Projective Unsupervised Flexible Embedding with Optimal Graph

Wei Wang[1]
wei.wang@unitn.it
Yan Yan[1]
yan.yan@unitn.it
Feiping Nie[2]
feipingnie@gmail.com
Xavier Alameda Pineda[1]
xavier.alamedapineda@unitn.it
Shuicheng Yan[3]
eleyans@nus.edu.sg
Nicu Sebe[1]
niculae.sebe@unitn.it

[1] DISI,
University of Trento
Trento, Italy

[2] OPTIMAL,
Northwestern Polytechnical University,
Xi'an, China

[3] ECE,
National University of Singapore,
Singapore

Abstract

Graph based dimensionality reduction techniques have been successfully applied to clustering and classification tasks. The fundamental basis of these algorithms is the constructed graph which dominates their performance. Usually, the graph is defined by the input affinity matrix. However, the affinity matrix is sub-optimal for dimension reduction as there is much noise in the data. To address this issue, we propose the projective unsupervised flexible embedding with optimal graph (PUFE-OG) model. We build an optimal graph by adjusting the affinity matrix. To tackle the out-of-sample problem, we employ a linear regression term to learn a projection matrix. The optimal graph and projection matrix are jointly learned by integrating the manifold regularizer and regression residual into a unified model. An efficient algorithm is derived to solve the challenging model. The experimental results on several public benchmark datasets demonstrate that the presented PUFE-OG outperforms other state-of-the-art methods.

1 Introduction

With the popularity of social networks, numerous data are generated everyday, such as images, videos, texts, *etc*. But usually these rich resources are not well organized and bring us big challenges to retrieve them. To address this issue, the most straightforward solution is to assign semantic labels to these data. However, the data we deal with (*e.g.*, images) are always represented by high-dimensional vectors. To better organize and represent the data, many feature selection [10] and probabilistic generative models with feature selection/missing data [2, 8], dimensionality reduction techniques, and clustering algorithms were proposed and benefited many related applications, such as image annotation [7], multi-view

clustering [29], human action recognition [32], human face aging [25], human headpose estimation [33], attribute detection [24, 26] and video annotation [14].

Many dimensionality reduction algorithms seek for an optimal projection matrix to map the data. These algorithms are always linked to clustering and classification algorithms and they work as an intermediate step. For instance, many works perform K-Means clustering after doing Principle Component Analysis (PCA), and spectral clustering (SC) uses Laplacian eigenmap to project the data which is followed by a K-Means clustering. Recently, the denoising auto-encoder [23] has been employed for dimensionality reduction before learning the classifiers. Auto-encoder is very similar to PCA when it only has one linear hidden layer. One main reason to do dimensionality reduction is that the low-dimensional embeddings can better represent the data and accelerate the clustering or classification process.

When the labels are available, the most popular dimensionality reduction algorithm is linear discriminant analysis (LDA). It has excellent performance as LDA utilizes discriminant information to learn the subspace. In addition, simultaneously performing clustering and subspace learning can yield even better clustering result. *Ye et al.* [35] proposed an effective discriminating K-Means (DisKmeans) algorithm by integrating LDA and K-Means. *Yang et al.* [34] proposed a local discriminant model and global integration (LDMGI) model by integrating manifold subspace learning and clustering into a unified framework. However, labeled data are often very costly to obtain.

When the labels are unavailable, unsupervised dimensionality reduction methods become the only choice. For example, PCA is widely used because of its simplicity and efficiency. The unsupervised graph based dimensionality reduction methods (*e.g.,* Neighbor Preserving Embedding [8], Locality Preserving Projections [17]) usually outperform PCA. This is because these methods take advantage of manifold information. Many graph based dimensionality reduction methods have been explored, such as locally linear embedding (LLE) [19], Laplacian eigenmap (LE) [1], and ISOMAP [22]. However, these methods suffer from out-of-sample problem. They can not map the new data points that are not included in the training set. To tackle this problem, many works [15] integrated the manifold regularizer with the ridge regression loss into the subspace learning framework. Similar to other manifold learning algorithms, their performance is also controlled by the graph constructed by the fixed affinity matrix, which might lead to a sub-optimal result [16]. To address this issue, we propose a projective unsupervised flexible embedding with optimal graph (PUFE-OG) framework. Instead of utilizing the fixed affinity matrix to preserve the manifold structure, we construct an optimal graph by adapting the affinity matrix for subspace learning. It is worthwhile to highlight the following contributions of our work: (1) The proposed framework exploits an unsupervised flexible embedding for high-dimensional data. The framework automatically adjusts the graph to an optimal graph based on the local manifold structure; (2) The optimal graph seamlessly integrates manifold learning with subspace learning into a unified framework. Besides, our framework has closed form solution when update its nested variables iteratively.

2 Related Work

Dimensionality Reduction Techniques: To better organize and represent data, various supervised and unsupervised dimensionality reduction methods are proposed. As a popular unsupervised dimensionality reduction method, PCA considers variance as the most important metric and aims to minimize the data reconstruction error by minimizing the variance.

