# Manifold Partition Discriminant Analysis

Yang Zhou and Shiliang Sun

**Abstract**—We propose a novel algorithm for supervised dimensionality reduction named Manifold Partition Discriminant Analysis (MPDA). It aims to find a linear embedding space where the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold, while nearby data belonging to different classes are well separated. By partitioning the data manifold into a number of linear subspaces and utilizing the first-order Taylor expansion, MPDA explicitly parameterizes the connections of tangent spaces and represents the data manifold in a piecewise manner. While graph Laplacian methods capture only the pairwise interaction between data points, our method capture both pairwise and higher order interactions (using regional consistency) between data points. This manifold representation can help to improve the measure of within-class similarity, which further leads to improved performance of dimensionality reduction. Experimental results on multiple real-world data sets demonstrate the effectiveness of the proposed method.

**Index Terms**—Discriminant Analysis, Supervised Learning, Manifold Learning, Tangent Space

## 1 Introduction

Linear Discriminant Analysis (LDA) is a classical supervised dimensionality reduction method. It aims to find an optimal low-dimensional projection along which data points from different classes are far away from each other, while those belonging to the same class are as close as possible. In the resultant low-dimensional space, the performance of classifiers could be improved. Because of this, LDA is especially useful for classification tasks. Due to its effectiveness, LDA is widely employed in different applications such as face recognition and information retrieval [1], [2], [3], [4]. However, when the input data are multimodal or mainly characterized by their variances, LDA cannot perform very well. This is caused by the assumption implicitly adopted by LDA that data points belonging to each class are generated from multivariate Gaussian distributions with the same covariance matrix but different means. If data are formed by several separate clusters or lie on a manifold, this assumption is violated, and thus LDA obtains undesired results.

To solve this problem, some extensions of LDA have been proposed, which resort to discovering local data structures. Marginal Fisher Analysis (MFA) [5] aims to gather the nearby examples of the same class, and separate the marginal examples belonging to different classes. Locality Sensitive Discriminant Analysis (LSDA) [6] maps data points into a subspace where the examples with the same label at each local area are close, while the nearby examples from different classes are apart from each other. Local Fisher Discriminant Analysis (LFDA) [7] also focuses on discovering local data structures. It can be viewed as performing LDA on the local area around each data point. LFDA is a very effective algorithm and has many applications. Recently, LFDA (combined with PCA) was applied to the pedestrian re-identification problem and achieved the state-of-the-art performance [8]. Despite of different names and motivations, these methods, in fact, fall into the same graph Laplacian based framework. All of them employ the Laplacian matrix on specific graphs to characterize data structures locally, and share the same idea that if nearby examples $x_i, x_j$ have the same class label $y$, they should be projected as close as possible, otherwise, they should be well separated. By exploiting the local structures around each data point, they are able to process the data on which LDA cannot achieve reasonable results. As widely recognized, graphs are often used as a proxy for the manifold. Therefore, these methods, to some extent, can be viewed as the combinations of manifold learning and LDA.

Although the above methods overcome the drawback of LDA, they rely on the graph Laplacian to capture the manifold structure, where only pairwise differences are considered whereas regional consistency is ignored. The regional consistency can be characterized by tangent spaces of the data manifold, which could be very useful to enhance the performance of discriminant analysis in some situations [9] [10]. Moreover, the definition of closeness of these graph Laplacian based methods is rather vague. Along which direction can we decide if the closeness of the mapped data points is achieved? We advocate that in order to preserve the manifold structure as much as possible, the closeness of the embeddings should be achieved along the direction that is consistent with the local variation of the data manifold.

During recent years, tangent space based methods have received considerable interest in the area of man-

• *Yang Zhou and Shiliang Sun (corresponding author) are with Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, P. R. China (email: shiliangsun@gmail.com, slsun@cs.ecnu.edu.cn)*

*Manuscript received Dec. 21, 2014; revised Apr. 2, 2015, Sep. 10, 2015 and Jan. 21, 2016; accepted Feb. 9, 2016.*

ifold learning [10], [11], [12], [13]. They utilize tangent spaces to estimate and extract the topological and geometrical structure of the underlying manifold. Local Tangent Space Alignment (LTSA) [11] constructs tangent spaces at each data point and then aligns them to obtain a global coordinate through minimizing the reconstruction error. Similar to LTSA, Manifold Charting [12] tries to unfold the manifold by aligning local charts. Tangent Space Intrinsic Manifold Regularization (TSIMR) [10] estimates a local linear function on the manifold which has constant manifold derivatives. Parallel Vector Field Embedding (PFE) [13] represents a function along the manifold from the perspective of vector fields and requires the vector field at each data point to be as parallel as possible. Due to exploiting the regional consistency reflected by tangent spaces, these tangent space based methods work well for representing the manifold structure. However, because of their unsupervised nature, they have no ability to capture the discriminative information from class labels, and thus are not optimal for supervised dimensionality reduction. Then how should we utilize the regional consistency of tangent spaces to improve the performance of supervised dimensionality reduction?

Besides the methods mentioned above, there are many other works that have been done in the field of dimensionality reduction. Supervised Local Subspace Learning ($SL^2$) [14] learns a mixture of local tangent spaces that are robust to under-sampled regions for continuous head pose estimation, so that it can avoid overfitting and be robust to noise. Linear Spherical Discriminant Analysis (LSDA) [15] performs discriminant analysis based on the cosine distance metric to improve speaker clustering performance. By building a sparse projection matrix for dimension reduction, Double Shrinking Algorithm (DSA) [16] compresses image data on both dimensionality and cardinality to obtain better embedding or classification performance. Least-Squares Dimension Reduction (LSDR) [17] adopts a squared-loss variant of mutual information as a dependency measure to perform sufficient dimensionality reduction. Wang et al. proposed an exponential framework for dimensionality reduction [18]. By using matrix exponential to measure data similarity, this framework emphasizes small distance pairs, and can avoid the small sample size problem. Although all of these methods have their own merits, none of them solves the above mentioned two problems.

In this paper, we propose a novel supervised dimensionality reduction method called Manifold Partition Discriminant Analysis (MPDA), which solves the above two problems. In MPDA, pairwise differences and piecewise regional consistency are considered simultaneously, so that the manifold structure can be well preserved. MPDA aims to find a linear embedding space where the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold, while nearby data belonging to different classes are well separated. Compared with existing methods, MPDA has several desirable properties that should be highlighted:

- MPDA partitions the data manifold into a number of non-overlapping linear subspaces and discovers regional manifold structures in a piecewise manner.
- With the partitioned manifold, MPDA is able to construct tangent spaces with varied numbers of dimensions. This provides MPDA with more flexibility to handle non-uniformly distributed or complex data.
- By using the first-order Taylor expansion, MPDA establishes a manifold representation which is characterized by both the pairwise differences and piecewise regional consistency of the underlying manifold.
- Thanks to the proposed manifold representation, MPDA improves the measure of within-class similarity, and is able to obtain a projection that is consistent with the local variation of the underlying manifold.

The rest of this paper is organized as follows. In Section 2, we briefly introduce the graph Laplacian based framework, under which many supervised dimensionality reduction methods can be considered within the same category. Then the Manifold Partition Discriminant Analysis (MPDA) algorithm is presented in Section 3. Section 4 discusses the connection and difference between MPDA and related works. In Section 5, MPDA is tested on multiple real-world data sets compared with existing supervised dimensionality reduction algorithms. Finally, we give concluding remarks in Section 6.

## 2 Graph Laplacian Based Framework for Discriminant Analysis

Representing data on a specific graph is a popular way to characterize the relationships among data points. Given an undirected weighted $G = \{X, W\}$ with a vertex set X and a symmetric weight matrix $W \in \mathbb{R}^{n\times n}$, these relationships can be easily characterized by $G$, where each example serves as a vertex of $G$, and $W$ records the weight on the edge of each pair of vertices. Generally, if two examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close", the corresponding weight $W_{ij}$ is large, whereas if they are "far away", then the $W_{ij}$ is small. Provided a certain $W$, the intrinsic geometry of graph $G$ can be represented by the Laplacian matrix [19], which is defined as

$$L = D - W, \tag{1}$$

where $D$ is a diagonal matrix with the $i$-th diagonal element being $D_{ii} = \sum_{j\neq i} W_{ij}$. The Laplacian matrix is capable of representing certain geometry of data according to a specific weight matrix. This property is very helpful for developing dimensionality reduction methods.

Let $X$ be a data set consisting of $n$ examples and labels, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes a $d$-dimensional

example, $y_i \in \{1, 2, \ldots, C\}$ denotes the class label corresponding to $\boldsymbol{x}_i$, and $C$ is the total number of classes. Classical LDA aims to find an optimal linear projection $\boldsymbol{t}$ along which the between-class scatter is maximized and the within-class scatter is minimized [20]. The objective function of LDA can be written as:

$$\boldsymbol{t}^* = \arg\max_{\boldsymbol{t}} \frac{\boldsymbol{t}^\top S_b \boldsymbol{t}}{\boldsymbol{t}^\top S_w \boldsymbol{t}}, \tag{2}$$

where $\top$ denotes the transpose of a matrix or a vector, $S_b$ and $S_w$ denote the between-class and within-class scatter matrices, respectively. The definitions of $S_b$ and $S_w$ are given as follows:

$$S_b = \sum_{c=1}^{C} n_c (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^\top, \tag{3}$$

$$S_w = \sum_{c=1}^{C} \sum_{\{i|y_i=c\}} (\boldsymbol{x}_i - \boldsymbol{\mu}_c)(\boldsymbol{x}_i - \boldsymbol{\mu}_c)^\top, \tag{4}$$

where $n_c$ is the number of data from the $c$-th class, $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$ is the mean of all the data points, and $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{\{i|y_i=c\}} \boldsymbol{x}_i$ is the mean of the data from class $c$. Apart from the above formulations, $S_b$ and $S_w$ can also be formulated via the graph Laplacian [7]:

$$S_b = \sum_{ij} W_{ij}^b \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = 2XL^bX^\top, \tag{5}$$

$$S_w = \sum_{ij} W_{ij}^w \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = 2XL^wX^\top, \tag{6}$$

where $L^w$ and $L^b$ are the Laplacian matrices constructed by the weight matrices $W^w$ and $W^b$ with

$$W_{ij}^b = \begin{cases} (1/n - 1/n_c) & if\ y_i = y_j = c \\ 1/n & if\ y_i \neq y_j, \end{cases}$$

$$W_{ij}^w = \begin{cases} 1/n_c & if\ y_i = y_j = c \\ 0 & if\ y_i \neq y_j. \end{cases}$$

The objective function (2) can be converted to a generalized eigenvalue problem:

$$XL^wX^\top \boldsymbol{t} = \lambda XL^bX^\top \boldsymbol{t} \tag{7}$$

whose solution can be easily given by the eigenvector with respect to the largest eigenvalue. From the above formulations, it is clear that the graph Laplacian plays a key role in deriving LDA, where the weight matrices $W^b$ and $W^w$ measure the similarity of each pair of data points, and their characteristics varies as the criterion of similarity changes. This provides a general and flexible framework to develop new dimensionality reduction algorithms by constructing appropriate Laplacian matrices.

In order to improve the performance of LDA, many local structure based extensions of LDA have been proposed in the recent decades. Representative methods include Marginal Fisher Analysis (MFA) [5], Locality Sensitive Discriminant Analysis (LSDA) [6], Local Fisher Discriminant Analysis (LFDA) [7], etc. Unlike traditional LDA, they compute the between-class and within-class scatter based on local data structures rather than the global mean values. Although these methods improve the performance of discriminant analysis by solving the problem caused by the improper assumption adopted by LDA, none of them extends beyond the graph Laplacian based framework. Their differences merely lie in the different ways of constructing the Laplacian matrices $L^b$ and $L^w$.

