Discriminative Unsupervised Dimensionality Reduction

Xiaoqian Wang, Yun Liu, Feiping Nie, Heng Huang*
University of Texas at Arlington
Arlington, Texas 76019, USA
xqwang1991@gmail.com, yun.liu@mavs.uta.edu, feipingnie@gmail.com, heng@uta.edu

Abstract

As an important machine learning topic, dimensionality reduction has been widely studied and utilized in various kinds of areas. A multitude of dimensionality reduction methods have been developed, among which unsupervised dimensionality reduction is more desirable when obtaining label information requires onerous work. However, most previous unsupervised dimensionality reduction methods call for an affinity graph constructed beforehand, with which the following dimensionality reduction steps can be then performed. Separation of graph construction and dimensionality reduction leads the dimensionality reduction process highly dependent on quality of the input graph. In this paper, we propose a novel graph embedding method for unsupervised dimensionality reduction. We simultaneously conduct dimensionality reduction along with graph construction by assigning adaptive and optimal neighbors according to the projected local distances. Our method doesn't need an affinity graph constructed in advance, but instead learns the graph concurrently with dimensionality reduction. Thus, the learned graph is optimal for dimensionality reduction. Meanwhile, our learned graph has an explicit block diagonal structure, from which the clustering results could be directly revealed without any postprocessing steps. Extensive empirical results on dimensionality reduction as well as clustering are presented to corroborate the performance of our method.

1 Introduction

Natural and social science applications are crowded with high-dimensional data in this day and age. However, in most cases these data are literally characterized by an underlying low-dimensional space. This interesting phenomenon draws high attention to dimensionality reduction researches which focus on discovering intrinsic manifold structure from the high dimensional ambient. Multitudinous supervised and unsupervised dimensionality reduction methods have been put forward, such as PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), LLE [Roweis and Saul, 2000], LPP [Niyogi, 2004], shift invariant LPP [Nie *et al.*, 2014a], NMMP [Nie *et al.*, 2007], TRACK [Wang *et al.*, 2014], *etc.* In the circumstances where accessing label of the data is intricate, unsupervised dimensionality reduction methods are more favorable. Meanwhile, among the tremendous number of unsupervised dimensionality reduction methods, graph embedding method is laid more emphasis on since graph and manifold information are utilized within.

However, most of state-of-the-art graph based dimensionality reduction methods require an affinity graph constructed before hand, which makes their projection ability dependent heavily on the input of graph. However, due to the separated learning processes, the constructed graph may not be optimal for the later dimensionality reduction. To address this problem, in this paper, we propose a novel graph embedding method for unsupervised dimensionality reduction which asks for no input of the graph. Instead, graph construction in our model is conducted simultaneously with dimensionality reduction. We assign adaptive and optimal neighbors on the basis of the projected local distances. Our main assumption is that data with lower distance apart usually has a larger probability to be connected, which is a common hypothesis in previous graph based methods [Nie *et al.*, 2014b]. Also, we constrain the learned graph to an ideal structure where the graph is block diagonal with the number of connected components to be exactly the number of clusters in the data, such that the constructed graph also uncovers the data cluster structure to enhance the graph quality.

2 Related Work

PCA is the most famous dimensionality reduction method, which is meant to find the projection direction where the variance of data is maximized. Since the variance of projected data can be rewritten as $\sum_{i}^{n}\sum_{j}^{n}\left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 = tr(W^TXHX^TW)$, where H is the centering matrix as:

$$H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T, \tag{1}$$

*Corresponding Author. X. Wang and Y. Liu contribute equally to this paper. This work was partially supported by NSF IIS-1117965, IIS-1302675, IIS-1344152, DBI-1356628.

and the goal of PCA is to solve:

$$\max_{W^TW=I} tr(W^T X H X^T W)\,.$$

However, as is pointed out in [Welling, 2005], PCA goes incapable when distances between classes lower down while the scale of each class is still large in a certain degree, as we can imagine the case where several cigars are placed closely, with each cigar representing the distribution of data in one class.

