Neighbors-based graph construction for dimensionality reduction

HUI TIAN\(^1\), LONG LAN\(^2\), AND XIANG ZHANG\(^3\), AND ZHIGANG LUO\(^4\)

\(^1\)\(^2\)\(^3\)\(^4\) National University of Defense Technology

Corresponding author: Long Lan (e-mail: tianhui95@sina.com)

ABSTRACT Dimensionality reduction is a fundamental task in the field of data mining and machine learning. In many scenes, examples in high-dimensional space usually lie on low-dimensional manifolds; thus, learning the low-dimensional embedding is important. Some well-known methods, such as LPP and LE, adopt a locality-preserving strategy by constructing an adjacent graph and using the graph Laplacian to project raw examples into subspace in order to obtain the low-dimensional representation. Accordingly, in this paper, we propose a novel neighbors-based distance that measures the distance of two examples through their neighbors. To be more specific, we create a virtual bridge point from the neighbors of each example and use it to link with others. Instead of computing their direct Euclidean distance, we derive the distance of any two examples using their bridge points. We note that the introduced metric shows a high discriminative ability for the examples on the boundary, which known to be infamously hard examples. Extensive experiments on classification and clustering demonstrate that our proposed graph construction method can achieve a large margin improvement in spite of its simple form.

INDEX TERMS Dimensionality Reduction; Graph, Metric; Manifold Learning

I. INTRODUCTION

The ability to learn low-dimensional representation embedded in high-dimensional space plays a very important role in the field of data mining and machine learning. In recent years, many classical methods have been proposed to address this issue. These methods can be roughly categorized into two groups; namely, the supervised and unsupervised dimensionality reduction methods. Principal components analysis (PCA) [17], as one of the most popular dimensionality reduction methods, maps the observed examples into the subspace where all examples share the maximal variance. Nonnegative matrix factorization (NMF) [19] is also a classical unsupervised method. The non-negative property of NMF provides the benefit of parts-based representation which is similar to how the human brain understands the world. The coefficients of all parts consist of the low-dimensional representations. Linear discriminant analysis (LDA) [12] is a supervised method, which guarantees that examples from different classes have large distance between them in the subspace. In addition to the supervised and unsupervised method, there is also semi-supervised dimensionality reduction (SSDR) [28], which utilizes part of the label information to search for low-dimensional subspace. In spite of the huge success achieved by the previous methods, how to capture the underlying structure from the observed data remains a challenging one. We note here that these methods mentioned above consider the observed examples from a global perspective. Many impressive works have found another path to the locality-preserving property. Locally linear embedding (LLE) [22] considers that each example can be linearly reconstructed using its nearby examples. The base examples and the corresponding coefficients for each example are enforced to be same in both high and low dimensional space. The local linearity is thus preserved in the low dimensional space.

We note that the adjacent graph plays a very important role in the manifold learning of dimensionality reduction. Many studies have addressed the significances of the adjacent graph [8], [11], [25], [30], and successfully applied diverse graphs to clustering and classification tasks [21], [24], [27], while more recent works drive the graph construction dimensionality reduction methods into more real-world and complex scenes. Khoreva et al. [18] constructed a spatio-temporal graph for video segmentation, while Zhu et al [29] proposed a robust graph construction for dimensionality reduction (RGDR): instead of directly constructing a graph matrix to preserve the local similarity of data, RGDR adaptively learns three matrices simultaneously, using the transformation matrix to project the original data into their
low-dimensional space. However, many works mainly address subspace searching linearly or nonlinearly based on the existing adjacent graph; the construction of the adjacent graph itself is less widely studied.

