}$$

$$\begin{aligned} X^r(q,k) &= f^r(X(q,k)\cup\{q\}) \\ &= f^r(\{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,q\}) \end{aligned}$$

RLHC does not add any parameter and has the same complexity as that of HKNN to perform the classification for the query sample.This is because that the relative transformation is performed locally in $O(k^2)$ for each query sample, where k can be regarded as a constant.

4. Experimental results

4.1. Experimental setup

To validate RLHC on the classification performance, we compare it with the baseline approaches through experiments on benchmark data sets. These baseline approaches are KNN, FKNN [39], EKNN [38], LMC [31], LPC [2], HKNN [21,22], and ALH [37]. As it is empirically validated that ALH outperforms many other classifiers [37], including LDA, SVM, NFL, NNL, and CNN, these classifiers are not compared in this experiments.

In experiments, the error rate is taken as the measure of the performance [2,31]. The parameter k takes the value from $\{3,6,9,\ldots,30\}$ while the kernel parameter γ for LPC takes the values from $\{0.1,0.2,\ldots,0.9\}$. The parameter λ for HKNN and RLHC takes the values from $\{0.1,0.2,\ldots,0.9\}$. As T is for ALH uniquely and this is no experience to define the range, we take the values from $1:0,1.2,1.4,\ldots,3.0$ according to Fig. 2 in [37]. Euclidean distance is taken in all compared classifiers. Because k-fold cross validations is the most popular method to measure the quality of the classifier [40], we perform 5 times five-fold cross validations on each data. On each partition, the parameters are determined for each classifier through five-fold cross validations on the training samples, and then applied to perform the classification over the testing samples. Finally the average error rate of each method is reported.

4.2. On artificial data sets

Using artificial data, we can control the number and the dimension of the available samples and can add noise according to the experimental purpose. In the experiments, three artificial data sets are used. The first artificial data is ring norm that can be regarded as a benchmark data [42]. It is a 20-dimensional 2 class classification example. Each class is drawn from a multivariate normal distribution. Class 1 has mean zero and covariance 4 times the identity. Class 2 has mean (2/sqrt(20), 2/sqrt(20), …, 2/sqrt(20)) and unit covariance. The second one is two-spiral-patterns data representing a difficult problem that has been considered as a benchmark in pattern classification [41]. The data is represented in Fig. 4. The third data is *p-dimensional norm data* that has two classes [2]. Class 1 is represented by a multivariate normal distribution with zero mean and a standard deviation which is equal to one in all dimensions, whereas class 2 is represented by a normal distribution with zero mean and a standard deviation which is equal to two in all dimensions. Clearly, the classes are linearly nonseparable and the optimal decision boundary is quadratic. This data is usually referred to the hard task and taken as benchmarks to evaluate the performance of classifiers.

Generally the performance of a classifier is severely influenced by the outliers or noisy samples, particularly in small training sample size situations [31]. To compare RLHC with baseline approaches on noisy data, we perform the experiments on two benchmark artificial data sets: two spiral pattern data [41] and ring norm data [42]. Each data is appended with different random Gaussian noises. It can be observed from Fig. 5 that RLHC performs

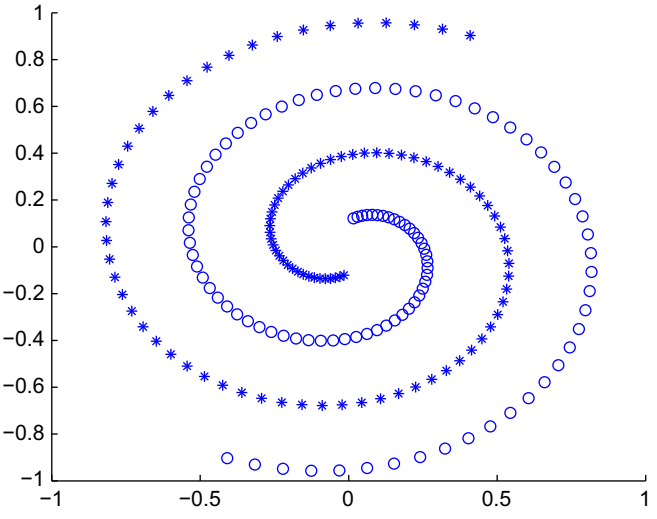


Fig. 4. Two spiral pattern data.