Afterwards, LDA was proposed to accomplish a better dimensionality reduction task as it was devoted to minimizing the within class distance, S_w, while maximizing the between class distance, S_b. Nevertheless, LDA also encounters several problems. For example, the *small sample size problem* occurs when the number of samples is smaller than the number of data dimensions. Researchers have came up with numerous ways to overcome this obstacle. Authors in [Chen *et al.*, 2000] put forward an approach to finding the most discriminative information in the null space of S_w which then evades the computational difficulty generated by the singularity of S_w in Fisher LDA. Another research [Yu and Yang, 2001] indicates that by discarding the null space of S_b, which is non-informative, one can solve the traditional LDA problem from a better point of view.

Whereas, all these LDA methods mentioned above necessitate the knowledge of data labels, which may not be always easily accessible, especially in the cases where labeling a data calls for mountains of work. Researches in the state of art come up with graph embedding methods, among which LPP can be seen as an representative example. Given data $X \in \mathbb{R}^{d\times n}$, suppose we are to learn a projection matrix $W \in \mathbb{R}^{d\times m}$, where m is the number of dimension we'd like to reduce to. The idea of LPP works like this: firstly learn an affinity graph S showing pairwise affinity between data points and then find a "good" mapping by tackling the following problem [Belkin and Niyogi, 2001; Niyogi, 2004]:

$$\min_{W^TXDX^TW=I} \sum_i^n \sum_j^n \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}\,,$$

where D is the degree matrix of S.

Also, there is another variant of this model published in [Kokiopoulou and Saad, 2007]:

$$\min_{W^TW=I} \sum_i^n \sum_j^n \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}\,.$$

Despite the fact that no more do these graph embedding methods require label information, it is still a must that an affinity graph constructed ahead of time to work as the input. These methods separate dimensionality reduction and graph construction, thus depend on quality of the input affinity graph to a large extent, where poor quality graph gives rise to erroneous dimensionality reduction. In this paper, we put forward a novel graph embedding dimensionality reduction method which combines these two steps together. In our method, we concurrently carry out dimensionality reduction as well as graph construction by assigning adaptive and optimal neighbors on the basis of the projected local distances. Our fundamental standing point is that data with lower distance apart usually has a larger probability to be connected, in other words, in the same cluster. Also, we constrain the learned graph to an ideal structure where the graph is block diagonal with the number of connected components to be exactly the number of clusters in the data. The detailed description on how we implement dimensionality reduction and graph construction at the same time and what remarkable properties of the learned graph have will be exhibited in the next section.

3 Graph Embedding Discriminative Unsupervised Dimension Reduction

We hope to learn the optimal affinity graph S to optimize the projection matrix W for unsupervised dimensionality reduction. Ideally, we should learn both S and W simultaneously in a unified objective. To design the proper learning model, we hope to emphasize the following ideas: 1) The affinity matrix $S \in \mathbb{R}^{n\times n}$ and projection matrix W are mutually learned, *e.g.* optimize S and W simultaneously in $\min_{W,S} \sum_i^n \sum_j^n \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}$ (previous methods learned S separately and only optimize W in dimensionality reduction). 2) The learned affinity matrix S implies the probability of each data point in X to connect with its neighbors, *i.e.* a larger probability should be assigned to a pair with smaller distance, such that the graph structure is interpretable. 3) The data variance in the embedding space is maximized to retain most information.

Based the above considerations, we can build the following objective:

$$\begin{aligned} &\min_{S,W} \frac{\sum_{i,j=1}^n \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}}{tr(W^TXHX^TW)} \\ &s.t. \quad \forall i, \mathbf{s}_i^T\mathbf{1} = 1, 0 \le s_{ij} \le 1, W^TW = I\,, \end{aligned} \tag{2}$$

where H is the centering matrix define in Eq. (1).

However, Problem (2) has a trivial solution that only the nearest data point of W^Tx_i is assigned a probability as 1 while all others assigned 0, that is to say, x_i is connected with only its nearest neighbor in the projected space.

That is definitely not what we expect. Instead, we hope the learned affinity graph can maintain or enhance the data cluster relations, such that the projected data don't destroy such important structure. The desired result is that we project the data to a low-dimensional subspace where the probability within cluster is nonzero and evenly distributed while the probability between clusters is zero.

