Bo Tang and Haibo He
Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA

ENN: Extended Nearest Neighbor Method for Pattern Recognition

Abstract

This article introduces a new supervised classification method – the extended nearest neighbor (ENN) – that predicts input patterns according to the *maximum gain of intra-class coherence*. Unlike the classic k-nearest neighbor (KNN) method, in which only the nearest neighbors of a test sample are used to estimate a group membership, the ENN method makes a prediction in a "two-way communication" style: it considers *not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors*. By exploiting the generalized class-wise statistics from all training data by iteratively assuming all the possible class memberships of a test sample, the ENN is able to learn from the global distribution, therefore improving pattern recognition performance and providing a powerful technique for a wide range of data analysis applications.

I. Introduction

With the continuous expansion of data availability in many areas of engineering and science, such as biology, communications, social networks, global climate, and remote sensing, it becomes critical to identify patterns from vast amounts of data, to identify members of a predefined class (classification), or to group patterns based on their similarities (clustering). In the setting of a classification, there are two kinds of classifiers which scientists and engineers may use: *parametric* classifiers in which the underlying joint distributions/models of the data are assumed to be known but certain parameters need to be estimated, and *nonparametric* classifiers in which the classification rules do not depend explicitly on the data's underlying distributions [1]. With the coming of the Big Data era [2] [3] [4], nonparametric classifiers have received particular attention because the data distributions/models of many classification problems are either unknown or very difficult to obtain in practice. Previous nonparametric classification methods, such as k-nearest neighbor (KNN), assume that data which are close together based upon some metrics, such as Euclidean distance, more likely belong to the same category. Therefore, given an unknown sample to be classified, its nearest neighbors are first ranked and counted, and then a class membership assignment is made.

©DIGITAL VISION

Nearest neighbor-based pattern recognition methods have several key advantages, such as easy implementation, competitive performance, and a nonparametric computational basis which is independent of the underlying data distribution. The first modern study of the nearest neighbor approach can be traced back to 1951 by Fix and Hodges [5]. In their formalization of nonparametric discrimination, the consistency of KNN was established using a probability density estimation. They proved that if $k \to \infty$ and $k/n \to 0$, where k is the number of selected nearest neighbors and n is the sample size of the whole data set, the classification error of the nearest neighbor method (R_n) can asymptotically converge to the Bayesian rule (R^*): $R_n \to R^*$. In another representative work [6], Cover and Hart proved that for N-class classification problems, when $k = 1$, the classification error of the nearest neighbor method will be bounded by $R^* \leq R_n \leq R^* \; (2-(N/(N-1))R^*)$ when there are infinite samples. This basically means in the large sample case, the nearest neighbor method has "a probability of error which is less than twice the Bayes probability of error" [6]. These early classic works laid down a strong theoretical foundation for nearest neighbor based methods, which have been witnessed in considerable applications in many different disciplines, such as biological and

Digital Object Identifier 10.1109/MCI.2015.2437512
Date of publication: 16 July 2015

Corresponding author: Haibo He (Email: he@ele.uri.edu).

1556-603X/15©2015IEEE

Online Resource for ENN:

http://www.ele.uri.edu/faculty/he/ENN/ENN.html

Supplementary materials and source code implementation of the ENN classifier have been provided. The source code includes all three versions of the ENN classifier, **ENN, ENN.V1** and **ENN.V2**, in Matlab M-file format.

chemical data analysis [7], disease classification and clinical outcome prediction [8], among others. In addition to pattern recognition, KNN based methods are also widely used for clustering [9], regression [10], density estimation [11], and outlier detection [12], to name a few. In fact, KNN has been identified as one of the "top 10 algorithms in data mining" by the IEEE International Conference on Data Mining (ICDM) presented in Hong Kong in 2006 [13].

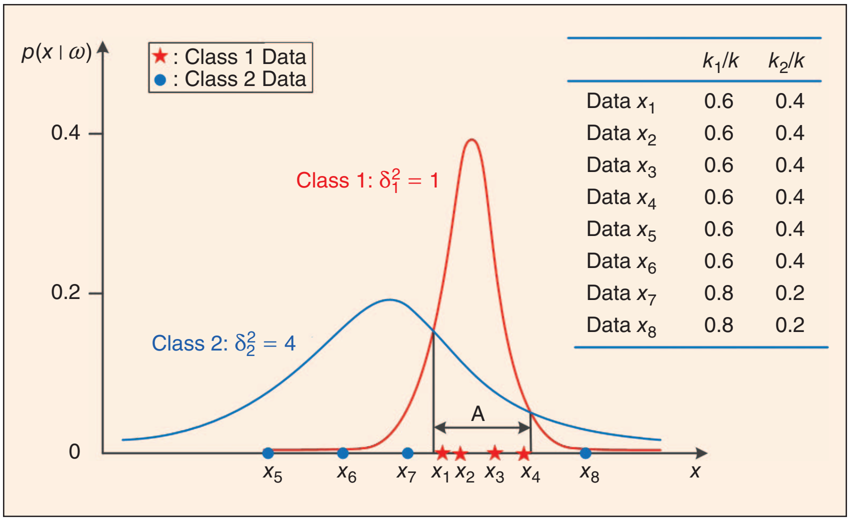
There are several important issues in the KNN method that have been of great interest since it was originally proposed. The first one is how to choose the user-defined parameter k, the number of nearest neighbors. To determine the best k, a straightforward method is through cross validation by trying various k and choose the one with the best performance [14]. The second one is how to determine an appropriate dissimilarity measurement. The Euclidean distance is commonly used for this purpose, but it is easily influenced by irrelevant features therefore suffering from noisy samples, especially with high-dimensional data (i.e., the curse of dimensionality problem). Feature normalization and dimensionality reduction methods are able to remedy this issue in certain ways. More recently, distance metric learning from training data is of great interest to improve the prediction performance of KNN based methods [15]–[21]. For example, Goldberger et al. proposed a stochastic nearest neighbor approach for distance metric learning with the optimum leave-one-out performance on training data [15]. Weinberger et al. proposed to learn a Mahalanobis distance metric by semidefinite programming [16] [17]. Hastie and Tibshirani used a local linear discriminant analysis to estimate distance metrics for computing neighbors and building local decision boundaries for classification [21]. The third issue in the KNN method is its high computational complexity for large data sets. To address this issue, Gowda et al. proposed a condensed nearest neighbor approach using a concept of mutual nearest neighbors to reduce the size of training data for decision making [22]. Bagui et al. employed a concept of ranking to reduce the computational complexity [23]. To speed up the searching of nearest neighbors and overcome the memory limitation, many data structures, named fast KNN, have been proposed to implement the KNN method, such as k-d tree [24], nearest feature line [25], orthogonal search tree [26], ball-tree [27], and principal axis search tree [28], to name a few.