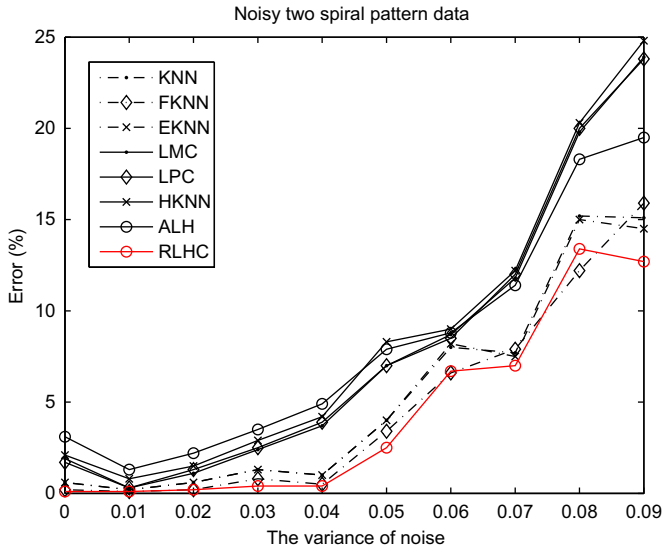


Fig. 5. Classification errors against noise levels.

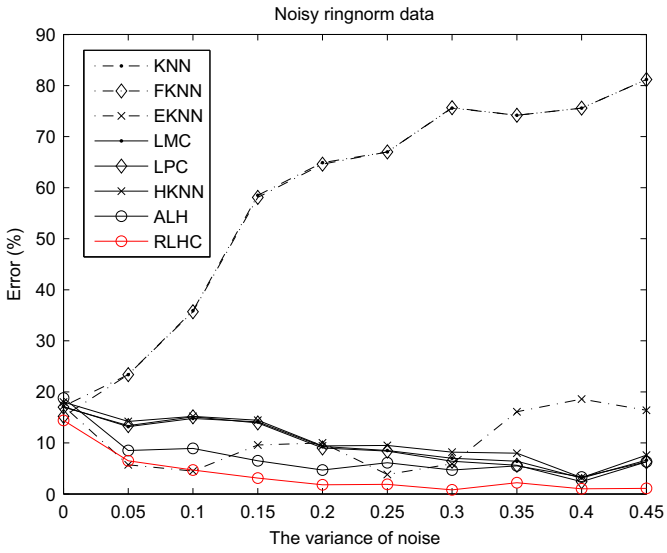


Fig. 6. Classification errors against noise levels.

best on two spiral pattern data. Its average classification error rate is 4.35%, followed by FKNN with the average classification error rate 4.78%. HKNN is the worst performer whose average classification error rate is 8.61%. On ring norm data, shown as Fig. 6, RLHC performs best at many noise cases, whose average classification error is 3.74%, followed by ALH with the average classification error 7.33%. FKNN becomes the much bad performer on this data whose average classification error rate is up to 57.07%, much different from that it performs on two spiral pattern data, illustrating FKNN is much sensitive to the data. The average classification error of HKNN is 10.77%, more than RLHC over 7.03%. These experimental results mean that RLHC is strongest to resist in noise disturbance.

It is well known that the curse of the dimensionality is a hard issue for pattern recognition, as in high dimensional data there may be redundant dimensions and exists a high degree of correlation among these dimensions. This may be a severe drawback to classifiers when dealing with small high dimensional data where the data may be sparsely distributed. To validate RLHC with the better ability to deal with the curse of the dimensionality problem, we do experiments on ring norm data [42] and p-dimensional norm data [2]. It can be observed from Fig. 7 that RLHC performs best on ring norm data with the average classification error 6.36%, followed by ALH with the average classification error 21.84%. The average classification error of HKNN is 43.30%, more than RLHC over 36.94%. As the dimension increases, the average classification error of RLHC decreases while that of KNN and FKNN go up quickly.

On p-dimensional norm data, shown as Fig. 8, RLHC performs best at any dimension, whose average classification error is 4.05%, followed by EKNN with average classification error 6.02%. The average classification error of HKNN is 13.48%, more than RLHC over 9.43%. These experimental results suggest that RLHC may be more robust to the dimensionality and shows a favorable behavior in high dimensional data spaces.

