Flexible Orthogonal Neighborhood Preserving Embedding

Tianji Pang[1], **Feiping Nie**[1,2], **Junwei Han**[1]
[1]Northwestern Polytechnical University, Xian 710072, P. R. China.
[2]University of Texas at Arlington, USA
{pangtj911,feipingnie,junweihan2010}@gmail.com

Abstract

In this paper, we propose a novel linear subspace learning algorithm called Flexible Orthogonal Neighborhood Preserving Embedding (FONPE), which is a linear approximation of Locally Linear Embedding (LLE) algorithm. Our novel objective function integrates two terms related to manifold smoothness and a flexible penalty defined on the projection fitness. Different from Neighborhood Preserving Embedding (NPE), we relax the hard constraint $\mathbf{P}^T\mathbf{X} = \mathbf{Y}$ by modeling the mismatch between $\mathbf{P}^T\mathbf{X}$ and $\mathbf{Y}$, which makes it better cope with the data sampled from a non-linear manifold. Besides, instead of enforcing an orthogonality between the projected points, i.e. $(\mathbf{P}^T\mathbf{X})(\mathbf{P}^T\mathbf{X})^T = \mathbf{I}$, we enforce the mapping to be orthogonal, i.e. $\mathbf{P}^T\mathbf{P} = \mathbf{I}$. By using this method, FONPE tends to preserve distances so that the overall geometry can be preserved. Unlike LLE, as FONPE has an explicit linear mapping between the input and the reduced spaces, it can handle novel testing data straightforwardly. Moreover, when $\mathbf{P}$ becomes an identity matrix, our model can be transformed into denoising LLE (DLLE). Compared with the standard LLE, we demonstrate that DLLE can handle data with noise better. Comprehensive experiments on several benchmark databases demonstrate the effectiveness of our algorithm.

1 Introduction

Nowadays, most of data we obtain is always very high-dimensional, but its potentially meaningful structure is of much lower dimensionality. A large number of dimension reduction techniques[Xu *et al.*, 2016] [Huang *et al.*, 2015] [Elbagoury and Ibrahim, 2016] have been proposed to extract the underlying low-dimensional structure and have been widely used in many fields, such as data mining, machine learning, computer vision, etc., as an important preprocessing step. By using the dimension reduction methods, data can be better understood and its intrinsic structure can be better visualized. Moreover, dimension reduction methods can overcome the curse of dimensionality effectively.

Two popular dimension reduction techniques are Principal Component Analysis (PCA) [Turk and Pentland, 1991] and Linear Discriminant Analysis (LDA) [Belhumeur *et al.*, 1997]. Their modified methods [Wang *et al.*, 2015b] [Li and Tao, 2012] are also widely used in many different fields, including face recognition, fingers movement recognition and so on. In recent years, there has been a growing interest in discovering the manifold of data since Roweis and Saul [Roweis and Saul, 2000] and Tenenbaum et al. [Tenenbaum *et al.*, 2000] who indicate that many types of high dimensional data can be characterized by a low dimensional underlying manifold. However, both PCA and LDA effectively discover only the Euclidean structure. They fail to discover the underlying structure if the data lie on a manifold. Besides, a manifold of data usually exhibits significant non-linear structure while PCA and LDA are both linear dimension reduction methods.

To discover the intrinsic geometry structure of a data set, many non-linear manifold learning methods have been proposed, such as locally linear embedding (LLE) [Roweis and Saul, 2000], Isomap [Tenenbaum *et al.*, 2000], and Laplacian Eigenmap [Belkin and Niyogi, 2003]. These non-linear methods do yield impressive results on some artificial data sets. However, they often suffer from the so-called out-of-sample problem [Bengio *et al.*, 2004], which means they can not handle new data points which are not included in the training set. Besides, their non-linear property makes them computationally expensive. Thus, they might not be suitable for many real world tasks.

At the same time, a large number of kernel based dimension reduction methods, such as kernel PCA [Schölkopf *et al.*, 1998] and kernel LDA [Liu *et al.*, 2002] have also be proposed. These method can also discover the non-linear structure of data. However, most of these methods do not consider the structure of the manifold of data explicitly. Besides, they are also computationally expensive.