Euclidean distance is the most popular metric used to measure the similarity of pair-wise examples. However, this metric still ignores the distribution of the neighborhood examples for each example. Accordingly, in this paper, we design a novel metric to measure the distance of any two examples with the help of their neighbors. More specifically, we introduce a bridge point for each example to connect it with other examples. The final distance between the concerned example and other examples are calculated by the bridge point rather than the example itself. One encouraging observation in the learning of the embedded low-dimensional manifold is that the hard examples are pushed away from their locations in high-dimensional space with our new metric. Furthermore, many studies point out that hard examples, namely those the boundaries of different classes, are the most difficult part of any dataset of this kind. The incorrect nearby neighbors of boundary examples can easily pull these hard examples into the wrong classes. As our graph construction method decouples the k-nearest examples to the hard examples, the manifold learning from the adjacent graph achieves better performance in both classification and clustering tasks. Moreover, in the interests of fair comparison, we equip the classical LE [4], LPP [15] and Isomaps [2], [23] with our proposed graph to demonstrate the improvements.

The contributions of this paper can be summarized as follows: a) We propose a new distance metric that considers the neighbor information. This new metric can handle the hard examples better than the Euclidean distance; b) We provide a theoretical analysis of why our new method works, in Theorem 1; c) We present classification and clustering results using our proposed method on four public datasets, showing that our method achieves better performance than the baseline methods.

II. RELATED WORKS
In this section, we discuss the graph construction methods and several classical manifold learning methods which we equip with our defined graph to make a comparison include Laplacian eigenmaps, Locality preserving projections and Isomaps.

A. ADJACENT GRAPH CONSTRUCTION
A discriminative and informative graph is very important for many applications, in the graph, the weight of edge reflect the similarity between examples. The most used measurement of similarity is the Euclidean distance, generally, the larger the distance, the smaller the similarity. Cosine distance calculate the angle of involved examples to measure their distance. The nearest neighbors strategy is proved an effective skill to optimize the graph as the most nearby neighbors are always the most informative examples. K-nearest neighbors and ε-neighborhoods [18] are two solutions to obtain the informative neighbors. K-nearest neighbors find the nearest k neighbors for each examples no matter how far its neighbors away from it. ε-neighborhoods draws a ball for each example, the examples inside the ball are considered as its neighbors no matter whether there are how many examples in the ball. We note that in many real-world dataset, neither the simple Euclidean distance and Cosine distance nor the nearest neighbors strategy achieves a promising similarity, as they could not accurately compute the distance between different examples in high dimensional space.

B. MANIFOLD LEARNING METHODS FOR DIMENSIONALITY REDUCTION

1) Laplacian eigenmaps

Laplacian eigenmaps [3], [4] considers the problem of constructing a representation for data lying on a low dimensional manifold embedded in a high dimensional space. LE firstly build a graph to incorporate the neighborhood information of the concerned dataset. The core algorithm of LE is very simple. Specifically, LE constructs the adjacent graph G, each value in G denotes the similarity of the involved two examples and is calculated via the heat kernel or simple 0-1 weight based on the Euclidean distance. To find a low embedding in the such as m-dimensional Euclidean space. We only need to solve the m+1 eigenvectors of the following problem

\[ Lf = \lambda Df, \]

where D is diagonal weight matrix, its non-zero values are the sums of the corresponding rows or columns of G, \( D_{ii} = \sum_j G_{ji} \). \( I = D - G \) is the Laplacian matrix. Obviously, the Eq. 1 has the eigenvalue of 0, this eigenvalue its corresponding eigenvector have no meaning in the embedding. Thus, only the next m small eigenvalues (from small to large) and the corresponding eigenvectors are kept. The set of \((f_1, f_2, \ldots, f_m) \in \mathbb{R}^{n \times m}\) consist of the n examples in the m low dimensional space.

