LEARNING STRICTLY ORTHOGONAL p-ORDER NONNEGATIVE LAPLACIAN EMBEDDING VIA SMOOTHED ITERATIVE REWEIGHTED METHOD

by

Haoxuan Yang

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Computer Science).

Golden, Colorado

Date _____

Signed: _____
Haoxuan Yang

Signed: _____
Dr. Hua Wang
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Tracy Camp
Department Head and Professor
Department of Computer Science Department

ABSTRACT

Laplacian embedding is a powerful graph based method with its ability in spectral clustering to reveal the intrinsic geometry of data in the high dimensional space. Imposing the orthogonality and the nonnegativity constraints can avoid degenerate and negative solutions, respectively. These two attributes are critical yet challenging to achieve simultaneously.

Although, in recent years, many attempts have been made to overcome this, this problem is still not perfectly handled. We propose an effective algorithm to solve the Laplacian embedding problem that satisfies the both constraints. To promote the robustness of our embedding model against outliers, we exploit the p-order of the ℓ_2-norm distances to find the best solution of the spectral embedding from the input graph.

Optimization with both orthonormal and nonnegative constraints is highly nonlinear and nonconvex in feasible domain. The p-order term in our objective further makes it nonsmooth and difficult to efficiently solve in general. We introduce a novel **smoothed iterative reweighted method with a smoothness term** to tackle this challenging optimization problem and rigorously analyze its convergence. We demonstrate the effectiveness and potential of our proposed method by extensive empirical studies on both synthetic and real data sets.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

Alternating Direction Method of Multipliers . . . . . ADMM

Karush-Kuhn-Tucker . . . . . KKT

Laplacian Embedding . . . . . LE

Non-negative Laplacian Embedding . . . . . NLE

Normalized Cut . . . . . NCut

Very-large-scale integration . . . . . VLSI

p-Order Nonnegative Laplacian Embedding . . . . . PO-NLE

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Dr. Hua Wang, who continually and convincingly conveys a spirit of adventure in regard to research. He deserves thanks for many things. Notably, for giving many suggestions, sharing his knowledge and creating the research environment in which I have performed my graduate studies. He has provided really useful guidance at key moments in my work while at the same time allowing me to work independently the majority of the time. I also would like to thank Dr. Kai Liu who gave me some valuable advice and support in my research. Additionally, I consider myself fortunate indeed to have Dr. Dinesh Mehta and Dr. Dejun Yang as my committee members. This thesis dissertation would not have been possible without the support of all these people.

I am grateful to my parents who have provided me moral and emotional support in my life. They encouraged and educated me in a way that helped me develop my interest in science early in my life. I am also grateful to my other family members and friends who have supported me along the way.

CHAPTER 1

INTRODUCTION

1.1 Dimensionality Reduction

Data resided in high dimensional space makes it intractable due to the lack of intuition. In fact, many data sets in real world applications are low dimensional data lying in high dimensional space. For example, the images in face recognition task are usually larger than 60×60, which correspond to a vector with more than 3600 dimensions, but not every single pixel in the picture can help identify the face. Developing appropriate representations for complex data is one of the central problems in machine learning and recognition. Dimensionality reduction is designed to reduce the number of features in data and minimize the redundancy. It can also discover the intrinsic structure or latent variables which can better explain the data so that the irrelevant and noisy features are removed. The idea of dimensionality reduction is to reduce the number of feature of the data to a more manageable and less redundant value. For instance, the pixel in high resolution images tends to have the same value as the nearby pixels. The embedding strategy is one of the most useful tool in dimensionality reduction. An embedding is translation process from high dimensional space into relatively low dimensional space. Embeddings make it easier to do machine learning applications such as information retrieval and data mining with large or sparse inputs. Figure 1.1 shows an intuitive example of dimensionality reduction.

1.2 Embedding

Generally, linear and non-linear embedding are two main types of widely used embedding approaches. Linear transformation has been used in the linear embedding to embed the data into a linear space such as Principle Component Analysis (PCA) [1] which seeks to maximize the covariance among data points. PCA identifies dimensions that are orthogonal to each other. These dimensions, which are also known as the principal components, are developed based on the original data. The redundancy of the data is minimized due to the orthogonality of the dimensions.