The assumption of this ideal structure describes a block diagonal graph whose number of connected components is the same as the the number of clusters. However, how to translate this ideal structure to an equation language seems to be a fairly intractable task. Here we come up with a novel and simple idea to accomplish this challenge.

Given $F \in \mathbb{R}^{n\times k}$, suppose each node i is assigned a function value as $\mathbf{f}_i \in \mathbb{R}^{1\times k}$, then it can be verified that:

$$\sum_{i,j} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 s_{ij} = 2tr(F^T L_S F)\,, \tag{3}$$

where $L_S = D_S - \frac{S^T+S}{2}$ is the Laplacian matrix in graph theory, and the degree matrix $D_S \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose i-th diagonal element is $\sum_j (s_{ij} + s_{ji})/2$. If S is nonnegative, then the Laplacian matrix has an important property as follows [Mohar, 1991].

Theorem 1 *The multiplicity k of the eigenvalue 0 of the Laplacian matrix L_S is equal to the number of connected components in the graph associated with S.*

Given the probability matrix S, Theorem 1 indicates that if $r(L_S) = n - k$, the graph could explicitly partition the data points into exactly k clusters according to the block diagonal structure.

What's more, to guarantee that the probability within clusters is evenly distributed, we add a regularization term $\gamma \|S\|_F^2$, where a large enough γ could force the s_{ij} value within a block to be the same.

Thus our model becomes:

$$\begin{aligned} \min_{S,W} & \frac{\sum_{i,j=1}^{n} \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}}{tr(W^T X H X^T W)} + \gamma \|S\|_F^2 \\ s.t. \quad & \forall i, \mathbf{s}_i^T\mathbf{1} = 1, 0 \le s_{ij} \le 1, W^T W = I, \\ & rank(L_S) = n - k\,. \end{aligned} \tag{4}$$

It is notable that our model learns the projection matrix W and the affinity matrix S at the same time, which is significantly different from previous works. Moreover, our learned affinity matrix S has an ideal block diagonal structure. But Problem (4) seems very difficult to solve especially when there is a strict constraint on $rank(L_S)$. In the following section, we propose a novel algorithm to fulfil the optimization.

4 Optimization Algorithm Solving Problem (4)

In Problem (4), suppose $\sigma_i(L_S)$ is the i-th smallest eigenvalue of L_S. It is easy to see that $\sigma_i(L_S) \ge 0$ since L_S is positive semi-definite. So for a large enough λ, Problem (4) would be equivalent to:

$$\begin{aligned} \min_{S,W} & \frac{\sum_{i,j=1}^{n} \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}}{tr(W^T X H X^T W)} + \gamma \|S\|_F^2 \\ & + 2\lambda \sum_{i=1}^{k} \sigma_i(L_S) \\ s.t. \quad & \forall i, \mathbf{s}_i^T\mathbf{1} = 1, 0 \le s_{ij} \le 1, W^T W = I\,, \end{aligned} \tag{5}$$

where a large enough λ guarantees that the k smallest eigenvalues of L_S are all zero and thus the rank of L_S is $n - k$.

According to the Ky Fan's Theorem [Fan, 1949], we have:

$$\sum_{i=1}^{k} \sigma_i(L_S) = \min_{F\in R^{n\times k}, F^T F = I} tr(F^T L_S F)\,. \tag{6}$$

So we turn to solve:

$$\begin{aligned} \min_{S,W,F} & \frac{\sum_{i,j=1}^{n} \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}}{tr(W^T X H X^T W)} + \gamma \|S\|_F^2 \\ & + 2\lambda tr(F^T L_S F) \\ s.t. \quad & \forall i, \mathbf{s}_i^T\mathbf{1} = 1, 0 \le s_{ij} \le 1, W^T W = I, \\ & F \in R^{n\times k}, F^T F = I\,. \end{aligned} \tag{7}$$

We can solve Problem (7) by means of the alternative optimization method.

The first step is fixing W, S and solving F. Then Problem (7) becomes:

$$\min_{F\in R^{n\times k}, F^T F = I} tr(F^T L_S F) \tag{8}$$

The optimal solution of F in Problem (8) is formed by the k eigenvectors corresponding to the k smallest eigenvalues of L_S.