However, PCA can not guarantee that the learned subspace has discriminant power and it may not improve the clustering result. Different from PCA, spectral clustering [6] is a graph based clustering method. SC utilizes LE for dimensionality reduction based on the affinity matrix. SC is more robust compared with PCA as LE preserves the local manifold structure of the data points. Thus, the data are tailored to low-dimensional space and it accelerates K-Means clustering. To further improve the performance of SC, many works focused on learning a better graph. *Liu et al.* [11] proposed to compress the original large scale graph into a sparse bipartite graph. *Shao et al.* [20] adopted deep networks in SC. But SC can not cluster the data which are not included in the training set as LE can not map new data to the low-dimensional space. Thus, SC suffers from the out-of-sample problem.

In the supervised settings, Fisher LDA is widely used for subspace learning. LDA [12] minimizes the within-cluster scatter matrix while maximizes the between-cluster matrix. Then the cluster centers will be as far as possible while the within-cluster data will be as close as possible. Thus, the learned subspace has greater discriminantive power compared with PCA. Dictionary learning can also be employed as a clustering method. *Ramirez et al.* [18] proposed training a dictionary for each cluster. A cross-incoherence promoting term was integrated into the dictionary learning framework. This term encourages the dictionaries to be as independent as possible, which can lead to greater discriminative power. But the performance of these methods highly relies on the sufficiency of the labeled training data which is very time-consuming to obtain.

Manifold Learning: When labels are unavailable, graph based dimensionality reduction methods can usually provide a better low-dimensional representation and outperform PCA by utilizing the local manifold structure. Given a dataset $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_n]$, where n is the number of samples and $\boldsymbol{x}_i\in\mathbb{R}^d,\forall i$, manifold information is preserved in the undirected graph $\boldsymbol{G}=\{\boldsymbol{X},\boldsymbol{A}\}$ with data matrix $\boldsymbol{X}$ and affinity matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$. The $(i,j)^{th}$ element $A_{i,j}$ in $\boldsymbol{A}$ denotes the connectivity between sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $A_{i,j}$ can be viewed as the weight of the edge in the graph. Let us denote the low-dimensional representations of $\boldsymbol{x}_i$ as $\boldsymbol{f}_i\in\mathbb{R}^{1\times c}$, where c is the lower dimension and $\boldsymbol{f}_i$ is a row vector. The goal of manifold learning is to construct another graph $\widehat{\boldsymbol{G}}=\{\boldsymbol{F},\widehat{\boldsymbol{A}}\}$, where $\boldsymbol{F}\in\mathbb{R}^{n\times c}$ is the low-dimensional representation of $\boldsymbol{X}$. Ideally, we should have $\widehat{\boldsymbol{A}}=\boldsymbol{A}$, which means the connectivity between sample $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ should be the same as the connectivity between sample $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Gaussian kernel is widely used to calculate the affinity matrix. $\boldsymbol{A}$ is defined as follows:

$$A_{i,j}=\begin{cases}\exp(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}), & \text{if } \boldsymbol{x}_i,\boldsymbol{x}_j \text{ are } k \text{ nearest neighbors.}\\ 0, & \text{otherwise.}\end{cases}$$

σ controls the weight of the edge. Most manifold learning methods (*e.g.,* LE [1]) represent the manifold structure with this graph, and require the low-dimensional representations to preserve the same manifold structure by employing a manifold regularizer as follows:

$$\min_{\boldsymbol{F},\boldsymbol{F}^T\boldsymbol{D}\boldsymbol{F}=\boldsymbol{I}} \boldsymbol{F}^T\boldsymbol{L}_{\boldsymbol{A}}\boldsymbol{F} \tag{1}$$

Eq. (1) is the objective function of LE, $\boldsymbol{D}$ is a diagonal matrix with the element $D_{i,i}=\sum_j A_{i,j}$. The graph Laplacian matrix $\boldsymbol{L}_{\boldsymbol{A}}=\boldsymbol{D}-\boldsymbol{A}$.

To better understand the intuition why this term can preserve the manifold structure, we reformulate Eq. (1) as Eq. (2) ($\boldsymbol{A}$ is symmetric).

$$\mathrm{Tr}(\boldsymbol{F}^T\boldsymbol{L}_A\boldsymbol{F})=\frac{1}{2}\sum_{i,j=1}^{n}\|\boldsymbol{f}_i-\boldsymbol{f}_j\|_2^2 A_{i,j} \tag{2}$$

where $\boldsymbol{A}$ can be viewed as a similarity matrix which measures the similarity between every two data points. A larger $A_{i,j}$ implies a more similar pair of samples. The closer the two samples are, the larger $A_{i,j}$ will become. Correspondingly, $\|\boldsymbol{f}_i-\boldsymbol{f}_j\|_2^2$ will become smaller to keep the balance.

