Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **3**, May **2017**

The Distance-Weighted K-nearest Centroid Neighbor Classification

Ping Li[1],Jianping Gou[2], Hebiao Yang[2]

[1] : School of Internet of Things
Wuxi Institute of Technology
1600 Gaolang West Road, Binhu District, Wuxi 214121, China
lip@wxit.edu.cn

[2] :School of Computer Science and Telecommunication Engineering
Jiangsu University
301 Xuefu Road, Jingkou District, Zhenjiang 212013, China
goujianping@ujs.edu.cn; yhbjj@ujs.edu.cn

Received May, 2016; revised February, 2017

Abstract. *The k-Nearest Centroid Neighbor rule is one of the effective algorithms in pattern classification. In this paper, with the goal of overcoming the sensitivity issue on the choice of the neighborhood size k and improving the classification performance, two new distance-weighted k-nearest centroid neighbor rules are proposed. According to the geometric distribution and the similarity between the nearest centroid neighbors and the query pattern, the proposed rules mainly employ the new weighted voting function to give weights in the classification voting. In order to verify the classification behavior of the proposed classifiers, we conduct extensive experiments on twelve real data sets, in comparison with the other KNN-based classifiers. Experimental results show that the new classifiers are effective algorithms for the classification tasks, owing to their satisfactory classification performance and robustness over a wide range of k.*
Keywords: Pattern classification, k-nearest neighbor rule, k-nearest Centroid neighbor rule, Distance-weighted voting.

1. **Introduction.** Pattern recognition system is an important part of modern information science and artificial intelligence. It is mainly composed of four parts: data acquisition, data preprocessing, feature extraction and classification decision[1]. As an important part in pattern recognition, the research of the classification decision algorithm has become a hot research topic. In pattern recognition, since the k-nearest neighbor (KNN) rule was first introduced by Fix and Hodges[2], it has been one of the top ten algorithms in data mining[3], and has been widely used in many practical applications, such as image processing, speech recognition and text classification.

The basic rationale for the KNN rule is such that each query pattern is assigned to the class, represented by a majority of its k-nearest neighbors sought from the training set by Euclidean distance. The major characteristic of the KNN rule is its good asymptotic performance. If the number of training samples approaches to infinity, the error rate of the KNN rule is bounded above twice the optimal Bayesian error rate. And when the number of the samples N and the number of neighbors k tend to infinity and $k/N \to 0$, the error rate of the KNN rule approaches to the Bayesian error rate[4]. However, the KNN rule has two main limitations. Firstly, the sensitivity issue on the choice of the neighborhood

size k exists in the KNN rule. If k is too small, the classification result of the query is sensitive to the data sparseness and the noisy, ambiguous or mislabeled points. If k is too large, its neighborhood may include many outliers from other classes[5, 6, 7]. Secondly, by virtue of the majority voting for making decision in the KNN rule, the k neighbors of each query have an identical weight. The ties of vote can easily give rise to the unpromising classification results.

To deal with the problem, Dudani developed a weighted voting scheme, called distance-weighted k-Nearest Neighbor rule (WKNN)[8], with the basic idea of weighting closer neighbors more heavily according to their distances to the query. Gou, et al. developed a new distance-weighted k-nearest neighbor rule (DWKNN)[9, 10] which can deal with the outliers in the local region of a data space, so as to degrade the sensitivity of the choice of k.

In order to improve the classification accuracy, a great many of alternative extensions of the traditional KNN have been developed. Among them, the Nearest Centroid Neighbor rule (NCN) is one of the alternative methods[11, 12, 13]. Based on the concept of the NCN rule, Sánchez developed the k-Nearest Centroid Neighbor rule (KNCN)[12]. Instead of directly selecting k nearest neighbors for a query, KNCN chooses k nearest centroid neighbors that are not only close enough to the query, but also well symmetrically distributed around it. Many experimental studies have indicated that the KNCN rule performs very well in terms of the classification accuracy. However, just like the KNN rule, there are still several problems in the KNCN rule. With the goal of improving the KNCN classification performance, the weighted voting schemes for KNCN have been put forward[14].

Although the KNCN and WKNCN rules can get better classification performance, there are still main problem which will reduce the classification accuracy. This issue in the KNCN and WKNCN is that the far centroid neighbors with more similarities have identical or small contributions for classification. In fact, the far centroid neighbors may be more related to the query pattern, we should give larger weights to them for classification. On the contrary, the similarities among the nearest centroid neighbors with the query pattern are lower, especially the neighbors may be the outliers, we should give smaller weights to them. In order to overcome this problem and improve the classification performance, in this paper, we propose two new classifiers on basis of DWKNN and KNCN, called Distance-weighted k-nearest Centroid Neighbor rules (DWKNCN). In the DWKNCN, we design two weighted voting functions for DWKNCN. The experimental results show the effectiveness of the proposed classifiers in many practical situations.

The rest of this article is organized as follows. In section 2, we briefly summarize the related work. In section 3, we introduce two weighted voting methods for KNCN-based classification. Section 4 presents the experimental results and section 5 offers our conclusion.