2) Locality preserving projections

Locality preserving projections (LPP) [15], [26] aim to find the intrinsic dimensionality of the dataset in the high dimensional space. LPP is a linear dimensionality reduction method, it however shares several useful nonlinear properties which address by only the nonlinear methods such as LE and LLE. The adjacent graph construction of LPP is similar to LE, also adopts the heat kernel or simple 0-1 strategy to represent the edge of graph according to Euclidean distance. LPP solve the following generalized eigenvector problem to obtain a transformation matrix

\[ XLX^T a = \lambda XDX^T a. \]

The L and D is the Laplacian matrix and diagonal matrix respectively as defined in Eq 1. X is the set of observed dataset. By solving the minimum eigenvalue and their corresponding eigenvectors of the generalized Eq 2, it is readily to achieve the transformation matrix \( A = (a_0, a_1, \ldots, a_{m-1}) \in \mathbb{R}^{n \times m} \).
R^{n \times m} (n > m), through which the examples in the high $n$-dimensional space could be projected into the low-$m$-dimensional subspace. Different from LE, LLE and Isomaps, LPP is defined everywhere of the dataset, that is to say, the transformation matrix learned from training examples can be directly applied to test examples. However, for the methods of LE, LLE and Isomaps, it is unclear how to evaluate the mapping for the new test examples.

3) Isomaps

Isomaps [2], [23] provides a global geometric framework for nonlinear dimensionality reduction. In the case of classical Swiss roll dataset, examples distribute in the different roll are far apart if measured by the geodesic or shortest path, however, if using the straight-line Euclidean distance, the paths are close. Thus, in this special case, only the geodesic distances reflect the true low-dimensional geometry of the manifold. Isomaps preserves the intrinsic geometry of the observed data, for the neighboring examples, the adopted geodesic metric approximates the Euclidean metric thus the neighborhoods of each examples are almost kept the similar, for the faraway examples the distance between them are measured by several middle examples, and the final distance is the sum up of all paths over these middle examples. The middle examples are searched through the famous Floyd algorithm to guarantee the paths connect the two end examples are shortest. Specially, Isomaps calculates the distance between two examples using two different strategies, for the nearest neighbors, Isomaps sets the value as their direct Euclidean distance. For the faraway examples, the distance value is replaced by summing up the Euclidean distance of examples stationed on the shortest path. To achieve a low dimensional embedding leverage the geodesic graph, Isomaps also formulates the mapping as a eigenfunction problem similar to LE.

### C. KNN AND K-MEANS

K-Nearest Neighbor [9], [14] is a typical supervised method for classification. It assigns the sample to the class most of its neighbors belong to. K-means is a typical method for clustering. After given the class number $k$, k-means [1], [13] separates the samples into $k$ class according to their distance. We therefore define a novel distance to measure the similarity of any two examples and construct a discriminative and informative graph to reduce the dimensionality of the original data for both classification and clustering. In this section, we first introduce the motivation of our designed metric for the adjacent graph, then provide a more detailed definition of the metric. We further analyze the property of the proposed distance in the final part of this section.

### III. METHOD

Based on the analysis from the above sections, we can conclude that the adjacent graph plays an important role in dimensionality reduction. Despite this, however, few previous works emphasize the significance of the graph itself. We therefore define a novel metric to measure the similarity of any two examples and construct a discriminative and informative graph to reduce the dimensionality of the original data for both classification and clustering. In this section, we first introduce the motivation of our designed metric for the adjacent graph, then provide a more detailed definition of the metric. We further analyze the property of the proposed distance in the final part of this section.