The second step is fixing S, F and solving W. Then Problem (7) becomes:

$$\min_{W^T W = I} \frac{\sum_{i,j=1}^{n} \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}}{tr(W^T X H X^T W)}\,, \tag{9}$$

which can be rewritten as

$$\min_{W^T W = I} \frac{tr(W^T X L_S X^T W)}{tr(W^T X H X^T W)}\,. \tag{10}$$

We can solve W using the iterative method introduced in [Nie *et al.*, 2009]. The Lagrangian function of Problem (10) is:

$$\mathcal{L}(W, \Lambda) = \frac{tr(W^T X L_S X^T W)}{tr(W^T X H X^T W)} - tr(\Lambda(W^T W - I))\,. \tag{11}$$

Taking derivative w.r.t. W and set it to zero, we have:

$$\begin{aligned} & (X L_S X^T - \frac{tr(W^T X L_S X^T W)}{tr(W^T X H X^T W)} X H X^T) W \\ = \quad & \Lambda W\,. \end{aligned} \tag{12}$$

The solution of W in Problem (12) is formed by the m eigenvectors corresponding to the m smallest eigenvalues of the matrix:

$$(X L_S X^T - \frac{tr(W^T X L_S X^T W)}{tr(W^T X H X^T W)} X H X^T)\,. \tag{13}$$

We can iteratively update W until $K.K.T.$ condition in Eq. (12) is satisfied.

The third step is fixing W, F and solving S. Then according to Eq. (3), Problem (7) becomes:

$$\begin{aligned} \min_{\forall i, \mathbf{s}_i^T\mathbf{1}=1, 0\le s_{ij}\le 1} & \frac{\sum_{i,j=1}^{n} \left\|W^T\mathbf{x}_i - W^T\mathbf{x}_j\right\|_2^2 s_{ij}}{tr(W^T X H X^T W)} \\ & + \gamma \|S\|_F^2 + \lambda \sum_{i,j=1}^{n} \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 s_{ij}\,. \end{aligned} \tag{14}$$

Problem (14) can be solved separately for each $\mathbf{s}_i$ as follows:

$$\min_{\mathbf{s}_i} \sum_{j=1}^{n} \left(d_{ij}^{wx} s_{ij} + \gamma s_{ij}^2 + \lambda d_{ij}^f s_{ij} \right) \quad s.t. \quad \mathbf{s}_i^T \mathbf{1} = 1, 0 \leq s_{ij} \leq 1 \,, \tag{15}$$

where $d_{ij}^{wx} = \frac{\left\| W^T \mathbf{x}_i - W^T \mathbf{x}_j \right\|_2^2}{tr(W^T X H X^T W)}$ and $d_{ij}^f = \|\mathbf{f}_i - \mathbf{f}_j\|_2^2$.

Then Problem (15) can be rewritten as:

$$\min_{\mathbf{s}_i^T \mathbf{1}=1, 0 \leq s_{ij} \leq 1} \left\| \mathbf{s}_i + \frac{1}{2\gamma} \mathbf{d}_i \right\|_2^2 , \tag{16}$$

where $d_{ij} = d_{ij}^{wx} + \lambda d_{ij}^f$. Then we can update $\mathbf{s}_i$ accordingly.

We can iteratively update F, W and S with the three alternative steps mentioned above and the algorithm for solving Problem (7) is summarized in Algorithm 1.

Algorithm 1 Algorithm to solve Problem (7).

Input:

Data matrix $X \in \mathbb{R}^{d \times n}$, number of clusters k, reduced dimension number m, parameter γ and λ.

Output:

Projection $W \in \mathbb{R}^{d \times m}$ and probability matrix $S \in \mathbb{R}^{n \times n}$ with exactly k connected components.