In general, two types of errors may occur in KNN based methods: for the samples in the areas of higher density, the k nearest neighbors could lie in the areas of slightly lower density, or vice versa [29]. These types of errors may result in misclassification, especially when nearest neighbors are dominated by the samples from other classes. Error rates are increased when sample sizes are small or data are imbalanced (see a comprehensive survey on imbalanced learning [30]). In this paper, we introduce extended nearest neighbor (ENN), a new method based on generalized class-wise statistics which can represent intra-class coherence. Unlike the classic KNN method where only the nearest neighbors are considered for classification, our proposed ENN method takes advantage of all available training data to make a classification decision: it assigns a class membership to an unknown sample to maximize the intra-class coherence. By exploiting the information from all available data to maximize the intra-class coherence, ENN is able to learn from the global distribution, therefore, improving pattern recognition performance.

II. Problem Formulation: Limitations of KNN

The reason of the "two types of errors" [29] is that KNN method is sensitive to the scale or variance of the distributions of the predefined classes [31]. The nearest neighbors of an unknown observation will tend to be dominated by the class with the highest density. For example, in a two-class classification problem, if we assume that class 1 has a smaller variance than class 2, then the class 1 samples dominate their near neighborhood with higher density (i.e., more concentrated distribution), whereas the class 2 samples are distributed in regions with lower density (i.e., more spread out distribution). In this case, for class 2 samples, the number of nearest neighbors from class 1 may exceed that from class 2. For instance, for those class 2 samples which are close to the region of class 1, there may be a large number of class 1 neighbors because of the higher density of class 1. Only for those class 2 samples which are far away from the region of higher density of class 1, their nearest neighbors from class 2 will be dominant.

Fig. 1 shows one example (a two-class classification scenario) of decision making in the classic KNN method for two Gaussian distributions with different means and variances, in which we assume that their prior distribution $p(\omega)$ are the same. In this figure, the x-axis represents the data value (one dimensional data in this case), and the y-axis represents the class-conditional probability $p(x|\omega)$. For the eight data points illustrated here, we assume that data points x_1, x_2, x_3 and x_4 are sampled from the class 1 distribution, and data points x_5, x_6, x_7 and x_8 are sampled from the class 2 distribution. On the side of this figure, we also listed their corresponding k_1/k and k_2/k ratio (here $k = 5$ is the parameter of KNN, and k_1 and k_2 are the number of nearest neighbors belong to class 1 and class 2, respectively). In this case, the Bayesian method can correctly classify all these eight data points by calculating their corresponding posterior probabilities according to the Bayes' theorem. However, under the KNN method, all of these eight data points are predicted as class 1. This means



	k_1/k	k_2/k
Data x_1	0.6	0.4
Data x_2	0.6	0.4
Data x_3	0.6	0.4
Data x_4	0.6	0.4
Data x_5	0.6	0.4
Data x_6	0.6	0.4
Data x_7	0.8	0.2
Data x_8	0.8	0.2

FIGURE 1 One example of the KNN rule in comparison with the Bayesian rule for a two-class classification problem. Assume that data points x_1, x_2, x_3 and x_4 are sampled from the class 1 distribution, and data points x_5, x_6, x_7 and x_8 are sampled from the class 2 distribution. In this case, the Bayesian rule will correctly classify all these eight data points. However, the KNN rule (with $k = 5$) will misclassify the four data points from class 2: x_5, x_6, x_7 and x_8. This figure demonstrates the well-known limitation of the classic KNN method.

that the four data points from class 2, x_5, x_6, x_7 and x_8, are misclassified because their k_1/k ratio values are larger than their corresponding k_2/k ratio values. The reason is that the nearest neighbors of these four data points are dominated by the class 1 data because of the distribution. This has been a long-standing limitation of the classic KNN method in literature [29] [31].

III. Proposed Approach: ENN for Classification

Here we describe the Extended Nearest Neighbor (ENN) method to solve this deficiency in the classic KNN method. The advantages of the original KNN approach are retained in our ENN classifier, such as easy implementation and competitive classification performance. However, unlike the classic KNN approach, the new *ENN classifier makes a prediction by not only considering who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors.*

A. ENN Classifier

We describe our ENN method starting with the *two-class classification* problem. The generalized statistic based on nearest neighbors was first proposed in the two-sample test problem to evaluate whether two distributions are mixed well or widely spread [32], [33]. In our approach, unlike the generalized statistic over all samples as discussed in [33], we first build our generalized class-wise statistic regarding the pooled samples S_1 and S_2 for each class along with its nearest neighbors. We define the generalized class-wise statistic T_i for class i as the following:

$$T_i = \frac{1}{n_i k}\sum_{\mathbf{x}\in S_i}\sum_{r=1}^{k} I_r(\mathrm{x}, S = S_1 \cup S_2)$$

$$i = 1,2 \qquad (1)$$

where S_1 and S_2 denote the samples in class 1 and class 2, respectively, x denotes one single sample in $S = S_1 \cup S_2$, n_i is the number of samples in S_i, and k is the user-defined parameter of the number of the nearest neighbors. The indicator function $I_r(\mathrm{x}, S)$ indicates whether both the sample x and its r-th nearest neighbor belong to the same class, defined as follows:

$$I_r(\mathrm{x}, S) = \begin{cases} 1, & \text{if } \mathrm{x} \in S_i \text{ and } \mathrm{NN}_r(\mathrm{x}, S) \in S_i \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

where $\mathrm{NN}_r(\mathrm{x}, S)$ denotes the r-th nearest neighbor of x in S. This equation means for either class, if both the sample x and its r-th nearest neighbor in the pool of S belong to the same class, then the outcome of the indicator function $I_r(\mathrm{x}, S)$ equals 1; otherwise, it equals 0. In this way, the generalized class-wise statistic T_i in Eq. (1) is the ratio of the number of nearest neighbors belonging to the same class for class i with respect to the product of the sample size of class i (i.e., n_i) and the number of nearest neighbors under consideration (k). A large T_i indicates the samples in S_i are much closer together and their nearest neighbors are dominated by the same class samples, whereas a small T_i indicates that samples in S_i have an excess of nearest neighbors from the other class. Note that the generalized class-wise statistic has $0 \le T_i \le 1$ with $T_i = 1$ when all the nearest neighbors of class i data are also from the same class i, and $T_i = 0$ when all the nearest neighbors are from other classes. Based on this discussion, we can use T_i to represent the data distribution across multiple classes. Therefore, we introduce the concept of *intra-class coherence*, defined as follows:

$$\Theta = \sum_{i=1}^{2} T_i \qquad (3)$$

Given an unknown sample Z to be classified, we iteratively assign it to class 1 and class 2, respectively, to obtain two new generalized class-wise statistics T_i^j, where $j = 1, 2$.

