A Novel Weighted Voting for K-Nearest Neighbor Rule

Jianping Gou
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, P. R. China
Email: cherish.gjp@gmail.com

Taisong Xiong and Yin Kuang
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, P. R. China
College of Computer Science, Sichuan University, Chengdu, 610065, P. R. China
Email: xiongtaisong@gmail.com, yingkuang@163.com

***Abstract*—K-nearest neighbor rule (KNN) is the well-known non-parametric technique in the statistical pattern classification, owing to its simplicity, intuitiveness and effectiveness. In this paper, we firstly review the related works in brief and detailedly analyze the sensitivity issue on the choice of the neighborhood size *k*, existed in the KNN rule. Motivated by the problem, a novel dual weighted voting scheme for KNN is developed. With the goal of overcoming the sensitivity of the choice of the neighborhood size *k* and improving the classification performance, the proposed classifier mainly employs the dual weighted voting function to reduce the effect of the outliers in the *k* nearest neighbors of each query object.**

To verify the superiority of the proposed classifier, the experiments are conducted on one artificial data set and twelve real data sets, in comparison with the other classifiers. Experimental results suggest that our proposed classifier is an effective algorithm for the classification tasks in many practical situations, owing to its satisfactory classification performance and robustness over a wide range of *k*.

***Index Terms*—*K*-nearest neighbor rule, Weighted voting, Distance-weighted *k*-nearest neighbor rule**

I. Introduction

Since the *k*-nearest neighbor (KNN) rule was firstly introduced by Fix and Hodges [1], it has been one of the well-known and widely used nonparametric classifiers in pattern classification. The basic idea of this rule is that an unclassified object is assigned to the class, represented by a majority of its *k* nearest neighbors in the training set. The main characteristic of the rule is its good asymptotic performance: (a) the error rate of the *k*-nearest neighbor rule asymptotically approximates the optimal Bayesian error rate if the number of training objects approaches infinity, and (b) if k=1, its asymptotic error rate is bounded above by twice the optimal Bayesian error rate [2, 3, 4]. The distance-weighted *k*-nearest neighbor rule (WKNN), which weights more heavily close neighbors based on their distances to the query object, was proposed by Dudani [5]. It is a weighted voting method as an improvement for KNN. Nevertheless, the number of the training samples is usually too small to obtain a good asymptotic performance, which often leads to dramatic degradation of classification performance in the KNN and WKNN rules.

Currently, one major outstanding issue in the KNN rule, yet to be resolved in the research community, is the sensitivity of choice of the neighborhood size *k* [6]. The problem intrinsically results from the estimate of the conditional class probabilities from samples in a local region of data space, which contains *k* nearest neighbors of each query object. Because of the radius of the region determined by the distance of the *k*-furthest neighbor, the results of conditional class probabilities depend on the selection of neighborhood size *k*. If *k* is very small, the local estimate tends to be very poor, due to the influence of the data sparseness and the noisy, ambiguous or mislabelled points. Obviously, we can increase *k* and take into account a large region around the query object in order to smooth the estimate. However, a large value of *k* easily decreases the estimate and results in the dramatic degradation of the performance, because more outliers from the other classes are introduced. To address the issue, the related research works have been done to improve the classification performance of the KNN classifier [5, 7, 8, 9, 10].

Motivated by the issue of sensitivity of choice of *k*, we propose a new dual weighted voting method, based on the distance-weighted nearest neighbor rule, in order to reduce the influences of the sensitivity of *k* on pattern classification and improve the performance of KNN and WKNN. In this new rule, we employ the dual weights of *k* nearest neighbors to determine the class of the query object by the majority weighted voting. The experimental results show the effectiveness of the proposed classifier in such practical situations.

II. The Related Works

A. The KNN classifier

In this section, we review the related works in brief. First of all, we discuss the *k*-nearest neighbor rule.

doi:10.4304/jcp.6.5.833-840

The *k*-nearest neighbor rule (KNN), also called the majority voting *k*-nearest neighbor, is one of the oldest and simplest non-parametric techniques in the pattern classification literature. In this rule, a query pattern is assigned to the class, represented by a majority of its *k* nearest neighbors in the training set. As a matter of fact, the NN rule is the simplest and most special form of the KNN rule, when *k*=1. No matter what the distance measures are employed in pattern classification, it has been proved that the asymptotic error rate of the KNN rule approaches the optimal Bayesian error rate, when both the number *n* of the samples and the number *k* of neighbors tend to infinity, i.e., $k/n \to \infty$. If *k*=1, the error rate of the NN rule is bounded by at most twice the optimal Bayesian error rate [2, 3, 4].