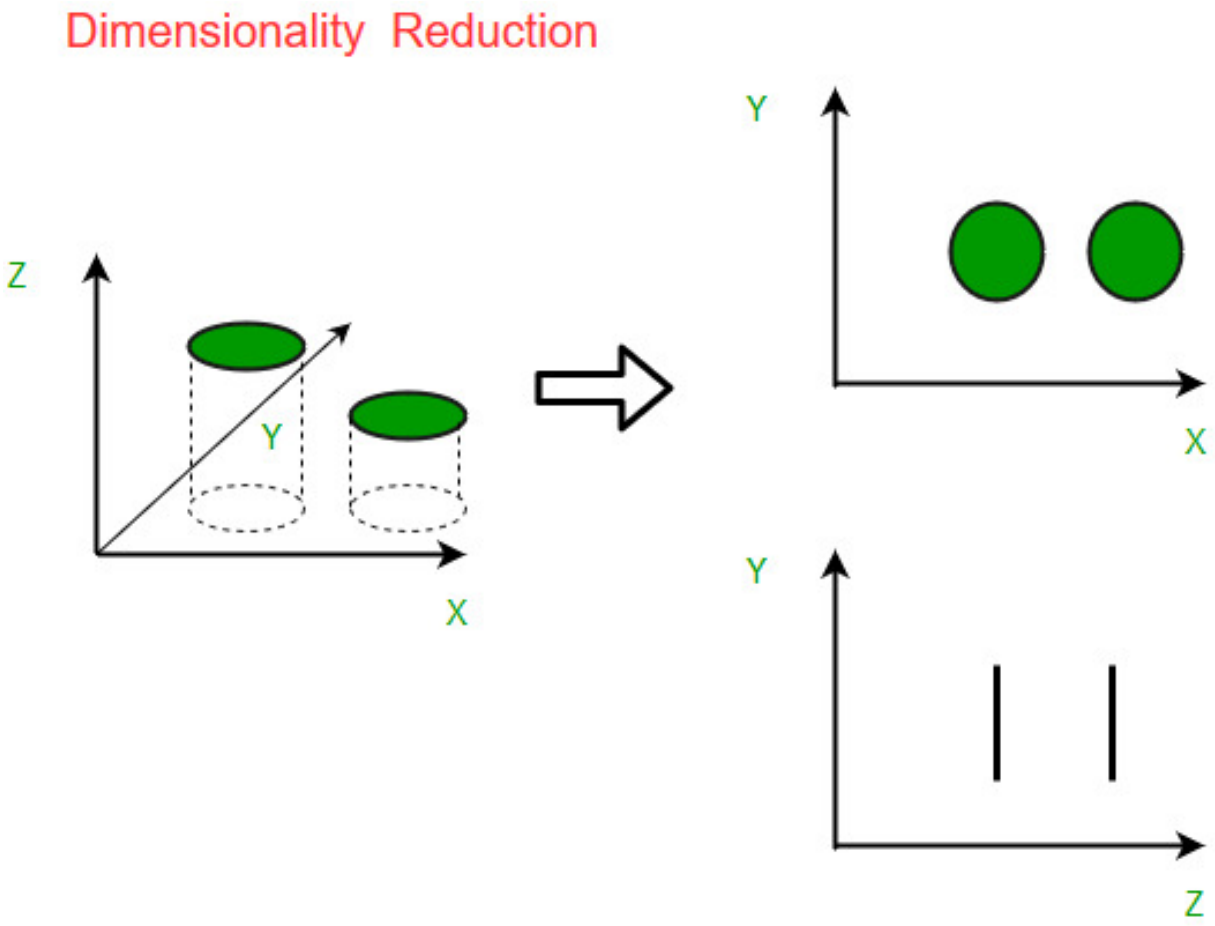


Figure 1.1: A three Dimensional objects can be mapped to a two dimensional space, or even further mapped to a simple line which is one dimensional space. This figure is provided by https://www.geeksforgeeks.org/dimensionality-reduction

.