To solve these problems above, i.e. the out-of-sample problem and learning the non-linear manifold of data, many linear dimensionality reduction methods based on manifold learning were proposed. Local learning projection (LPP) [He and Niyogi, 2004] and neighborhood preserving embedding (NPE) [He *et al.*, 2005] are the representative ones among these methods. In [Wang *et al.*, 2015a], Wang et al. also proposed a modified method of LPP called robust LPP (rLPP). Different from PCA which aims at preserving the glob-

al Euclidean structure, LPP, NPE and rLPP aim at preserving the local neighborhood structure. Actually, LPP is the optimal linear approximations to the eigenfunctions of the Laplace Betrami operator on the manifold [He and Niyogi, 2004] while NPE is a linear approximation to the LLE [He *et al.*, 2005]. rLPP improves the robustness of LPP by using ℓ_1-norm or not-squared ℓ_2-norm formulations instead of squared ℓ_2-norm formulations. Thus, though rLPP, LPP, NPE and PCA are all linear dimensionality reduction methods, rLPP, LPP and NPE can successfully maintain the non-linear manifold structure of data but PCA can not. Besides, as they can map the data in the original feature space to a lower dimensional space by a simple linear transformation, they can be used for faster training and testing in real applications, as well as the interpretation of the data. However, LPP, rLPP and NPE all used the hard constraint, i.e. $\mathbf{P}^T\mathbf{X} = \mathbf{Y}$ to obtain a linear approximation of the non-linear manifold structure of data, which is supposed to be too strict to cope with the data sampled from a non-linear manifold very well. What is more, both LPP and NPE require an orthogonality relationship between the projected points, i.e. $\mathbf{Y}\mathbf{Y}^T = \mathbf{I}$. This kind of orthogonality may lead to a criterion that is similar to that of PCA: the projected data points tend to be different from one another which results in the problem that the overall geometry cannot be preserved very well [Kokiopoulou and Saad, 2007].

In this paper, we propose a novel linear dimensionality reduction algorithm, called Flexible Orthogonal Neighborhood Preserving Embedding (FONPE). The dimensionality reduction matrix $\mathbf{P}$ is obtained by minimizing an objective function which is constructed by a manifold smoothness term and a flexible penalty term defined on the projection fitness. Specifically, given a set of data points, we first build a weighted matrix which explicitly models the topology of the data. Similar to LLE, each data point can be represented as a linear combination of its neighboring data points and the weight matrix is constructed by the corresponding combination coefficients. Then, we can obtain a dimensionality reduction matrix $\mathbf{P}$ by solving an eigenvalue decomposition. Actually, FONPE is a linear approximation to the LLE. Thus, like most other manifold learning based linear dimensionality reduction methods, FONPE can learn the non-linear manifold of data though it is a linear method and it can also overcome the out-of-sample problem since it employs an explicit mapping between the input and the reduced space. But different from NPE which is also an LLE linear approximation algorithm, FONPE relaxes the hard constraint, i.e. $\mathbf{P}^T\mathbf{X} = \mathbf{Y}$, in NPE by introducing a flexible mismatch penalty between $\mathbf{P}^T\mathbf{X}$ and $\mathbf{Y}$. With this relaxation, FONPE can better deal with the samples which reside on a non-linear constraint. Besides, we enforce the mapping to be orthogonal instead of the projected points as NPE does. In [Kokiopoulou and Saad, 2007], Kokiopoulou et al. had demonstrated that this kind of orthogonality can preserve the overall geometry better. What is more, we also find an interesting fact that when the projection matrix $\mathbf{P}$ becomes an identity matrix, our model can be transformed into denoising LLE (DLLE), which can handle noised data better than LLE.

2 Related Works

In this section, we will give a brief introduction to linear dimension reduction and a brief review of LLE and NPE.

2.1 Linear Dimension Reduction

Given a dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$, where $\mathbf{x}_i \in \mathbb{R}^{m\times 1}$ is the i-th sample, linear dimension reduction aims to find a transformation matrix $\mathbf{P} \in \mathbb{R}^{m\times d}$ that can maps the data from the orignally m-dimensional space to the d-dimensional space, where $d << m$, via a simple linear transformation as follows:

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X} \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^{d\times n}$ is the compact low dimensional data which can represents $\mathbf{X}$.

2.2 Local Linear Embedding

LLE [Roweis and Saul, 2000] aims to preserve the local linear reconstruction relationship among the data points. LLE expects each data point and its neighbors to lie on or be closed to a locally linear manifold. Thus, each data point can be reconstructed from its k nearest neighbors by a weighted linear combination. The weight coefficients $\mathbf{W}$ are calculated by minimizing the following cost function:

$$\begin{aligned} &\varepsilon(\mathbf{W}) = \sum_i \|\mathbf{x}_i - \sum_{j\in N_k(\mathbf{x}_i)} W_{ij}\mathbf{x}_j\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{W}^T\|_F^2 \\ &s.t. \mathbf{W}\mathbf{1} = \mathbf{1} \end{aligned} \tag{2}$$

where $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is a column vector with all its entries being 1, $N_k(\mathbf{x}_i)$ denotes the k nearest neighbors of $\mathbf{x}_i$ and the weight W_{ij} summarizes the contribution of the jth data point to the construction of the ith data point.