4.3. On real data sets

While simulated data are informative for comparison studies, it is highly likely that artificially constructed examples will not correspond to situations that are likely to occur in practice. Thus, in this section, we examine the performance of the competing

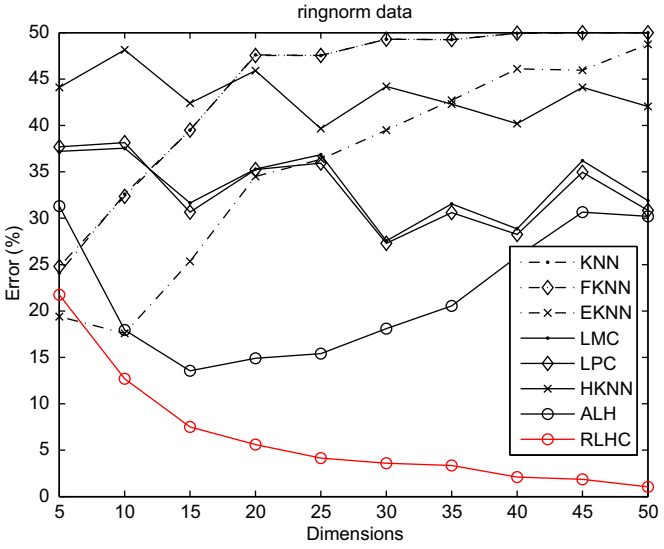


Fig. 7. Classification errors against data dimensions.

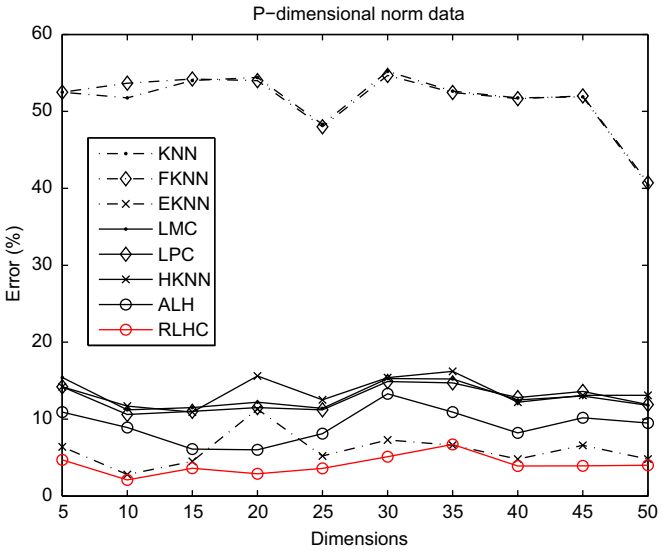


Fig. 8. Classification errors against data dimensions.

classification methods using real data. One of the advantages of real data is that they are generated without any knowledge of the classification procedures that it will be used to test. In our experiments we used eight different real data sets. These data sets are all taken from the UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html), illustrated as Table 1. It can be observed from Table 2 that RLHC outperforms all compared classifiers on the average of classification errors in eight real data sets. The average standard deviation of the errors of RLHC is also the smallest, indicating that its performance is most stable. No classifier can perform best in all data sets. RLHC performs much best in 2/8 data sets with large gap to the second best classifier, while it outperforms HKNN in 5/8 data sets, illustrating the significance of the relative transformation to the classification. It can be also observed that HKNN performs better than RLHC on the rest three data sets, but with much small gap.

Generally every method has its strengths and weaknesses. It is necessary to apply a measure to evaluate the robustness of the different methods. We take the usually used measure to quantify the robustness by computing the ratio b_m of the error rate e_m of method m and the smallest error rate over all methods being compared in a particular data set: $b_m = e_m / \min_{1 \le k \le 10} e_k$ [1,2,14]. Thus, the best method m^* for that problem has $b_m = 1$, and all other methods have larger values $b_m > 1$. The larger the value of b_m, the worse the performance of the method. This means that the distribution of b_m will be a good indicator reflecting its robustness. Fig. 9 shows the distribution of b_m for each method over the 10 real data sets, which is drawn using matlab function: boxplot. Clearly, the spread for RLHC is much narrower and closer to one. This result demonstrates that it obtains the most robust performance over these data sets.

4.4. Parameter analysis

As there is no structured way to choose the optimal parameters, it is expected that the proposed RLHC can be less sensitive to the choice of their parameters. As the neighborhood size k is common for all compared classifiers, we investigate how this parameter affects the average classification error rate through experiments on four real data sets. For the purpose of clarity, here, we only give performance curves of HKNN, ALH, and RLHC, as they share the similar idea and have better performance over the other compared classifiers as illustrated as b_m. It can be observed from Figs. 10 to 13 that RLHC does not vary drastically against the neighborhood sizes with a high confidence, showing its best stability. It surpasses other approaches at many values of the neighborhood size. This means that selection of k for RLHC is easier than that for the baseline classifiers.

