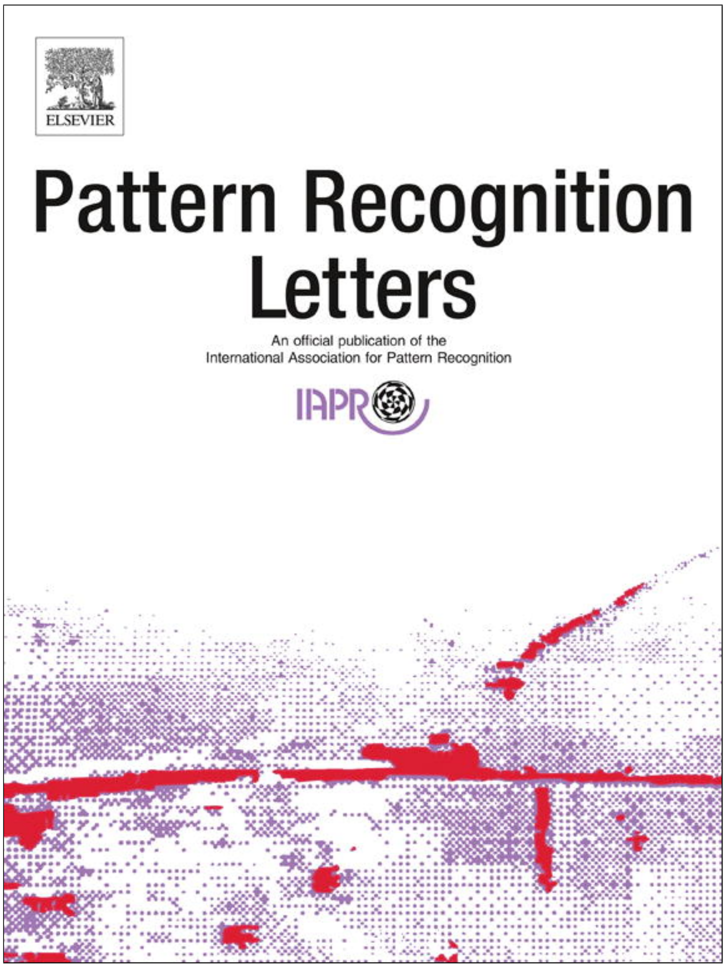
ELSEVIER
Pattern Recognition Letters
An official publication of the International Association for Pattern Recognition
IAPR

Pattern Recognition Letters 34 (2013) 980–986

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

Coarse to fine K nearest neighbor classifier

Yong Xu [a,b], Qi Zhu [a], Zizhu Fan [a,c], Minna Qiu [a], Yan Chen [a,*], Hong Liu [d]

[a] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, 518055 Shenzhen, China
[b] Key Laboratory of Network Oriented Intelligent Computation, Shenzhen, China
[c] School of Basic Science, East China Jiaotong University, Nanchang, Jiangxi, China
[d] Key Laboratory of Machine Perception and Intelligence, Shenzhen Graduate School, Peking University, Shenzhen, China

ARTICLE INFO

Article history:
Received 20 January 2012
Available online 16 February 2013
Communicated by Prof. S. Sarkar

Keywords:
Nearest neighbor classifier
Classification
Biometrics
Face recognition
Palmprint recognition

ABSTRACT

In this paper, we propose a coarse to fine K nearest neighbor (KNN) classifier (CFKNNC). CFKNNC differs from the conventional KNN classifier (CKNNC) as follows: CFKNNC first coarsely determines a small number of training samples that are "close" to the test sample and then finely identifies the K nearest neighbors of the test sample. The main difference between CFKNNC and CKNNC is that they exploit the "representation-based distances" and Euclidean distances to determine the nearest neighbors of the test sample from the set of training samples, respectively. The analysis shows that the "representation-based distances" are able to take into account the dependent relationship between different training samples. Actually, the nearest neighbors determined by the proposed method are optimal from the point of view of representing the test sample. Moreover, the nearest neighbors obtained using our method contain less redundant information than those obtained using CKNNC. The experimental results show that CFKNNC can classify much more accurately than CKNNC and various improvements to CKNNC such as the nearest feature line (NFL) classifier, the nearest feature space (NFS) classifier, nearest neighbor line classifier (NNLC) and center-based nearest neighbor classifier (CBNNC).

1. Introduction

The KNN classifier has been widely used in the fields of pattern classification and machine learning. For example, the KNN classifier has been applied to feature selection (Tahir et al., 2007) and dimensionality reduction (Villegas and Paredes, 2011). As we know, the conventional KNN classifier (CKNNC) (Cover and Hart, 1967) simply uses the K training samples that are closest to the test sample to classify it. Owing to the simplicity of the classification rule of CKNNC, many improvements have been proposed for achieving a higher accuracy.

