

Article

# Laplacian Eigenmaps Dimensionality Reduction Based on Clustering-Adjusted Similarity

Honghu Zhou <sup>1,†</sup> and Jun Wang <sup>1,\*</sup>

<sup>1</sup> College of Computer and Information Science, Southwest University, Chongqing 400715, China; zhh143919@e-mail.swu.edu.cn

\* Correspondence: kingjun@swu.edu.cn, Tel: +86-23-6825-4396

† Current address: College of Computer and Information Science, Southwest University, No.2 Tiansheng Road, Beibei District, Chongqing 400715, China

Received: 25 July 2019; Accepted: 30 September 2019; Published: 4 October 2019

**Abstract:** Euclidean distance between instances is widely used to capture the manifold structure of data and for graph-based dimensionality reduction. However, in some circumstances, the basic Euclidean distance cannot accurately capture the similarity between instances; some instances from different classes but close to the decision boundary may be close to each other, which may mislead the graph-based dimensionality reduction and compromise the performance. To mitigate this issue, in this paper, we proposed an approach called Laplacian Eigenmaps based on Clustering-Adjusted Similarity (LE-CAS). LE-CAS first performs clustering on all instances to explore the global structure and discrimination of instances, and quantifies the similarity between cluster centers. Then, it adjusts the similarity between pairwise instances by multiplying the similarity between centers of clusters, which these two instances respectively belong to. In this way, if two instances are from different clusters, the similarity between them is reduced; otherwise, it is unchanged. Finally, LE-CAS performs graph-based dimensionality reduction (via Laplacian Eigenmaps) based on the adjusted similarity. We conducted comprehensive empirical studies on UCI datasets and show that LE-CAS not only has a better performance than other relevant comparing methods, but also is more robust to input parameters.

**Keywords:** Laplacian Eigenmaps; dimensionality reduction; clustering-adjusted similarity

---

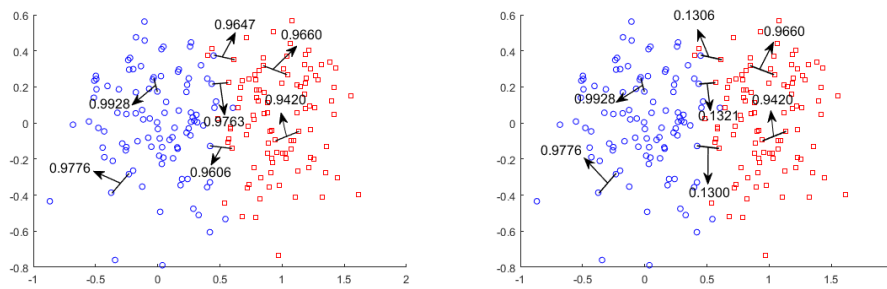
## 1. Introduction

Dimensionality reduction is a typical data preprocessing step in data mining and pattern recognition [1–3]. It aims to project original high-dimensional data into a low-dimensional subspace while preserving the geometric structure of them as much as possible. These low-dimensional representations of the original data can be used for different follow-up tasks, such as visualization, clustering, classification and so on. Dimensionality reduction has been studied for several decades [4–6]. By exploring and exploiting the geometric structure of samples from different perspectives, various unsupervised dimensionality reduction methods have been proposed [7]. Principal Component Analysis (PCA) is a representative unsupervised method [8], it seeks to maximize the internal information of data after dimension reduction, and measures the importance of the direction by measuring the variance of the data in the direction of projection. However, such projection does not play a big role in data differentiation, and may make data points undistinguishable by mixing them together. In this case, Locally linear embedding (LLE) [9] and its variants [10–14] were proposed to seek the low-dimensional embedding of high-dimensional samples by preserving the local geometric structure of samples. Further on, to make well use of valuable labeled samples, some semi-supervised dimensionality reduction methods have also been introduced [15–18].

In this paper, we focus on the Graph Embedding-based Dimensionality Reduction (GEDR), which can unify most dimensionality reduction solutions [19]. GEDR highlights that the main difference between existing dimensionality reduction solutions is the adopted graph structure. GEDR methods typically rely on the adopted graph structures to capture the geometric relation between samples in the high-dimensional space. This kind of graph is usually called an affinity graph [20], since its edge set conveys information about the proximity of the data in the input space. Once the affinity graph is constructed, these methods derive the low-dimensional samples by imposing that certain graph properties are preserved in the reduced low-dimensional subspace. This typically results in an optimization problem, whose solution provides the reduced data, or a mechanism to project data from the original space to low-dimensional space. For example, Belkin et al. [21] introduced Laplacian Eigenmaps (LE), which constructs a neighborhood graph (normally are  $k$ -nearest-neighbor ( $k$ NN) or  $\varepsilon$ -nearest-neighbor ( $\varepsilon$ NN) graph [22]) to capture the local structure of samples. Tenenbaum et al. [23] proposed Isomap, which estimates the geodesic distance between samples and then uses multidimensional scaling to induce a low-dimensional representation. Weinberger et al. [24,25] introduced an approach called Maximum Variance Unfolding (MVU) to preserve both the local distances and angles between the samples. He et al. [13] extends LE for linear dimensionality reduction, which can output a linear projective matrix to project new samples into the low-dimensional subspace. In these studies, researchers have been recognizing that the constructed graph on the instances determines the performance of GEDR. However, how to construct a graph that correctly reflects the similarity between instances is a public problem [2]. The distance between instances becomes isometric as the dimensionality of instances increases [26] and many traditional similarity metrics are distorted by noisy or redundant features of high-dimensional data. Thus, researchers have developed several graph construction methods to improve the performance of GEDR.

Some efforts have been made to improve the performance of LE, a classical and representative GEDR method. Zeng et al. [27] proposed geodesic distance-based generalized Gaussian Laplacian Eigenmap (GGLE) method using different generalized Gaussian functions to measure the similarity between high-dimensional data points. Raducanu et al. [28] introduced Self-regulation of neighborhood parameter for Laplacian Eigenmaps (S-LE) by measuring similarity between instances via the ratio of geodesic distance and Euclidean distance between the samples and their neighborhood nodes, and the adopted neighborhood parameters are adjusted and optimized. Ning et al. [29] developed Supervised Cluster Preserving Laplacian Eigenmap (SCPLE), which constructs an intra-class graph and an inter-class graph, and determines the edge weights by class label information and adaptive thresholds. By maximizing the weighted neighbor distances between heterogeneous samples and minimizing the weighted neighbor distances between homogeneous samples, SCPLE maps homogeneous samples closer and heterogeneous samples faraway in the low-dimensional space.

Although these approaches improve the performance of LE, they still have problems when embedding instances close to the decision boundary. For example, as shown in Figure 1(a), instances from different clusters close to the decision boundary may be much closer than instances from the same cluster. As a result, they will be placed nearby in the reduced low-dimensional space, which misleads the data distribution and compromises the learning performance.