Initialize S by setting its i-th column $\mathbf{s}_i$ as the optimal solution to $\min_{\mathbf{s}_i^T \mathbf{1}=1, 0 \leq s_{ij} \leq 1} \sum_{j=1}^{n} (\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 s_{ij} + \gamma s_{ij}^2)$.

while not converge **do**

1. Update $L_S = D_S - \frac{S^T + S}{2}$, where $D_S \in \mathbb{R}^{n \times n}$ is a diagonal matrix whose i-th diagonal element is $\sum_j (s_{ij} + s_{ji})/2$.
2. Update F, whose columns are the k eigenvectors of L_S corresponding to the k smallest eigenvalues.
2. Update W, whose columns are the m eigenvectors of matrix in (13) corresponding to the m smallest eigenvalues. Update W iteratively until converges.
3. For each i, update the i-th column of S by solving Problem (16).

end while

5 Discussion of Algorithm 1

Our algorithm uses the alternative optimization method, whose convergence has already been proved in [Bezdek and Hathaway, 2003]. In our method, the slowest step in each iteration is the eigen-decomposition step for Eq. (8) and Eq. (13), which can be efficiently solved by many existing numerical computation packages. The time complexity of our method is $O((d^2 m + n^2 k)T)$, where T is the number of iterations. In practice, our method usually converges within 20 iterations, so it is fairly efficient in both dimensionality reduction and clustering tasks. It is also worth mentioning that we get clustering results immediately with the structure of S, with no need of post processing like K-Means in spectral clustering methods.

There are three hyper parameters in Problem (7), which are k, γ and λ. The number of clusters k can be set via priori knowledge of the data, or discovered using some efficient methods like [Rodriguez and Laio, 2014]. As for λ, it can be determined in an heuristic approach: first set λ with an initial guess, then in each iteration, we compute the number of zero eigenvalues, if it's larger than k, then divide λ by 2; if smaller than k then multiply λ by 2; otherwise we stop the iteration. So in our algorithm, the only parameter we need to tune is γ.

Even though Algorithm 1 is proposed for linear dimensionality reduction, it can be easily extended to non-linear cases. By assorting to the data transformation technique proposed in [Zhang *et al.*, 2010], we can simply perform Algorithm 1 on the transformed data and achieve the same effect as kernel methods do.

6 Experimental Results

In this part, we will firstly validate performance of our dimensionality reduction method (Model (7)) on both synthetic data and real-world benchmark data sets. Afterwards we will present clustering results achieved by our method. For simplicity, we denote our clustering method as DUDR (Discriminative Unsupervised Dimensionality Reduction) in the following context.

6.1 Experiments on Synthetic Data

The synthetic data in this experiment is a randomly generated two-Gaussian matrix. We stochastically generate two clusters of data which obeys Gaussian distribution. Our goal is to find an effective projection direction in which the two clusters could be explicitly set apart. We compared our dimensionality reduction method DUDR with two related methods, PCA and LPP, and displayed comparison results in Fig. 1. Seen from Fig. 1, we know that when these two clusters are far from each other, all these three methods could easily find a good projection direction. However, as the distance between two clusters lower down, PCA becomes incompetent. As the two clusters draw closer, LPP also lose its way to find a "good" projection direction. In contrast, DUDR performs consistently well under all occasions. The reason for this phenomenon goes as follows: PCA is focused on the global structure of data, so when the distance between clusters becomes smaller than the length of each cluster, it is unable to distinguish two clusters thus fails immediately. As for LPP, it pays more attention to the local structure, thus works well when two clusters are relatively close. However, when the distance further lower down, LPP is not capable any more. Whereas, our method DUDR lays more emphasis on the discriminative structure, hence is able to preserve its projection ability in all circumstances.

6.2 Experiments on Real Benchmark Datasets

We tested both projection and clustering ability of DUDR on 8 benchmark image datasets: Pathbased, Compound, Spiral, Movements [Asuncion and Newman, 2007], Jaffe [Lyons *et al.*, 1998], AR_ImData [Martınez and Benavente, 1998], XM2VTS [Messer *et al.*, 1999] and Coil20 [Nene *et al.*,

1996], among which the first three are shape set data [1], while the latter five are image data sets. Description of these 8 datasets is summarized in Table 1.

