Research Article

A New Nearest Neighbor Classification Algorithm Based on Local Probability Centers

I-Jing Li and Jiunn-Lin Wu

Department of Computer Science and Engineering, National Chung Hsing University, 250 Kuo Kuang Road, Taichung 402, Taiwan

Correspondence should be addressed to Jiunn-Lin Wu; jlwu@cs.nchu.edu.tw

Received 14 August 2013; Revised 29 December 2013; Accepted 29 December 2013; Published 9 February 2014

Academic Editor: Massimo Scalia

Copyright © 2014 I.-J. Li and J.-L. Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The nearest neighbor is one of the most popular classifiers, and it has been successfully used in pattern recognition and machine learning. One drawback of k NN is that it performs poorly when class distributions are overlapping. Recently, local probability center (LPC) algorithm is proposed to solve this problem; its main idea is giving weight to samples according to their posterior probability. However, LPC performs poorly when the value of k is very small and the higher-dimensional datasets are used. To deal with this problem, this paper suggests that the gradient of the posterior probability function can be estimated under sufficient assumption. The theoretic property is beneficial to faithfully calculate the inner product of two vectors. To increase the performance in high-dimensional datasets, the multidimensional Parzen window and Euler-Richardson method are utilized, and a new classifier based on local probability centers is developed in this paper. Experimental results show that the proposed method yields stable performance with a wide range of k for usage, robust performance to overlapping issue, and good performance to dimensionality. The proposed theorem can be applied to mathematical problems and other applications. Furthermore, the proposed method is an attractive classifier because of its simplicity.

1. Introduction

The k nearest neighbor (k NN) [1] algorithm is a supervised classification technique that has been implemented in successfully many applications, such as pattern recognition [2] and machine learning task [3]. There are many attractive properties of k NN. First, the classification rule is intuitive; that is a query sample is classified by a majority vote of its k nearest neighbors through Euclidean distance function. Second, it has been shown that when both the number of samples and the value of k tend to be infinite, the error rate of the k NN method approaches the optimal Bayes error rate [1]. Third, k NN is a nonparametric classifier which means it has no assumptions for probability distributions. Despite of the aforementioned advantages, k NN suffers from several drawbacks. The major problem is dimensionality curse. It means that this algorithm becomes less effective in high-dimensional space. A common way to solve this problem is to select an appropriate distance measure [4–12]. Because the Euclidean distance considers all features having equal contributions, some popular methods assign

different weights to features according to their influence [5, 7, 10, 11]. Larger weights are offered to more important features and lesser weights are given to less influential ones. In general, using an appropriate distance function results in good performance in high-dimensional space [10, 12].

k NN is a lazy learning algorithm that does not obtain learning models. In the classification stage, it is necessary to compare all training data. Obviously, when the size of the dataset is large, it yields slow classification time and large storage requirement. Many approaches have been suggested based on the concept of data reduction. Many papers have addressed prototype generation [13–17] or prototype selection [18–21] methods to obtain fewer samples and faster classification time. The aim of these methods is to make a tradeoff between the reduction rate and the accuracy rate. Although these methods can provide fewer samples and decrease the classification time, how to determine the optimal prototypes is still an open problem.

Nowadays, few studies have been focused on an interesting problem: k NN fails to predict the correct class when class distributions are overlapping. In the heavily overlapped

○ Prototype of class 1
● Prototype of class 2

FIGURE 1: Classification results by an appropriate metric.

region, samples with different classes are comparable around a query pattern. As a result, the count of the number of classes becomes incorrect, and it affects the performance of k NN. Overlapping issue is general in recognition problem, but limited attention is paid to how to decrease the effect of overlapping problem in nearest-neighbors-based classifiers. Recently, researchers have proposed a weighting method called the local probability centers (LPC) algorithm [22]. The LPC algorithm is based on the categorical average pattern (CAP) [23] method. The CAP method used the categorical k nearest neighbors of a query sample for classification because it can get more local information. The categorical k nearest neighbors are from the same class around a query pattern; the categorical average patterns are then the average of these categorical k nearest neighbors. Finally, the classification rule for a query pattern is based on the distance between categorical average patterns and the query pattern. CAP performs well in the high-dimensional dataset. The shortcoming of the CAP method is that it assumes that the categorical k nearest neighbors have the same importance. Therefore, it should provide different weights to training samples. The aim of weighting method is to let the border samples have smaller influence and the interior points have larger influence. The concept of classification rule is described in Figure 1. As can be seen, the square query pattern would incorrectly be classified by traditional Euclidean distance. However, the query pattern is correctly classified by selecting an appropriate metric.

The goal of LPC algorithm is to select an appropriate metric for classification, and then it gives different weights to samples based on their posterior probabilities. Border samples have smaller posterior probabilities because they have lower confidence for classification. On the contrary, interior points are more credible and they should obtain larger posterior probabilities. Figure 2 shows an example of the LPC algorithm where $p(x | w_1)$ is the probability density function of class w_1 and $p(x | w_2)$ is the probability density function of class w_2 . A query pattern is represented by q , and Ω_1 and Ω_2 are local probability centers in two classes,

FIGURE 2: An example of LPC algorithm in the overlapping region.

respectively. If we classify q according to the Euclidean distance function, Ω_2 is its nearest neighbor and q makes an incorrect prediction. However, if we choose another measure, such as the posterior probability of the sample, then q will make a better prediction. Thus, the posterior probability is more credible and the weighting average mechanism can decrease the overlapping degree of samples.

Although the LPC algorithm is an attractive classifier, it suffers from the following drawbacks. First, the LPC algorithm gives different weights to samples based on their posterior probability. It uses the one-dimensional Parzen window [24] to estimate class-conditional density. This technique is inappropriate in the multidimensional case. Second, the LPC algorithm estimates the posterior probability of query samples using the Taylor polynomial approximation method. The Taylor theorem is described as follows.

Theorem 1 (Taylor's theorem). *Let $f(x)$ have $n+1$ continuous derivatives on $[a, b]$ for some $n \geq 0$, and let $x, x_0 \in [a, b]$; it yields*

$$f(x) = p_n(x) + R_n(x), \quad (1)$$

$$p_n(x) = f(x_0) + \frac{(x-x_0)}{1!} f'(x_0) + \cdots + \frac{(x-x_0)^n}{n!} f^{(n)}(x_0), \quad (2)$$

$$R_n(x) = \frac{1}{n!} \int_{x_0}^x (x-t)^n f^{(n+1)}(t) dt = \frac{(x-x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\xi) \quad (3)$$

for some ξ between x and x_0 , where $p_n(x)$ is called the Taylor polynomial of order n based at x_0 and $R_n(x)$ is the remainder term. To estimate the posterior probability of query samples, the LPC algorithm uses the Taylor polynomial of order 1 approximation method given by

$$p_1(x) = f(x_0) + f'(x_0)(x-x_0). \quad (4)$$

To estimate the posterior probability of the query pattern in Figure 2, the formulation (4) becomes

$$p(w_j | q) = p(w_j | \Omega_j) + \nabla p(w_j | \Omega_j)^T (q - \Omega_j), \quad (5)$$

$$j = 1, 2, \dots, N_c,$$

where N_c is the number of classes. According to (5), it is necessary to calculate the gradient of the posterior probability function. However, the LPC algorithm utilizes a parameter to represent the vector of the gradient of the posterior probability function and then multiplies it by the distance between the query sample and local probability centers. The final outcome is not equal to the inner product of two vectors. In other words, the product of the two values is not equal to the inner product of two vectors. This is the most serious problem in the LPC algorithm. Finally, the nearest neighbors of query samples in high dimensional space are further than those in lower dimensional space. Thus, the error of the Taylor formulation of order 1 becomes larger in high-dimensional space. Therefore, it should adopt another formulation to obtain better performance.

