AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 5(5): 4702–4721.
DOI:10.3934/math.2020301
Received: 14 January 2020
Accepted: 17 May 2020
Published: 28 May 2020

Research article

A data partition strategy for dimension reduction

Li Liu, Long Zhang, Huaxiang Zhang*, Shuang Gao, Dongmei Liu and Tianshi Wang

School of Information science and Engineering, Shandong Normal University, 250358, Jinan, China

* **Correspondence:** Email: huaxzhang@163.com.

Abstract: Based on the idea that different data contributes differently to dimension reduction, we propose a weighted affinity propagation strategy to partition the data into representative data and common data. The representative data have dominant features while the common data have less importance. In the dimension reduction, the sparse relationship and geodesic distances between pairs of representative data are preserved, and the common data are recovered through a linear combination of the adjacent representative data in the projection space. Experiments on benchmark datasets demonstrate the competitive performance of the proposed method with other methods.

Keywords: dimension reduction; affinity propagation; geodesic distance; linear combination
Mathematics Subject Classification: 68TXX, 68U35

1. Introduction

With the rapid growth of digital information in recent years, the effective management and rational utilization of large-scale information become urgent in practical applications, such as cross-modal retrieval [1,2], control system [3–6], and data analysis [7,8]. Dimension reduction becomes an effective tool to find valuable information with less information loss and strong discriminant ability in low-dimensional space.

A large number of dimension reduction methods have been proposed in the past few decades. Principal Components Analysis (PCA) [9] and Linear Discriminant Analysis (LDA) [10–12] are the most popular methods due to their simplification and efficiency. PCA maximizes the global variance to obtain the low-dimensional subspace, while LDA maximizes the ratio between the trace of between-class scatter and the trace of within-class scatter. PCA and LDA are linear methods, and they cannot achieve appreciated projection results for the nonlinear high-dimensional data, so many manifold learning methods have been developed. Representative algorithms mainly include Isometric Mapping [13], Kernel Principal Component Analysis [14], and Maximum Variance Unfolding [15]. These methods are nonlinear dimension reduction methods and can effectively show the structure of

the original data, but they preserve solely the global structure information.

Locality Preserving Projection aims to find an optimal subspace that can preserve the local relationship between adjacent samples as much as possible [16–20]. Neighborhood Preserving Embedding [21], Locality Sensitive Discriminant Analysis [22], and Marginal Fisher Analysis [23] are all local methods. Sparsity Preserving Projection [24, 25] is also one of the most influential methods and preserves the sparse reconstructive relationship of the data. The sparse representation structure means the sparse reconstructive relationship of the data through basis vectors with few nonzero elements in sparse subspace. To extend the Sparsity Preserving Projection to a semi-supervised embedding method, Weng et al. [26] introduce the idea of in-class constraints and propose a semi-supervised method for data embedding named Constrained Sparsity Preserving Embedding. Besides, the representative methods also include Sparsity Preserving Discriminant Analysis [27] and Multi-view Sparsity Preserving Projection [24]. Taking into account the local and global structure information, Cai et al. [28] present a linear dimension reduction algorithm named Global and Local Structure Preserving Projection to keep both global and local clustering structures hidden in data. A sparse dimension reduction method named Sparse Global-Local Preserving Projections is proposed [29]. The method preserves both the global and local structures of the data set and extracts sparse transformation vectors that can reveal meaningful correlations between variables. Liu et al. [30] utilize the collaborative representation to construct within-class and between-class graphs and transform the dimensionality reduction task into a graph embedding framework. Hinton et al. [31] describe an effective way of initializing the weights that allows deep auto-encoder networks to learn low-dimensional codes that work much better than PCA to reduce the dimensionality. A deep variational auto-encoder [32] is utilized for unsupervised dimension reduction of single-cell RNA-seq data. The method can explicitly model the dropout events and find the nonlinear hierarchical feature representations of the original data.

Figure 1. Three pictures from one person in Yale B dataset.

In fact, in real applications, some data have special meanings and the importance of each point is different. One example is illustrated by the three figures in Figure 1 which describe one person. The discriminative features that can be extracted from the figures are different. This inspires us to pay more attention to the important points in the process of dimension reduction, whereas the existing approaches usually ignore the different importance of points. In this paper, we partition the data in a high-dimensional space into two classes: the representative data and the common data, and deal with different kinds of data with different methods. The representative data dominate the feature of the entire model and their projection largely determines the projection of the whole model. Therefore, we focus on the representative data in dimension reduction, and then the projections of the common data are then recovered from the projected representative data. The contributions of this paper are stated as follows:

- A weighted affinity propagation algorithm is proposed to cluster the data with different weights. The weight of each point provides an importance factor for clustering, which makes the data with high weights play more important roles in the clustering process.
- Two approaches of dimension reduction are utilized to preserve the structure information of the representative data and common data with different aims. The data that have more discriminative features or more importance are paid more attention to in the process of dimension reduction.

The rest of this paper is organized as follows. The related work is introduced in Section 2. Section 3 describes the details of the proposed method. The contents include data partition and structure information preserving projections of the representative data and common data. The experimental results and discussions are presented in Section 4. Finally, Section 5 summarizes this work.

2. Related work

2.1. Affinity propagation algorithm

Affinity Propagation (AP) is a clustering method proposed by Frey and Dueck [33]. Compared with other methods, AP clustering achieves the number of clusters automatically through message exchange. By exchanging messages among data points simultaneously, AP determines whether a point should be an exemplar (cluster center), or the exemplar to which it should belong.

AP algorithm passes real-valued messages between points until a high-quality set of exemplars and corresponding clusters are found. The responsibility $r(i,k)$ is the message sent from data x_i to candidate exemplar point x_k and reflects the possibility of point x_k to be the exemplar. The availability $a(i,k)$ reflects how appropriate it would be for point x_i to choose candidate exemplar x_k as its exemplar. Mathematically, the responsibility $r(i,k)$ and the availability $a(i,k)$ are updated as

$$r(i,k) = s(i,k) - \max_{k' \neq k}\{a(i,k') + r(i,k')\} \tag{2.1}$$

$$a(i,k) = \begin{cases} \min\{0, r(k,k) + \sum_{i' \neq i,k} \max\{0, r(i',k)\}\}, & i \neq k \\ \sum_{i' \neq k} \max\{0, r(i',k)\}\}, & i = k \end{cases} \tag{2.2}$$

where $s(i, j)$ describes the similarity relationship between point x_i and point x_j and the negative squared Euclidean distance may be considered as the similarity measure.

The responsibility and the availability are updated repeatedly until some step criterion is met. For example, the iterative process can be terminated after a fixed number of iterations. Namely, for point x_i the value of k is

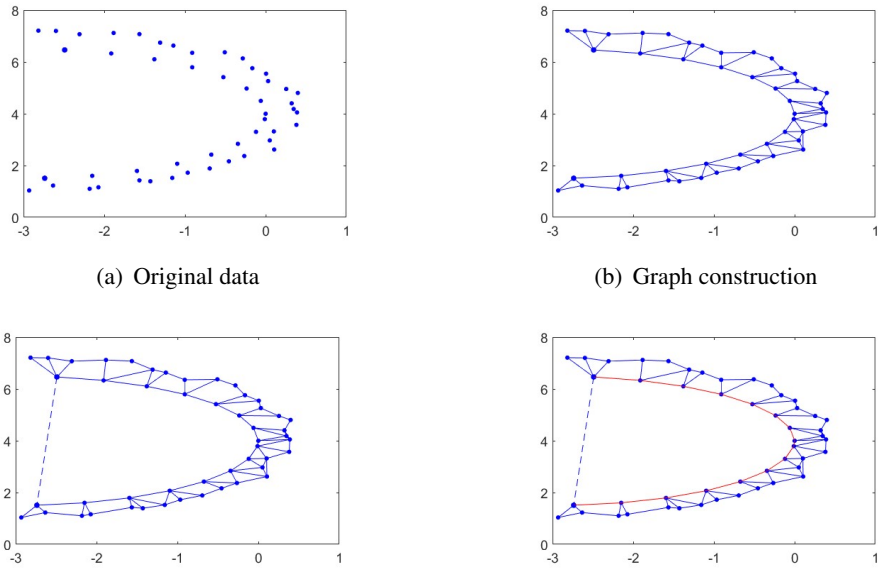
$$k = \arg\max_{j}\{a(i, j) + r(i, j)\} \tag{2.3}$$

If $k = i$, then point x_i is an exemplar or cluster center. If $k \neq i$, then point x_k is an exemplar for point x_i.