Following the *k*-nearest neighbor rule, we give a high-level summary of the nearest neighbor classifier. Let $T = \{x_1,...,x_N\}$ be the training set, i.e., a corpus of training vectors or class-labeled points (x_i, c_j). The training vectors $x_i \in R^m$ are vectors in the *m*-dimensional feature space, $1 \le i \le N$, and c_j, $j = 1,..., N_c$ are their corresponding class labels (typically $N > N_c$). Given a query object (x', c') randomly, its unknown class c' is determined as follow:

a) select the set $T' = \{x_1^{NN},...,x_k^{NN}\}$, the set of *k* nearest training objects to the selected query object x', arranged in an ascending order in terms of the distance (or dissimilarity) measure between x_i^{NN} and x', $d(x', x_i^{NN})$. And $d(x', x_i^{NN})$ is popularly computed by Euclidean distance metric:

$$d(x', x_i^{NN}) = \left\| x' - x_i^{NN} \right\|_{L_2} \quad . \tag{1}$$

b) assign the class label to the query object x', based on the majority voting class of its nearest neighbors:

$$c' = \arg\max_{c} \sum_{x_i^{NN}, \in T'} I(c = c_i^{NN}) \cdot \tag{2}$$

where c is a class label, c_i^{NN} is the class label for the *i*-th neighbor among *k* nearest neighbors of the query object. $I(c = c_i^{NN})$, an indicator function, takes a value of one if the class c_i^{NN} of the neighbor x_i^{NN} is the same as the class c and zero otherwise. Note that the Euclidean distance metric is often chosen for computing the dissimilarities of the nearest neighbors. But the distance measures are variant according to the practical problem at hand.

B. The WKNN classifier

As mentioned above, the KNN classifier assigns to an unclassified object the class label, which has the majority voting amount in the set of its *k* nearest neighbors. However, this rule implicitly supposes that *k* nearest neighbors have an identical weight in making decision for classification, regardless of their distances to the query object, and also neglects that the nearest neighbor close to the query pattern should contribute more to the classification. It is intuitively appealing to define a weighted voting method for giving *k* nearest neighbors to different weights, depending on their corresponding distances to the object in question.

To weigh the closer neighbors more heavily than the farther ones, Dudani firstly proposed a weighted voting rule, called the distance-weighted KNN rule (WKNN), in which the votes of the different members of the k nearest neighbor set are weighted by a function of their distances to the query object [5].

This scheme is defined as follow:

a) select the set $T' = \{x_1^{NN},...,x_k^{NN}\}$, the same as the representation of the KNN classifier.

b) assign a weight w_i to the *i*-th nearest neighbor x_i^{NN}, using the distance-weighted function.

$$w_i = \begin{cases} \dfrac{d_k^{NN} - d_i^{NN}}{d_k^{NN} - d_1^{NN}}, & d_k^{NN} \ne d_1^{NN}, \\ 1 \qquad\qquad , & d_k^{NN} = d_1^{NN}. \end{cases} \tag{3}$$

where d_i^{NN} is the distance of the *i*-th nearest neighbor to the query object, d_1^{NN} is the distance of the nearest neighbor, and d_k^{NN} is the distance of the *k*-furthest neighbor.

c) assign the query object a majority weighted voting class label $c_{j_{\max}}$ using the following rule:

$$c_{j_{\max}} = \arg\max_{c_j} \sum_{x_i^{NN} \in T'} w_i \times I(c = c_i^{NN}) \quad . \tag{4}$$

With (3), a neighbor with smaller distance is weighted more heavily than one with greater distance: the nearest neighbor gets weight of 1, the furthest neighbor a weight of 0 and the other neighbors' weights are scaled linearly to the interval in between.

In addition, we here discuss another weighted voting function, called uniform function in [10], for KNN (UWKNN). The weight w_i lies on the rank of the nearest neighbor x_i^{NN} rather than the corresponding distance d_i^{NN} to the query object.

$$w_i = \frac{1}{i} \qquad i = 1,...,k \quad . \tag{5}$$

where w_i takes only values from 1 to $1/k$. It keeps from giving too much weight to the proximate nearest neighbor and needs less computation time than the others, but can not adjust the distribution of weights, depending on nearness of the neighbors to the query object.

III. The Proposed Weighted Voting Method

A. The motivation

As discussed in section II, the KNN rule and WKNN rule are the most well-known nonparametric classifiers in pattern classification. Nevertheless, the issue of the sensitivity of the selection of the neighborhood size *k* largely affects the classification performance of the KNN and WKNN rules at present [6], thus, we will address the problem in this article.

For KNN rule, the choice of the optimal or sub-optimal *k* is greatly affected by the finite sample space in the practical problem. The issue is mainly generated by two reasons: (i) if *k* is too small, the classification result of the query is sensitive to the data sparseness and the noisy, ambiguous or mislabelled points; (ii) if *k* is too large, its neighborhood may include many points (outliers) from other classes. Clearly, the classification performance of the *k*-nearest neighbor classifier is very sensitive to the selected value of *k*. Moreover, a majority vote is the simplest method of combining the class labels for KNN, but this can be a problem if the nearest neighbors vary widely in their distances and the closer ones more reliably indicate the class of the query pattern. In order to deal with the sensitivity of the choice of *k*, a great variety of weighted voting methods for KNN have been developed, for instance, in [6, 11, 12].