To deal with the out-of-sample problem and preserve the manifold smoothness, [21] proposed a LapRLS/L method, which exploited ridge regression to learn the subspace and a manifold regularizer term to make use of the local geometry information. Let $\boldsymbol{W}\in\mathbb{R}^{d\times c}$ be the projection matrix and $\boldsymbol{b}\in\mathbb{R}^c$ be the bias term. The ridge regression function is defined as

$$\frac{1}{n}\|\boldsymbol{X}^T\boldsymbol{W}+\mathbf{1}\boldsymbol{b}^T-\boldsymbol{Y}\|_F^2+\gamma\|\boldsymbol{W}\|_F^2 \tag{3}$$

where $\mathbf{1}\in\mathbb{R}^n$, and with all its entries being 1; $\boldsymbol{Y}$ is a binary class assignment matrix. $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,...,\boldsymbol{y}_n]\in\mathbb{R}^{n\times c}$, where $\boldsymbol{y}_i\in\{0,1\}^{c\times 1}$ is the class indicator vector for $\boldsymbol{x}_i$. The j-th element of $\boldsymbol{y}_i$ is 1 if $\boldsymbol{x}_i$ belongs to the j-th class, and 0 otherwise. The second term $\gamma\|\boldsymbol{W}\|_F^2$ is a regularizer to prevent the entries in the projection matrix $\boldsymbol{W}$ becoming too large. The manifold regularizer is

$$\lambda_I Tr(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}_{\boldsymbol{A}}\boldsymbol{X}^T\boldsymbol{W}) \tag{4}$$

where $\boldsymbol{X}^T\boldsymbol{W}$ is similar to $\boldsymbol{F}$ in Eq. (1). After combining Eq. (3) and Eq. (4), the projection matrix and manifold structure can be jointly learned. Then the low-dimensional representation of new data can be obtained using the linear projection function $h(\boldsymbol{x})=\boldsymbol{x}^T\boldsymbol{W}+\boldsymbol{b}^T$. However, this model still requires labeled training data. *Nie et al.* [15] further extended the LapRLS/L to deal with the unlabeled data. The performance of these graph based methods which integrate manifold regularizer is dominated by the graph.

Thus, learning an effective graph is very essential. However, the graph is always *fixed* by the affinity matrix $\boldsymbol{A}$. The affinity matrix is a hard constraint, and the learned low-dimensional representations are restricted to have the same neighbourhood relationship ($\widehat{\boldsymbol{A}}=\boldsymbol{A}$) without any violation. But the neighbourhood relationship may be inaccurate as there is much noise in the data. To address this issue, we propose constructing an optimal graph by adjusting the neighbourhood relationships for projective unsupervised *flexible* manifold embedding which can learn the projection matrix and the optimal graph simultaneously.

3 Projective Unsupervised Flexible Embedding Framework (PUFE)

To take advantage of manifold information and solve the out-of-sample problem, a unified PUFE framework is proposed by integrating the manifold regularizer with the ridge regression terms. The PUFE is as follows:

$$\min_{\substack{\boldsymbol{F}\in\mathbb{R}^{n\times c},\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}\\ \boldsymbol{W}\in\mathbb{R}^{d\times c},\boldsymbol{b}\in\mathbb{R}^c}} Tr(\boldsymbol{F}^T\boldsymbol{L}_A\boldsymbol{F})+\mu\|\boldsymbol{X}^T\boldsymbol{W}+\mathbf{1}\boldsymbol{b}^T-\boldsymbol{F}\|_F^2+\gamma\|\boldsymbol{W}\|_F^2 \tag{5}$$

The functionality of each term is described as follows:

• The first term, $Tr(\boldsymbol{F}^T\boldsymbol{L}_A\boldsymbol{F})$, is a manifold regularizer as in Eq. (1), which is in charge of maintaining the manifold smoothness of the data.

• The second term denotes the ridge regression error. This linear regression model learns

a projection matrix $\boldsymbol{W}$ to project samples to a low-dimensional space. Usually, the dimensionality reduction methods, such as PCA, LDA, use a linear function to project data matrix $\boldsymbol{X}$: $\boldsymbol{F}=\boldsymbol{X}^T\boldsymbol{W}$, and $\boldsymbol{F}$ lies in the space spanned by $\boldsymbol{X}$. But the second term relaxes the rigid form $\boldsymbol{F}=\boldsymbol{X}^T\boldsymbol{W}$ to $\boldsymbol{F}\approx\boldsymbol{X}^T\boldsymbol{W}+\mathbf{1}\boldsymbol{b}^T$ by minimizing $\|\boldsymbol{X}^T\boldsymbol{W}+\mathbf{1}\boldsymbol{b}^T-\boldsymbol{F}\|_F^2$ which is more flexible. Moreover, this term allows its interaction with manifold regularizer through the variable $\boldsymbol{F}$.

• The third term is a regularizer which is always combined with the ridge regression term. Its function is to prevent overfitting. The weights, μ, γ balance the importance of each term.

Optimal Graph Construction: As a graph based method, PUFE tackles the out-of-sample problem, and the affinity matrix $\boldsymbol{A}$ has a significant influence on its performance. The affinity matrix $\boldsymbol{A}$ can be viewed as a probability matrix. If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar to each other, a large value will be assigned to $A_{i,j}$ and it represents that $\boldsymbol{x}_i$ has a higher possibility to be the neighbor of $\boldsymbol{x}_j$.