Table 1: Description of 8 benchmark data sets

Data sets	# of Instances	Dimensions	Classes
Pathbased	300	2	3
Compound	399	2	6
Spiral	312	2	3
Movements	360	90	15
Jaffe	213	1024	10
AR_ImData	840	768	120
XM2VTS50	1180	1024	295
Coil20	1440	1024	20

Experiments on Projection

We evaluated our dimensionality reduction method on the 5 benchmark data sets with high dimensions: AR_ImData, Movements, Coil20, Jaffe, XM2VT. Similar to that in the synthetic data experiment, we compared DUDR with PCA and LPP methods.

The comparison is based on the clustering experiments, where we first learned the projection matrix separately with these three methods and then ran K-Means on the projected data. For each method we repeated K-Means for 100 times with the same initialization and recorded the best result w.r.t. the K-means objective function value in these 100 runs.

Among these three methods, LPP requires an affinity matrix constructed before hand, so in this experiment we constructed the graph with the self-tune Gaussian method [Chen *et al.*, 2011]. We set the number of neighbors to be 5 and the parameter σ to be self-tuned so as to guarantee the input graph quality. As for DUDR, we tuned the γ value via cross validation.

For Movement data set, we compared the performance by setting reduced dimensions to the range of 1 to 16, while for all other four data sets, we set the scale to be 1 to 100. Besides the three dimensionality reduction methods, we also included the baseline results by conducting K-means clustering on the original data.

The comparison results are reported in Fig. 6.2, from which we obtain two interesting observations: 1. Clustering results in the projected space tend to outperform the one in the original space, especially when the number of dimensions of the projected space increases. This is because noise may occur in the original space, whereas dimensionality reduction methods can find the subspace with discriminative power and discard the distracting information, thus cluster more accurately and rapidly. 2. DUDR outperforms PCA and LPP under different circumstances, and the superiority is especially evident when the number of projected dimension is small. DUDR method is able to project the original data to a subspace with quite small dimensions(k-1), where k is the number of clusters in the data set. Such low-dimensional subspace projected by our method even gains an advantage over that obtained by PCA and LPP with higher dimensions. So with DUDR we can project the data to a much lower dimensional space with clustering ability not weakened, which makes the dimensionality reduction process more efficient and effective.

[1]Downloaded from http://cs.joensuu.fi/sipu/datasets/

Experiments on Clustering

We evaluated the clustering ability of DUDR on all 8 benchmark data sets and compared it with several famous clustering methods, which are K-Means, Ratio Cut, Normalized Cut and NMF methods.

In the clustering experiment, we set the number of clusters to be the ground truth k in each data set and we set the projected dimension in DUDR to be k-1. Similar to that of the previous subsection, for all methods in need of an affinity matrix as an input, like Ratio Cut, Normalized Cut and NMF, the graph was constructed using the self-tune Gaussian method. For all methods involving K-Means, including K-Means, Ratio Cut and Normalized Cut, we ran K-Means for 100 times with the same initialization and wrote down their average performance, standard deviation and the performance corresponding to the best K-Means objection function value. As for NMF and DUDR, we ran only once and recorded the results.

The evaluation is based on two widely used clustering metrics: accuracy, NMI (normalized mutual information). Results summarized in Table 2 prove that DUDR outperforms all counterparts on most data sets. Under most occasions DUDR acquires an equivalent or even better accuracy and NMI with less time consumed, since K-Means, Ratio Cut and Normalized Cut need 100 times run but DUDR only calls for a few numbers of iteration; NMF requires a graph constructed before hand but DUDR doesn't. Moreover, clustering results of DUDR are steady for a certain setting while other methods are astable and heavily dependent on the initialization.