The objective of this paper is to address the aforementioned disadvantages of the LPC algorithm. Among these disadvantages, the first and the third problems are easier to solve. To deal with the first problem, the multidimensional Parzen window [24] can be used to estimate class-conditional density. This is a suitable choice in a general case. With respect to the third problem, the Taylor polynomial of order 1 can be replaced by the Euler-Richardson formulation [25]. Euler-Richardson method uses nearer points to estimate the posterior probability; therefore, the Taylor polynomial of order 1 approximation method is accurate to order $(\Delta x)^2$ where $\Delta x = |x - x_0|$, whereas the Euler-Richardson method is accurate to order $(\Delta x)^3$. It would be a better numerical approximation method compared to Taylor polynomial approximation method.

Considering the second problem, the most difficult work has to be solved. The study presents the gradient of the posterior probability function through mathematical derivation. This paper proves that the gradient of the posterior probability function can be estimated under sufficient assumptions. This is a novel step of the proposed method. This theoretic property makes it truly calculate the inner product of two vectors. Therefore, we have developed an improved version of LPC algorithm called ILPC.

In the study, both synthetic datasets and real datasets are used to evaluate performance. LPC method adopts an incorrect formulation which makes it perform poorly when the value of k is very small and in higher-dimensional datasets. However, the proposed method achieves robust performance with a wide range of k for usage. This indicates that correct formation provides a good model. The advantages of the proposed method are summarized in the following.

- (1) The proposed method is the best performing method for overlapping issue. Few classifiers present good performance in the situation.
- (2) Multidimensional Parzen window and Euler-Richardson method increase the performance of the

proposed method. It is helpful to the proposed new classifier to implement in real applications.

- (3) The proposed method has robust performance with wide range of k for usage. It is easy to select an appropriate value of k in the proposed method.
- (4) The proposed method is based on k nearest neighbor classifier. It is simple that it only adds weights to training samples.
- (5) In the paper, the gradient of posterior probability function can be estimated under sufficient assumptions. This property can be applied to other mathematical problems or applications.

This paper is organized as follows. Section 2 introduces the related works of classifiers including k NN, CAP, and LPC algorithms. The proposed method and mathematical derivation are described in Section 3. Experimental results and discussions are displayed in Section 4. Finally, conclusions are drawn in Section 5.

2. Related Work

This section reviews related works for the proposed method. After a review of some improved k NN classifiers, the section describes the details of LPC method.

The k nearest neighbor is one of the most popular classification methods and has been widely used in pattern recognition [2] and other applications [3]. If q is a query pattern and y is a training sample, then the common metric for k nearest neighbor is defined by the formulation

$$D(q, y) = \left(\sum_{i=1}^d (q_i - y_i)^2 \right)^{1/2}, \quad (6)$$

where d denotes the number of features. In (6), the query pattern is classified in terms of the Euclidean distance function. The Euclidean distance function is simple and easily implemented. The weakness of the Euclidean distance function is that it assumes that all features have the same importance. In general, there is a high degree correlation among features and noisy features in datasets. Thus, numerous studies have addressed how to select an appropriate distance function, such as the Chi-square distance [4], weighted distance [5, 7, 10, 11], optimal distance [9], and adaptive distance [6, 8]. Some of these functions perform effectively in general cases [4, 8].

Another issue for k NN is that, when class distributions are overlapping, the class label becomes unreliable. This makes k NN method predict a wrong class. Researchers recently proposed the local probability centers algorithm to improve this situation [22]. The LPC algorithm is based on the categorical average pattern (CAP) [23], which uses the categorical k nearest neighbors for a query pattern, while k NN uses global nearest neighbor which are from different classes. However, the categorical k nearest neighbors are from the same class around a query pattern. The CAP algorithm is illustrated as follows. Let $x_i = [x_{i1}^{w_j}, \dots, x_{id}^{w_j}]^T$ be a d -dimensional vector belonging to class w_j , where $j = 1, \dots, N_c$, $i = 1, \dots, N_j$,

N_c is the number of classes and N_j is the number of samples belonging to w_j , and $N = \sum_{j=1}^{N_c} N_j$ is the total number of training samples. Given a query pattern $q = [q_1, \dots, q_d]^T$, the categorical k nearest neighbors in the class w_j of q are denoted by $\chi_k^{w_j}(q)$. The class label w of the test sample is determined by

$$w = \arg \min_j \left\{ \left\| \frac{1}{k} \sum_{x_i^{w_j} \in \chi_k^{w_j}(q)} x_i^{w_j} - q \right\| \right\}. \quad (7)$$

The benefits of the CAP method are that it can reduce the effect of outliers when the value of k is very small and it performs well in high-dimensional space. Hotta et al. proposed a kernel version of the CAP method called KCAP [23]. The classification rule is given by

$$w = \arg \min_j \left\{ \left\| \frac{1}{k} \sum_{x_i^{w_j} \in \chi_k^{w_j}(q)} \Phi(x_i^{w_j}) - \Phi(q) \right\| \right\}, \quad (8)$$

$j = 1, \dots, N_c,$

where $\Phi(\cdot)$ is a mapping function that maps samples from a data space to another data space. Choosing an appropriate kernel function can improve CAP performance. For example, the Gaussian kernel makes CAP obtain better accuracy rate.

One disadvantage of the CAP method is that it assumes that the categorical k nearest neighbors around the query pattern have equal contributions. Zeng et al. [26] proposed a pseudonearest neighbor rule that gives different weights to the categorical k nearest neighbors in terms of the reciprocal value of k . The weight formulation Y_i is defined as

$$Y_i = \frac{1}{i} \quad i = 1, \dots, k. \quad (9)$$

On the basis of the formulation, the nearer categorical k samples around query samples have greater influence, and the farther categorical k samples around query samples have lesser influence. The weighting formulation is incorrect because there is no difference among samples in different classes. Hence, this method achieves poor performance, even worse than the CAP method in some cases.