As pointed out by Weinberger et al., the accuracy of K nearest neighbor (KNN) classification significantly depends on the metric used to compute distances between different samples (Weinberger et al., 2009; Weinberger et al., 2006). A number of metrics have been proposed to improve CKNNC, such as the Mahalanobis distance metric (Weinberger et al., 2009; Park et al., 2011), adaptive distance (Wang et al., 2007) and local metric (Noh et al., 2010). Li and Lu defined a distance between a test sample and a "line" constructed by two training samples from a same class and extended the nearest neighbor classifier to the nearest feature line (NFL) classifier (Li and Lu, 1999). NFL treats all points on feature lines which pass through each pair of training samples from the same class as virtual training samples. If a class has n training samples, then NFL obtains $n(n-1)/2$ feature lines for this class. Chien and Wu (Chien and Wu, 2002) devised the nearest feature plane (NFP) and the nearest feature space (NFS) classifiers, which used the distance between the test sample and a plane and the distance between the test sample and the space spanned by the training samples from a class, respectively. NFP classifies a test sample to a class that contains the nearest feature plane to the test sample and NFS classifies a test sample to a class whose feature space is closest to the test sample. NFL, NFP and NFS can be regarded as nearest neighbor classifiers with some constraints. A modification of NFL, center-based nearest neighbor classifier (CBNNC) calculated only the distances between the test sample and the line passing through a sample point with known label and the center of the sample class (Gao and Wang, 2007). Nearest neighbor line classifier (NNLC) is also a modification of NFL NNLC exploits only the feature line whose corresponding prototypes are the neighbors of the test sample and is more computationally efficient than NFL (Zheng et al., 2004). Moreover, some researchers changed the distance structure of samples by setting a weight to each feature component of the sample (Wang et al., 2007). For example, Hu et al. proposed an approach to learn sample weights for enlarging the margin by minimizing margin based classification loss (Hu et al., 2011).

Recently, the dynamic time warping constraint was also exploited to improve CKNNC (Yu et al., 2011). Besides the above improvements to CKNNC, a number of new and valuable

* Corresponding author.
E-mail address: laterfall2@yahoo.com.cn (Y. Chen).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.01.028

definitions of the nearest neighbor, such as meaningful nearest neighbor (MNN) (Omercevic et al., 2007), MaxNearestDist (Samet, 2008), probably correct k-nearest neighbor (PCKN) (Toyama et al., 2010) and distance lower bound based nearest neighbor (Chen et al., 2007) have also been proposed.

The satisfactory performance of coarse to fine classification has drawn much attention in many cases. For example, coarse to fine image classification (Descampe et al., 2011) and face recognition (Feng et al., 2004) have achieved a high accuracy. The coarse to fine strategy has also been applied to resolve other problems such as shape matching (Rube et al., 2004), object detection (Pedersoli et al., 2011) and textures retrieval (Descampe et al., 2006).

In this paper, an integration of coarse to fine classification and KNN classifier is proposed. The proposed classifier, CFKNNC, first selects a small number of training samples that are "similar" ("close") to the test sample from the original set of training samples. CFKNNC then computes a representation-based "distance" to determine K training samples that are the most "similar" to the test sample. Finally, CFKNNC classifies a test sample to the class which the most of the determined K training samples come from. The experimental results show that CFKNNC performs very well and obtain a much higher accuracy than CKNNC. CFKNNC also outperforms NFL, NFS, NNLC and CBNNC. CFKNNC has the following rationales: the first step of CFKNNC eliminates side-effects on classification of the training samples that are far from the test sample, so it will be easier to ultimately classify the test sample into one of the classes that are genuinely close to the test sample. Furthermore, the second step of CFKNNC allows the distance between the test sample and each of its K neighbors to be evaluated in a competitive way rather than in an independent way. The competitive way evaluates the distance more reliable. Our analysis also shows that the nearest neighbors obtained by our method contain less redundant information than those obtained by CKNNC. As a result, under the condition that our method and CKNNC determine a same number of nearest neighbors, the nearest neighbors obtained using our method is able to provide a better representation for the test sample than those obtained using CKNNC. In other words, as the nearest neighbors obtained by our method contain less redundant information, the weighted sum of the nearest neighbors obtained using our method has a greater potential to approximate the test sample than that of the nearest neighbors obtained using CKNNC.

The rest of the paper is organized as follows: Section 2 describes our method. Section 3 shows the rationales of the proposed method. Section 4 presents the experimental results and Section 5 offers our conclusion.

2. Coarse to fine K nearest neighbor classifier

Coarse to fine K nearest neighbor classifier (CFKNNC) exploits column vectors $X_1 \ldots X_N$ to stand for all N training samples. CFKNNC first coarsely determines n training samples that are closest to the test sample. CFKNNC then finely selects K ($K \leqslant n$) nearest neighbors of the test sample from the n training samples determined. Finally, CFKNNC uses the class labels of the K nearest neighbors to classify the test sample. If $X_1 \ldots X_N$ and test sample Y are not unit vectors with length of 1, CFKNNC will convert them into unit vectors in advance. CFKNNC tries to express test sample Y as a linear combination of all the training samples. In other words, CFKNNC assumes that the following equation is almost satisfied:

$$Y = \sum_{i=1}^{N} \gamma_i X_i. \tag{1}$$

Eq. (1) can be rewritten as

$$Y = X\gamma, \tag{2}$$

where $\gamma = (\gamma_1 \ldots \gamma_N)^T$, $X = (X_1 \ldots \quad X_N)$.

We exploit the Lagrangian algorithm to obtain the solution of Eq. (2). As we expect that $X\gamma$ best approximate Y, we require that γ minimizes $||Y - X\gamma||^2$. Moreover, theories of numerical analysis have proven that if n has a small norm, the solution of Eq. (2) can generalize well. Therefore, we also expect that $||\gamma||^2$ be as small as possible. For the above factors, we define a Lagrangian function $f(\gamma) = ||Y - X\gamma||^2 + \mu||\gamma||^2$, where μ is a positive constant. The optimal solution of Eq. (2) should minimize $f(\gamma)$ and satisfy $\frac{\partial f(\gamma)}{\partial \gamma} = 0$ as well. Because $\frac{\partial f(\gamma)}{\partial \gamma} = 0$ means $2(X^TX + \mu I)\gamma = 2X^TY$, As a result, we obtain the solution of Eq. (2) using $\hat{\gamma} = (X^TX + \mu I)^{-1}X^TY$, where I is the identity matrix. CFKNNC then calculates