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*. Dudani [2] firstly proposed a weighted voting method, called the distance-weighted KNN rule (WKNN), and then it has been demonstrated that WKNN is very much superior to the other weighted voting methods through the empirical comparison [13, 14]. However, the classification performance of WKNN still substantially suffers from the sensitivity of the selection of *k*, owing to the existing outliers in the set of *k* nearest neighbors, particularly in small training sample size situations [15]. In addition, the class distribution of a data set has a dramatic effect on the performance of WKNN and KNN. When the class distribution of a data set is irregular and imbalanced, that is, some classes have rather more training samples than the others, they are not robust to the sensitivity of the selection of *k*. In such cases, the estimate of each conditional class probability of the query pattern in a local region of the data space is to be very unreliable, because the query may be largely subject to the outliers from other classes, and the sum weight of each class in the *k* nearest neighbors is not suitable to make the decision for pattern classification.

Motivated by the problem as mentioned above and weighted voting method, we proposed a dual weighted voting method for KNN rule (DWKNN). In this new rule, the weight for each nearest neighbor in the WKNN rule is replaced by the corresponding dual weight. Compared to WKNN, the new rule reduces the weight of each nearest neighbor except the first closest and *k*-furthest nearest neighbors. It can solve the problem of the sensitivity of neighborhood size *k* without influencing the classification performance, especially in the case of the irregular class distribution of data sets and small sample size. Therefore, it is robust to the sensitivity of choice of *k* and improves the classification performance. The experimental results show the effectiveness of the new classifier in such practical situations.

B. The proposed dual weighted voting for KNN rule

In order to address the sensitivity of selection of *k*, a simple and effective classifier, a dual weighted voting *k*-nearest neighbor rule (DWKNN), is proposed. The new rule can overcome the influence of the sensitivity and improve the classification performance of KNN and WKNN. Let $x_1^{NN},\dots,x_k^{NN}$ be the *k*-nearest neighbors to the query object x', and $d_1^{NN},\dots,d_k^{NN}$ be the corresponding distances, arranged in an increasing order, in terms of the dissimilarity between x_i^{NN} and x'. The DWKNN rule is based on both the distance-weighted voting and uniform weighted voting functions: to give different weights to the *k*-nearest neighbors depending on their distances and their ranks of the nearest neighbors, with closer neighbors having greater weights. And then, the proposed dual weighted voting function is defined as follow:

$$w_i = \begin{cases} \dfrac{d_k^{NN} - d_i^{NN}}{d_k^{NN} - d_1^{NN}} \times \dfrac{1}{i}, & d_k^{NN} \neq d_1^{NN}, \\ 1 \quad , & d_k^{NN} = d_1^{NN}. \end{cases} \tag{6}$$

In the following, we predict the class of the query object by the majority weighted voting, the same as (4).

$$c_{j_{\max}} = \arg\max_{c_j} \sum_{x_i^{NN} \in T'} w_i \times I(c = c_i^{NN}) \ . \tag{7}$$

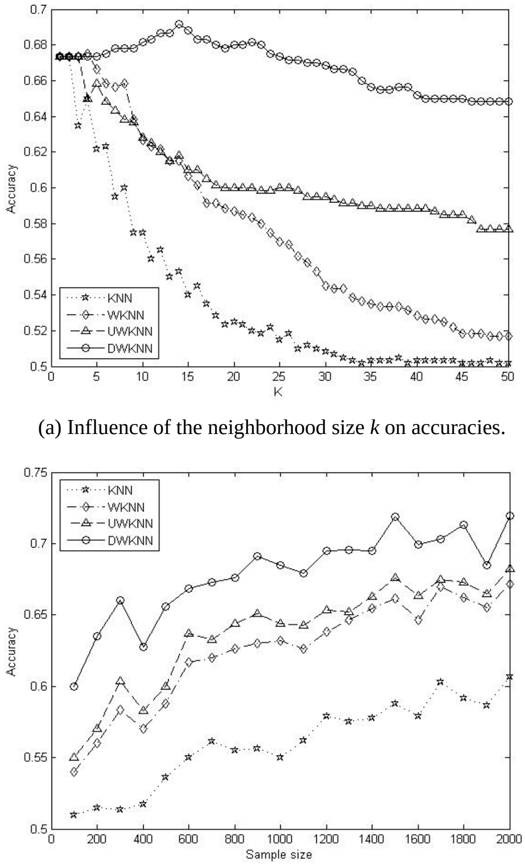
It is clear that the dual weight of each nearest neighbor consists of two parts: the first part is same as the weight of the WKNN rule, the second is the weight of the UWKNN rule. As can be seen from (6), it is obvious that the dual weight w_i is less than the weight computed by (3) and (5), except the weights of the first and *k*-further nearest neighbors. As a consequence, the corresponding neighbor x_i^{NN} has less influence on the classification performance of the query object. We can see that the new rule not only adjusts the distribution of the weights according to the nearness of neighbors, but also relies on the ranks of neighbors. In the new rule, the nearest neighbor gets weight of 1, and the DWKNN rule is equal to the NN rule when *k*=1. Moreover, the furthest *k*-th neighbor gets a weight of 0 and the other weights are scaled linearly between 0 and 1.