As mentioned previously, various improved k NN classifiers, including the CAP and KCAP algorithms, are all distance-based methods. Therefore, these methods have problem described in Figures 1 and 2. The LPC algorithm, which is based on statistical principles, was proposed to deal with this problem. The key idea of the LPC is using posterior probability for classification. The LPC algorithm uses the one-dimensional Parzen window to estimate the probability density function

$$\hat{p}(x) = \frac{1}{N\tau^n(2\pi)^{n/2}} \sum_{i=1}^N e^{-(1/2\tau^2)(x_i-x)^T(x_i-x)}, \quad (10)$$

where $n = 1$ and τ is a parameter that controls the width of Gaussian kernel. The posterior probability $p(w_j | x_i^{w_j})$ of

samples is calculated through Bayes Theorem. Finally, LPC method uses the Taylor polynomial of order 1 to estimate the posterior probability of the query sample

$$p(w_j | q) = p(w_j | \Omega_j) - \nabla p_j \|\Omega_j - q\|, \quad j = 1, \dots, N_c, \quad (11)$$

where Ω_j represents the local probability centers of classes. Comparing (5) and (11), (11) is incorrect form because the term $\nabla p_j \|\Omega_j - q\|$ is not equal to the inner product of two vectors. Referring to the Cauchy-Schwarz inequality, for any two vectors a and b in d -dimensional space,

$$\left(\sum_{m=1}^d a_m b_m \right)^2 \leq \left\| \sum_{m=1}^d a_m^2 \right\| \left\| \sum_{m=1}^d b_m^2 \right\|. \quad (12)$$

The equality is tenable if and only if a and b are linearly dependent, but it is improbable in this case. However, the LPC method adopts the term $\nabla p_j = \alpha d / R_j$ to calculate the gradient vector under incorrect assumptions, where R_j is the maximal radius of $\chi_k^{w_j}(q)$. There is an incorrect relationship among the gradient vector and the dimension d and the maximal radius R_j . The incorrect formula may produce unfavorable results. For example, LPC presents unsatisfied performance in the high-dimensional datasets. The best way is to estimate the gradient of the posterior probability function. It is logical choice to faithfully calculate the outcome of the inner product of two vectors. Then the correct formulation improves performance of LPC algorithm.

3. The Proposed Method

This section is devoted to illustrating the method proposed in this study. The proposed algorithm consists of three parts. The first part includes the estimation of the posterior probability of all samples by multidimensional Parzen window. The analysis of the Euler-Richardson method is described in the second part. The third part presents classification rule based on the posterior probability of a query pattern. The theoretic property is also proofed in the subsection.

3.1. Preprocessing. The first part of the proposed method utilizes the multidimensional Parzen window to estimate the probability density of each point. The formulation is given as follows:

$$p(x_i | w_j) = \frac{1}{k_0} \sum_{x_s^{w_j} \in \chi_{k_0}^{w_j}(x_i)} \phi(x_s^{w_j}), \quad (13)$$

$i = 1, \dots, N, \quad j = 1, \dots, N_c,$

where $p(\cdot)$ stands for the probability density function, and $x_s^{w_j} \in \chi_{k_0}^{w_j}(x_i)$ is the k_0 nearest neighbor of $x_i^{w_j}$. This study chooses the most general Gaussian kernel, which is defined as

$$\phi(x_i^{w_j}) = \frac{1}{h^d(2\pi)^{d/2}} e^{-(1/2h^2)(x_i - x_s^{w_j})^T(x_i - x_s^{w_j})}, \quad (14)$$

FIGURE 3: The proposed method based on the Euler-Richardson formulation.

where h controls the kernel width size and d represents the size of features. Calculating the probability density of other classes in a similar way leads to

$$p(x_i | w_t) = \frac{1}{k_0} \sum_{x_s^{w_t} \in \mathcal{S}_{k_0}^{w_t}} \phi(x_s^{w_t}), \quad t \neq j \cap t = 1, \dots, N_c. \quad (15)$$

Next, the class posterior probability of point x_i is computed in terms of the Bayes Theorem

$$p(w_j | x_i) = \frac{p(x_i | w_j) p(w_j)}{\sum_{t=1}^{N_c} p(x_i | w_t) p(w_t)}, \quad \text{if } x_i \in w_j, \quad (16)$$

where $p(w_t) = (N_t/N)$ denotes a prior probability of the t th class. The preprocessing step is to offer different weights to all samples based on their credibility. The value implies data distributions.

3.2. Analysis of the Euler-Richardson Method. The proposed method replaces the Taylor polynomial of order 1 with the Euler-Richardson method. Because the Euler-Richardson method is not a well-known algorithm, it shows the details of Euler-Richardson formulation in the subsection.

The benefit of the Euler-Richardson method is that it is accurate to $(\Delta x)^3$, which is the same accuracy given by the Taylor formulation of order 2, but it does not need to calculate the second differential of $f(x)$. Figure 3 illustrates the concept of the proposed method based on the Euler-Richardson method. The term $\nabla p(w_1 | (\Omega_1 + q)/2)$ is more precise to estimate $p(w_1 | q)$ because the midpoint $(\Omega_1 + q)/2$ is nearer to the query pattern than the local center Ω_1 . Similarly, the midpoint $(\Omega_2 + q)/2$ is nearer to the query pattern than Ω_2 and $\nabla p(w_2 | (\Omega_2 + q)/2)$ is beneficial for estimating $p(w_2 | q)$. Therefore, the Euler-Richardson method yields more precise results in high-dimensional space. The local centers Ω_1 and Ω_2 are further apart in the high-dimensional dataset, and error increases with the application of the traditional Taylor polynomial of order 1. Hence, the proposed method achieves reasonable improvement when the dimension is high.

In the following, we show mathematical derivation of Euler-Richardson method. This study defines the function $y = f(x)$. The estimate of y_1 through the Taylor polynomial of order 2 is given by

$$y_1 = f(x + \Delta x) = f(x) + f'(x) \Delta x + \frac{1}{2} f''(x) (\Delta x)^2, \quad (17)$$

where $\Delta x = |x_1 - x|$. Divide the step Δx into two half steps. The first half step is defined as

$$f\left(x + \frac{1}{2} \Delta x\right) = f(x) + f'(x) \frac{1}{2} \Delta x + \frac{1}{2} f''(x) \left(\frac{1}{2} \Delta x\right)^2. \quad (18)$$

Then, the second half step can be written as

$$y_2 = f(x + \Delta x) = f\left(x + \frac{1}{2} \Delta x\right) + f'\left(x + \frac{1}{2} \Delta x\right) \frac{1}{2} \Delta x + \frac{1}{2} f''\left(x + \frac{1}{2} \Delta x\right) \left(\frac{1}{2} \Delta x\right)^2. \quad (19)$$

Substituting (19) into (18) leads to

$$y_2 = f(x + \Delta x) = f(x) + \frac{1}{2} \left[f'(x) + f'\left(x + \frac{1}{2} \Delta x\right) \right] \Delta x + \frac{1}{2} \left[f''(x) + f''\left(x + \frac{1}{2} \Delta x\right) \right] \left(\frac{1}{2} \Delta x\right)^2. \quad (20)$$

Recall that $f''(x + (1/2)\Delta x) = f''(x) + (1/2)f'''(x)\Delta x + \dots$. This study considers order $(\Delta x)^2$, and the formulation becomes

$$y_2 = f(x + \Delta x) = f(x) + \frac{1}{2} \left[f'(x) + f'\left(x + \frac{1}{2} \Delta x\right) \right] \Delta x + \frac{1}{2} \left[2f''(x) \right] \left(\frac{1}{2} \Delta x\right)^2. \quad (21)$$

Combining (17) and (21) cancels the terms of order $(\Delta x)^2$. Finally, the Euler-Richardson method is defined as follows:

$$y_1 = 2y_2 - y_1 = f(x) + f'\left(x + \frac{1}{2} \Delta x\right) \Delta x + O(\Delta x)^3. \quad (22)$$

Obviously, the formulation (22) is accurate to $(\Delta x)^3$, which is the same as the accuracy for the Taylor polynomial of order 2, but it does not need to calculate the second differential of $f(x)$. For this reason, Euler-Richardson method is adopted to increase the accuracy.

