Available online at www.sciencedirect.com
SCIENCE DIRECT
Pattern Recognition 39 (2006) 635–645
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Improving nearest neighbor classification with cam weighted distance

Chang Yin Zhou, Yan Qiu Chen*

Department of Computer Science and Engineering, School of Information Science and Engineering, Fudan University, Shanghai 200433, China

Received 29 July 2005; accepted 16 September 2005

Abstract

Nearest neighbor (NN) classification assumes locally constant class conditional probabilities, and suffers from bias in high dimensions with a small sample set. In this paper, we propose a novel cam weighted distance to ameliorate the curse of dimensionality. Different from the existing neighborhood-based methods which only analyze a small space emanating from the query sample, the proposed nearest neighbor classification using the cam weighted distance (CamNN) optimizes the distance measure based on the analysis of inter-prototype relationship. Our motivation comes from the observation that the prototypes are not isolated. Prototypes with different surroundings should have different effects in the classification. The proposed cam weighted distance is orientation and scale adaptive to take advantage of the relevant information of inter-prototype relationship, so that a better classification performance can be achieved. Experiments show that CamNN significantly outperforms one nearest neighbor classification (1-NN) and k-nearest neighbor classification (k-NN) in most benchmarks, while its computational complexity is comparable with that of 1-NN classification.

Keywords: Classification; Nearest neighbors; Cam distribution; Distance measure

1. Introduction

In a classification problem, given C pattern classes and N labeled training observations (prototypes), the nearest neighbor (NN) classifier, a simple yet appealing approach, assigns to a query pattern x_0 the class label of its NN [1,2]. With the sample size approaching infinity, the error rate of NN classifier converges asymptotically for all sample distributions, to a value between L^* and $2L^*(1 - L^*)$, where L^* is the Bayes risk [2–5].

The finite sample size of many real world problems poses a new challenge. The statistics of x_0 may no longer be the same as that of its NN. Therefore, the estimate of risk varies greatly, depending on the choice of the distance metric [6]. To improve the performance, many methods [7–10] have been proposed to modify the distance metric (measure) such that the finite sample risk will be closer to the asymptotic risk.

However, the existing methods tackle this problem only from the aspect of the query point. These methods [7–10] take advantage of the local information around the query point. They analyze the measurement space emanating from the query point, and study how the distance measure should be changed or weighted. These approaches only examine a small local region surrounding the query sample, so that the most of the inter-prototype information is neglected.

The proposed CamNN classifier optimizes the distance measure with respect to the analysis of the inter-prototype relations. Our motivation comes from the understanding that prototypes are not isolated instances. The nearby prototypes actively affect the confidence level of the information provided by the prototype being considered. Not only should the distance measure to one prototype vary with orientation, the distance measure to each prototype should also be treated discriminately according to its different surroundings.

As shown in Fig. 1, the equi-distance contour in an traditional NN classification is circular, since it assumes all the samples to be isolated and homogeneous. When deflective cam contours are adopted for the prototypes, reflecting the attraction and repulsion effects they receive from their

* Corresponding author. Tel./fax: +86 21 65643842.
E-mail addresses: cyzhou@fudan.edu.cn (C.Y. Zhou), chenyq@fudan.edu.cn (Y.Q. Chen).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.09.004

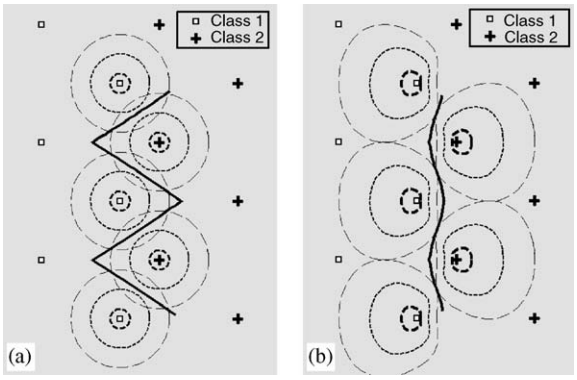

Fig. 1. (a) shows a traditional 1-NN classification. (b) shows 1-NN classifier with ideal cam contours. The dash lines are the equi-distance contours around the prototypes. The black solid line in each figure is the corresponding decision boundary. The cam contours of prototypes can be deemed as a result of the attraction and repulsion effects from its neighbors.

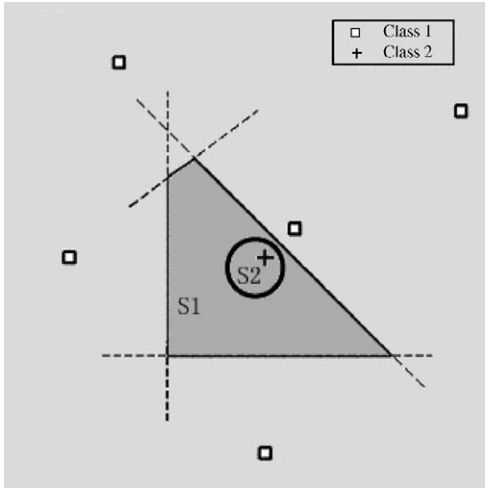

Fig. 2. S1 (containing S2) is the region of Class 2 by the traditional NN classification. S2 is the region of Class 2 by a revised NN classification, who considers the inter-prototype relations and compresses the distance scale when measuring the distance to the solitary prototype of Class 2.

neighbors, the decision boundary becomes smoother and more desirable. From the comparison, it can be seen that an orientation adaptive distance measure can greatly improve the classification performance.

Fig. 2 presents another common situation, where one prototype of Class 2 falls into an area with many prototypes of Class 1. The traditional NN classification treats all the prototypes equally regardless of their surroundings, so that a large region S1 will be decided to belong to Class 2. However, the prototypes are not isolated instances, the inter-prototype relationship should not be neglected. Because of the great weakening effects the solitary prototype suffers from its opposite neighbors, the distance measure scale of this prototype is diminished and then the region belonging to Class 2 is compressed from S1 to the smaller S2. Such a compressed region is usually more desirable. This comparison shows that it could be more reasonable if each prototype is granted a different but appropriate distance scale.