2.2. Geodesic distance measure

The distance function determines how close the data are. Euclidean distance is widely used to calculate the distance between two data points, while sometimes it cannot reflect the real distance [34, 35]. For the points in Figure 2(a), the Euclidean distance between two specified points is represented

by dotted line in Figure 2(c). According to this metric the two points appear deceptively close, while they are on the opposite parts. Geodesic distance between two points is defined usually as the length of the shortest paths. Figure 2(d) shows the benefit of Geodesic distance. In this case, the two points linked by dottted line are not neighbors according to Geodesci distance measure.



(a) Original data (b) Graph construction

(c) Euclidean distance (d) Geodesic distance

Figure 2. The different distance expression. The blue dotted line represents the Euclidean distance, the red line represents the Geodesic distance.

For two points $s, t \in X$, there exists a shortest Geodesic path from s to t. To compute the Geodesic distance between s and t, we approximate the Geodesic path on graph G. If the Euclidean distance between two points is lower than a given threshold, the line-segment between two points is chosen as the edge of graph G as shown in Figure 2(b). The Euclidean distance is assigned to the edge as the weight.

The Geodesic distance between s and t is defined as follows

$$d_G(s,t) = \min Path(s,t) = \min_{p \in Path} \min_{x_i \in p, x_i \neq s,t} d(s,x_1) + d(x_1,x_2) + \cdots + d(x_{|p|},t) \tag{2.4}$$

where $d(x_i, x_j) = \|x_i - x_j\|$, $Path$ denotes the path set that contains all the paths between points s and t, p is one of the paths, x_i is the point on the path p.

The graph G is a weighted undirected graph and there exist several algorithms to achieve the shortest distance of the graph G, such as Floyd-Warshall algorithm, which gets the shortest distance between any two points and outputs the correct result as long as no negative cycles exist in the graph G. The shortest distance is taken as the approximation of the Geodesic distance, which reflects the distance more accurately. The distance measure in this paper is based on Geodesic distance and applied in the search of k nearest neighbors.

3. Description of the proposed method

For the data that have special meaning or different discriminative features, the dimension reduction approach based on data partition and structure information preserving is proposed. Firstly, data points are partitioned into representative data and common data through the weighted affinity propagation clustering algorithm. Furthermore, sparse relationship and Geodesic distance preserving projections are adopted to project the representative data from the original space to the low-dimensional space. Finally, the location of the common data can be achieved through the linear combination of its adjacent representative points. The outline of the proposed method is shown in Figure 3.

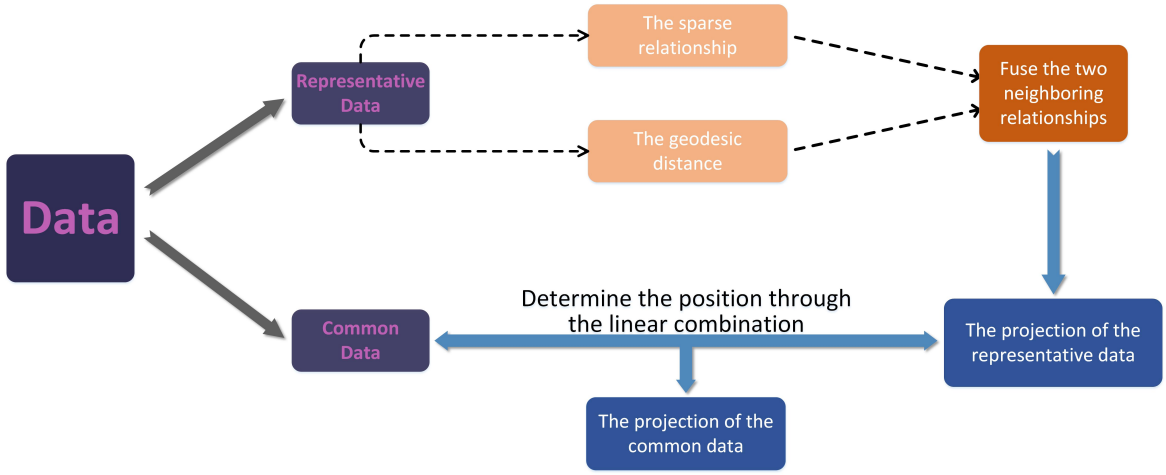


Figure 3. Outline of the proposed method.

3.1. Weight estimation of data points

In the process of dimension reduction, the data have different discriminative features or importance, so it is not a good choice to regard the data as the same. In many practical applications, the weights should be estimated except the data that have weights itself.

The density has some connection with the importance. In general, the bigger the value of density is, the more important it is. Based on the assumption, we introduce the kernel function to describe the importance as the weight. For any $x \in X$, where $X = \{x_1, \cdots, x_n\}$ is the dataset, the kernel weight estimator $f(x; h)$ is given by the following formula

$$f(x;h) = \frac{1}{nh}\sum_{i=1}^{n} k(\frac{x_i - x}{h}) \tag{3.1}$$

where $h > 0$ is the bandwidth, and $k(\cdot)$ is the Gaussian kernel function.

Kernel weight estimator uses a fixed bandwidth h over the whole data support. We use the data-driven least squares cross-validation method to select the bandwidth h by minimizing the cross-validation function $CV_f(h)$ [36]

$$CV_f(h) = \frac{1}{n^2 h}\sum_{i=1}^{n}\sum_{j=1}^{n} \bar{k}(\frac{x_i - x_j}{h}) - \frac{1}{n(n-1)h}\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} k(\frac{x_i - x_j}{h}) \tag{3.2}$$

In the practical applications, the weights of the data have special meaning. Three-dimensional points have not topological information, so it is difficult to do some operations, such as deformation and surface reconstruction. Curvature is the geometrical property and a good invariant feature of local surfaces. Curvature in 3D space usually refers to Gaussian curvature, mean curvature and principal curvature. Gaussian curvature K and mean curvature H can be estimated using the method proposed by [37]. The principal curvatures k_1, k_2 are calculated by Gaussian curvature K and mean curvature H:

$$\begin{cases} k_1 = H - \sqrt{H^2 - K} \\ k_2 = H + \sqrt{H^2 - K} \end{cases} \tag{3.3}$$

Curvature can be negative or positive according to whether it is consistent with the manifold orientation. Here, the absolute value of the curvature is taken as the weight. The points with large weights are more important than the others and should be paid more attention to in the process of dimension reduction.

3.2. Weighted affinity propagation clustering algorithm

In many applications, each point has a special meaning and the importance of each point is different. It is obvious that the data should have different possibilities to become an exemplar. To give the same possibility of becoming the exemplar for the data with different importance is not a good choice. That ignores the different importance of points, so we introduce the weighted affinity propagation clustering algorithm to solve the problem.

In the traditional AP algorithm, a point has more possibility to be the exemplar if there are more neighbors around it. Therefore, for the point x with high weight, we add virtual points around it to increase the possibility of the point x to be the exemplar. The values and position of virtual points are the same as the point x. For the number of virtual points, it depends on the weight of the given point x. The function $V(w)$ reflecting the relationship between the number of virtual points and the weight of the given point is constructed. As the range of the weights is different in real applications, the number of virtual points cannot be approximated accurately.

Let x_i be the point with weight w_i, the number of virtual points is n_i. For points set X, the sum of the virtual points is $\sum_{i=1}^{N} n_i$, where N is the number of points in the set X. Before the virtual points are added to the points set X, the responsibility $r(i,k)$ and the availability $a(i,k)$ are $N \times N$ matrices. After M virtual points are added, the two matrices change into $(N+M) \times (N+M)$ matrices. The virtual points are the copies of the original point in the dataset S, so the responsibility $r(i,k)$ and the availability $a(i,k)$ of virtual points are the same for the original points.