$$e_i = \|Y - \hat{\gamma}_i X_i\|^2, \tag{3}$$

where $\hat{\gamma}_i$ stands for the i-th entry of $\hat{\gamma}$. CFKNNC selects n training samples that have the first n smallest e_i s and denotes them as $Z_1 \ldots Z_n$, respectively.CFKNNC then uses a weighted sum of $Z_1 \ldots Z_n$ shown as Eq. (4) to express test sample Y

$$Y = \sum_{i=1}^{n} w_i Z_i. \tag{4}$$

Eq. (4) can be rewritten as $Y = Zw$, where $w = [w_1 \ldots w_n]^T$ and $Z = [Z_1 \ldots Z_n]$. CFKNNC obtains the solution of Eq. (4) using $\hat{w} = (Z^TZ + \mu I)^{-1}Z^TY$, where μ is a positive constant and I is the identity matrix. CFKNNC views $d_i = \|Y - \hat{w}_i Z_i\|^2$ ($\hat{w}_i$ stands for the i-th entry of $\hat{w}$) as the similarity metric between Y and Z_i. It is regarded that a smaller d_i means a higher similarity between Y and Z_i. CFKNNC then selects K training samples that have the first K smallest d_i s from $Z_1 \ldots Z_n$ and denotes them as $s_1 \ldots s_K$, respectively.For $s_1 \ldots s_K$, CFKNNC counts the number of the training samples from the j-th ($j = 1, \ldots, C$) class. C is the number of all the classes. Let m_j be the number of the training samples from the j-th ($j = 1, \ldots, C$) class. It is clear that $K = \sum_{j=1}^{C} m_j$. If $t = \arg\max_j m_j$, then CFKNNC assigns the test sample Y into the t-th class.

3. Interpretation and rationale of the proposed method

In this section, we interpret the difference and relationship between CKNNC and the proposed method and show the rationale of the proposed method.

We show the flowcharts of CKNNC and the proposed method, CFKNNC, in Fig. 1. The main similarity between CKNNC and CFKNNC is that both of them classify a test sample by exploiting the labels of the K nearest neighbors of the test sample. Suppose that the class labels of these K nearest neighbors are $c_1, \ldots, c_K \in \{1, 2, \ldots, C\}$, respectively. If l is the class label that the most of these K nearest neighbors have, then both CKNNC and CFKNNC will classify the test sample into the l-th class.

As shown in Fig. 1, the main difference between CKNNC and CFKNNC is that CKNNC uses the Euclidean distance to determine the nearest neighbors of the test sample, whereas CFKNNC uses "representation-based distances" to do so. Moreover, the algorithm of CFKNNC seems to be more elaborate and has more steps than that of CKNNC.

As we know, CKNNC calculates the distances between the test sample and each training sample in an independent way. However, the method proposed in this paper calculates the distances in a dependent way. In order to simply interpret the above difference, we assume that there are only two training samples. If the second training sample changes and the first training sample does not do so, then in CKNNC the distance between the test sample and the second training sample also changes, whereas the distance between the test sample and the first training sample is the same

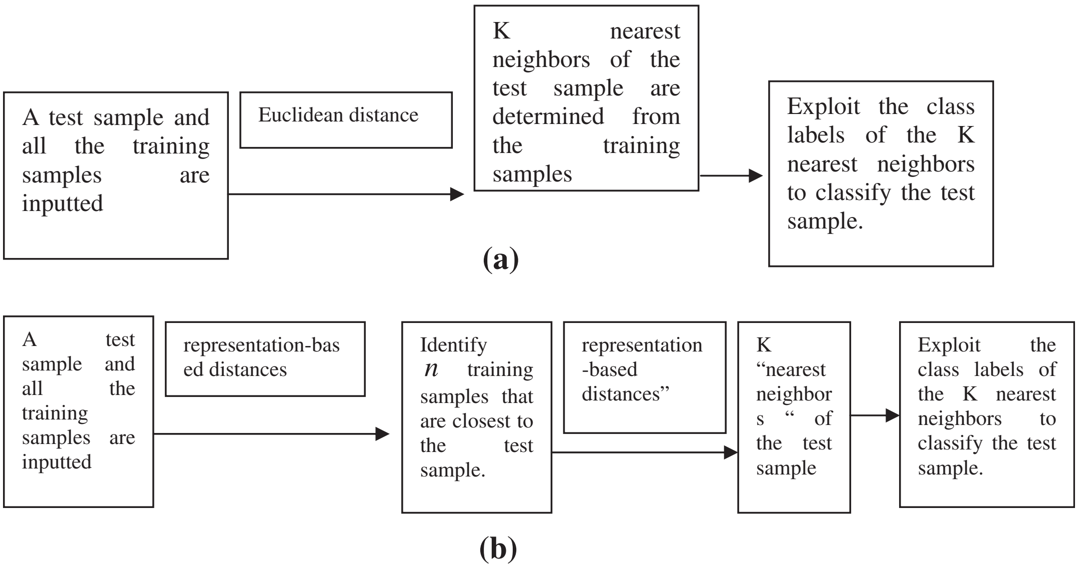

Fig. 1. Flowcharts of CKNNC and CFKNNC. (a) and (b) show the flowcharts of CKNNC and CFKNNC, respectively.

as its previous value. However, in CFKNNC if either of the training samples changes, the distance between the test sample and each of the two training samples must change. Actually, the proposed method has the following idea: it partially views classification as a representation problem of the test sample and considers that a number of training samples that can provide a good approximation for the test sample should be selected as its "nearest neighbors". The method proposed indeed selects the training samples that have much contribution to this approximation as the nearest neighbors of the test sample. Moreover, as shown in Eq. (4), the weighted sum of all the training samples is expected to approximate the test sample and different training samples make contribution to this approximation in a competitive way. Therefore, it seems that the K training samples selected by our method can provide a good approximation, i.e. representation for the test sample and the approximation does not contain much redundant information. From the point of view of representation-based classification, a good representation of the test sample usually can lead to a high classification accuracy. In contrast, as CKNNC separately evaluates the similarity (distance) between the test sample and each training sample, the nearest neighbors selected by CKNNC might contain much more redundant information to represent the test sample.