In spite of its effectiveness, the graph Laplacian based framework still has several limitations. The between-class and within-class scatter are computed by only aggregating all pairwise differences between data points across the entire graph, whereas the regional consistency, which is reflected by the regional structure around a local area of the underlying manifold, is ignored. Moreover, by minimizing the aggregation of within-class data pairs (6), the objective function (2) tends to find a direction along which some "averaged" within-class similarity is achieved. However, it is unclear that how the "averaged" similarity can precisely reflect the topological and geometrical structure of the underlying manifold.

## 3 Manifold Partition Discriminant Analysis

In this section, we propose a novel supervised dimensionality reduction algorithm named Manifold Partition Discriminant Analysis (MPDA). Unlike previous methods that mainly rely on the graph Laplacian [5], [6], [7], MPDA exploits both pairwise differences and piecewise regional consistency to preserve the manifold structure. It aims to find a linear embedding space where the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold, while nearby data belonging to different classes are well separated. To this end, we first need to extract the piecewise consistency from the data manifold, which can be achieved by partitioning the data manifold into non-overlapping pieces, and estimating tangent spaces for each piece. Then we can represent the data manifold by combining pairwise differences with piecewise consistency. The resultant manifold representation is able to characterize the local variation of the data manifold, and improve the measure of within-class similarity, which eventually leads to the MPDA algorithm. Specifically, we mainly solve the following problems:

- *P1* How to partition the data manifold into a number of non-overlapping pieces, and estimate an accurate tangent space?
- *P2* How to combine pairwise differences with piecewise regional consistency in representing the data manifold?
- *P3* How to find a linear subspace where the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold?

Next, we first solve *P2* and *P3* in Section 3.1 and 3.2, respectively, and defer the treatment of *P1* to Section 3.3.

### 3.1 Manifold Representation

In order to combine pairwise differences with piecewise regional consistency in representing the data manifold, we are interested in estimating a function $f$ defined on an $m$-dimensional smooth manifold $\mathcal{M}$, where $\mathcal{M}$ is embedded in $\mathbb{R}^d$. This function $f$ can serve as a direct connection between the data representation in $d$ and $m$-dimensional spaces. For simplicity, we first consider to represent data in a one-dimensional Euclidean space $\mathbb{R}$. Define $f : \mathbb{R}^d \to \mathbb{R}$ as a function along the manifold $\mathcal{M}$. Let $\mathcal{T}_{\boldsymbol{x}_0}\mathcal{M}$ be the tangent space of $\boldsymbol{x}_0$ on $\mathcal{M}$, where $\boldsymbol{x}_0 \in \mathbb{R}^d$ is a single point on the manifold $\mathcal{M}$. According to the first-order Taylor expansion at $\boldsymbol{x}_0$, $f$ can be expressed as follows [10], [13], [21]:

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \boldsymbol{v}_{\boldsymbol{x}_0}^\top \boldsymbol{u}_{\boldsymbol{x}_0}(\boldsymbol{x}) + O(\|\boldsymbol{x} - \boldsymbol{x}_0\|^2),$$

where $\boldsymbol{u}_{\boldsymbol{x}_0}(\boldsymbol{x}) = T_{\boldsymbol{x}_0}^\top(\boldsymbol{x} - \boldsymbol{x}_0)$ is an $m$-dimensional vector which gives a representation of $\boldsymbol{x}$ in the tangent space $\mathcal{T}_{\boldsymbol{x}_0}\mathcal{M}$. $T_{\boldsymbol{x}_0} \in \mathbb{R}^{d\times m}$ is a matrix formed by the orthonormal bases of $\mathcal{T}_{\boldsymbol{x}_0}\mathcal{M}$, and characterizes the regional consistency of the manifold structure around $\boldsymbol{x}_0$. Generally, $T_{\boldsymbol{x}_0}$ can be estimated by performing PCA on the neighborhood of $\boldsymbol{x}_0$ [11], [22]. $\boldsymbol{v}_{\boldsymbol{x}_0}$ is an $m$-dimensional tangent vector and represents the manifold derivative of $f$ at $\boldsymbol{x}_0$ with respect to $\boldsymbol{u}_{\boldsymbol{x}_0}(\boldsymbol{x})$, which reflects the local variation of the manifold at $\boldsymbol{x}_0$.

Given two nearby data points $\boldsymbol{z}$ and $\boldsymbol{z}'$ lying on the manifold $\mathcal{M}$, we can use the first-order Taylor expansion at $\boldsymbol{z}'$ to express $f(\boldsymbol{z})$ as follows:

$$f(\boldsymbol{z}) = f(\boldsymbol{z}') + \boldsymbol{v}_{\boldsymbol{z}'}^\top T_{\boldsymbol{z}'}^\top(\boldsymbol{z} - \boldsymbol{z}') + O(\|\boldsymbol{z} - \boldsymbol{z}'\|^2). \quad (8)$$

If $\mathcal{M}$ is smooth enough, the second-order derivatives of $f$ tend to vanish. Furthermore, when $\boldsymbol{z}$ and $\boldsymbol{z}'$ are close to each other, $\|\boldsymbol{z} - \boldsymbol{z}'\|^2$ becomes very small. Therefore, the remainder in (8) can be omitted, which leads to:

$$f(\boldsymbol{z}) \approx f(\boldsymbol{z}') + \boldsymbol{v}_{\boldsymbol{z}'}^\top T_{\boldsymbol{z}'}^\top(\boldsymbol{z} - \boldsymbol{z}'). \quad (9)$$

With the above results, it is clear that for any nearby data points $\boldsymbol{z}$ and $\boldsymbol{z}'$ lying on the manifold $\mathcal{M}$, the low-dimensional embeddings $f(\boldsymbol{z})$ and $f(\boldsymbol{z}')$ should satisfy (9), and the difference between both sides of (9) should be as small as possible. This can serve as a good criterion to preserve the manifold structure, which establishes the connection between each pair of nearby data points.

Assume that the data manifold can be well approximated by the union of a number of non-overlapping linear subspaces. In this case, each linear subspace can serve as a tangent space, and each tangent space has a tangent vector. With the partitioned manifold, we are able to construct tangent spaces and tangent vectors for each linear subspace rather than each data point. If $\boldsymbol{z}'$ lies in a tangent space $\mathcal{T}_p\mathcal{M}$ with a tangent vector $\boldsymbol{v}_p$, (9) becomes:

$$f(\boldsymbol{z}) \approx f(\boldsymbol{z}') + \boldsymbol{v}_p^\top T_p^\top(\boldsymbol{z} - \boldsymbol{z}'), \quad (10)$$

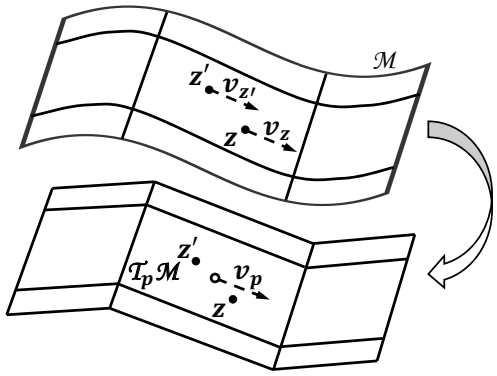



Fig. 1. Conceptual illustration of the manifold partition strategy.

where $T_p$ is estimated by performing PCA on the data falling into $\mathcal{T}_p\mathcal{M}$. This can be justified by the fact that the manifold derivative of a linear subspace is a constant function. This means that for the data falling into the same linear subspace, their corresponding tangent vectors are equal, and can be represented by only one tangent vector $\boldsymbol{v}_p$. It is worth noting that since PCA entails mean subtraction, each tangent space estimated by PCA will have a separate mean. This seems to cause the discrepancy of tangent spaces. However, this discrepancy is not a problem in our case. Once the orthonormal basis $T_p$ has been estimated, the effect of mean subtraction is just to center data to the origin of the corresponding subspace. Notice that only the data falling into $\mathcal{T}_p\mathcal{M}$ or those around $\mathcal{T}_p\mathcal{M}$ are involved in the projection of $T_p$. These data points implicitly reflect the mean of the corresponding subspace. Therefore, we can directly use the orthonormal basis $T_p$ to compute the projection without mean subtraction. Figure 1 illustrates the concept of the above strategy (we call it the manifold partition). Intuitively, after partitioning the manifold, $\mathcal{M}$ is approximated by the union of the linear subspaces, where each linear subspace serves as a tangent space. Therefore, (10) combines pairwise differences with piecewise regional consistency in representing the data manifold.

### 3.2 The MPDA Algorithm

Based on the above results, we propose our MPDA algorithm. Consider a data set $X = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ belonging to $C$ classes where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \ldots, C\}$ is the class label associated with the data point $\boldsymbol{x}_i$. Generally, we assume that data in different classes are generated from different manifolds. Provided that $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} = \bigcup_{p=1}^P X^p$ has been partitioned into $P$ patches, where data of each patch have the same class label, and we have obtained the orthonormal basis matrices $\{T_p\}_{p=1}^P$ of tangent spaces for each data patch. Our goal is to find an embedding space where the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold, while nearby data

belonging to different classes are well separated.

In order to gather within-class data based on the manifold structure, we first construct the within-class graph $G = \{X, W\}$ to represent the geometry of the data manifold. If $\boldsymbol{x}_i$ is among the $k$-nearest neighbors of $\boldsymbol{x}_j$ with $y_i = y_j$, an edge connecting $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is added with the weight $W_{ij} = W_{ji} = 1$. If there is no edge connecting $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$, $W_{ij} = 0$. With the results in Section 3.1, for each pair of nearby within-class data points, we can obtain:

$$f(\boldsymbol{x}_i) \approx f(\boldsymbol{x}_j) + \boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{11}$$

$$f(\boldsymbol{x}_j) \approx f(\boldsymbol{x}_i) + \boldsymbol{v}_{\pi_i}^\top T_{\pi_i}^\top (\boldsymbol{x}_j - \boldsymbol{x}_i). \tag{12}$$

We require the difference between both sides of (11) to be as small as possible. In the scenario of linear dimensionality reduction, $f(\boldsymbol{x})$ represents a one-dimensional embedding of $\boldsymbol{x}$, and we aim to find a linear projection. To this end, $f(\boldsymbol{x})$ is further approximated as a linear function $f(\boldsymbol{x}) = \boldsymbol{t}^\top \boldsymbol{x}$ where $\boldsymbol{t} \in \mathbb{R}^d$ is a linear projection vector. Then, if nearby data points $\boldsymbol{x}_i, \boldsymbol{x}_j$ belong to the same class, we can measure their similarity as follows:

$$\begin{aligned} &\big(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j) - \boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2 \\ =&\big(\boldsymbol{t}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) - \boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2, \end{aligned} \tag{13}$$

where $\pi_i \in \{1, \ldots, P\}$ is an index indicating the patch $\boldsymbol{x}_i$ belongs to. Moreover, we also need to measure the similarity between nearby tangent spaces. By substituting (11) into (12), we have:

$$(T_{\pi_j} \boldsymbol{v}_{\pi_j} - T_{\pi_i} \boldsymbol{v}_{\pi_i})^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) \approx 0.$$

From the above equation, we know that the two vectors are approximately perpendicular or the row vector $(T_{\pi_j} \boldsymbol{v}_{\pi_j} - T_{\pi_i} \boldsymbol{v}_{\pi_i})^\top$ approximately equals to a zero vector. However, the perpendicular case can not be satisfied for every pair of nearby data points on the manifold. For instance, consider there are three nearby data points on the manifold. Each pair of them should satisfy the above equation, while only two of them are, in general, justified in the perpendicular case. On the other side, the case of zero row vectors can be justified for all the data pairs, and leads to $T_{\pi_j} \boldsymbol{v}_{\pi_j} \approx T_{\pi_i} \boldsymbol{v}_{\pi_i}$. Finally, by multiplying both sides of this equation with $T_{\pi_i}^\top$ and using $T_{\pi_i}^\top T_{\pi_i} = I$, it follows that:

$$\boldsymbol{v}_{\pi_i} \approx T_{\pi_i}^\top T_{\pi_j} \boldsymbol{v}_{\pi_j}. \tag{14}$$

It is clear that for each pair of nearby tangent spaces the difference between both sides of (14) should be as small as possible. Therefore, the similarity between nearby tangent spaces can be measured as follows:

$$\|\boldsymbol{v}_{\pi_i} - T_{\pi_i}^\top T_{\pi_j} \boldsymbol{v}_{\pi_j}\|_2^2. \tag{15}$$

With the above results, the data manifold with respect to each class can be estimated by relating data with a discrete weight $W_{ij}$, which leads to an objective function as follows:

$$\begin{aligned} \min_{\boldsymbol{t}, \boldsymbol{v}} \sum_{i,j}^{n} W_{ij} \Big[ &\big(\boldsymbol{t}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) - \boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2 \\ &+ \gamma \|\boldsymbol{v}_{\pi_i} - T_{\pi_i}^\top T_{\pi_j} \boldsymbol{v}_{\pi_j}\|_2^2 \Big], \end{aligned} \tag{16}$$

where $\gamma$ is a trade-off parameter controlling the influence between (13) and (15). It is clear that if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class and fall into the same tangent space, their similarity only depends on their pairwise difference. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same class but lie in different tangent spaces, apart from the pairwise difference, their similarity also depends on the angle between $\boldsymbol{v}_{\pi_j}$ and $T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)$, which means that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be viewed as similar data points when $\boldsymbol{v}_{\pi_j}$ and $T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)$ have similar directions. Since $\boldsymbol{v}_{\pi_j}$ reflects the varying direction of the data manifold around $\boldsymbol{x}_j$, by optimizing (16), we can deem that the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold.