(a) Euclidean distance-based similarity between samples (b) Cluster-adjusted similarity between samples

**Figure 1.** Comparisons between two types of similarities between samples, shown as (a) and (b). The clustering-adjusted similarity clearly tunes down the Euclidean-based similarity between two instances from different clusters.

To remedy the issue illustrated in Figure 1, we proposed Laplacian Eigenmaps dimensionality reduction based on Clustering-Adjusted Similarity (LE-CAS). LE-CAS applies clustering technique on the original instances to explore the underlying data distribution and global structure of instances. At this stage, we initially intended to use  $k$ -means clustering [22] as the method to obtain the decision boundaries. However, as our feature space has large scale and complex structure, the clusters produced by  $k$ -means clustering depends largely on the distribution of samples and may not be related to the structure of the boundaries. To solve this kind of problem, we decided to employ an optimization method of  $k$ -means clustering called kernel  $k$ -means [30]. Kernel  $k$ -means maps the data to a higher-dimensional feature space through a nonlinear mapping and performs cluster analysis in the feature space. This method of mapping data to high-dimensional space can highlight the feature differences between sample classes and obtain more accurate clustering results. After that, LE-CAS uses the cluster structure to adjust the similarity between instances based on Gaussian heat kernel. In particular, if pairwise instances belong to different clusters, the similarity between them will be reduced based on the original, otherwise the similarity is unchanged. As shown in Figure 1(b), the similarities between pairwise samples from different clusters are reduced, while the similarities between samples from the same cluster remain high. In this way, the global structure information revealed by clustering is embedded into the adjusted similarity, which can better capture the structure between samples. Finally, LE-CAS executes Laplacian Eigenmaps dimensionality reduction based on this clustering-adjusted similarity. Extensive experimental results on UCI datasets from different domains show that LE-CAS significantly outperforms other approaches, which aim to improve LE by different techniques.

The structure of this paper is organized as follows. In Section 2, we give the details of how to adjust similarity between samples and list the procedures of LE-CAS. The preparatory works of the experiment are introduced in Section 3, experimental results and analysis on UCI datasets are presented in Section 4, followed with conclusions and future work in Section 5.

## 2. Methodology

Suppose  $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_n] \in \mathbb{R}^{n \times d}$  be  $n$  instances in the  $d$ -dimensional space. LE-CAS targets to project  $\mathbf{X}$  into a low-dimensional subspace with new representation  $\mathbf{Y} \in \mathbb{R}^{n \times c}$  with  $c \ll d$ . At the beginning of this section, we would like to introduce the basic idea inside our method called Clustering-Adjusted Similarity (CAS) [31]. In addition, then we will briefly illustrate the proceeding of our LE-CAS.

### 2.1. Clustering-Adjusted Similarity

The GEDR methods often resort to a typical similarity metric to capture the similarity between samples and structure among them. In this paper, we start with the widely used Gaussian heat kernel as follows:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_m^2}\right) \tag{1}$$

where  $\sigma_m > 0$  is Gaussian heat kernel width.  $W_{ij} \in (0, 1]$  represents the similarity between  $x_i$  and  $x_j$ , and  $W_{ii} = 1$ .  $W \in \mathbb{R}^{n \times n}$  can be viewed as a weighted adjacency matrix of a graph, which stores the pairwise similarity between  $n$  samples. This way of graph construction is proved to be a simple and effective solution by previous studies [13].

However, for pairs of instances located close to the decision boundary but from different classes, they have a high similarity (see two instances in different clusters but with high similarity 0.9674 in Figure 1(a)), which mislead them being close in the projected subspace. To mitigate the problem, we aim at reducing the similarity between instances close to decision boundary, which are from different clusters, while remaining the high similarity between samples of the same cluster. To reach that target, we apply CAS as described below.

CAS is based on the idea that samples in the same cluster are similar and those in different clusters should be dissimilar as much as much possible. To explore the clusters of the origin instances, we perform kernel  $k$ -means clustering on all instances. Suppose the set of the final clusters is  $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$ ,  $m$  is the number of clusters and  $\mathcal{C}_h$  represents the  $h$ -th cluster.  $\mathcal{C}_{h_1} \cap \mathcal{C}_{h_2} = \emptyset (\forall h_1 \neq h_2, h_1, h_2 = 1, 2, \dots, m)$ , and  $\cup_{h=1}^m \mathcal{C}_h = \mathbf{X}$ . The cluster centroids ( $\mathbf{u}_h \in \mathbb{R}^d$ ) of the  $h$ -th cluster is:

$$\mathbf{u}_h = \frac{\sum_{x_i \in \mathcal{C}_h} x_i}{|\mathcal{C}_h|} \tag{2}$$

$|\mathcal{C}_h|$  is the number of instances placed into the  $h$ -th cluster. Then the similarity between two cluster centroids is similarly defined based on the Gaussian heat kernel as follows:

$$S_{h_1 h_2} = \exp\left(-\frac{\|\mathbf{u}_{h_1} - \mathbf{u}_{h_2}\|^2}{\sigma_m^2}\right) \tag{3}$$

$S_{h_1 h_2} \in (0, 1)$  is the Gaussian heat kernel similarity between  $\mathbf{u}_{h_1}$  and  $\mathbf{u}_{h_2}$ . Obviously, the similarity between two cluster centroids is small when the distance in Euclidian measurement is large.

To facilitate the clustering-based adjustment, we define  $\mathbf{I}$  as follows:

$$\begin{aligned} \mathbf{I}_i &= h, \text{ if } x_i \in \mathcal{C}_h; i = 1, 2, 3, \dots, n \\ &h = 1, 2, 3, \dots, m \end{aligned} \tag{4}$$

$\mathbf{I} \in \mathbb{R}^{n \times d}$  is the label indicator matrix, each row corresponds to an instance and each column for a distinct label.  $d$  represents the number of clusters of the whole instances. If instance  $\mathbf{X}_i$  belong to the  $h$  th cluster,  $\mathbf{I}_i = h$ , otherwise,  $\mathbf{I}_i = 0$ .

Finally, the original Gaussian heat kernel similarity data matrix  $W$  is adjusted into  $\tilde{W}$  as follows:

$$\tilde{W}_{ij} = \mathbf{S}_{\mathbf{I}_i \mathbf{I}_j} W_{ij} \tag{5}$$

From the definition of  $\mathbf{I}$  and  $\mathbf{S}$ , we can find that for two samples ( $i$  and  $j$ ) placed into the same cluster  $\mathbf{S}_{ij} = 1$ , since  $\mathbf{I}_i = \mathbf{I}_j$ ,  $\tilde{W}_{ij} = W_{ij}$ . On the other hand, for two samples from different clusters,  $\tilde{W}_{ij}$  shrinks to  $\mathbf{S}_{\mathbf{I}_i \mathbf{I}_j} W_{ij}$ , since  $\mathbf{I}_i \neq \mathbf{I}_j$  and  $\mathbf{S}_{\mathbf{I}_i \mathbf{I}_j} < 1$ .