The rules for message propagation are updated as follows

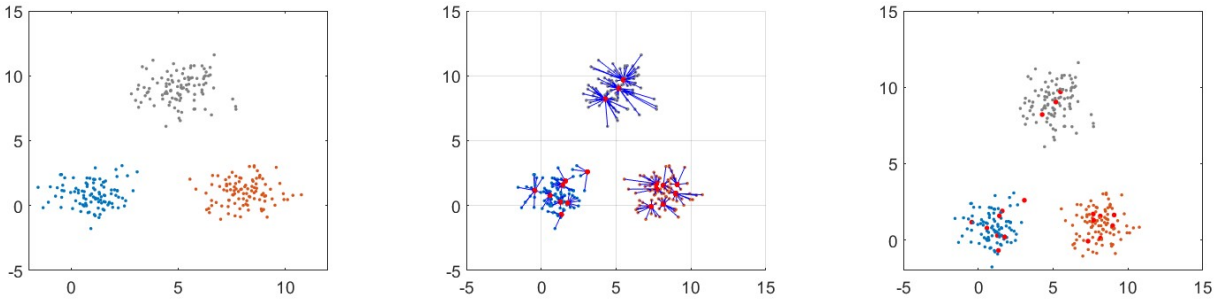
$$r^{(t+1)}(i,k) = (1-\lambda) \times (s(i,k) - \max_{k' \neq k}\{a^{(t)}(i,k') + s(i,k')\}) + \lambda \times r^{(t)}(i,k) \tag{3.4}$$

$$a^{(t+1)}(i,k) = \begin{cases} (1-\lambda) \times \min\{0, r^{(t)}(k,k) + \sum_{i' \neq i,k} \max\{0, r^{(t)}(i',k)\}\} + \lambda \times a^{(t)}(i,k), & i \neq k \\ (1-\lambda) \times \sum_{i' \neq k} \max\{0, r^{(t)}(i',k)\}\}, & i = k \end{cases} \tag{3.5}$$

where λ is damping factor in the iterative steps.

The higher value of the damping factor λ leads to slower convergence rates. In the iterative process of the algorithm, oscillation and non-convergence occur when two or more points are suitable as

representative points in the same class cluster. The original points are copied many times, so the values of $r(i,k)$ and $a(i,k)$ of an original point and its virtual points should be unified and the sum is considered to judge whether the point is the exemplar. An example of the weighted AP algorithm is shown in Figure 4.



(a) Original data (b) Graph construction (c) Euclidean distance

Figure 4. The different distance expression. The blue dotted line represents the Euclidean distance, the red line represents the Geodesic distance.

The exemplar means a cluster center and is assumed to be more important than the other points in one class. Therefore, we choose the exemplars as the representative points and the rest as the common ones. Besides the exemplars, the points whose weights are higher than a given threshold are also taken as the representative points. These points are important but far away from the other points. In Figure 4(c), the red points are the representative data and the rest points are the common data.

3.3. Structure information preserving for the representative data

In this section, we give the structure information preserving projection for the representative data. The structure information refers to sparse neighborhood and Geodesic distance. The representative data have more importance and determines the distribution of the original data, so the dimension reduction of the representative data is expected to obtain more discriminative features.

3.3.1. Sparse neighborhood preserving projection

The basic idea of sparse neighborhood preserving projection is that the same weight matrix that reconstructs the point x_i should also reconstruct its projection y_i through its neighbors. Sparse neighborhood preserving projection constructs sparse reconstructive weights between representative data and preserves the neighborhood similarity relationship in the low-dimensional subspace. Assume X_i denotes the set of neighbors of point x_i, to minimize the reconstruction error, the weight matrix w_i^S corresponding to x_i are obtained by solving an l_1-minimization problem

$$\min \|w_i^S\|_1, s.t. \|x_i - X_i w_i^S\| < \varepsilon \tag{3.6}$$

The weight matrix w_i^S denotes the reconstruction coefficients following the sparse rule and can be achieved by calculating formula (3.6). The sparse reconstruction error function $E_S(x_i)$ in the projection space can be expressed as the following

$$E_S(x_i) = \sum_{i=1}^{N} \|P_i^T x_i - P_i^T X w_i^S\|^2 \tag{3.7}$$

3.3.2. Geodesic distance preserving projection

To preserve the neighborhood relationship between the point and its neighbors, the sparse neighborhood preserving projection is utilized. However, for some data like Swiss Roll, the sparse neighborhood relationship is not enough to reflect the real relationship, so the Geodesic distance preserving projection is proposed to describe the structure of data points in another way.

Geodesic distance projection preserves the proportion of the Geodesic distances between these points in the set. The matrix w_i^G gives the Geodesic distance relationship between point x_i and its neighbors. The component $w_{i,j}^G$ in the matrix w_i^G can be defined as the following

$$w_{i,j}^G = \frac{e^{-d_G(x_i,x_j)}}{\sum_{x_j \in InClass(x_i)} e^{-d_G(x_i,x_j)}} \tag{3.8}$$

The weight value $w_{i,j}^G$ denotes the distance relationship between x_i and x_j. In the projection space, to keep the Geodesic distance relationship the reconstructive error function $E_D(x_i)$ is as follows

$$E_D(x_i) = \sum_{i=1}^{N} \|P_i^T x_i - P_i^T X w_i^G\|^2 \tag{3.9}$$

3.3.3. Sparse neighborhood and Geodesic distance preserving projection

The neighbors of the point x_i can be k-NN or ε-Neighbors. Here, k-NN is used to choose the neighbors of point x_i. To keep sparse neighborhood relationship, we minimize the error function $E_S(x_i)$. At the same time, we minimize the error function $E_D(x_i)$ to keep the Geodesic distance. To preserve the two relationships in the process of dimension reduction, we get the following total error function

$$\begin{aligned} E(X) &= \sum_{i=1}^{N} E(x_i) \\ &= \lambda_1 \sum_{i=1}^{N} E_S(x_i) + \lambda_2 \sum_{i=1}^{N} E_D(x_i) \\ &= \lambda_1 \sum_{i=1}^{N} \|P_i^T x_i - P_i^T X w_i^S\|^2 + \lambda_2 \sum_{i=1}^{N} \|P_i^T x_i - P_i^T X w_i^G\|^2 \end{aligned} \tag{3.10}$$

where λ_1 and λ_2 are parameters to adjust the balance of the sparse neighborhood relationship and Geodesic distance. The N projection matrices are independent of each other and thus the objective function can be considered as the sum of N subfunctions $E(x_i)$ which are separately optimized, so we can get the following formula

$$
\begin{aligned}
E(X_i) &= \lambda_1 \sum_{i=1}^{M} \|P_i^T x_i - P_i^T X w_{i,j}^S\|^2 + \lambda_2 \sum_{i=1}^{M} \|P_i^T x_i - P_i^T X w_{i,j}^G\|^2 \\
&= \lambda_1 \sum_{i=1}^{M} tr[P_i^T (x_{i,j} - X_i w_{i,j}^S)(x_{i,j} - X_i w_{i,j}^S)^T P_i] + \lambda_2 \sum_{i=1}^{M} tr[P_i^T (x_{i,j} - X_i w_{i,j}^G)(x_{i,j} - X_i w_{i,j}^G)^T P_i] \\
&= \sum_{i=1}^{M} tr[P_i^T (\lambda_1 M_i^S + \lambda_2 M_i^G) P_i]
\end{aligned} \tag{3.11}
$$

where $M_i^S = (x_{i,j} - X_i w_{i,j}^S)(x_{i,j} - X_i w_{i,j}^S)^T$ and $M_i^G = (x_{i,j} - X_i w_{i,j}^G)(x_{i,j} - X_i w_{i,j}^G)^T$.