2. **Outline of Related Work.**

2.1. **KNN, WKNN and DWKNN.**

The KNN-based classification rule is one of the top ten algorithms in data mining. In the KNN, given a set of training samples and a query, it first finds k nearest neighbors, and then assigns the class label to the query object that has the majority voting among its nearest neighbors[15]. We give a summary of the KNN algorithmic procedure. Let $T = \{(x_i, l_i)\}_{i=1}^{N}$ be the training set with M classes in the m-dimensional feature space, where $x_i \in R^m$, $l_i \in \{c_1, c_2, \ldots c_M\}$, N is the number of training samples. Given a query sample $\bar{x}$, its unkown class $\bar{l}$ is determined as follows:

a) A set of k similar labeled target neighbors for $\bar{x}$ is identified by Euclidean distance. Denote the set $T_k^{NN}(\bar{x}) = \left\{\left(x_i{}^{NN}, l_i{}^{NN}\right)\right\}_{i=1}^{k}$, arranged in an increasing order in terms

of Euclidean distance $d(\bar{x}, {x_i}^{NN})$ between $\bar{x}$ and ${x_i}^{NN}$:

$$d\left(\bar{x}, {x_i}^{NN}\right) = \sqrt{\left(\bar{x} - {x_i}^{NN}\right)^T \left(\bar{x} - {x_i}^{NN}\right)} \tag{1}$$

b) The class label of the query object $\bar{x}$ is predicted by the majority voting of those identified neighbors:

$$\bar{l} = arg\max_{c_j} \sum_{{x_i}^{NN} \in T_k^{NN}(\bar{x})} \delta\left(c_j = {l_i}^{NN}\right) \tag{2}$$

Where $j \in \{1, 2, \ldots M\}$, $\delta\left(c_j = {l_i}^{NN}\right)$ is the Kronecker delta function that takes a value of one if $c_j = {l_i}^{NN}$ and zero otherwise.

In the KNN rule, an implicit assumption that k nearest neighbors of each query share an identical weight is not always appropriate in pattern classification. Dudani first introduced a weighted voting method for KNN, called the WKNN rule[8]. In the WKNN, the closer neighbors are weighted more heavily than the farther ones. The weight w_i for the i-th nearest neighbor of the query $\bar{x}$ is defined as follows:

$$w_i = \begin{cases} \frac{d\left(\bar{x}, {x_k}^{NN}\right) - d\left(\bar{x}, {x_i}^{NN}\right)}{d\left(\bar{x}, {x_k}^{NN}\right) - d\left(\bar{x}, {x_1}^{NN}\right)} & d\left(\bar{x}, {x_k}^{NN}\right) \neq d\left(\bar{x}, {x_1}^{NN}\right), \\ 1 & d\left(\bar{x}, {x_k}^{NN}\right) = d\left(\bar{x}, {x_1}^{NN}\right) \end{cases} \tag{3}$$

Then, the classification result of the query is made by the majority weighted voting:

$$\bar{l} = arg\max_{c_j} \sum_{{x_i}^{NN} \in T_k^{NN}(\bar{x})} w_i \times \delta\left(c_j = {l_i}^{NN}\right) \tag{4}$$

In contrast to WKNN, Gou, et al. introduced a new distance-weighted k-nearest neighbor rule (DWKNN)[8, 9]. It can deal with the outliers in the local region of a data space, in order that the degree of the sensitivity of different choices of k can be degraded. Different from the weights in WKNN, the new weight ${w_i}'$ for the i-th nearest neighbor of the query $\bar{x}$ in DWKNN is defined as follows:

$${w_i}' = \begin{cases} \frac{d\left(\bar{x}, {x_k}^{NN}\right) - d\left(\bar{x}, {x_i}^{NN}\right)}{d\left(\bar{x}, {x_k}^{NN}\right) - d\left(\bar{x}, {x_1}^{NN}\right)} \times \frac{d\left(\bar{x}, {x_k}^{NN}\right) + d\left(\bar{x}, {x_1}^{NN}\right)}{d\left(\bar{x}, {x_k}^{NN}\right) + d\left(\bar{x}, {x_i}^{NN}\right)} & d\left(\bar{x}, {x_k}^{NN}\right) \neq d\left(\bar{x}, {x_1}^{NN}\right), \\ 1 & d\left(\bar{x}, {x_k}^{NN}\right) = d\left(\bar{x}, {x_1}^{NN}\right) \end{cases} \tag{5}$$

And then, the classification result of the query is made by the majority weighted voting:

$$\bar{l}' = arg\max_{c_j} \sum_{{x_i}^{NN} \in T_k^{NN}(\bar{x})} {w_i}' \times \delta\left(c_j = {l_i}^{NN}\right) \tag{6}$$

2.2. **KNCN and WKNCN.** NCN is a good way to choose the nearest neighbors[12, 13]. The basic idea of NCN is that the neighbors are not only close to the query objects as much as possible, but also they are distributed around the query objects as geometrically as possible. For the query object $\bar{x}$, NCN should be subject to two complementary constraints:

a) The distance criterion: the centriod neighbors should be close to $\bar{x}$ as much as possible.
b) The symmetry criterion: the centriod neighbors should be placed around $\bar{x}$ as homogeneously as possible.

The centroid of a set of points $X = \{x_1, x_2, \ldots x_r\}$ can be defined as $x_r^c = \frac{1}{r}\sum_{i=1}^{r} x_i$.

According to the NCN concept, the KNCN rule is introduced by Sánchez [9]. Compared with the KNN, the KNCN predicts the class label of the query object in terms of both the proximity and symmetrical distribution of the neighbors by the majority voting.

Given a query $\bar{x}$, the KNCN classifier can be obtained as follows:

a) Find the k nearest centriod neighbors of a query $\bar{x}$ from the training set T, indicated by $T_k^{NCN} = \left\{x_i^{NCN} \in R^m\right\}_{i=1}^{k}$.
b) Assign $\bar{x}$ to the class c with the greatest voted class among k nearest centroid neighbors in the set T_k^{NCN}(resolve ties randmly), according to Eq.(2)

With the goal of improving the KNCN classification performance, the weighted voting schemes for KNCN have been proposed[14]. It implies a common weighted voting function, i.e., uniform kernel function:

$$w_i^{NCN} = \frac{1}{i}, i = 1, 2, \ldots, k \tag{7}$$

And then, the classification result of the query is made by the majority weighted voting according to Eq.(4) or (6).