$$T_i^j = \frac{1}{n_i' k}\sum_{\mathbf{x}\in S'_{i,j}}\sum_{r=1}^{k} I_r(\mathbf{x}, S' = S_1 \cup S_2 \cup \{Z\})$$

$$i,j = 1,2 \qquad (4)$$

where n_i' is the size of $S'_{i,j}$ and $S'_{i,j}$ is defined as

$$S'_{i,j} = \begin{cases} S_i \cup \{Z\}, & \text{when } j = i \\ S_i, & \text{when } j \ne i \end{cases} \qquad (5)$$

For this two-class classification problem, we have four generalized class-wise statistics: T_1^1, T_2^1, T_1^2 and T_2^2. Unlike the previous KNN classifiers which predict that the unknown sample Z belongs to a class based only on its nearest neighbors, the ENN classifier predicts its class membership according to the following target function:

$$f_{\mathrm{ENN}} = \arg\max_{j\in 1,2} \sum_{i=1}^{2} T_i^j = \arg\max_{j\in 1,2} \Theta^j \qquad (6)$$

where

$$\Theta^j = \sum_{i=1}^{2} T_i^j \tag{7}$$

This means that our ENN method makes the prediction based on which decision resulting the largest intra-class coherence when we iteratively assume the test sample Z to be each possible class.

We now present the detailed calculation steps for a two-class classification problem using our ENN method. First, let's assume that the observation Z should be classified as class 1. Then, for each training sample, we re-calculate its k nearest neighbors and obtain the new generalized class-wise statistic according to Eq. (4), denoted as T_1^1 for class 1 and T_2^1 for class 2. Second, we assume that the observation Z should be classified as class 2; then, for each training sample, we also re-calculate its k nearest neighbors and obtain the new generalized class-wise statistic according to Eq. (4), denoted as T_1^2 for class 1 and T_2^2 for class 2. Then a classification decision is made according to Eq. (6).

The aforementioned two-class ENN decision rule can be easily extended to *multi-class classification* problems. Specifically, for an *N-class classification* problem, we have an ENN Classifier algorithm (see the box below).

B. ENN.V1 Classifier: An Equivalent Version of ENN

The underlying idea of the ENN rule shown in Eq. (8) is that we classify the test sample Z based on which decision results in the greatest intra-class coherence among all possible classes. We can further demonstrate that the classification of Z in our ENN rule depends not only on who are the nearest neighbors of Z, but also on who consider Z as one of their nearest neighbors.

ENN Classifier: Given an unknown sample Z to be classified, we iteratively assign it to each possible class $j, j = 1, 2, \cdots, N,$ and compute the generalized class-wise statistic T_i^j for each class $i, i = 1, 2, \cdots, N$. Then, the sample Z is classified according to:

$$f_{\text{ENN}} = \underset{j \in 1,2,\cdots,N}{\arg\max} \sum_{i=1}^{N} T_i^j \tag{8}$$

For the new observation Z, let's assume that there are k_1 nearest neighbors from class 1 and k_2 nearest neighbors from class 2, where $k_1 + k_2 = k$ is the total number of nearest neighbors investigated. First, let's assume the observation Z to be classified as class 1; then, we count the number of class 1 data who have an *increased* number of class 1 samples in its k nearest neighbors (denote this number as Δn_1^1), and we also count the number of class 2 data who have a *decreased* number of class 2 samples in its k nearest neighbors (denote this number as Δn_2^1). Then, in a similar way, we further assume the observation Z to be a member of class 2, and count the number of class 1 data who have a *decreased* number of class 1 samples in its k nearest neighbors (denote this number as Δn_1^2), and we also count the number of class 2 data who have an *increased* number of class 2 samples in its k nearest neighbors (denote this number as Δn_2^2). In this way, Δn_i^j represents how many of the k nearest neighbors from each class will change because of the introduction of the new sample Z, when it is iteratively assumed to be each possible class. To clearly demonstrate this concept, we present a detailed calculation example in our Supplementary Material Section 1 for a two-class classification problem.

With these discussions, we present an equivalent version of the ENN classifier in the box ENN.V1 (see below).

Proof of ENN.V1. We iteratively assign the unknown sample Z to class j to obtain new generalized class-wise statistic T_i^j for class i. According to Eq. (4), we have,

when $i = j$

$$\begin{aligned} T_i^j &= \frac{1}{n_i' k} \sum_{X \in S_i \cup \{Z\}} \sum_{r=1}^{k} I_r(X, S' = S_1 \cup S_2 \cup \{Z\}) \\ &= \frac{1}{(n_i+1)k}\Bigg[\sum_{X \in S_i} \sum_{r=1}^{k} I_r(X, S_1 \cup S_2 \cup \{Z\}) \\ &\qquad + \sum_{r=1}^{k} I_r(Z, S_1 \cup S_2 \cup \{Z\})\Bigg] \\ &= \frac{1}{(n_i+1)k}\Bigg[\sum_{X \in S_i} \sum_{r=1}^{k} I_r(X, S_1 \cup S_2) + \Delta n_i^j \\ &\qquad + \sum_{r=1}^{k} I_r(Z, S_1 \cup S_2)\Bigg] \\ &= \frac{1}{(n_i+1)k}(n_i k T_i + \Delta n_i^j + k_i) \end{aligned} \tag{10}$$

and when $i \neq j$

$$\begin{aligned} T_i^j &= \frac{1}{n_i' k} \sum_{X \in S_i} \sum_{r=1}^{k} I_r(X, S' = S_1 \cup S_2 \cup \{Z\}) \\ &= \frac{1}{n_i k}\left[\sum_{X \in S_i} \sum_{r=1}^{k} I_r(X, S_1 \cup S_2) - \Delta n_i^j\right] \\ &= \frac{1}{n_i k}(n_i k T_i - \Delta n_i^j) \\ &= T_i - \frac{\Delta n_i^j}{n_i k} \end{aligned} \tag{11}$$

Therefore, from Eq. (8), we have:

$$\begin{aligned} f_{\text{ENN}} &= \underset{j \in 1,2,\cdots,N}{\arg\max} \sum_{i=1}^{N} T_i^j \\ &= \underset{j \in 1,2,\cdots,N}{\arg\max} \sum_{i=1}^{N} (T_i^j - T_i) \\ &= \underset{j \in 1,2,\cdots,N}{\arg\max} \Bigg\{ (T_i^j - T_i)_{i=j} \\ &\qquad + \sum_{i \neq j}^{N} (T_i^j - T_i) \Bigg\} \end{aligned}$$