#### A. MOTIVATION AND DEFINITION

Euclidean distance is widely used to measure affinity, such as in the classical K-Nearest Neighbor (KNN) and LPP. However, the Euclidean distance can only measure the similarity between two directly involved samples while ignoring the distribution of their neighbors. As we all know, neighborhood distribution information is vital for classification and regression task; despite this, the question of how to integrate the neighborhood distribution information into the metric merits further study. Here, we design a novel distance, referred to as neighbors-based distance, which we expect to increase the inter-class distance and decrease the intra-class distance for hard examples by taking the neighborhood distribution information into consideration. First, let us con-
and symmetry of distance, we modify the final distance of
We notice that ρ(P1, Q1) = d(P2, Q2), d(x, y) represents the Euclidean distance between x and y. However, intuitively, by taking neighbor information into consideration, we can easily draw the conclusion that the distance between P1 and Q1 should be larger than that between P2 and Q2. Based on this observation, our designed metric aims to narrow the distance of P2 and Q2 while widen the distance of P1 and Q1, as P1 and Q1 are more likely to be from two different classes, while P2 and Q2 more likely to be from the same class when their neighborhood information is taken into account. Thus, in our defined metric ρ, ρ(P1, Q1) > ρ(P2, Q2). It is clear that Euclidean distance cannot work well in such a situation.

A more detailed definition of ρ is presented below. As shown in Fig. 1(b), to make a clear illustration, we consider the 2-nearest neighbors case. For every example, taking Q1 as an example, we define the distance of Q1 and P1 as

\[
\rho'(P_1, Q_1) = \left\| P_1 - Q_1 \right\|_2 + \left\| Q_1 - Q_2 \right\|_2. \tag{3}
\]

We notice that \( \rho'(P_1, Q_1) \neq \rho'(Q_1, P_1) \). To maintain the symmetry of distance, we modify the final distance of Q1 and P1 as follow:

\[
\rho(P_1, Q_1) = \frac{\rho'(P_1, Q_1) + \rho'(Q_1, P_1)}{2}, \tag{4}
\]

which is the distance \( \rho \) shown in Fig. 1 (b).

### B. ALGORITHM

To verify the effectiveness of the defined metric, we apply the Neighbors-based graph to several classical dimensionality reduction methods, such as Local Preserving Projections (LPP), Laplacian Eigenmap(LE) and Isomaps. The main flow is shown in Fig. 2. The detailed process to construct \( W \) is displayed in Fig. 3. The former three part of Fig. 3 is clear. The only problem is how to construct \( W \) according to \( D' \). We follow the routing of LPP and LE. First, find the k-nearest neighbor according to \( D' \). Second, we modify the matrix \( D' \), all the non-k-nearest neighbor points are set to zero. The modified \( D' \) is \( W \). After getting \( W \), the method to solve this problem is the same with LPP or LE. The time complexity of constructing \( W \) is \( O(n^2d) \), where \( n \) is sample number and \( d \) is the dimension of sample.

### C. ANALYSIS OF NEIGHBORS-BASED DISTANCE

In this subsection, we will show some promising properties of the defined metric. First of all, we prove \( \rho(P_1, Q_1) > \rho(P_2, Q_2) \) as mentioned in section 3.1 with the theorem 1. Then, we explain why the neighbors-based distance can handle the infamous hard examples. Finally, we will face a weakness of this distance.

1) Theorem 1
Given the definition of \( \rho \), let’s reconsider the situation of in Fig. 1. We human can get an intuitive idea of \( \rho(P_1, Q_1) > \rho(P_2, Q_2) \) from Fig. 1(b) at a first glance. Here, we give a proof of \( \rho(P_1, Q_1) > \rho(P_2, Q_2) \) when some reasonable assumptions are satisfied.

**Theorem 1:** Assumption: all the \( l_2 \)-norm of Neighborhood vector is the same, marked with \( d_1, d_1 = d(Q_1, Q_1) = ... \)
$d(Q_2', Q_2) = d(P_2', P_2) = d(P_1', P_1).$ Consider point $P_1$, $Q_1$ and $P_2$, $Q_2$ in Fig. 1. $P_1P_1', P_2P_2', Q_1Q_1', Q_2Q_2'$ are Neighborhood vector. $\angle Q_1'Q_1P_1 > \angle Q_2'Q_2P_2$ and $\angle P_1'P_1Q_1 > \angle P_2'P_2Q_2$. Prove that $\rho(P_1, Q_1) > \rho(P_2, Q_2)$. 