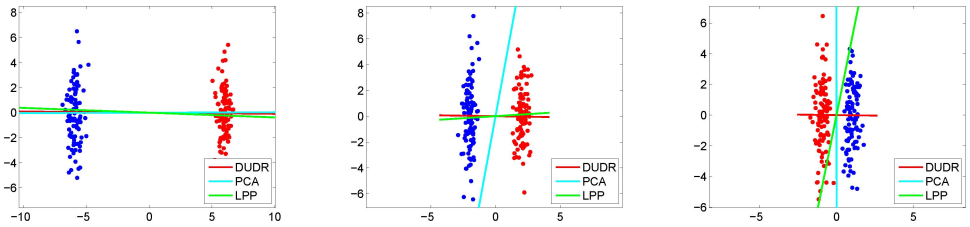
7 Conclusions

In this paper, we proposed a novel graph embedding dimensionality reduction model. Instead of learning a probabilistic affinity matrix before dimensionality reduction, we simultaneously conducted these two processes. We assigned adaptive and optimal neighbors according to the projected local connectivity. In our new graph embedding dimensionality reduction method, the learned graph has a promising block diagonal structure with exactly k connected components, where k denotes the number of clusters. We attained this goal by imposing rank constraint on the Laplacian matrix of graph. We derived an efficient algorithm to optimize the proposed objective and conducted rich experiments on both synthetic data and 8 real-world benchmark data sets to elucidate the superiority of our model.

References

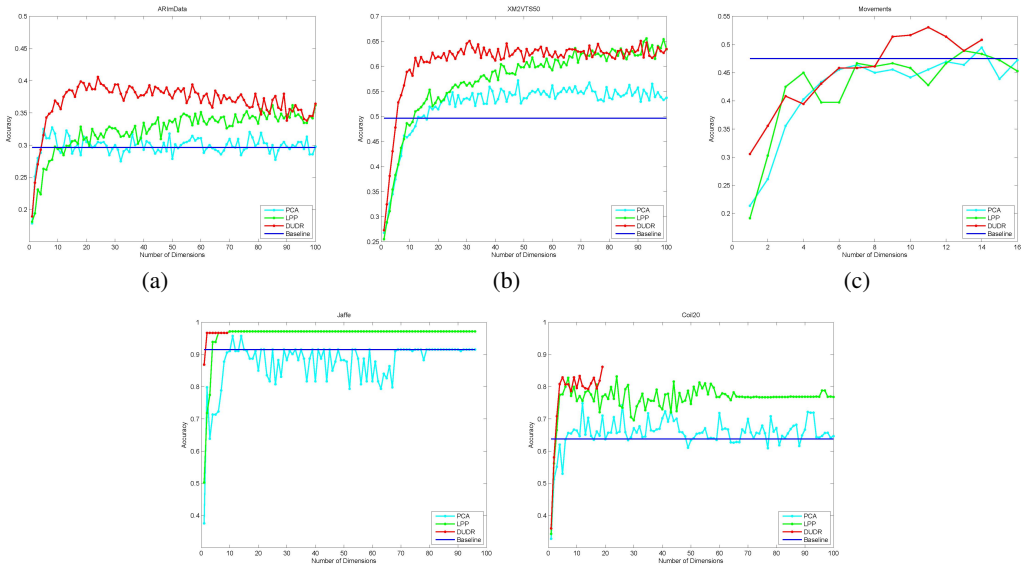
[Asuncion and Newman, 2007] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[Belkin and Niyogi, 2001] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.



(a) Far away (b) Relatively close (c) Fairly close

Figure 1: Projection results on the two-Gaussian synthetic data in three different settings of between cluster distance. Our goal is to find an effective projection direction in which the two clusters could be explicitly set apart.



(a) (b) (c)

(d) (e)

Figure 2: Projection results on five high-dimensional benchmark data sets. The baseline result is obtained by conducting K-means on the original data.