3. **The Distance-weighted KNCN Classifiers.**

3.1. **Problem representation.** As we know, the classification rules based on the KNN rule have two limitations: (1) the classification performances is sensitive to choose the neighborhood size k; (2) a majority vote is the simplest method of combining the class labels for the KNN, and the k neighbors of each query have an identical weight. The ties of vote can easily give rise to the unpromising classification results. Moreover, although the weighted voting methods are less sensitive to the choice of k than the k-nearest neighbor rule, their classification results are still affected by the sensitivity of k, owing to the existing outliers in the neighborhood region of k nearest neighbors, especially in small training sample size situations[16, 17, 18].

Due to the distance and symmetry criterions used in the KNCN, it has been observed that the classification importance of centroid neighbors' proximity can be overestimated[14]. Given a NCN neighborhood, some nearest centroid neighbors may be indeed too far from the query pattern, but the query may be located closer to the most distant centroid neighbors. This problem could result in slower or worse classification accuracy.

Motivated by the problems as mentioned above, we can find that the distance of farther nearest centroid neighbors are probably smaller than the closer nearest centroid neighbors[19]. In this situation, on one hand, the farther nearest centroid neighbors may be more effective for the classification result; on the other hand, the closer nearest centroid neighbors may be outliers, it will reduce the classification accuracy. So during the classification process, we should give more weight to highly reliable centroid neighbors while reducing the impact of unreliable centroid neighbors. We assume that the weights of the closer nearest centroid neighbors with the lower classification contribution should be smaller, otherwise the weights of the farther nearest centroid neighbors with the higher classification contribution should be larger. Therefore, we propose two Distance-weighted k-nearest Centroid Neighbor rules (DWKNCN). The proposed methods can not only use the advantage of the geometric distribution of the nearest centroid neighbors, but also use the similarity between the nearest centroid neighbors and the query pattern. Moreover, we can give more contribution with more weight for classification, although the nearest centroid neighbors with more similarities are farther. They can resolve the problem of the sensitivity issue on the choice of the neighborhood size k and the identical weight to each neighbors, so as to improve the classification performance.

Next, we borrow the ideas of the distance-weighted voting method to develop two new voting schemes for KNCN.

3.2. **DWKNCN1.** We design a simple and effective classifier, i.e. DWKNCN1, to attempt to solve the aforementioned problems and improve the classification accuracy.

Let $\bar{T} = \left\{\left({x_i}^{NCN}, {l_i}^{NCN}\right)\right\}_{i=1}^{k}$ be the set of the k-nearest neighbors to the query $\bar{x}$ arranged in an increasing order according to the distance $d(\bar{x}, {x_i}^{NCN})$ between $\bar{x}$ and ${x_i}^{NCN}$, and $\bar{W} = \{\bar{w}_1, \bar{w}_2 \dots \bar{w}_k\}$ be the set of the homologous weights. DWKNCN1 is based on WKNN and KNCN: to give different weights to k nearest centroid neighbors according to their distances, with closer neighbors having greater weights. The weight $\bar{w}_i$ for the i-th nearest centroid neighbor of the query $\bar{x}$ is defined as follows:

$$\bar{w}_i = \begin{cases} \frac{d_{max} - d\left(\bar{x}, {x_i}^{NCN}\right)}{d_{max} - d_{min}} & d_{max} \neq d_{min}, \\ 1 & d_{max} = d_{min} \end{cases} \tag{8}$$

where $d_{max} = max\left\{d\left(\bar{x}, {x_i}^{NCN}\right)\right\}_{i=1}^{k}$, $d_{min} = min\left\{d\left(\bar{x}, {x_i}^{NCN}\right)\right\}_{i=1}^{k}$

And then, we label the query $\bar{x}$ by the weighted voting, the same as Eq.(4) or (6).

$$\bar{l} = arg\max_{c_j} \sum_{\left({x_i}^{NCN}, {l_i}^{NCN}\right) \in \bar{T}} \bar{w}_i \times \delta\left(c_j = {l_i}^{NCN}\right) \tag{9}$$

According to the Eq.(8), we can see that a centroid neighbor with smaller distance is weighted more heavily than one with larger distance: the nearest centroid neighbor with smallest distance gets weight of 1, the nearest centroid neighbor with largest distance among the k neighbors gets weight of 0 and the other centroid neighbors' weights are scaled linearly to the distances in between.

3.3. **DWKNCN2.** Compared with the DWKNCN1, we use the weighted function of Eq.(5) in DWKNN to put forward a new weighted voting method(DWKNCN2) which reduces the weight of each nearest centroid neighbor except the first nearest neighbor. It can keep from giving too much weight to the outliers by reducing the weights of the neighbors in the set of k nearest centroid neighbors for each query.

The new weight $\bar{w}_i'$ for the i-th nearest neighbor of the query $\bar{x}$ is defined as follows:

$$\bar{w}_i' = \begin{cases} \frac{d_{max} - d\left(\bar{x}, {x_i}^{NCN}\right)}{d_{max} - d_{min}} \times \frac{d_{max} + d_{min}}{d_{max} + d(\bar{x}, {x_i}^{NCN})} & d_{max} \neq d_{min}, \\ 1 & d_{max} = d_{min} \end{cases} \tag{10}$$

And then, we label the query $\bar{x}$ by the weighted voting, the same as Eq. (9).