C. The DWKNN algorithm

We summarize the novel dual weighted voting for the majority *k*-nearest neighbor rule in the algorithmic form as below. It should be noted that the optimal parameter value of neighborhood size *k* is empirically selected by the Leave-One-Out method for each classification task.

```
Inputs:
        $x'$ : a query object.
        $\{(x_i, c_i), i = 1,\dots, N\}$ : the training set.
Step 1: Calculate the distances.
        for $i$=1 to $N$
            $d_i = \sqrt{(x' - x_i)^T (x' - x_i)}$
        end
Step 2: Sort the distances in an ascending order.
```

```
    $[c, sorted\_dist] = sort(d_i, 'ascend')$
Step 3: Find the set of $k$ nearest neighbors of the query object $x'$, $T' = \{x_1^{NN}, \ldots, x_k^{NN}\}$.
    for $i$= 1 to $k$
        $x_i^{NN} = x_{sorted\_index(i)}, c_i^{NN} = c_{sorted\_index(i)}$
    end
Step 4: Compute the weights of the $k$ nearest neighbors.
    for $i$=1 to $k$
        case1: $d_k^{NN} \neq d_1^{NN}$
            $w_i = \frac{d_k^{NN} - d_i^{NN}}{d_k^{NN} - d_1^{NN}} \times \frac{1}{i}$
        case2: $d_k^{NN} = d_1^{NN}$
            $w_i = 1$
    end
Step 5: Compute the sum of weights for each class in the $k$ nearest neighbors set.
    for $i$=1 to $k$
        $sum\_w_c = \sum_{x_i^{NN} \in T'} w_i \times I(c = c_i^{NN})$
    end
Step 6: Assign a majority weighted voting class label $c_{\max}$ to the query object.
    $c_{\max} = \arg\max_c (sum\_w_c)$
```



(a) Influence of the neighborhood size k on accuracies.

(b) Influence of sample size on accuracies.

Figure 1. The classification results on the artificial date set

TABLE I.
SOME CHARACTERISTICS OF THE UCI DATA SETS USED: THE NUMBER OF INSTANCES, ATTRIBUTES, AND CLASSES.

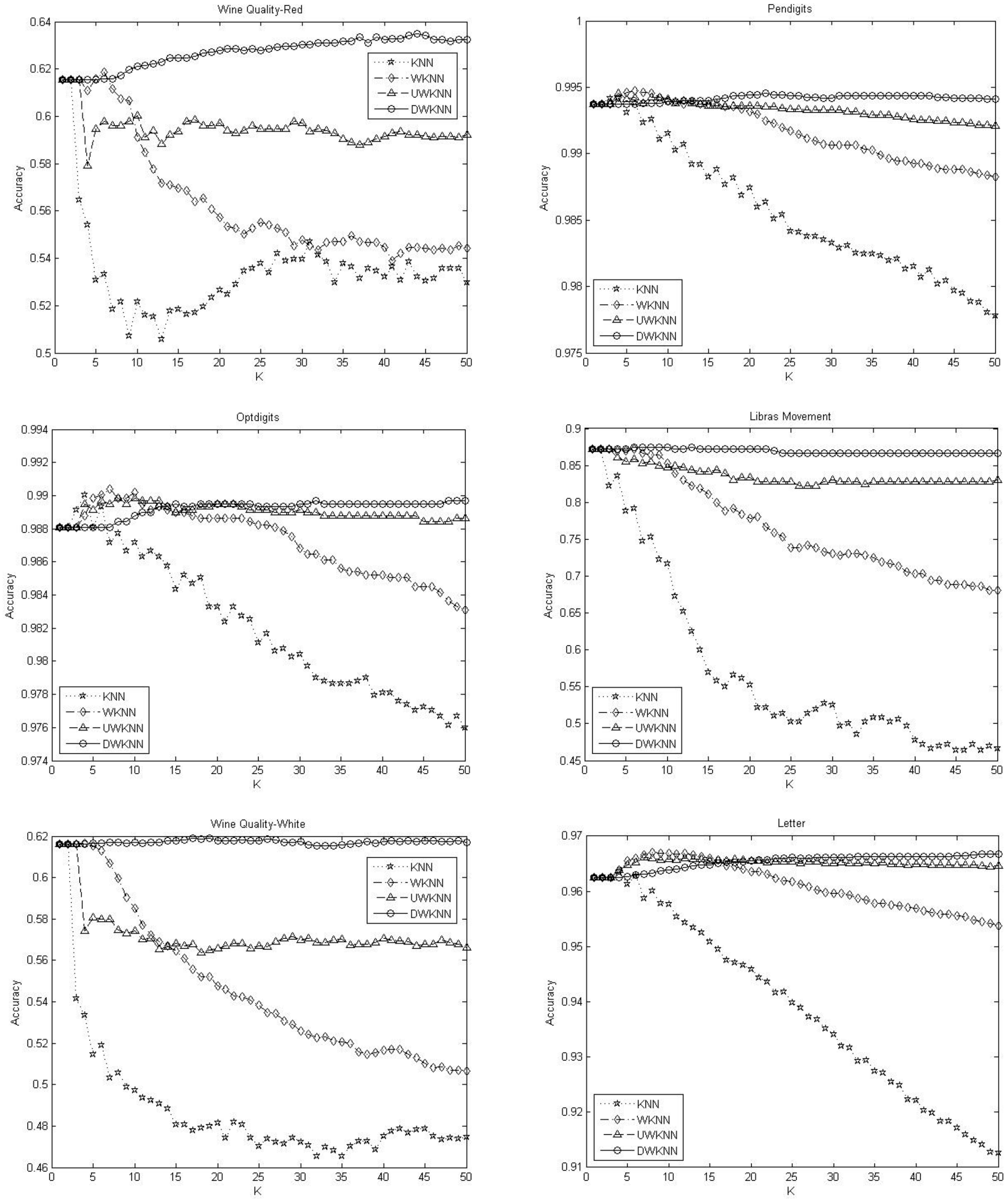
Data set	Attributes	Instances	Classes
Pendigits	16	10992	10
Optdigits	64	5620	10
Ionosphere	34	351	2
Glass	10	214	7
Landsat Satellite	36	6435	7
Libras Movement	90	360	15
Wine Quality-Red	11	1599	11
Zoo	17	101	7
Vehicle	18	946	4
Wine Quality-White	11	4898	11
Letter	16	20000	26
Image Segmentation	19	2310	7