### 2.2. Laplacian Eigenmap-Based Clustering-Adjusted Similarity

Based on the clustering-adjusted similarity, we present the procedure of LE-CAS as follows, and the influence of our improved method is also shown in Figure 1.

1. Carry out kernel  $k$ -means clustering on the original dataset  $\mathbf{X}$ , the original data is clustered into  $k_1$  classes. During the clustering, we first use a nonlinear mapping function to map the instances from the original space  $\mathbb{R}^{n \times d}$  to a higher-dimensional space  $F$ , and then clustering in this space.

$$\phi : \mathbb{R}^{n \times d} \rightarrow F, x \rightarrow \phi(x) \tag{6}$$

The instances of the original space becomes

$$[\phi(x_1); \dots; \phi(x_n)] \tag{7}$$

On this basis, kernel clustering is to minimize the following criterion function,  $t_i^\phi$  is the class-mean of  $i$ -th cluster.

$$J^\phi = \sum_{i=1}^{k_1} \sum_{p \in \mathcal{C}_i} \|\phi(p) - t_i^\phi\|^2 \tag{8}$$

$$t_i^\phi = \frac{1}{|\mathcal{C}_i|} \sum_{p \in \mathcal{C}_i} \phi(p) \tag{9}$$

After clustering, we get the following groups:

$$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{k_1}\} \tag{10}$$

2. Construct a graph with the edge weight between  $\mathbf{x}_i$  and  $\mathbf{x}_j$  specified as Equation (1) ( $\mathbf{x}_i, \mathbf{x}_j$  are instances in  $\mathbf{X}$ ). Set up edges between each point and its nearest  $k_2$  points via  $k$ NN method,  $k_2$  is a preset value. This graph will continue to be used in the following steps.
3. To determine the weight between points, which is different from typical LE method, the adjusted weight matrix  $\tilde{\mathbf{W}}_{ij}$  calculated according to Equation (11) before is selected as the final weight matrix of the previous graph.  $\sigma_m > 0$  is Gaussian heat kernel width,  $\mathbf{u}_{h_1}, \mathbf{u}_{h_2}$  represents the cluster centroids of the cluster which  $\mathbf{x}_i$  and  $\mathbf{x}_j$  belongs to.  $\mathbf{l}_i, \mathbf{l}_j$  are indicative vectors.

$$\tilde{\mathbf{W}}_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_m^2}\right), & \text{if } \mathbf{l}_i = \mathbf{l}_j \\ \exp\left(-\frac{\|\mathbf{u}_{h_1} - \mathbf{u}_{h_2}\|^2}{\sigma_m^2}\right) \times \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_m^2}\right), & \text{if } \mathbf{l}_i \neq \mathbf{l}_j \end{cases} \tag{11}$$

4. Construct graph Laplace matrix  $\mathbf{L}$ ,  $\mathbf{D}$  is a diagonal matrix with its  $(i, i)$ -element equal to the sum of the  $i$  th row of  $\tilde{\mathbf{W}}_{ij}$

$$\mathbf{D}_{ii} = \sum_{j=1}^n \tilde{\mathbf{W}}_{ij} \tag{12}$$

$$\mathbf{L} = \mathbf{D} - \tilde{\mathbf{W}}_{ij} \tag{13}$$

5. The objective function of Laplace feature mapping optimization is as follows:

$$\min \sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \tilde{\mathbf{W}}_{ij}$$

$y_i$  and  $y_j$  are the representation of instance  $i, j$  in the target  $c$ -dimensional subspace  $\mathbf{Y}$ . The objective function after transform is as follows:

$$\min \text{trace} \left( \mathbf{Y}^T \mathbf{L} \mathbf{Y} \right), \text{ s.t. } \mathbf{Y}^T \mathbf{L} \mathbf{Y} = \mathbf{I} \quad (14)$$

Where the constraint function  $\text{s.t. } \mathbf{Y}^T \mathbf{L} \mathbf{Y} = \mathbf{I}$  guarantees the optimization problem has solutions.

- Do feature mapping, and calculate the eigenvectors and eigenvalues of  $\mathbf{L}$ . The column vectors of  $\mathbf{Y}$  that minimize the formula are the eigenvectors corresponding to  $c$  minimum non-zero eigenvalues (including multiple roots) of the generalized eigenvalue problem. The smallest  $c$  eigenvectors which are correspond to the non-zero eigenvalues are used as the output after dimensionality reduction.

$$\mathbf{L} \mathbf{y} = \lambda \mathbf{D} \mathbf{y} \quad (15)$$

### 3. Experiment Setup

#### 3.1. Datasets

We employ experiments on ten publicly available UCI datasets to quantitatively evaluate the performance of LE-CAS. The statistics of these datasets are listed in Table 1. These datasets are with different numbers of features and of samples: Msplice represents dataset on molecular biology splice online learning, which is used for multiclass clustering. W1a data set records information from a TV series called W1A. Soccer-sub1 stores the information on players registered in FIFA. Madelon is an artificial dataset, which was part of the NIPS 2003 feature selection challenge. This is a two-class classification problem with continuous input variables. FG-NET is a dataset used for face recognition. ORL contains 400 images of 40 different people, was created by the Olivetti research laboratory in Cambridge, England, between April 1992 and April 1994. Musk describes a set of 102 molecules of which 39 are judged by human experts to be musks and the remaining 63 molecules are judged to be non-musks. CNAE-9 is a data set containing 1080 documents of free text business descriptions of Brazilian companies categorized into a subset of 9 categories cataloged in a table called National Classification of Economic Activities. SECOM is about data from a semi-conductor manufacturing process. DrivFace contains images sequences of subjects while driving in real scenarios. All data sets are available from the UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/index.php>).

**Table 1.** Statistics of datasets used for experiments.

Datasets	Instance	Dimensions
Msplice	3175	240
W1a	2477	300
Soccer-sub1	1735	279
Madelon	2600	500
FG-NET	1002	262
ORL	400	1024
Musk	6598	223
CNAE-9	1080	857
SECOM	1567	591
DrivFace	1606	525

#### 3.2. Comparing Methods

We compare LE-CAS against four different GDR methods: the original LE [21], GGLE [27], S-LE [28] and SCPL [29]. The latter three comparing methods try to improve LE from different aspects and have close connection with our work. These comparing methods were introduced in section 1. We first apply these dimensionality reduction methods to project the high-dimensional samples into a

low-dimensional subspace. After that, the widely adopted  $k$ -means [22] clustering is applied to cluster the samples projected in the subspace by the respective comparing method.