To avoid degenerate solutions, a constraint $P_i^T X_i X_i^T P_i = I$ is added. Therefore, the objective function can be expressed as the following optimization function

$$
\min tr(P_i^T X_i (I - w_i - w_i^T + w_i^T w_i) X_i^T P_i), s.t. P_i^T X_i X_i^T P_i = I \tag{3.12}
$$

where $w_i = \lambda_1 w_i^S + \lambda_2 w_i^G$, λ_1 and λ_2 satisfy this condition that $\lambda_1 + \lambda_2 = 1$.

The projection matrix that minimizes the objective function is given by the minimum eigenvalue solution to the generalized eigenvalue problem

$$
X_i (I - w_i - w_i^T + w_i^T w_i) X_i^T P_i = \lambda X_i X_i^T P_i \tag{3.13}
$$

The optimization problem can be solved with the generalized eigenvalue decomposition mentioned in the related work as $E(x_i) = \lambda_i^k P_i$. For the data point x_i, the projection in the low-dimensional space is $y_i = P_i^T x_i$.

Algorithm 1: Sparse neighborhood and Geodesic distance preserving projection.

Input: Representative dataset $X = x_1, x_2, \ldots, x_N$, the graph G and the value of λ_1 and λ_2.

Output: Project matrices $P_i, i = 1, 2, \ldots, N$.

Step 1: Transform the optimization problem to a generalized eigenvalue problem; calculate sparse weight matrix w_i^S and gastric distance matrix w_i^G;

Step 2: Solve problem and sort the eigenvalue as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0 \geq \lambda_{d+1} \geq \cdots \geq \lambda_D$;

Step 3: Select the eigenvectors that correspond to positive eigenvalues to form the projection matrix.

3.4. Structure information preserving for the common data

For a common point, because of the uncertain projection position of its adjacent common data, the performance and computational cost of dimension reduction cannot be guaranteed if the location of the common data depends on all the adjacent representative and common data. On the other hand, the representative data have more importance than the common one, so the projection location of the common data can be achieved solely according to its adjacent representative ones. Here, the structure information preserving is to keep the Geodesic distance relationship between the common point and

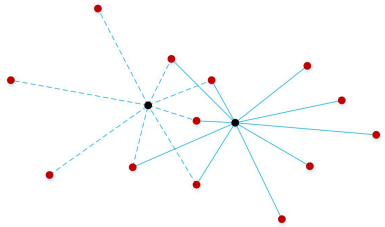


Figure 5. Linear combination of representative points. The red points are the representative points, the black point represents the common point.

its adjacent representative points, which not only preserves the distance relationship but also reduces the running time.

The unknown position of a common point in the low-dimensional space is achieved through the linear combination of its adjacent representative points as shown in Figure 5. The crucial issue for the success of the linear combination method is whether we can obtain appropriate weights for all component systems in an efficient manner. The linear combination can be expressed as follows

$$y_i = \sum_{j=1}^{n} \lambda_j y_j \tag{3.14}$$

where y_i is the projection coordinate of the j-th representative point in the adjacent set Q_i, λ_j is the weight assigned to the projection point y_j. The adjacent representative points set Q_i of the common point x_i is defined as follows: $y_j \in Q_i$, if $d_G(x_i, x_j) < d$ and $y_j \notin Q_i$, otherwise.

Here, we utilize the distance representation method to construct the relationship between the common point and its adjacent representative points in the original space. The distance representation can discover the local relationship and has natural discriminating power. Given the weight threshold, all points within the threshold band are included into the neighbor data set. The definition of the weight λ_j is the following

$$\lambda_j = \frac{e^{-d_G(x_i,x_j)^p}}{\sum_{x_j \in Q_i} e^{-d_G(x_i,x_j)^p}} \tag{3.15}$$

where p is the adjustable parameter.

Through the linear combination of the adjacent representative points of common data x_i, we directly achieve the projection location of the common data x_i in the low-dimensional space. The method preserves the Geodesic distance relationship between the common and its representative points and avoids the solution of equations.

4. Experiments and analysis

To evaluate the performance of the proposed method, we do comparison experiments with the related methods such as Linear Discriminant Analysis (LDA), Locality Preserving Projection (LPP), Multidimensional Scaling (MDS), Factor Analysis (FA) [38], Principal Component Analysis (PCA), Probabilistic Principal Component Analysis (PPCA) [39, 40], Stochastic Neighbor Embedding (SNE) [41], Kernel t-distributed Stochastic Neighbor Embedding (Kt-SNE) [42], Symmetric

Stochastic Neighbor Embedding (SSNE) [43] and Variational Auto-encoder(VAE) [32] on benchmark datasets. FA is a statistical method used to describe variability among correlated variables in terms of a potentially lower number of unobserved variables called factors. The difference between FA and PCA is whether the objective function clearly identify certain unobservable factors from the observed variables. PPCA is an expression of PCA as the maximum likelihood solution of a probabilistic latent variable model, so the information about uncertainty of measurements can be utilized to produce a more accurate model. Kt-SNE and SSNE are proposed based on Stochastic Neighbor Embedding which minimizes the Kullback-Leibler divergence between the similarity matrix of a high-dimensional data set and its counterpart from a low-dimensional embedding. Kt-SNE preserves the flexibility of basic t-SNE, but enables explicit out-of-sample extensions. SSNE uses a symmetrized cost function with simpler gradients. VAE is an autoencoder whose training is regularized to avoid overfitting and ensures the latent space has good properties. The mentioned methods include classical methods and the improved methods.

The datasets with class labels are projected to the low-dimensional space and the projection points are classified using AP algorithm and SVM classifier in different experiments. Here, classification accuracy is introduced to evaluate the performance of different methods and is described as the following

$$Acc = \frac{\sum_{i=1}^{N} \delta(y_i, c_i)}{N} \tag{4.1}$$

where N is the number of data points, y_i and c_i denote the class label after and before dimension reduction, $\delta(y, c)$ is a function that equals one if $y = c$ and equals zero otherwise.

4.1. Experiments on three-dimensional datasets

To visualize the results of dimension reduction, 3D datasets are projected on 2D surface to intuitively show the performance of different methods. In our experiments, 3D datasets include Vowel, Haberman, Swiss Roll and Rabbit model. Vowel dataset has 871 samples and 6 classes, and Haberman dataset has 36 samples and 2 classes. The visualization results of dimension reduction are shown in the Figure 6 and 7 and the classification accuracy is given in Table 1.

Table 1. Classification accuracy on three different datasets.

	Method									
Dataset	FA	LDA	LPP	MDS	PCA	PPCA	Kt-SNE	SNE	SSNE	Ours
Swiss Roll	0.5847	0.8423	0.6980	0.9290	0.9290	0.8760	0.9067	0.9713	0.5847	0.9823
Vowel	0.5109	0.5993	0.4811	0.6383	0.6383	0.4937	0.7141	0.6923	0.2181	0.7566
Haberman	0.5131	0.4547	0.5327	0.5948	0.5948	0.5797	0.6634	0.683	0.5784	0.6830

Swiss Roll is a typical dataset where the Euclidean distance of two points may be small but the Geodesic distance is large. As Swiss Roll is developable and Gaussian curvature of every point is zero, the absolute values of mean curvature are taken as the weights. In this experiment, Swiss Roll is constructed using Matlab program as shown in Figure 8(a) and the generation process of the dataset is as follows: $N = 3000$; $t = (3\pi/2) \times (1 + 2 \times rand(1, N))$; $s = 20 \times rand(1, N)$; $X = [t \cdot cos(t); s; t \cdot sin(t)]$.