The coarse to fine strategy used in the proposed method also has the following positive effect. It allows the first step to exclude the training samples that are "far" from the test sample and allows the second step to finally obtain the "optimal" nearest neighbors of the test sample. In particular, it seems that the coarse to fine strategy enables the proposed method first to determine a number of candidates to the K nearest neighbors and enables it then to identify the most competent candidates in terms of the contribution to the approximation of the test sample. In the experimental section, the results show that our coarse to fine strategy can obtain a higher accuracy than CKNNC.

Now we use a two-class problem to illustrate the difference between CKNNC and CFKNNC. Fig. 2 shows the first seven nearest neighbors of the test sample obtained using CKNNC and CFKNNC. In this figure, the notations with a same color denote samples that are from a same class. We see that the nearest neighbors determined by CFKNNC might be very different from those determined by CKNNC. There are two main reasons for this difference. First, CKNNC uses the original sample vectors to determine the nearest neighbors of the test sample, whereas CFKNNC uses the normalized sample vectors to do so. Second, CFKNNC exploits the representation-based "distance" shown in Section 2, whereas CKNNC uses the Euclidean distance. Fig. 2 visually illustrates the difference

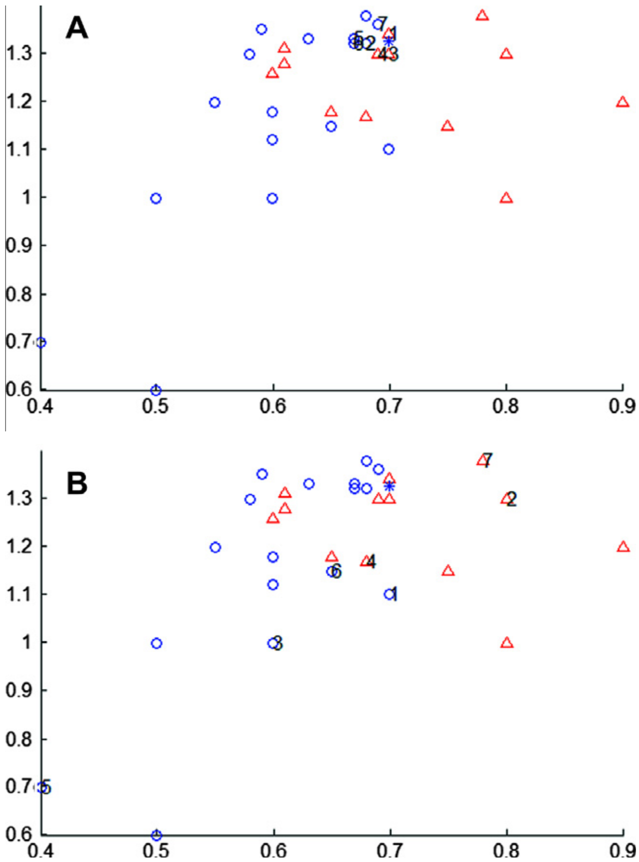

Fig. 2. The first seven nearest neighbors (denoted by black digitals) of the test sample obtained using CKNNC and CFKNNC. (a) and (b) show the results of CKNNC and CFKNNC, respectively. In (a) and (b), the samples are the same. The test sample is denoted by star (∗). The training samples are denoted by circles (O) (the first class) and triangles (the second class), respectively. The genuine class of the test sample is the first class. The first step of CFKNNC selects 26 training samples from the set of all the training samples.

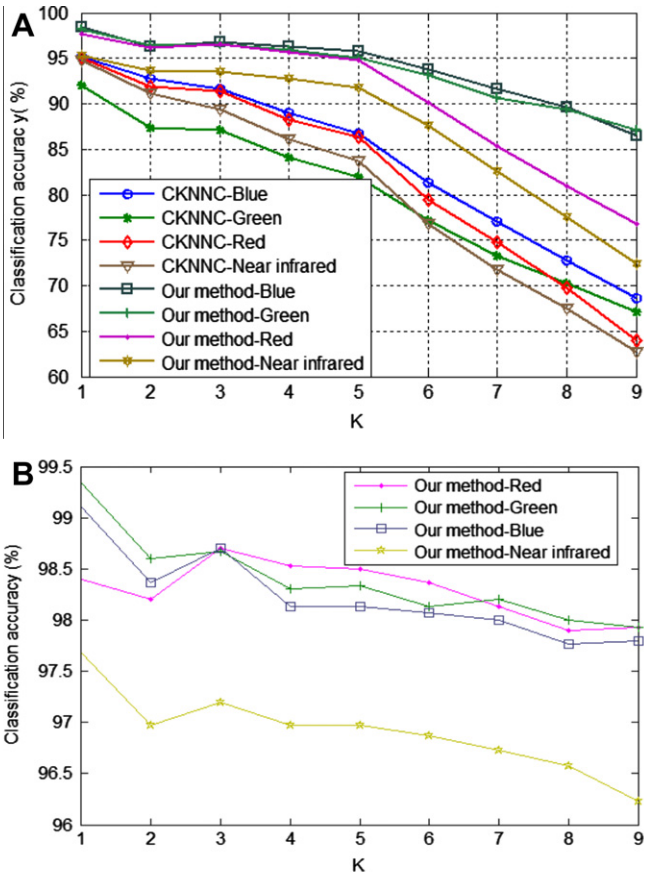

Fig. 3. Experimental results of our method and CKNNC on the multispectral palmprint image dataset. All the images captured in the second session were used as test samples. In (a), the first three images captured in the first session were used as training samples and the first step of our method selects 150 ($n = 150$) samples from the set of all the training samples. In (b), the first six images captured in the first session were used as training samples and the first step of our method selects 300 ($n = 300$) samples from the set of all the training samples.

between CKNNC and our representation-based method, CFKNNC. It is clear that when using 1, 3 or 5 nearest neighbors of the test sample to perform classification, CKNNC will erroneously classify the test sample into the second class. However, when using 1, 3 or 5 nearest neighbors of the test sample to perform classification, CFKNNC will correctly classify the test sample into the first class.