TABLE II.
THE LOWEST ERROR (%) OF EACH METHOD WITH THE CORRESPONDING K IN THE PARENTHESIS FOR ALL DATA SETS

Data set	KNN	WKNN	UWKNN	DWKNN
Ionosphere	13.39(1)	13.39(1)	13.39(1)	**12.54**(**14**)
Landsat Satellite	8.44(4)	8.45(7)	**8.34**(**6**)	**8.34**(**46**)
Libras Movement	12.78(1)	12.78(1)	12.78(1)	**12.50**(**6**)
Glass	26.64(1)	26.64(1)	26.64(1)	**25.23(8)**
Vehicle	33.45(5)	**32.74**(**6**)	33.45(7)	33.92(30)
Wine Quality-White	38.38(1)	38.36(4)	38.38(1)	**38.10**(**17**)
Zoo	**1.98**(**1**)	**1.98**(**1**)	**1.98**(**1**)	**1.98**(**1**)
Wine Quality-Red	38.46(1)	38.15(6)	38.46(1)	**36.52**(**44**)
Pendigits	0.58(3)	**0.53**(**6**)	0.55(4)	0.55(22)
Optdigits	1.00(4)	**0.96**(**7**)	1.01(8)	1.03(32)
Image Segmentation	3.33(1)	**3.12**(**6**)	3.33(1)	3.29(9)
Letter	3.64(4)	**3.29**(**8**)	3.39(7)	3.32(50)
Average Error	15.17	15.03	15.14	**14.78**

IV. EXPERIMENTAL RESULTS

In this section, we systematically assess classification performance of the proposed classifier. The classification accuracy (or error rate) is the most effective measure of the performance of a classifier in pattern classification. In order to attain the reliable classification performance of our proposed classifier, we conduct experiments on twelve real data sets and one artificial date set respectively. These real data sets are selected from the UCI machine learning repository [16], with numeric attributes only, for simplicity of the distance metric. The Euclidean distance is used as the distance metric. To verify the estimated accuracies or error rates to be reliable in predicting the performance of the classifiers, we adopt the Leave-One-Out (LOO) method. LOO treats each


Wine Quality-Red

Pendigits

Optdigits

Libras Movement

Wine Quality-White

Letter

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

object in a data set as a query object once, and the others as training objects. LOO can maintain the statistical independence between the training set and test samples. In our experiments, we select the optimal value of the neighborhood size *k*, which minimizes the error rate estimated by LOO method.