3.3. Classification Rule with Local Probability Centers. The final part of the proposed method is to estimate the posterior probability of the query pattern in the classification step. For a query pattern q , finding its k nearest neighbors in the class

w_i is denoted by $\chi_k^{w_j}(q)$. The local probability centers Ω_j are computed according to the formulation

$$\Omega_j = \frac{\sum_{x_i \in \chi_k^{w_j}(q)} p(w_j | x_i^{w_t}) \cdot x_i^{w_t}}{\sum_{x_i \in \chi_k^{w_j}(q)} p(w_j | x_i^{w_t})}. \quad (23)$$

As mentioned earlier, the classification rule is carried out using the distance between the local probability and the query pattern would lead to an incorrect prediction. Thus, the posterior probability of the local probability center can be simply calculated by

$$p(w_j | \Omega_j) = \frac{\sum_{x_i \in \chi_k^{w_j}(q)} p(w_j | x_i^{w_j})}{k}. \quad (24)$$

Finally, the posterior probability of the j th class of the query pattern which combined the Euler-Richardson method is defined as

$$p(w_j | q) = p(w_j | \Omega_j) - \nabla p\left(w_j | \frac{\Omega_j + q}{2}\right)^T (\Omega_j - q), \quad j = 1, \dots, N_c. \quad (25)$$

It is noticed that this formulation is different from (11). The midpoint $(\Omega_j + q)/2$ provides good estimate and it minimizes the error according to the statement. Now, it is a challenging work of calculating the term $\nabla p(w_j | (\Omega_j + q)/2)^T$. The theoretic property is given in the following.

Theorem 2. Let $p(w_j | x)$ be the posterior probability of x of the j th class, and $p(x | w_j)$ is the class-conditional density of x then

$$\frac{d}{dx} p(w_j | x) = \frac{p(w_j | x)}{p(x | w_j)} \frac{d}{dx} p(x | w_j). \quad (26)$$

Proof. According to Bayes Theorem, for every observing point x conforms to

$$p(w_j | x) = \frac{p(x | w_j) p(w_j)}{p(x)}. \quad (27)$$

Involve logarithm function $\ln(\cdot)$ and the differential, we have

$$\frac{d}{dx} (\ln p(w_j | x)) = \frac{d}{dx} \left(\ln \frac{p(x | w_j) p(w_j)}{p(x)} \right). \quad (28)$$

□

Expanding, we obtain

$$\begin{aligned} \frac{d}{dx} (\ln p(w_j | x)) \\ = \frac{d}{dx} (\ln p(x | w_j) + \ln p(w_j) - \ln p(x)). \end{aligned} \quad (29)$$

Since $p(w_j)$ is a constant, then (29) becomes

$$\frac{d}{dx} (\ln p(w_j | x)) = \frac{d}{dx} (\ln p(x | w_j) - \ln p(x)). \quad (30)$$

The nonparametric density estimation [1, 27] can be written as

$$p(x) \approx \frac{k_N}{NV_N}, \quad (31)$$

where V_N is the volume of x and k_N is the number of samples inside the volume. We assume that the estimation density converges to real $p(x)$ that satisfies the following three conditions:

$$\begin{aligned} \lim_{N \rightarrow \infty} V_N &\rightarrow 0, \\ \lim_{N \rightarrow \infty} k_N &\rightarrow \infty, \\ \lim_{N \rightarrow \infty} \frac{k_N}{N} &\rightarrow 0. \end{aligned} \quad (32)$$

Here, we choose $V = 1/\sqrt{N}$; it yields $p(x) \approx k_N/\sqrt{N}$, and (30) becomes

$$\begin{aligned} \frac{d}{dx} (\ln p(w_j | x)) &= \frac{d}{dx} \left(\ln p(x | w_j) - \ln \frac{k_N}{\sqrt{N}} \right) \\ &= \frac{d}{dx} (\ln p(x | w_j) - 1) \\ &= \frac{d}{dx} \ln p(x | w_j). \end{aligned} \quad (33)$$

It shows that $p(x)$ can be estimated under sufficient assumption.

Calculating the differential of logarithm function of two sides, we obtain

$$\frac{1}{p(w_j | x)} \frac{d}{dx} p(w_j | x) = \frac{1}{p(x | w_j)} \frac{d}{dx} p(x | w_j). \quad (34)$$

Finally, the differential of posterior probability of x is defined as

$$\frac{d}{dx} p(w_j | x) = \frac{p(w_j | x)}{p(x | w_j)} \frac{d}{dx} p(x | w_j). \quad (35)$$

Now, the multidimensional Parzen window with Gaussian kernel is described as follows:

$$p(x | w_j) = \frac{1}{Nh^d(2\pi)^{d/2}} \sum_{i=1}^N e^{-(1/2h^2)(x_i-x)^T(x_i-x)}, \quad (36)$$

if $x_i \in w_j$.

Substituting (36) into (35), it yields

$$\begin{aligned} \frac{d}{dx} \ln p(w_j | x) &= \frac{p(w_j | x)}{p(x | w_j)} \frac{1}{Nh^{d+2}(2\pi)^{d/2}} \\ &\quad \times \sum_{i=1}^N (x - x_i) e^{-(1/2h^2)(x_i-x)^T(x_i-x)}. \end{aligned} \quad (37)$$

We only consider local probability centers near the query pattern; therefore, $N = 1$ for each class. Then (37) can be written as

$$\begin{aligned} \frac{d}{dx} p\left(w_j \mid \frac{\Omega_j + q}{2}\right) \\ = \frac{p(w_j \mid (\Omega_j + q)/2)}{p((\Omega_j + q)/2 \mid w_j)} \frac{1}{Nh^{d+2}(2\pi)^{d/2}} \\ \times \left(\frac{\Omega_j - q}{2}\right) e^{-(1/2h^2)(q - (\Omega_j + q)/2)^T (q - (\Omega_j + q)/2)}. \end{aligned} \quad (38)$$

We mentioned that the midpoint $(\Omega_j + q)/2$ is nearer to the query pattern than the local center Ω_j . However, the differential of $p(w_j \mid (\Omega_j + q)/2)$ can be calculated by (38), and its error is small if Ω_j is very near to q . Since the term $p(w_j \mid (\Omega_j + q)/2)/p((\Omega_j + q)/2 \mid w_j)$ has to be calculated in the classification phase, it consumes the classification time. In order to speed up the computation, the differential of $p(w_j \mid (\Omega_j + q)/2)$ can be approximated to the following formula:

$$\begin{aligned} \frac{d}{dx} p\left(w_j \mid \frac{\Omega_j + q}{2}\right) \approx \frac{p(w_j \mid \Omega_j)}{p(\Omega_j \mid w_j)} \frac{1}{h^{d+2}(2\pi)^{d/2}} \\ \times \left(\frac{\Omega_j - q}{2}\right) e^{-(1/8h^2)(q - \Omega_j)^T (q - \Omega_j)}. \end{aligned} \quad (39)$$

By substituting (39) into (25), we obtain

$$\begin{aligned} p(w_j \mid x) = p(w_j \mid \Omega_j) - \frac{p(w_j \mid \Omega_j)}{p(\Omega_j \mid w_j)} \\ \times \frac{1}{h^{d+2}(2\pi)^{d/2}} e^{-(1/8h^2)(q - \Omega_j)^T (q - \Omega_j)} \\ \times \frac{(q - \Omega_j)^T (q - \Omega_j)}{2}. \end{aligned} \quad (40)$$

Compared with (11), the formulation (40) shows a numerical model in terms of the proposed method and Euler-Richardson method. However, the LPC method uses $\nabla p_j = \alpha d/R_j$ to represent the gradient vector; this is an unreasonable assumption. According to the assumptions, there is an incorrect relationship among the gradient vector, the dimension d , and the maximal radius R_j . In other words, there is no linear relationships between the gradient vector and the dimension d . Moreover, there is not a reciprocal relationship between the gradient vector and the maximal radius R_j . Regarding the proposed method, a novel step is proofed and it can make the inner product of two vectors be truly calculated. It seems that the modified formulation has a correct model. It can be found that the modified formulation does not have the term R_j and there is a more accurate relationship to the dimension d . The modified formulation is based on statistical principles and theorems. Using the new classification rule should reasonably achieve good performance.