In Eq.(10), the weight of each K-nearest centroid neighbor consists of two parts: the first part is the same as the weight in DWKNCN1, the second one is a new distinct weight. Generally, the similarity between the outliers and the query pattern is relatively small. When the similarity is smaller, the degree of weight reduction should be greater. According to Eq.(10), we can give more weights for the farther nearest centroid neighbors with more similarities, with the purpose of further reducing the weights of the outliers among the k nearest centroid neighbors. It is obvious that the weight $\bar{w}_i'$ is smaller than the weight $\bar{w}_i$ computed by Eq.(8), except the weights of the first and k-th nearest centroid neighbors.

3.4. **The Algorithm.** In summary, the algorithm form of the proposed DWKNCN is described in Algorithm 1. Before the start of the algorithm, we prepare the input as follows:

$\bar{x}$: the query pattern, k: the neighborhood, $T = \{x_i \in R^m\}_{i=1}^{N}$: the training samples,

$T_i = \{x_{ij} \in R^m\}_{j=1}^{N_i}$: the subset of T in each class, M: the numbers of classes,

$\{c_1, c_2, \dots c_M\}$: class labels, $N_1, N_2, \dots N_M$: the number of training samples in T_i

Then we can predict the class label of a query pattern by the DWKNCN rule.

Algorithm 1: The proposed DWKNCN algorithms

Step 1: Compute the distances of training samples in each class c_i to the query $\bar{x}$
for j=1 to N_i do
$(\bar{x}, x_{ij}) = \sqrt{(\bar{x} - x_{ij})^T (\bar{x} - x_{ij})}$
End for
Step 2: Find the first nearest centroid neighbor of $\bar{x}$ in each class c_i
$[min\_dist, min\_index] = min(d(\bar{x}, x_{ij}))$
$x_{i1}^{NCN} = x_{min\_index}$
Set $C_i^{NCN}(x) = \{x_{i1}^{NCN} \in R^m\}$
Step 3: Search k nearest centroid neighbors of x except the first one in class c_i, say $T_{ik}^{NCN}(x) = \left\{x_{ij}^{NCN} \in R^m\right\}_{j=1}^{k}$
For j=2 to k do
Set $S_i(x) = T_i - C_i^{NCN}(x)$, $S_i(x) = \{x_{il} \in R^m\}_{l=1}^{L_i(x)}$, $L_i(x) = length(S_i(x))$
Compute the sum of the previous $j-1$ nearest centroid neighbors.
$sum_i^{NCN}(x) = \sum_{r=1}^{j-1} x_{ir}^{NCN}$
Calculate the centroids in the set S_i for x and the distance between them.
For l=1 to $L_i(x)$ do
$x_{il}^c = \frac{1}{j}(x_{il} + sum_i^{NCN}(x))$
$d_{il}^c(x, x_{il}^c) = \sqrt{(x - x_{il}^c)^T (x - x_{il}^c)}$
End For
Find the j-th nearest centroid neighbors.
$[min\_index^{NCN}, min\_dist^{NCN}] = min(d_{il}^c(x, x_{il}^c))$
$x_{ij}^{NCN} = x_{min\_index^{NCN}}$
Add x_{ij}^{NCN} to the set $C_i^{NCN}(x)$.
End For
Set $T_{ik}^{NCN}(x) = C_i^{NCN}(x)$.
Step 4: Calculate the weights of k nearest centroid neighbors.
For i=1 to k do
$d_{\max} = \max\left(d\left(x, {x_i}^{NCN}\right)\right)$, $d_{\min} = \min\left(d\left(x, {x_i}^{NCN}\right)\right)$
If $d_{max} = d_{min}$ then
$\bar{w}_i = 1$
Else
$\bar{w}_i = \frac{d_{max} - d\left(\bar{x}, {x_i}^{NCN}\right)}{d_{max} - d_{min}}$ or $\bar{w}_i = \frac{d_{max} - d\left(\bar{x}, {x_i}^{NCN}\right)}{d_{max} - d_{min}} \times \frac{d_{max} + d_{min}}{d_{max} + d(\bar{x}, {x_i}^{NCN})}$
End if
End For
Step 5: Assign the class label to $\bar{x}$ by the weighted voting.
$\bar{l} = arg \max_{c_j} \sum_{\left({x_i}^{NCN}, {l_i}^{NCN}\right) \in T_k^{NCN}(\bar{x})} \bar{w}_i \times \delta\left(c_j = {l_i}^{NCN}\right)$

4. **Experimental Results.** In the pattern classification, the accuracy rate is one of the most important measures to evaluate the algorithm performance. In order to study the classification behavior of the proposed classifiers, we conduct many comparative experiments on twelve real data sets to compare with the aforementioned classifiers. These real data sets are selected from the UCI machine learning repository[20]. We consider the

classification performance through two aspects[21]: (1) the highest accuracy rates and the corresponding values of k, (2) the accuracy rates with varying the neighborhood size k.

4.1. **Experimental Data Sets.** The twelve real data sets in our experiments are selected from the UCI machine learning repository. For short, among these data sets, the abbreviated names for Image Segmentation, Parkinsons, Waveform, Landsat Satellite, Transfusion are Image, Park, Wave, Landsat and Trans. The overall properties of these data sets are described in Table 1, including dataset names, sample instances, feature space dimensions, class numbers and the number of testing samples.