While the idea of optimizing the distance measure from the aspect of prototypes and proposing an orientation and scale adaptive distance measure may seem obvious, few proposals along this line could be found in the literature. A literature review is provided in Section 4. Compared with the existing methods, our major contributions in this paper include:

I. This paper introduces a novel direction for the distance measure optimization for neighborhood-based classifiers. That is to analyze the inter-prototype relationship, and then utilize the relevant information to optimize the distance measure.
II. A novel cam weighted distance, which is orientation sensitive and scale variant, is proposed to describe and make use of the relevant information of inter-prototype relationship. Experiments show that, this distance weighting method, while fairly simple and with small computational complexity, has greatly improved the performance of the NN classifier.

2. Cam weighted distance

In a classification problem, each prototype can be regarded as the center of a probability distribution and the similarity to the prototype can be expressed by the corresponding conditional probability. In the traditional NN method, with the assumption that samples are isolated, the distribution can be a standard normal distribution so that the Euclidean distance is equivalent to the class-conditional probability. However, because of the attraction, repulsion, strengthening effect and weakening effect each prototype receives from its neighbors, the standard normal distributions have been greatly deformed. Obviously, neglecting such a deformation and still using the Euclidean distance to measure the similarity will lead to performance decline.

We construct a simple yet effective transformation $X = (a + b \cdot Y'\tau/\|Y\|) \cdot Y$ to simulate such a deformation, where Y denotes the original distribution and τ is a normalized vector denoting the deformation orientation. We call the deformed distribution cam distribution, if Y subjects to a standard normal distribution. For each prototype representing a cam distribution, its k-nearest neighbor prototypes are further assumed to be the samples of this cam distribution. Then, these samples can be used to estimate the corresponding distribution parameters a, b, and τ. When a, b, and τ are obtained, an inverse transformation can be performed, $Y = X/(a + b \cdot Y'\tau/\|Y\|)$, to eliminate the deformation. Such an inverse transformation will lead to our cam weighted distance.

2.1. Cam distribution

Definition 1 (*Cam distribution*). Consider a p-dimensional random vector $Y = (Y_1, Y_2, \ldots, Y_p)'$ that takes a standard p-dimensional normal distribution $N(0, I)$, that is, it has a probability density function

$$f(y) = \frac{1}{(2\pi)^{p/2}} \cdot \mathrm{e}^{-1/2 \cdot y'y}. \tag{1}$$

Let a random vector X be defined by the transformation

$$X = \left(a + b \cdot \frac{Y'\tau}{\|Y\|}\right) \cdot Y \tag{2}$$

or

$$X = (a + b \cdot \cos\theta) \cdot Y, \tag{3}$$

where $a > b \geqslant 0$, τ is a normalized vector, $\|Y\| = \sqrt{Y'Y}$, and θ is the included angle of vectors Y and τ. Then the distribution of X is called the *cam distribution* with parameters a and b in the direction τ, denoted as $X \sim Cam_p(a, b, \tau)$.

Theorem 1. *If a random vector X has a cam distribution $Cam_p(a, b, \tau)$, then the probability density function of X is as follows, for $x \in \Re^p$*

$$p(x) = \frac{1}{(2\pi)^{p/2}(a + b(x'\tau/\|x\|))^p} \cdot \exp\left[-\frac{1}{2}\left(\frac{\|x\|}{a + b(x'\tau/\|x\|)}\right)^2\right] \tag{4}$$

or

$$p(x) = \frac{1}{(2\pi)^{p/2}(a + b \cdot \cos\theta)^p} \cdot \exp\left[-\frac{1}{2}\left(\frac{\|x\|}{a + b \cdot \cos\theta}\right)^2\right], \tag{5}$$

where θ is the included angle of vectors x and τ (see Appendix A for a proof).

Theorem 2. *If a random vector $X \sim Cam_p(a, b, \tau)$, then*

$$E(X) = c_1 \cdot b \cdot \tau \tag{6}$$

and

$$E(\|X\|) = c_2 \cdot a, \tag{7}$$

where c_1 and c_2 are constants

$$c_1 = 2^{1/2} \cdot \frac{\Gamma((p+1)/2)}{\Gamma(p/2)} \Big/ p$$
$$c_2 = 2^{1/2} \cdot \frac{\Gamma((p+1)/2)}{\Gamma(p/2)}, \tag{8}$$

$\Gamma(\cdot)$ denoting the Gamma function $\Gamma(k) = \int_0^\infty t^{k-1}\,\mathrm{e}^{-t}\,\mathrm{d}t$ $(k > 0)$ (see Appendix B for a proof).

The role of each parameter in the transformation is shown clearly in this theorem. The parameter a is the determinant of the overall scale of the distribution; the parameter b determines the extent to which the peak deviates from the origin, and the parameter τ decides the orientation of the deviation.

2.2. Cam weighted distance

As mentioned above, the cam distribution is an eccentric distribution that biases towards a given direction. It is obtained from a standard normal distribution by the transformation $X = Y \cdot (a + b\cos\theta)$. In this model, the Euclidean distance is not suitable to describe the similarity directly, since the assumed normal distribution has been deformed. Instead, we firstly restore the deformation by an inverse transformation $Y = X/(a + b\cos\theta)$, and then measure the distance. Thus, we obtain a cam weighted distance. This weighted distance redresses the deformation and should be more suitable to describe the similarity.

Definition 2 (*Cam weighted distance*). Assume $x_0 \in \Re^p$ is the center of a cam distribution $Cam_p(a, b, \tau)$. The cam weighted distance from a point $x \in \Re^p$ to x_0 is defined to be

$$CamDist(x_0, x) = \|x - x_0\| \Big/ \left(a + b \cdot \frac{(x - x_0)'\tau}{\|x - x_0\|}\right) \tag{9}$$

or

$$CamDist(x_0, x) = \|x - x_0\|/(a + b\cos\theta), \tag{10}$$

where θ is the included angle of vectors $x - x_0$ and τ. Especially, $1/(a + b\cos\theta)$ is called the cam weight of the distance from x to x_0.

Three cam distributions $Cam_2(1, 0, [0.8, 0.6])$, $Cam_2(1, 0.4, [0.8, 0.6])$, and $Cam_2(1, 0.8, [0.8, 0.6])$ are shown up in Fig. 3, respectively, from the left to the right. The solid line in each figure is an equi-distance contour according to the cam weighted distance. By examining the equi-distance contour $CamDist(x_0, x) = d_0$ and the distance weight, $1/(a + b\cos\theta)$, we can find that the parameter a reflects the overall scale of the distance measure and b reflects the extent of orientation in distance measure. When $b = 0$, the contour is circular. As b increases, it looks more like a cam curve. When b approaches to a, the contour becomes a heart curve. In most cases, b is a medium value with respect to a, which indicates a cam curve. That is why we call it cam weighted distance. For a set of points with the same Euclidean distance to the origin, the one in the direction τ has the nearest cam weighted distance to the origin, while the one in the direction $-\tau$ is the farthest.