TABLE 1: The parameters of the proposed method used in the experiments.

Dataset	LPC	ILPC
$I - \Lambda$	$\tau = 0.1, \alpha = 5$	$h = 3$
$I - 4I$	$\tau = 0.1, \alpha = 10$	$h = 10$
$I - I$	$\tau = 0.1, \alpha = 0.01$	$h = 2$
10D	$\tau = 0.1, \alpha = 0.1$	$h = 5$
20D	$\tau = 0.1, \alpha = 0.5$	$h = 8$
30D	$\tau = 0.1, \alpha = 1$	$h = 10$
Iris	$\tau = 0.1, \alpha = 2.5$	$h = 0.5$
Wine	$\tau = 0.1, \alpha = 2$	$h = 5.5$
Iono	$\tau = 0.1, \alpha = 0.1$	$h = 3$
Sonar	$\tau = 0.1, \alpha = 2.5$	$h = 5.5$

4. Experimental Results

This section presents experimental results of this study. Both simulated datasets and real datasets are used to verify the proposed method. Since the LPC method has focused on the overlapping problem, Section 4.1 describes the performance of artificial datasets with different overlapping degrees. To show the effect of dimension, the performance of artificial datasets with different dimensions is displayed in Section 4.2. Then Section 4.3 shows the performance of the real dataset from the UCI machine learning repository. Finally, Section 4.4 provides discussions of the results.

In the study, four methods are used to compare performance: k NN, CAP, LPC, and ILPC. All of them are k NN-based methods. Both LPC and ILPC methods have parameters, and the fixed parameter $k_0 = \max\{10, 0.01N\}$ is set in all experiments. Table 1 shows other parameters of LPC and ILPC methods in experimental results. There are two tuned parameters in LPC algorithm: the width of Gaussian kernel τ and the value α . The ILPC has only one tuned parameter: the width of Gaussian kernel h .

4.1. Simulated Datasets with Different Overlapping Degree.

The first experiment uses artificial datasets with different degrees of overlapping to evaluate the performance of four methods because the LPC method states that it can improve the overlapping problem. These three artificial datasets are described as follows.

$I - \Lambda$ dataset:

$$\begin{aligned} u_1 &= \mathbf{0}_8, \\ u_2 &= [3.86, 3.10, 0.84, 0.84, 1.64, 1.08, 2.06, 0.01]^T, \\ \Sigma_1 &= I_8, \\ \Sigma_2 &= \text{diag}[8.41, 12.06, 0.12, 0.22, 1.49, 1.77, 0.35, 2.73], \end{aligned} \quad (41)$$

$I - 4I$ dataset:

$$u_1 = \mathbf{0}_8, \quad u_2 = \mathbf{0}_8, \quad \Sigma_1 = I_8, \quad \Sigma_2 = 4I_8, \quad (42)$$

FIGURE 4: The accuracy rate with different k values on the $I - \Lambda$ dataset.

$I - I$ dataset:

$$u_1 = \mathbf{0}_8, \quad u_2 = [2.56, 0, \dots, 0]^T, \quad \Sigma_1 = \Sigma_2 = I_8, \quad (43)$$

where Σ_i and u_i represent the covariance matrix and the mean vector of class i . I_k and $\text{diag}[\cdot]$ are denoted by $k \times k$ identity and diagonal matrix, respectively. The $I - \Lambda$ dataset consists of 8-dimensional Gaussian data with two classes, which are different means and different variances in all dimensions. Next, $I - 4I$ dataset consists of 8-dimensional Gaussian data with two classes. The means in all dimensions are the same, but the variance of the one class is four times to that of the other class. In the final, $I - I$ dataset consists of 8-dimensional Gaussian data with the same variance, and the mean vectors are slightly different in the first dimension. In summary, these three artificial datasets have different degrees of overlapping. The $I - I$ dataset is the most heavily overlapped dataset, followed by the $I - 4I$ dataset and the $I - \Lambda$ dataset. Each artificial dataset includes 2000 training samples, with 1000 samples in each class. In the classification stage, this study generates another 2000 test samples, with 1000 samples in each class.

Figure 4 illustrates the performance of all four methods in the $I - \Lambda$ dataset with different k values. Because the distributions are separable, all methods achieve high accuracy rate. In this case, it shows that the proposed method is the best performing method, with stable performance and nearly 100% accuracy rate. CAP method obtains better accuracy than kNN because it reduces the effect of noisy patterns. Notice that LPC performance is sensitive to the value of k . In particular, it obtains lower accuracy rate when $k < 4$. The phenomena implies that the assumption $\nabla p_j = \alpha d/R_j$ is inappropriate, where the term R_j is the maximal distance of $\chi_k^{w_j}(q)$. There is not a reciprocal relationship between the gradient vector and the maximal distance R_j . The term affects the estimate of the posterior probability when k is smaller. When distance between the local probability center and a query pattern is smaller, the pseudogradient value will be larger. Then, letting the pseudo gradient value multiply the distance between the query sample and local center, the reminder term of Taylor polynomial approximation method will be very large. In the situation, the posterior probability of

FIGURE 5: The accuracy rate with different k values on the $I - 4I$ dataset.

query sample would be incorrect. Hence, LPC presents lower accuracy rate when k is smaller.

Figure 5 illustrates the performance of the four methods in the $I - 4I$ dataset with different k values. The behavior of the LPC method in this case is similar to that in the $I - \Lambda$ dataset. The formulation of LPC is incorrect and it results in poor performance when k is very small. Considering kNN and CAP methods, they obtain similar performance. They have the same behavior that when the value k becomes larger, the accuracy rate is lower. In contrast, ILPC shows better performance when k becomes larger. This is because the density function is different in the two classes. When k is smaller, the one class has great advantage because its distribution is more centralized. In the situation, Parzen window method is difficult to precisely estimate the density function. When k becomes larger, Parzen window can obtain the reliable information from all of the classes. Therefore, the proposed method yields better performance. There is a medium overlap between two classes in the $I - 4I$ dataset. The class voting-based methods easily obtain unreliable information when the distributions are overlapping. The weighting average mechanism can generally reduce the degree of overlap in the local region. Hence, the ILPC method achieves robust performance in the $I - 4I$ dataset.