ENN.V1: Given an unknown sample Z to be classified, we iteratively assign it to each possible class $j, j = 1, 2, \cdots, N,$ and predict the class membership according to:

$$f_{\text{ENN.V1}} = \underset{j \in 1,2,\cdots,N}{\arg\max} \left\{ \left(\frac{\Delta n_i^j + k_i - kT_i}{(n_i+1)k} \right)_{i=j} - \sum_{i \neq j}^{N} \frac{\Delta n_i^j}{n_i k} \right\} \tag{9}$$

where k is the user-defined parameter of the number of the nearest neighbors, n_i is the number of training data for class i, k_i is the number of the nearest neighbors of the test sample Z from class $i, \Delta n_i^j$ represents the change of the k nearest neighbors for class i when the test sample Z is assumed to be class j, and T_i represents the generalized class-wise statistic of original class i (i.e., without the introduction of the test sample Z).

$$= \underset{j\in 1,2,\cdots,N}{\arg\max}\left\{\left(\frac{\Delta n_i^j + k_i - kT_i}{(n_i+1)\,k}\right)_{i=j} - \sum_{i\neq j}^{N}\frac{\Delta n_i^j}{n_i k}\right\}$$

$$= f_{\mathrm{ENN.V1}} \tag{12}$$

□

This proves our ENN.V1 is equivalent to the original ENN method. We now present an example in Fig. 2 for the detailed implementation of our two ENN versions, i.e., f_{ENN} in Eq. (8) and $f_{\mathrm{ENN.V1}}$ in Eq. (9). Note that this equivalent target function $f_{\mathrm{ENN.V1}}$ can provide the same classification result, without the recalculation of T_i^j as in f_{ENN}. Both target functions provide a general formula for our ENN method for multi-class classification problems. In practical applications, whether using f_{ENN} or $f_{\mathrm{ENN.V1}}$ is a user's preference. We would like to note that f_{ENN} is more straightforward to calculate and implement, but from a computational point of view, we recommend using $f_{\mathrm{ENN.V1}}$ as this target function does not require recalculating the generalized class-wise statistic T_i^j for every test sample.

C. ENN.V2 Classifier: An Approximate Version of ENN

The proposed ENN classifier can resolve the scale-sensitive problem in the classic KNN decision rule as we discussed previously in this paper. From Eq. (8), the ENN method classifies the test sample Z based on which decision results in the greatest intra-class coherence among all predefined classes. Moreover, from Eq. (9), the classification decision rule of our ENN method is based on both the samples who are the nearest neighbors of Z and who regard Z as one of their nearest neighbors. We provide an approximate but straightforward version of the ENN method under certain assumptions in the box ENN.V2.

> ENN.V2: Given an unknown sample Z to be classified, under the following two conditions:
>
> (1) All classes have the same number of data samples n, i.e., a balanced classification problem;
>
> (2) For all $i \neq j, (\Delta n_i^j/((n+1)nk)) \to 0$;
>
> The ENN decision rule can be approximated as follows:
>
> $$f_{\mathrm{ENN.V2}} = \underset{j\in 1,2,\cdots,N}{\arg\max}\{\Delta n_j + k_j - kT_j\} \tag{13}$$
>
> where k is the user-defined parameter of the number of the nearest neighbors, Δn_j denotes the number of samples in class j who consider Z as one of their k nearest neighbors, k_j is the number of the nearest neighbors of the test sample Z from class j, and T_j represents the generalized class-wise statistic of original class j (i.e., without the introduction of the test sample Z).

Proof of ENN.V2. Considering two-class classification scenario first, without loss of generality, if the ENN method classifies Z as class 2 according to Eq. (9), we have

$$\frac{\Delta n_1^1 + k_1 - kT_1}{(n+1)\,k} - \frac{\Delta n_2^1}{nk} < \frac{\Delta n_2^2 + k_2 - kT_2}{(n+1)\,k} - \frac{\Delta n_1^2}{nk}$$

$$\Rightarrow \frac{\Delta n_1^1 + \Delta n_1^2 + k_1 - kT_1}{(n+1)\,k} + \frac{\Delta n_1^2}{(n+1)\,nk} < \frac{\Delta n_2^2 + \Delta n_2^1 + k_2 - kT_2}{(n+1)\,k} + \frac{\Delta n_2^1}{(n+1)\,nk} \tag{14}$$

When $\Delta n_1^2/((n+1)\,nk)$ and $\Delta n_2^1/((n+1)\,nk)$ approximate to zero, we have

$$\Delta n_1^1 + \Delta n_1^2 + k_1 - kT_1 < \Delta n_2^2 + \Delta n_2^1 + k_2 - kT_2 \tag{15}$$

We now define

$$\Delta n_1 = \Delta n_1^1 + \Delta n_1^2 \tag{16}$$

$$\Delta n_2 = \Delta n_2^1 + \Delta n_2^2 \tag{17}$$

In this way, Δn_1 represents the total number of samples in class 1 who consider Z as one of their k nearest neighbors, and Δn_2 represents the total number of samples in class 2 who consider Z as one of their k nearest neighbors. Therefore, Eq. (15) can be expressed as follows:

$$\Delta n_1 + k_1 - kT_1 < \Delta n_2 + k_2 - kT_2 \tag{18}$$

Therefore, for a two-class classification problem, the classification rule is

$$f_{\mathrm{ENN.V2}} = \underset{j\in 1,2}{\arg\max}\{\Delta n_j + k_j - kT_j\} \tag{19}$$