To preserve the intrinsic geometric properties of the local neighborhoods, LLE assumes that nearby points in the high dimensional space remain nearby with respect to one another in the low dimensional space. Besides, LLE expects the reconstruction weights W_{ij} can reflect geometric properties of data. The same weights which reconstruct the point $\mathbf{x}_i$ by its neighbors in the high dimensional space should also reconstruct its embedded manifold coordinates in a low dimensional space. Thus, the final embedding coordinates can be calculated by fixing the weight coefficients and minimizing the following cost function:

$$\begin{aligned} &\varepsilon(\mathbf{Y}) = \sum_i \|\mathbf{y}_i - \sum_j W_{ij}\mathbf{y}_j\|^2 = \|\mathbf{Y} - \mathbf{Y}\mathbf{W}^T\|_F^2 \\ &s.t. \mathbf{Y}\mathbf{1} = \mathbf{0}, \mathbf{Y}\mathbf{Y}^T = N\mathbf{I} \end{aligned} \tag{3}$$

where $\mathbf{0} \in \mathbb{R}^{n\times 1}$ is a column vector with all its entries being 0 and $\mathbf{I}$ is a identity matrix. The constraint here is to make the problem well-posed. The problem will amount to computing the smallest $d+1$ eigenvalues of the matrix $\mathbf{M} = (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$ and the associated eigenvectors.

Since LLE suffers from the out-of-sample problem, He et al. [He *et al.*, 2005] proposed a linearization method called NPE.

2.3 Neighborhood Preserving Embedding

NPE [He *et al.*, 2005] is a linear approximation to the LLE algorithm. NPE first constructs an adjacency graph by using the k nearest neighbors method or the ε neighborhood method. Then, the weights on the edges can be computed by (2) as LLE does. Afterwards, NPE uses the strategy of linear approximation to the nonlinear mapping of LLE to learn the projection. By making $\mathbf{P}^T\mathbf{x}_i = \mathbf{y}_i$, the projection can be obtained by minimizing the following cost function:

$$\min_{\mathbf{P}^T\mathbf{X}\mathbf{X}^T\mathbf{P}=\mathbf{I}} \sum_i \|\mathbf{P}^T\mathbf{x}_i - \sum_j W_{ij}\mathbf{P}^T\mathbf{x}_i\|^2 \tag{4}$$

The constraint here is to remove an arbitrary scaling factor in the projection.

The optimal projections of (4) correspond to the minimum eigenvalue of the following standard eigenvalue problem:

$$\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{P} = \mathbf{X}\mathbf{X}^T\mathbf{P}\mathbf{\Lambda} \tag{5}$$

where $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$ and $\mathbf{\Lambda}$ is the eigenvalue matrix whose diagonal elements are the eigenvalues of $(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{M}\mathbf{X}^T$. Finally, the embedding can be obtained by (1).

As we can see, NPE replaces $\mathbf{Y}$ by $\mathbf{P}^T\mathbf{X}$ in LLE to calculate the projection matrix. This linear approximating may be overstrict to fit the data samples from a non-linear manifold. Besides, NPE imposes the orthogonal constraint $\mathbf{P}^T\mathbf{X}\mathbf{X}^T\mathbf{P} = \mathbf{I}$. However, Kokiopoulou et al. in [Kokiopoulou and Saad, 2007] pointed out that this kind of orthogonal constraint tends to make the projected points different from one another, which makes NPE can not preserve the overall geometry very well. To solve these two problems, we propose a novel linear dimension reduction method.

3 The Proposed Algorithm

In this section, we present the details of the proposed method, called Flexible Orthogonal Neighborhood Preserving Embedding (FONPE). An iterative method is also proposed to solve the objective function.

3.1 Problem Formulation

Our algorithm is a linear approximation to LLE. Different from NPE, we relax the hard mapping function $\mathbf{Y} = \mathbf{P}^T\mathbf{X}$ by introducing a flexible penalty term. By using this relaxation, our framework is more flexible and it can better cope with the samples which reside on the nonlinear manifold. We formulate our objective function as follows:

$$\min_{\mathbf{P},\mathbf{F},\mathbf{W}\mathbf{1}=\mathbf{1}} \|\mathbf{F} - \mathbf{F}\mathbf{W}^T\|_F^2 + \beta\|\mathbf{P}^T\mathbf{X} - \mathbf{F}\|_F^2 \tag{6}$$

where $\mathbf{F} \in \mathbb{R}^{d\times n}$ is the low-dimensional manifold embedding of $\mathbf{X}$, β is a positive parameter and $\mathbf{W}$ is an affinity matrix which contains geometry characteristics of data. In this paper, we build $\mathbf{W}$ by using the same strategy as LLE does. We assume that each data sample together with its k nearest neighbors lies on a locally linear manifold and it can be reconstructed by a linear combination of its k nearest neighbors. Thus, $\mathbf{W}$ is a matrix with W_{ij} having the weight if the ith and jth sample are neighbors, and 0 otherwise.

As we can see, the first term of (6) is the same as the manifold learning term in LLE which yields manifold smoothness and the second term is a residue which yields the linear approximation fitness. By using the parameter β, we can balance these two different terms.