In Eq. (5), graph $\boldsymbol{A}$ is a hard constraint, and $\boldsymbol{F}$ is forced to embed into the fixed graph $\boldsymbol{A}$ by the penalty term $Tr(\boldsymbol{F}^T\boldsymbol{L}_{\boldsymbol{A}}\boldsymbol{F})$ in the loss function. However, $\boldsymbol{A}$ is not an optimal graph for dimension reduction, and it may result in a sub-optimal low-dimensional representation. In this paper, we aim to construct an optimal graph $\boldsymbol{S}$ ($S_{i,j}\geq 0$) based on the affinity matrix $\boldsymbol{A}$. Similar to $\boldsymbol{A}$, the entry $S_{i,j}$ in $\boldsymbol{S}$ represents the probability that $\boldsymbol{x}_j$ being the neighbor of $\boldsymbol{x}_i$. Thus, we have $\sum_j S_{i,j}=1$, which can be formulated as $\boldsymbol{S}\mathbf{1}=\mathbf{1}$. $\boldsymbol{S}$ works as a soft constraint for $\boldsymbol{F}$. The loss $\|\boldsymbol{S}-\boldsymbol{A}\|_F^2$ can prevent $\boldsymbol{S}$ from deviating too far from $\boldsymbol{A}$. At the same time, $\boldsymbol{S}$ allows the neighborhood probability to be adjusted for dimension reduction. Then $\boldsymbol{F}$ is embedded into the optimal graph $\boldsymbol{S}$.

Let $\boldsymbol{L}_{\boldsymbol{S}}$ denote the graph Laplacian matrix, we propose the following objective function and the optimal graph for dimension reduction can be derived by solving the following function:

$$\min_{\substack{\boldsymbol{F}\in\mathbb{R}^{n\times c},\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}\\ \boldsymbol{S}\in\mathbb{R}^{n\times n},\boldsymbol{S}\mathbf{1}=\mathbf{1},\boldsymbol{S}\geq\mathbf{0}}} \|\boldsymbol{S}-\boldsymbol{A}\|_F^2+\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F}) \tag{6}$$

The graph Laplacian matrix $\boldsymbol{L}_{\boldsymbol{S}}\in\mathbb{R}^{n\times n}$ is denoted as $\boldsymbol{L}_{\boldsymbol{S}}=\boldsymbol{D}-\frac{\boldsymbol{S}+\boldsymbol{S}^T}{2}$, where $\boldsymbol{D}$ is a diagonal matrix with diagonal elements $D_{i,i}=\sum_j(S_{i,j}+S_{j,i})/2,\forall i$. Till now, we have constructed the optimal graph based on the similarity matrix. One very important advantage of this model is that the neighborhood probability can be tuned.

Projective Unsupervised Flexible Embedding with Optimal Graph (PUFE-OG): By replacing the affinity matrix $\boldsymbol{A}$ with an optimal neighborhood probability matrix $\boldsymbol{S}$. We can learn the optimal graph and projection matrix jointly. The following model is obtained after combining the optimal graph and unsupervised flexing embedding:

$$\begin{aligned}\min_{\boldsymbol{F},\boldsymbol{W},\boldsymbol{b},\boldsymbol{S}} \quad & \Big(\|\boldsymbol{S}-\boldsymbol{A}\|_F^2+\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F})\\ & +\mu\|\boldsymbol{X}^T\boldsymbol{W}+\mathbf{1}\boldsymbol{b}^T-\boldsymbol{F}\|_F^2+\gamma\|\boldsymbol{W}\|_F^2\Big)\\ \text{subject to} \quad & \boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I},\boldsymbol{F}\in\mathbb{R}^{n\times c}\\ & \boldsymbol{W}\in\mathbb{R}^{d\times c},\boldsymbol{b}\in\mathbb{R}^c\\ & \boldsymbol{S}\mathbf{1}=\mathbf{1},\boldsymbol{S}\geq\mathbf{0},\boldsymbol{S}\in\mathbb{R}^{n\times n}\end{aligned} \tag{7}$$

The functionality of each term is described as follows:

• The first term, $\|\boldsymbol{S}-\boldsymbol{A}\|_F^2$, is aimed at constructing an optimal graph $\boldsymbol{S}\in\mathbb{R}^{n\times n}$. The Frobenius norm measures the Euclidean distance between $\boldsymbol{S}$ and $\boldsymbol{A}$. This term allows the graph $\boldsymbol{S}$ to

adjust itself for the optimal graph construction while prevents it deviating too far away from the similarity matrix $\boldsymbol{A}$.

• The second term, $\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F})$, is in charge of maintaining the manifold smoothness of the data. Different from Eq. (5) whose Laplacian matrix $\boldsymbol{L}_{\boldsymbol{A}}$ is derived from the fixed affinity matrix $\boldsymbol{A}$, the Laplacian matrix $\boldsymbol{L}_{\boldsymbol{S}}$ in this term is based on a soft graph $\boldsymbol{S}$.

• The third term denotes the ridge regression residual. The forth term is a regularizer and its function is to prevent overfitting. The weights, λ, μ, γ represent the importance of each term.

Optimization of PUFE-OG: We solve the model in Eq. (7) by optimizing the variables alternatively which mainly consists of two parts.

Initialization: After setting $\boldsymbol{X}=[\boldsymbol{x}^1,\boldsymbol{x}^2,...,\boldsymbol{x}^n]$ and other parameters, $\boldsymbol{A}$ is set to affinity matrix $\boldsymbol{A}_0$. $\boldsymbol{S}_0$ is initialized according to Eq. (14) with $\boldsymbol{d}_i$ set to $\boldsymbol{0}$.