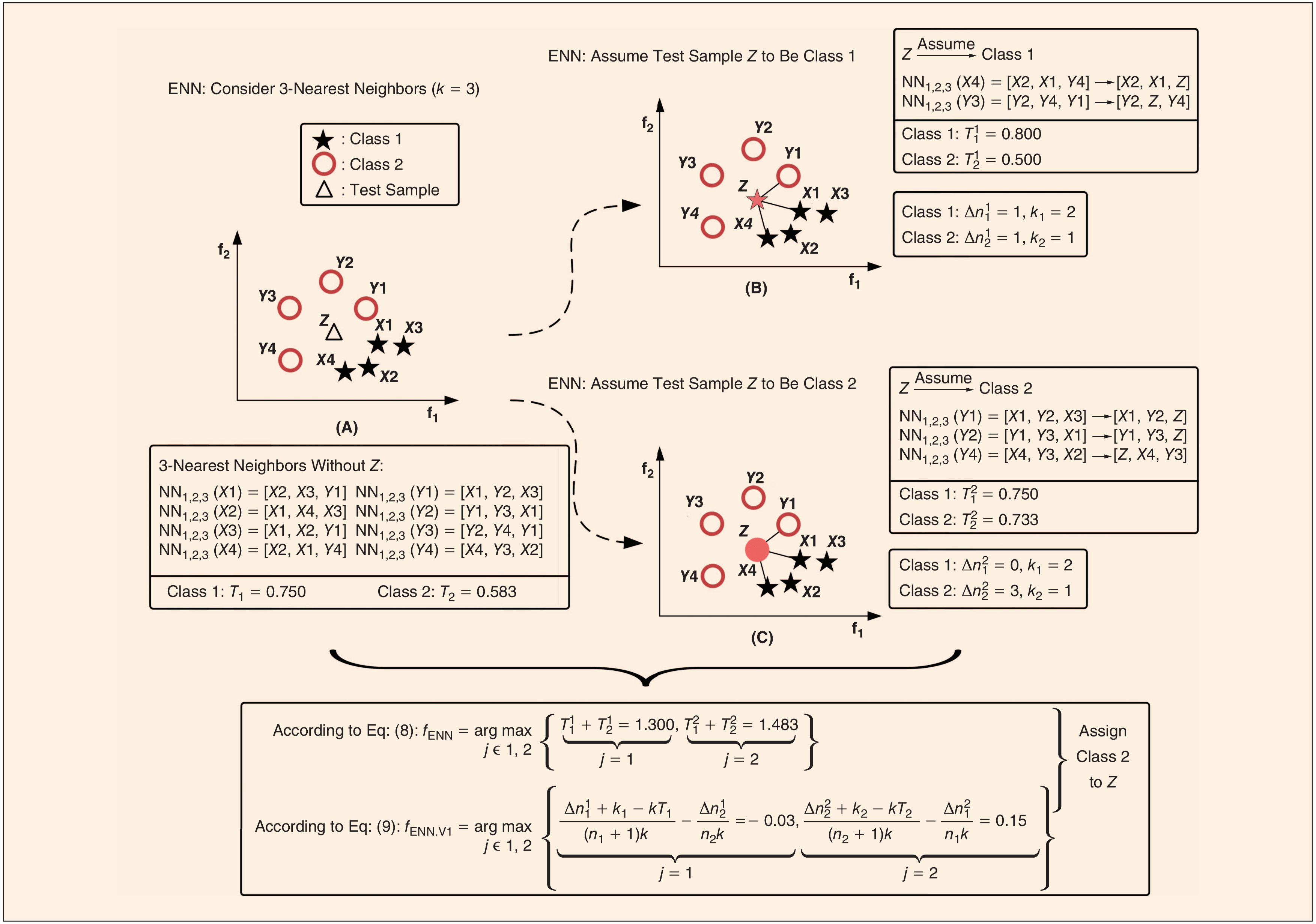
□

The above derivation can be easily extended to N-class classification problems, and we can obtain the approximate version of our ENN method as shown in Eq. (13).

This approximate version of ENN in Eq. (13) also explains why the proposed ENN method can address the scale-sensitive problem in the classic KNN rule. Whereas the KNN rule only considers k_j to make a decision, the proposed ENN method considers three factors to make a prediction decision: two "positive" terms k_j and Δn_j, and one "negative" term kT_j. The two positive terms demonstrate that our ENN approach considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. The negative term means the class that has a greater generalized class-wise statistic would be given a larger penalty value when we estimate the class membership of an unknown sample. The combination of these three factors provides the unique advantage of our ENN method to improve classification performance. We would also like to note that in many practical applications if the two conditions as described in ENN.V2 algorithm are satisfied, using Eq. (13) as a simple approximation of our ENN method can in general provide competitive classification performance.

D. Computational Analysis

As a simple and reliable technique, the nearest neighbor based methods have been widely used in both research and industry for pattern recognition, regression, feature reduction, clustering, among others. However, one major concern of this kind of method is its computational complexity. For the classic


ENN: Consider 3-Nearest Neighbors ($k = 3$)

★ : Class 1
○ : Class 2
△ : Test Sample

3-Nearest Neighbors Without Z:

$NN_{1,2,3}(X1) = [X2, X3, Y1]$ $NN_{1,2,3}(Y1) = [X1, Y2, X3]$
$NN_{1,2,3}(X2) = [X1, X4, X3]$ $NN_{1,2,3}(Y2) = [Y1, Y3, X1]$
$NN_{1,2,3}(X3) = [X1, X2, Y1]$ $NN_{1,2,3}(Y3) = [Y2, Y4, Y1]$
$NN_{1,2,3}(X4) = [X2, X1, Y4]$ $NN_{1,2,3}(Y4) = [X4, Y3, X2]$

Class 1: $T_1 = 0.750$ Class 2: $T_2 = 0.583$

ENN: Assume Test Sample Z to Be Class 1

$Z \xrightarrow{\text{Assume}} \text{Class 1}$

$NN_{1,2,3}(X4) = [X2, X1, Y4] \rightarrow [X2, X1, Z]$
$NN_{1,2,3}(Y3) = [Y2, Y4, Y1] \rightarrow [Y2, Z, Y4]$

Class 1: $T_1^1 = 0.800$
Class 2: $T_2^1 = 0.500$

Class 1: $\Delta n_1^1 = 1, k_1 = 2$
Class 2: $\Delta n_2^1 = 1, k_2 = 1$

ENN: Assume Test Sample Z to Be Class 2

$Z \xrightarrow{\text{Assume}} \text{Class 2}$

$NN_{1,2,3}(Y1) = [X1, Y2, X3] \rightarrow [X1, Y2, Z]$
$NN_{1,2,3}(Y2) = [Y1, Y3, X1] \rightarrow [Y1, Y3, Z]$
$NN_{1,2,3}(Y4) = [X4, Y3, X2] \rightarrow [Z, X4, Y3]$

Class 1: $T_1^2 = 0.750$
Class 2: $T_2^2 = 0.733$

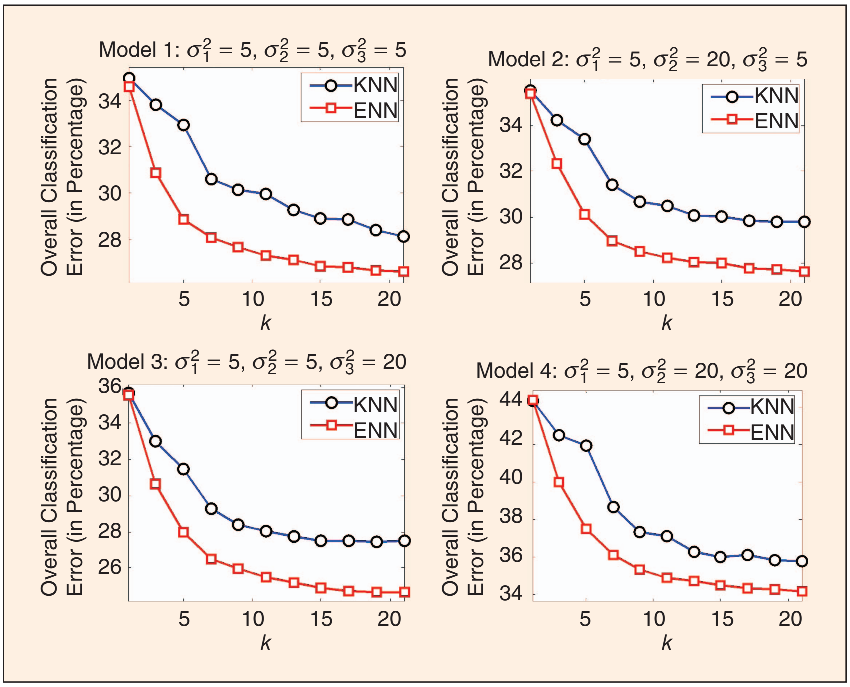
Class 1: $\Delta n_1^2 = 0, k_1 = 2$
Class 2: $\Delta n_2^2 = 3, k_2 = 1$