Finally, Figure 6 shows the accuracy rate with different k values on the $I - I$ dataset. The $I - I$ dataset is the most heavily overlapped dataset among the three synthetic datasets. kNN and CAP show poor performance because class label becomes unreliable in the overlapping region. As mentioned earlier, the basic idea behind the LPC method is to let the border samples have lesser influence. Thus, it yields better performance than CAP and kNN . However, the formulation which estimates the posterior probability is deficient. However, it can be observed that the proposed method provides excellent performance in the case. It implies that the proposed theorem provides a novel way to compute the inner product. As a result, in the heavily overlapped region, ILPC still can make correct prediction. As can be seen, among the four methods, only the proposed method achieves above 90% accuracy rate. Compared to kNN method, the

TABLE 2: The performance comparisons on the synthetic datasets with different overlapping degree.

	kNN		CAP		LPC		ILPC	
Dataset	Acc.	σ	Acc.	σ	Acc.	σ	Acc.	σ
$I - \Lambda$	98.51	0.160	98.74	0.246	98.35	0.750	99.80	0.000
$I - 4I$	94.81	1.597	95.58	2.053	94.97	4.319	96.15	0.504
$I - I$	87.31	2.032	86.54	1.285	88.25	1.356	93.49	0.443

The bold font means the result of the proposed method.

proposed method increases around 20% accuracy rate when $k = 1$.

The performance comparisons of the synthetic data with different degrees of overlapping in terms of average accuracy rate and variance are listed in Table 2. Obviously, as the degree of overlapping increases, all methods have lower performance. The class voting-based method cannot obtain reliable information in the overlapping region; thus, k NN and CAP yields lower accuracy rate in most heavily overlapping dataset. Although LPC uses the powerful metric which is based on posterior probability, the incorrect formulation affects the performance and stability. Because ILPC uses correct formulation for classification, the performance is above 90% accuracy rate on three synthetic datasets. In addition, the proposed method has stable and robust performance regardless of the value of k .

4.2. Simulated Datasets with Different Dimensions. Although the literature has shown that LPC reports performance comparisons from $d = 2$ to 8, it does not give the results in higher-dimensional space. In order to show the effect of dimension, this subsection uses artificial dataset with different dimensions to measure the performance. The artificial datasets with different dimensions are described as follows.

Dataset 1(10D), 2(20D) and 3(30D)

$$u_1 = \mathbf{0}_p, \quad u_2 = [0.5]_p^T, \quad \Sigma_1 = \Sigma_2 = I_p, \quad (44)$$

where $p \in \{10, 20, 30\}$ denotes the size of features. Similar to the first experiment, this study generates 4000 samples; half of the samples are for learning and the other half of the samples are for classification. There are different degrees of overlap between two classes in the artificial datasets. Figure 7 shows the accuracy rate as a function of k on 10D dataset. From results, it seems that there is a high overlapping in 10D dataset because all methods obtain lower accuracy rate. The proposed method yields the best performance, followed by LPC, k NN, and CAP. The obtained results in 10D dataset are similar to those in the $I - I$ dataset. It is noticed that k NN is superior to CAP in 10D dataset. k NN classifies a query sample by a majority vote of its k nearest neighbors. However, the classification rule of CAP is in terms of the distance between the query pattern and the local centers. It could be inferred that class voting-based method can obtain more reliable information than distance-based method. In the heavily overlapped region, it is difficult to determine the class

FIGURE 6: The accuracy rate with different k values on the $I - I$ dataset. FIGURE 7: The accuracy rate with different k values on the 10D dataset.

label of query samples based on Euclidean distance. Thus, CAP yields the worst performance among four methods.

The performance of four methods on 20D dataset is shown in Figure 8. All methods achieved better performance here than on 10D dataset because there are more features for usage. Compared to 10D dataset, there is a medium overlap in 20D dataset. Thus, CAP and k NN achieve similar performance in this case. LPC is the worst method in the case. The evidence points out that the assumption $\nabla p_j = \alpha d/R_j$ has a negative effect on the performance. In some sense, there is not a linear relationship between the dimension d and the gradient vector. When the dimension is larger, the outcome of the term will be very large. Then the reminder term would be very large, too. The outcome affects the posterior probability of a query sample. According to the proposed formulation, multidimensional Parzen window is used to estimate the gradient vector. It is a logical choice in high dimensional space. Furthermore, the proposed method uses Euler-Richardson method to reinforce the accuracy in the high-dimensional space. It can be observed that the proposed method outperforms other methods above 10% accuracy rate when k is small. The results may imply that the proposed method can obtain good performance in high dimensional space.

FIGURE 8: The accuracy rate with different k values on the 20D dataset.

FIGURE 9: The accuracy rate with different k values on the 30D dataset.

TABLE 3: The performance comparisons on the synthetic datasets with different dimensions.

Dataset	kNN		CAP		LPC		ILPC	
	Acc.	σ	Acc.	σ	Acc.	σ	Acc.	σ
10D	71.29	3.065	68.95	1.877	73.86	2.349	78.41	0.597
20D	78.63	3.104	78.85	2.730	76.25	4.023	88.73	0.706
30D	85.18	3.554	84.03	2.793	78.71	4.490	94.13	0.510

The bold font means the result of the proposed method.

Finally, we consider the highest dimensional dataset, 30D dataset. Figure 9 shows the performance of four methods as a function of k . The figure displays that k NN and CAP have similar performance in the dataset and LPC is still the worst performing method. ILPC method is superior to the other three approaches because its performance achieves above 90% accuracy rate only. Besides, it is apparent that the proposed method outperforms k NN method above 20% accuracy rate when the value of k is 1. The result indicates that the Euler-Richardson method is beneficial for handling high-dimensional dataset. The proposed method improves considerable accuracy rate to k NN.

Table 3 summaries the performance of four methods on synthetic datasets with different dimensions. All methods achieve better performance when the dimension becomes

TABLE 4: The characteristics of real datasets used in the experiments.

Dataset	Sample number	Feature number	Class number
Iris	150	4	3
Wine	178	13	3
Iono	315	34	2
Sonar	208	60	2

larger. It can be observed that LPC is the worst method among the three approaches; the average accuracy rate is lower than 80% in all three synthetic datasets. LPC adopts the incorrect assumption; therefore, it achieves poor performance in the second experiment. LPC only outperforms the k NN and CAP methods in the 10D dataset, the heavily overlapped dataset. It seems that the LPC method solves the overlapping problem, but it does not adopt an incorrect formulation related to dimension. As a result, LPC method achieves poor performance in higher-dimensional datasets. The proposed method offers considerable improvement in 20D and 30D datasets. It suggests that Euler-Richardson method and multidimensional Parzen window have advantages for high dimensional datasets.

4.3. Real Dataset. The final experiment uses several real datasets from the UCI machine learning repository [28]. Table 4 lists the characteristics of the real datasets used in the experiment in terms of the size of datasets, the number of features, and the number of classes. All features are normalized to zero mean and unit variance. These four real datasets have only numerical features and no missing values. Although the number of four real datasets is small, Iono and Sonar datasets have higher dimension. To get impartial results, the study utilizes leave-one-out cross validation to measure performance.