(1) Fix $\boldsymbol{S}$, update $\boldsymbol{W}$,$\boldsymbol{b}$,$\boldsymbol{F}$: When $\boldsymbol{S}$ is fixed, the original optimization model in Eq. (7) can be put in the following form,

$$\min_{\substack{\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}\\ \boldsymbol{W},\boldsymbol{b},\boldsymbol{F}\in\mathbb{R}^{n\times c}}} \left(\begin{array}{l}\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F})\\ +\mu\|\boldsymbol{X}^T\boldsymbol{W}+\boldsymbol{1}\boldsymbol{b}^T-\boldsymbol{F}\|_F^2+\gamma\|\boldsymbol{W}\|_F^2\end{array}\right) \tag{8}$$

Eq. (8) is convex with respect to $\boldsymbol{W}$,$\boldsymbol{b}$,$\boldsymbol{F}$. The proof is available in [15]. By setting the derivative with respect to $\boldsymbol{W}$ and $\boldsymbol{b}$ to $\boldsymbol{0}$, we have

$$\begin{aligned}\boldsymbol{b}&=\frac{1}{n}(\boldsymbol{F}^T\boldsymbol{1}-\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{1})\\ \boldsymbol{W}&=(\boldsymbol{X}\boldsymbol{H}_n\boldsymbol{X}^T+\frac{\gamma}{\mu}\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{H}_n\boldsymbol{F}=\boldsymbol{Q}\boldsymbol{F}\end{aligned} \tag{9}$$

where $\boldsymbol{Q}=(\boldsymbol{X}\boldsymbol{H}_n\boldsymbol{X}^T+\frac{\gamma}{\mu}\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{H}_n$.
Let $\boldsymbol{X}_n=\boldsymbol{X}\boldsymbol{H}_n$. The solution for $\boldsymbol{F}$ is

$$\boldsymbol{F}^*=\arg\min_{\boldsymbol{F},\boldsymbol{F}^T\boldsymbol{F}=\boldsymbol{I}} Tr\boldsymbol{F}^T(\lambda\boldsymbol{L}_{\boldsymbol{S}}+\mu\boldsymbol{H}_n-\mu\boldsymbol{N})\boldsymbol{F} \tag{10}$$

where

$$\boldsymbol{N}=\boldsymbol{X}_n^T(\boldsymbol{X}_n\boldsymbol{X}_n^T+\frac{\gamma}{\mu}\boldsymbol{I})^{-1}\boldsymbol{X}_n=\boldsymbol{X}_n^T\boldsymbol{X}_n(\boldsymbol{X}_n^T\boldsymbol{X}_n+\frac{\gamma}{\mu}\boldsymbol{I})^{-1}$$

A generalized eigenvalue decomposition [31] can be utilized to solve this objective function.

(2) Fix $\boldsymbol{W}$,$\boldsymbol{b}$,$\boldsymbol{F}$, update $\boldsymbol{S}$: When $\boldsymbol{W}$,$\boldsymbol{b}$,$\boldsymbol{F}$ are fixed, the original optimization model Eq. (7) can be put in the following form,

$$\min_{\boldsymbol{S}\in\mathbb{R}^{n\times n},\boldsymbol{S}*\boldsymbol{1}=\boldsymbol{1},\boldsymbol{S}\geq\boldsymbol{0}} \left(\|\boldsymbol{S}-\boldsymbol{A}\|_F^2+\lambda Tr(\boldsymbol{F}^T\boldsymbol{L}_S\boldsymbol{F})\right) \tag{11}$$

which is equivalent to the following

$$\min_{S_{i,j}\geq 0,\sum_j S_{i,j}=1}\sum_{i,j=1}^{n}(S_{i,j}-A_{i,j})^2+\frac{\lambda}{2}\sum_{i,j=1}^{n}\|\boldsymbol{f}_i-\boldsymbol{f}_j\|_2^2S_{i,j} \tag{12}$$

The rows of $\boldsymbol{S}$ are independent from each other. Thus, we can solve $\boldsymbol{S}$ row by row. For each row $\boldsymbol{s}_i$, we have

$$\min_{\boldsymbol{s}_i\geq 0,\boldsymbol{s}_i\boldsymbol{1}=1}\|\boldsymbol{s}_i-\boldsymbol{a}_i\|_2^2+\frac{\lambda}{2}\boldsymbol{s}_i\boldsymbol{d}_i^T \tag{13}$$

where $\boldsymbol{s}_i \in \mathbb{R}^{1\times n}$ whose j_{th} element is $S_{i,j}$, $\boldsymbol{a}_i \in \mathbb{R}^{1\times n}$ whose j_{th} element is $A_{i,j}$, and j_{th} element of d_i is $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|_2^2$. Eq. (13) can be further formulated as

$$\begin{aligned} &\min_{\boldsymbol{s}_i \geq 0, \boldsymbol{s}_i \mathbf{1}=1} \|\boldsymbol{s}_i - (\boldsymbol{a}_i - \frac{\lambda}{4}\boldsymbol{d}_i)\|_2^2 \\ =&\min_{x_j \geq 0, \sum_j x_j = 1} \|\boldsymbol{x} - \boldsymbol{v}\|_2^2 \end{aligned} \tag{14}$$

where $\boldsymbol{x} = \boldsymbol{s}_i$, $\boldsymbol{v} = \boldsymbol{a}_i - \frac{\lambda}{4}\boldsymbol{d}_i$. $\|\boldsymbol{x} - \boldsymbol{v}\|_2^2 = R^2$ is a Euclidean ball, and $x_j \geq 0, \sum_j x_j = 1$ defines a hyper-plane. The optimal solution is the tangent point between the Euclidean ball and the hyper-plane or the angle point of the hyper-lane. We summarize the structure of the solution in Algorithm 1.