In the experiments, we specify (or optimize) the input parameters of these comparing methods as the authors suggested in the original papers. As to LE-CAS,  $k_1$  is determined by calculating the contour coefficient to obtain a better result of initial clustering.  $k_2 = 10$  is used to construct  $k$ NN graph. The sensitivity of  $k_2$  for these  $k$ NN graph-based methods (LE[21], GGLE [27], and LE-CAS) and  $\sigma_m$  for Gaussian heat kernel function-based methods (GGLE [27], LE-CAS) will be studied later.

### 3.3. Evaluation Metrics

To evaluate the performance of different dimensionality reduction methods, we adopt four frequently used clustering-effect evaluation index: Fowlkes and Mallows Index (FMI) [32], F-measure [33] and Purity (PU) [34].

For dataset  $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_n]$ , it is assumed that the cluster obtained through clustering is divided into  $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$ , and the cluster given by the reference model is divided into  $\mathcal{C}^* = \{\mathcal{C}_1^*, \mathcal{C}_2^*, \dots, \mathcal{C}_m^*\}$ . Accordingly, let  $\lambda$  and  $\lambda^*$  respectively represent the cluster marker vectors corresponding to  $\mathcal{C}$  and  $\mathcal{C}^*$ . We consider the sample pairwise, as defined below:  $a = |\mathbf{SS}|$ ,  $\mathbf{SS} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \lambda_i = \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$ ,  $b = |\mathbf{SD}|$ ,  $\mathbf{SD} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \lambda_i = \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$ ,  $c = |\mathbf{DS}|$ ,  $\mathbf{DS} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* = \lambda_j^*, i < j\}$ ,  $d = |\mathbf{DD}|$ ,  $\mathbf{DD} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid \lambda_i \neq \lambda_j, \lambda_i^* \neq \lambda_j^*, i < j\}$ .

$|\mathbf{SS}|$  set contains sample pairs which belong to the same cluster in  $\mathcal{C}$  are still belong to the same cluster in  $\mathcal{C}^*$ ,  $|\mathbf{SD}|$  set contains sample pairs which belong to the same cluster in  $\mathcal{C}$  are not belong to the same cluster in  $\mathcal{C}^*$ .  $|\mathbf{DS}|$  set contains sample pairs which not belong to the same cluster in  $\mathcal{C}$  are belong to the same cluster in  $\mathcal{C}^*$ ,  $|\mathbf{DD}|$  set contains sample pairs which not belong to the same cluster in  $\mathcal{C}$  are still not belong to the same cluster in  $\mathcal{C}^*$ .  $a, b, c, d$  represents the number of data pairs in set  $|\mathbf{SS}|, |\mathbf{SD}|, |\mathbf{DS}|, |\mathbf{DD}|$ . Thus,

$$FMI = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}} \tag{16}$$

F-measure is based on precision and recall of clustering results:

$$F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

Purity represents the number of samples with correct clustering in the total number of samples.  $N$  represents total number of samples,  $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_m\}$  is the collection of cluster,  $\mathcal{C}_h$  represents  $h$ -th cluster,  $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$  is the collection of sample,  $\mathbf{X}_i$  represents  $i$ -th sample. Computational formula is shown as below:

$$PU(\mathcal{C}, \mathbf{X}) = \frac{1}{N} \sum_h \max_i |\mathcal{C}_h \cap \mathbf{X}_i| \tag{18}$$

## 4. Results

### 4.1. Results on Different Datasets

Figure 2 (for FMI's performance), Figure 3 (for F-measure's performance) and Figure 4 (for PU's performance) show the results with respect to different fixed target dimensionality after applying different dimensionality reduction solutions and with the same  $k$ -means clustering. To avoid random effects, we repeated the experiment for 30 times and calculated the mean and variance of the results of LE-CAS and comparing methods with respect to four evaluation metrics on respective datasets, which is shown on Table 2.

According to the table and figures, we can observe that the performances of all methods increases with the target dimensionality rising and LE-CAS always outperforms other comparing methods

across all the evaluation metrics and six datasets. LE-CAS and LE both uses Laplacian Eigenmaps, while LE-CAS achieves better performance on four evaluation indices than that of the LE. That is because LE-CAS changes the similarity calculation method of the instances on the decision boundary, and the adjacent instances are still close enough and the clusters they belong to remain the same after dimensionality reduction. The fact indicates the clustering similarity adjustment (CAS) can improve the performance of GEDR methods.

LE-CAS and GGLE both uses Gaussian kernel function (LE-CAS calculated the similarity between clusters and instances by Gaussian kernel function, GGLE measures similarity between instances via different generalized Gaussian kernel functions), while the performance of GGLE is much lower than that of LE-CAS. This fact shows that CAS used by LE-CAS can better reflect the real relationship between the instances than generalized Gaussian kernel functions used by GGLE.

The performance of LE-CAS is better than that of S-LE and SCPL, which use the ratio of Euclidean distance to geodesic distance between two instances adjusts the number of nearest neighbors. That is because the clustering-adjusted similarity can more well explore the geometric structure of instances than adaptive adjustment of neighborhood parameters. SCPL performs better than S-LE, which verifies that SCPL is an improved method based on S-LE via concerning neighborhood parameter of instances from same or different clusters.

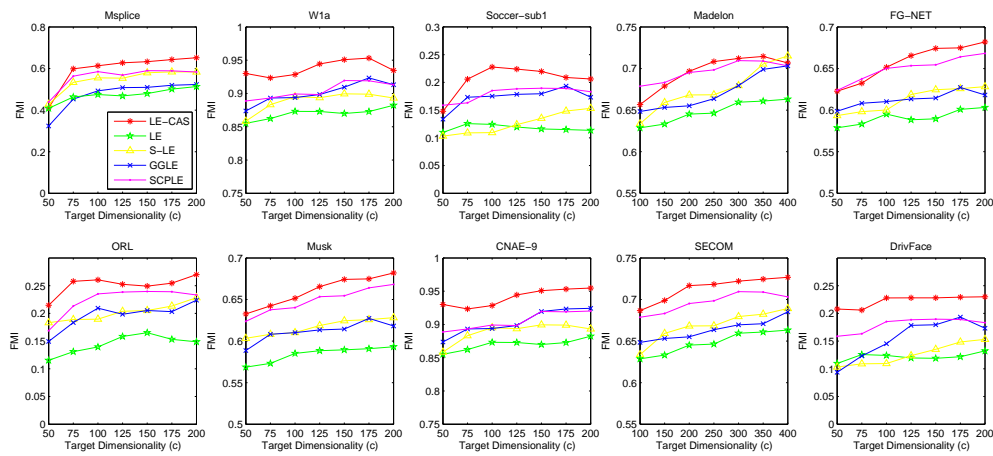
As for stability, LE-CAS performs well, the variance is basically stable and its effect of cluster can maintain good stability in different target dimensions, while the variance of other improved LE methods fluctuates and its effect of cluster has a complex turbulence into different target dimensions at the same time. That is because the similarity without CAS can be easily destroyed by noisy featured.