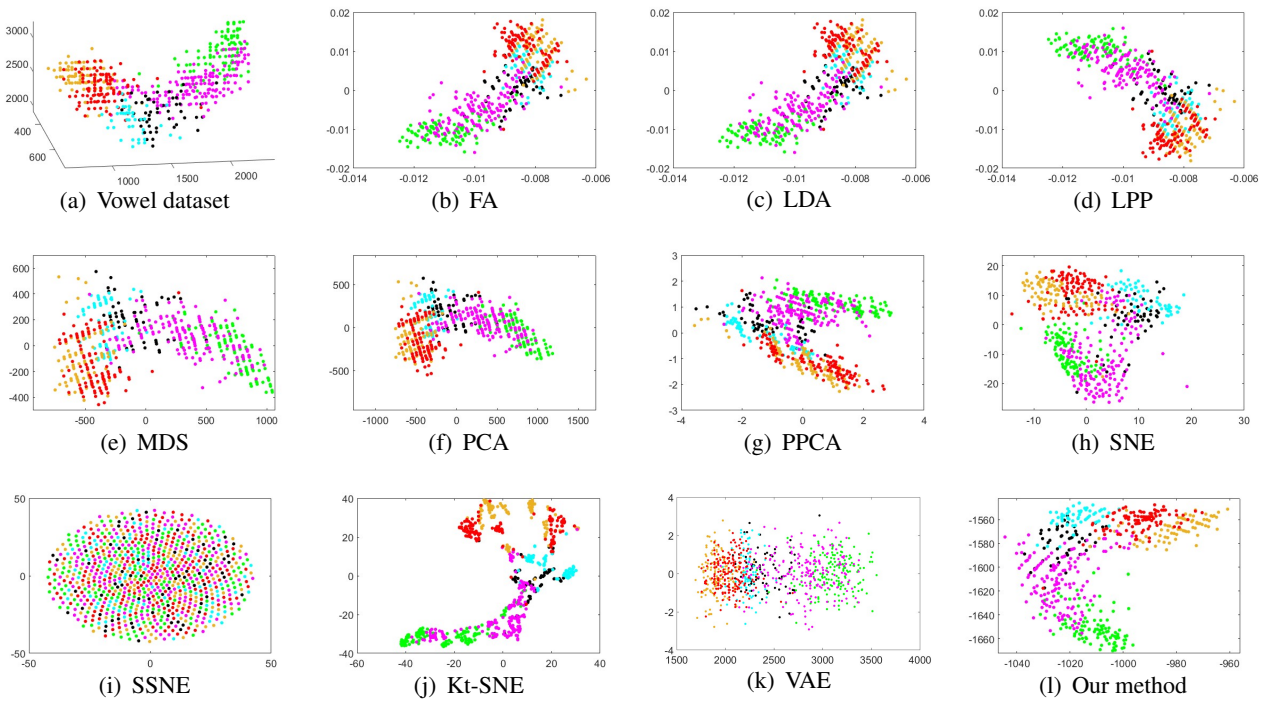


Figure 6. The dimension reduction of Vowel dataset.

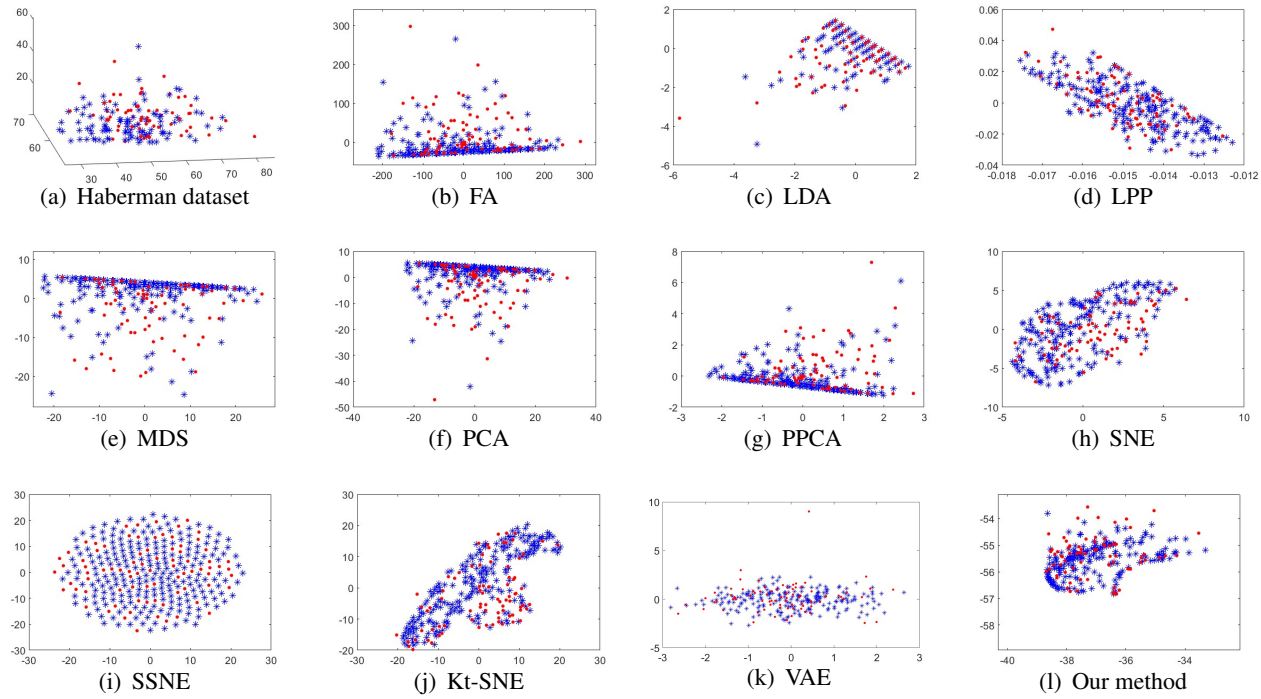
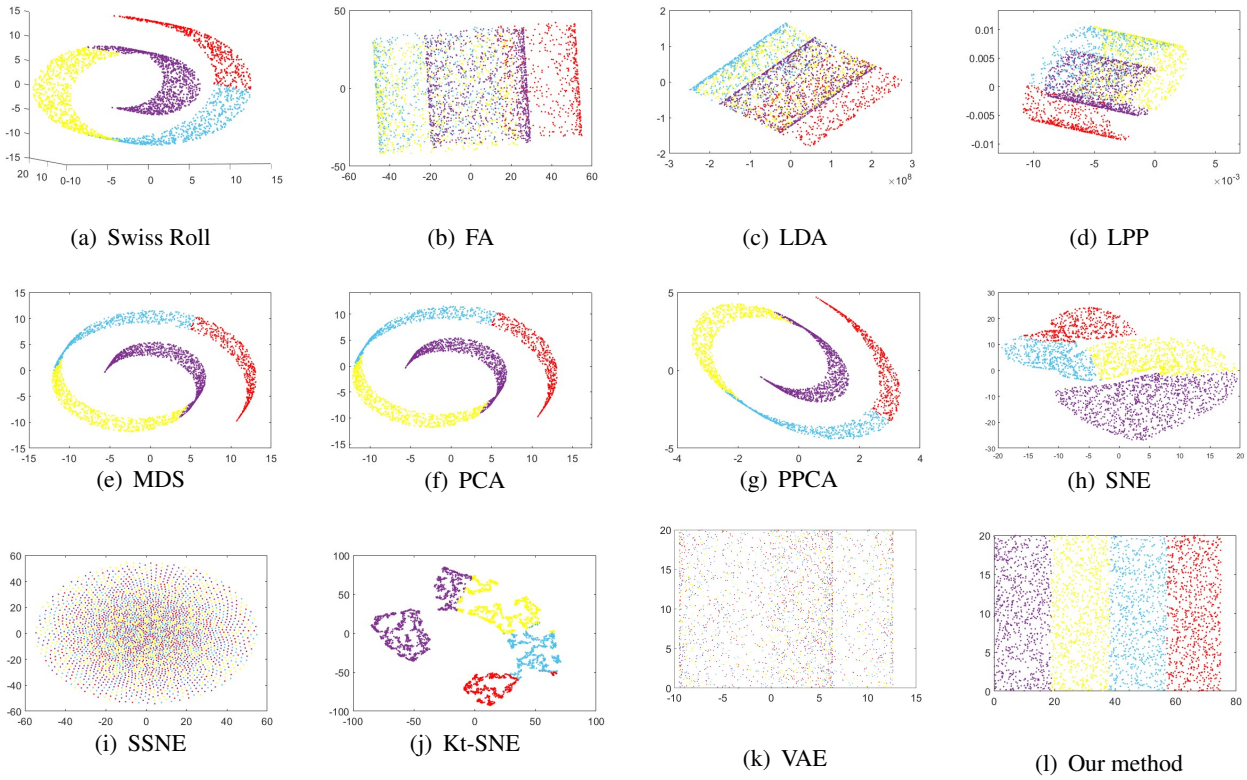


Figure 7. The dimension reduction of Haberman dataset.