$$\rho(P_1, Q_1) = \frac{d_1 + d(Q_1', P_1) + d_1 + d(P_1', Q_1)}{2},$$ (5)  

$$d(P_1, Q_1') = \sqrt{d_1^2 + d_1^2 - 2d_1 d(P_1, Q_1) \cos \alpha},$$ (6)  

$$d(Q_1, Q_1') = \sqrt{d_1^2 + d_1^2 - 2d_1 d(P_1, Q_1) \cos \beta}.$$(7)  

Therefore, $d(P_1, Q_1')$ is monotonically increasing with $\alpha$, $d(Q_1, Q_1')$ is monotonically increasing with $\beta$. $\rho(P_1, Q_1)$ is monotonically increasing with $\alpha$ and $\beta$. Where $\alpha$ is the included angle of $P_1Q_1$ and $Q_1A\bar{A}$'s Neighborhood vector $Q_1Q_1'$, $\beta$ is the included angle of $Q_1P_1$ and $P_1\bar{A}A$'s Neighborhood vector $P_1P_1'$ as shown in Fig. 1.  

$\angle Q_1'Q_1P_1 > \angle Q_2'Q_2P_2$, (8)  

$\angle Q_1P_1P_1' > \angle Q_2P_2P_2'$, (9)  

$\angle Q_1P_1Q_1' > \angle Q_2P_2Q_2'$, (10)  

$\rho(P_1, Q_1) = \frac{d_1 + d(Q_1', P_1) + d_1 + d(P_1', Q_1)}{2},$ (12)  

$\rho(P_2, Q_2) = \frac{d_1 + d(Q_2', P_2) + d_1 + d(P_2', Q_2)}{2},$ (13)  

$\rho(P_1, Q_1) > \rho(P_2, Q_2).$ (14)  

2) The superiority of the method  

A correct distance is extremely useful for the task of classification or clustering. Here we illustrate another property of our designed neighbors-based distance, which shows an encouraging discriminative ability for the application of classification and clustering. The promising merit of the proposed metric is that it can, to some extent, increase the inter-class distance and decrease the intra-class distance for hard examples. Hard examples lie near the boundary of different classes, thus it is difficult to distinguish them. Using the examples in the Fig. 1(b) for example. $P_2$ and $Q_2$ are two points from the same class, but their distance is far. $P_1$ and $Q_1$ are two points from different classes, but their distance is close. Examples like $P_1Q_1$ and $P_2Q_2$ are two types of hard examples. According to the Theorem 1, $\rho(P_1, Q_1) > \rho(P_2, Q_2) > d_1 = d_2$. Therefore, after using the Neighbors-based Distance, $\rho(P_1, Q_1)$ increased more than $\rho(P_2, Q_2)$. Because we only care about their relative distance, we can conclude that $\rho(P_1, Q_1)$ is increased and $\rho(P_2, Q_2)$ is decreased.  

3) Weakness of the proposed distance  

Let’s consider another situation shown in Fig. 5 (a), all the points are in the same class. However, $\overrightarrow{v_1} \cdot P_1P_2' < 0$ and $\overrightarrow{v_2} \cdot P_2P_2' < 0$. Therefore, the distance $\rho(P_1, Q_1)$ will be increased according to Theorem 1. The problem is that the Neighbors-based Distance of two points in the same class might be increased. However, if we look into the problem more carefully, the situation in Fig. 5 (a) might only occur between points in the inner part of the points group. In other word, the two points are not on the edge of the points group and they are easy to classify. In Fig. 5, (b), $\overrightarrow{v_3} \cdot P_3Q_1 > 0$ and $\overrightarrow{v_4} \cdot Q_1P_3 > 0$. Therefore, the distance $\rho(P_3, Q_1)$ will be decreased. The problem happens as the neighbors-based distance in the different class might be decreased. However, the situation in Fig. 5 (b) will only occur when points are far from each other like in Fig 5(b). Therefore, they are not hard to classify as well. In a word, the weakness of the proposed metric is easily handled.  