**Table 2.** Results (average $\pm$ std) of comparing methods on different datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace). The target dimensionality  $c$  is set to 10. The best (or comparable to the best) results are in **boldface**.

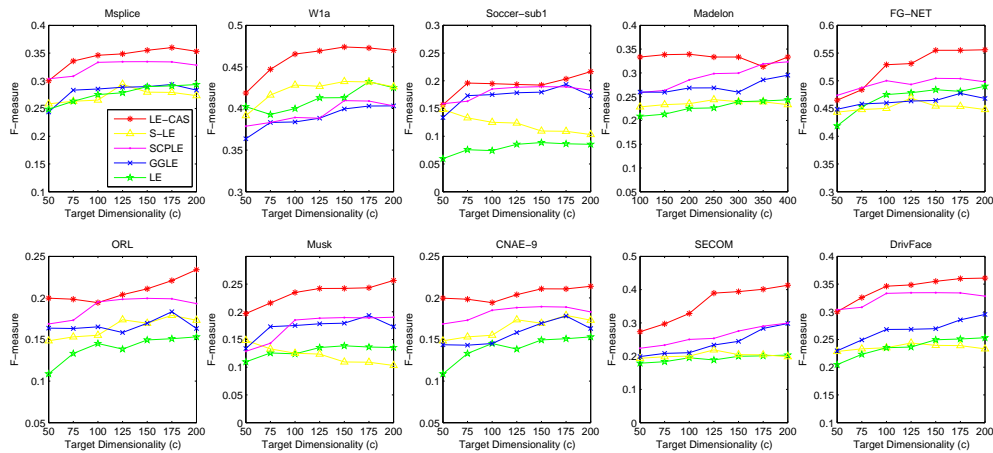
	FMI	F-measure	PU
Msplice			
LE-CAS	<b>0.6133<math>\pm</math>0.0124</b>	<b>0.5425<math>\pm</math>0.0302</b>	<b>0.7284<math>\pm</math>0.0638</b>
LE	0.3247 $\pm$ 0.0233	0.1871 $\pm$ 0.0292	0.4914 $\pm$ 0.0647
S-LE	0.5552 $\pm$ 0.0296	0.2956 $\pm$ 0.0239	0.5324 $\pm$ 0.0734
GGLE	0.4925 $\pm$ 0.0334	0.4025 $\pm$ 0.0108	0.6354 $\pm$ 0.0619
SCPL	0.5852 $\pm$ 0.0190	0.3928 $\pm$ 0.0099	0.6203 $\pm$ 0.0594
W1a			
LE-CAS	<b>0.9285<math>\pm</math>0.0514</b>	<b>0.4691<math>\pm</math>0.0375</b>	<b>0.9608<math>\pm</math>0.0461</b>
LE	0.6429 $\pm$ 0.0285	0.2488 $\pm$ 0.0215	0.6541 $\pm$ 0.0427
S-LE	0.7042 $\pm$ 0.0342	0.3158 $\pm$ 0.0261	0.7028 $\pm$ 0.0395
GGLE	0.8731 $\pm$ 0.0427	0.4021 $\pm$ 0.0125	0.7936 $\pm$ 0.0409
SCPL	0.8392 $\pm$ 0.0307	0.4210 $\pm$ 0.0325	0.8333 $\pm$ 0.0326
Soccer-sub1			
LE-CAS	<b>0.4268<math>\pm</math>0.0014</b>	<b>0.3172<math>\pm</math>0.0025</b>	<b>0.4774<math>\pm</math>0.0259</b>
LE	0.1242 $\pm$ 0.0062	0.1573 $\pm$ 0.0029	0.2358 $\pm$ 0.0291
S-LE	0.1995 $\pm$ 0.0051	0.2319 $\pm$ 0.0015	0.2936 $\pm$ 0.0245
GGLE	0.2452 $\pm$ 0.0024	0.2906 $\pm$ 0.0015	0.3534 $\pm$ 0.0208
SCPL	0.3052 $\pm$ 0.0099	<b>0.3173<math>\pm</math>0.0024</b>	0.3726 $\pm$ 0.0213
Medelon			
LE-CAS	<b>0.6967<math>\pm</math>0.0416</b>	<b>0.3024<math>\pm</math>0.0252</b>	<b>0.7804<math>\pm</math>0.1860</b>
LE	0.5852 $\pm$ 0.0437	0.1921 $\pm$ 0.0207	0.5046 $\pm$ 0.1643
S-LE	0.6082 $\pm$ 0.0367	0.2235 $\pm$ 0.0200	0.5478 $\pm$ 0.1302