It is worth noting that the above derivation is based on the first-order Taylor expansion of the function $f$. If we employ the zero-order Taylor expansion, the terms related to $\boldsymbol{v}_{\pi_i}$ $(i = 1, \ldots, n)$ vanish. Then the objective function is simplified as follows:

$$\min_{\boldsymbol{t}} \sum_{i,j}^{n} W_{ij} \big(\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j\big)^2 = \min_{\boldsymbol{t}} 2\boldsymbol{t}^\top X L X^\top \boldsymbol{t},$$

where $L = D - W$ is the Laplacian matrix and $D$ is a diagonal matrix with the $i$-th diagonal element being $D_{ii} = \sum_{j \neq i} W_{ij}$. This formulation is identical to the graph Laplacian based within-class scatter (6). From the aspect of manifold approximations, this means that in theory the proposed method is able to approximate the underlying manifold with a smaller approximation error $O(||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)$ than the graph Laplacian whose approximation error is $O(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$. Compared with (16), the graph Laplacian based scatter fails to consider the regional consistency that is explicitly parameterized by the proposed manifold representation. Although it can implicitly reflect regional relationships by minimizing the distance between each pair of nearby data points, the graph Laplacian has no ability to capture the regional consistency which is determined by all the nearby data around a given data point. On the other hand, the proposed manifold representation is capable of preserving both the pairwise geometry and the piecewise regional consistency, and thus can capture more structural information from the data manifold than the graph Laplacian. In other word, (16) better measures the within-class similarity than the graph Laplacian based scatter (6), because it can extract the regional consistency of each tangent space, and explicitly establishes the connections among tangent spaces by estimating tangent vectors $\{\boldsymbol{v}_p\}_{p=1}^P$.

Notice that although we assume that the data manifold can be approximated by a union of piece-wise subspaces,

it does not mean that the proposed manifold representation is inferior to the graph Laplacian. To verify this, we can split the within-class objective function (16) into two parts. The first part includes the terms related to $\boldsymbol{v}$, and the second part has the other terms. The piece-wise manifold assumption only affects the first part, while the second part is still based on the generic manifold assumption. In fact, the second part of (16) is just identical to the graph Laplacian based within-class scatter. This implies that the proposed manifold representation is at least as good as, if not better than, the graph Laplacian, as each tangent vector $\boldsymbol{v}_{\pi_i}$ can be a zero vector.

To separate data in different classes, we construct a between-class graph $G' = \{X, W'\}$. If $y_i \neq y_j$, we add an edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with the weight $W'_{ij} = 1/n$. If $y_i = y_j$, the corresponding weight is set to be $W'_{ij} = A_{ij}(1/n - 1/n_c)$. $n_c$ is the number of data points from the $c$-th class, and $A_{ij}$ is a weight that indicates the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, whose definition is given as follows:

$$A_{ij} = \begin{cases} \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma_i \sigma_j}) & \text{if } i \in \mathcal{N}_k(j) \text{ or } j \in \mathcal{N}_k(i) \\ 0 & \text{else,} \end{cases}$$

where $\mathcal{N}_k(i)$ denotes the $k$-nearest neighbor set of $\boldsymbol{x}_i$, and $\sigma_i$ is heuristically set to be the distance between $\boldsymbol{x}_i$ and its $k$-th nearest neighbor. Then we can formulate the following objective function to separate nearby between-class data points:

$$\max_{\boldsymbol{t}} \sum_{i,j}^{n} W'_{ij} (\boldsymbol{t}^\top \boldsymbol{x}_i - \boldsymbol{t}^\top \boldsymbol{x}_j)^2. \tag{17}$$

The methods for constructing $G'$ have been well studied in the literature [5], [6]. Here, we employed the one in [7] because of its effectiveness in enhancing the between-class separability.

It is easy to see that (16) can be reformulated as a canonical matrix quadratic as $(\ \boldsymbol{t}^\top \quad \boldsymbol{v}^\top\ ) S (\ \boldsymbol{t}^\top \quad \boldsymbol{v}^\top\ )^\top$ where $S$ is a $(d + mP) \times (d + mP)$ positive semi-definite matrix and $\boldsymbol{v} = (\boldsymbol{v}_1^\top, \boldsymbol{v}_2^\top, \ldots, \boldsymbol{v}_P^\top)^\top$. Due to the space limitation, the detailed derivation of $S$ is provided in the supplementary material, which is a modification of the derivation of a similar quantity used in [10]. By simple algebra formulations, (17) can also be reduced to $\begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{v} \end{pmatrix}^\top \begin{pmatrix} 2XL'X^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{v} \end{pmatrix} = (\ \boldsymbol{t}^\top \quad \boldsymbol{v}^\top\ ) S' (\ \boldsymbol{t}^\top \quad \boldsymbol{v}^\top\ )^\top$, where $L'$ is the Laplacian matrix constructed by $W'$. In order to preserve the manifold structure while separating nearby between-class data points, we can optimize the objective functions (16) and (17) simultaneously, which leads to the following objective function:

$$\arg\max_{\boldsymbol{f}} \frac{\boldsymbol{f}^\top S' \boldsymbol{f}}{\boldsymbol{f}^\top (S + \alpha I) \boldsymbol{f}}, \tag{18}$$

where we have defined $\boldsymbol{f} = (\boldsymbol{t}^\top, \boldsymbol{v}^\top)^\top$, and the Tikhonov regularizer with a trade-off parameter $\alpha$ has been employed to avoid the numerical singularity of $S$.

**Algorithm 1** MPDA

**Input:**
Labeled data $\{\boldsymbol{x}_i | \boldsymbol{x}_i \in \mathbb{R}^d\}$,
Class labels $\{y_i | y_i \in \{1, 2, \ldots, C\}\}_{i=1}^n$;
Dimensionality of embedding space $m$ ($1 \leq m \leq d$);
Trade-off parameters $\gamma$ ($\gamma > 0$).
**Output:**
$d \times r$ transformation matrix $T$.

Apply certain method to partition the data in each class into a total of $P$ patches $\{X^p\}_{p=1}^P$;
**for** $p = 1$ **to** $P$ **do**
  Construct $T_p$ by performing PCA on $X^p$;
**end for**
Construct the within-class graph $G$ and the between-class graph $G'$;
Compute the eigenvectors $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_m$ of (19) with respect to the top $m$ eigenvalues;
$T = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m)$.

The optimization of (18) is achieved by solving a generalized eigenvalue problem:

$$S' \boldsymbol{f} = \lambda (S + \alpha I) \boldsymbol{f} \tag{19}$$

whose solution is the eigenvector $\boldsymbol{f}^* = (\boldsymbol{t}^{*\top}, \boldsymbol{v}^{*\top})^\top$ with respect to the largest eigenvalue. Then we can use the first part of $\boldsymbol{f}^*$ to obtain a one-dimensional embedding of any $\boldsymbol{x} \in \mathbb{R}^d$ by computing $b = \boldsymbol{t}^{*\top} \boldsymbol{x}$. If we want to project $\boldsymbol{x}$ into an $m$-dimensional subspace, we can just compute $m$ eigenvectors $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m$ corresponding to the $m$ largest eigenvalues of (19). Then the $m$-dimensional embedding $\boldsymbol{b}$ of $\boldsymbol{x}$ is computed as $\boldsymbol{b} = T^\top \boldsymbol{x}$, where $T = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m)$. Algorithm 1 gives the pseudo-code for MPDA.

### 3.3 Partitioning the Manifold

In this section, we propose a manifold partition algorithm to solve the last problem (*P1*). Since tangent spaces are linear subspaces in essence, the better the data manifold can be linearly approximated by the partitioned pieces, the more accurately the resultant tangent spaces can reflect the regional consistency of the underlying manifold. In order to estimate tangent spaces which approximately lie on the manifold surface, we first need to introduce a criterion to measure the linearity of subspaces.

Given a data set $X$ as well as its pairwise Euclidean distance matrix $D^E$ and geodesic distance matrix $D^G$ (approximated by the shortest path algorithms such as Dijkstra's algorithm), we can measure the degree of linearity between two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ by computing the ratio $R_{ij} = D^G_{ij}/D^E_{ij}$, which is also referred to as the *tortuosity* [23]. $D^E_{ij}$ is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and $D^G_{ij}$ is their geodesic distance. It is clear that $D^G_{ij}$ is never smaller than $D^E_{ij}$. If $D^G_{ij} \approx D^E_{ij}$, then $R_{ij} \approx 1$ and we can deem that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ lie on a straight line.

When it comes to a data patch $X^p$, we can measure its linearity as follows:

$$R^p = \frac{1}{N_p^2} \sum_{\boldsymbol{x}_i \in X^p} \sum_{\boldsymbol{x}_j \in X^p} R_{ij}, \tag{20}$$

where $N_p$ denotes the number of data in $X^p$. It is clear that the smaller $R^p$ is, the better the data in $X^p$ fit a linear subspace.