The first dataset is the well-known Iris dataset. The performance with varying parameter k is shown in Figure 10. The Iris dataset is the smallest dataset of the four real datasets, with only 150 instances and four features. All methods show similar performance when $k < 20$. The performance of k NN and CAP is decreasing when $k > 20$. These results indicate that distance-based methods are sensitive to the value of k . In other words, the value of k is related to data distributions. The proposed method has stable performance with a wide range of k for usage. It seems that the posterior probability is a good measure for classification problem. LPC gets similar performance to LPC; however, it shows poor performance when k is very small. The detailed results are not shown in Figure 9; we list them as follows: 40.7% and 84.7% accuracy rate when $k = 1$ and 2 respectively. As mentioned earlier, the term R_j is inappropriate because the Euclidean distance is unreliable to the overlapping issue. It makes the term $\nabla p_j \|\Omega_j - q\|$ become incorrect, and LPC obtains unfavorable and incorrect results.

Figure 11 reports the classification accuracy of the four methods with varying parameter k for the Wine dataset. LPC method yields poor performance with an accuracy rate of 63.5% at $k = 1$. Besides, the performance of LPC is unstable

FIGURE 10: The comparison of classification accuracy with varying parameter k of four methods on the Iris dataset.

FIGURE 11: The comparison of classification accuracy with varying parameter k of four methods on the Wine dataset.

when $k < 11$. The CAP method presents good performance in this dataset. The idea of LPC is to give different weights to the categorical k nearest neighbors. Therefore, it should obtain better performance than CAP. However, the results show that the LPC method does not achieve better performance than CAP with varying parameter k . The results provide compelling evidence that there are some problems in weighting mechanism. However, the proposed method obtains better performance with a wide range k for usage. It could be indicated that the modified weighting formulation is more correct. All methods show a tendency of having accuracy rate when k becomes smaller. It means that there are sparse regions in Wine dataset. Therefore, it is difficult to obtain enough information from limited information in all methods.

The following experiments consider higher-dimensional datasets, Iono and Sonar. Figure 12 illustrates the accuracy rate as a function of k in Iono dataset. There is a high overlap between two classes, and noisy patterns are significant in Iono dataset. The proposed method shows excellent performance in this dataset. It suggests that the weighting average technique reduces the effect of noisy patterns and decreases the degree of overlap. Euler-Richardson method improves precision in a high-dimensional dataset. Thus, ILPC obtains better performance than CAP regardless of the

FIGURE 12: The comparison of classification accuracy with varying parameter k of four methods on the Iono dataset.

TABLE 5: The performance comparisons on real datasets.

	kNN		CAP		LPC		ILPC	
Dataset	Acc.	σ	Acc.	σ	Acc.	σ	Acc.	σ
Iris	94.65	1.803	94.51	1.893	94.09	8.872	95.66	0.553
Wine	96.77	0.878	97.68	0.766	96.99	6.518	98.32	1.050
Iono	82.85	2.724	89.70	0.965	82.64	2.091	92.86	0.412
Sonar	74.63	5.161	85.21	2.134	82.39	5.411	88.22	0.480

The bold font means the result of the proposed method.

value of k . CAP method only reduces the effect of noisy patterns. Therefore, CAP is the second best method. The LPC algorithm is the worst method among the four methods because it adopts an inappropriate formulation related to dimension. The results are consistent with those found in the second experiment with different dimensions. Compared to other real datasets, k NN method does not perform well in the Iono dataset. k NN suffers from the overlap problem and there are no strategies for dealing with noisy samples. In the situation, it is difficult to classify a query pattern correctly.

The Sonar is the highest-dimensional dataset of four real datasets. There is a medium overlap in Sonar dataset, but the noisy patterns are not significant. The performance of the four methods in the Sonar dataset with varying parameter k is depicted in Figure 13. It shows that ILPC has robust performance even the dimension is high. Only the proposed method achieves nearly 90% accuracy rate. The LPC method still reports poor performance when k is very small. The incorrect weighting formulation has a negative effect which prevents it from having higher accuracy rate than CAP method. Apparently, k NN shows the worst performance of all methods due to dimensionality curse [29, 30]. The dimension of the Sonar dataset is 60, but it includes only 208 samples. When the dimension is high, there is a high degree of correlation among features. In addition, Sonar has sparse regions because there are 208 samples in a high-dimensional space. In this situation, k NN easily obtain unreliable information by counting the major vote of k nearest neighbors.

The performance comparisons of each method on real datasets are listed in Table 5. It shows that all methods achieve above 90% accuracy rate in the smaller datasets, Wine and

FIGURE 13: The comparison of classification accuracy with varying parameter k of four methods on the Sonar dataset.

Iris. Considering Iono and Sonar dataset, ILPC obtains 92% and 88% accurate rate, respectively. It can be observed that ILPC improves above 10% accuracy rate compared to k NN method. The results are consistent with that found in the second experiment: the proposed method improves more in high-dimensional dataset. Besides, the proposed theorem plays an important role in our algorithm; multidimensional Parzen window and Euler-Richardson seem to provide a sufficient way to estimate the posterior probability of a sample in high dimensional datasets.

k NN method reveals its drawback in Sonar dataset. Only its performance is lower than 80% accuracy rate. However, ILPC method increases around 14% accuracy rate in Sonar dataset. ILPC has good performance in Iono and Sonar datasets. It seems that the weighting mechanism not only decreases the degree of overlap but also minimizes the effect of noisy pattern. On the whole, the proposed method yields above 92% accuracy rate in real datasets except Sonar dataset. It could be inferred that ILPC has robust performance in high-dimensional space.

4.4. Discussions. Numerous prior works have been focused on distance selection in k NN method. To our knowledge, limited work was done for overlapping issue. Although LPC method demonstrates improved performance when class distributions are overlapping, it suffers from many disadvantages. The most serious problem is it uses the product of two values to represent the inner product of two vectors. The incorrect assumption makes it perform poorly when k is smaller or in high-dimensional datasets. In this paper, the proposed theorem makes it faithfully calculate the inner product of two vectors. To reinforce its performance in high-dimensional space; multidimensional Parzen window and Euler-Richardson method are utilized for the proposed method.

To verify the performance of the proposed method, both artificial datasets and real datasets are used in our experiments. Since LPC has been proposed to overcome the overlapping problem, the first experiment uses artificial datasets with different degrees of overlapping to show performance. From the results, it can be observed that LPC reports worse performance when k is smaller. This is because there are no reciprocal relationships between the gradient vector

and the maximal radius R_j . Using insufficient assumption may lead to unsatisfying results. In general, ILPC is the best performing method, followed by LPC, CAP, and k NN in the first experiment. When the degree of overlapping increases, all methods have lower performance. It is noticed that CAP has worse performance than k NN in the $I - I$ dataset. It implies that distance-based method suffers from the overlapping problem. Choosing a proper metric will make a better prediction, such as the posterior probability.

We also notice that dimensionality is a hard issue for classification. In the second experiment, we use three artificial datasets with different dimensions to measure performance. It can be observed that all methods achieve better performance when the dimension becomes larger. Compared to k NN, ILPC has considerable improvement when k is smaller. It suggests that it is hard to determine the nearest neighbor in the high-dimensional dataset. Thus, k NN performs poorly in high-dimensional dataset. LPC obtains the worst performance in the second experiment because it adopts an incorrect formulation related to dimension. ILPC shows robust performance in high-dimensional dataset. The results gained benefits from multidimensional Parzen window and Euler-Richardson method.

The final experiment uses several real datasets to measure performance. The results of the real datasets are consistent with those found in previous experiments. LPC performs poorly when k is smaller or in high-dimensional datasets. In the experiment, we found that weighting mechanism not only decreases the degree of overlap but also minimizes the effect of noisy pattern. Thus, the proposed method achieves good performance in Iono and Sonar datasets.