We should point out that cam weighted distance measure is just a weighted distance, but not a metric, since $CamDist(x_0, x)$ may not equal to $CamDist(x, x_0)$, even $CamDist(x, x_0)$ is not defined.

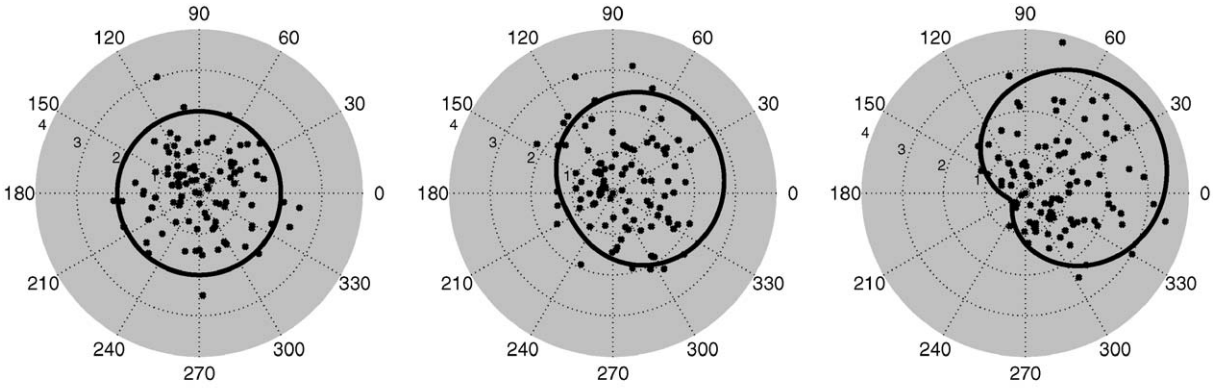

Fig. 3. Three cam distributions $Cam_2(1, 0, [0.8, 0.6])$, $Cam_2(1, 0.4, [0.8, 0.6])$, $Cam_2(1, 0.8, [0.8, 0.6])$ are shown up, respectively, from the left to the right, each one with one hundred samples. The samples are marked by black dots. The black dash line in each figure is an equi-distance contour according to the cam weighted distance.

2.3. Parameter estimation

The properties presented in Theorem 2 have considerably facilitated parameter estimation. For an arbitrary prototype $x_i \in D$, we assume that it represents a cam distribution and is the origin of this cam distribution, and that the prototypes nearby subject to this distribution. Then, we use its k-nearest neighbors $X_i = \{x_{i1}, x_{i2}, \ldots, x_{ik}\}$ to estimate the parameters of the cam distribution, including a_i, b_i and τ_i.

First, we convert X_i to a set of vectors $V_i = \{v_{i1}, v_{i2}, \ldots, v_{ik}\}$, where $v_{ij} = x_{ij} - x_i$, $j = 1, 2, \ldots, k$. Then, we use $\widehat{G_i}$ and $\widehat{L_i}$, which are the center of mass and the averaged vector length of V_i.

$$\begin{aligned} \widehat{G_i} &= \textstyle\sum_{j=1}^{k} v_{ij}/k, \\ \widehat{L_i} &= \textstyle\sum_{j=1}^{k} \|v_{ij}\|/k \end{aligned} \tag{11}$$

to estimate $E(\eta)$ and $E(\|\eta\|)$, respectively. According to Theorem 2, we get an estimation to a_i, b_i, and τ_i:

$$\begin{aligned} \widehat{a_i} &= \widehat{L_i}/c_2, \\ \widehat{b_i} &= \|\widehat{G_i}\|/c_1, \\ \widehat{\tau_i} &= \widehat{G_i}/\|\widehat{G_i}\|. \end{aligned} \tag{12}$$

It can be easily proved that it is an unbiased estimation.

2.4. More details of parameter estimation

It should be noted that the above estimation focuses on a single class situation and assumes all k-nearest neighbors of x_i have the same class label as x_i. However, in a two-class or multiple-class classification problem, for an arbitrary prototype x_i, its k-nearest neighbors $X_i = \{x_{i1}, x_{i2}, \ldots, x_{ik}\}$ may come from other opposite classes, so we should not use these neighbor prototypes directly for parameter estimation. A simple skill is employed in our implementation to solve this problem. Assume y_{i0} is the label of x_i and y_{ij} is the label of the neighbors x_{ij}, $j = 0, 1, \ldots, k$. We convert V_i in Eq. (11) to W_i, according to

$$w_{ij} = \begin{cases} v_{ij} & \text{if } y_{ij} = y_{i0}, \\ -\frac{1}{2} \cdot v_{ij} & \text{if } y_{ij} \neq y_{i0}, \end{cases} \tag{13}$$

where $j = 1, 2, \ldots, k$. Then, Eq. (11) is revised to be

$$\begin{aligned} \widehat{G_i} &= \textstyle\sum_{j=1}^{k} w_{ij}/k, \\ \widehat{L_i} &= \textstyle\sum_{j=1}^{k} \|w_{ij}\|/k. \end{aligned} \tag{14}$$

Such a simple transformation not only reserves most of the sample scatter information, but also reflects the relative position of the current class to the nearby opposite classes, so that the orientation information can be reserved.

3. CamNN classification

3.1. The CamNN algorithm

The proposed cam weighted distance can be more suitable to measure the similarity than the Euclidean distance in many cases, since it makes use of the relevant information of the inter-prototype relationship. Accordingly, we propose a novel classification method CamNN which uses cam weighted distance to improve the neighborhood-based classifier.

The simplicity of the parameter estimation of cam weighted distance makes the CamNN straightforward. The whole process consists of two phases. In the first training phase, for each prototype x_i in the training set D, CamNN firstly finds its k-nearest prototypes by the Euclidean

Table 1
The CamNN algorithm

Phase 1: Training
Given a prototype set $D = \{x_i\}$, the corresponding class labels $C = \{y_i\}$ and a parameter k, for each prototype $x_i \in D$:

(1) Find its k-nearest neighbors $X_i = \{x_{i1}, x_{i2}, \ldots, x_{ik}\}$, $X_i \subset D$.
(2) Obtain V_i from X_i by $v_{ij} = x_{ij} - x_i$, $j = 1, \ldots, k$.
(3) Obtain W_i from V_i according to Eq. (13).
(4) Calculate $\widehat{G_i}$ and $\widehat{L_i}$ with W_i, according to Eq. (14).
(5) Estimate a_i, b_i, τ_i by using $\widehat{G_i}$ and $\widehat{L_i}$, according to Eq. (12).
(6) Save a_i, b_i, τ_i to A_i.