In order to remove an arbitrary scaling factor in the projection, we need to impose some kinds of constraints. Here, we do not use the same constraint as NPE does, because the orthogonality of the projected data may make them tend to be different from one another. To better preserve the overall geometry of data, we employ a different orthogonality strategy. We enforce the projection to be orthogonal. Thus, the constraint can be written as follows:

$$\mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{7}$$

Thus, our objective function can be reformulated as follows:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I},\mathbf{F},\mathbf{W}\mathbf{1}=\mathbf{1}} \|\mathbf{F} - \mathbf{F}\mathbf{W}^T\|_F^2 + \beta\|\mathbf{P}^T\mathbf{X} - \mathbf{F}\|_F^2 \tag{8}$$

By solving (8), we can obtain the projection matrix $\mathbf{P}$. Then, the embedding of $\mathbf{X}$ can be obtained by using a simple linear transformation as follows:

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X} \tag{9}$$

Because our method has an explicit linear mapping between the high-dimensional data and its corresponding low-dimensional embedding, it does not suffer from the out-of-sample problem as LLE does.

3.2 Optimization

We introduce an efficient algorithm to tackle problem (8) alternatively and iteratively.

The first step is to find the k nearest neighbors for each data sample.

The second step is to fix $\mathbf{F}$, $\mathbf{P}$ and solve $\mathbf{W}$. Then, we need to solve the following subproblem:

$$\min_{\mathbf{W}\mathbf{1}=\mathbf{1}} \|\mathbf{F} - \mathbf{F}\mathbf{W}^T\|_F^2 \tag{10}$$

This is the same problem as (2). For the details about how to solve the above minimization problem, please refer to [Roweis and Saul, 2000].

The third step is to fix $\mathbf{F}$, $\mathbf{W}$ and solve $\mathbf{P}$. Taking derivative of (8) w.r.t. $\mathbf{F}$ and setting it to zero, we have:

$$\mathbf{F} = \mathbf{P}^T\mathbf{B} \tag{11}$$

where,

$$\mathbf{B} = \beta\mathbf{X}[(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W}) + \beta\mathbf{I}]^{-1} \tag{12}$$

Then, bringing (11) back to (8), we have:

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} tr(\mathbf{P}^T\mathbf{Q}\mathbf{P}) \tag{13}$$

where,

$$\begin{aligned}\mathbf{Q} =&[\mathbf{B}(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})\mathbf{B}^T \\ &+ \beta(\mathbf{X}\mathbf{X}^T - \mathbf{X}\mathbf{B}^T - \mathbf{B}\mathbf{X}^T + \mathbf{B}\mathbf{B}^T)]\end{aligned} \tag{14}$$

According to Ky Fan′s Theorem [H, 2007], given the value of d which denotes the dimension of the output data, the optimal value of (13) is $\sum_{i=1}^{d}\lambda_i$, where $\lambda_i(i = 1, 2, ..., d)$ are

the first d smallest eigenvalue of $\mathbf{Q}$. The columns of $\mathbf{P}$, i.e $\mathbf{p}_i(i = 1, 2, ..., d)$, are the smallest eigenvectors of $\mathbf{Q}$. Therefore, the projection matrix $\mathbf{P}$ can be calculated by eigenvalue decomposition.

The forth step is to fix $\mathbf{P}$, $\mathbf{W}$ and solve $\mathbf{F}$. This problem can be solved by (11) and (12).

We iteratively and alternatively update $\mathbf{W}$, $\mathbf{P}$ and $\mathbf{F}$ according to the last three steps. The method can be achieved by **Algorithm 1**.

It is worth noting that in this paper, we initialize $\mathbf{P}$ by $\mathbf{I}$ and $\mathbf{F}$ by $\mathbf{X}$. By using this method, (8) becomes the following function:

$$\min_{\mathbf{W1=1}} \|\mathbf{X} - \mathbf{X}\mathbf{W}^T\|_F^2 \tag{15}$$

which is same as LLE of learning an affinity matrix. In this case, we can obtain a $\mathbf{W}$ which contains the geometric structure of the original data. This ensures us to obtain a low-dimensional embedding which maintains the original geometric structure.

Algorithm 1: The Algorithm of Solving (8)

Input: Training data $\mathbf{X} \in \mathbb{R}^{m\times n}$, parameter β, the reduction dimension d, the number of nearest neighbors k.

Initialize $\mathbf{P} = \mathbf{I}$ and $\mathbf{F} = \mathbf{X}$;

while *not converge* **do**

1. Update $\mathbf{W}$ by solving (10);
2. Update $\mathbf{P}$. The columns of the updated $\mathbf{P}$ are the first d eigenvectors of $\mathbf{Q}$ corresponding to the first d smallest eigenvalues, where $\mathbf{Q}$ can be calculate by (14);
3. Update $\mathbf{F}$ by (11).