With the above measure of linearity, we can partition the manifold by hierarchical clustering [24]. There are mainly two branches of hierarchical clustering depending on their search strategies. In this paper, we use the top-down hierarchical divisive clustering rather than the bottom-up hierarchical agglomerative clustering because of two reasons. For one thing, if we need to partition the data set into $P$ patches, as $P$ is usually much smaller than the number of data points, top-down methods are more efficient than bottom-up ones. For another, top-down methods tend to construct patches with the same or similar sizes. As a result, the tangent spaces estimated by these patches tend to have similar dimensionalities, which fits the manifold assumption better. Specifically, given a data set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, our top-down partition algorithm aims to partition $X$ into a number of patches (subsets) until there is no patch (subset) containing more than $M$ data points, which consists of the following steps:

1) Initialize $P = 1$, $X = \{X^p\}_{p=1}^P = \{X^1\} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_1}\}$, where $N_1 = N$. Compute the Euclidean distance matrix $D^E$, the geodesic distance matrix $D^G$ (approximated by the shortest path algorithms such as Dijkstra's algorithm), and the patch linearity $R^1$ according to (20).
2) From $\{X^p\}_{p=1}^P$, select the patch $X^p$ $(p \in 1, \ldots, P)$ having the highest value of $R^p \cdot N_p$. From $X^p$, select two data points $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ having the largest geodesic distance $D_{lr}^G$. Create two new patches $X_l^p = \{\boldsymbol{x}_l\}$ and $X_r^p = \{\boldsymbol{x}_r\}$. Update $X^p \leftarrow X^p \setminus \{\boldsymbol{x}_l, \boldsymbol{x}_r\}$.
3) Construct the $k'$-nearest neighbor sets of $X_l^p$ and $X_r^p$ denoted by $\mathcal{N}_l^p$ and $\mathcal{N}_r^p$, respectively. Construct the joint neighbor set $\mathcal{N}_{joint}^p = \mathcal{N}_l^p \cap \mathcal{N}_r^p$. Update $\mathcal{N}_l^p \leftarrow \mathcal{N}_l^p \setminus \mathcal{N}_{joint}^p$, $\mathcal{N}_r^p \leftarrow \mathcal{N}_r^p \setminus \mathcal{N}_{joint}^p$.
4) Update $X_l^p \leftarrow X_l^p \cup (\mathcal{N}_l^p \cap X^p)$, $X^p \leftarrow X^p \setminus (\mathcal{N}_l^p \cap X^p)$, $X_r^p \leftarrow X_r^p \cup (\mathcal{N}_r^p \cap X^p)$, $X^p \leftarrow X^p \setminus (\mathcal{N}_r^p \cap X^p)$.
5) Compute the patch linearity $R_l^p$ and $R_r^p$ for $X_l^p$ and $X_r^p$, respectively. Let $N_l$ and $N_r$ be the number of data in $X_l^p$ and $X_r^p$. If $R_l^p \cdot N_l > R_r^p \cdot N_r$, update $X_r^p \leftarrow X_r^p \cup \mathcal{N}_{joint}^p$, or update $X_l^p \leftarrow X_l^p \cup \mathcal{N}_{joint}^p$ otherwise. Repeat steps $3) \sim 5)$ until $X^p = \varnothing$.
6) $X^p$ has been partitioned into $X_l^p$ and $X_r^p$. Update $P \leftarrow P + 1$, $X^p \leftarrow X_l^p$, $X^P \leftarrow X_r^p$. Go to step 2), until there is no patch having $N_p > M$, where $M$ is the maximum patch size.

Generally, in order to obtain the patch in which data lie in a linear subspace, we should divide the patch with the largest $R^p$ in each turn of partition. In our algorithm, we combine the patch linearity $R^p$ and its size $N_p$ together to select the patch that should be further divided, because the scope of subspaces should be small enough so that the Taylor expansion in (11) and (12) can be justified. Two parameters in the proposed partition algorithm should be determined, i.e., the neighborhood size $k'$ and the maximum patch size $M$. It is worth noting that to estimate tangent spaces accurately, each patch should satisfy two competing requirements. On the one hand, we should keep sufficient data in each patch so that the tangent space can be well estimated. On the other hand, the patch should be small enough to preserve the local manifold structure. Therefore, we use $M$ rather than the number of subspaces $P$ as the threshold to control the termination of the algorithm.

Besides extracting the piecewise regional consistency, partitioning the manifold can provide additional benefits. It is clear that an accurate estimation of tangent spaces is crucial for tangent space based methods. Usually, tangent spaces are estimated by performing PCA on the $k$-nearest neighbors of each data point. This approach fixes the neighborhood size, which may fail to estimate the correct tangent spaces when data are sampled non-uniformly or the manifold has a varying curvature. In contrast, the proposed MPDA method is more likely to get a robust estimation, because PCA is performed on the data in each linear subspace where data naturally lie on the manifold surface. Figure 2 shows an example that performing PCA on the fixed-sized neighborhood fails to capture the correct tangent space. As can be seen, $\mathcal{T}_p\mathcal{M}$ and $\mathcal{T}_{p'}\mathcal{M}$ reflect the correct manifold structure, whereas $\mathcal{T}_{\boldsymbol{z}}\mathcal{M}$ computed by $\boldsymbol{z}$ and its two-nearest neighbors is incorrect. In addition, real data are often complex whose underlying manifold dimensionality could vary at different regions. Therefore, it would be better to adjust the manifold dimensionality for different parts of the manifold instead of setting a fixed one. As the number of data varies in each linear subspace, MPDA adaptively determines the number of dimensions of each linear subspace by simply employing PCA to preserve certain percentages of energy, say 95%. This provides MPDA with more flexibility to handle complex data in practice.

### 3.4 Pairwise-variate MPDA

For now, we have presented the MPDA algorithm which considers both preserving the manifold structure and distinguishing data from different classes. Since the data manifold is partitioned into a number of non-overlapping tangent spaces, MPDA discovers the regional consistency of the data manifold in a piecewise manner, where the manifold partition strategy plays a key role in deriving MPDA. If we relax the piece-wise manifold assumption to the general one, and directly derive the proposed method from (9) rather than (10) without partitioning the manifold, we can obtain a Pairwise-variate MPDA (PMPDA). Then, the objective

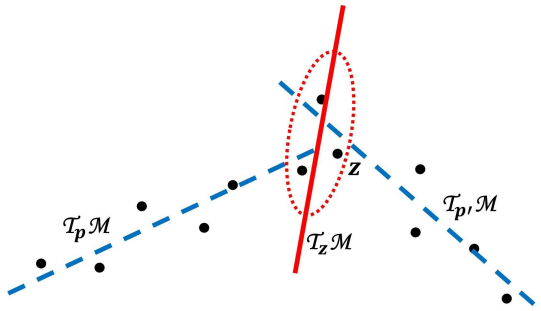



Fig. 2. An example of how performing PCA on the fixed-sized neighborhood fails to capture the correct tangent space. Dashed lines indicate the tangent spaces of two patches $X^p$ and $X_{p'}$. The dotted ellipse indicates the two-nearest neighborhood of $\boldsymbol{z}$. The solid line shows the tangent space estimated by performing PCA on the two-nearest neighborhood of $\boldsymbol{z}$.

function (16) becomes:

$$\min_{\boldsymbol{t},\boldsymbol{v}} \sum_{i,j}^{n} W_{ij} \Big[ \big(\boldsymbol{t}^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) - \boldsymbol{v}_{\boldsymbol{x}_j}^\top T_{\boldsymbol{x}_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2 + \gamma \|\boldsymbol{v}_{\boldsymbol{x}_i} - T_{\boldsymbol{x}_i}^\top T_{\boldsymbol{x}_j} \boldsymbol{v}_{\boldsymbol{x}_j}\|_2^2 \Big], \tag{21}$$

where the orthonormal basis matrix $T_{\boldsymbol{x}_i}$ is computed by performing PCA on the $k$-nearest within-class neighbors of $\boldsymbol{x}_i$.

Similar to (16), (21) can also be reformulated as a quadratic form $(\ \boldsymbol{t}^\top \quad \boldsymbol{v}^\top\ ) S_p (\ \boldsymbol{t}^\top \quad \boldsymbol{v}^\top\ )^\top$ where $S_p$ is a $(d + mn) \times (d + mn)$ positive semi-definite matrix, and $\boldsymbol{v} = (\boldsymbol{v}_{\boldsymbol{x}_1}^\top, \boldsymbol{v}_{\boldsymbol{x}_2}^\top, \ldots, \boldsymbol{v}_{\boldsymbol{x}_n}^\top)^\top$. The rest steps of PMPDA are just the same as MPDA except that $S$ is replaced by $S_p$. Finally, PMPDA solves the following generalized eigenvalue problem:

$$S'\boldsymbol{f} = \lambda(S_p + \alpha I)\boldsymbol{f}. \tag{22}$$

Compared with MPDA, PMPDA no longer needs to partition the manifold, but has to estimate tangent vectors and tangent spaces for each data point, while MPDA estimates only $P$ of them. On the one hand, PMPDA is more effective to preserve the manifold geometry, because it is based on a more general manifold assumption. On the other hand, PMPDA has to determine tangent vectors and construct tangent spaces at each data point, which not only leads to tremendous computational overheads, but results in severe storage problems when it is performed on large data sets. Consequently, PMPDA can only be performed on small data sets and is hardly practical. In brief, PMPDA is able to make a better manifold estimation at the expense of its efficiency, and the strategy of partitioning the manifold can be viewed as a trade-off between effectiveness and efficiency of the manifold estimation.

Like PMPDA, some tangent space based methods such as TSIMR [10] and PFE [13] also suffer from similar computational and storage problems. This implies that although the manifold partition strategy is a crucial part of MPDA, we can also apply it to the tangent space based methods to make them more efficient. We provide some preliminary results in the supplementary material.

### 3.5 Time Complexity

In this section, we briefly analyze the computational complexity of both PMPDA and MPDA. The main computational costs of PMPDA lie in building tangent spaces for $n$ data points and solving the generalized eigenvalue problem. PMPDA takes $O((d^2k + k^2d) \times n)$ for estimating $n$ tangent spaces by performing PCA on the $k$-nearest neighborhood of each data point. Note that we can obtain at most $k$ meaningful orthonormal bases for each tangent space, since there are only $k+1$ data points as the inputs of PCA. Therefore, the dimensionalities of tangent vectors and tangent spaces are at most $k$. This means that PMPDA takes $O((d + kn)^3)$ for solving the generalized eigenvalue problem (22).

MPDA first partitions the manifold into $P$ linear subspaces, whose time complexity is dominated by computing the geodesic distance matrix $D^G$ and the hierarchical divisive clustering for data in each class. Computing $D^G$ based on a $k'$-NN graph by the Dijkstra's algorithm with Fibonacci heaps takes $O(n^2 \log n + k'n^2/2)$. The computational complexity of the hierarchical divisive clustering can be approximated as $O(\sum_{c=1}^{C} \sum_{p=1}^{P_c} (2^p(n_c/2^p))^2) \approx O(\sum_{c=1}^{C} n_c^2)$, where $n_c$ is the number of data in the $c$-th class, and $P_c$ is the number of patches partitioned from the data in the $c$-th class with $\sum_{c=1}^{C} P_c = P$. Then MPDA takes $O(\sum_{p=1}^{P}(d^2 N_p + {N_p}^2 d))$ for estimating $P$ tangent spaces and $O((d + \sum_{p=1}^{P} N_p)^3) = O((d+n)^3)$ for solving the generalized eigenvalue problem.

With the above results, we can find that the most consuming parts of PMPDA and MPDA lie in the generalized eigenvalue decomposition. PMPDA needs to decompose $S_p$, a large matrix sized $(d+mn) \times (d+mn)$, which will takes $O((d+kn)^3)$. Compared with PMPDA, MPDA manipulates a much smaller matrix $S$ sized $(d + mP) \times (d + mP)$ and only needs to estimate $P$ tangent spaces rather than $n$. Since $P \ll n$, this leads to significant computational savings.

### 3.6 Further Improvement

Based on the above analysis, it is clear that MPDA avoids both computing tangent spaces for every data point and solving the eigenvalue problem with a large matrix, so that it have less computational complexity. In fact, we can make it more scalable. Note that we use the product $R^p \cdot N_p$ to determine the patch that should be further divided and the manner that how this patch is divided, where $N_p$ is predominant to control our partition algorithm. When the number of data is very large, we can deem that the Euclidean distance is approximately equal to the geodesic distance within

a small region, which leads to $R^p \approx 1$. Therefore, the partition algorithm can be simplified by omitting the computation of $D^G$ and $R^p$, so that we can save the time for performing Dijkstra's algorithm and computing $R^p$.

In addition, the computational costs of MPDA can be reduced by estimating tangent vectors and tangent spaces only at anchor points. In this case, we are interested in selecting a portion of points from the original data set as the anchor points, and the rest can be represented according to the first-order Taylor expansion at their nearest anchor points. Therefore, the data manifold can be estimated by using the anchor points only. It is natural to specify the center of each linear subspaces, which is not necessary a data point among the training set, as the anchor point. As a result, MPDA can be performed on only $P$ anchor points rather than the whole data set, such that the corresponding computational complexity for solving the generalized eigenvalue problem can be reduced to $O((d+P)^3)$.

Moreover, the two-stage strategy [25] can be adopted to further reduce the computational costs. We can separate the generalized eigenvalue problem (19) into two stages. The first stage maximizes (17) via QR decomposition to find its solution space. The second one solves (19) in the solution space of (17). Since $S'$ is just the extension of $2XL'X^\top$, the rank of $S'$ is at most $d$. Consequently, the time for solving (19) can be reduced to $O(d^3)$. Please refer to [25] for more details.