Phase 2: Classification
For an arbitrary query $q \in \Re^p$,
(7) Calculate the cam weighted distance from q to each prototype x_i according to Eq. (10):

$$CamDist(x_i, q) = \|q - x_i\|/(a_i + b_i \cos\theta_i),$$

where θ_i is the included angle of vectors $q - x_i$ and τ_i.
(8) Find the nearest neighbor $x^* \in D$, which satisfies

$$CamDist(x^*, q) = \min_{x_i \in D} CamDist(x_i, q).$$

(9) Return the label y^*, where y^* is the class label of x^*.
*By modifying the classification phase, we can also combine cam weighted distance with k-NN or some other classification methods.

distance, and then uses these k-nearest prototype to estimate the three cam weighting parameters a_i, b_i and τ_i, according to Eqs. (13), (14) and (12). After this phase, a parameter matrix A is obtained: $A_i = [a_i, b_i, \tau_i]$, $i = 1, 2, \ldots, |D|$, so that we will be able to calculate the cam weighted distance $CamDist(x_i, q)$ from any query point $q \in \Re^p$ to an arbitrary prototype $x_i \in D$, according to Eq. (10).

In the following classification phase, many neighborhood-based classifiers can be applied, including 1-NN classifier, k-NN classifier, or some other methods [8–10], and the proposed cam weighted distance can be used to improve their classification performance. To simplify the problem, we only apply cam weighted distance on the traditional 1-NN classification in our implementation. That is, for any query $q \in \Re^p$, we find the prototype with the shortest cam weighted distance and assign to q the label of this prototype. The detailed steps of this proposed method CamNN are listed in Table 1.

3.2. The computational complexity

It is remarkable that CamNN is computationally competitive with the traditional 1-NN classification, while CamNN has significantly outperformed the k-NN classification (see Section 5). Given a classification problem with M prototypes and N queries, the computational complexity of CamNN in the training phase is $O(k * M)$. In the classification phase, similar with 1-NN, CamNN only needs to find one nearest neighbor. In each cam weighted distance computation, according to Eq. (9), there are two inner product operations. So, CamNN's computational complexity in the classification phase is $O(2 * M * N)$, close to that of 1-NN $O(M * N)$. Compared with k-NN whose complexity is $O(k * M * N)$ or with many other adaptive NN methods, which will be much more complicated, such as [7–10], CamNN has obvious computational advantage.

4. Literature review

Hastie [10] introduces discriminate adaptive NN classification (DANN) metric which combines the advantage of linear discriminant (LDA) classifier and NN classifier to ameliorate the curse of dimensionality. For each query, DANN iteratively adjusts its metric while searching for the k-nearest neighbors. DANN elongates the distance along the linear discriminate boundary, which is believed to have contributed to the improvement of the performance of k-NN.

Short [7] uses the k-nearest neighbors of the query point to construct a direction vector, defines the distance as the multiplication of a vector with this direction vector and then selects the nearest one from k-nearest neighbors to classify the query x_0.

Friedman [9] integrates tree-structured recursive partitioning techniques and regular k-NN methods, to estimate the local relevance of each query point, and then uses this information to customize the metric measure centered at the query.

Similar to [7,9,10], Domeniconi [8] proposes another method to take advantage of the local relevance around the query point. This method employs a Chi-squared distance to iteratively compute a flexible metric for producing neighborhoods that are highly adaptive to query locations.

Also, there are some other methods like [11,12], which estimate some global information and use them to improve the performance of the NN classification. Fukunaga [11] presents a global quadratic metric $d_{A_0}(X, Y) = [(X - Y)^{\mathrm{T}} A_0 (X - Y)]^{1/2}$ to minimizes the mean-squared error between the NN asymptotic risk and the finite sample risk. In [12], a class-dependent weighted (CDW) dissimilarity measure in vector spaces is proposed to improve the performance of the NN classifier.

From the view point of information retrieval, all of these methods are very different from our proposed CamNN. Methods [7–10] and many other methods like [13,14] take advantage of the local information around the input point. They analyze the measurement space emanating from the input point, and study how the neighbors should be weighted according to their relations with the input point. And methods [11,12] only use some global information to improve the performance. In contrast, the proposed CamNN analyzes and takes advantage of the inter-prototype relationship. In many cases, the information of the inter-prototype relationship is very important, but is difficult to be obtained either from the aspect of the query point or from the global aspect.

5. Experimental evaluation

We have performed three sets of experiments to evaluate the effect of the cam weighted distance on the performance of the NN classification. The proposed CamNN is also further compared with the k-NN classifier in the experiments. The performance of k-NN is highly dependent on the choice of the parameter k, so the selection of the parameter k is crucial to the k-NN classifier. To make the comparison fair and objective, we always choose the best parameter k for k-NN in each experiment.

5.1. Experiments on two artificial problems

In this set of experiments, we examine whether CamNN performs up to our motivation explained in the Introduction. The first experiment in this set is performed on the problem shown in Fig. 1, and the results of 1-NN, 5-NN and our proposed CamNN are shown in Fig. 4. In the second experiment, we apply 1-NN, 5-NN and CamNN to classify two classes which follow independent standard bidimensional normal distribution $N(0, I)$, centered at $(-1, 0)$ and $(1, 0)$, respectively. The classification results are shown in Fig. 5.

As can be seen in Figs. 4 and 5 that, the decision boundary of CamNN is smoother and closer to the Bayesian decision boundary than those of 1-NN and 5-NN. By comparing Fig. 4 with Fig. 1, it is noted that the orientation adaptivity has been well realized in our proposed method; by comparing Fig. 5 with Fig. 2, we can see that the proposed CamNN has also achieved another objective—scale adaptivity.

5.2. Experiments on Elena artificial database

The Elena artificial database is a famous set of benchmarks for classification, including Concentric, Clouds and Gaussian datasets (*ftp://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases*). For each dataset, we randomly select half of the entries as training set, and test using the other half. To obtain a reliable result, we repeat each experiment for 20 times independently, and the average cross-validation error rates are reported in Table 2.