Output: $\mathbf{P} \in \mathbb{R}^{m\times d}$

3.3 Denosing LLE

An interesting finding is that by making $\mathbf{P}$ to be an identity matrix in our objective function, it can be transformed into denoising LLE (DLLE). DLLE can learn embedding manifold structure from high-dimensional data which contains noise. In this case, our objective function can be written as follows:

$$\min_{\mathbf{F},\mathbf{W1=1}} \|\mathbf{F} - \mathbf{F}\mathbf{W}^T\|_F^2 + \beta\|\mathbf{X} - \mathbf{F}\|_F^2 \tag{16}$$

Actually, in many cases, the obtained data $\mathbf{X}$ always contain some noise. It may destroy the manifold structure of data to some extent. The traditional LLE always can not capture the manifold structure of data very well. In fact, $\mathbf{F}$ in (16) can be viewed as an approximation to $\mathbf{X}$. By choosing an appropriate β, we can make $\mathbf{F}$ be the data which has better manifold structure compared to $\mathbf{X}$. By using this method, we can obtain an affinity matrix $\mathbf{W}$ which contains a more accurate geometric structure of data. Thus, we can learn a better embedding than the original LLE.

The method of solving (16) is almost like the method of solving (8). We also tackle (16) in an alternative and iterative way:

The first step is to find the k nearest neighbors for each data sample.

The second step is to fix $\mathbf{F}$ and solve $\mathbf{W}$. Then, we need to solve the following subproblem:

$$\min_{\mathbf{W1=1}} \|\mathbf{F} - \mathbf{F}\mathbf{W}^T\|_F^2 \tag{17}$$

The third step is to fix $\mathbf{W}$ and solve $\mathbf{P}$. Taking derivative of (16) w.r.t. $\mathbf{F}$ and setting it to zero, we have:

$$\mathbf{F} = \beta\mathbf{X}[(\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W}) + \beta\mathbf{I}]^{-1} \tag{18}$$

We iteratively and alternatively update $\mathbf{W}$ and $\mathbf{F}$ according to the last two steps. To solve (16), we initialize $\mathbf{F}$ by $\mathbf{X}$. The final embedding of DLLE can be obtained by solving (3) after we obtain $\mathbf{W}$.

3.4 Convergence Analysis

As the methods of solving (8) and (16) are almost the same, here we only analyse the convergence of **Algorithm 1**.

As seen above, we solve the problem (8) in an alternative way. The following proposition shows that the proposed algorithm can find a globally optimal solution in each iteration.

Proposition 1 *When $\mathbf{F}$ and $\mathbf{P}$ are fixed, the optimal $\mathbf{W}$ can be obtained by our method. The same conclusion can also be obtained by $\mathbf{F}$ and $\mathbf{P}$*

Proof Briefly, when $\mathbf{F}$ and $\mathbf{P}$ are fixed, the optimization problem (8) is equivalent to the problem (10). Roweis and Saul in [Roweis and Saul, 2000] gave a detailed description on the method to find the global solution of the problem (10). In this paper, we adopt the same way to solve (10). When $\mathbf{W}$ and $\mathbf{F}$ are fixed, solving the problem (8) is equivalent to solving (13). According to Ky Fan′s Theorem [H, 2007], if the value of d which denotes the dimension of the output data is given, the optimal solution of (13) can be obtained by finding the first d smallest eigenvalues and its corresponding eigenvectors of $\mathbf{Q}$. When $\mathbf{W}$ and $\mathbf{P}$ are fixed, it is obviously that the problem (8) is a convex problem without constraint. Its global optimal solution can be easily obtained by taking derivative of (8) w.r.t. F and setting it to zero. □

The above proposition only shows that the proposed algorithm could find a global optimization in each iteration. Based on this result, we will propose another proposition.It indicates that our iteration can monotonically decrease the objective function of the problem (8).

Proposition 2 *The proposed iterative procedure can monotonically decrease the objective function of the problem (8) in each iteration.*

Proof Assume that we have derived $\mathbf{W}^{(s)}$ and $\mathbf{P}^{(s)}$ in the sth iteration. We now update $\mathbf{F}$. The following results hold:

$$\begin{aligned}\{\mathbf{F}^{(s)}\} = \operatorname*{argmin}_{\mathbf{F}} &\|\mathbf{F} - \mathbf{F}(\mathbf{W}^{(s)})^T\|_F^2 \\ &+ \beta\|(\mathbf{P}^{(s)})^T\mathbf{X} - \mathbf{F}\|_F^2\end{aligned} \tag{19}$$

In the $(s + 1)$th iteration, we fix $\mathbf{F}$, $\mathbf{P}$ as $\mathbf{F}^{(s)}$, $\mathbf{P}^{(s)}$,respectively, and update $\mathbf{W}$ by solving (8). We have the

following results:

$$\{\mathbf{W}^{(s+1)}\} = \underset{\mathbf{W1=1}}{\operatorname{argmin}} \|\mathbf{F}^{(s)} - \mathbf{F}^{(s)}\mathbf{W}^T\|_F^2 + \beta\|(\mathbf{P}^{(s)})^T\mathbf{X} - \mathbf{F}^{(s)}\|_F^2 \tag{20}$$