In summary, the impact of the weakness of the proposed method is less than that of its strengths.
D. MODIFICATION OF THE NEIGHBORHOOD VECTOR

Let’s consider the situation in Fig. 6. If point $P_i$ has three neighbors $P_2$, $P_3$ and $P_4$, the original Neighborhood vector is

$$\vec{v} = P_1P_2 + P_1P_3 + P_1P_4,$$  \hspace{1cm} (15)

The closest neighbor should be the most important. To treat different neighbor differently, we apply the weight to the construction of the Neighborhood vector. The Neighborhood vector is

$$\vec{v} = w_1P_1P_2 + w_2P_1P_3 + w_3P_1P_4,$$  \hspace{1cm} (16)

where $w_i = k\sum_{i=1}^{k}e^{-\|P_iP_{i+1}\|/\sigma_i^2}$, $i = 1, 2, \ldots, k$, $k = 3$. The results in section 4 are based on the modified definition.

IV. EXPERIMENTS

In this section, we provide extensive experimental results on five datasets. We first introduce the selected dataset in Section 4.1. Second, we present the experimental settings in detail setting in Section 4.2; also, we will present the results on the above five datasets and compare them with other methods. Finally, we will provide robustness analysis of four important parameters in Section 4.3.

A. DATASET

The scope of the datasets encompasses face recognition (YaleB and ORL [5]–[7], [16]), image segmentation (UCIs Image Segmentation [10]), wine classification (UCIs Wine Quality [20]), and glass classification (UCIs Glass Identification [10]). The UCI Machine Learning Repository provides a large number of datasets on which users can verify their algorithm. A brief introduction to the chosen datasets is presented in Table 1. To facilitate a comprehensive evaluation, the chosen datasets exhibit significant variance; for example, the number of examples in each dataset varies from 214 to 60000, the number of attributes varies from 9 to 1024, and the class number varies from 3 to 40. Therefore, these datasets are diverse enough to enable comprehensive validation of our proposed method.

### TABLE 1. brief introduction to dataset.

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>Task description</th>
<th>Instance Number</th>
<th>Attribute Number</th>
<th>Class Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>YaleB</td>
<td>Face recognition</td>
<td>2432</td>
<td>1024</td>
<td>38</td>
</tr>
<tr>
<td>ORL</td>
<td>Face recognition</td>
<td>400</td>
<td>1024</td>
<td>40</td>
</tr>
<tr>
<td>UCI Image Segmentation</td>
<td>Image segmentation</td>
<td>2310</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>UCI wine quality</td>
<td>Wine classification</td>
<td>4898</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>UCI glass identification</td>
<td>Glass classification</td>
<td>214</td>
<td>9</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>The number of instances to construct Neighborhood vector</td>
</tr>
<tr>
<td>K2</td>
<td>The K in KNN classifier</td>
</tr>
<tr>
<td>RDIM</td>
<td>The reduction dimension</td>
</tr>
<tr>
<td>t</td>
<td>HeatKernel parameter</td>
</tr>
</tbody>
</table>

### TABLE 2. 4 main parameters in the experiments.

<table>
<thead>
<tr>
<th>name</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>The number of instances to construct Neighborhood vector</td>
</tr>
<tr>
<td>K2</td>
<td>The K in KNN classifier</td>
</tr>
<tr>
<td>RDIM</td>
<td>The reduction dimension</td>
</tr>
<tr>
<td>t</td>
<td>HeatKernel parameter</td>
</tr>
</tbody>
</table>

B. EXPERIMENTAL SETUPS AND RESULTS

One direct method of validating the effectiveness of the proposed metric would be to equip the neighbor distance-based adjacent graph to the classical dimensionality reduction methods in order to determine whether or not the new metric-derived graph works.