All the 'Gaussian' datasets correspond to the same problem, but with dimensionality ranging from 2 to 8. Each 'Gaussian' dataset consists of two classes, one class represented by a multivariate normal distribution with zero mean and standard deviation equal to 1 in all dimensions, and another class represented by a normal distribution with zero mean and standard deviation equal to 2 in all dimensions. This allows the study of the classifier behavior for different dimensionalities of the input vectors, for heavy overlapped distributions and for nonlinear separability [15]. From Table 2, it can be seen that CamNN outperforms 1-NN significantly for all Elena artificial datasets, and can soon preponderates over the k-NN classifier as the dimensionality increases. CamNN has been by far the best performer in Gaussian-6, Gaussian-7 and Gaussian-8, though CamNN is inferior to the k-NN classifier in several low-dimensional datasets. The results on Gaussian datasets show that, the proposed CamNN can better handle the dimensionality increase than k-NN.

The dataset 'Concentric' with bidimensional uniform concentric circular distributions may be used to study the linear separability of the classifier when some classes are nested in other without overlapping. Despite of the large sample and low-dimension, CamNN still shows a large advantage over the k-NN classifier on the dataset 'Concentric'. CamNN's outstanding result on 'Concentric' shows its great linear separability.

Further experiments are performed to evaluate CamNN with respect to the sample size. Table 2 has shown that when the training set is large ($M = 5000 * \frac{1}{2}$) CamNN only slightly outperforms k-NN on the dataset Gaussian-4. Now, we are going to examine how the comparison results will change when the size of the training set M is set to 3200, 1600, 800, 400, 200, 100 and 50. For each setting, the training samples are randomly selected from the whole 5000 samples and the rest is used as test samples. Again, each experiment is repeated 20 times for a reliable result, and the averaged cross-validation error rates are illustrated in Fig. 6.

It is shown clearly in Fig. 6 that with the sample size decreasing, CamNN's advantage over k-NN becomes more and more obvious. When the training set is very large, the performance of k-NN and CamNN is very close. However, when the performance of k-NN declines quickly with the training set size decreasing, CamNN still can keep a relative low error rate. This trend indicates that CamNN

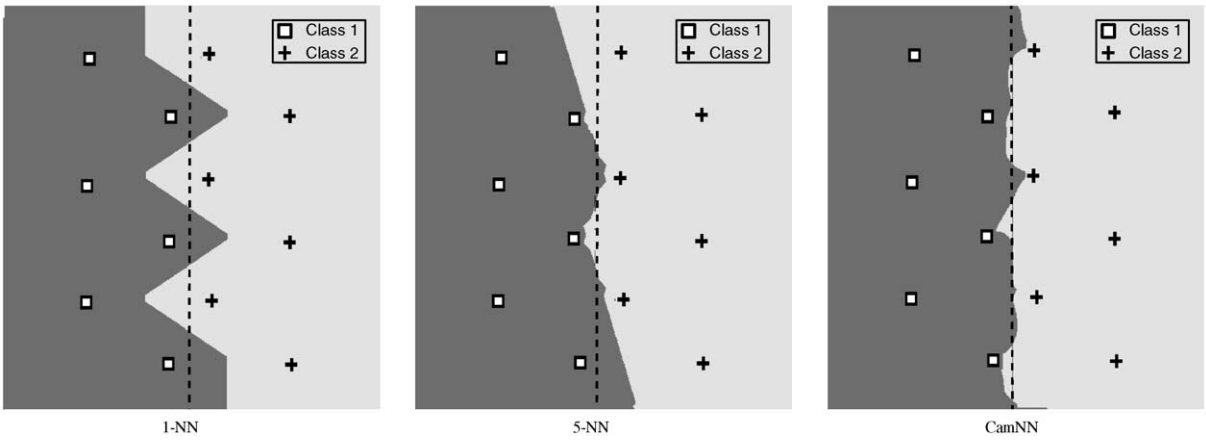

Fig. 4. The results of 1-NN, 5-NN and CamNN ($K = 5$) are shown up, respectively, from the left to the right. Any points in the left grayed area will be classified to Class 1. It can be seen that the decision boundary of CamNN is more desirable.

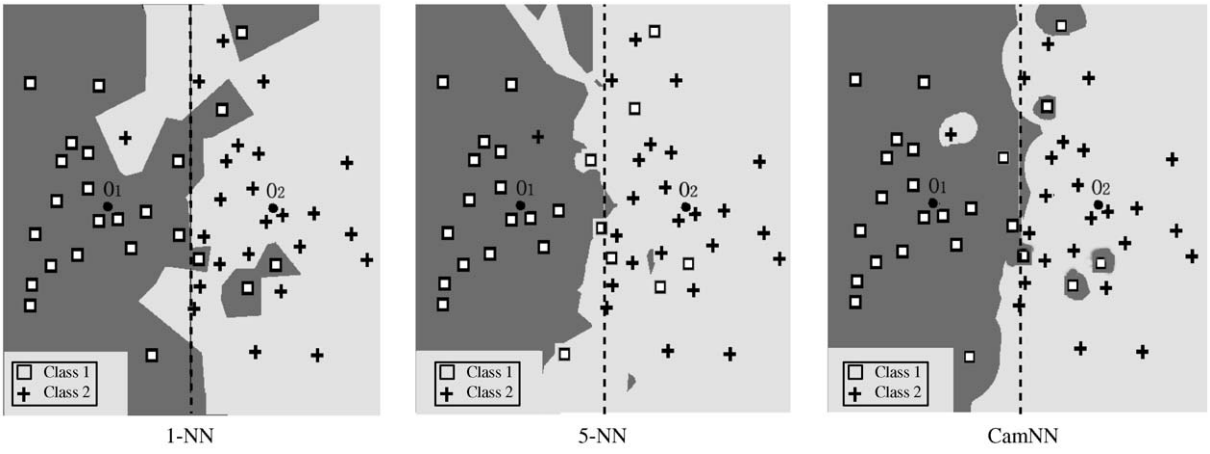

Fig. 5. The marked points are training data coming from two independent standard normal distributions which are centered at O_1 and O_2 ($\|O_1 - O_2\| = 2$), respectively. The central dash line in each figure is the Bayes decision line. The classification results of 1-NN, 5-NN and CamNN ($K = 5$) are shown up from the left to the right, respectively. Any points in the grayed area will be classified to Class 1.