## 4 Discussion

Several works have been done to manipulate data in local subspaces for dimensionality reduction [26], [27], [28], [29]. Basically, they share the same spirit in aligning local subspaces to build a global coordinate, where the connections of local subspaces are considered implicitly. The main difference between MPDA and these methods is that MPDA constructs tangent spaces in a piecewise manner and explicitly characterizes their connections by estimating tangent vectors.

Local Linear Coordination (LLC) [26] and Coordinated Factor Analysis (CFA) [27] construct linear subspaces through the mixture of factor analyzers (MFA) which can serve as an alternative way to partition the data manifold. However, MFA is optimized by the expectation-maximization (EM) algorithm, which can be slow and unstable. Moreover, the number of factor analyzers and the dimensionality of each linear subspace should be specified as a priori knowledge, which are difficult to determine. In contrast, the proposed manifold partition algorithm for constructing linear subspaces is more efficient, and the dimensionality of each linear subspace can be determined automatically by using PCA.

Compared with MPDA, Maximal Linear Embedding (MLE) [29] also constructs a number of linear subspaces based on the measure of linearity but follows a different principle. MLE prefers to construct the linear subspaces whose sizes should be as large as possible, while MPDA constructs a number of linear subspaces with similar and relatively small sizes to justify the manifold assumption as well as the Taylor expansion. Another difference between MPDA and MLE is that although MPDA partitions the data manifold to extract piecewise regional consistency, it still use all the data to discover the underlying manifold, whereas MLE only uses a portion of data to obtain the resultant global coordinate. This means that MPDA utilizes more information from data sets than MLE.

Locally Multidimensional Scaling (LMDS) [28] can be seen as a sparsified version of LTSA. It constructs tangent spaces based on a set of overlapping local subspaces where the number of subspaces should be as small as possible. This strategy allows LMDS to avoid estimating tangent spaces for each data point, and thus makes LMDS more efficient than LTSA. However, if the local subspaces are non-overlapping, LMDS cannot work normally any more, because as an alignment based method, it needs the overlapping parts of local subspaces to serve as the implicit connections for aligning a global coordinate. In contrast, since MPDA explicitly characterizes the connections among tangent spaces by estimating tangent vectors, it can construct a global coordinate based on non-overlapping local subspaces.

TABLE 1
Statistics of the data sets: $n$ is the number of data points, $d$ is the data dimensionality, $C$ is the number of classes, and $\delta$ is the percentages of training data.

| Data Set | $n$ | $d$ | $C$ | $\delta$ |
|---|---|---|---|---|
| COIL20 | 1440 | 1024 | 20 | 25% |
| COIL100 | 7200 | 1024 | 100 | 25% |
| FaceDetection | 2000 | 361 | 2 | 25% |
| MNIST | 4000 | 784 | 10 | 25% |
| OptDigits | 5620 | 64 | 10 | 25% |
| Semeion | 1593 | 256 | 10 | 25% |
| Vehicle | 846 | 18 | 4 | 50% |

## 5 Experiment

### 5.1 Real-World Data Sets

We focus on supervised dimensionality reduction tasks and test the proposed PMPDA and MPDA on multiple real-world data sets. Comparisons are made with: 1) Classical baseline methods including PCA and LDA; 2) Graph Laplacian based methods including Marginal Fisher Analysis (MFA) [5], Locality Sensitive Discriminant Analysis (LSDA) [6] and Local Fisher Discriminant Analysis (LFDA) [7], which are the most related counterparts of MPDA; 3) Tangent space based methods Linear Local Tangent Space Alignment (LLTSA) [30] and linearized PFE (we call it LPFE) which are the linear variations of LTSA and PFE, respectively; 4) Other types

TABLE 2
Average error rates (dimensionality) on different data sets.

| Methods | COIL20 | COIL100 | FaceDetection | MNIST | OptDigits | Semeion | Vehicle |
|---|---|---|---|---|---|---|---|
| Baseline | 4.71%(1024) | 10.49%(1024) | 7.97%(361) | 12.78%(784) | 2.11%(64) | 14.51%(256) | 36.75%(18) |
| PCA | 3.24%(23.95) | 7.48%(38.55) | 4.85%(23.45) | 10.90%(32.75) | 2.03%(35.05) | 11.83%(34.75) | 36.62%(13.95) |
| LDA | 3.11%(19) | 13.99%(99) | 7.44%(1) | 18.98%(9) | 4.66%(9) | 14.66%(9) | 26.67%(3) |
| MFA | 2.30%(15.8) | 7.50%(27.35) | 2.93%(22.35) | 12.21%(47.4) | 2.52%(34.65) | 12.69%(30.65) | 20.20%(11.15) |
| LSDA | 3.31%(19.6) | 8.79%(27.65) | 3.54%(27.65) | 11.77%(34.65) | 2.52%(28.45) | 12.66%(19.1) | 22.09%(10.6) |
| LFDA | 1.89%(17.2) | 7.39%(33.1) | 2.82%(44.8) | 13.53%(34.75) | 2.33%(27.95) | 12.22%(29.2) | **19.63%(10.65)** |
| LLTSA | 6.49%(23.65) | 15.72%(49) | 4.85%(26.55) | 17.27%(32) | 3.94%(22.55) | 19.92%(17.3) | 23.84%(17.45) |
| LSDR | 4.12%(48.15) | - | 7.68%(85.6) | 12.40%(123.4) | - | 14.30%(120.05) | 36.37%(14.45) |
| LPFE | 3.23%(50.05) | 9.60%(155.35) | 5.74%(71.85) | 18.42%(124.3) | 8.68%(28.1) | 21.01%(112.6) | 49.43%(11.4) |
| PMPDA | **1.45%(14.45)** | - | **1.64%(25.1)** | **9.70%(22)** | - | **9.26%(22.6)** | 22.27%(11.6) |
| MPDA | **1.25%(13.85)** | **6.69%(25.75)** | 2.15%(26.9) | 10.09%(28.7) | **1.90%(23.9)** | **8.86%(22.45)** | **19.55%(8.9)** |

of supervised dimensionality reduction methods Least-Squares Dimension Reduction (LSDR) [17]. Seven real-word data sets are used including COIL20, COIL100 [31], Face Detection [32], a subset of MNIST [33] containing the first 2k training and test images, and three UCI data sets including OptDigits, Semeion Handwritten and Vehicle [34]. The configuration of each data set is shown in Table 1.

The parameters $k$, $\alpha$ and $\gamma$ for both PMPDA and MPDA are determined by 4-fold cross validation, and the parameters $k'$ and $M$ for the partition algorithm in MPDA are set to be $k' = 6$ and $M = 10$ empirically. Furthermore, all the parameters for MFA, LSDA, LFDA, LLTSA, and LPFE are selected by 4-fold cross validation. The measure for each round of cross validation is the classification accuracy on the validation set. Specifically, after training different dimensionality reduction algorithms on the training set, we first perform dimensionality reduction on both the training and validation sets, and then train a classifier using the training set in the discovered subspace. Finally, by classifying data in the validation set, we can determine the values of parameters according to the classification results. Originally, LPFE is an unsupervised method. For a fair comparison, LPFE is performed based on a supervised graph which is identical to the within-class graph $G$ used in MPDA. For each data set, we randomly split certain rates of data as the training set to compute the subspace, and then classify the rest of data by the nearest neighbor classifier (1-NN) in the discovered subspace. Every experimental result is obtained from the average over 20 splits. For computational efficiency, we use PCA to preserve 95% energy for the data sets whose dimensionality are larger than 100. In addition, we also compare the baseline method that just employs the 1-NN classifier in the original space without performing dimensionality reduction.

Generally, the classification performance varies with the dimensionality of the subspace. For each method, the best performance as well as the corresponding dimensionality of the subspace are reported. PMPDA is not tested on the COIL100 and OptDigits data sets because of out of memory. LSDR is not tested on the COIL100 and OptDigits data sets since the execution time is too long. Table 2 shows the average error rates of each method with corresponding dimensionality on different data sets, where the best method and the comparable one based on Student's t-test with a p-value of 0.05 are highlighted in bold font. We see that PMPDA or MPDA outperforms other methods in a statistical significant manner for all the data sets except the Vehicle data set. This means that compared with the graph Laplacian based methods, our methods improve the performance of supervised dimensionality reduction by taking advantage of the regional consistency from tangent spaces and keeping in mind that the within-class similarity shall be achieved along the varying direction of the data manifold. LPFE fails to get reasonable results, which is probably because it has no ability to separate data from different classes, and it may lose too much (non-linear) information due to the linearization. It is worth noting that although MPDA can be viewed as the approximation of PMPDA, it still obtains comparable or better results than PMPDA. This suggests that the manifold partition strategy itself is able to improve the performance of dimensionality reduction, because it provides MPDA with more flexibility to estimate tangent spaces. In addition, although not shown in Table 2, if we remove PMPDA out of the comparison, MPDA becomes the best method based on t-test with p=0.05. This demonstrates that MPDA is consistently better than its counterparts. Figure 3 depicts how the mean classification accuracy varies with respect to the dimensionality of embedding spaces on different data sets. It shows that MPDA and PMPDA work quite well. Particularly, except for the Vehicle data set, MPDA and PMPDA (if applicable) consistently obtain the best results with respect to the dimensionality of embedding spaces. To further evaluate the effectiveness of MPDA, we also conduct experiments on the data sets that have been tested by the authors of its counterparts. In this case, results from the existing algorithms can be cited from the corresponding original paper for fairer comparisons. According to Table 2, LFDA seems to be the best algorithm except for MPDR and PMPDR. Because of this, we focus on comparing MPDA with LFDA on the USPS handwritten digit data set according to the configuration of LFDA's original

paper. Again, MPDR and PMPDR outperform their counterparts with statistical significance. Please refer to the supplementary material for details.

### 5.2 Parameter Sensitivity

In this section, we evaluate the parameter sensitivity of MPDA on the Semeion Handwritten data set. Specifically, we aim to test how the performance of MPDA varies with its parameters $k$, $\gamma$, $M$, and $k'$, respectively. To this end, the default values of $k$, $\gamma$, $M$ and $k'$ are set to be $5$, $1$, $10$ and $6$, respectively. And we alternately change one of these parameters to evaluate the performance of MPDA when the other parameters are fixed. Figure 4 implies that $k$ and $M$ are more important than $\gamma$ and $k'$. Their values should be determined properly, while those of $\gamma$ and $k'$ seem to have no significant influence on the performance of MPDA. Overall, MPDA get stable results as its parameters change, where the classification accuracy ranges from $90\%$ to $92\%$. Therefore, MPDA is relatively insensitive to the changes of parameters.

## 6 Conclusion

In this paper we have proposed a tangent space based linear dimensionality reduction method named Manifold Partition Discriminant Analysis (MPDA). By considering both pairwise differences and piecewise regional consistency, MPDA can find a linear embedding space where the within-class similarity is achieved along the direction that is consistent with the local variation of the data manifold, while nearby data belonging to different classes are well separated. Different to graph Laplacian methods that capture only the pairwise interaction between data points, our method capture both pairwise as well as higher order interactions (using regional consistency) between data points.

As a crucial part of MPDA, the manifold partition strategy plays a key role in preserving the manifold structure to improve the measure of the within-class similarity. It not only enables MPDA to adaptively determine the number of dimensions of each linear subspace, but also can be adopted by other tangent space base methods to make them more efficient. The experiments on multiple real-world data sets have shown that compared with existing works MPDA can obtain better classification results.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Projects 61370175, and Shanghai Knowledge Service Platform Project (No. ZF1213).

## References

[1] D. Q. Dai and P. C. Yuen, "Face recognition by regularized discriminant analysis," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 4, pp. 1080–1085, 2007.

[2] H. Zhao and P. C. Yuen, "Incremental linear discriminant analysis for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 1, pp. 210–221, 2008.