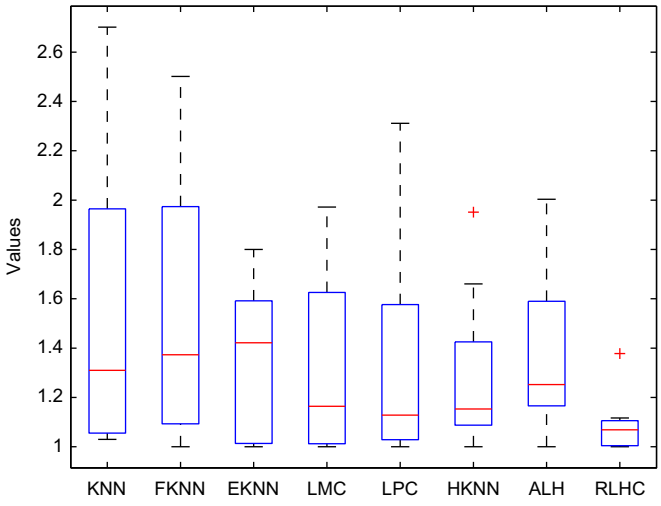


Fig. 9. Average performance distribution of different classifiers.

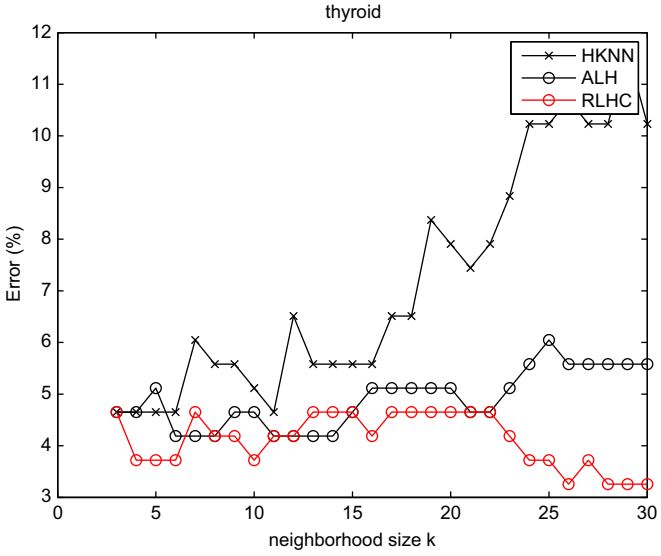


Fig. 10. Classification errors against neighborhood sizes.

Table 1
Data sets used in experiments.

No.	Problem	Attributes	Classes	Size
1	Wine	13	3	178
2	Dermatology	34	6	358
3	Diabetes	8	2	768
4	Ionosphere	34	2	351
5	Glass	9	7	214
6	Optdigits	64	10	1797
7	Segmentation	19	7	210
8	Yeast	8	10	1484

Table 2
Average classification errors (%) for real data.

No.	KNN	FKNN	EKNN	LMC	LPC	HKNN	ALH	RLHC
1	3.92 ± 1.50	3.95 ± 1.61	2.78 ± 1.91	3.32 ± 2.94	3.89 ± 3.10	*2.80 ± 1.91*	**1.68 ± 1.54**	*1.70 ± 1.55*
2	3.08 ± 2.52	3.36 ± 2.57	3.36 ± 2.37	3.35 ± 1.23	2.79 ± 0.98	**2.23 ± 1.58**	4.46 ± 2.48	3.07 ± 1.82
3	24.74 ± 3.86	25.65 ± 3.73	23.31 ± 4.23	**23.31 ± 4.15**	23.83 ± 3.52	25.91 ± 3.10	28.00 ± 1.66	26.04 ± 3.46
4	15.37 ± 5.87	14.23 ± 5.00	10.24 ± 4.72	9.95 ± 5.73	10.81 ± 5.43	11.10 ± 2.28	10.81 ± 4.27	**5.69 ± 2.22**
5	27.99 ± 5.42	28.94 ± 5.45	26.99 ± 5.53	33.64 ± 6.26	32.25 ± 8.36	31.27 ± 5.57	34.14 ± 3.36	**26.67 ± 3.84**
6	1.33 ± 0.45	1.33 ± 0.45	1.28 ± 0.50	0.89 ± 0.36	0.83 ± 0.48	0.95 ± 0.32	1.06 ± 0.36	0.89 ± 0.53
7	12.38 ± 5.43	12.38 ± 3.91	13.33 ± 5.22	**10.00 ± 4.58**	10.48 ± 3.19	11.90 ± 3.76	12.38 ± 4.26	10.95 ± 1.30
8	41.78 ± 1.82	**40.57 ± 1.67**	41.17 ± 3.12	41.58 ± 2.72	41.98 ± 2.96	43.13 ± 1.50	45.89 ± 2.64	43.46 ± 2.75
Avg.	16.33 ± 3.36	16.30 ± 3.05	15.31 ± 3.45	15.75 ± 3.50	15.86 ± 3.50	16.16 ± 2.50	17.30 ± 2.57	**14.81 ± 2.18**