Table 2
Comparison results on Elena datasets

Dataset	#C	#Dim	#Samples	1-NN	k-NN		CamNN	
				Error rate (%)	Error rate (%)	K	Error rate (%)	K
1 Gaussian-2	2	2	5000	35.4 ± 0.7	$\mathbf{27.7 \pm 0.5}$	21	34.2 ± 0.4	16
2 Gaussian-3	2	3	5000	32.0 ± 0.7	$\mathbf{23.4 \pm 0.6}$	23	26.8 ± 0.6	5
3 Gaussian-4	2	4	5000	27.5 ± 0.7	21.3 ± 0.6	19	$\mathbf{21.2 \pm 0.5}$	6
4 Gaussian-5	2	5	5000	24.6 ± 0.5	19.5 ± 0.4	13	$\mathbf{18.5 \pm 0.4}$	6
5 Gaussian-6	2	6	5000	22.0 ± 0.7	18.3 ± 0.5	9	$\mathbf{15.6 \pm 0.4}$	6
6 Gaussian-7	2	7	5000	20.6 ± 0.6	17.9 ± 0.6	5	$\mathbf{14.2 \pm 0.4}$	6
7 Gaussian-8	2	8	5000	20.3 ± 0.7	18.6 ± 0.5	3	$\mathbf{12.8 \pm 0.3}$	6
8 Clouds	2	2	5000	15.3 ± 0.3	$\mathbf{11.9 \pm 0.3}$	23	12.8 ± 0.4	45
9 Concentric	2	2	2500	1.9 ± 0.2	1.8 ± 0.3	9	$\mathbf{1.1 \pm 0.2}$	11

The best performer for each dataset is bolded.
Best k is selected for k-NN classification in each experiment.

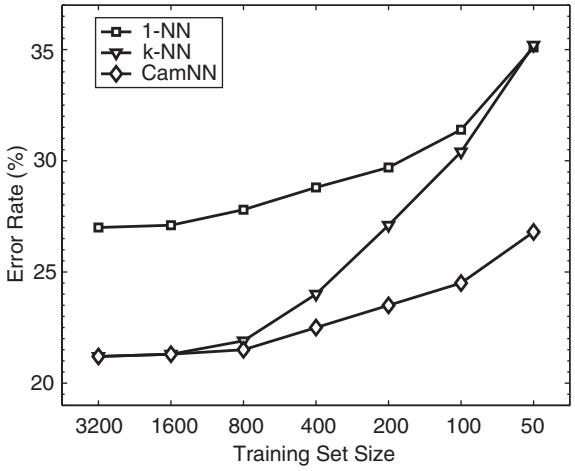

Fig. 6. 1-NN, k-NN and CamNN are compared on the dataset Gaussian-4 with respect to the decreasing sample size.

can be a promising method to resolve some small sample problems.

5.3. Experiments on UCI machine learning database

Though comparative studies on real world data tend to be less informative than those based on artificial data because of the unknown underlying structure and limited sample size, comparisons on real world data is still very useful for several reasons. First, the real world data is produced without favoring any particular algorithm, so that the comparison results will be more objective. Second, it is always suspected that the results on the artificial data cannot be generalized to the problems in the real world. So, our experiments are also performed on some real data to evaluate the proposed CamNN method.

Ten different real world datasets are taken from the well-known UCI Machine Learning database at *http://www.ics.uci.edu/~mlearn/MLRepository.html*. For each dataset, the features are firstly normalized to have zero mean and unit variance and then leave-one-out [16] cross-validation is performed to measure the performance. The comparison results of 1-NN, k-NN and CamNN on UCI database are given in Table 3.

Again, CamNN significantly outperforms 1-NN and k-NN, and is the best performer for eight of the 10 datasets. For the remaining two dataset, CamNN is only slightly inferior to the k-NN classifier. Particularly, it can be observed that on 'balance-scale', 'ionosphere' and 'wine', CamNN is by far the best performer. It is remarkable that in our implementation, CamNN is essentially an adaptive 1-NN classifier using cam weighted distance instead of Euclidean distance. So, CamNN outperforming 1-NN and k-NN in these experiments shows the effectiveness of cam weighted distance in measuring the similarity.

6. Summary and conclusions

This paper presents a novel way to optimize the distance measure for the neighborhood-based classifiers. Our motivation is that the prototypes are not isolated and by analyzing the inter-prototype relationship, we are able to obtain useful information to optimize the distance measure.

We have proposed a method CamNN to analyze and take advantage of the inter-prototype relationship. The cam weighted distance, the core of CamNN, has two essential characteristics, orientation and scale adaptivity, which enable it to reflect the inter-prototype relationship effectively, so that a better classification performance is achieved. The efficacy of our method is validated by the experiments using both artificial and real data.

Remarkably, CamNN is computationally competitive with 1-NN while its performance has significantly outperformed k-NN in most datasets. Furthermore, CamNN confines the

Table 3
Comparison results on UCI datasets

Dataset	#C	#Dim	#Samples	1-NN	k-NN		CamNN	
				Error rate (%)	Error rate (%)	K	Error rate (%)	K
1 Auto-MPG	3	7	392	26.7	26.5	7	**24.2**	8
2 Balance-Scale	3	4	625	19.7	9.8	7	**8.4**	5
3 Bcw	2	9	699	4.9	**3.3**	7	**3.3**	9
4 Wdbc	2	30	569	4.9	**3.2**	9	3.5	5
5 Glass	6	9	214	29.9	**27.6**	3	**27.6**	11
6 Ionosphere	2	33	351	13.4	13.4	1	**6.8**	60
7 Iris	3	4	150	5.3	4	7	**3.3**	6
8 Liver-disorder	2	6	345	36.8	**34.5**	3	35.3	11
9 Pima	2	8	768	29.3	25.8	5	**24.7**	4
10 Wine	3	10	178	6.7	4.5	3	**2.8**	7

The best performer for each dataset is bolded.
The best k is selected for k-NN classification in each experiment.

analysis to the training phase, so that its classification phase is fairly straightforward and fast.

Acknowledgements

The research work presented in this paper is supported by National Natural Science Foundation of China, Grant No. 60275010, and Science and Technology Commission of Shanghai Municipality, Grant No. 04JC14014.