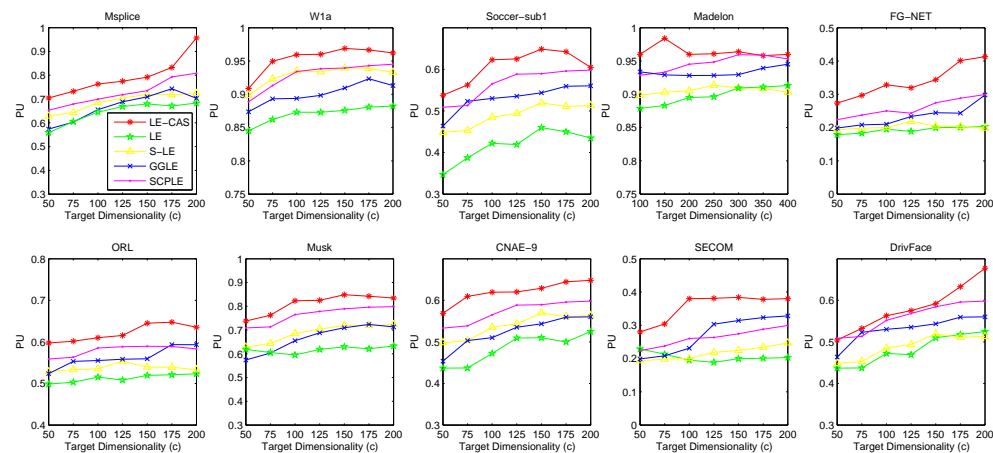
GGLE	0.6552±0.0345	<b>0.3023±0.0132</b>	0.6532±0.1033
SCPLE	<b>0.6923±0.0570</b>	0.3008±0.0219	0.6378±0.1126
FG-NET			
LE-CAS	<b>0.6515±0.0200</b>	<b>0.5042±0.0016</b>	<b>0.7520±0.0619</b>
LE	0.4952±0.0138	0.3895±0.0021	0.5923±0.0573
S-LE	0.4652±0.0207	0.4014±0.0015	0.6752±0.0592
GGLE	0.5558±0.0169	0.4977±0.0020	0.6936±0.0499
SCPLE	0.5874±0.0245	<b>0.5044±0.0021</b>	0.7233±0.0601
ORL			
LE-CAS	<b>0.5635±0.0452</b>	<b>0.3388±0.0092</b>	<b>0.6710±0.0229</b>
LE	0.1398±0.0226	0.1247±0.0084	0.4215±0.0198
S-LE	0.2635±0.0356	0.1964±0.0078	0.5024±0.0099
GGLE	0.4545±0.0279	0.2311±0.0065	0.5924±0.0187
SCPLE	0.4933±0.0281	0.2295±0.0081	0.6011±0.0125
Musk			
LE-CAS	<b>0.6832±0.0216</b>	<b>0.4724±0.0222</b>	<b>0.8804±0.1846</b>
LE	0.4956±0.0375	0.2121±0.0247	0.5376±0.1643
S-LE	0.5982±0.0299	0.3055±0.0200	0.5895±0.1302
GGLE	0.6458±0.0321	0.3623±0.0132	0.6989±0.1033
SCPLE	<b>0.6823±0.0370</b>	0.4078±0.0219	0.7277±0.1023
CNAE-9			
LE-CAS	<b>0.9285±0.0514</b>	<b>0.7051±0.0193</b>	<b>0.9409±0.1860</b>
LE	0.6541±0.0427	0.4621±0.0237	0.6746±0.1283
S-LE	0.7028±0.0365	0.5738±0.0224	0.5478±0.2302
GGLE	0.8731±0.0417	<b>0.7023±0.0122</b>	0.8032±0.1053
SCPLE	0.8392±0.0347	0.6708±0.0238	0.7978±0.1926
SECOM			
LE-CAS	<b>0.6967±0.0416</b>	<b>0.5930±0.0192</b>	<b>0.7931±0.1427</b>
LE	0.5852±0.0437	0.3247±0.0233	0.4388±0.1904
S-LE	0.6082±0.0367	0.5552±0.0296	0.5478±0.1309
GGLE	0.6552±0.0345	0.4925±0.0334	0.6546±0.1071
SCPLE	<b>0.6923±0.0570</b>	<b>0.5892±0.0270</b>	0.6339±0.1286
DrivFace			
LE-CAS	<b>0.6738±0.0227</b>	<b>0.5771±0.0280</b>	<b>0.7563±0.0470</b>
LE	0.3691±0.0322	0.1917±0.0267	0.4614±0.0234
S-LE	0.4452±0.0237	0.3156±0.0193	0.5324±0.0193
GGLE	0.5892±0.0334	0.4622±0.0138	0.6474±0.0291
SCPLE	0.6037±0.0190	0.5112±0.0142	0.6890±0.0391



**Figure 2.** Results of FMI on ten datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace) under different target dimensionalities (c).



**Figure 3.** Results of F-measure on ten datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace) under different target dimensionalities (c).



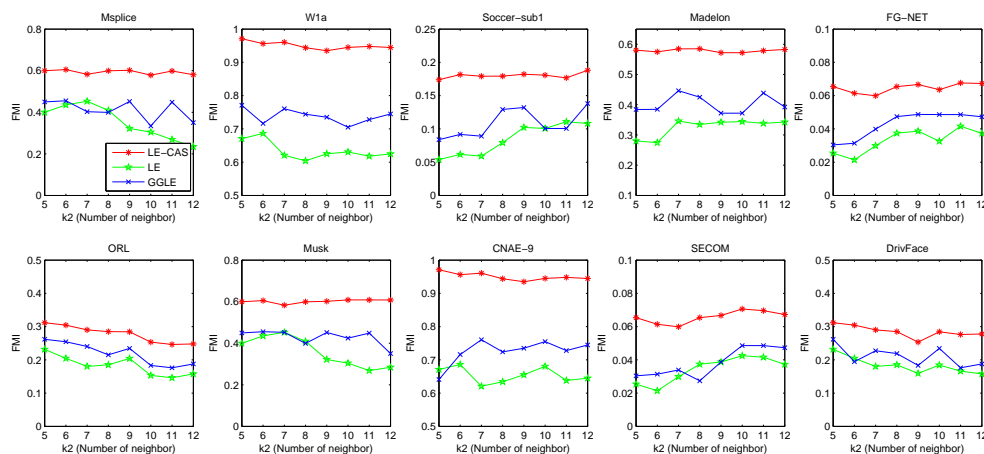
**Figure 4.** Results of PU on ten datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace) under different target dimensionalities (c).

After detailed observations of the results on different datasets, we found that the FMI results on several datasets are much lower than others (Look at the FMI values on dataset soccer-sub1, ORL, DrivFace). Since FMI is an evaluation index which measures the consistency between the clustering results and the real labels as we mentioned before, there are good reasons to believe that clustering occur in these datasets are not so perfect, which means that the clusters identified by the clustering method we used in LE-CAS (kernel  $k$ -means) are not well suitable for the true decision boundaries. Fortunately, even though the clusters we obtained are not so satisfied, the performance of the LE-CAS is still better than the performance of GGLE, S-LE, SCPLC and LE, we can still get relatively good results.

#### 4.2. Parameter Sensitivity Analysis

Because we adopted kernel  $k$ -means just as data preprocessing in our approach, so for the determination of  $k_1$  value, we make the  $k_1$  from 2 to 10, repeat several times kernel  $k$ -means on each  $k_1$  value (to avoid local optimal solution), and calculate the current average contour coefficient. Finally, the  $k_1$  corresponding to the maximum contour coefficient is selected as the final number of clusters.

LE-CAS, LE, and GGLE make use of a  $k$ NN graph to set up the adjacency matrix and the weights between samples, and hence for graph-based dimensionality reduction. To study the sensitivity of these methods to the input value of  $k_2$ , we increase  $k_2$  from 5 to 12, and report the FMI values of these algorithms under each input value of  $k_2$  (number of neighbors) in Figure 5.