[3] F. Dornaika and A. Bosaghzadeh, "Exponential local discriminant embedding and its application to face recognition," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 921–934, 2013.

[4] H. Wang, X. Lu, Z. Hu, and W. Zheng, "Fisher discriminant analysis with L1-norm," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 828–841, 2014.

[5] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[6] D. Cai, X. He, K. Zhou, J. Han, and H. Bao, "Locality sensitive discriminant analysis," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 708–713.

[7] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *Journal of Machine Learning Research*, vol. 8, pp. 1027–1061, 2007.

[8] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3318–3325.

[9] P. Simard, Y. LeCun, and J. S. Denker, "Efficient pattern recognition using a new transformation distance," S. Hanson, J. Cowan, and C. Giles, Eds. Cambridge, MA: Morgan-Kaufmann, 1993, pp. 50–58.

[10] S. Sun, "Tangent space intrinsic manifold regularization for data representation," in *Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 179–183.

[11] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2004.

[12] M. Brand, "Charting a manifold," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 985–992.

[13] B. Lin, X. He, C. Zhang, and M. Ji, "Parallel vector field embedding," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2945–2977, 2013.

[14] D. Huang, M. Storer, F. De la Torre, and H. Bischof, "Supervised local subspace learning for continuous head pose estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2921–2928.

[15] H. Tang, S. M. Chu, M. Hasegawa-Johnson, and T. S. Huang, "Partially supervised speaker clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 959–971, 2012.

[16] T. Zhou and D. Tao, "Double shrinking sparse dimension reduction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 244–257, 2013.

[17] T. Suzuki and M. Sugiyama, "Sufficient dimension reduction via squared-loss mutual information estimation," *Neural Computation*, vol. 25, no. 3, pp. 725–758, 2013.

[18] S. Wang, S. Yan, J. Yang, C. Zhou, and X. Fu, "A general exponential framework for dimensionality reduction," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 920–930, 2014.

[19] F. R. K. Chung, *Spectral Graph Theory*. Rhode Island: American Mathematical Society, 1997.

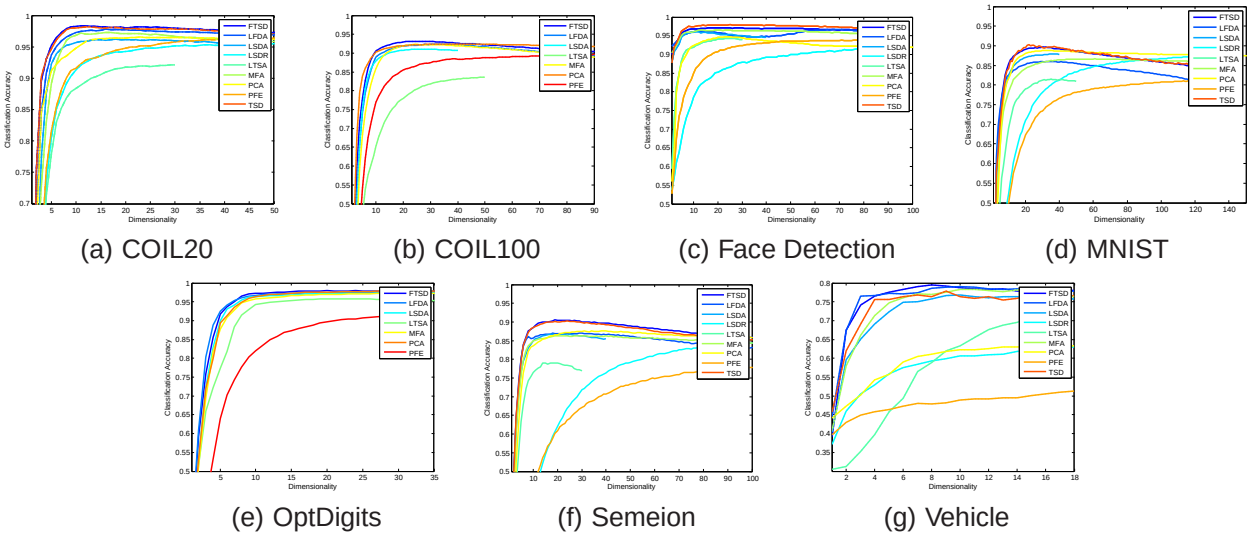
[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, 1990.

[21] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition–Tangent distance and tangent propagation," in *Neural Networks: Tricks of the Trade*. Springer, 2012, vol. 7700, pp. 235–269.

[22] W. Min, K. Lu, and X. He, "Locality pursuit embedding," *Pattern Recognition*, vol. 37, no. 4, pp. 781–788, 2004.

[23] M. B. Clennell, "Tortuosity: A guide through the maze," *Geological Society Special Publications*, vol. 122, pp. 299–344, 1997.

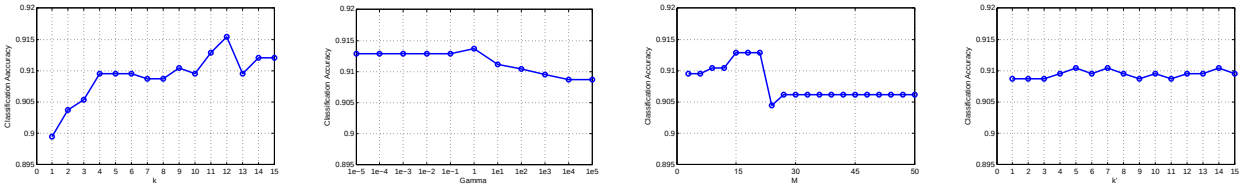
[24] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 2009.




(a) COIL20 (b) COIL100 (c) Face Detection (d) MNIST

(e) OptDigits (f) Semeion (g) Vehicle

Fig. 3. Classification accuracy versus embedding dimensionality on different data sets (better viewed in color).




(a) $k \in \{1, 2, \ldots, 15\}$ (b) $\gamma \in \{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$ (c) $M \in \{5, 10, \ldots, 50\}$ (d) $k' \in \{1, 2, \ldots, 15\}$

Fig. 4. Average classification accuracy of MPDA with respect to the values of different parameters on the Semeion data set.


[25] J. Ye and Q. Li, "A two-stage linear discriminant analysis via QR-decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 929–941, 2005.

[26] Y. W. Teh and S. T. Roweis, "Automatic alignment of local representations," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 865–872.

[27] J. Verbeek, "Learning nonlinear image manifolds by global alignment of local linear models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1236–1250, 2006.

[28] L. Yang, "Alignment of overlapping locally scaled patches for multidimensional scaling and dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 438–450, 2008.

[29] R. Wang, S. Shan, X. Chen, J. Chen, and W. Gao, "Maximal linear embedding for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1776–1792, 2011.

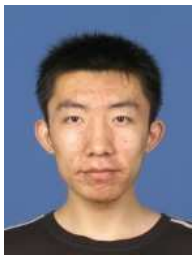
[30] T. Zhang, J. Yang, D. Zhao, and X. Ge, "Linear local tangent space alignment and application to face recognition," *Neurocomputing*, vol. 70, no. 7, pp. 1547–1553, 2007.

[31] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library," Department of Computer Science, Columbia University, Tech. Rep. CUCS-006-96, 1996. [Online]. Available: http://www.cs.columbia.edu/CAVE

[32] M. Alvira and R. Rifkin, "An empirical comparison of SNoW and SVMs for face detection," Center for Biological and Computational Learning, MIT, Cambridge, MA, Tech. Rep. 193, 2001.

[33] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[34] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

**Yang Zhou** is a master student in the Pattern Recognition and Machine Learning Research Group, Department of Computer Science and Technology, East China Normal University. His research interests include pattern recognition, manifold learning, dimensionality reduction, etc.

**Shiliang Sun** is a professor at the Department of Computer Science and Technology and the head of the Pattern Recognition and Machine Learning Research Group, East China Normal University. He received the Ph.D. degree in pattern recognition and intelligent systems from the Department of Automation and the State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, China, in 2007. From 2009 to 2010, he was a visiting researcher at the Department of Computer Science, University College London, working within the Centre for Computational Statistics and Machine Learning. In July 2014, he was a visiting researcher at the Department of Electrical Engineering, Columbia University, New York. He is on the editorial boards of multiple international journals including Neurocomputing and IEEE Transactions on Intelligent Transportation Systems. His research interests include kernel methods, learning theory, multi-view learning, approximate inference, sequential modeling and their applications, etc.

# Supplementary Material for Manifold Partition Discriminant Analysis

Yang Zhou and Shiliang Sun

## 1 Detailed Derivation of S

By representing $S$ as a block matrix, the within-class objective function becomes:

$$\min_{\boldsymbol{t},\boldsymbol{v}} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{v} \end{pmatrix}^\top \begin{pmatrix} S_1 & S_2 \\ S_2^\top & S_3 \end{pmatrix} \begin{pmatrix} \boldsymbol{t} \\ \boldsymbol{v} \end{pmatrix} = \boldsymbol{f}^\top S \boldsymbol{f}. \quad (1)$$

In order to fix $S$, we decompose (1) into four additive terms as follows:

$$\begin{aligned} \boldsymbol{f}^\top S \boldsymbol{f} = & \underbrace{\sum_{i,j=1}^{n} W_{ij}((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t})^2}_{\text{term one}} + \\ & \underbrace{\sum_{i,j=1}^{n} W_{ij}\big(\boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2}_{\text{term two}} + \\ & \underbrace{\sum_{i,j=1}^{n} W_{ij}\big[-2((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t})\boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big]}_{\text{term three}} + \\ & \underbrace{\gamma \sum_{i,j=1}^{n} W_{ij} \|\boldsymbol{v}_{\pi_i} - T_{\pi_i} T_{\pi_j}^\top \boldsymbol{v}_{\pi_j}\|_2^2}_{\text{term four}}, \end{aligned}$$

and examine their separate contributions to the whole $S_p$.

### Term One

$$\begin{aligned} & \sum_{i,j=1}^{n} W_{ij}((\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{t})^2 \\ =& 2\boldsymbol{t}^\top X(D - W)X^\top \boldsymbol{t} = 2\boldsymbol{t}^\top X L X^\top \boldsymbol{t}, \end{aligned}$$

where $D$ is a diagonal weight matrix with $D_{ii} = \sum_{j=1}^{n} W_{ij}$, and $L = D - W$ is the Laplacian matrix. Thus term one contributes to $S_1$ in (1).


- *Yang Zhou and Shiliang Sun (corresponding author) are with Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, P. R. China (email: shiliangsun@gmail.com, slsun@cs.ecnu.edu.cn)*


### Term Two

Define $B_{\pi_j i} = T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)$. Then

$$\begin{aligned} & \sum_{i,j=1}^{n} W_{ij}\big(\boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j)\big)^2 \\ = & \sum_{i,j=1}^{n} W_{ij}(\boldsymbol{v}_{\pi_j}^\top B_{\pi_j i})^2 = \sum_{j=1}^{n} \boldsymbol{v}_{\pi_j}^\top \big(\sum_{i=1}^{n} W_{ij} B_{\pi_j i} B_{\pi_j i}^\top\big)\boldsymbol{v}_{\pi_j}. \end{aligned}$$

Let $\Pi_p = \{i | \pi_i = p, i \in \{1, \ldots, n\}\}$ be a set that consists of the indices of the data belonging to the $p$-th linear subspace. Then we can group the terms with respect to $\boldsymbol{v}_{\pi_j}$ $(j = 1, \ldots, n)$ into $P$ terms as follows:

$$\begin{aligned} & \sum_{j=1}^{n} \boldsymbol{v}_{\pi_j}^\top \big(\sum_{i=1}^{n} W_{ij} B_{\pi_j i} B_{\pi_j i}^\top\big)\boldsymbol{v}_{\pi_j} \quad (2) \\ = & \sum_{p=1}^{P} \boldsymbol{v}_p^\top \big(\sum_{j \in \Pi_p} H_j\big)\boldsymbol{v}_p, \end{aligned}$$

where we have defined matrices $\{H_j\}_{j=1}^n$ with $H_j = \sum_{i=1}^{n} W_{ij} B_{\pi_j i} B_{\pi_j i}^\top$.