PCA is aiming to maximize the following objective:

$$\max_{PP^T=I} \mathbf{tr}\left(P^T Q P^T\right) \quad , \tag{1.1}$$

where $\mathbf{Q}$ is a diagonal of eigenvalues of the data and $\mathbf{P}$ is the resultant projection. After the maximization, the learned projection $\mathbf{P}$ can map the original data to a new transformed space, in which the principal components are orthogonal and ordered. In this way, the first few principal components based on the ordering in the eigenvalue decmposition are often sufficient enough to represent the original data. Figure 1.2 shows an intuitive example of PCA in a simple case.

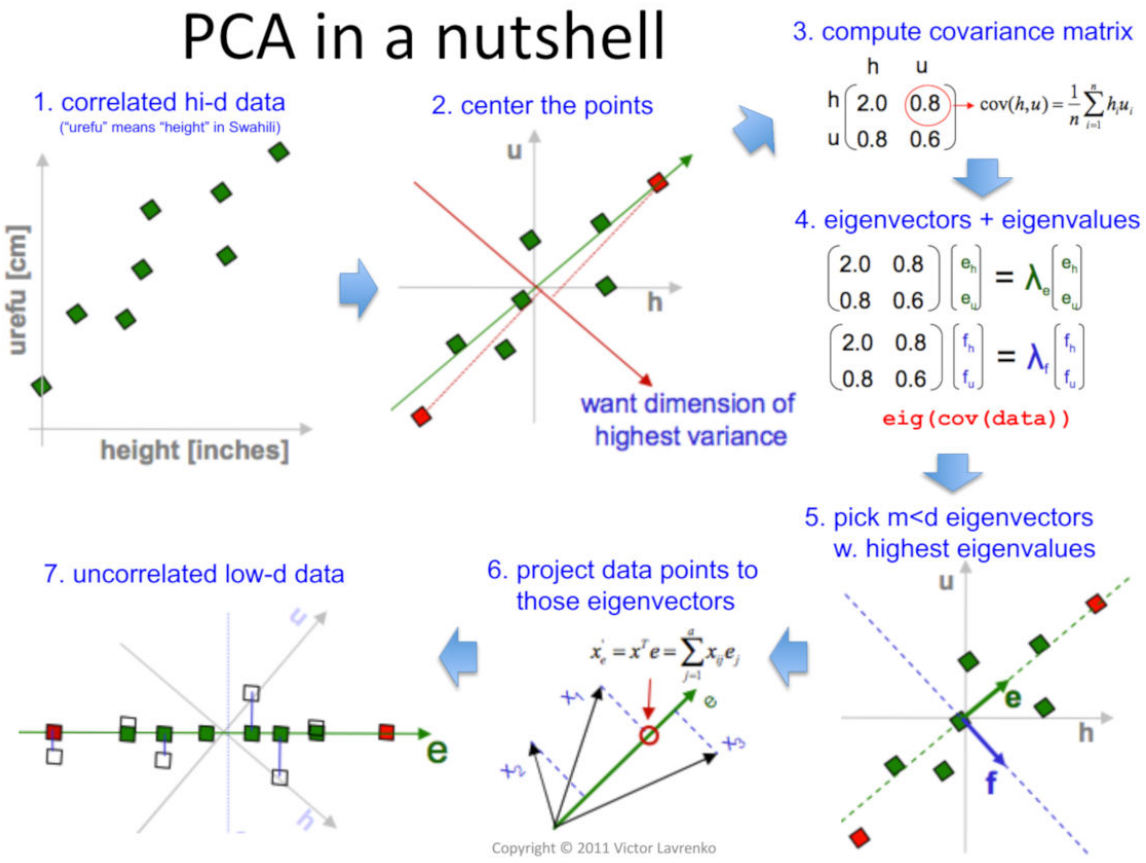


Figure 1.2: Consider a dataset in only two dimensions which can be plotted in a plane. PCA finds a new coordinate system to tease out the highest variation from the data. The new axis, "e", in the figure does not mean anything physical. It is called "principal component" that is chosen to give one axis a lot of variation. This figure is provided by https://devopedia.org/principal-component-analysis.