A. Experimental data sets

We briefly describe the overall properties of the selected data sets. The artificial data set, with two classes and 12 dimensionalities, is generated by using a singular normal density function. Each class has the equal probability. In the data set, u_i is the mean vector, $u_1 = u_2 = 0$, $\sum_i$ is the covariance matrix from class c_i, $\sum_1 = 2I_{12}$, $\sum_2 = 6I_{12}$. The real data sets are depicted in Table 1. The number of instances varies from 101 (Zoo) to 20000 (Letter) while the number of attributes varies from 10 (Glass) to 90 (Libras Movement). Amongst the twelve real data sets, eleven data sets are multi-class classification tasks, one data set (Ionosphere) is two-class classification task. There are two data sets (Libras Movement and Image Segmentation) with the same number of samples of each class, and four data sets

(Vehicle, Pendigits, Letter and Optdigits) with the approximately same number of samples of each class. However, the rest data sets (Zoo, Ionosphere, Landsat Satellite, Glass, Wine Quality-Red and Wine Quality-White) have the imbalanced or irregular class distribution respectively. For instance, the smallest class of Wine Quality-White has 5 samples, and the largest class has 2198 samples.

B. Experiment 1

In order to test the classification performance of our proposed classifier, experiments are firstly conducted on the artificial data set varying with the neighborhood size *k* and the number of the training samples. Fig. 1 shows the classification results. Fig. 1 (a) shows the influences of the neighborhood size *k* on the classification accuracies with total 600 samples in artificial data set. The interval of neighborhood size *k* ranges from 1 to 50. Fig. 1 (b) shows the influences of sample size on the classification accuracies. We randomly generate training samples with number of 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900 and 2000 respectively, and set the value of *k* to be 15. The experimental results in Fig. 1 suggest that the classification performance of the proposed classifier is better than other methods with increasing neighborhood size *k* and sample size in the data set.

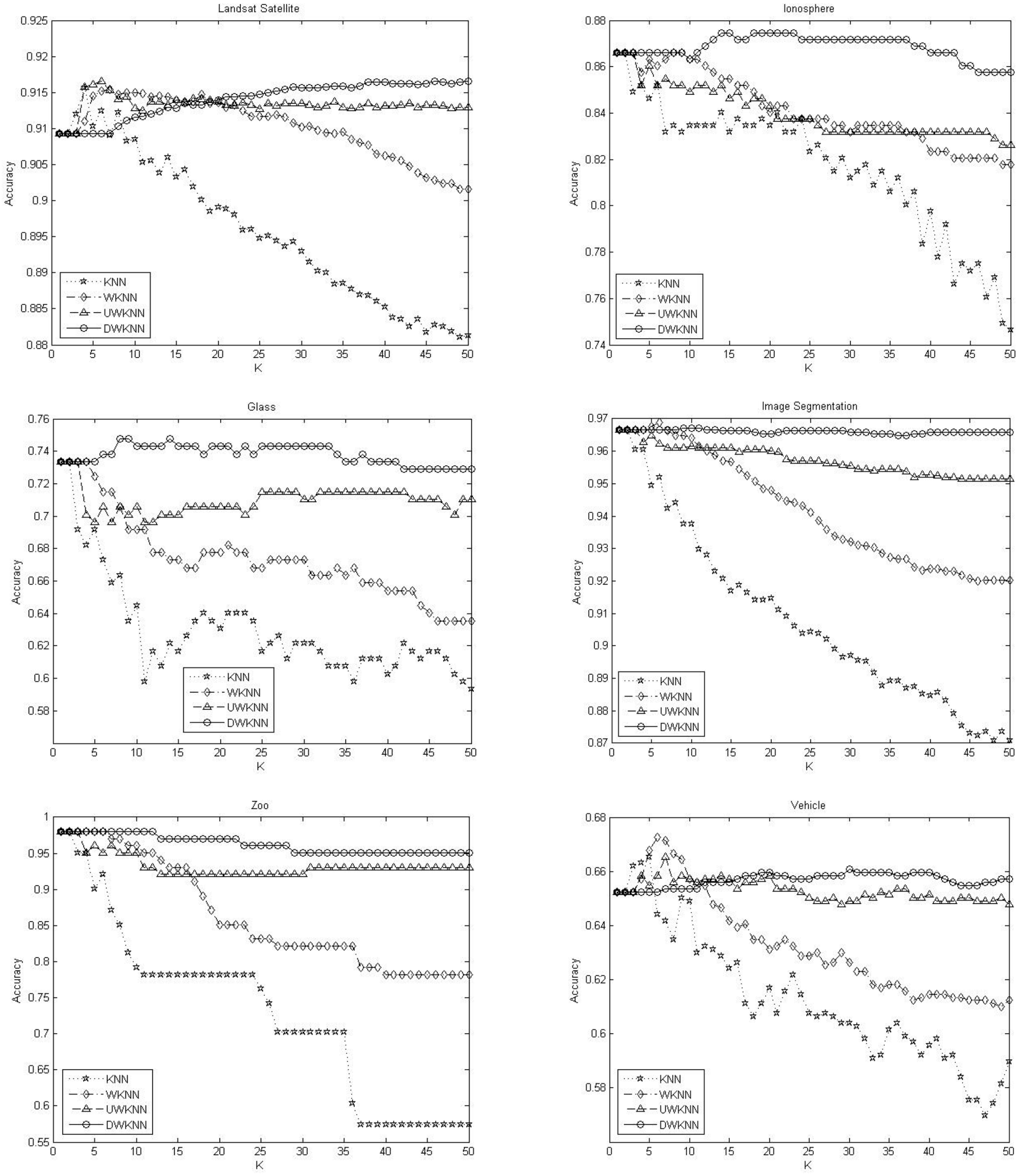


Figure 3. The classification accuracies via the neighborhood size k on each real data set.