(a) Swiss Roll (b) FA (c) LDA (d) LPP

(e) MDS (f) PCA (g) PPCA (h) SNE

(i) SSNE (j) Kt-SNE (k) VAE (l) Our method

Figure 8. The dimension reduction of Swiss Roll.

The projection results using comparative methods are shown in Figure 8(b)-8(k) and Figure 8(l) is the projection results of our method.

The comparative methods based on Euclidean distance, such as MDS, PCA, PPCA and SNE, achieve good dimension reduction results on the Swiss Roll dataset, while the distribution shapes of projection points cannot meet the need in the field of surface construction. Swiss Roll is developable, which means that Swiss Roll can be fully flattened on the 2D surface. In this experiment our method preserves the Geodesic distance of representative data by adjusting λ_1=0, λ_2=1 in formula (3.10), then Swiss Roll is unfolded on the 2D surface as shown in Figure 8(l). Compared with these methods, our method not only achieves higher classification accuracy but also is useful in the model design.

(a) Point cloud of rabbit model (b) Clustering result using weighted AP algorithm

Figure 9. 3D rabbit model.

Rabbit model is widely used to test the results in the computer aided design. The rabbit model has 30,000 points as shown in Figure 9(a) and is divided into 9 classes through the weighted AP clustering algorithm mentioned in Section 3.2.

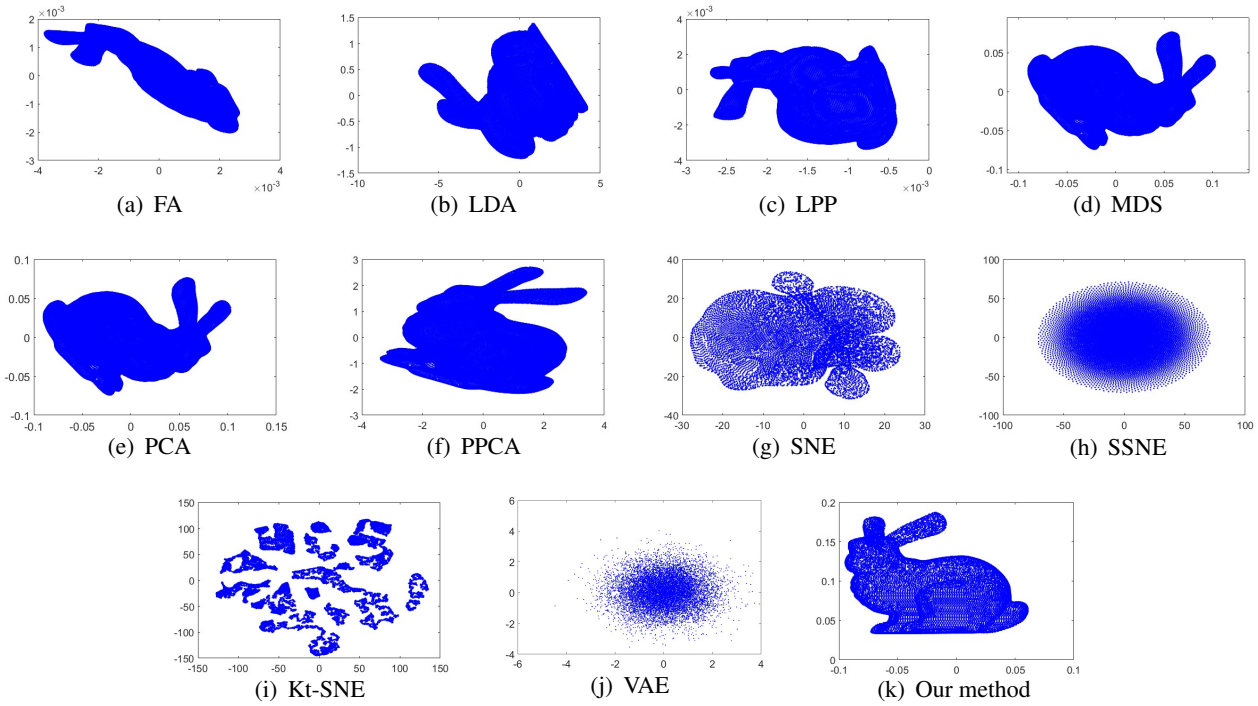


Figure 10. The dimension reduction of Swiss Roll.

We can see from the results shown in Figure 10 that the results using LDA, LPP, MDS, PCA, PPCA and our method can reflect the shape of the rabbit. The difference between these methods is the different viewpoint. From the results, we also observe that the projected points are overlapped in the 2D space. The reason is that the rabbit model is closed and no method can avoid overlapping except splitting the rabbit model into different parts. From the experiments results, we can find that the proposed method get better experimental results than the mentioned methods for 3D data points.

4.2. Experiments on high-dimensional datasets

In this subsection, we show the classification performance of the proposed methods on three different types of high-dimensional datasets. The high-dimensional datasets include control_uni, YaleB and caltech101_silhouettes_16_uni. Control_uni dataset is a measure dataset and there are 600 samples and 6 classes. For each data, the original dimension is 60. Yale B dataset contains 2414 front-view face images of 38 individuals as shown in Figure 11. The original dimension of each sample is 1024. The data in caltech101_silhouettes_16_uni dataset is sparse and the original dimension is 256. There are 8641 samples and 101 classes in the dataset. In our experiments, these methods are compared under different dimensions.

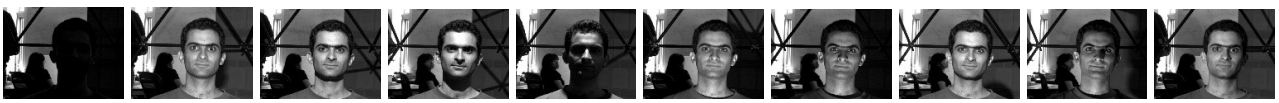
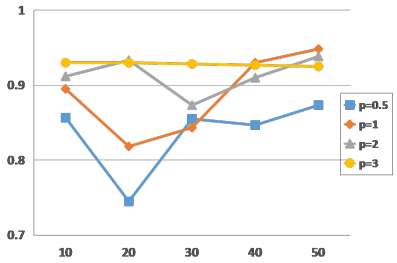


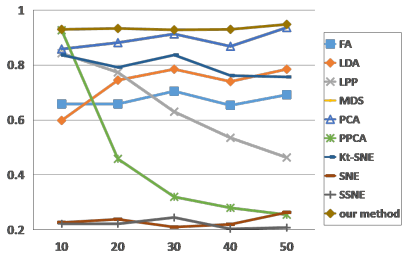
Figure 11. Yale B dataset.

In the experiments, the values of δ_1 and δ_2 in formula (3.14) are both 0.5, which makes the projection of representative points preserve sparse relationship and Geodesic distance at the same

time. The value p in formula (3.15) changes from 0.5 to 3 to get the influence of the common data on the projection results. The classification accuracy graphs with different values of p on control_uni dataset, YaleB dataset and caltech101_silhouettes_16_uni dataset are shown in Figure 12(a),13(a) and 14(a), respectively. On control_uni dataset, the best result in our method with $p = 3$ is compared with the results of other methods and the classification accuracy graph of different methods is shown in Figure 12(b). On YaleB dataset and caltech101_silhouettes_16_uni dataset, classification accuracy graphs are shown in Figure 13(b) and 14(b), where the values of p are both 2 in our method. In these graphs, the vertical axis is the classification accuracy and the horizontal axis represents the dimension.