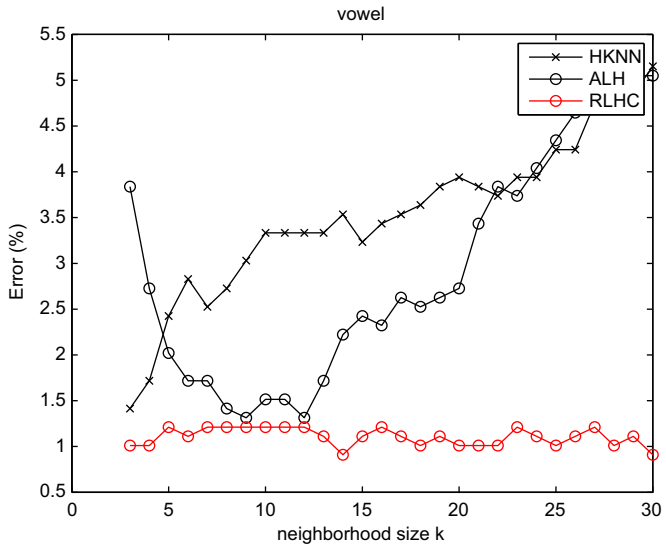


Fig. 11. Classification errors against neighborhood sizes.

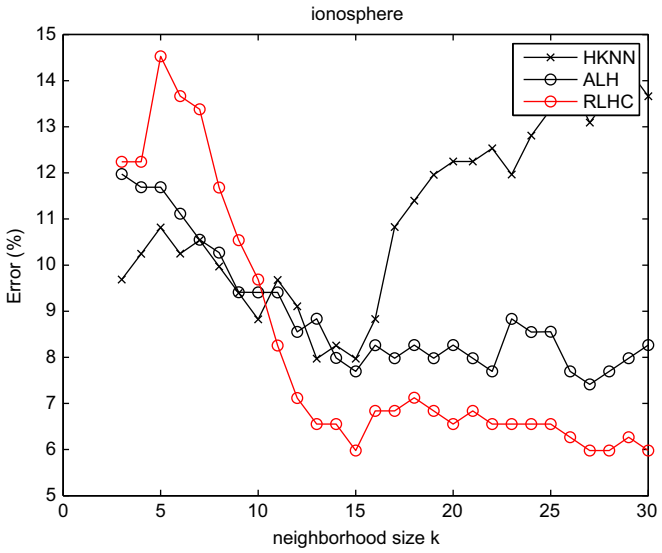


Fig. 13. Classification errors against neighborhood sizes.

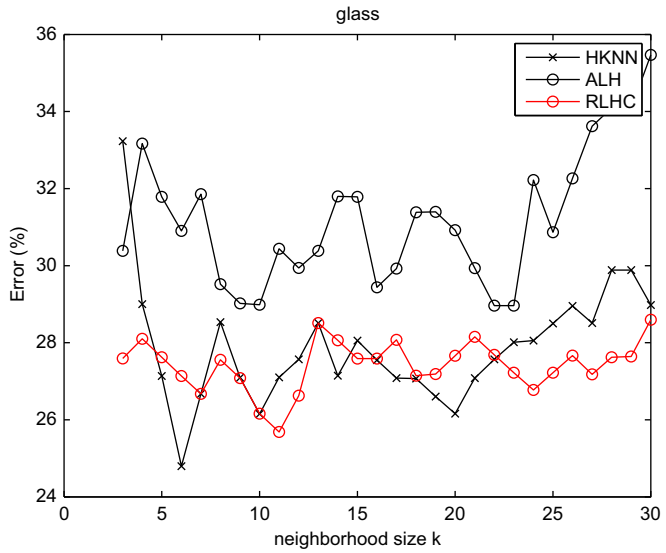


Fig. 12. Classification errors against neighborhood sizes.

5. Related work

RLHC takes the similar principle to those local manifold approaches, including nearest feature line (NFL) [20], nearest feature space (NFS) method is proposed [55], k-local hyperplane distance nearest neighbor (HKNN) [21], nearest neighbor line (NNL) [24], and center-based nearest neighbor (CNN) [25]. Both NFL and NFS metrics have been used in the classification phase and in the feature extraction [26–28]. HKNN shares the similar idea with NFS but perform very well in a variety of applications [33,22,23]. RLHC is directly enhanced from HKNN.