TABLE 1. The data sets used in the experiments

Dataset	Instances	Dimensions	Classes	Training set	Testing set
Wine	178	13	3	118	60
Iris	150	4	3	100	50
Seeds	210	7	3	140	70
DUser	403	6	4	260	143
Image	2310	19	7	1400	910
Ecoli	336	7	8	216	120
Park	195	22	2	130	65
Wave	5000	40	3	3000	2000
Landsat	6435	36	6	4290	2145
Musk	476	166	2	317	159
Pen	10992	16	10	7323	3669
Trans	748	4	2	498	250

The attributes of the data sets are both numeric, which can make us directly use Euclidean distance to calculate the similarity between two samples. Among the 12 data sets, there are 3 data sets that belong to two-class classification tasks, while the others are multi-class classification tasks. We conduct experiments by 20-fold cross validation. The training sets are randomly taken from each data set, while the remaining samples are the testing sets. We do experiments 20 times and obtain 20 different training and testing sets for performance evaluation on each data set. Twenty averaged classification accuracy with 95% confidence is achieved as the final performance. We should note that the values of the neighborhood size k in each data set vary from 1 to 15. The optimal or sub-optimal value of k on each data set which obtains the highest accuracy rate is chosen within the interval.

4.2. **Experimental Comparisons.** We thoroughly explore the performance of our two proposed DWKNCNC classifiers, compared to KNN, WKNN, DWKNN, KNCN, WKNCN. The average best accuracy rates, the corresponding standard deviations and values of k of each method are shown in Table 2.

Among these methods, the best classification results are marked bold-face type in the table. As shown in Table 2, we can clearly see that the proposed DWKNCN1 and DWKNCN2 classifier always perform better than other five methods on the real data sets, except Wave and Trans data set.

To further explore the superiority of the proposed DWKNCN rules, the average classification results of the seven methods with varying the neighborhood size k on each data set are shown in Figure 1 and 2.

It is clear that the performance of DWKNCN is almost superior to the other methods, especially when the value of k is large. In consequence, we can draw a conclusion that

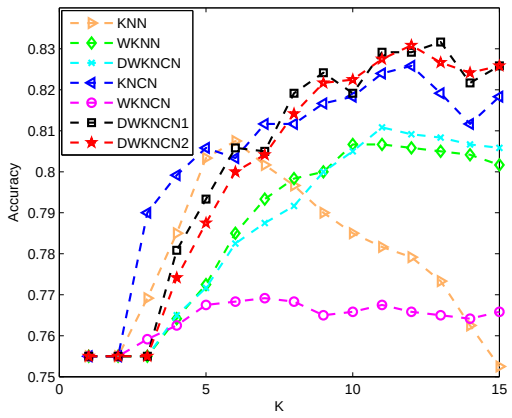
TABLE 2. The best average accuracy rates (%) of each method with the corresponding stds and value of k in the parentheses for the 12 UCI data set (the best recognition performance is described in bold-face on each data set)

Datasets	KNN	WKNN	DWKNN	KNCN	WKNCN	DWKNCN1	DWKNCN2
Wine	80.75±1.74 (6)	80.67±2.09 (10)	81.08±2.17 (11)	82.58±2.22 (12)	76.92±0.45 (7)	**83.17±2.87 (13)**	**83.08±2.89 (12)**
Iris	96.80±0.57 (10)	96.40±0.60 (10)	96.60±0.68 (10)	96.40±0.43 (9)	95.40±0.10 (7)	**97.00±0.66 (11)**	**97.00±0.57 (11)**
Seeds	90.27±0.83 (7)	90.40±1.07 (13)	90.40±1.10 (11)	90.13±0.83 (9)	88.20±0.20 (13)	**90.67±1.13 (12)**	**90.57±1.08 (12)**
DUser	85.11±1.46 (4)	85.22±1.20 (7)	85.33±1.25 (7)	85.89±1.18 (9)	82.67±0.21 (9)	**86.11±1.45 (12)**	**86.44±1.63 (13)**
Image	95.96±1.88 (1)	96.09±0.54 (5)	96.06±0.32 (5)	96.35±0.37 (5)	96.09±0.09 (11)	**96.36±0.20 (8)**	**96.52±0.22 (9)**
Ecoli	84.72±1.21 (3)	86.06±2.10 (13)	85.78±2.05 (13)	86.00±1.63 (9)	81.44±0.20 (8)	**86.33±2.13 (10)**	**86.67±2.28 (10)**
Park	82.41±0.62 (8)	82.77±0.73 (13)	82.70±0.75 (13)	82.48±0.78 (8)	81.61±0.23 (12)	82.63±0.63 (11)	**82.92±0.80 (11)**
Wave	**82.67±2.41 (15)**	81.11±2.46 (15)	81.06±2.46 (15)	82.22±2.59 (14)	80.50±2.26 (11)	80.89±2.46 (15)	80.89±2.51 (11)
Landsat	90.80±3.20 (1)	90.80±1.00 (1)	90.85±0.9 (4)	92.15±1.34 (6)	91.50±0.23 (6)	**92.20±0.69 (5)**	**92.25±0.60 (5)**
Musk	86.73±2.58 (4)	86.86±0.67 (6)	86.92±0.55 (8)	88.05±0.77 (3)	89.69±1.58 (3)	**89.94±1.61 (11)**	**90.19±1.70 (11)**
Pen	95.00±4.77 (1)	95.00±1.17 (1)	95.00±0.41 (1)	95.19±2.28 (4)	94.55±0.40 (9)	95.00±0.97 (1)	**95.59±0.90 (4)**
Trans	77.96±2.32 (14)	76.08±1.94 (14)	75.52±1.86 (15)	**78.52±2.54 (13)**	76.28±2.15 (13)	76.32±2.13 (14)	76.04±2.08 (15)