We have reported that the proposed method has excellent and robust performance through experiments. In addition, the proposed theorem by mathematical derivation is shown in the paper. The theoretic property provides a novel way to calculate the inner product of two vectors. Then we have developed a numerical model to estimate the posterior probability. The experimental result is in agreement with the fundamental theorem. The proposed theorem can also be applied to applications such as pattern recognition and numerical problem.

In this paper, we have proposed a new classifier based on local probability centers. The proposed ILPC has only one parameter: the width of Gaussian kernel h . The best values of h are selected through experiment. Selecting the best value of k is another problem for k NN method. However, the proposed method shows stable performance with a wide range of k for usage. This is because the posterior probabilities of samples are based on statistical principle. Using nearer samples would make a good precise prediction to query pattern. As a result, it is a robust metric for classification even the class distributions are overlapping. According to previous discussions, we conclude that ILPC is a promising classifier which has advantages in the following.

- (1) Simplicity: ILPC is simple and is based on k nearest neighbor classifier. It is easy to obtain good performance because ILPC has only one parameter.

- (2) Robustness for overlapping issue: ILPC shows good performance in the artificial datasets with different degrees of overlapping.
- (3) Stable performance: ILPC yields good performance with a wide range of k for usage.
- (4) Robustness for dimensionality: ILPC shows good performance in the high-dimensional datasets.

Time complexity in classifiers is an important consideration. The computational costs of the k NN and CAP methods are both kN . The time complexity of the LPC method is k_0N^2 in the preprocessing step and $kN + kN_c$ in the classification stage. The term $p(w_j | (\Omega_j + q)/2)/p((\Omega_j + q)/2 | w_j)$ wastes classification time and can be simplified as $p(w_j | \Omega_j + q)/p(\Omega_j + q | w_j)$. Hence, ILPC method requires the same computational cost as LPC method. The classification time for k NN method is inefficient for application in large datasets because it must compare all training samples. Many papers have addressed prototype generation [13–17] or prototype selection methods [18–21] to obtain fewer samples and decrease classification time. Future works should choose an appropriate algorithm to reduce the classification time for ILPC.

5. Conclusions

In this paper, we have developed a new nearest neighbor algorithm called ILPC which is based on local probability centers. The original local probability centers method utilizes incorrect formulations which makes it achieve poor performance when k is very small and in high-dimensional datasets. We found that the major problem of LPC is it uses the product of two values to represent the inner product of two vectors. To deal with this problem, the paper shows that the gradient of the posterior probability function can be estimated in sufficient assumptions. The proposed theorem provides a correct model. In addition, multidimensional Parzen window and Euler-Richardson method are used to improve the accuracy rate in high dimensional space. Thus, the proposed method has offered a powerful model based on fundamental theorems. Experimental results show that the proposed method is superior to other methods. The proposed method achieves stable performance with a wide range of k for usage, robust performance to overlapping issue, and good performance to dimensionality. The theoretic property can be applied for numerical problem and other applications. The proposed algorithm has only one parameter and it is a promising classifier. Future work should select an appropriate algorithm to reduce the classification time for ILPC.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] P. E. Hart, D. G. Stock, and R. O. Duda, *Pattern Classification*, Wiley, Hoboken, NJ, USA, 2nd edition, 2001.
- [2] S. A. Mahmoud and W. G. Al-Khatib, "Recognition of Arabic (Indian) bank check digits using log-gabor filters," *Applied Intelligence*, vol. 35, no. 3, pp. 445–456, 2011.
- [3] H. Malek, M. M. Ebadzadeh, and M. Rahmati, "Three new fuzzy neural networks learning algorithms based on clustering, training error and genetic algorithm," *Applied Intelligence*, vol. 37, no. 3, pp. 280–289, 2012.
- [4] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest-neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1281–1285, 2002.
- [5] R. Paredes and E. Vidal, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1100–1110, 2006.
- [6] J. Wang, P. Neskovic, and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognition Letters*, vol. 28, no. 2, pp. 207–213, 2007.
- [7] M. Z. Jahromi, E. Parvinnia, and R. John, "A method of learning weighted similarity function to improve the performance of nearest neighbor," *Information Sciences*, vol. 179, no. 17, pp. 2964–2973, 2009.
- [8] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 6, pp. 607–616, 1996.
- [9] R. D. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 622–627, 1981.
- [10] B. Chen, H. Liu, J. Chai, and Z. Bao, "Large margin feature weighting method via linear programming," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1475–1488, 2009.
- [11] W. Hui, "Nearest neighbors by neighborhood counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 942–953, 2006.
- [12] C.-M. Hsu and M.-S. Chen, "On the design and applicability of distance functions in high-dimensional data space," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 4, pp. 523–536, 2009.
- [13] C. J. Veenman and M. J. T. Reinders, "The nearest subclass classifier: a compromise between the nearest mean and nearest neighbor classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1417–1429, 2005.
- [14] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
- [15] L. Nanni and A. Lumini, "Particle swarm optimization for prototype reduction," *Neurocomputing*, vol. 72, no. 4–6, pp. 1092–1097, 2009.
- [16] W. Lam, C.-K. Keung, and D. Liu, "Discovering useful concept prototypes for classification based on filtering and abstraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1075–1090, 2002.
- [17] I. Triguero, J. Derrac, S. García, and F. Herrera, "A taxonomy and experimental study on prototype generation for nearest neighbor classification," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 42, no. 1, pp. 86–100, 2012.
- [18] S. García, J. Derrac, J. R. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

- [19] H. A. Fayed and A. F. Atiya, "A novel template reduction approach for the K-nearest neighbor method," *IEEE Transactions on Neural Networks*, vol. 20, no. 5, pp. 890–896, 2009.
- [20] F. Angiulli, "Fast nearest neighbor condensation for large data sets classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 11, pp. 1450–1464, 2007.
- [21] D. Randall Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 257–286, 2000.
- [22] B. Li, Y. W. Chen, and Y. Q. Chen, "The nearest neighbor algorithm of local probability centers," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 38, no. 1, pp. 141–154, 2008.
- [23] S. Hotta, S. Kiyasu, and S. Miyahara, "Pattern recognition using average patterns of categorical k-nearest neighbors," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 412–415, August 2004.
- [24] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 3rd edition, 2006.
- [25] H. Gould, J. Tobochnik, and W. Christian, *An Introduction to Computer Simulation Methods: Applications to Physical System*, Addison Wesley, 3rd edition, 2006.
- [26] Y. Zeng, Y. Yang, and L. Zhao, "Pseudo nearest neighbor rule for pattern classification," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3587–3595, 2009.
- [27] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 2nd edition, 1990.
- [28] C. Blake, E. Keogh, and C. J. Merz, "UCI Repository of Machine Learning Databases," Department of Information and Computer Science, University of California, 2009, <http://www.ics.uci.edu/~mllearn>.
- [29] F. Korn, B.-U. Pagel, and C. Faloutsos, "On the "dimensionality curse" and the "self-similarity blessing"," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 96–111, 2001.
- [30] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, "What is the nearest neighbor in high dimensional spaces?" in *Proc. the 26th International Conference on Very Large Data Bases*, pp. 506–515, 2000.