Firstly, train/Test split is an important setting. As noted above, there is no feasible way to transfer the mapping from the training data to the test data for LE and Isomap; thus, we do not split up the training and test data for LE and Isomap. As we can learn a project matrix from the training data with LPP, we follow Deng Cai’s train/test [7] split for LPP. Furthermore, LPP is suitable for both classification and clustering, while Isomap and LE can only be applied to clustering.

Based on this observation, in order to conduct comprehensive analysis and comparison for our proposed metric, we conduct classification and clustering experiments on all five of these datasets, recording the LPP’s results for both experiments, while recording LE and LLE’s performance for the clustering experiment only. Moreover, the simplify matters, for the classification experiments, we utilize the popular K-NN classifier to separate the mapping outputs of these dimensionality reduction methods, while for the clustering experiments, we use K-means to cluster the low-dimensional outputs.

In these experiments, there are four important parameters that need to be tuned; these parameters are listed in table 2. We search for the most suitable result via grid search methods. The final results are presented in Tables 3 and 4. However, the best parameter can not guarantee that the neighbor distance works well. Thus, we will also present a sensitivity analysis of these parameters below.

In the classification experiments, we follow the settings of Deng Cai [5]–[7], [16]. More specifically, in the YaleB dataset, every face has 64 images from different views; we choose 50 images for training and 16 images for testing. In the ORL dataset, every face has 10 images, and we choose eight of these for training and two images for test. In the remain two UCIs dataset, we choose 5/6 instances for training and 1/6 instances for testing.

We firstly list the classification performances of compared methods on all five datasets. In Table 3, five dimensionality reduction methods are considered. Among them, LDA and...
### TABLE 3. KNN accuracy on 5 datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YALEB</th>
<th>ORL</th>
<th>UCI Image Segmentation</th>
<th>UCI wine quality</th>
<th>UCI glass identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA(supervised)</td>
<td>90.81</td>
<td>85.25</td>
<td>82.86</td>
<td>100</td>
<td>57.57</td>
</tr>
<tr>
<td>PCA</td>
<td>55.74</td>
<td>75.96</td>
<td>82.86</td>
<td>82.14</td>
<td>63.63</td>
</tr>
<tr>
<td>Supervised LPP(supervised)</td>
<td>89.54</td>
<td>75.96</td>
<td>82.86</td>
<td>85.71</td>
<td>69.69</td>
</tr>
<tr>
<td>LPP(baseline)</td>
<td>76.78</td>
<td>75.96</td>
<td>82.86</td>
<td>71.43</td>
<td>66.66</td>
</tr>
<tr>
<td>NLPP(our methods)</td>
<td>87.42</td>
<td>79.38</td>
<td>88.57</td>
<td>96.42</td>
<td>72.72</td>
</tr>
</tbody>
</table>

### TABLE 4. K-Means accuracy on 5 datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YALEB</th>
<th>ORL</th>
<th>UCI Image Segmentation</th>
<th>UCI wine quality</th>
<th>UCI glass identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>9.61</td>
<td>63.76</td>
<td>66.35</td>
<td>63.76</td>
<td>44.28</td>
</tr>
<tr>
<td>LPP(baseline)</td>
<td>14.13</td>
<td>65.28</td>
<td>66.35</td>
<td>65.28</td>
<td>45.00</td>
</tr>
<tr>
<td>LE(baseline)</td>
<td>21.75</td>
<td>54.33</td>
<td>60.00</td>
<td>54.33</td>
<td>45.72</td>
</tr>
<tr>
<td>Isomaps(baseline)</td>
<td>20.63</td>
<td>63.88</td>
<td>51.25</td>
<td>63.88</td>
<td>42.25</td>
</tr>
<tr>
<td>NLPP(our methods)</td>
<td>15.28</td>
<td>66.29</td>
<td>68.20</td>
<td>66.29</td>
<td>45.72</td>
</tr>
<tr>
<td>NLE(our methods)</td>
<td>24.03</td>
<td>56.01</td>
<td>52.33</td>
<td>56.01</td>
<td>51.25</td>
</tr>
<tr>
<td>NIsomaps(our methods)</td>
<td>27.24</td>
<td>67.25</td>
<td>53.25</td>
<td>67.25</td>
<td>45.72</td>
</tr>
</tbody>
</table>

**FIGURE 7.** Sensitivity analysis of K1 and K2. We can observe that the curves of the proposed NLPP and LPP are relatively stable on both of these datasets, while NLPP achieves better performances than LPP.