As shown in the experimental section, our method obtains a smaller mean of the cosine correlation coefficients of different nearest neighbors than CKNNC. This illustrates that the nearest neighbors obtained using our method contain less redundant information than those obtained using CKNNC. As a result, under the condition that our method and CKNNC determine a same number of nearest neighbors, the weighted sum of the nearest neighbors obtained using our method can better approximate the test sample

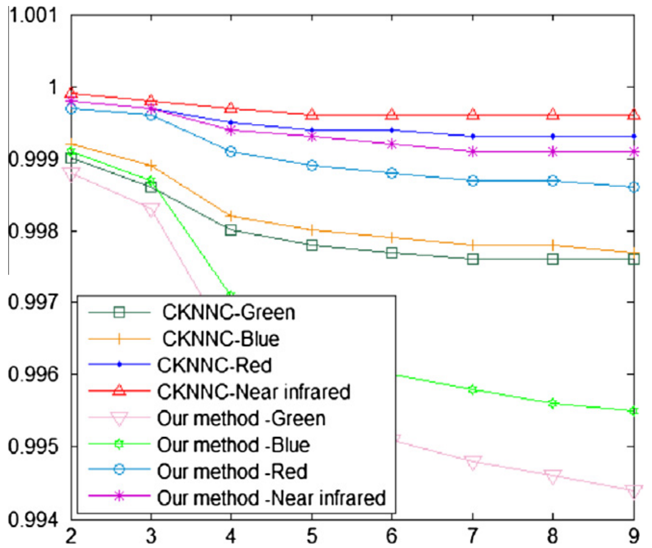

Fig. 4. Mean of the correlation coefficients between different nearest neighbors. The vertical axis shows the mean of the cosine correlation coefficients. The horizontal axis shows the number (i.e. the value of K) of the nearest neighbors, of a test sample, which are used for classification of the test sample.

than that of the nearest neighbors obtained using CKNNC. In the field of pattern recognition a better approximation is usually favored. For example, the widely used principal component analysis has the goal to represent the samples with the smallest error (Xu et al., 2010; Yang et al., 2010).

The cosine correlation coefficient is calculated using

$$sim_{ij} = \frac{s_i^T s_j}{\|s_i\| \|s_j\|}, \tag{5}$$

where s_j denotes the j-th ($j = 1, \ldots, K$) nearest neighbor obtained using our method or CKNNC. sim_{ij} is referred to as the correlation coefficient between the i-th and j-th nearest neighbors of the test sample. For a test sample, the mean of the cosine correlation coefficients is defined as

$$\overline{sim} = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} sim_{ij}. \tag{6}$$

In the experimental section, Fig. 4 shows the mean of $\overline{sim}$ on all the test samples.

4. Experimental results

In this section we show the experimental results of our method, NFL, NFS, NNLC and CBNNC. The main reasons to compare the method proposed in this paper to NFL, NFS, NNLC and CBNNC are as follows. First, these methods are improvements to K nearest neighbor classifier. Second, these methods and our method all

Table 1
Classification accuracies (%) of NFL, NFS, CBNNC, NNLC and the competitive coding method on the multispectral palmprint image dataset.

		Blue	Green	Red	Near infrared
Three training samples per palm	NFL	95.00	92.63	95.17	95.37
	NFS	95.10	92.87	95.40	95.63
	CBNNC	95.20	92.10	95.03	94.73
	NNLC	94.50	91.10	94.30	93.33
	Competitive coding	92.00	90.60	95.33	95.13
Six training samples per palm	NFL	96.57	95.57	97.60	97.47
	NFS	97.30	96.37	97.97	98.17
	CBNNC	96.23	94.93	96.87	96.57
	NNLC	96.30	94.77	96.80	96.20
	Competitive coding	93.83	93.00	95.50	95.83

use a similar way to modify K nearest neighbor classifier. In other words, these methods modify K nearest neighbor classifier by weighting the training samples. In particular, they first replace the original training sample with weighted training sample or take a weighted sum of some training samples as a training sample. Then they calculate the distances between the test sample and weighted training sample or the weighted sum of the training samples and use the distances to classify the test sample.

Before testing these methods, we converted each sample into a unit vector with length of 1 in advance. We also tested competitive coding method. The codes of our method will be available at http://www.yongxu.org/lunwen.html.