RLHC also shares the similar decision rules with other approaches, such as local mean classifier (LMC) [31], local probability center (LPC) [2], and HKNN. HKNN has been extended by the feature weight [37] and the kernel technique [43]. RLHC extends HKNN using the relative transformation. The relative transformation is a newly proposed model based on the idea that perception is relative, which has been validated in manifold learning [52] and extended by the kernel technique [45], high order techniques [44], as well as geodesic distance [46]. However they are only applied to enhance the manifold learning, instead of classification. Besides, some other feature space transformation are also proposed for clustering [57], which can be further investigated to perform the classification.

Many perceptual laws can account for the amazing ability of human, as illustrated in topological invariants [47], Gestalt theory of perception [49] and topological psychology [48]. In machine vision, some computational theories apply these perceptual laws to solve the problems such as to find Gestalts in digital images [50], to perform object grouping on SAR images by Gestalt laws [54], or to apply Gestalt law of continuity to perform visual structure inference [53]. They used specific properties of images, instead of taking each image as independent objects for classification. Another novel work is the proposal of principle of homology continuity [51]. This approach is important but not for designing nearest neighbor classifiers. As a matter of fact, to apply Gestalt laws to perform classification on independent objects is a much significant task [56].

6. Conclusion

To nicely perform classification on the noisy, the sparse, and the imbalance data, this paper presents a new classifier that integrates the relative transformation with the local manifold-based classifier. We showed that this approach achieves error rates less than those local classifiers, while it does not increase the time complexity, as where the relative transformation is performed locally. This approach is also robust to the parameters and does not need any additional parameter, so that it is more practical and has wider applications. Further research is warranted in order to gain a better insight into the method's theoretical properties and to investigate its performance in different cases where the different relative transformation can be defined and applied properly.

Acknowledgment

This work was supported by China National Science Foundation under Grants 60973083, 61003174, State Key Laboratory of Brain and Cognitive Science under Grant 08B12, program for New Century

Excellent Talents in University (Project No. NCET-11-0165), Natural Science Foundation of Guangdong Province (Project No. S2011010000264 and 10451064101004233), and Fundamental Research Funds for the Central Universities, SCUT.