PCA are two baselines; LDA is the supervised method, while PCA is unsupervised. We can see that NLPP outperforms LPP in all datasets. NLPP is even comparable with the supervised method. The reason for this result is that neighbor distance can enlarge the distance of hard negative pairs and decrease the distance of hard positive pairs.

We list the clustering performances of methods in Table 4. Here, we use PCA as a baseline. Three classical dimensionality reduction methods and their variants are included in the comparison. LPP and its variant NLPP share the same parameter the only different is the metric used. LE and NLE, Isomaps and NIsomaps also differ only in terms of the metric used to facilitate fair comparison. We use the clustering accuracy (ACC) to evaluate the performances. Neighbor-based distance show better performance for all three dimensionality reduction methods.

**C. PARAMETER SENSITIVITY ANALYSIS**

In this section, we take a look at how the parameters affect performance and demonstrate the robustness of our proposed metric to these parameters. To do so, we provide parameter sensitivity analysis for K1, K2, RDIM and t. Due to space limitations, we here only provide the sensitivity analysis for classification experiments on two representative datasets, YaleB and UCI Wine Quality. The methods used for these tasks are LPP and NLPP.
The performances are relatively stable to K1 and t and a little sensitive to K2 and RDIM. A important observation is that whether performances are sensitive, the NLPP achieves better performance than LPP. Therefore, we can conclude that the proposed neighbor-based distance has an encouraging positive impact on stable dimensionality reduction.

V. CONCLUSION
Dimensionality reduction is a fundamental task in the fields of machine learning and data mining. Our proposed neighbor-distance can enlarge the distance between hard negative pairs and decrease the distance between hard positive pairs. Thus it can perform well on many tasks. Extensive experiments on a number of datasets demonstrate that the proposed metric can be applied to classical dimensionality reduction methods, whether linear or nonlinear, and achieve substantial performance. However, due to time limitations, we did not conduct experiment on very large-scale datasets such as Imagenet or Amazon Reviewers. When handling such large scale real-world problem, deep learning based methods are often superior to others. Accordingly, combining neighbor distance with deep methods represents an avenue for future work. Another potential avenue would be to replace the Euclidean distance in contrastive loss or triplet loss with our neighbor distance.

REFERENCES

HUI TIAN received the B.S. degree from National University of Defense Technology, Changsha, China. He is currently pursuing the master’s degree with the National University of Defense Technology, Changsha, China. His current research interests include computer vision and machine learning.

LONG LAN received the Ph.D. degree in computer science from the National University of Defense Technology, in 2017. He was a visiting Ph.D. student with the University of Technology Sydney, from 2015 to 2017. He is currently a Lecturer with the College of Computer, National University of Defense Technology. His research interests include multi-object tracking, computer vision, and discrete optimization.

XIANG ZHANG received the M.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2010 and 2015, respectively. He is currently a Research Assistant with the Institute for Quantum Information and State Key Laboratory of High Performance Computing, College of Computer, National University of Defense Technology. His current research interests include computer vision and machine learning.

ZHIGANG LUO received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1981, 1993, and 2000, respectively. He is currently a Professor with the College of Computer, National University of Defense Technology. His current research interests include machine learning, computer vision, and bioinformatics.

HUI TIAN

LONG LAN

XIAO ZHANG

ZHIGANG LUO

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/.