Now we can define a block diagonal matrix $S_3^H$ sized $mP \times mP$, where the block size is $m \times m$. Set the $(i,i)$-th block $(i = 1, \ldots, P)$ of $S_3^H$ to be $\sum_{j \in \Pi_p} H_j$. Then the resultant $S_3^H$ is the contribution of term two for $S_3$ in (1).

### Term Three

Define vectors $\{F_p\}_{p=1}^P$ with $F_p = \sum_{i=1}^{n} \sum_{j \in \Pi_p} W_{ij} B_{\pi_j i} \boldsymbol{x}_i^\top$. Then term three can be

rewritten as:

$$\begin{aligned}
&\sum_{i,j=1}^{n} W_{ij}\big[-2((\boldsymbol{x}_i-\boldsymbol{x}_j)^\top \boldsymbol{t})\boldsymbol{v}_{\pi_j}^\top T_{\pi_j}^\top(\boldsymbol{x}_i-\boldsymbol{x}_j)\big]\\
=&\sum_{i,j=1}^{n} 2W_{ij}\big[((\boldsymbol{x}_j-\boldsymbol{x}_i)^\top \boldsymbol{t})\boldsymbol{v}_{\pi_j}^\top B_{\pi_j i}\big]\\
=&\sum_{p=1}^{P}\boldsymbol{t}^\top(\sum_{i=1}^{n}\sum_{j\in\Pi_p} -W_{ij}\boldsymbol{x}_i B_{\pi_j i}^\top)\boldsymbol{v}_p+\\
&\sum_{p=1}^{P}\boldsymbol{v}_p^\top(\sum_{i=1}^{n}\sum_{j\in\Pi_p} -W_{ij}B_{\pi_j i}\boldsymbol{x}_i^\top)\boldsymbol{t}+\\
&\sum_{p=1}^{P}\boldsymbol{t}^\top F_p^\top \boldsymbol{v}_p+\sum_{p=1}^{P}\boldsymbol{v}_p^\top F_p\boldsymbol{t}.
\end{aligned}$$

From this expression, we can give the formulation of $S_2$. Then the block matrix $S_2^\top$ in (1), which is its transpose, is ready to get.

Suppose we define two block matrices $S_2^1$ and $S_2^2$ sized $d\times mP$ each where the block size is $d\times m$, and $S_2^2$ is a block diagonal matrix. Set the $p$-th block $(p=1,\ldots,P)$ of $S_2^1$ to be $\sum_{i=1}^n\sum_{j\in\Pi_p} -W_{ij}\boldsymbol{x}_i B_{\pi_j i}^\top$, and the $(p,p)$-th block $(p=1,\ldots,P)$ of $S_2^2$ to be $F_p^\top$. Then, term three can be rewritten as: $\boldsymbol{t}^\top(S_2^1+S_2^2)\boldsymbol{v}+\boldsymbol{v}^\top(S_2^1+S_2^2)^\top\boldsymbol{t}$. It is clear that $S_2=S_2^1+S_2^2$.

### Term Four

Denote matrix $T_{\pi_i}T_{\pi_j}^\top$ by $A_{\pi_i\pi_j}$. Then, with $\gamma$ omitted temporarily,

$$\begin{aligned}
&\sum_{i,j=1}^{n} W_{ij}\|\boldsymbol{v}_{\pi_i}-T_{\pi_i}T_{\pi_j}^\top\boldsymbol{v}_{\pi_j}\|_2^2\\
=&\sum_{i,j=1}^{n} W_{ij}\boldsymbol{v}_{\pi_j}^\top A_{\pi_i\pi_j}^\top A_{\pi_i\pi_j}\boldsymbol{v}_{\pi_j}+\\
&\sum_{i,j=1}^{n} W_{ij}\boldsymbol{v}_{\pi_i}^\top\boldsymbol{v}_{\pi_i}-\sum_{i,j=1}^{n}2W_{ij}\boldsymbol{v}_{\pi_i}^\top A_{\pi_i\pi_j}\boldsymbol{v}_{\pi_j}.
\end{aligned}$$

Similarly, we can sum up the terms with respect to $\boldsymbol{v}_{\pi_j}$, which further leads to:

$$\begin{aligned}
&\sum_{i,j=1}^{n} W_{ij}\boldsymbol{v}_{\pi_j}^\top A_{\pi_i\pi_j}^\top A_{\pi_i\pi_j}\boldsymbol{v}_{\pi_j}+\\
&\sum_{i=1}^{n} D_{ii}\boldsymbol{v}_{\pi_i}^\top I\boldsymbol{v}_{\pi_i}-\sum_{i,j=1}^{n}2W_{ij}\boldsymbol{v}_{\pi_i}^\top A_{\pi_i\pi_j}\boldsymbol{v}_{\pi_j}\\
=&\sum_{p=1}^{P}\boldsymbol{v}_p^\top(\sum_{j\in\Pi_p}(D_{jj}I+C_j))\boldsymbol{v}_p-\\
&\sum_{p=1}^{P}\sum_{q=1}^{P}\boldsymbol{v}_p^\top(\sum_{i\in\Pi_p}\sum_{j\in\Pi_q}2W_{ij}A_{\pi_i\pi_j})\boldsymbol{v}_q.
\end{aligned}$$

where in the third line we have defined matrices $\{C_j\}_{j=1}^n$ with $C_j=\sum_{i=1}^n W_{ij}A_{\pi_i\pi_j}^\top A_{\pi_i\pi_j}$.

Now suppose we define two block matrices $S_3^1$ and $S_3^2$ sized $mP\times mP$ each where the block size is $m\times m$, and $S_3^1$ is a block diagonal matrix. Set the $(p,p)$-th block $(p=1,\ldots,P)$ of $S_3^1$ to be $\sum_{j\in\Pi_p}(D_{jj}I+C_j)$, and the $(p,q)$-th block $(p,q=1,\ldots,P)$ of $S_3^2$ to be $\sum_{i\in\Pi_p}\sum_{j\in\Pi_q}2W_{ij}A_{\pi_i\pi_j}$. Then the contribution of term four for $S_3$ would be $\gamma(S_3^1-S_3^2)$. Further considering the contribution of term two for $S_3$, we finally have $S_3=S_3^H+\gamma(S_3^1-S_3^2)$.

TABLE 1
Average error rates (dimensionality) on the USPS data sets.

| Methods | USPS-eo | USPS-sl |
|---|---|---|
| Baseline | 2.72%(256) | 3.25%(256) |
| PCA | 2.47%(37.9) | 2.93%(38.75) |
| LDA | 10.93%(9) | 21.25%(9) |
| MFA | 2.79%(26.3) | 3.57%(29.45) |
| LSDA | 3.39%(26.8) | 4.21%(27.1) |
| LFDA | 7.89%(39.25) | 9.84%(20.9) |
| LLTSA | 8.16%(28.4) | 8.47%(29.85) |
| LSDR | - | - |
| LPFE | 4.33%(184.4) | 3.61%(180.25) |
| PMPDA | **1.76%(28.65)** | **2.20%(31.6)** |
| MPDA | **1.67%(35.25)** | **2.28%(30.2)** |

## 2 USPS Data Set

In this section, we further evaluate the effectiveness of MPDA by conducting experiments on the data sets that have been tested by the authors of its counterparts. In this case, results from the existing algorithms can be cited from the corresponding original paper for fairer comparisons. According to Table 2 in our paper, LFDA seems to be the best algorithm except for MPDR and PMPDR. Because of this, we focus on comparing MPDA with LFDA. Specifically, our experiments are conducted on two binary classification data sets created from the USPS handwritten digit data set. The first task (USPS-eo) is to separate even numbers from odd numbers, and the second task (USPS-sl) is to separate small numbers ("0" to "4") from large numbers ("5" to "9"). We randomly chose 100 data points from each digit to form both the training and test set, so that there are 1000 data points for training and testing, respectively. The strategy of model selection is the same as that in our paper. We report the best classification results obtained by each method and the corresponding dimensionality at which the results are achieved. Every experimental result is obtained from the average over 20 times, where the best method and the comparable one based on Student's t-test with a p-value of 0.05 are highlighted in bold font. The above experimental configuration is identical to the one used in LFDAs original paper [1] except for two differences. The first is that the neighborhood size $k$ is treated as a parameter for graph construction. We determine $k$ via 4-fold cross validation, while $k$ is set to be 7 in LFDA's original paper [1]. The second is that we search all the possible dimensionalities of embedding subspaces to report the best classification results, whereas the results

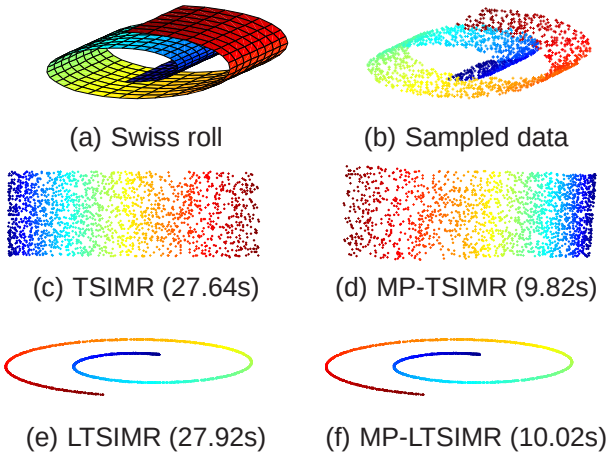



Fig. 1. The "Swiss roll" manifold and the corresponding embedding results (time) obtained by the dimensionality reduction methods with and without partitioning the manifold.

in [1] are obtained by selecting the dimensionality of embedding subspaces via 20-fold cross validation. This two differences imply that we are expected to get relatively better results than those reported in [1]. Table 1 shows that MPDA and PMPAD still outperform its counterparts with statistical significance. LSDR is not tested since the execution time is too long. It is worth noting that the reported classification results of LFDA in [1] are $9.0\%$ for UPSP-eo and $12.9\%$ for UPSP-sl, respectively. In our experiments, these results are improved as expected. However, they are still much worse than the results of MPDA and PMPDA.

## 3 Effectiveness of Partitioning the Manifold

As a crucial part of MPDA, the manifold partition strategy plays a key role in preserving the manifold structure with computational and storage efficiency. In fact, it can also be adopted by other tangent space base methods [2], [3] to make them more efficient. To verify its effectiveness, we apply the manifold partition strategy to Tangent Space Intrinsic Manifold Regularization (TSIMR) [3] and its linear variation (we call it LTSIMR). This further leads to two algorithms called MP-TSIMR and MP-LTSIMR, which can be viewed as the approximated versions of TSIMR and LTSIMR. To show the effectiveness of the manifold partition strategy, we compare the embedding results of these methods on the "Swiss roll" manifold.

The "Swiss roll" is a two-dimensional manifold in a three-dimensional ambient space as shown in Figure 1(a). The data set consists of $2000$ points sampled from the manifold, which are depicted in Figure 1(b). The construction of the adjacency graph uses $10$ nearest neighbors with the heat kernel. The parameter $t$ of the heat kernel is fixed as the average of the squared distances between all points and their most nearest neighbors. The parameter $\gamma$ is set to be $\gamma = 0$ for TSIMR and MP-TSIMR, and $\gamma = 10^9$ for LTSIMR and MP-LTSIMR. We fix the parameters $k' = 31$ and $M = 48$ for the manifold partitioning algorithm.