Appendix A. Proof of the Theorem 1

Theorem 1. *If a random vector X has a cam distribution* $Cam_p(a, b, \tau)$, *then the probability density function of X is as follows, for* $x \in \Re^p$

$$p(x) = \frac{1}{(2\pi)^{p/2}(a + b(x'\tau/\|x\|))^p} \cdot \exp\left[-\frac{1}{2}\left(\frac{\|x\|}{a + b(x'\tau/\|x\|)}\right)^2\right] \tag{15}$$

or

$$p(x) = \frac{1}{(2\pi)^{p/2}(a + b\cdot\cos\theta)^p} \cdot \exp\left[-\frac{1}{2}\left(\frac{\|x\|}{a + b\cdot\cos\theta}\right)^2\right], \tag{16}$$

where $\|x\| = \sqrt{x'x}$, *and* θ *is the included angle of vectors x and* τ.

Proof. According to Definition 1, there exists a transformation $X = (a + b \cdot Y' \cdot \tau/\|Y\|) \cdot Y$ from a standard normal random vector $Y \sim N(0, I_p)$ to a cam random vector $X \sim (Cam_p a, b, \tau)$. This is a one-to-one transformation, so we can write

$$x = \left(a + b \cdot \frac{y' \cdot \tau}{\|y\|}\right) \cdot y. \tag{17}$$

Thus, the density of X is then given by

$$p(x) = f(y(x)) \cdot |J(y \to x)|, \tag{18}$$

where $f(\cdot)$ is the probability density function of Y and $J(y \to x)$ is the Jacobian of the transformation $y = y(x)$.

(i) By Eq. (17), we get

$$x'\tau = \left(a + b \cdot \frac{y' \cdot \tau}{\|y\|}\right) \cdot y'\tau,$$

$$x'x = \left(a + b \cdot \frac{y' \cdot \tau}{\|y\|}\right)^2 \cdot y'y,$$

$$\frac{y' \cdot \tau}{\|y\|} = \frac{x' \cdot \tau}{\|x\|}. \tag{19}$$

Hence,

$$y = \left(a + b\frac{x'\tau}{\|x\|}\right)^{-1} x. \tag{20}$$

(ii) From Eq. (17), we have

$$\begin{aligned} \mathrm{d}x &= \mathrm{d}\left(a + b\frac{y'\tau}{\|y\|}\right) \cdot y + \left(a + b\frac{y'\tau}{\|y\|}\right)\mathrm{d}y \\ &= b\left(\frac{\mathrm{d}y'\tau}{\|y\|} - \frac{y'\tau}{(\|y\|)^3} \cdot y'\,\mathrm{d}y\right) y + \left(a + b\frac{y'\tau}{\|y\|}\right)\mathrm{d}y \\ &= \frac{b}{\|y\|^3} \cdot y \cdot (y'y\tau - y'\tau y') \cdot \mathrm{d}y + \left(a + b\frac{y'\tau}{\|y\|}\right)\mathrm{d}y, \end{aligned}$$

so that

$$\frac{\mathrm{d}x}{\mathrm{d}y} = \frac{b}{\|y\|^3} \cdot y \cdot \left(y'y\tau - y'\tau y'\right) + c \cdot I_p, \quad c = b\frac{y'\tau}{\|y\|}.$$

Thus,

$$J(x \to y) = det\left[\frac{\mathrm{d}x}{\mathrm{d}y}\right] \tag{21}$$

$$= |c^p| \cdot \left|\frac{1}{\|y\|^2 y'\tau} y\left(y'y\tau' - y'\tau y'\right) + I_p\right|. \tag{22}$$

Since

$$|I_p + A_{p\times q}B_{q\times p}| = |I_q + B_{q\times p}A_{p\times q}|, \tag{23}$$

we have

$$J(x \to y)$$

$$= |c^p| \cdot \left(1 + \frac{1}{\|y\|^2 y'\tau} \cdot \left(y'y\tau'y - y'\tau y'y\right)\right) \tag{24}$$

$$= |c^p| \tag{25}$$

$$= \left|a + b\frac{y'\tau}{\|y\|}\right|^p. \tag{26}$$

Thus, we obtain the Jacobian as follows

$$J(y \to x) = \frac{1}{J(x \to y)} = \left|a + b\frac{y'\tau}{\|y\|}\right|^{-p}.$$

According to Eq. (19), we have

$$J(y \to x) = \left|a + b\frac{x'\tau}{\|x\|}\right|^{-p}. \tag{27}$$

(iii) By Eqs. (20) and (27), the p.d.f of X is

$$\begin{aligned} p(x) &= f[y(x)] \cdot J(y \to x) \\ &= (2\pi)^{-p/2} \cdot \exp\left[-\tfrac{1}{2} y'y\right] \cdot J(y \to x) \\ &= (2\pi)^{-p/2} \cdot \exp\left[-\frac{1}{2}\left(a + b\frac{x'\tau}{\|x\|}\right)^{-2} x'x\right] \\ &\quad \cdot \left(a + b\frac{x'\tau}{\|x\|}\right)^{-p} \\ &= \frac{1}{(2\pi)^{p/2}(a + b\,(x'\tau/\|x\|)\,)^p} \\ &\quad \cdot \exp\left[-\frac{1}{2}\left(\frac{\|x\|}{a + b(x'\tau/\|x\|)}\right)^2\right]. \end{aligned}$$

Since $\cos\theta = x'\tau/\|x\|$, we also have

$$\begin{aligned} p(x) &= \frac{1}{(2\pi)^{p/2}(a + b\cdot\cos\theta)^p} \\ &\quad \cdot \exp\left[-\frac{1}{2}\left(\frac{\|x\|}{a + b\cdot\cos\theta}\right)^2\right]. \end{aligned}$$

□

Appendix B. Proof of the Theorem 2

Theorem 2. *If a random vector $X \sim Cam_p(a, b, \tau)$, then*

$$E(X) = c_1 \cdot b \cdot \tau \tag{28}$$

and

$$E(\|X\|) = c_2 \cdot a, \tag{29}$$

where c_1 and c_2 are constants with

$$\begin{aligned} c_1 &= 2^{1/2} \cdot \frac{\Gamma((p+1)/2)}{\Gamma(p/2)} \Big/ p \\ c_2 &= 2^{1/2} \cdot \frac{\Gamma((p+1)/2)}{\Gamma(p/2)}, \end{aligned} \tag{30}$$

$\Gamma(\cdot)$ denoting the Gamma function $\Gamma(k) = \int_0^\infty t^{k-1}\,\mathrm{e}^{-t}\,\mathrm{d}t$ $(k > 0)$.