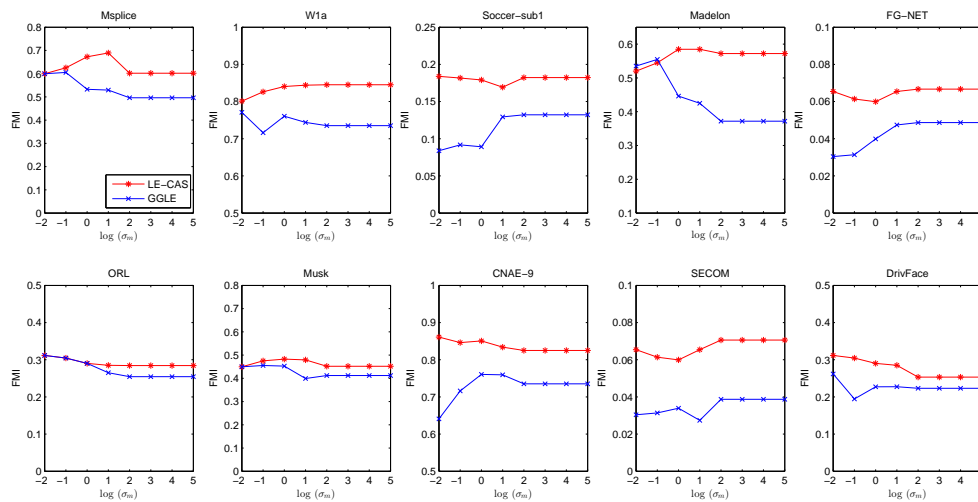


**Figure 5.** Results (FMI) vs.  $k_2$  on ten datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace).

From this figure, we can observe that no matter how the  $k_2$  value changes, the results of LE-CAS on these datasets are better than those of another two comparing algorithms. Fluctuations mainly happen in LE and GGLE, while LE-CAS gets a relatively smooth curve. GGLE adopts the  $k$ NN graph constructed in the original high-dimensional space to explore the local geometric structure of samples. It uses the  $k_2$  nearest neighbors of a sample to seek the linear relationships between  $k_2$  nearest neighbors. Therefore, it still focuses on the local manifold, and is sensitive to  $k_2$ . As  $k_2$  increase, the linear relationship between them becomes increasingly complicated, which causes the increase of error rate. LE and LE-CAS also adopt the  $k$ NN graph, LE-CAS additionally uses the global cluster structure of samples to adjust the similarity between neighborhood samples. The performance margin between LE-CAS and LE, and the better stability of LE-CAS proves the effectiveness of clustering-adjusted similarity. From these results, we can conclude LE-CAS is robust to noise and can work well under a wide range of input values of  $k_2$ .

Both LE-CAS and GGLE specify edge weight by Gaussian heat kernel function. From Equation (11), we can see that they both rely on a suitable Gaussian heat kernel width  $\sigma_m$ .  $\sigma_m$  should not be too

small or too large. If  $\sigma_m$  is too small, the similarity in Equation (11) will be close to 0, On the other hand,  $\sigma_m$  should not be too big. If  $\sigma_m$  is too big, the similarity in Equation (11) will be close to 1. In our previous experiments, we set  $\sigma_m$  as the mean of square Euclidean distance between all training instances for both LE-CAS and GGLE. To investigate the sensitivity of  $\sigma_m$  on LE-CAS and GGLE, we conduct experiments to investigate the influence of  $\sigma_m$ . In the following experiments, we increase  $\sigma_m$  from  $1 \times 10^{-2}$  to  $1 \times 10^5$  for LE-CAS and GGLE. Other parameter settings are kept the same with previous experiments. Similarly, we run 30 independent experiments for each fixed  $\sigma_m$  and report the FMI under each  $\sigma_m$  in Figure 6.

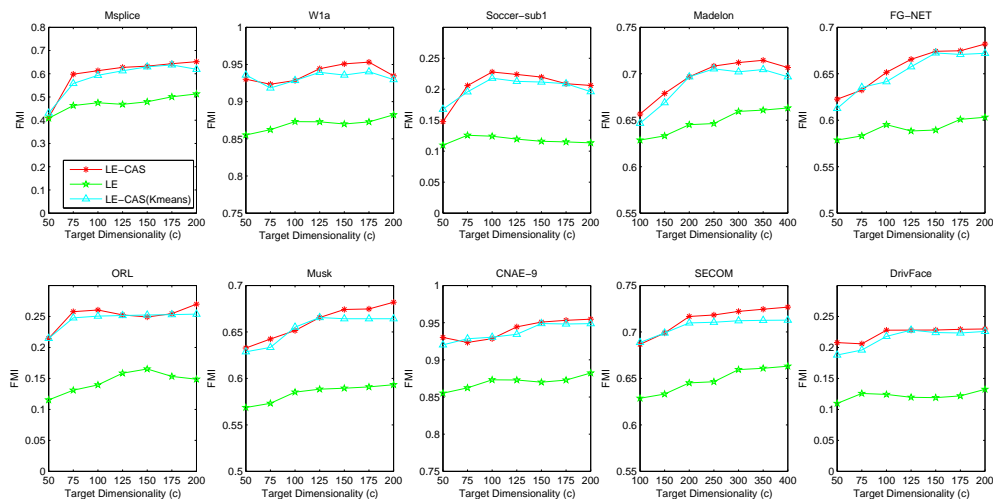


**Figure 6.** Results (FMI) vs.  $\sigma_m$  on ten datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace).

From Figure 6, we can find that LE-CAS outperforms LGC in a wide range of  $\sigma_m$ . Both GGLE and LE-CAS depend on a suitable  $\sigma_m$ . The performance of two methods reaches relatively stable with the increase in  $\sigma_m$ . The FMI of LE-CAS is similar to GGLE when  $\sigma_m$  is too small. This is because if  $\sigma_m$  is too small, the clustering-adjusted similarity is similar to the original Gaussian heat kernel similarity. The performance of both LE-CAS and GGLE becomes relatively stable when  $\sigma_m \geq 100$  in our experiments, and LE-CAS performs significantly better than GGLE. The reason is that the clustering-adjusted similarity has its effect. From the above results, we can conclude that LE-CAS effectively improves the performance of GEDR methods and makes stable performance in a wide range of  $\sigma_m$ .

### 4.3. Robustness Analysis

Since our method combines clustering with dimensionality reduction, the performance of LE-CAS seems heavily depends on whether the clustering method used in previous steps. To clarify the robustness of our method, we employed two different clustering methods,  $k$ -means and kernel  $k$ -means, as the initial clustering method in the course of LE-CAS. The performance of these methods are shown as Figure 7.



**Figure 7.** Results of FMI on ten datasets (Msplice, W1a, Soccer-sub1, Medelon, FG-NET, ORL, Musk, CNAE-9, SECOM, DrivFace) under different target dimensionalities (c).

From Figure 7, we can illustrate that the performances of two clustering methods are merely close to each other on ten datasets and each method performs better than LE method, which indicates that the performance of our LE-CAS cannot be influenced much by the clustering method used in initial steps.

## 5. Conclusions

In this paper, we introduced the Laplacian Eigenmaps dimensionality reduction based on Clustering-Adjusted Similarity(LE-CAS), which leverages local manifold structure and global cluster structures to adjust the similarity between neighborhood samples. In particular, the adjusted similarity can reduce the similarity between pairwise samples from different clusters while maintain the similarity between samples of the same cluster. Experimental results on public benchmark datasets show that that the clustering-adjusted similarity improve the performance of classical LE and outperforms other related competitive solutions.