There are some other different linear embedding approaches. For example, Latent Dirichlet Allocation (LDA) [2] aims at maximizing the separability between classes. However, linear methods can be effective only when data exhibits linear trends. They often fail to map the data which exhibits relational structures such as a manifold. Data which resides in the non-linear space is quite common in the real world and it is unsuitable to use these methods to discover the locality information encoded in the data.

Non-linear embedding approaches are designed to tackle the problem mentioned above. Some typical algorithms are Locally Linear Embedding (LLE) [3], IsoMAP [4], Local Tangent Space Alignment (LTSA) [5], Locality Preserving Projection (LPP) [6], Structure Preserving Embedding (SPE) [7], etc. They all have different purposes and can detect the non-linear manifold embedded in the data.

1.3 Laplacian Embedding

Laplacian Embedding is a very unique non-linear approach because of its relation to graph clustering [8–11] and the usage of eigenvectors of graph Laplacian matrix. In traditional Laplacian embedding, the intrinsic subspace/manifold in high-dimensional space can be explored in such a way that the inherent data structures are well reserved and made more apparent due to the fact that the features less related to others will be rigorously pruned. Laplacian embedding is a very special nonlinear graph based embedding method which was first introduced as "quadratic placement" in 1970s [12]. This model has found a myriad of applications in many industrial tasks and becomes popular in 1990s because of the circuit layout in VLSI community [13] and graph partitioning [14] in domain fundamental problem of distributed memory computing. Recently, the real power of Laplacian embedding was revealed as its relation to graph clustering [8–10]. The eigenvector of Laplacian matrix provides the approximation solution to the Ratio Cut spectral clustering [9] and it has been mathematically proved that Laplacian embedding and ratio cut clustering are actually identical [15]. Figure 1.3 and Figure 1.4 provide a typical example of the performance of Laplacian Embedding on a synthetic swiss roll data. Figure 1.5 shows the result of PCA approach for a

comparision.