Similarly, when we fix $\mathbf{W}$, $\mathbf{F}$ as $\mathbf{W}^{(s+1)}$, $\mathbf{F}^{(s)}$, respectively, and optimize $\mathbf{P}$ by solving (8), the following result holds:

$$\{\mathbf{P}^{(s+1)}\} = \underset{\mathbf{P}^T\mathbf{P}=\mathbf{I}}{\operatorname{argmin}} \|\mathbf{F}^{(s)} - \mathbf{F}^{(s)}(\mathbf{W}^{(s+1)})^T\|_F^2 + \beta\|\mathbf{P}^T\mathbf{X} - \mathbf{F}^{(s)}\|_F^2 \tag{21}$$

Then, when we fix $\mathbf{W}$, $\mathbf{P}$ as $\mathbf{W}^{(s+1)}$, $\mathbf{P}^{(s+1)}$, respectively, and optimize $\mathbf{F}$ by solving (8), the following result holds:

$$\{\mathbf{F}^{(s+1)}\} = \underset{\mathbf{F}}{\operatorname{argmin}} \|\mathbf{F} - \mathbf{F}(\mathbf{W}^{(s+1)})^T\|_F^2 + \beta\|(\mathbf{P}^{(s+1)})^T\mathbf{X} - \mathbf{F}\|_F^2 \tag{22}$$

As we known, $\{\mathbf{F}^{(s)}, \mathbf{W}^{(s)}, \mathbf{P}^{(s)}\}$ and $\{\mathbf{F}^{(s+1)}, \mathbf{W}^{(s+1)}, \mathbf{P}^{(s+1)}\}$ are both feasible solution to the problem (8). More importantly, recalling the results in Proposition 1, we have the following inquality:

$$\begin{aligned} &G(\mathbf{F}^{(s+1)}, \mathbf{W}^{(s+1)}, \mathbf{P}^{(s+1)}) \\ &\leq G(\mathbf{F}^{(s+1)}, \mathbf{W}^{(s+1)}, \mathbf{P}^{(s)}) \\ &\leq G(\mathbf{F}^{(s+1)}, \mathbf{W}^{(s)}, \mathbf{P}^{(s)}) \\ &\leq G(\mathbf{F}^{(s)}, \mathbf{W}^{(s)}, \mathbf{P}^{(s)}) \end{aligned} \tag{23}$$

where

$$\begin{aligned} &G(\mathbf{F}, \mathbf{W}, \mathbf{P}) = \|\mathbf{F} - \mathbf{F}\mathbf{W}^T\|_F^2 + \beta\|\mathbf{P}^T\mathbf{X} - \mathbf{F}\|_F^2 \\ &s.t. \mathbf{P}^T\mathbf{P} = \mathbf{I}, \mathbf{W1} = \mathbf{1} \end{aligned} \tag{24}$$

It indicates the decrease of objective function during iteration. □

4 Experiment

In this section, we perform extensive experiments to test the performance of FONPE and DLLE. We compare FONPE with several manifold learning based linear dimension reduction methods, i.e. PCA, LPP and NPE, on synthetic data and together with the other two recently proposed dimension reduction methods, i.e. FME [Nie *et al.*, 2010] and ULGE-K [Feiping Nie and Li, 2017], on real data. As for DLLE, we only compare its performance with LLE on synthetic data since it suffers out-of-sample problem.

4.1 Synthetic Data

We first evaluate the performance of FONPE on synthetic data. In this paper, we employ two well known synthetic data from [Roweis and Saul, 2000]: the s-curve and the swissroll. We compare FONPE with PCA, LPP and NPE. Data points in **Fig 1**(b) ($N = 2000$), **Fig 2**(b) ($N = 2000$) are randomly sampled from the manifold shown in **Fig 1**(a), **Fig 2**(a), respectively. The affinity graphs of LPP, NPE and FONPE are all constructed using $k = 6$ nearest neighbor points.

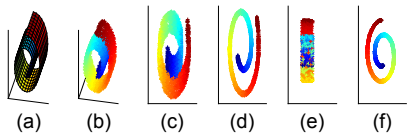

Figure 1: Results on swissroll by linear dimensionality reduction methods. (a): swissroll; (b): random points of swissroll; (c): PCA; (d): LPP; (e): NPE; (f): FONPE.

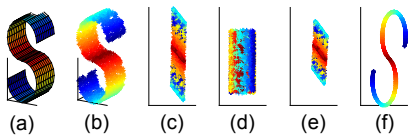

Figure 2: Results on s-curve by linear dimensionality reduction methods. (a): s-curve; (b): random points of s-curve; (c): PCA; (d): LPP; (e): NPE; (f): FONPE.