Algorithm 1 Optimization Algorithm of PUFE-OG

Input: cluster number c, parameter λ, μ, σ, $\boldsymbol{X}$.
Output: learned projection parameters $\boldsymbol{W}$, $\boldsymbol{b}$.
Initialization: $\boldsymbol{A} \leftarrow \boldsymbol{A}_0$, $\boldsymbol{S} \leftarrow \boldsymbol{S}_0$, $\boldsymbol{L_S} = Laplace(\boldsymbol{S})$.
repeat
  fix $\boldsymbol{S}$, *update* $\boldsymbol{W}$,$\boldsymbol{b}$,$\boldsymbol{F}$:
    Update $\boldsymbol{F}$ *according to Eq. (10);*
    Update $\boldsymbol{W}$,$\boldsymbol{b}$ *according to Eq. (9).*
  fix $\boldsymbol{W}$,$\boldsymbol{b}$,$\boldsymbol{F}$, *update* $\boldsymbol{S}$:
    For each row $\boldsymbol{s}_i$, *update it according to Eq. (14).*
    $\boldsymbol{L_S} = \boldsymbol{D_S} - \frac{\boldsymbol{S}+\boldsymbol{S}^T}{2}$;
until converges

4 Experiments

Datasets & Setup: To evaluate our model, we choose three multiview action recognition datasets, three face datasets and one handwritten digit recognition dataset. The action datasets are the IXMAS dataset [27], the newer version of of IXMAS dataset referred to as NIXMAS, and the partially occluded dataset OIXMAS [28] dataset. The three face datasets are the JAFFE dataset [13], the UMIST face dataset [9], and the YaleB dataset [3]. We also use the USPS dataset [9] to validate the performance on handwritten digit recognition. Table 1 shows the details of the 7 benchmark datasets, such as the number of instances, the dimension of each instance, as well as the number of classes.

Datasets	Instance No.	Dimension	Class No.
IXMAS [27]	1789	500	12
NIXMAS [27]	1148	500	11
OIXMAS [28]	1800	500	12
JAFFE [13]	213	676	10
UMIST [9]	575	644	20
USPS [9]	9298	256	10
Yaleb [3]	575	644	20

Table 1: Description of 7 Benchmark Datasets

In the experiment, we found that 15 is a good neighborhood number to build the graph. For parameters λ, μ, σ, we tune them in the range of $[10^{-3}; 10^{-2}; ...10^{3}]$. The dimension of the low-dimensional space is set to $[10; 20; ...; 100]$.

Table 2: Performance comparison (Mean Accuracy% ± Standard Deviation)

	IXMAS	NIXMAS	OIXMAS	JAFFE	UMIST	USPS	Yaleb
Raw Feature	20.99±2.40	20.64±0.86	18.47±0.95	73.29±7.27	41.56±2.47	65.08±2.34	11.73±0.80
PCA	21.11±2.42	20.50±0.80	18.63±0.96	73.18±7.20	43.29±2.52	66.31±2.36	11.71±0.71
Atuo Enc. [23]	25.59±2.33	22.30±0.14	16.39±1.09	67.81±5.17	37.58±2.42	46.90±0.79	7.33±0.68
NPE [8]	19.44±2.32	19.10±0.68	18.78±1.05	75.25±7.92	40.93±2.44	62.74±2.71	20.70±1.18
LPP [17]	22.57±2.03	19.84±0.80	18.53±0.73	79.28±7.81	44.50±2.74	68.62±3.48	28.89±1.51
PUFE [16]	28.28±1.71	21.46±0.54	20.94±0.58	80.98±8.39	55.01±2.65	71.17±3.64	45.07±2.47
PUFE-OG	**28.59±1.72**	**24.02±0.52**	**21.50±0.77**	**83.30±7.24**	**71.13±1.18**	**75.22±2.65**	**60.05±2.43**

Table 3: Performance comparison (Mean NMI% ± Standard Deviation)

	IXMAS	NIXMAS	OIXMAS	JAFFE	UMIST	USPS	Yaleb
Raw Feature	27.14±2.73	15.91±0.97	16.78±2.11	80.08±5.06	64.10±1.76	66.97±0.86	16.48±1.09
PCA	27.11±2.81	15.79±0.77	16.95±2.04	80.18±4.92	63.86±1.71	61.03±0.81	16.54±0.98
Atuo Enc. [23]	25.59±1.99	20.34±0.04	13.89±1.03	75.26±2.57	54.04±1.96	42.49±0.14	9.64±0.67
NPE [8]	25.09±2.62	12.60±0.76	19.03±2.09	82.53±4.43	61.01±2.01	59.42±1.36	27.06±0.97
LPP [17]	28.19±2.62	12.97±0.96	14.38±0.11	86.47±4.33	62.79±2.25	65.37±0.99	41.10±1.00
PUFE [16]	31.75±2.64	16.90±0.86	23.53±0.58	86.44±5.05	74.51±1.46	70.82±1.64	56.85±1.59
PUFE-OG	**33.68±2.15**	**28.83±0.65**	**24.95±1.23**	**88.47±4.54**	**86.71±2.26**	**73.71±0.85**	**73.14±1.61**

Experimental Results: To evaluate the performance with its projection ability, we compare it with raw feature, PCA, denoising auto-encoder (pretraining stage) [23], Neighbor Preserving Embedding (NPE) [8], Locality Preserving Projections (LPP) [17], as well as the PUFE [16] without optimal graph. To evaluate the projection ability of each method, we first learn the projection matrix and obtain the corresponding low-dimensional representations. Then we rely on K-Means to do the clustering. The result is shown in Table 2 and Table 3. We run the K-Means 100 times each time, and report the average value and standard deviation in the table.