C. Experiment 2

We further conduct the experiments to investigate the classification performance of the proposed classifier on the real data sets. The proposed classifier is compared with the KNN, WKNN and UWKNN classifiers, in terms of the classification accuracy or error rate on the query object.

The best resulting error rates (%) of each classifier on the benchmark data sets are shown in Table 2. The numbers in the parenthesis represent the corresponding optimal neighborhood size *k* of each method on each data set. It should be noted that the neighborhood size *k* ranges from 1 to 50. The error rates in bold-faces are the smallest error rates between the four classifiers on each data set, and the bold numbers in the parenthesis are the corresponding optimal value of *k*. As shown in Table 2 at the best cases, the DWKNN classifier is found to be superior or comparable to the KNN, WKNN and UWKNN classifiers, and the average error rates of DWKNN on all the real data sets is lower than the other methods. We can see that the classification performance of DWKNN is better than that of KNN on whole data sets except Zoo, Vehicle and Optdigits, WKNN in 6 out 12 data sets, and UWKNN in 7 out of 12 data sets.

Furthermore, the most important observation is that the best neighborhood size *k* of many data sets is determined as one as shown in Table 2. It has been well known that the error rate of NN rule is bounded above by twice the optimal Bayesian error [4]. On the one hand, a larger value of the neighborhood size *k* guarantees a lower error rate only if the distribution around a query object is dense enough, that is, there are a great many training samples [17]. However, in the case of the data sets with finite number of training samples, as in our experiments, a large value of *k* may not guarantee a better classification result owing to the outliers from other classes. Consequently, the NN rule results in the best performance for many data sets in Table 2. On the other hand, with a small value of *k*, the classification result can be unreliable due to data sparseness or noise objects [6, 14]. It is more interesting that, unlike the other methods, DWKNN has the large optimal values of neighborhood size *k* at almost cases. For our proposed classifier, the best classification performance is always obtained by equal or greater than 6 nearest neighbors, that is, the values of *k* is usually larger than 6, except that the same performance is attained on Zoo data set. And the optimal value of neighborhood size *k* of the DWKNN rule is larger than the KNN, WKNN and UWKNN rules besides Zoo. So we can see that the achieved optimal classification performance of DWKNN is attained by utilizing more nearest neighbors and it overcomes the negative effects of oversmoothing with increasing the value of the neighborhood size *k*. Hence, the proposed dual weighted voting scheme, which further reduces the each weight of the *k* nearest neighbors, is definitely verified on the real data sets.

In order to investigate the classification performance of the new dual weighted voting method for KNN via the neighborhood size *k* in terms of the classification accuracies, Fig. 2 and Fig. 3 show the classification results of all methods for each real date set. We can see that all data sets result in a similar pattern. As depicted in Fig. 2 and Fig. 3, it can be found that the classification performance of DWKNN is superior to which of the other methods at almost cases, especially in the case of a large value of the neighborhood size *k* on each data set. As for the proposed DWKNN classifier, in contrast to the other three methods on each real data set, the larger the value of *k* is, the better the performance is. Moreover, the classification accuracies of KNN, WKNN and UWKNN descend rapidly as *k* increases, however, DWKNN remains stable or even increased for some data sets (Optdigits, Wine Quality-Red and Letter). In addition, the proposed classifier is robust to the data sets with the imbalanced class distribution and remarkably improves the classification performance (Zoo, Ionosphere, Landsat Satellite, Wine Quality-White, Wine Quality-Red and Glass). As a consequent, DWKNN is less sensitive to the choice of *k* than the others and is robust over a wide range of *k*, and obtains the satisfactory classification performance.

Therefore, we can draw a conclusion that the proposed classifier has five benefits: (a) it can employ more nearest neighbors to keep the estimate of the majority voting smooth, in other words, it takes into account a large region of data space with a large value of neighborhood size *k* to improve the classification performance [14]. (b) it can reduce the influence of the outliers from other classes when *k* is too large, and overcome the sensitivity of noise points when *k* is too small [6]. (c) it is robust to the data sets with the imbalanced class distribution. (d) the classification performance is less sensitive to the choice of *k* and robust over a wide range of *k*. (e) the satisfactory classification performance results from a larger value of *k* than that of the other methods. So the proposed classifier is a robust and effective algorithm for the classification task in many practical situations.