According to Eq: (8): $f_{\text{ENN}} = \underset{j \in 1,2}{\arg\max} \left\{ \underbrace{T_1^1 + T_2^1 = 1.300}_{j=1}, \underbrace{T_1^2 + T_2^2 = 1.483}_{j=2} \right\}$

According to Eq: (9): $f_{\text{ENN.V1}} = \underset{j \in 1,2}{\arg\max} \left\{ \underbrace{\frac{\Delta n_1^1 + k_1 - kT_1}{(n_1+1)k} - \frac{\Delta n_2^1}{n_2 k} = -0.03}_{j=1}, \underbrace{\frac{\Delta n_2^2 + k_2 - kT_2}{(n_2+1)k} - \frac{\Delta n_1^2}{n_1 k} = 0.15}_{j=2} \right\}$

Assign Class 2 to Z

FIGURE 2 (A), (B) and (C) demonstrate the detailed procedure of two versions of ENN classifier. In (A), training samples are stored and their k nearest neighbors and distances are calculated to compute the generalized class-wise statistic T_1 and T_2. (B) and (C) illustrate the changes of the generalized class-wise statistic for each class, when we iteratively assume Z to be class 1 and class 2. Both of them provide the same classification result, but the recalculation of generalized class-wise statistic T_i^j is avoided in ENN.V1.


Model 1: $\sigma_1^2 = 5$, $\sigma_2^2 = 5$, $\sigma_3^2 = 5$

Model 2: $\sigma_1^2 = 5$, $\sigma_2^2 = 20$, $\sigma_3^2 = 5$

Model 3: $\sigma_1^2 = 5$, $\sigma_2^2 = 5$, $\sigma_3^2 = 20$

Model 4: $\sigma_1^2 = 5$, $\sigma_2^2 = 20$, $\sigma_3^2 = 20$

FIGURE 3 Overall classification error rate (in percentage) of ENN and KNN for four Gaussian data models.

KNN method, there is no training stage (i.e., the so-called instance-based learning or lazy learning), with a computational complexity of $O(M \log M)$ in a testing stage for every test sample, where M is the number of training data. To improve the efficiency, it makes more sense to take a preprocessing stage where data structures are built to construct the relationship among training data, such as k-d tree [24], ball tree [27], and nearest feature line [25], or to reduce the size of training data by eliminating the data which may not provide useful information for decision making, such as the condensed nearest neighbor rule by using a concept of mutual nearest neighborhood to only select samples close to the decision boundary [22].

In our ENN method, a preprocessing stage can be developed to calculate the generalized class-wise statistic T_i for each class to build weighted KNN graphs, in which training samples are the vertices and the distances of the sample to its nearest neighbors are the edges. The weighted KNN graphs can then be used to calculate Δn_i^j efficiently for a given test sample. To do so, we calculate distances between the test sample and every training data to obtain k_i for each class at a computational complexity of $O(M \log M)$. After that, we compare this distance within the weighted KNN graphs to obtain Δn_i^j with only computational complexity of $O(M)$. Therefore, the total computational complexity of our proposed ENN method is $O(M \log M) + O(M)$, which is on the same scale as $O(M \log M)$ in the KNN method. Notice that we have computational complexity of $O(M^2 \log M)$ in a preprocessing stage of building the weighted KNN graphs, if we perform it in a direct manner (i.e., for each training sample, we calculate and order the mutual distances to all the other training data to search its nearest neighbors). Fortunately, the existing techniques proposed in literature to improve the efficiency of KNN based methods can be easily integrated into our ENN method to speed up the searching of nearest neighbors, and hence reduce the complexity of the ENN method. For example, using the structure of k-d trees [24] can reduce the computational complexity of the preprocessing stage and the testing stage to $O(M \log M)$ and $O(\log M)$, respectively.

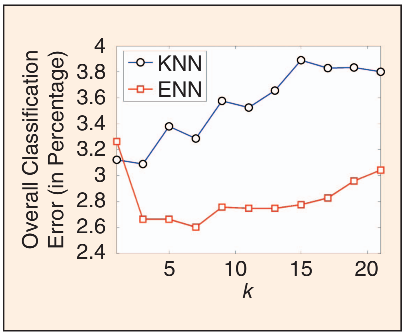


FIGURE 4 Overall classification error rate (in percentage) of ENN and KNN for handwritten digits classification.

IV. Experiments and Results Analysis

To evaluate the performance of our ENN classifier, we conduct several experiments with Gaussian data classification, hand-written digits classification [34], and twenty real-world datasets from UCI Machine Learning Repository [35]. In all these experiments, we explore every odd number of k from 1 to 21, i.e., $k = 1, 3, 5, 7, \ldots, 21$.

We first test our proposed ENN classifier for a 3-dimensional Gaussian data with 3 classes, in comparison with the classic KNN rule and the Maximum a Posterior (MAP) rule. In the MAP rule, we estimate the parameters of Gaussian distribution for each class with the given training data. In our current experiments, we use 250 training samples and 250 test samples for each class with the following parameters:

$$\boldsymbol{\mu}_1 = [8\ 5\ 5]^T \quad \boldsymbol{C}_1 = \sigma_1^2\mathbf{I}$$
$$\boldsymbol{\mu}_2 = [5\ 8\ 5]^T \quad \boldsymbol{C}_2 = \sigma_2^2\mathbf{I}$$
$$\boldsymbol{\mu}_3 = [5\ 5\ 8]^T \quad \boldsymbol{C}_3 = \sigma_3^2\mathbf{I}$$

To demonstrate the effectiveness of the proposed ENN classifier to solve the deficiency of the KNN rule, we examine the classification error rate of each class with different variances. Fig. 3 shows the overall classification error rates (in percentage) averaged over 100 random runs. The detailed classification performance for each class among these three comparative methods, i.e., ENN, KNN, and MAP, is provided in Supplementary Material Section 2. Our results show that the proposed ENN method performs much better than the classic KNN method.