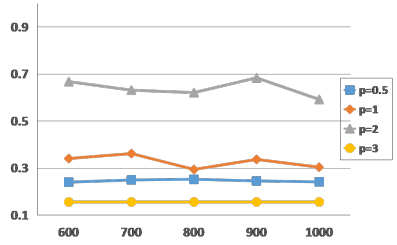


(a) Classification accuracy with different value p

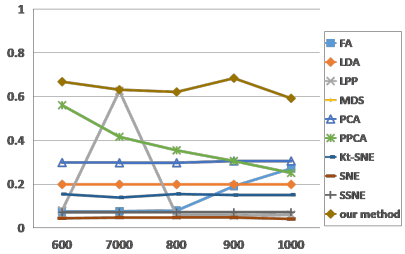


(b) Classification accuracy of different methods

Figure 12. Classification accuracy on control_uni dataset.

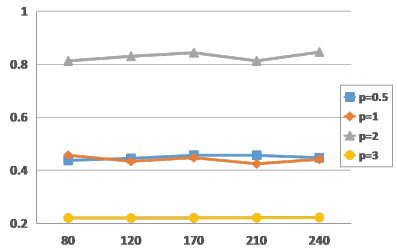


(a) Classification accuracy with different value p

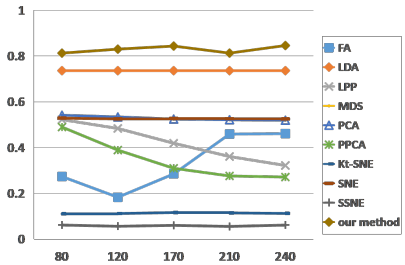


(b) Classification accuracy of different methods

Figure 13. Classification accuracy on Yale B dataset.



(a) Classification accuracy with different value p



(b) Classification accuracy of different methods

Figure 14. Classification accuracy on caltech101_silhouettes_16_uni dataset.

Different value p affects the relationships between the common point and its adjacent representative points. The results shown in Figure 12(a),13(a) and 14(a) illustrate that the distribution of the common data affects the classification accuracy, and the value p in the best case varies for different datasets.

The classification accuracy graphs of different methods in Figure 12(b),13(b) and 14(b) show that the classification results using the methods such as LDA, LPP, PCA, SNE and our method vary in a small range, which means that the dimension has little influence on the classification results using these methods and these methods are stable for the dimensional factors. From the classification accuracy we can see that the worst results of three datasets are less than 30%, 10% and 10%, while our results are the best under any given dimensions and more than 90%, 68% and 80%. From the stability and accuracy, our method achieves a good performance and is superior to the mentioned methods.

4.3. Parameter analysis of λ_1 and λ_2

In this subsection, we mainly analyze the effects of parameter λ_1 and λ_2 on the dimension reduction results. In the objective formula (3.10), λ_1 and λ_2 are utilized to adjust the proportion of sparse neighborhood to Geodesic distance relationship. For the three high-dimensional datasets, the values of classification accuracy with different parameters λ_1 and λ_2 are listed in Table 2,3 and 4.

Table 2. The classification accuracy on control_uni dataset with $p = 3$.

Dim	$\lambda_1, \lambda_2 = 0, 1$	$\lambda_1, \lambda_2 = 0.2, 0.8$	$\lambda_1, \lambda_2 = 0.5, 0.5$	$\lambda_1, \lambda_2 = 0.8, 0.2$	$\lambda_1, \lambda_2 = 1, 0$
50	0.9367	0.9367	0.9400	0.9400	0.9400
40	0.9333	0.9333	0.9367	0.9367	0.9367
30	0.9333	0.9300	0.9367	0.9400	0.9400
20	0.9333	0.9333	0.9400	0.9400	0.9400
10	0.9233	0.9333	0.9433	0.9400	0.9267

Table 3. The classification accuracy on Yale B dataset with $p = 2$.

Dim	$\lambda_1, \lambda_2 = 0, 1$	$\lambda_1, \lambda_2 = 0.2, 0.8$	$\lambda_1, \lambda_2 = 0.5, 0.5$	$\lambda_1, \lambda_2 = 0.8, 0.2$	$\lambda_1, \lambda_2 = 1, 0$
1000	0.5527	0.5222	0.5752	0.5405	0.5924
900	0.6023	0.6046	0.5434	0.6695	0.6443
800	0.5297	0.5846	0.6350	0.6665	0.5814
700	0.5246	0.5890	0.5717	0.6282	0.5917
600	0.5477	0.5587	0.6095	0.6339	0.6682

Table 4. The classification accuracy on caltech101_silhouettes_16_uni dataset with $p = 2$.

Dim	$\lambda_1, \lambda_2 = 0, 1$	$\lambda_1, \lambda_2 = 0.2, 0.8$	$\lambda_1, \lambda_2 = 0.5, 0.5$	$\lambda_1, \lambda_2 = 0.8, 0.2$	$\lambda_1, \lambda_2 = 1, 0$
240	0.7180	0.7246	0.7213	0.7235	0.7191
210	0.6998	0.6730	0.7250	0.7179	0.7254
170	0.6619	0.7534	0.7285	0.6776	0.6863
120	0.6664	0.7379	0.7531	0.7254	0.7201
80	0.7155	0.7131	0.7052	0.7209	0.6765

Table 2,3 and 4 illustrate that the classification accuracies vary slightly with the change of λ_1 and λ_2. This is to say that different values of λ_1 and λ_2 have little influence on the extraction of discriminate

features in most cases, but the structure of feature vectors in the projection space are different. Maybe there are not important for the high-dimensional datasets, whereas for the 3D datasets the visualization results in 2D space are fully different. As shown in Figure 8, the classification accuracies using SNE and our method are both high, but the distributions of the projection points have great difference.

On the other hand, SVM is used as the classifier in this subsection. 80% of the data are used for training and the remaining data are used for testing, while the projection points in the other experiments are classified using AP algorithm. The experimental results are shown in Table 2,3 and 4 with the parameters $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$, and the corresponding classification accuracy values of our methods in Figure 12(b),13(b) and 14(b) are different. The reason is that the projection points are classified using different classifiers, which implies that the classifier has effect on the classification accuracy.

4.4. Computational efficiency

To evaluate the computational complexity of our method, we conduct experiments to compare the running time of each method. Table 5 records the running time of each method on three high-dimensional datasets. For each method, we choose the optimal parameter setting in terms of optimal results. The results of each method on the three datasets are repeated 10 independent times, and the mean results are reported. The calculations are executed on a Linux server with 32 processors (2.10GHz for each) and 64 GB RAM memory by MATLAB implementations.

Table 5 illustrates that the comparative methods, such as LDA, MDS and PCA, run faster than the proposed method, while the computational efficiency of these methods based on SNE is lower. In our method, data partition is performed before the solution of objective function, and the calculation of Geodesic distance and clustering of the weighted points occupy most of the time.

Table 5. The running time of different methods on high-dimensional datasets (in seconds).

	Dataset		
Method	control_uni	YaleB	caltech101_silhouettes_16_uni
FA	0.1873	90.8678	16.3910
LDA	0.0089	1.6779	0.0580
LPP	0.0476	2.6535	12.14476
MDS	0.0018	0.1924	0.0405
PCA	0.0013	0.1888	0.0378
PPCA	5.7341	131.0023	167.1789
Kt-SNE	6.4335	25.3993	66.5125
SNE	104.1011	361.7535	114.2378
SSNE	99.7138	396.2319	164.1256
Ours	0.2604	29.2127	16.3784

Floyd-Warshall algorithm is adopted to construct the graph to calculate the Geodesic distance. Though there are algorithms with a better worst-case runtime than Floyd-Warshall algorithm, but these algorithms are much more complicated and involve complicated data structure. Therefore, Floyd-Warshall algorithm with time complexity is $O(n^3)$ is still the best choice in our experiment. Compared with AP algorithm, the weighted AP algorithm adds virtual points in the dataset and has

the same the time complexity $O(n^3)$ in each iteration. To reduce the running time, we optimize the data structure and algorithm process. In short, some methods are faster than our method, whereas our method achieves a significant improvement in the classification accuracy and the visualization results.