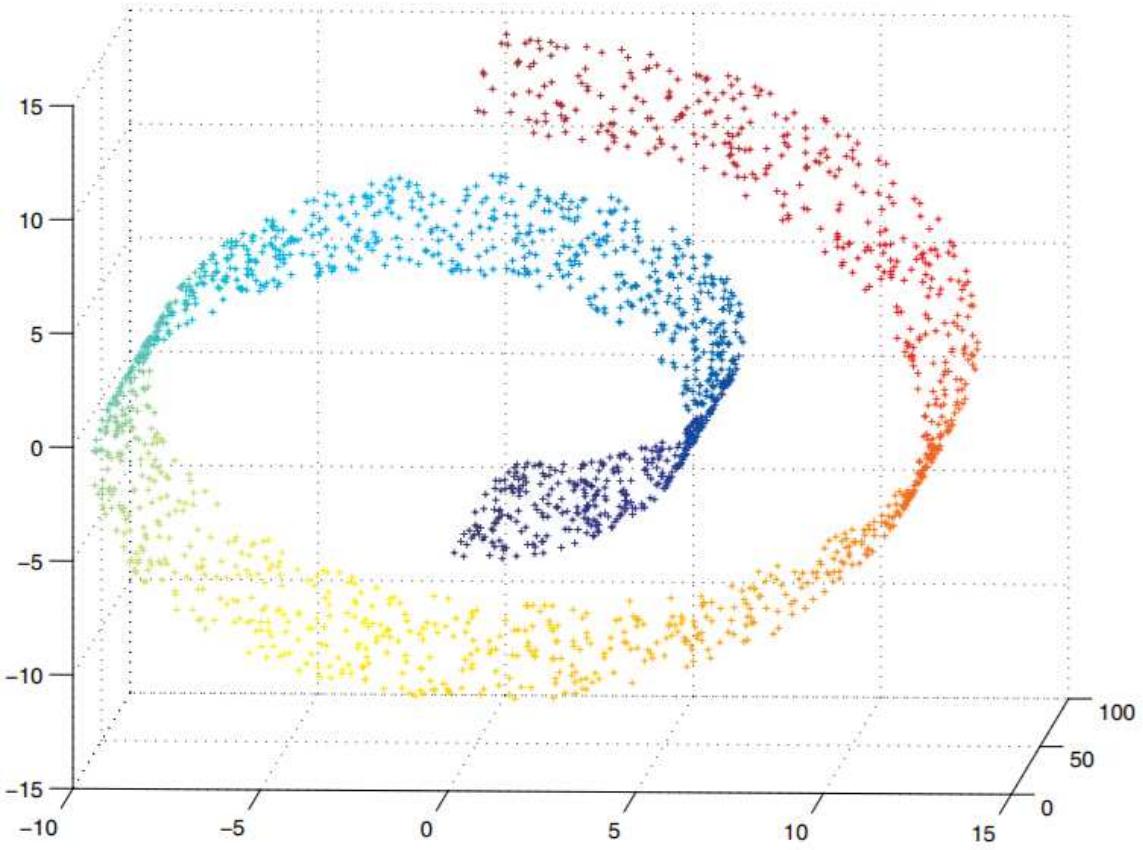


Figure 1.3: The synthetic data set of 2000 points on a swiss roll is shown in Figure 1. This swiss roll is acutally a flat two dimensional submanifold resided in a three dimensional space. This figure is provided by [16].

Given a set of n data points, we can represent the pairwise similarities between these data points by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the data points are represented by the vertices $\mathcal{V}$ and $|\mathcal{V}| = n$. Suppose that $U \in \Re^{n \times n}$ denotes the affinity matrix of the graph $\mathcal{G}$ where w_{ij} measures the similarity between the i-th and the j-th vertices, quadratic placement [12] aims to embed the vertices of the graph into the 1-dimensional space with coordinates $(x_1, \ldots, x_n)$, such that if the i-th and the j-th vertices are similar (*i.e.*, w_{ij} is large), they should be adjacent in embedded space, *i.e.*, $(x_i - x_j)^2$ should be small. This can be achieved by [12]:

$$\min_{\|\mathbf{x}\|_2^2=1} \sum_{i,j} w_{ij} \left(x_i - x_j\right)^2 = 2\mathbf{x}^T \left(D - U\right) \mathbf{x} \ , \tag{1.2}$$