4.1. Experiment on the multispectral palmprint dataset

We first conducted experiments on the PolyU multispectral palmprint dataset. This dataset was generated from the Hong Kong Polytechnic University. It was collected from 250 persons (55 women and 195 men) (Zhang et al., 2010; Han et al., 2008; Zhang et al., 2003). As every person provided palmprint images of both the left and right palms, there are 500 palms. This dataset contains four kinds of palmprint images that are captured under the red, green, blue and near infrared illuminations, respectively. We refer to these images as red, green, blue and near infrared palmprint images, respectively. These multispectral palmprint images were collected in two separate sessions. In each session, every palm provided 6 palmprint images at each illumination. As a result, for each illumination there are 6000 palmprint images and the dataset includes 24,000 palmprint images in total from 500 different palms. The resolution of the original palmprint image was 352×288. The 128×128 region of interest (ROI) domain was extracted from each palmprint image using the method proposed in Zhang et al. (2003). In the following experiments, for palmprint ROI images of a palm captured under one illumination, the first three or six images captured in the first session were used as training samples and all the images captured in the second session were used as test samples. We resized each ROI image to a 32×32 image and then converted it into a one-dimensional unit vector with length of 1. Fig. 3 shows the experimental results of our method and CKNNC on the multispectral palmprint image dataset. In Fig. 3(a), the first three images captured in the first session were used as training samples and the first step of our method selects 150 samples from the set of all the training samples. In Fig. 3(b), the first six images captured in the first session were used as training samples and the first step of our method selects 300 samples from the set of all the training samples. From this figure, we also see that if there are more training samples, the method will obtain a higher accuracy. Table 1 shows the classification accuracies of NFL, NFS, CBNNC and NNLC on the multispectral palmprint image dataset. We see that our method always obtains a higher classification accuracy than CKNNC. For example, when only the nearest neighbor was used for classification, the classification accuracies of CKNNC on the blue, green, red and near infrared illumination palmprint ROI images are 95.17%, 92.07%, 95.03% and 94.77%, respectively. However, as shown in Fig. 3, the corresponding classification accuracies of our method on the blue, green, red and near infrared illumination palmprint ROI images are 98.37%, 98.17%, 97.67% and 95.27%, respectively. Our method also outperforms NFL, NFS, CBNNC and NNLC. Both our method and CKNNC obtain the best performance with $K = 1$. The main reason is that palmprint images usually have no much deformation and the "nearest neighbor" of the test sample has a very high probability of being from the same palm as the test sample.

From Table 2, we can see the variation with parameter n of the classification accuracy of our method. Fig. 4 shows the mean of the correlation coefficients between different nearest neighbors. From Fig. 4, we see that our method obtains a smaller mean of the cosine correlation coefficients between different nearest neighbors than

Table 2
Classification accuracies (%) of our method and CKNNC.

	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$
Our method on the blue palmprint images($n = 75$)	97.43	95.00	96.33	95.40	95.23	94.17	92.97	90.83	88.60
Our method on the blue palmprint images($n = 150$)	98.37	96.30	96.73	96.23	95.83	93.77	91.60	89.57	86.50
Our method on the blue palmprint images($n = 300$)	98.57	96.40	96.87	96.37	95.80	93.07	90.47	87.57	84.63
Our method on the 2D palmprint images in the 2D + 3D palmprint dataset($n = 160$)	96.60	95.70	95.85	96.00	95.75	95.90	95.65	94.45	93.30
CKNNC on the 2D palmprint images in the 2D + 3D palmprint dataset	91.15	89.00	87.85	85.50	83.95	82.10	80.30	76.85	74.90
Our method on the noisy multispectral palmprint (R, $n = 300$)	97.70	97.07	98.00	97.97	98.00	97.47	97.13	96.77	96.60
Our method on the noisy multispectral palmprint (G, $n = 300$)	98.77	97.83	98.47	98.30	98.17	97.97	97.90	97.80	97.67
Our method on the noisy multispectral palmprint (B, $n = 300$)	98.83	97.97	98.07	98.07	98.07	97.97	97.90	97.73	97.53
Our method on the noisy multispectral palmprint (I, $n = 300$)	96.40	95.50	95.67	95.87	96.03	95.50	95.43	95.10	95.10
CKNNC on the noisy multispectral palmprint (R)	96.50	95.33	95.03	94.40	93.83	93.10	92.47	91.73	90.70
CKNNC on the noisy multispectral palmprint (G)	94.37	93.00	92.67	91.57	91.17	907	89.37	88.60	87.73
CKNNC on the noisy multispectral palmprint (B)	96.13	95.30	95.33	94.70	94.43	93.27	92.73	92.03	91.33
CKNNC on the noisy multispectral palmprint (I)	96.07	94.73	93.70	93.27	92.63	91.10	90.17	89.33	87.93
Our method on the noisy 2D images in the 2D + 3D dataset ($n = 300$)	95.35	94.00	94.75	94.85	95.15	94.85	94.85	94.30	93.85
CKNNC on the noisy 2D images in the 2D + 3D dataset	87.30	81.20	82.60	80.05	79.00	77.10	75.60	71.90	69.05
Our method on the AR face dataset ($n = 300$)	86.81	71.39	72.22	72.57	74.03	75.35	75.83	76.25	76.18
CKNNC on the AR face dataset	79.44	62.78	62.50	63.06	64.72	65.56	65.56	65.00	63.61

Table 3
Classification accuracies (%) of NFL, NFS, CBNNC, NNLC and the competitive coding method.

	NFL	NFS	CBNNC	NNLC	Competitive coding
2D palmprint images in the 2D + 3D palmprint dataset	92.10	91.95	91.15	90.10	92.35
Noisy multispectral palmprint dataset (R)	97.07	97.60	96.50	96.23	92.17
Noisy multispectral palmprint dataset (G)	95.30	95.80	94.36	94.43	91.50
Noisy multispectral palmprint dataset (B)	96.33	97.03	96.13	96.03	92.17
Noisy multispectral palmprint dataset (I)	96.90	97.80	96.03	95.80	91.83
Noisy 2D images in the 2D + 3D palmprint dataset	89.45	89.50	87.35	89.05	87.65
AR face dataset	84.03	79.72	73.82	86.04	/

CKNNC for each illumination. This illustrates that the nearest neighbors obtained using our method contain less redundant information than those obtained using CKNNC. Table 2 also shows that our method is somewhat robust to parameter n. When n varies from 75 to 300, the accuracy of our method changes only a little.