In our future work, we want to explore other principle ways to refine the similarity between instances and further improve the performance of GEDR methods. In addition, we will pay more attention to weakly supervised graph-based dimensionality reduction. Otherwise, for problems happened near the decision boundary, fuzzy set theory may also can help. We are willing to compare our method to fuzzy set theory in the future.

**Author Contributions:** Conceptualization, Jun Wang; Investigation, Honghu Zhou; Methodology, Honghu Zhou and Jun Wang; Supervision, Jun Wang; Validation, Honghu Zhou; Writing – original draft, Honghu Zhou; Writing – review & editing, Jun Wang.

**Acknowledgments:** This work is supported by Natural Science Foundation of China (61873214), Fundamental Research Funds for the Central Universities (XDJK2019B024).

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## References

1. Song, M.; Yang, H.; Siadat, S.H.; Pechenizkiy, M. A comparative study of dimensionality reduction techniques to enhance trace clustering performances. *Expert Syst. Appl.* **2013**, *40*, 3722–3737.
2. Fan, M.; Gu, N.; Hong, Q.; Bo, Z. Dimensionality reduction: An interpretation from manifold regularization perspective. *Inf. Sci.* **2014**, *277*, 694–714.

3. Vlachos, M.; Domeniconi, C.; Gunopulos, D.; Kollios, G.; Koudas, N. Non-linear dimensionality reduction techniques for classification and visualization. In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Edmonton, AB, Canada, 23–26 July 2002; pp. 645–651.
4. Tao, M.; Yuan, X. Recovering Low-Rank and Sparse Components of Matrices from Incomplete and Noisy Observations. *SIAM J. Optim.* **2011**, *21*, 57–81.
5. Shi, L.; He, P.; Liu, E. An incremental nonlinear dimensionality reduction algorithm based on ISOMAP. In Proceedings of Australian Joint Conference on Advances in Artificial Intelligence, Sydney, Australia, 5–9 December 2005; pp. 892–895.
6. Massini, G.; Terzi, S. A new method of Multi Dimensional Scaling. *Fuzzy Information Processing Society*; IEEE: Toronto, ON, Canada, 2010; ISBN 978-1-4244-7859-0.
7. Vel, O.D.; Li, S.; Coomans, D. Non-Linear Dimensionality Reduction: A Comparative Performance Analysis. In *Learning from Data*; Springer: Berlin, Germany, 1996; pp. 323–331.
8. Leinonen, T. *Principal Component Analysis and Factor Analysis*, 2nd ed.; Springer: New York, NY, USA, 2004; ISBN 978-0-387-95442-4.
9. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326.
10. Belkin, M.; Niyogi, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 585–591.
11. Dornaika, F.; Assoum, A. Enhanced and parameterless Locality Preserving Projections for face recognition. *Neurocomputing* **2013**, *99*, 448–457.
12. Qiang, Y.; Rong, W.; Bing, N.L.; Yang, X.; Yao, M. Robust Locality Preserving Projections With Cosine-Based Dissimilarity for Linear Dimensionality Reduction. *IEEE Access* **2017**, *5*, 2676–2684.
13. He, X. Locality preserving projections. Ph.D. Thesis, University of Chicago Chicago, IL, USA, 2005.
14. Yu, G.; Hong, P.; Jia, W.; Ma, Q. Enhanced locality preserving projections using robust path based similarity. *Neurocomputing* **2011**, *74*, 598–605.
15. Deng, C.; He, X.; Han, J. Semi-supervised Discriminant Analysis. In Proceedings of IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
16. Bunte, K.; Schneider, P.; Hammer, B.; Schleif, F.M.; Villmann, T.; Biehl, M. Limited Rank Matrix Learning, discriminative dimension reduction and visualization. *Neural Netw.* **2012**, *26*, 159–173.
17. Song, Y.; Nie, F.; Zhang, C.; Xiang, S. A unified framework for semi-supervised dimensionality reduction. *Pattern Recognit.* **2008**, *41*, 2789–2799.
18. Yu, G.X.; Peng, H.; Wei, J.; Ma, Q.L. Mixture graph based semi-supervised dimensionality reduction. *Pattern Recognit. Image Anal.* **2010**, *20*, 536–541.
19. Yan, S.; Xu, D.; Zhang, B.; Zhang, H.J.; Yang, Q.; Lin, S. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 40–51.
20. Effrosyni, K.; Yousef, S. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 2143–2156.
21. Belkin, M.; Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* **2003**, *15*, 1373–1396.
22. Trad, M.R.; Joly, A.; Boujemaa, N. Large Scale KNN-Graph Approximation. In Proceedings of 2012 IEEE 12th International Conference on Data Mining Workshops, Brussels, Belgium, 10 December 2012, pp. 439–448.
23. Tenenbaum, J.B.; De Silva, V.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323.
24. Weinberger, K.Q.; Saul, L.K. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *AAAI*, **2006**, *6*, 1683–1686.
25. Weinberger, K.Q.; Saul, L.K. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vis.* **2006**, *70*, 77–90.
26. Parsons, L.; Haque, E.; Liu, H. Subspace clustering for high dimensional data: A review. *ACM Sigkdd Explor. Newsl.* **2004**, *6*, 90–105.
27. Zeng, X.; Luo, S.; Wang, J.; Zhao, J. Geodesic distance-based generalized Gaussian Laplacian Eigenmap. *J. Softw.* **2009**, *20*, 815–824.

28. Raducanu, B.; Dornaika, F. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognit.* **2012**, *45*, 2432–2444.
29. Ning, W.; Liu, J.; Deng, T. A supervised class-preserving Laplacian eigenmaps for dimensionality reduction. In Proceedings of International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, Changsha, China, 13–15 August 2016; pp. 383–389.
30. Dhillon, I.S.; Guan, Y.; Kulis, B. Kernel k-means, Spectral Clustering and Normalized Cuts. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, Seattle, WA, USA, 22–25 August 2004; pp. 551–556.
31. Chen, X.; Lu, C.; Tan, Q.; Yu, G. Semi-supervised classification based on clustering adjusted similarity. *Int. J. Computer. Appl.* **2017**, *39*, 210–219.
32. Nemec, A.F.L.; Brinkhurst, R.O.B. The Fowlkes Mallows statistic and the comparison of two independently determined dendrograms. *Can. J. Fish. Aquat. Sci.* **1988**, *45*, 971–975.
33. Liu, Z.; Tan, M.; Jiang, F. Regularized F-measure maximization for feature selection and classification. *BioMed Res. Int.* **2009**, *2009*, 617946.
34. Aliguliyev, R.M. Performance evaluation of density-based clustering methods. *Inf. Sci.* **2009**, *179*, 3583–3602.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).