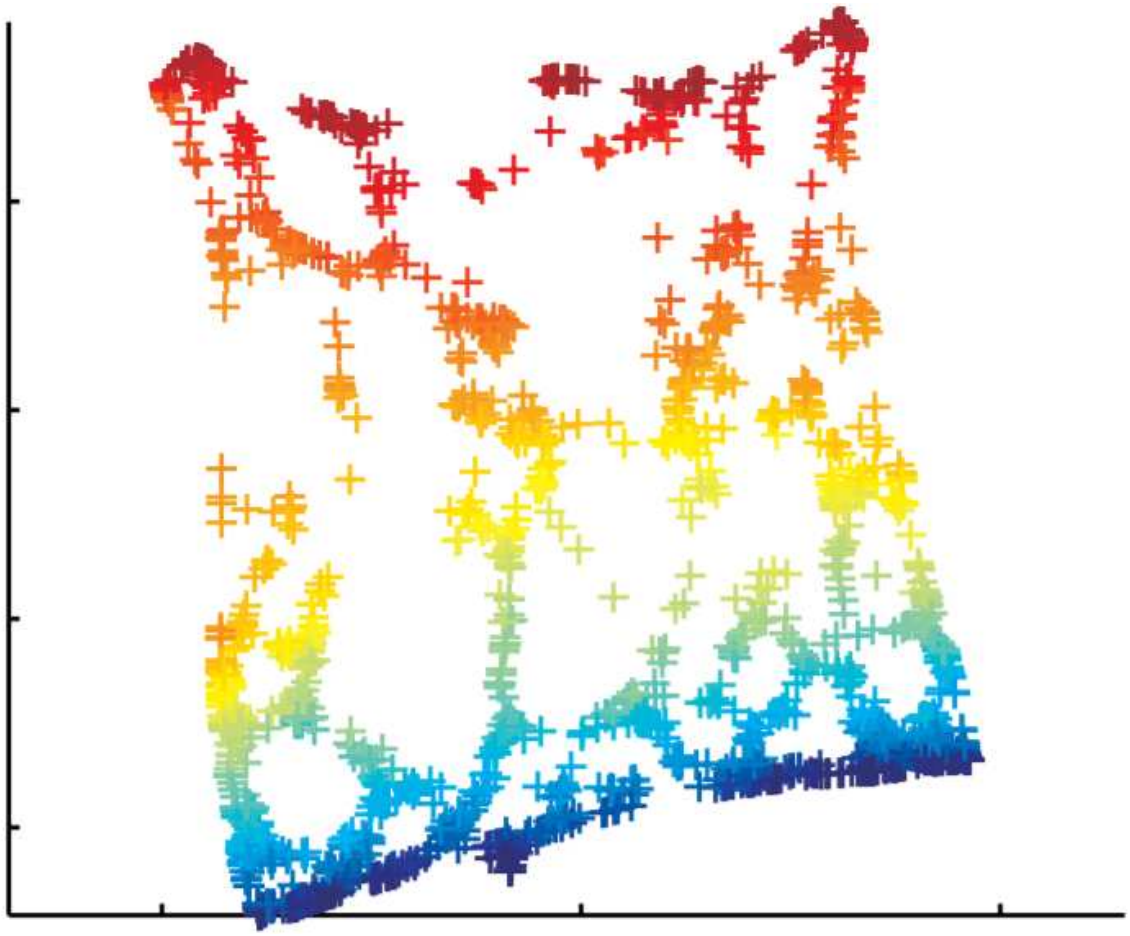


Figure 1.4: The result of Laplacian Embedding performed on a synthetic swiss roll data looks like an ”unfolded” two-dimensional representations of the swiss roll. This figure is provided by [16].