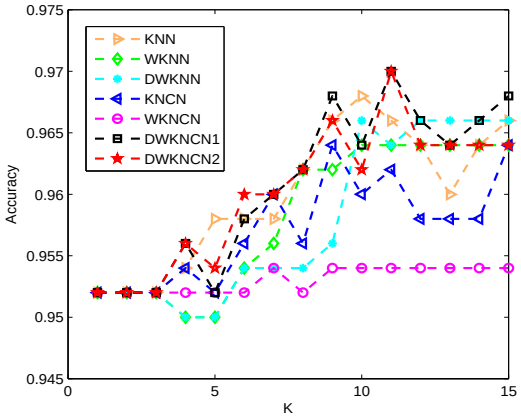
the proposed DWKNCN classifiers have the robustness to the sensitivity choices of the neighborhood size k with good performance to some degree.

In order to further study on classification performance of the proposed DWKNCN, we obtain the classification results about a given query pattern on DUser data set given the value of k in the WKNCN and DWKNCN classifiers. The given query pattern $\bar{x}$ originally belongs to class 1. Find the k nearest centriod neighbors of the given query pattern $\bar{x}$ from the training set, $\bar{x}$ is wrongly classified by WKNCN when $k = 1, \ldots, 8$ but $\bar{x}$ is correctly classfied by DWKNCN when $k = 5, 6, 7, 8$. Table 3 illustrates the distances between each nearest centroid neighbor and the given query pattern, the labels and the corresponding weights of nearest centroid neighbors on DUser data set when $k = 8$.

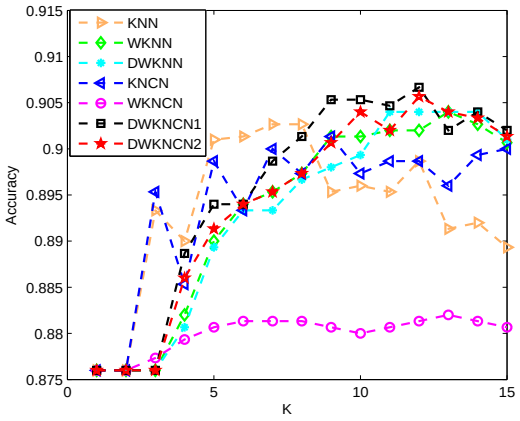
According to Eq. (7), (8) and (10), we can calculate the weights of each nearest centroid neighbor in WKNCN, DWKNCN1 and DWKNCN2 rule, i.e., w_i, $\bar{w}_i$ and $\bar{w}_i{}'$. From Table 3, we can easily find the weights of the 2^{th}, 3^{th} and 5^{th} nearest centroid neighbor using DWKNCN1 and DWKNCN2 rules are greater than the weights using WKNCN. Just in the time the labels of these nearest centroid neighbors are as same as the label of the query pattern. On the other hand, the label of the farthest nearest centroid neighbor, i.e.,



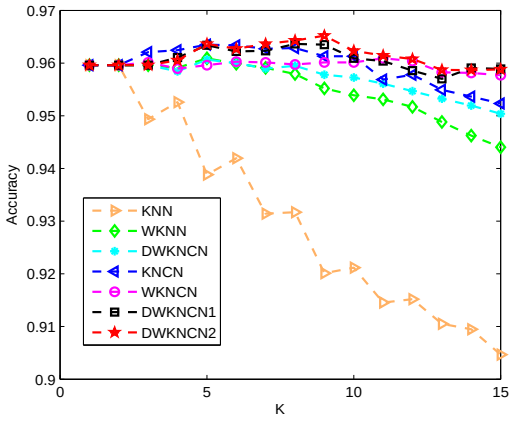
(a)Wine.



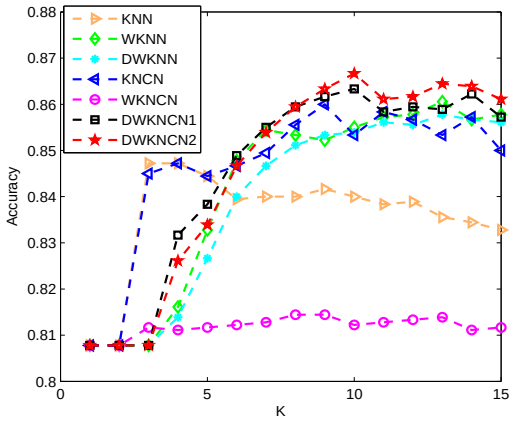
(b)Iris.



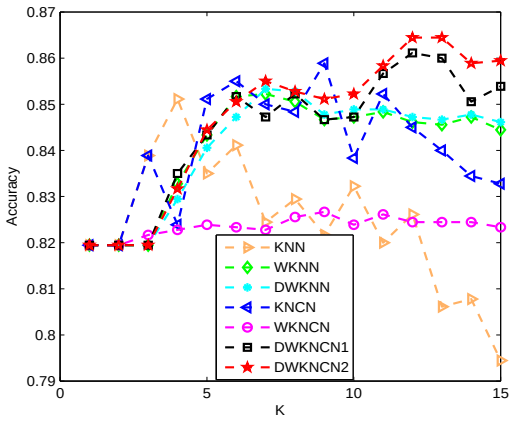
(c)Seeds.



(d)Image.