4.2. Experiment on the 2D + 3D palmprint dataset

We also used a 2D + 3D palmprint dataset to perform experiments. This dataset contains 8000 palmprint samples collected from 400 different palms (Li et al., 2010). Twenty samples from each of these palms were collected in two separated sessions, where 10 samples were captured in each session, respectively. The average time interval between the two sessions is one month. Each sample contains a 3D ROI (region of interest) and its corresponding 2D ROI. Each 3D ROI is recorded by a binary file which contains $128 * 128$ float values denoting the mean curve ratio, and each 2D ROI is recorded by a BMP format image file.

We used the first four 2D ROI images collected in the first session as training samples and took the first five 2D ROI images collected in the second session as test samples. We also resized each image to a 32 by 32 matrix in advance. Each sample in this experiment was also converted into a unit vector with length of 1 in advance. Tables 2 and 3 show the classification accuracies of our method, CKNNC, NFL, NFS, CBNNC, NNLC and the competitive coding method. They also indicate that our method is superior to the other methods in classification accuracy!. The experimental results also show that when the ratio of n to the number of all the training samples ranges from 0.05 to 0.2, the accuracy of our method changes only a little.

4.3. Experiments on noisy test samples

In this subsection, we test the robustness of different methods by imposing Gaussian noise to the test samples (no noise is imposed to the training samples). We use Matlab function "imnoise (I,'gaussian',0,variance)" to generate Gaussian noise, where I stands for the test sample, "0" means that the mean of the noise is zero and the parameter "variance" is set to 0.01.

4.3.1. Experiment on the multispectral palmprint dataset

For palmprint ROI images of a palm captured under one illumination, the first six images captured in the first session were used as training samples and all the images captured in the second session were used as test samples. Tables 2 and 3 show that when recognizing the noisy palmprint images, our method still outperforms the other methods.

4.3.2. Experiments on the 2D + 3D palmprint dataset

The experimental results on the noisy 2D images from the 2D + 3D palmprint dataset are also shown in Tables 2 and 3. The Gaussian noise is also only imposed to the test sample. Tables 2 and 3 show again that when the test sample is corrupted by the noise, our method is also superior to the other methods in classification accuracy.

4.4. Further exploration of CFKNNC

As mentioned in Section 3, the main rationale of the proposed method, i.e. CFKNNC is as follows: the K training samples selected by CFKNNC can provide a good representation for the test sample, and the representation contains less redundant information. In contrast, as CKNNC separately evaluates the similarity (distance) between the test sample and each training sample, the nearest neighbors selected by CKNNC might contain much more redundant information. As a result, the K training samples determined using CFKNNC can better represent the test sample than the K training samples determined using CKNNC. It is likely that if the training samples provide a good representation for the test sample, the test sample will have a higher probability of being from the same class as these training samples. Therefore, CFKNNC can obtain a higher accuracy than CKNNC.

We can also describe the rationale of CFKNNC from an intuitive viewpoint. We take two training samples, respectively from two different classes as an example. We assume that one training sample is from the same class as the test sample and the other is not. We calculate the Euclidean distance between each training sample and the test sample. Because the sample is captured in a noisy condition, it is possible that these two training samples have identical Euclidean distances to the test sample and both of them belong to K nearest neighbors of the test sample. As a consequence, CKNNC might erroneously classify the test sample. However, as CFKNNC can effectively exploit the data structure of the sample, it might correctly obtain the distance relationship between the test sample and these two training samples.

The classification accuracy also show that when K increases, the accuracy of CFKNNC decreases slowly; whereas the accuracy of CKNNC decreases quickly with the increase of K. Under the condition that K is less than or equal to 4, CFKNNC obtain a satisfactory accuracy. In the real-world applications, we can combine the training set and a validation set to determine a proper value for K. On the other hand, all the palmprint recognition experiments show that when $K = 1$, CFKNNC is able to obtain the highest accuracy. As a result, for palmprint recognition, it is feasible to simply set K to 1.

We would like to point out that as our method includes more steps and matrix operations, it has a higher computational complexity than CKNNC.

4.5. Face recognition experiment

In order to verify whether our method is applicable for other biometrics issues, we use the AR face dataset (http://cobweb.ecn.purdue.edu/~aleix/aleix−face−DB.html) to conduct face recognition experiment. The face images of this database were obtained under the condition of varying pose, facial expression, or lighting. Occluded face images are also included in the AR face database. There are 120 subjects and 3120 gray face images captured in two sessions. We resized each image to a 40 by 50 image. The first 14 face images and the remaining face images of each subject are used as training samples and test samples, respectively. Tables 2 and 3 show that our method can also perform very well in face recognition. This implies that our method might be applicable for other biometrics issues.

5. Conclusions

The CFKNNC method proposed in this paper has the following rationales: it depends on an elaborate distance metric, i.e. the representation-based distance to determine the nearest neighbors of the test sample from the set of training samples. The first and second steps of CFKNNC coarsely and finely identify the nearest neighbors, respectively. These two steps are very useful for eliminating the side-effect on classification of the test sample of the training samples that are far from the test sample and for enhancing the positive effect on classification of the test sample of the training samples that are close to the test sample and are probably from the same class as the test sample. Another potential reason why our method outperforms CKNNC is that the nearest neighbors obtained using our method contain less redundant information than those obtained using CKNNC. As a result, under the condition that our method and CKNNC determine a same number of nearest neighbors, the weighted sum of the nearest neighbors obtained using our method is able to better approximate the test sample than that of the nearest neighbors obtained using CKNNC.