References

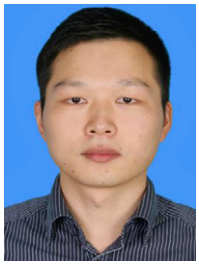
[1] C. Domeniconi, J. Peng, D. Gunopulos, Locally adaptive metric nearest-neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1281–1285.
[2] B. Li, Y.W. Chen, Y.Q. Chen, The nearest neighbor algorithm of local probability centers, IEEE Trans. Syst. Man Cybern. 38 (1) (2008) 141–153.
[3] S.N. Karl, Q.N. Truong, An adaptable k-nearest neighbors algorithm for MMSE image interpolation, IEEE Trans. Image Process. 18 (9) (2009) 1976–1987.
[4] M.H. Tristan, R. Stephane, Tailored aggregation for classification, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2009) 2098–2105.
[5] D. Ramanan, S. Baker, Local distance functions: a taxonomy,new algorithms, and an evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (4) (2011) 794–806.
[6] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory (1967) 21–27.
[7] L. Devroye, L. Gyorfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, Springer-Verlag, 1996, pp. 71–75.
[8] E. Blanzieri, F. Melgan, Nearest neighbor classification of remote sensing images with the maximal margin principle, IEEE Trans. Geosci. Remote Sensing 46 (6) (2008) 1804–1811.
[9] G. Guo, C.R. Dyer, Learning from examples in the small sample case: face expression recognition, IEEE Trans. Syst. Man Cybern. 35 (2005) 477–488.
[10] B. Zhang, S.N. Srihari, Fast k-nearest neighbor classification using cluster-based trees, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 525–528.
[11] C. Zhou, Y. Yan, Q. Chen, Improving nearest neighbor classification with cam weighted distance, Pattern Recognition 39 (2006) 635–645.
[12] C. Shen, J. Kim, L. Wang, Scalable large-margin Mahalanobis distance metric learning, IEEE Trans. Neural Networks 21 (9) (2010) 1524–1532.
[13] H. Altincay, Improving the k-nearest neighbour rule: using geometrical neighbourhoods and manifold-based metrics, Expert Syst. (2010) 1–16.
[14] J. Peng, D. Heisterkamp, H.K. Dai, LDA/SVM driven nearest neighbor classification, IEEE Trans. Neural Network 14 (4) (2003) 940–942.
[15] G. Wen, L. Jiang, N.R. Shadbolt, Using graph algebra to optimize neighborhood for isometric mapping, in: Proceedings of 20th International Joint Conference on Artificial Intelligence (IJCAI-07), India, 2007.
[16] G. Wen, L. Jiang, J. Wen, Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding, Pattern Recognition 41 (7) (2008) 2226–2236.
[17] J. Peng, D.R. Heisterkamp, H.K. Dai, Adaptive quasiconformal kernel nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 656–661.
[18] H. Wang, Nearest neighbors by neighborhood counting, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 942–953.
[19] L. Peng, B. Yang, Y. Chen, A. Abraham, Data gravitation based classification, Inf. Sci. (179) (2009) 809–819.
[20] S.Z. Li, J.W. Lu, Face recognition using the nearest feature line method, IEEE Trans. Neural Networks 10 (2) (1999) 439–443.
[21] P. Vincent, Y. Bengio, K-local hyperplane and convex distance nearest neighbour algorithms, Adv. Neural Inf. Process. Syst. 14 (2002) 985–992.
[22] O.G. Okun, K-local hyperplane distance nearest neighbor algorithm and protein fold recognition, Pattern Recognition Image Anal. 17 (4) (2007) 621–630.
[23] V. Kecman, T. Yang, Adaptive local hyperplane for regression tasks, in: Proceedings of the International Joint Conference on Neural Networks, 2009, pp. 1566–1570.
[24] W. Zheng, L. Zhao, C. Zou, Locally nearest neighbor classifiers for pattern classification, Pattern Recognition 37 (2004) 1307–1309.
[25] Q.B. Gao, Z.Z. Wang, Center-based nearest neighbor classifier, Pattern Recognition 40 (2007) 346–349.
[26] Y. Pang, Y. Yuan, X. Li, Generalized nearest feature line for feature extraction, Electron. Lett. 43 (20) (2007) 1079–1080.
[27] J. Lu, Y.P. Tan, Uncorrelated discriminant nearest feature line analysis for face recognition, IEEE Signal Process. Lett. 17 (2) (2010) 185–188.
[28] J. Lu, Y.-P. Tan, Nearest feature space analysis for classification, IEEE Signal Process. Lett. 18 (1) (2011) 51–58.
[29] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: Proceedings of the 15th European Conference on Machine Learning (ECML), Pisa, Italy, 2004, pp. 39–50.
[30] Y.-H. Liu, Y.-T. Chen, Face recognition using total margin-based adaptive fuzzy support vector machines, IEEE Trans. Neural Network 18 (1) (2007) 178–192.
[31] Y. Mitani, Y. Hamamoto, A local mean-based nonparametric classifier, Pattern Recognition Lett. 27 (10) (2006) 1151–1159.
[32] H. Zhu, O. Basir, An adaptive fuzzy evidential nearest neighbor formulation for classifying remote sensing images, IEEE Trans. Geosci. Remote Sensing 43 (8) (2005) 1874–1889.
[33] L. Nanni, A. Lumini, An ensemble of K-local hyperplanes for predicting protein–protein interactions, Bioinformatics 22 (2006) 1207–1210.
[34] I. Triguero, S. Garcia, F. Herrera, IPADE: iterative prototype adjustment for nearest neighbor classification, IEEE Trans. Neural Networks 21 (12) (2010) 1984–1991.
[35] H. Zhang, A.C. Berg, M. Maire, J. Malik, SVM-KNN: discriminative nearest neighbor classification for visual category recognition, in: Proceeding of conference on computer vision and pattern recognition (CVPR 2006), IEEE Computer Society, New York, USA, 2006, pp. 2126–2136.
[36] T.J. Bergman, J.C. Beehner, D.L. Cheney, R.M. Seyfarth, Hierarchical classification by rank and kinship in baboons, Science 302 (5648) (2003) 1234–1236.
[37] T. Yang, V. Kecman, Adaptive local hyperplane classification, Neurocomputing 71 (2008) 3001–3004.
[38] T. Denoex, A k-nearest neighbor classification rule based on Dempster–Shafer theory, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 804–813.
[39] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy K-nearest neighbor algorithm, IEEE Trans. Syst. Man Cybern. 15 (4) (1985) 580–585.
[40] J.D. Rodriguez, A. Perez, J.A. Lozano, Sensitivity analysis of k-fold cross validation in prediction error estimation, IEEE Trans. Pattern Anal. Mach. Intell. 32 (3) (2010) 569–575.
[41] S. Singh, 2D spiral pattern recognition with possibilistic measure, Pattern Recognition Lett. 19 (1998) 131.
[42] L. Breiman, Arcing classifiers, Ann. Statist. 26 (1998) 801.
[43] Q. Ni, Z. Wang, X. Wang, Kernel K-local hyperplanes for predicting protein–protein interactions, in: Fourth International Conference on Natural Computation, 2008, pp. 66–69.
[44] G. Wen, H. Liu, L. Jiang, High order relative transformation for isometric embedding, J. Comput. Inf. Syst. 4 (6) (2008) 2447–2454.
[45] G. Wen, L. Jiang, J. Wen, Kernel relative transformation with applications to enhancing locally linear embedding, in: Proceedings of the International Joint Conference on Neural Networks, 2008, pp. 3401–3406.
[46] G. Wen, T. Lu, L. Jiang, J. Wen, Locally linear embedding based on relative manifold, J. Software 20 (9) (2009) 2376–2386.
[47] L. Chen, Visual Cognit. 12 (2005) 553.
[48] K. Lewin, Principles of Topological Pyschology, McGraw-Hill Book Company, Inc., 1936.
[49] K. Koffka, Principles of Gestalt Psychology, Routledge, London, 1955.
[50] A. Desolneux, L. Moisan, J. Morel, Computational gestalts and perception thresholds, J. Physiol. Paris 97 (2003) 311.
[51] S. Wang, X. Chen, Biomimetic (topological) pattern recognition—a new model of pattern recognition theory and its application, in: Proceedings of the International Joint Conference on Neural Network, 2003, p. 2258.
[52] G. Wen, et al., Relative transformation-based neighborhood optimization for isometric embedding, Neurocomputing 72 (4–6) (2009) 1205–1213.
[53] O. Ben-Shahar, S.W. Zucker, General geometric good continuation: from Taylor to Laplace via level sets, Int. J. Comput. Vis. 86 (2010) 48–71.
[54] E. Michaelsen, U. Stilla, U. Soergel, L. Doktorski, Extraction of building polygons from SAR images: Grouping and decision-level in the GESTALT system, Pattern Recognition Lett. 31 (2010) 1071–1076.
[55] J.T. Chien, C.C. Wu, Discriminant waveletfaces and nearest feature classifiers for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1644–1649.
[56] G. Wen, Modeling Gestalt laws for classification, in: Proceedings of the 9th IEEE International Conference on Cognitive Informatics, China, 2010.
[57] J. Wu, L. Al Nimer, O. Al Azzam, C. Chitraranjan, S. Salem, A.M. Denton, Sequential data clustering, in: ICMLA, 2010, pp. 1015–1020.