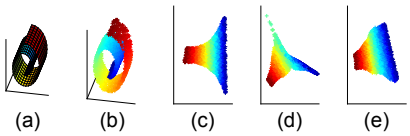

Figure 3: Results on swissroll by non linear dimensionality reduction methods. (a): swissroll; (b): random points of swissroll; (c): LLE result on data without noise; (d): LLE result on data with noise; (e): DLLE result on data with noise.

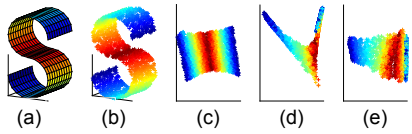

Figure 4: Results on s-curve by non linear dimensionality reduction methods. (a): s-curve; (b): random points of s-curve; (c): LLE result on data without noise; (d): LLE result on data with noise; (e): DLLE result on data with noise.

As we can see from **Fig 1** and **Fig 2**, all these three manifold learning based linear dimension reduction methods perform much better than PCA since they can share many of the data representation properties of non-linear techniques and preserve local information of data which is indicated by the color shading. Moreover, it can be seen, FONPE performs better than LPP and NPE since it can preserve global geometric characteristics as well.

Then, we evaluate the effectiveness of DLLE also on s-curve and swissroll. Different from above, when building training data, we add 1 by 2000 uniformly distributed random noise into each dimension of s-curve and swissroll.

As can be seen, the structure of the low-dimensional data obtained from data with noise by DLLE is much closer to that of LLE obtained from data without noise. This indicates FONPE can better capture the manifold structure of data with noise compared to LLE.

Table 1: Datasets Description

Data	# Samples	# Features	# Classes
YaleB	2414	1024	38
PIE	3329	1024	68
ORL	400	1024	40
UMIST	9145	10304	102

4.2 Face Recognition

In this part, we evaluate the performance of FONPE on four benchmark data sets for the task of face recognition. The data descriptions are summarized in **Table 1**. The YaleB data set [Georghiades *et al.*, 2001] contains 2414 near frontal images from 38 persons under different illuminations. Each image is resized to 32×32 and the pixel value is used as feature representation. The PIE data set [Sim *et al.*, 2002] contains 68 individuals with 41368 face images as a whole. We used 24 face images for each individual in our experiment. The ORL data set [Samaria and Harter, 1994] cosists of 400 face images of 40 people. The images are captured at different times and have different variations.Each image is resized to 32×32 and the pixel value is used as feature representation. The UMIST data set [Graham and Allinson, 1995] contains 20 people under different poses.

For each data set, PCA is used for preprocessing. In our experiments, we keep 95 percent information in the sense of reconstruction error. After that, 50 percent images of each individual are randomly selected with labels to form the training set and the rest of the data set is used as the testing set. The training samples are used to learn a projection. The testing samples are then projected into the reduced space. Recognition is performed using a SVM classifier. We utilize the linear kernel with the parameter $C = 1$. Generally, how to select a proper k directly from data is an open problem in manifold learning. In this paper, we run LPP, NPE and FONPE when $k = 3, 5, 7, 9, 11, 13$, respectively. After that, we choose the optimal recognition rate among these experiments as their final results. For FME, we set each parameter to $\{10^{-9}, 10^{-6}, 10^{-3}, 10^{0}, 10^{3}, 10^{6}, 10^{9}\}$, and then we report the best accuracy as its final result. As for ULGE-K, we set its parameters as suggested in [Feiping Nie and Li, 2017].

For FONPE, it has an extra parameter, i.e. β, which can balance the manifold smoothness and the linear projection fitness. In the limiting case, namely, $\lambda \to +\infty$, the penalty term can force data mapping into a complete linear space which yields a hard projection constraint as NPE does. For more general case, we demonstrate the recognition rate result with different β on UMIST database in **Fig 5**. The parameter is selected from $10^{-7}, 10^{-6}, ..., 10^{17}$. As can be seen from **Fig 5**, the recognition rates have slight variations when k ranges from 10^{-1} to 10^{12}. This shows FONPE is quite robust to the value of β in a large range. Besides, it can be seen that when the value of β is too large, FONPE yields poor performance. This clarifies that the methods using the relaxed projection strategy can deal with non-linear data better comparing with methods using hard projection strategy. Similar properties exist on other databases.

The recognition rates versus the dimension of each method on these databases are shown in **Fig 6**. As we can see, our approach outperforms other methods significantly.

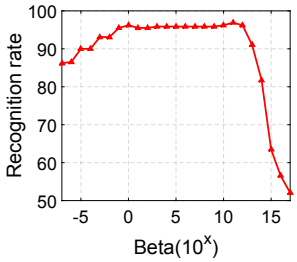

Figure 5: Recognition rates versus the parameter Beta on UMIST database.