Figure 1 shows the two-dimensional embedding results obtained by different algorithms. As can be seen in Figure 1(c), TSIMR precisely reflects the intrinsic manifold structure. In addtion, Figure 1(d) shows that MP-TSIMR also gets a good embedding result besides a little distortion. In the case of linear dimensionality reduction, LTSIMR and MP-LTSIMR almost get identical results as shown in Figures 1(e) and 1(f). These embedding results demonstrate that MP-TSIMR and MP-LTSIMR have similar or the same embedding performance compared with TSIMR and LTSIMR. When it comes to the computational time, MP-TSIMR and MP-LTSIMR are three times faster than their counterparts. This is because both TSIMR and LTSIMR estimate 2000 tangent vectors and tangent spaces to discover the intrinsic manifold structure, whereas MP-TSIMR and MP-LTSIMR only need to estimate 65 tangent vectors and tangent spaces. Therefore, it is clear that the manifold partition strategy not only is useful for MPDA, but can accelerate other tangent space based methods without sacrificing their performances much.

## 4 Storage Overheads

Apart from high computational costs, many tangent based methods such as TSIMR [3] and PFE [2] are also storage consuming, which greatly hampers their applications. In contrast, the proposed MPDA algorithm is free from this limitation. In this section, we evaluate the storage overheads of MPDA compared with its tangent space based counterparts including PMPDA and LPFE on the COIL20, COIL100, Face Detection, MNIST, Opt-Digits, Semeion Handwritten and Vehicle data sets. The number of the training data for each data set is the same with the setting used in our paper. In order to obtain consistent results, the neighborhood size for constructing the within-class graph $G$ is set to be $k = 7$, because the storage costs of PMPDA and LPFE directly depend on $k$. Figure 2 shows the peak memory costs of PMPDA, LPFE and MPDA on different data sets. As can be seen, MPDA has the least storage overheads in all cases (about 150 times and 60 times less than PMPDA and LPFE, respectively). In Figures 2(b) and 2(e), the peak memory costs of PMPDA and LPFE are extremely high as the number of data becomes relatively large, whereas MPDA still has very low memory costs. It should be noted that the above results are obtained under the condition of $k = 7$. In practice, $k$ may be larger, say $k = 10$ or $k = 15$. In this case, the storage overheads of PMPDA and LPFE grow quickly as $k$ becomes large. Consequently, LPFE can barely work normally, and PMPDA is no longer applicable because of out of memory. In contrast, MPDA still costs the same amount of memory because its storage overheads are independent of $k$. Therefore,

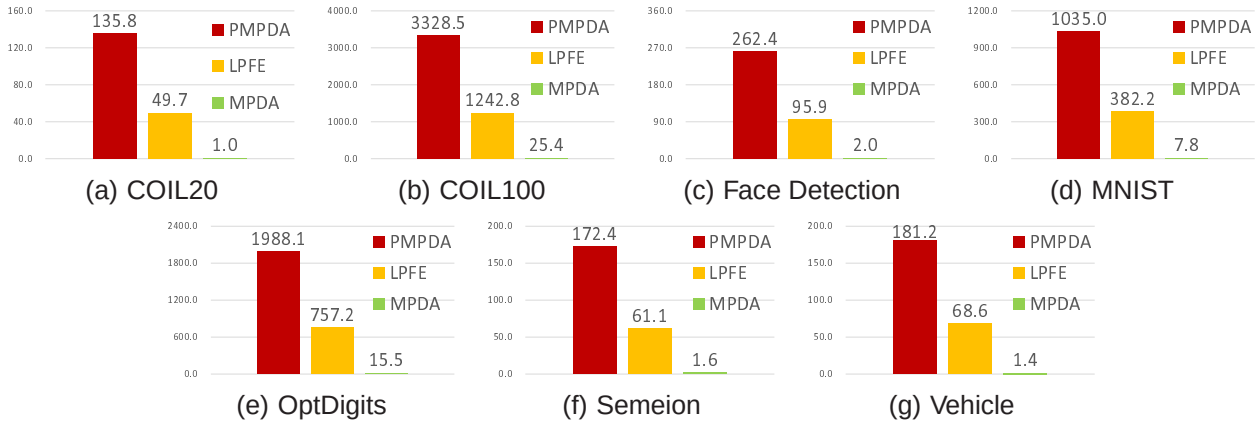



Fig. 2. Peak memory costs (Mb) of PMPDA, LPFE and MPDA on different data sets.

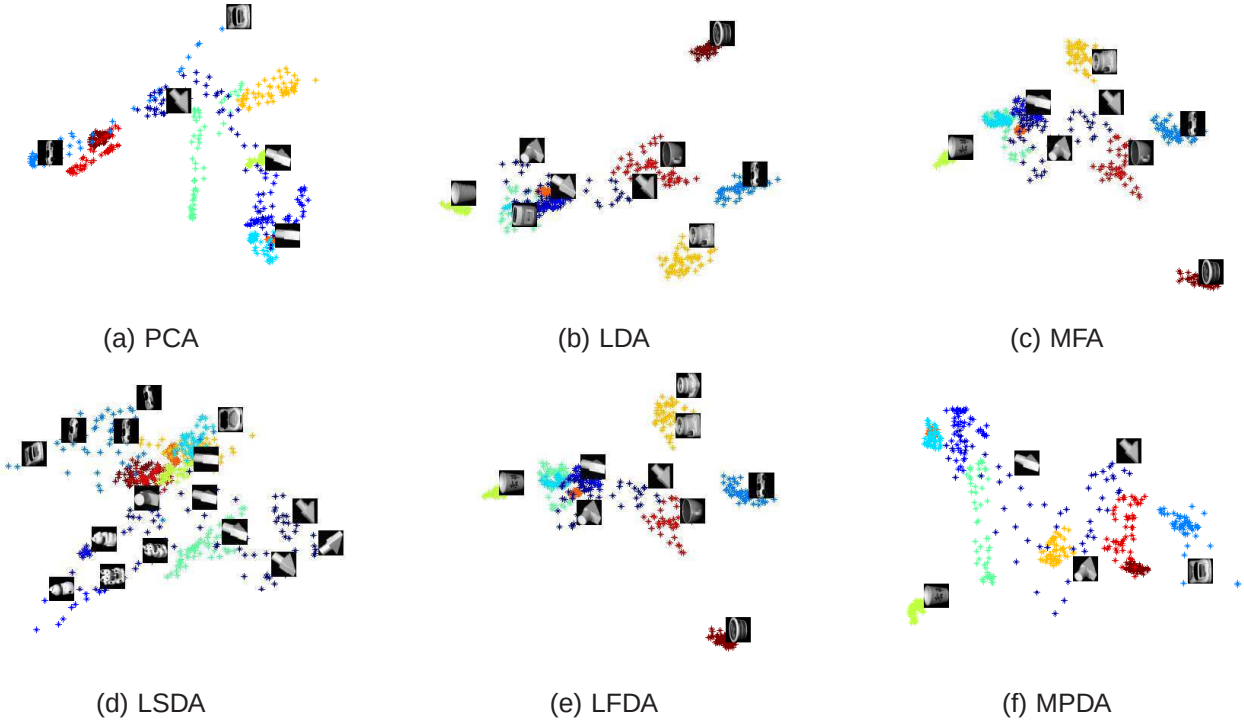


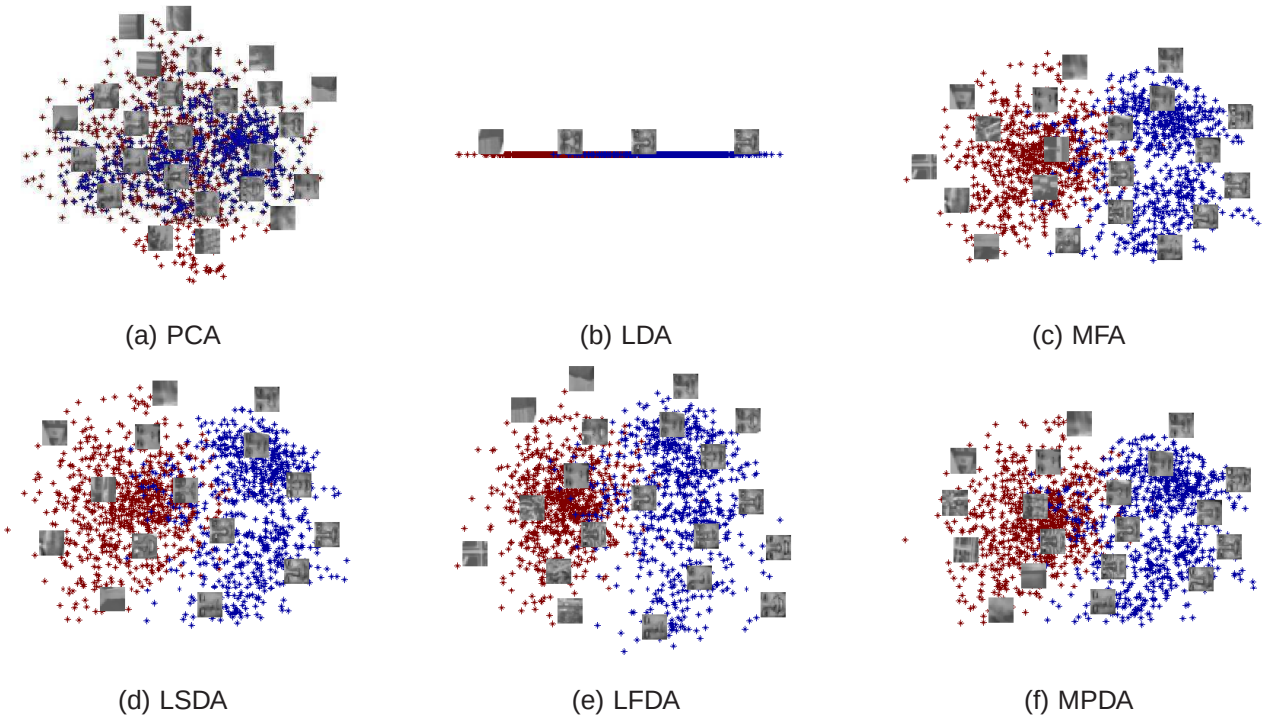

Fig. 3. Two-dimensional embedding results of the COIL20 dataset (better viewed in color).

the proposed MPDA algorithm is more scalable and practical than its counterparts.

## 5 2D Embedding Results

MPDA aims to find an embedding space where the manifold structure with respect to each class is preserved as much as possible, while nearby data from different classes are well separated. To verify whether this goal is achieved, we test the two-dimensional embedding results obtained by MPDA on the COIL20 and Face Detection image data sets. The data embeddings belonging to different classes are represented by different colors, and thumbnails of some images are also shown. In order to represent the data embeddings clearly, we only show the data from 10 out of 20 classes in the COIL20 data set. The embedding results obtained by other methods including PCA, LDA, MFA, LSDA and LFDA are also provided.

In Figure 3, all the methods except LSDA obtain reasonable results for the COIL20 data set. LDA, MFA and LFDA get similar embedding results because they essentially fall into the same framework [4]. Compared with other methods, the embeddings obtained by MPDA are quite different but still reasonable in gathering the data in the same class as well as separating those from different classes. As can be seen in Figure 4, the embeddings of each class obtained by PCA seriously overlap each other, because it has no ability to utilize the discriminant information from class labels. In addition, LDA also gets bad embedding results, because the Face Detection is a binary classification data set so that LDA can only map the data into a one-dimensional space. On



(a) PCA (b) LDA (c) MFA

(d) LSDA (e) LFDA (f) MPDA

Fig. 4. Two-dimensional embedding results of the Face Detection dataset (better viewed in color).

the other side, MFA, LSDA, LFDA and MPDA obtain reasonable results because they share the same spirit of gathering the data in the same class and separating those in different classes. It is worth noting that since Figures 3 and 4 only reflect the embedding performance of MPDA, we cannot draw any conclusion that whether or not the classification performance of MPDA is superior to other methods just from these figures..

## REFERENCES


[1] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *Journal of Machine Learning Research*, vol. 8, pp. 1027–1061, 2007.

[2] B. Lin, X. He, C. Zhang, and M. Ji, "Parallel vector field embedding," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2945–2977, 2013.

[3] S. Sun, "Tangent space intrinsic manifold regularization for data representation," in *Proceedings of the IEEE China Summit and International Conference on Signal and Information Processing*, 2013, pp. 179–183.

[4] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.