Wen Guihua, born in 1968, Ph.D., professor, doctor supervisor. In 2005–2006, he did visiting research on machine learning and semantic web in School of Electronics and Computer, University of Southampton, UK. His main research interests are computational creativity, data mining and knowledge discovery, machine learning, and cognitive geometry. Since 2006, he proposed some original methods based on the computation of cognitive laws, which can effectively solve difficult problems in information science. The research results have been published in the international journals, including Pattern Recognition (2008), Neurocomputing (2008), Journal of Software (2008, 2009), Journal of computer Research and Development (2008). He also published some papers in the international conferences such as IJCAI (2007).

Since 2006, he directed the projects from the China National Natural Science Foundation, State Key Laboratory of Brain and Cognitive Science, the Ministry of Education Scientific Research Foundation for returned overseas students, Guangdong Provincial Science and Technology research project, the Fundamental Research Funds for the Central Universities, SCUT. He also directed many projects from enterprises, with applications of his research results to the practical problems. He has ever been a Council Member of Chinese Association for Artificial Intelligence and a program committee member of many international conferences. He is also a reviewer for China National Natural Science Foundation.

Lijun Jiang is now a lecturer at School of Materials Science and Engineering, South China University of Technology. Her research fields include creative education and intelligent CAD.

Jun Wen is an associate professor in Hubei Institute for Nationalities. His research fields include creativity computation, machine learning, and intelligent software.



Jia Wei received his B.Sc. and M.Sc. degrees both in Computer Science from Harbin Institute of Technology in 2003 and 2006, respectively. He received his Ph.D. degree in Computer Science from South China University of Technology in 2009. He is now a lecturer at the School of Computer Science and Engineering, South China University of Technology. His research fields include machine learning and artificial intelligence.

Zhiwen Yu is an associate professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received the B.Sc. and M.Phil. degrees from the Sun Yat-Sen University in China in 2001 and 2004 respectively, and the Ph.D. degree in Computer Science from the City University of Hong Kong, in 2008. His research interests include pattern recognition, bioinformatics, multimedia, intelligent computing and data mining. He has published more than 50 technical articles in referred journals and conference proceedings in the areas of pattern recognition, bioinformatics and multimedia.