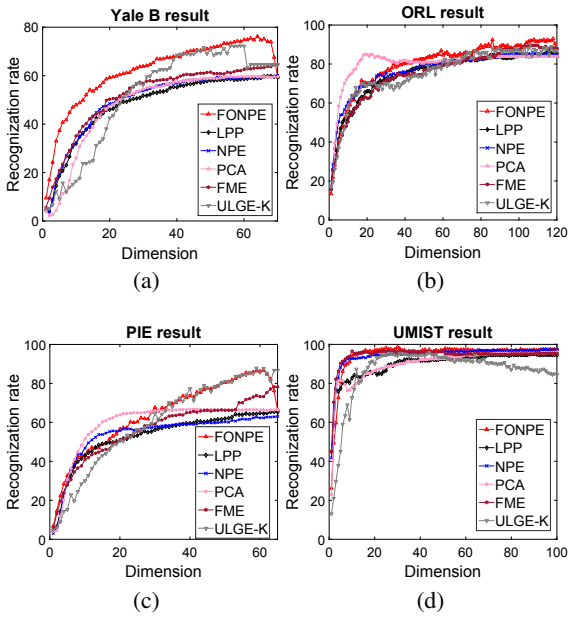

Figure 6: Recognition rates versus the dimension of each method on four benchmark databases.

5 Conclusion

In this paper, we have proposed a novel linear dimensionality reduction algorithm called Flexible Orthogonal Neighbor Preserving Embedding. Though our algorithm is a linear method, it shares many non-linear properties of LLE since it is based on the same variational principle as LLE. By using a projection residue penalty term, our model can fit non-linear structure of data flexibly. Extensive experiments on synthetic data and four benchmark data sets showed our method can outperform other three kinds of state-of-the-art linear dimensionality reduction methods. Besides, our model can also be transformed into denoising LLE which is a non-linear manifold learning method that can tolerate the noise of data to some extent. The effectiveness of DLLE also was demonstrated on two well known synthetic data by experiments.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grants 61522207 and 61473231.

References

[Belhumeur *et al.*, 1997] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Bengio *et al.*, 2004] Yoshua Bengio, Jean-François Paiement, Pascal Vincent, Olivier Delalleau, Nicolas Le Roux, and Marie Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in Neural Information Processing Systems*, 16:177–184, 2004.

[Elbagoury and Ibrahim, 2016] Ahmed Elbagoury and Rania Ibrahim. Ebek: Exemplar-based kernel preserving embedding. In *The International Joint Conference on Artificial Intelligence IJCAI*, 2016.

[Feiping Nie and Li, 2017] Wei Zhu Feiping Nie and Xuelong Li. Unsupervised large graph embedding. In *AAAI Conference on Artificial Intelligence.*, 2017.

[Georghiades *et al.*, 2001] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[Graham and Allinson, 1995] Daniel B. Graham and Nigel Allinson. Characterizing virtual eigensignatures for general purpose face recognition. *Journal of Nursing Management*, 3(2):87–91, 1995.

[H, 2007] William H. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[He and Niyogi, 2004] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, pages 153–160. MIT Press, 2004.

[He *et al.*, 2005] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 1208–1213. IEEE, 2005.

[Huang *et al.*, 2015] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection metric learning on grassmann manifold with application to video based face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[Kokiopoulou and Saad, 2007] Effrosyni Kokiopoulou and Yousef Saad. Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2143–2156, 2007.

[Li and Tao, 2012] Jun Li and Dacheng Tao. On preserving original variables in bayesian pca with application to image analysis. *IEEE Transactions on Image Processing*, 21(12):4830–4843, 2012.

[Liu *et al.*, 2002] Qingshan Liu, Rui Huang, Hanqing Lu, and Songde Ma. Face recognition using kernel-based fisher discriminant analysis. In *IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 197–201. IEEE, 2002.

[Nie *et al.*, 2010] Feiping Nie, Dong Xu, Wai Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932, 2010.

[Roweis and Saul, 2000] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Samaria and Harter, 1994] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision.*, pages 138 – 142. IEEE, 1994.

[Schölkopf *et al.*, 1998] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[Sim *et al.*, 2002] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database. In *IEEE International Conference on Automatic Face and Gesture Recognition.*, pages 46 – 51. IEEE, 2002.

[Tenenbaum *et al.*, 2000] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[Turk and Pentland, 1991] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991.

[Wang *et al.*, 2015a] Hua Wang, Feiping Nie, and Heng Huang. Learning robust locality preserving projection via p-order minimization. In *AAAI Conference on Artificial Intelligence.*, pages 3059–3065, 2015.

[Wang *et al.*, 2015b] Rong Wang, Feiping Nie, Xiaojun Yang, Feifei Gao, and Minli Yao. Robust 2dpca with non-greedy-norm maximization for image analysis. *IEEE transactions on cybernetics*, 45(5):1108–1112, 2015.

[Xu *et al.*, 2016] Jinglin Xu, Junwei Han, and Feiping Nie. Discriminatively embedded k-means for multi-view clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.