The experimental results are listed in Table 2 and Table 3. We can observe that the performance of our method is consistently better than the others under different evaluation metrics. We also observe that:

• The graph based dimensionality reduction methods (NPE, LPP, PUFE, PUFE-OG) usually achieve better performance compared with PCA and raw feature. This observation suggests that low-dimensional representations can have better performance if it is based on a weighted graph which contains the structure information of the local cliques.

• The performance of PCA is always similar to the performance of raw feature. Sometimes it has little improvement or inferior performance compared with the raw feature. The underlying reason of the observation is that PCA does not preserve the manifold smoothness.

• Denoising auto-encoder have relatively better performance for the first two datasets than the raw feature. This observations suggests that denoising auto-encoder is robust to view changes compared with other methods. However, it fails capturing the inter-class difference of other datasets and has the worst performance among all the methods.

• PUFE always outperforms LPP and NPE and it is very stable across all the datasets. This is because PUFE relaxes the rigid projection form, namely, $\boldsymbol{F}=\boldsymbol{X}^T\boldsymbol{W}$, to a more flexible version, which is $\boldsymbol{F}\approx\boldsymbol{X}^T\boldsymbol{W}+\mathbf{1}\boldsymbol{b}^T$.

• Our PUFE-OG consistently outperforms other graph based algorithms and it is very stable. This advantage is attributed to the optimal neighborhood probability graph construction.

• The data in UMIST is more regular compared with other datasets. Thus, its manifold structure is more label consistent (the samples from the same cluster are lying close by). Then the improvement in accuracy is higher compared with other datasets.

Parameter Sensitivity Analysis: We choose UMIST dataset to study the sensitivity of the

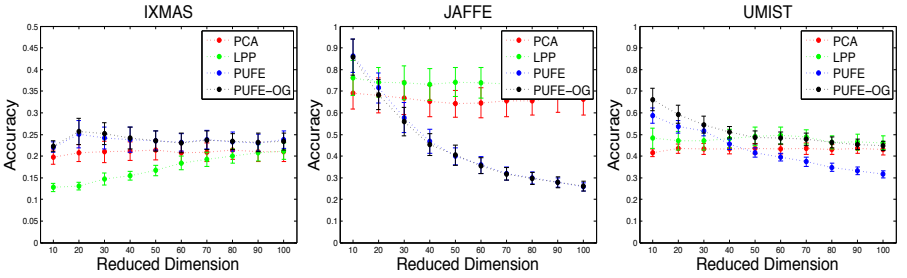

Figure 1: Results of Projection with Different Dimensions on 3 Benchmark Datasets

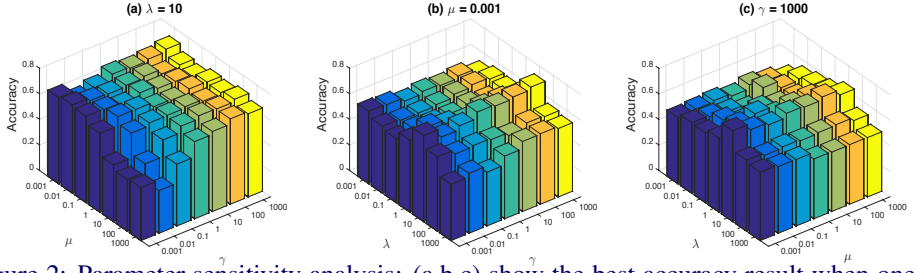

Figure 2: Parameter sensitivity analysis: (a,b,c) show the best accuracy result when one of the three parameters λ, μ, γ is fixed

parameters. UMIST has 20 classes, and it consists of 575 images with the dimension of 644. The best clustering accuracy is obtained when $\lambda = 10, \mu = 0.001, \gamma = 1000$.

We study the sensitivity of the three parameters by fixing one of them to the optimal settings. Figure 2 shows the clustering accuracy. From Figure 2(a), we can observe that when λ is fixed, a smaller μ and a larger γ can lead to a better performance. Figure 2(b) shows that λ is more sensitive than γ and its optimal setting is $\lambda = 10$. A similar result can be observed in Figure 2(c). To sum up, λ is the most sensitive parameter, and a smaller μ and larger γ can lead to a better performance.

We also test the sensitivity of the reduced dimensions of the low-dimensional space. As shown in Table 2, the most stable methods across different benchmark datasets are PCA, LPP, PUFE, and PUFE-OG. Thus, we mainly focus on these 4 methods. Figure 1 shows the results on 3 benchmark datasets. We can observe that PUFE and PUFE-OG have better performance when the reduced dimension is low. The performance of PCA and LPP become better as the dimension increases. But different from PCA and LPP, a high dimension degrades the performance of PUFE and PUFE-OG. This is because the quality of the manifold structure has a great influence to the performance. The local manifold structure is often more reliable (the k nearest neighbors are more likely from the same cluster when k is small) and the distance can be approximated to be linear for the local structure. When the dimension is low, PUFE-OG will focus on learning the local manifold structure and yield a significant performance gain. However, when dimension is too high, the linear measurement will become inaccurate. Then the performance will drop dramatically. Thus, PUFE-OG is more sensitive to dimension change compared with other methods, and a low dimension is preferable for PUFE-OG.