Proof. (i) $E(X)=E((a+b\cdot Y'\tau/\|Y\|)Y)=b\cdot E((Y'\tau/\|Y\|)Y)$. Construct an orthogonal matrix A that satisfies $\tau = A \cdot (1, 0, \ldots, 0)'$, and then let $Z = A'Y$. Obviously, Z still follows standard p-dimension normal distribution, $Z \sim N(0, I_p)$. Then,

$$\begin{aligned} E(X) &= b \cdot E\left(\frac{Y'A \cdot (1, 0, \ldots, 0)'}{\|Y\|} Y\right) \\ &= b \cdot E\left(\frac{Z' \cdot (1, 0, \ldots, 0)'}{\|Z\|} AZ\right) \\ &= b \cdot A \cdot E\left(\frac{Z_1}{\|Z\|} Z\right) \\ &= b \cdot A \cdot E\left(\frac{Z_1 Z_1}{\|Z\|}, \frac{Z_1 Z_2}{\|Z\|}, \ldots, \frac{Z_1 Z_p}{\|Z\|}\right)'. \end{aligned}$$

For the antisymmetry of $Z_i Z_j/\|Z\|$, we know $E(Z_i Z_j/\|Z\|) = 0$ for $i \neq j$. Thus

$$\begin{aligned} E(X) &= b \cdot A \cdot \left(E\frac{Z_1^2}{\|Z\|}, 0, \ldots, 0\right)' \\ &= b \cdot A \cdot (1, 0, \ldots, 0)' \cdot E\frac{Z_1^2}{\|Z\|} \\ &= b \cdot \tau \cdot \frac{1}{p} \cdot E\left(\frac{\sum Z_i^2}{\|Z\|}\right) \\ &= \frac{1}{p} \cdot E(\|Z\|) \cdot b \cdot \tau. \end{aligned}$$

Let $W = \|Z\|^2$, then W has a Chi-Squared distribution χ_p^2, whose p.d.f is

$$g(w) = \frac{1}{2^{p/2}\Gamma(p/2)} \cdot \mathrm{e}^{-p/2} \cdot w^{(p/2)-1}, \quad w > 0.$$

Then, we have

$$\begin{aligned} E(\sqrt{W}) &= \int_0^\infty w^{1/2} \frac{1}{2^{p/2}\Gamma(p/2)} \mathrm{e}^{-w/2} \cdot x^{(p/2)-1}\,\mathrm{d}w \\ &= 2^{1/2} \cdot \frac{\Gamma((p+1)/2)}{\Gamma(p/2)}, \end{aligned} \tag{31}$$

so that,

$$\begin{aligned} E(X) &= b \cdot \frac{1}{p} \cdot E(\|Z\|) \cdot \tau \\ &= 2^{1/2} \cdot \frac{1}{p} \cdot \frac{\Gamma((p+1)/2)}{\Gamma(p/2)} \cdot b \cdot \tau. \end{aligned} \tag{32}$$

(ii)

$$
\begin{aligned}
E\|X\| &= E\left\|\left(a + b\frac{Y'\tau}{\|Y\|}\right) Y\right\| \\
&= E\left\|\left(a + b\frac{Z'(1,0,\ldots,0)'}{\|Z\|}\right) AZ\right\| \\
&= E\left\|\left(a + b\frac{Z_1}{\|Z\|}\right) Z\right\| \\
&= E\left(a + b\frac{Z_1}{\|Z\|}\right)\|Z\| \\
&= aE\|Z\| \\
&= 2^{1/2}\cdot\frac{\Gamma((p+1)/2)}{\Gamma(p/2)}\cdot a. \qquad \square
\end{aligned}
$$

References

[1] P. Hart, R. Duda, Pattern classification and scene analysis, Pattern Classification and Scene Analysis, Wiley, New York, 1974.

[2] P. Hart, T. Cover, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.

[3] L. Devroye, On the inequality of cover and hart in nearest neighbor discrimination, IEEE Trans. Pattern Anal. Mach. Intell. 3 (1981) 75–79.

[4] S.R. Kulkarni, G. Lugosi, S.S. Venkatesh, Learning pattern classification—a survey, IEEE Trans. Inf. Theory 44 (6) (1998) 2178–2206 (TY—JOUR).

[5] T. Wagner, Convergence of the nearest neighbor rule, IEEE Trans. Inf. Theory 17 (5) (1971) 566–571 (TY—JOUR).

[6] R. Bellman, R. Kalaba, On adaptive control processes, IRE Trans. Autom. Control 4 (2) (1959) 1–9 (TY—JOUR).

[7] R. Short II, K. Fukunaga, The optimal distance measure for nearest neighbor classification, IEEE Trans. Inf. Theory 27 (5) (1981) 622–627 (TY—JOUR).

[8] C. Domeniconi, J. Peng, D. Gunopulos, Locally adaptive metric nearest-neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (9) (2002) 1281–1285 (TY—JOUR).

[9] J.H. Friedman, Flexible Metric Nearest Neighbor Classification, The Pennsylvania State University CiteSeer Archives September 24, 1999.

[10] T. Hastie, R. Tibshirani, Discriminant adaptive nearest neighbor classification, IEEE Trans. Pattern Anal. Mach. Intell. 18 (6) (1996) 607–616 (TY—JOUR).

[11] K. Fukunaga, T.E. Flick, An optimal global nearest neighbor metric, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 314–318.

[12] R. Paredes, E. Vidal, A class-dependent weighted dissimilarity measure for nearest neighbor classification problems, Pattern Recognition Lett. 21 (12) (2000) 1027–1036.

[13] J. Peng, D.R. Heisterkamp, H.K. Dai, Lda/svm driven nearest neighbor classification, IEEE Trans. Neural Networks 14 (4) (2003) 940–942 (TY—JOUR).

[14] A. Djouadi, On the reduction of the nearest-neighbor variation for more accurate classification and error estimates, IEEE Trans. Pattern Anal. Mach. Intell. 20 (5) (1998) 567–571 (TY—JOUR).

[15] G. Barna, T. Kohonen, R. Chrisley (Eds.), Statistical pattern recognition with neural networks: benchmarking studies, Proceedings of the IEEE International Conference on Neural Networks, vol. 61–67, San Diego, 1988.

[16] R.R. Hayes, K. Fukunaga, Estimation of classifier performance, IEEE Trans. Pattern Anal. Mach. Intell. 11 (10) (1989) 1087–1101.