The experimental results show that CFKNNC performs very well and can obtain a much higher accuracy than CKNNC. Moreover, CFKNNC also outperforms the state-of-art improvements to CKNNC.

Acknowledgements

This article is partly supported by Program for New Century Excellent Talents in University (Nos. NCET-08-0156 and NCET-08-0155), NSFC under Grant nos. 61071179, 61173086, 61020106004, 60902099 and 61001037 as well as the Fundamental Research Funds for the Central Universities (HIT.NSRIF. 2009130).

References

Chen, Y.-S., Hung, Y.-P., Yen, T.F., Fuh, C.-S., 2007. Fast and versatile algorithm for nearest neighbor search based on a lower bound tree. Pattern Recognition 40 (2), 360–375.

Chien, J.-T., Wu, C.-C., 2002. Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Trans. Pattern Anal. Machine Intell. 24, 1644–1649.

Cover, T.M., Hart, P.E., 1967. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory 13 (1), 21–27.

Descampe, A., Vleeschouwer, C.D., Vandergheynst, P., Macq, B., 2011. Scalable feature extraction for coarse-to-fine JPEG 2000 image classification. IEEE Trans. Image Process. 20 (9), 2636–2649.

Descampe A., Vandergheynst P., Vleeschouwer C.D., Macq B., 2006. Coarse-to-fine textures retrieval in the JPEG 2000 compressed domain for fast browsing of large image datasets. In: MRCS: 282–289.

Feng X., Hadid A., Pietikäinen M., 2004. A coarse-to-fine classification scheme for facial expression recognition. In: ICIAR, (2): 668–675.

Gao, Q.-B., Wang, Z.-Z., 2007. Center-based nearest neighbor classifier. Pattern Recognition 40 (1), 346–349.

Han D., Guo Z., Zhang D., 2008. Multispectral Palmprint Recognition using wavelet-based Image Fusion. In: Internat. Conf. on Signal Process., pp. 2074–2077.

Hu, Q., Zhu, P., Yang, Y., Yu, D., 2011. Large-margin nearest neighbor classifiers via sample weight learning. Neurocomputing 74 (4), 656–660.

Li, S.Z., Lu, J., 1999. Face recognition using the nearest feature line method. IEEE Trans. Neural Networks 10 (2), 439–443.

Li W., Zhang L., Zhang D., Lu G., Yan J., 2010. Efficient joint 2D and 3D palmprint matching with alignment refinement. In: CVPR,pp. 795–801.

Noh Y.-K., Zhang B.-T., Lee D.D., 2010. Generative local metric learning for nearest neighbor classification. In: NIPS, pp. 1822–1830.

Omercevic D., Drbohlav O., Leonardis A., 2007. High-dimensional feature matching: employing the concept of meaningful nearest neighbors. In: IEEE 11th Internat. Conf. on Computer Vision, pp. 1–8.

Park K., Shen C., Hao Z., Kim J., 2011. Efficiently learning a distance metric for large margin nearest neighbor classification. In: Proc. 25th AAAI Conf. on Artificial Intelligence.

Pedersoli M., Vedaldi A., Gonzàlez J., 2011. A coarse-to-fine approach for fast deformable object detection. In: CVPR: 1353–1360.

Rube I.A.E., Ahmed M., Kamel M., 2004. Coarse-to-fine multiscale affine invariant shape matching and classification. (ICPR02): 163–166.

Samet, H., 2008. K-nearest neighbor finding using MaxNearestDist. IEEE Trans. Pattern Anal. Machine Intell. 30 (2), 243–252.

Tahir, M.A., Bouridane, A., Kurugollu, F., 2007. Simultaneous feature selection and feature weighting using hybrid Tabu search/K-nearest neighbor classifier. Pattern Recognition Lett. 28 (4), 438–446.

Toyama, J., Kudo, M., Imai, H., 2010. Probably correct k-nearest neighbor search in high dimensions. Pattern Recognition 43 (4), 1361–1372.

Villegas, M., Paredes, R., 2011. Dimensionality reduction by minimizing nearest-neighbor classification error. Pattern Recognition Lett. 32 (4), 633–639.

Wang, J., Neskovic, P., Cooper, L.N., 2007. Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Lett. 28, 207–213.

Weinberger, K.Q., Blitzer, J., Saul, L.K., 2009. Distance metric learning for large margin nearest neighbor classification. J. Machine Learn. Res. 10, 207–244.

Weinberger, K.Q., Blitzer, J., Saul, L.K., 2006. Distance metric learning for large margin nearest neighbor classification. Adv. Neural Inform. Process. Systems 18, 1473–1480.

Xu, Y., Zhang, D., Yang, J.-Y., 2010. A feature extraction method for use with bimodal biometics. Pattern Recognition 43 (3), 1106–1115.

Yu, D., Yu, X., Hu, Q., Liu, J., Wu, A., 2011. Dynamic time warping constraint learning for large margin nearest neighbor classification. Inform. Sci. 181 (13), 2787–2796.

Yang, W., Sun, C., Zhang, L., Ricanek, K., 2010. Laplacian bidirectional PCA for face recognition. Neurocomputing 74 (1–3), 487–493.

Zhang, D., Kong, W.-K., You, J., Wong, M., 2003. On-line palmprint identification. IEEE Trans. Pattern Anal. Machine Intell. 25 (9), 1041–1050.

Zhang, D., Guo, Z., Lu, G., Zhang, L., Zuo, W., 2010. An online system of multi-spectral palmprint verification. IEEE Trans. Instrum. Meas. 59 (2), 480–490.

Zheng, W., Zhao, L., Zou, C., 2004. Locally nearest neighbor classifiers for pattern classification. Pattern Recognition 37 (6), 1307–1309.