TABLE 1 **The average testing error rate and standard derivation of ENN $(k=3)$ compared with KNN $(k=3)$, naive Bayes, LDA, and neural network. All results are shown in percentage. Each result is averaged over 100 random runs: in each run, we randomly select half of the data as the training data and the remaining half as the test data. For each dataset, we highlight the best result with Bold value among all these five methods. To specifically compare the results of ENN and KNN, we also underline the value if ENN performs significantly better than KNN under one-tailed *t*-test $(p=0.01)$.**

DATASETS	ENN	KNN	NAIVE BAYES	LDA	NEURAL NETWORK
IONOSPHERE	**17.35** ± 2.69	18.55 ± 2.94	19.83 ± 2.86	20.68 ± 3.00	18.48 ± 2.90
VOWEL	**8.50** ± 1.92	11.73 ± 1.80	43.90 ± 2.98	40.94 ± 1.97	45.17 ± 3.15
SONAR	**22.67** ± 3.97	24.49 ± 4.06	29.22 ± 4.16	33.75 ± 5.11	27.24 ± 4.37
WINE	4.49 ± 2.16	7.08 ± 2.20	5.07 ± 1.71	**2.58** ± 2.13	7.21 ± 2.88
BREAST-CANCER	**4.04** ± 0.87	4.44 ± 1.07	5.76 ± 1.04	5.88 ± 1.18	4.57 ± 1.17
HABERMAN	**31.32** ± 6.53	32.13 ± 5.79	36.35 ± 10.85	34.63 ± 9.92	37.40 ± 10.58
BREAST TISSUE	**36.71** ± 6.37	42.40 ± 6.19	44.02 ± 6.18	41.24 ± 6.60	67.62 ± 5.22
MOVEMENT LIBRAS	**26.33** ± 2.88	32.16 ± 2.97	45.41 ± 3.39	39.90 ± 3.31	40.87 ± 4.34
MAMMOGRAPHIC MASSES	21.16 ± 1.43	22.27 ± 1.55	**18.96** ± 1.57	19.17 ± 1.72	49.40 ± 0.29
SEGMENTATION	24.71 ± 3.07	27.85 ± 3.04	**12.64** ± 2.93	12.79 ± 2.88	23.06 ± 5.95
ILPD	40.04 ± 3.58	40.91 ± 3.68	**26.87** ± 2.69	29.64 ± 3.39	32.09 ± 3.53
PIMA INDIANS DIABETES	31.22 ± 2.15	33.08 ± 2.37	29.44 ± 2.19	28.29 ± 2.01	**25.38** ± 2.77
KNOWLEDGE	23.93 ± 4.69	27.11 ± 4.45	12.66 ± 2.45	**6.97** ± 2.53	14.42 ± 3.86
VERTEBRAL	**35.13** ± 4.83	37.64 ± 5.06	47.93 ± 3.41	36.88 ± 4.83	45.11 ± 3.12
BANK NOTE	**0.09** ± 0.18	0.12 ± 0.23	15.27 ± 1.25	2.60 ± 0.60	0.18 ± 0.37
MAGIC	**20.10** ± 0.33	20.42 ± 0.36	25.69 ± 0.61	23.30 ± 0.34	29.62 ± 0.38
PEN DIGITS	**0.74** ± 0.15	0.94 ± 0.17	15.38 ± 0.41	11.22 ± 0.52	11.65 ± 0.70
FAULTS	0.91 ± 0.52	1.65 ± 0.86	**0.00** ± 0.00	**0.00** ± 0.00	**0.00** ± 0.00
LETTER	**5.60** ± 0.25	7.44 ± 0.25	40.09 ± 0.47	29.80 ± 0.37	28.33 ± 0.52
SPAM	10.08 ± 0.59	11.52 ± 0.63	10.31 ± 0.78	**9.64** ± 0.61	15.32 ± 1.02

We also evaluate the classification performance of our proposed ENN classifier on the entire MNIST handwritten digits dataset [34], which is a widely used benchmark in the community. In this experiment, we use 60,000 images as training data and 10,000 images as test data. Fig. 4 shows the comparison of classification error rates (in percentage) between classic KNN rule and our ENN rule, where the minimum error of 2.61% is obtained at $k=7$ for our ENN method. The overall classification error rates using the ENN rule are notably less than those with the KNN rule (Fig. 4).

We further apply our ENN classifier to twenty real world datasets from UCI Machine Learning Repository [35]. Table I presents the classification error rates (in percentage) for these twenty UCI datasets in comparison to the classic KNN, naive Bayes, linear discriminant analysis (LDA), and neural network. It shows that ENN always performs better than KNN, and in 17 out of these 20 datasets, the performance improvement is significant (one-tailed t-test, $p=0.01$). In the neural network implementation, we use the classic multilayer perceptron (MLP) structure with 10 hidden neurons, and with 800 back-propagation iterations for training at the learning rate of 0.01. The results are averaged over 100 random runs, and in every run, we randomly select half of the data as the training data and the remaining half as the test data. Let's consider the Spam database as an example. This database includes 4601 e-mail messages, in which 2788 are legitimate messages and 1813 are spam messages. Each message is represented by 57 attributes, of which 48 are the frequency of a particular word (FW), 6 are based on the frequency of a particular character (FC), and 3 are continuous attributes that reflect the use of capital letters (SCL) in the e-mails. Fig. 5 shows detailed classification error rates (in percentage) with different parameters of k, which clearly demonstrates that ENN method can achieve consistently lower error rates than those of KNN rule. For a detailed performance comparison between ENN and KNN for all these twenty datasets, please refer to the Supplementary Material Section 3 for further information.

We would like to note that for all these experiments, there appears to be no significant difference between ENN and KNN when $k=1$. The reason for this might be that under such a small value of $k=1$, the classification performance will be determined by the single closest neighbor. Therefore, when $k=1$, both methods do not consider the data distribution anyway. That might explain why under $k=1$, both methods show a very close performance.

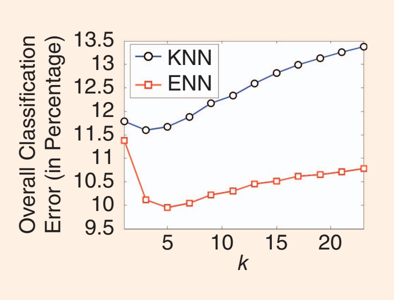


FIGURE 5 Overall classification error rate (in percentage) of ENN and KNN for spam e-mail classification.