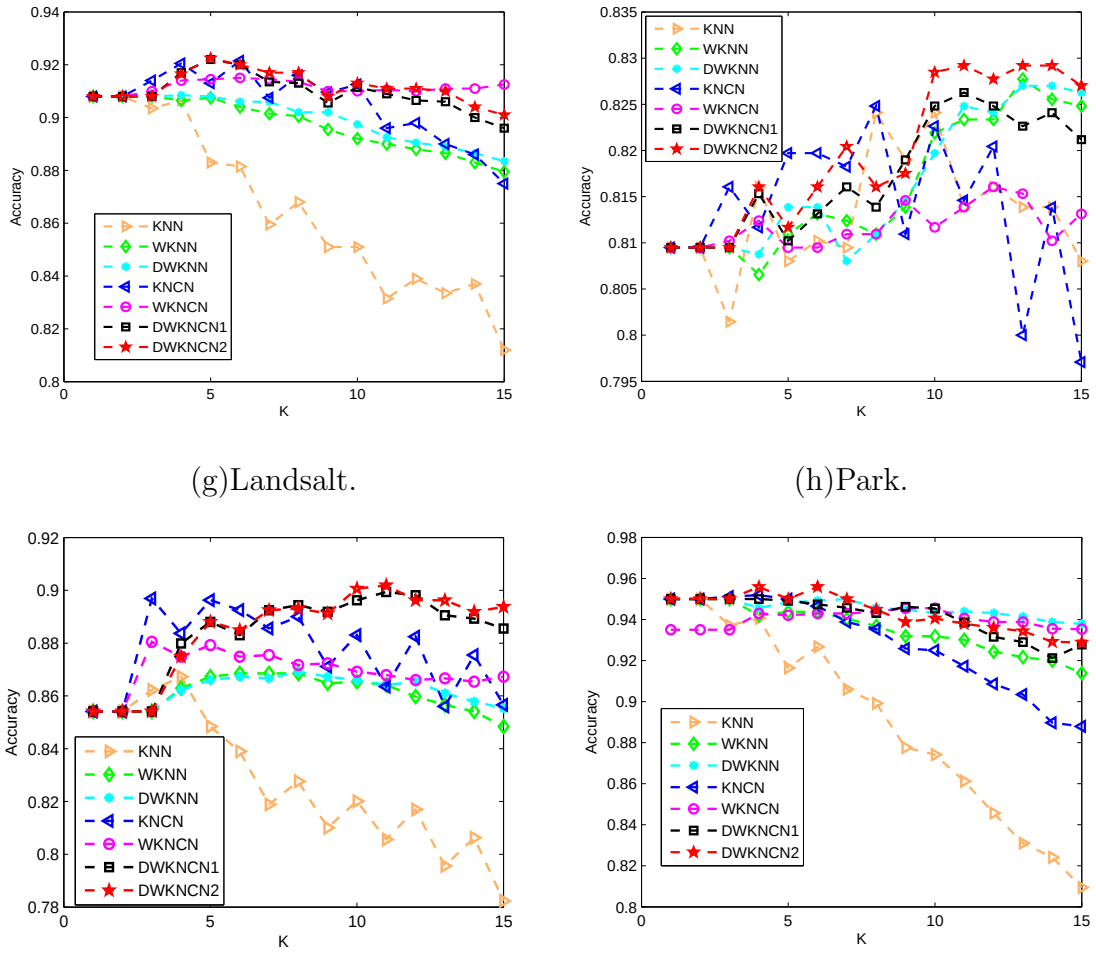


(e)Ecoli.



(f)Duser.

FIGURE 1. The classification accuracies via the neighborhood size k on each data set.



(g)Landsalt. (h)Park.

(i)Musk. (k)Pen.

FIGURE 2. The classification accuracies via the neighborhood size k on each data set.

the 6^{th} nearest centroid neighbor is inconsistent with the query pattern, the weights of two DWKNCN rules are zero, as a result, it greatly reduces the possibility of inconsistent value. Consequently the classification result using WKNCN is wrong, the results using DWKNCN1 and DWKNCN2 are right. The reason for the aforementioned results is that when the close centroid neighbors have too large distances or the far centroid neighbors have too small distances, the weighted voting scheme of WKNCN is not suitable. In our proposed classification, the distances are considered in the weighted voting scheme, the experimental results show it outperforms the others.

5. **Conclusions.** In this article, we propose new distance-weighted k-nearest centroid neighbor classification rules which are based on KNCN and WKNCN. The new classifiers aim at improving the classification performance. Compared to the other k-nearest neighbor classifiers, the experiments of the proposed method are conducted on 12 real data sets. The experimental results suggest that the new classifiers outperform the others, especially in case of a large value of neighborhood size k. In the proposed classification methods, our focus is how to give weights when the close centroid neighbors have too large distances or the far centroid neighbors have too small distances. Through the comprehensive analysis, it suggests that the proposed classifiers have the following strengths: a) when the

TABLE 3. The weights of WKNCN, DWKNCN1 and DWKNCN2 rule, the distance and label of each nearest centroid neighbor when $k = 8$ and the classification results on DUser data set(The symbols $'\heartsuit'$ and $'\diamondsuit'$ denote the labels of the class 1 and 2. The symbols $'\times'$ and $'\surd'$ indicate the wrong and right classification. The i denotes the i^{th} nearest centroid neighbor, the d_i denotes the distance between a query pattern and the i^{th} nearest centroid neighbor, the l_i denotes the label of the i^{th} nearest centroid neighbor. The w_i, $\bar{w}_i$ and $\bar{w}_i{}'$ denote the weights of three methods. The specific weights are describes in bold-face in table 3)