5. Conclusion

Since the importance of data is different in many applications, we partition them into representative data and common data. Two structure preserving methods are adopted to reduce the dimension. For the representative data, the projection preserves the sparse neighborhood relationship and the Geodesic distance. According to the distribution of the representative data, the position of the common data can be located through the linear combination. The proposed method reduces the distortion in the process of the dimension reduction and extracts more discriminative information. In the future work, we will focus on the dimension reduction of weighted points with large-scale size.

Acknowledgments

The work is partially supported by the National Natural Science Foundation of China (Nos. 61702310, 61803237, 61772322, U1836216), the major fundamental research project of Shandong, China (No. ZR2019ZD03), and the Taishan Scholar Project of Shandong, China (No. ts20190924).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. F. Shang, H. Zhang, J. Sun, et al. *Semantic Consistency Cross-Modal Dictionary Learning with Rank Constraint*, J. Vis. Commun. Image R., **62** (2019), 259–266.
2. F. Shang, H. Zhang, L. Zhu, et al. *Adversarial Cross-Modal Retrieval Based on Dictionary Learning*, Neurocomputing, **355** (2019), 93–104.
3. D. Yang, X. Li, J. Qiu, *Output tracking control of delayed switched systems via state-dependent switching and dynamic output feedback*, Nonlinear Anal-Hybri., **32** (2019), 294–305.
4. X. Yang, X. Li, Q. Xi, et al. *Review of stability and stabilization for impulsive delayed systems*, Math. Biosci. Eng., **15** (2018), 1495–1515.
5. X. Li, J. Shen, R. Rakkiyappan, *Persistent impulsive effects on stability of functional differential equations with finite or infinite delay*, Appl. Math. Comput., **329** (2018), 14–22.
6. D. Yang, X. Li, J. Shen, et al. *State-dependent switching control of delayed switched systems with stable and unstable modes*, Math. Method. Appl. Sci., **41** (2018), 6968–6983.
7. R. Agarwal, M. P. Yadav, D. Baleanu, et al.*Existence and uniqueness of miscible flow equation through porous media with a non-singular fractional derivative*, AIMS Mathematics, **5** (2020), 1062–1073.

8. J. Jonnalagadda, B. Debananda, *Lyapunov-type inequalities for Hadamard type fractional boundary value problems*, AIMS Mathematics, **5** (2020), 1127–1146.
9. S. Wold, *Principal component analysis*, Chemometr. Intell. Lab., **2** (1987), 37–52.
10. M. Li, B. Yuan, *2D-LDA: A statistical linear discriminant analysis for image matrix*, Pattern Recogn. Lett., **26** (2005), 527–532.
11. G. Shao, N. Sang, *Regularized max-min linear discriminant analysis*, Pattern Recogn., **66** (2017), 353–363.
12. S. Wang, J. Lu, X. Gu, et al. *Semi-supervised linear discriminant analysis for dimension reduction and classification*, Pattern Recogn., **57** (2016), 179–189.
13. T. M. Rassias, *Properties of Isometric Mappings*, J. Math. Anal. Appl., **235** (1999), 108–121.
14. F. Kuang, W. Xu, S. Zhang, *A novel hybrid KPCA and SVM with GA model for intrusion detection*, Appl. Soft Comput., **18** (2014), 178–184.
15. Y. Liu, T. Chen, Y. Yao, *Nonlinear process monitoring and fault isolation using extended maximum variance unfolding*, J. Process Contr., **24** (2014), 880–891.
16. X. He, P. Niyogi, *Locality preserving projections*, In: *Advances in neural information processing systems*, 2004, 153–160.
17. L. Zhang, L. Qiao, S. Chen, *Graph-optimized locality preserving projections*, Pattern Recogn., **43** (2010), 1993–2002.
18. W. Xu, C. Luo, A. Ji, et al. *Coupled locality preserving projections for cross-view gait recognition*, Neurocomputing, **224** (2017), 37–44.
19. S. Huang, L. Zhuang, *Exponential discriminant locality preserving projection for face recognition*, Neurocomputing, **208** (2016), 373–377.
20. S. B. Chen, J. Wang, C. Y. Liu, et al. *Two-dimensional discriminant locality preserving projection based on l1-norm Maximization*, Pattern Recogn. Lett., **87** (2017), 147–154.
21. X. He, D. Cai, S. Yan, et al. *Neighborhood preserving embedding*, Tenth IEEE International Conference on Computer Vision, 2005, 1208–1213.
22. D. Cai, X. He, K. Zhou, et al. *Locality sensitive discriminant analysis*, In: *IJCAI International Joint Conference on Artificial Intelligence*, 2007, 708–713.
23. D. Xu, S. Yan, D. Tao, et al. *Marginal Fisher analysis and its variants for human gait recognition and content-based image retrieval*, IEEE T. Image Process., **16** (2007), 2811–2821.
24. H. Wang, L. Feng, L. Yu, et al. *Multi-view sparsity preserving projection for dimension reduction*, Neurocomputing, **216** (2016), 286–295.
25. L. Qiao, S. Chen, X. Tan, *Sparsity preserving projections with applications to face recognition*, Pattern Recogn., **43** (2010), 331–341.
26. L. Weng, F. Dornaika, Z. Jin, *Flexible constrained sparsity preserving embedding*, Pattern Recogn., **60** (2016), 813–823.
27. Q. Gao, Y. Huang, H. Zhang, et al. *Discriminative sparsity preserving projections for image recognition*, Pattern Recogn., **48** (2015), 2543–2553.

28. W. Cai, *A dimension reduction algorithm preserving both global and local clustering structure*, Knowl-Based. Syst., **118** (2017), 191–203.

29. S. Bao, L. Luo, J. Mao, et al. *Improved fault detection and diagnosis using sparse global-local preserving projection*, J. Process Contr., **47** (2016), 121–135.

30. L. Liu, B. Zhang, H. Zhang, et al. *Graph steered discriminative projections based on collaborative representation for Image recognition*, Multimed. Tools Appl., **78** (2019), 24501–24518.

31. G. E. Hinton, R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science, **313** (2006), 504–507.

32. D. Wang, J. Gu, *VASC: Dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder*, Genom. Proteom. Bioinf., **16** (2018), 320–331.

33. B. J. Frey, D. Dueck, *Clustering by passing messages between data points*, Science, **315** (2007), 972–976.

34. G. Wen, L. Jiang, J. Wen, *Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding*, Pattern Recogn., **41** (2008), 2226–2236.

35. G. Shamai, R. Kimmel, *Geodesic distance descriptors*, In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 6410–6418.

36. Y. Fan, L. Hou, K. X. Yan, *On the density estimation of air pollution in Beijing*, Econ. lett., **163** (2018), 110–113.

37. B. He, Z. Lin, Y. F. Li, *An automatic registration algorithm for the scattered point clouds based on the curvature feature*, Opt. Laser Technol., **46** (2013), 53–60.

38. M. U. Ali, S. Ahmed, J. Ferzund, et al. *Using PCA and Factor Analysis for dimensionality reduction of Bio-informatics Data*, IJACSA., **8** (2017), 415–426.

39. G. Nyamundanda, L. Brennan, I. C. Gormley, *Probabilistic principal component analysis for metabolomic data*, BMC Bioinformatics, **11** (2010), 571.

40. R. Sharifi, R. Langari, *Nonlinear sensor fault diagnosis using mixture of probabilistic PCA models*, Mech. Syst. Signal Pr., **85** (2017), 638–650.

41. K. Bunte, S. Haase, M. Biehl, et al. *Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences*, Neurocomputing, **90** (2012), 23–45.

42. A. Gisbrecht, A. Schulz, B. Hammer, *Parametric nonlinear dimensionality reduction using kernel t-SNE*, Neurocomputing, **147** (2015), 71–82.

43. J. A. Cook, I. Sutskever, A. Mnih, et al. *Visualizing similarity data with a mixture of maps*, PMLR., **2** (2007), 67–74.

AIMS Press