V. Conclusions and Future Work

In this paper, we proposed an innovative ENN classification methodology based on the maximum gain of intra-class coherence. By analyzing the generalized class-wise statistics, ENN is able to learn from the global distribution to improve pattern recognition performance. Unlike the classic KNN rule which only considers the nearest neighbors of a test sample to make a classification decision, ENN method considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. We have developed three versions of the ENN classifier, **ENN, ENN.V1** and **ENN.V2**, and analyzed their foundations and relationships to each other. The experimental results on numerous benchmarks demonstrated the effectiveness of our ENN method.

As a new classification method, we are currently exploring the power of the ENN method and developing its variants for numerous machine learning and data mining problems, such as imbalanced learning, class conditional density estimation, clustering, regression, among others. For example, for imbalanced learning problems [30] [36] [37], notice that the distribution scale sensitive issue addressed by our ENN method can also be considered as an unequal distribution learning problem, therefore, we expect that our ENN method could be applied easily to the learning from imbalanced data. Meanwhile, the idea of the proposed ENN method may also benefit the class conditional density estimation [38] [39], if one considers a different size of neighborhood for each class according to our defined generalized class-wise statistic T_i. One similar idea is adaptive or variable-bandwidth kernel density estimation where the width of kernels is varied for different samples [11] [40]. Furthermore, similar to different variations of the KNN method, other forms of ENN classifiers could be developed, such as distance-weighted ENN. As nearest neighbor-based classification methods are used in many scientific applications because of their easy implementation, non-parametric nature, and competitive classification performance, we would expect the new ENN method and its future variations could have widespread use in many areas of data and information processing.

VI. Acknowledgment

This research was partially supported by National Science Foundation (NSF) under grant ECCS 1053717 and CCF 1439011, and the Army Research Office under grant W911NF-12-1-0378.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[2] Nature. (2008). Big data. [Online]. Available: http://www.nature.com/news/specials/bigdata/index.html

[3] Z. Zhou, N. Chawla, Y. Jin, and G. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives," *IEEE Comput. Intell. Mag.*, vol. 9, no. 4, pp. 62–74, 2014.

[4] Y. Zhai, Y. Ong, and I. Tsang, "The emerging big dimensionality," *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, 2014.

[5] E. Fix and J. L. Hodges Jr, "Discriminatory analysis-nonparametric discrimination: Consistency properties," U.S. Air Force Sch. Aviation Medicine, Randolph Field, Texas, Project 21-49-004, Tech. Rep. 4, 1951.

[6] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[7] P. Horton and K. Nakai, "Better prediction of protein cellular localization sites with the k nearest neighbors classifier," in *Proc. 5th Int. Conf. Intelligent Systems Molecular Biology*, 1997, vol. 5, pp. 147–152.

[8] R. Parry, W. Jones, T. Stokes, J. Phan, R. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. Wang, "k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction," *Pharmacogenomics J.*, vol. 10, no. 4, pp. 292–312, 2010.

[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics Probability*, California, 1967, vol. 1, pp. 281–297.

[10] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Stat.*, vol. 46, no. 3, pp. 175–185, 1992.

[11] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *Ann. Stat.*, vol. 20, no. 3, pp. 1236–1265, 1992.

[12] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *ACM SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[13] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inform. Syst.*, vol. 14, no. 1, pp. 1–37, 2007.

[14] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. Artificial Intelligence*, vol. 14, pp. 1137–1145, 1995.

[15] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Advances Neural Information Processing Systems*, 2005, pp. 513–520.

[16] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Advances Neural Information Processing Systems*, 2005, pp. 1473–1480.

[17] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Machine Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[18] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Computer Society Conf. Vision Pattern Recognition*, 2005, pp. 539–546.

[19] J. Wang, P. Neskovic, and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognit. Lett.*, vol. 28, no. 2, pp. 207–213, 2007.

[20] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *Comput. Res. Repository*, 2013, to be published.

[21] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 6, pp. 607–616, 1996.

[22] K. C. Gowda and G. Krishna, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood," *IEEE Trans. Inform. Theory*, vol. 25, no. 4, pp. 488–490, 1979.

[23] S. C. Bagui, S. Bagui, K. Pal, and N. R. Pal, "Breast cancer detection using rank nearest neighbor classification rules," *Pattern Recognit.*, vol. 36, no. 1, pp. 25–34, 2003.

[24] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.

[25] S. Z. Li, K. L. Chan, and C. Wang, "Performance evaluation of the nearest feature line method in image classification and retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 11, pp. 1335–1349, 2000.

[26] J. McNames, "A fast nearest-neighbor algorithm based on a principal axis search tree," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 9, pp. 964–976, 2001.

[27] T. Liu, A. W. Moore, and A. Gray, "New algorithms for efficient high-dimensional nonparametric classification," *J. Machine Learn. Res.*, vol. 7, pp. 1135–1158, 2006.

[28] V. Garcia, E. Debreuve, and M. Barlaud, "Fast k nearest neighbor search using GPU," in *Proc. IEEE Computer Society Conf. Computer Vision Pattern Recognition Workshops*, 2008, pp. 1–6.

[29] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. 30th Annu. ACM Symp. Theory Computing*, 1998, pp. 604–613.

[30] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Know. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.

[31] J. H. Friedman, S. Steppel, and J. Tukey, *A Nonparametric Procedure for Comparing Multivariate Point Sets*. no. 153, Stanford Linear Accelerator Center Computation Research Group Technical Memo, 1973.

[32] M. F. Schilling, "Multivariate two-sample tests based on nearest neighbors," *J. Amer. Stat. Assoc.*, vol. 81, no. 395, pp. 799–806, 1986.

[33] M. Schilling, "Mutual and shared neighbor probabilities: Finite-and infinite-dimensional results," *Adv. Appl. Probab.*, vol. 18, no. 2, pp. 388–405, 1986.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[35] M. Lichman. (2013). UCI Machine Learning Repository. School of Information and Computer Science. Irvine, CA: Univ. California. [Online]. Available: http://archive.ics.uci.edu/ml/.

[36] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Networks*, 2008, pp. 1322–1328.

[37] S. Chen, H. He, and E. A. Garcia, "RAMOBoost: Ranked minority over-sampling in boosting," *IEEE Trans. Neural Networks*, vol. 21, no. 10, pp. 1624–1642, 2010.

[38] G. Biau, F. Chazal, D. Cohen-Steiner, L. Devroye, and C. Rodriguez, "A weighted k-nearest neighbor density estimate for geometric inference," *Electron. J. Stat.*, vol. 5, pp. 204–237, 2011.

[39] U. von Luxburg and M. Alamgir, "Density estimation from unweighted k-nearest neighbor graphs: A roadmap," in *Proc. Advances Neural Information Processing Systems*, 2013, pp. 225–233.

[40] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Hoboken, NJ: Wiley, 2009.