Table 2: Clustering accuracy and NMI on eight benchmark data sets

		K-Means		RatioCut		NormalizedCut		NMF	DUDR
		%(min_obj)	Average%	%(min_obj)	Average%	%(min_obj)	Average%		
ACC.	Pathbased	74.33	74.24±0.97	77.67	77.67±0.00	77.67	77.67±0.00	78.00	**87.00**
	Compound	69.42	63.93±10.66	53.63	53.12±4.48	53.13	52.64±3.56	52.38	**80.20**
	Spiral	33.97	34.54±0.29	99.68	98.10±7.80	99.68	97.36±9.60	91.03	**100.00**
	Movements	45.28	44.24±2.19	45.83	45.79±2.33	45.56	45.10±2.08	46.11	**51.11**
	Jaffe	91.08	74.83±8.36	**96.71**	85.17±7.38	**96.71**	80.76±8.29	**96.71**	**96.71**
	AR_ImData	28.57	27.43±1.03	34.88	35.32±0.75	36.19	36.54±0.77	37.14	**39.05**
	XM2VTS50	51.78	48.47±1.21	57.80	57.44±0.90	65.51	64.77±1.12	67.80	**68.64**
	Coil20	65.83	56.16±4.85	78.75	70.73±4.49	79.38	71.43±4.81	70.42	**82.99**
		K-Means		RatioCut		NormalizedCut		NMF	DUDR
		%(min_obj)	Average%	%(min_obj)	Average%	%(min_obj)	Average%		
NMI	Pathbased	51.28	51.17±1.29	55.16	55.16±0.00	55.16	55.16±0.00	52.51	**75.63**
	Compound	69.68	69.60±6.18	73.37	70.67±4.92	73.34	70.49±4.46	73.26	**79.27**
	Spiral	0.04	0.05±0.02	98.35	96.43±9.46	98.35	95.88±11.69	75.95	**100.00**
	Movements	58.50	57.35±1.84	60.64	61.45±2.17	60.89	59.89±1.95	64.08	**64.53**
	Jaffe	91.75	82.71±5.17	96.23	90.51±3.85	96.23	88.67±4.22	96.23	**96.25**
	AR_ImData	62.63	61.08±0.91	67.99	68.22±0.53	70.53	69.91±0.36	**70.56**	64.68
	XM2VTS50	82.41	81.26±0.52	81.24	81.29±0.95	88.63	88.46±0.27	**89.03**	82.93
	Coil20	79.08	73.16±2.43	88.43	84.34±2.00	**89.17**	84.99±2.10	81.22	88.95

[Bezdek and Hathaway, 2003] James C Bezdek and Richard J Hathaway. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, 11(4):351–368, 2003.

[Chen *et al.*, 2000] Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. A new lda-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.

[Chen *et al.*, 2011] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[Fan, 1949] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations. i. 35(11):652–655, 1949.

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

[Lyons *et al.*, 1998] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 200–205. IEEE, 1998.

[Martınez and Benavente, 1998] Aleix Martınez and Robert Benavente. The ar face database. *Rapport technique*, 24, 1998.

[Messer *et al.*, 1999] K. Messer, J. Matas, J. Kittler, and K. Jonsson. Xm2vtsdb: The extended m2vts database. In *In Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, 1999.

[Mohar, 1991] Bojan Mohar. The laplacian spectrum of graphs. In *Graph Theory, Combinatorics, and Applications*, pages 871–898. Wiley, 1991.

[Nene *et al.*, 1996] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-20). Technical report, Technical Report CUCS-005-96, 1996.

[Nie *et al.*, 2007] Feiping Nie, Shiming Xiang, and Changshui Zhang. Neighborhood minmax projections. In *IJCAI*, pages 993–998, 2007.

[Nie *et al.*, 2009] Feiping Nie, Shiming Xiang, Yangqing Jia, and Changshui Zhang. Semi-supervised orthogonal discriminant analysis via label propagation. *Pattern Recognition*, 42(11):2615–2627, 2009.

[Nie *et al.*, 2014a] Feiping Nie, Xiao Cai, and Heng Huang. Flexible shift-invariant locality and globality preserving projections. *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 485–500, 2014.

[Nie *et al.*, 2014b] Feiping Nie, Xiaoqian Wang, and Heng Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 977–986. ACM, 2014.

[Niyogi, 2004] X Niyogi. Locality preserving projections. In *Neural information processing systems*, volume 16, page 153, 2004.

[Rodriguez and Laio, 2014] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

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

[Wang *et al.*, 2014] De Wang, Feiping Nie, and Heng Huang. Unsupervised feature selection via unified trace ratio formulation and k-means clustering (track). *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 306–321, 2014.

[Welling, 2005] Max Welling. Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto*, 3, 2005.

[Yu and Yang, 2001] Hua Yu and Jie Yang. A direct lda algorithm for high-dimensional datałwith application to face recognition. *Pattern recognition*, 34(10):2067–2070, 2001.

[Zhang *et al.*, 2010] Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel pca. *Neurocomputing*, 73(4):959–967, 2010.