i	d_i	l_i	The weights of three methods		
			w_i(WKNCN)	$\bar{w}_i$(DWKNCN1)	$\bar{w}_i{}'$(DWKNCN2)
1	0.1565	$\diamondsuit$	1	1	1
2	0.2022	$\heartsuit$	0.5	**0.7901**	**0.7274**
3	0.2653	$\heartsuit$	0.33	**0.5002**	**0.4151**
4	0.2791	$\diamondsuit$	0.25	0.4369	0.3549
5	0.2142	$\heartsuit$	0.2	**0.7349**	**0.6628**
6	0.3742	$\diamondsuit$	0.1667	**0**	**0**
7	0.3419	$\heartsuit$	0.1429	0.1484	0.1100
8	0.3356	$\diamondsuit$	0.125	0.1775	0.1328
The classification result			$\diamondsuit(\times)$	$\heartsuit(\surd)$	$\heartsuit(\surd)$

close centroid neighbors have too large distances or the far centroid neighbors have too small distances, the weights of our weighted voting scheme are relatively suitable than the others. b) it is more robust to the neighborhood size k with the preferable performance. Based on our study, it can be concluded that our proposed classifiers can be effectively used in the field of pattern classification.

Acknowledgment. This work was supported in part by National Natural Science Foundation of China (Grant No. 61502208), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 14KJB520007), China Postdoctoral Science Foundation (Grant No. 2015M570411), Natural Science Foundation of Jiangsu Province of China (Grant No. BK20150522), Senior Visiting Scholar Project of Jiangsu Higher Vocational College of China(Grant No. 2015FX082) and Research Foundation for Talented Scholars of JiangSu University (Grant No. 14JDG037). The authors would like to thank to the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] S.Theodoridis, *Pattern Recognition(4th Editon)*, Electronic Industry Press, China, 2006.
[2] E. Fix, J. L. Hodges, Discriminatory Analysis, Nonparametric discrimination: ConsistencyProperties, *Technique Report No. 4 USAF School of Aviation Medcine*, Randolf Field Texas, 1951.
[3] X. D. Wu, V. Kumar et al., Top 10 algorithems in data mining, *Knowledge Information System*, vol. 14, pp. 1-37, 2008.
[4] T. Wagner, Convergence of the nearest neighbor rule, *IEEE Transations of Information Theory*, vol. 17, no. 5, pp. 566-571, 1971.
[5] D. M. Garcia, J. G. Gutirrez, J.C.R Santos, An-evolutionary voting for k-nearest neighbours, *Expert System with Applications*, vol. 43, pp. 9-14, 2016.
[6] Y. Abraham, S. Hanoch, K-nearest neighbors optimization-based outlier removal,*Journal of Computational Chemistry*, vol. 36, no. 8, pp. 493-506, 2015.
[7] X. X. Wang, L. Y. Ma, A Compact K Nearest Neighbor Classification for Power Plant Fault Diagnosis,*Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 3, pp. 508-517, 2014.

[8] S.A.Dudani, The Distance-weighted k-Nearest Neighbor Rule, *IEEE Transations On System Man and Cybernetics*, vol. SMC-6, pp. 325-327, 1976.
[9] J.P.Gou, L.Du, Y.H.Zhang, T.S.Xiong, A new Distance-weighted k-nearest Neighbor Classifier,*Journal of Information & Computational Science*, vol. 9, no. 6, pp. 1429-1436, 2012.
[10] J. P. Gou, T. S. Xiong, Y. Kuang, A Novel Weighted Voting for K-Nearest Neighbor Rule, *Journal of Computers*, vol. 6, no. 5, pp. 833-840, 2011.
[11] B. B. Chaudhuri, A new definition of neighbourhood of a point in multi-dimensional space,*Pattern Recognition Lett.*, vol. 17, no. 1, pp. 11-17, 1996.
[12] J. S. Sánchez, F.Pla, F.J.Ferri, On the user if neighbourhoodbased non-parametric classifiers, *Pattern Recognition Lett.*, vol. 18, pp. 1179-1186, 1997.
[13] J.S . Sánchez, A. I. Marqés, Enhanced neighborhood specification for pattern classification, *Pattern recognition and string matching*, Kluwer Academic Publishers, 2002.
[14] J. P. Gou, L. Du, T.S. Xiong, Weighted K-nearest Centroid Neighbor Classification, *Journal of Computational Information Systems*, vol. 8, no. 2, pp. 851-860, 2012.
[15] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification,*IEEE Transations of Information Theory*, vol. 13, no. 1, pp. 21-27, 1967.
[16] M.Zhao, J.C.Chen, Improvement and Comparison of Weighted k Nearest Neighbors Classifiers for Model Selection,*Journal of Software Engineering*, vol. 10, no. 1, pp. 109-118, 2016.
[17] E. Chavez, M. Graff, G.Navarro, E. S. Tellez, Near neighbor searching with K nearest references,*Information Systems*, vol. 51, pp. 43-61, 2015.
[18] J.W.Yoon, N.Friel, Efficient model selection for probabilistic K nearest neighbor classification, *Neurocomputing*, vol. 149, pp. 1098-1108, 2015.
[19] J.S.Pan, Y.L.Qiao,S.H.Sun, A Fast K Nearest Neighbors Classification Algorithm, *IEICE Transactions on Fundamentals of Electronics, Communication and Computer Sciences*, vol. E-87-A, no. 3, pp. 961-963, 2004.
[20] A. Frank, A. Asuncion, UCI machine learning repository, *<http://archive.ics.uci.edu/ml/>.Irvine, CA: University if California, School of Information and Computer Science*, 2010.
[21] Y. Xu, Q. Zhu et al., Coarse to fine K nearest neighbor classifier,*Pattern Recognition letters*,vol. 34, pp. 980-986, 2013.