V. Conclusions

In this paper, a dual weighted voting method, based on the distance-weighted voting and uniform weighted voting functions for KNN, has been proposed. The new classifier aims at overcoming the influence of the sensitivity of the selection of the neighborhood size *k*, and improving the classification performance. Compared to KNN, WKNN and UWKNN, the experiments of the proposed classifier is conducted on each data set, in terms of the classification accuracies or error rates. The experimental results suggest that this new classifier always outperforms the other classifiers, especially in the case of a large value of neighborhood size *k*. Therefore, it can be concluded that the proposed classifier is a promising algorithm owing to its robustness and effectiveness. However, it should be noted that our study has been explored only on the classification tasks with numerical data, and the dual weighted voting reduces impact of the nearest neighbors in the same class as the query object and may have a negative effect on the classification performance, these problem should be further studied in the future.

ACKNOWLEDGMENT

The authors wish to thank to the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] E. Fix, and J. L. Hodges, “Discriminatory analysis, nonparametric discrimination: Consistency properties,” *Technique Report* No. 4, U.S. Air Force School of Aviation Medicine, Randolf Field Texas, 1951

[2] T. M. Cover, and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, Vol. 13(1), pp. 21-27, 1967.

[3] T. Wagner, “Convergence of the nearest neighbor rule,” *IEEE Transactions on Information Theory*, Vol. 17(5), pp. 566-571, 1971.

[4] T. Hastie, R. Tibshirani and J. Friedman, “The elements of statical learning,” *Springer*, New York, NY, USA, 2001.

[5] S. A. Dudani, “The Distance-weighted k-Nearest Neighbor Rule,” *IEEE Transactions on System, Man, and Cybernetics*, Vol. SMC-6, pp. 325-327, 1976.

[6] X. D. Wu, V. Kumar et al., “Top 10 algorithms in data mining,” *Knowl Inf Syst*, Vol. 14, pp. 1-37, 2008.

[7] Y. Mitani, and Y. A. Hamamoto, “local mean-based nonparametric classifier,” *Pattern Recognition Letters*, Vol. 27, pp. 1151-1159, 2006 .

[8] Y. Zeng, Y. Yang and L. Zhao, “Pseudo nearest neighbor rule for pattern classification,” *Expert Systems with Applications*, Vol. 36, pp. 3587-3595, 2009.

[9] J. Wang, P. Neskovic and L. N. Cooper, “Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence,” *Pattern Recognition*, Vol. 39, pp. 417-423, 2006.

[10] P. Kang and S. Cho, “Locally linear reconstruction for instance-based learning,” *Pattern Recognition*, Vol. 41, pp. 3507-3518, 2008.

[11] H. G. Lewis, M. Brown and A. R. L. Tatnall, “Incorporating Uncertainty in Land Cover Classification from Remote Sensing Imagery,” *Advances in Space Research*, Vol. 26(7), pp. 1123-1126, 2000.

[12] R. N. Shepard, “Toward a universal law of generalization for psychological science,” *Science*, Vol. 237, pp.1317-1323, 1987.

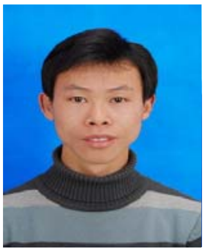
[13] J. E. S. Macleod, A. Luk and D. M. Titterington, “A re-examination of the distance-weighted k-nearest neighbor classification rule,” *IEEE Transactions on System, Man, and Cybernetics*, Vol. 17(4), pp. 689-696, 1987.

[14] J. Zavrel, “An empirical re-examination of weighted voting for K-NN,” In: Daelemans W, Flach P, van den Bosch A (eds) *Proceedings of the 7th Belgian-Dutch Conference on Machine Learning*, Tilburg, pp 139-148, 1997.

[15] K. Fukunaga, “Introduction to Statistical Pattern Recognition,” second ed. *Academic Press*, 1990.

[16] A. Frank and A. Asuncion, UCI Machine Learning Repository<*http://archive.ics.uci.edu/~mlearn/MLReposito-ry.html*>. Irvine, CA: University of California, School of Information and Computer Science, 2010.

[17] R. O. Duda, P. E. Hart and D. G. Stork, “Pattern classification,” 2ed edition, *A Wiley-Interscience Publication*, New York, NY, USA, 2001.



Jianping Gou was born in Sichuan, China. He received the B.S. degree in computer science from the North University for Nationalities of China, Yinchuan, in 2005, and the M.S. degrees in computer science from the Southwest Jiaotong University of China, Chengdu, in 2008. He is currently working toward the Ph.D. degree in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include neural networks, pattern classification, machine learning. Now he is IEEE student member.

Taisong Xiong was born in Sichuan, China. He received the B.S. degree and the M.S. degrees in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and 2006, respectively. He is currently working toward the Ph.D. degree in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include neural networks, computer vision. Now he is IEEE student member.

Yin Kuang was born in Sichuan, China. He received the M.S. degrees in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2006. He is currently working toward the Ph.D. degree in the College of Computer Science, Sichuan University, Chengdu, China. His current research interests include neural networks and pattern recognition. Now he is IEEE student member.