5 Conclusion

To cope with the out-of-sample problem, most unsupervised graph based dimension reduction techqiues integrate the ridge regression term with the manifold regularizer. The performance of these graph based methods highly depends on the input affinity matrix. However, the graph might be sub-optimal for dimension reduction. To tackle this problem, we propose a novel projective unsupervised flexible embedding framework by constructing an optimal graph.

Instead of using the affinity matrix to build the graph, we construct a probability graph based on the Gaussian similarity graph. The probability graph is more smooth. Thus, an optimal graph can be learned and the experimental results prove that our PUFE-OG model always outperforms the other projective dimension reduction techniques. We derive an efficient algorithm to solve the problem. The extensive experiments demonstrate the effectiveness and the stability of our model. This optimal graph construction method can also be applied to other graphs, such as l^1 graph [30] and LLE graph. In the future, we will explore its application in other graph based methods, as well as its application in the semi-supervised setting.

6 Acknowledgement

This work has been partially supported by the EC project QoSTREAM.

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[2] Israel-Dejene Gebru, Xavier Alameda-Pineda, Florence Forbes, and Radu Horaud. EM algorithms for weighted-data clustering with application to audio-visual scene analysis. *TPAMI*, 2016.

[3] 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. *TPAMI*, 2001.

[4] Daniel B Graham, Nigel Allinson, and M. Characterising virtual eigensignatures for general purpose face recognition. In *Face Recognition*. 1998.

[5] Xiaofei He, Deng Cai, Shuicheng Yan, and Hong-Jiang Zhang. Neighborhood preserving embedding. In *ICCV*, 2005.

[6] Jin Huang, Feiping Nie, and Heng Huang. Spectral rotation versus k-means in spectral clustering. In *AAAI*, 2013.

[7] Jin Huang, Feiping Nie, Heng Huang, and Chris Ding. Supervised and projected sparse coding for image classification. In *AAAI*, 2013.

[8] Thomas Hueber, Laurent Girin, Xavier Alameda-Pineda, and Gerard Bailly. Speaker-adaptive acoustic-articulatory inversion using cascaded gaussian mixture regression. *TASLP*, 2015.

[9] B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1990.

[10] Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, and Hanqing Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, 2012.

[11] Jialu Liu, Chi Wang, Marina Danilevsky, and Jiawei Han. Large-scale spectral clustering on graphs. In *IJCAI*, 2013.

[12] Dijun Luo, Chris HQ Ding, and Heng Huang. Linear discriminant analysis: New formulations and overfit analysis. In *AAAI*, 2011.

[13] Michael J Lyons, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba, and Julien Budynek. The japanese female facial expression (jaffe) database. 1998.

[14] Adeel Mumtaz, Emanuele Coviello, Gert RG Lanckriet, and Antoni B Chan. Clustering dynamic textures with the hierarchical em algorithm for modeling video. *TPAMI*, 2013.

[15] Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *TIP*, 2010.

[16] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *SIGKDD*. ACM, 2014.

[17] X Niyogi. Locality preserving projections. In *NIPS*, 2004.

[18] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *CVPR*, 2010.

[19] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000.

[20] Ling Shao, Di Wu, and Xuelong Li. Learning deep and wide: A spectral method for learning deep networks. *TNNLS*, 2014.

[21] Vikas Sindhwani, Partha Niyogi, Mikhail Belkin, and Sathiya Keerthi. Linear manifold regularization for large scale semi-supervised learning. In *ICML Workshop on Learning with Partially Classified Training Data*, 2005.

[22] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[23] Pascal Vin., Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[24] Wei Wang, Yan Yan, and Sebe Nicu. Attribute guided dictionary learning. In *ICMR*, 2015.

[25] Wei Wang, Zhen Cui, Yan Yan, Jiashi Feng, Shuicheng Yan, Xiangbo Shu, and Sebe Nicu. Recurrent face aging. In *CVPR*, 2016.

[26] Wei Wang, Yan Yan, Stefan Winkler, and Nicu Sebe. Category specific dictionary learning for attribute specific feature selection. *TIP*, 2016.

[27] Daniel Wein., Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 2006.

[28] Daniel Wein., Mustafa Özuysal, and Pascal Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*. 2010.

[29] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2014.

[30] Shuicheng Yan and Huan Wang. Semi-supervised learning by sparse representation. In *SDM*, 2009.

[31] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *TPAMI*, 2007.

[32] Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, and Nicu Sebe. Multitask linear discriminant analysis for view invariant action recognition. *TIP*, 2014.

[33] Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, Oswald Lanz, and Nicu Sebe. A multi-task learning framework for head pose estimation under target motion. *TPAMI*, 2016.

[34] Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. Image clustering using local discriminant models and global integration. *TIP*, 2010.

[35] Jieping Ye, Zheng Zhao, and Mingrui Wu. Discriminative k-means for clustering. In *NIPS*, 2008.