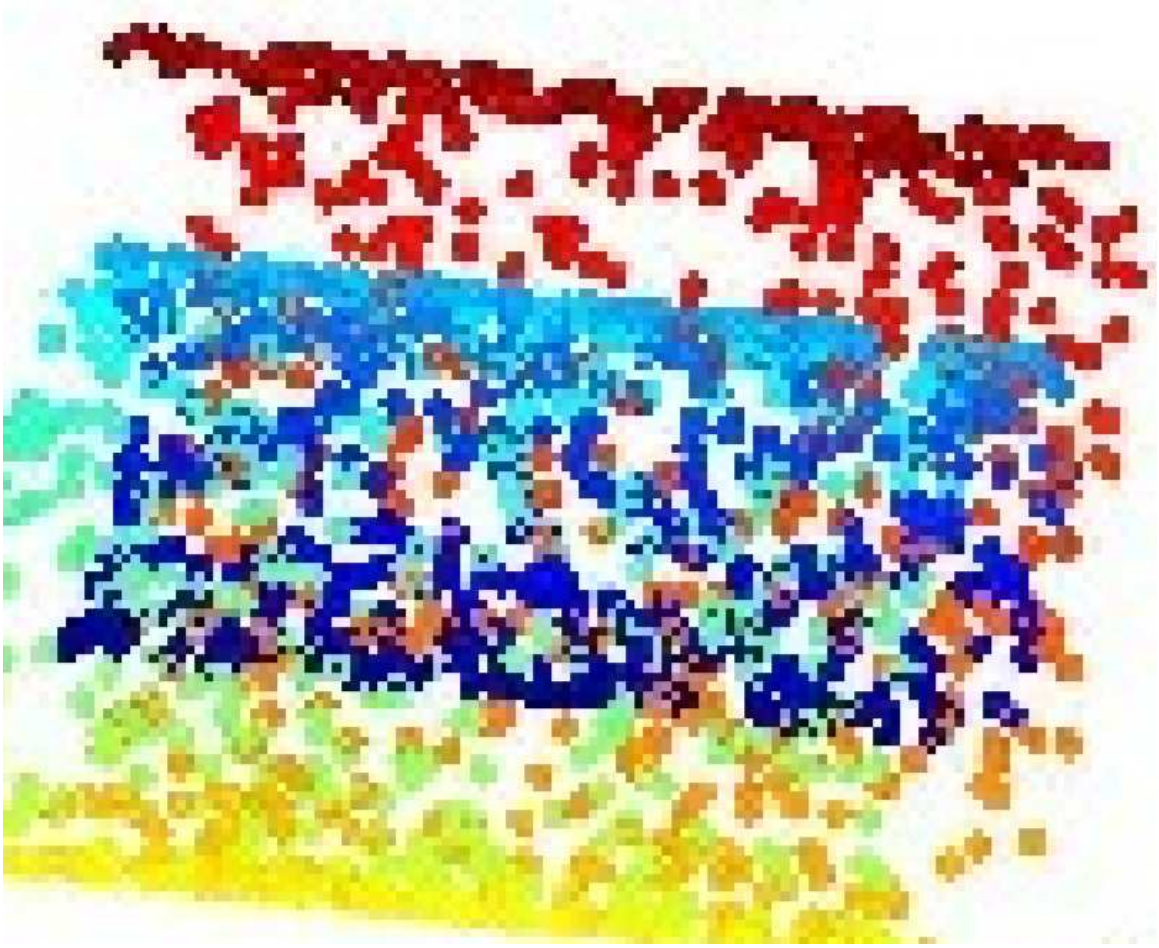


Figure 1.5: The two-dimensional result of PCA performed on a synthetic swiss roll data fails to ”unfold” the swiss roll. This figure is provided by https://www.cs.cmu.edu/ efros/courses/AP06/presentations/melchior_isomap_demo.pdf.

where $\mathbf{x} = [x_1, \ldots, x_n]^T$, and $D = diag(d_1, d_2, \ldots, d_n)$ is the degree matrix of the graph and $d_i = \sum_j w_{ij}$.

The 1-dimensional quadratic placement in Eq. (1.2) can be generalized to r-dimensional Laplacian embedding, for which we can minimize the following objective [15]:

$$\min_{X^T X = I} \sum_{i,j} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \mathbf{tr}\left(X^T (D - U) X\right) \quad , \tag{1.3}$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \Re^{n \times r}$. Obviously, the i-th row of X, *i.e.*, $\mathbf{x}_i^T \in \Re^r$, is the embedding of the i-th data point in the r-dimensional space. Here, the orthonormal constraint of $X^T X = I$ in Eq. (1.3) is imposed to avoid degenerate solutions, which is important as analyzed in [15, 17].

1.4 Laplacian Embedding and Graph Clustering

As we know, a task of partitioning the vertices of a given graph into clusters is a graph clustering approach where the low weights in the edges between the clusters are desired. Spectral clustering is a powerful tool for graph clustering which is originated from spectral graph partitioning [14, 18, 19]. Min-cut algorithm was proposed to minimize the cut weights. However, this often leads to a highly unbalanced partition due to the fact that very small subgraphs are more likely to be divided out which is not desired. This has led to a simple version Ratio Cut solution proposed by Cheng and Wei[20] and it was developed in circuit placement field. Hagen and Kahng [8] later provide a more powerful Ratio Cut solution using Fiedler vector (second eigenvector of the graph Laplacian). Chan et al. [9] further generalized this two-way Ratio Cut clustering to multi-way clustering and show that the embedding of the objective function is identical to the Laplacian Embedding with the same orthogonality constraint. Shi and Malik [10] propose a Normalized-cut clustering solution by using the eigenvector of the generalized/normalized Laplacian [21]. Ding et al [22] further developed this into the min-max cut clustering.

The real power of Laplacian Embedding is graph based clustering due to the fundamental relationship between Laplacian Embedding and Ratio Cut spectral clustering. These two things are actually identical [15]. In fact, the approximation solution of the Ratio Cut clustering can be

provided by the eigenvectors of the Laplacian matrix [23]. Also, the generalized eigenvectors of the Laplacian matrix gives an approximation solution of the Normalized Cut clustering [24] and min-max clustering [22]. As a powerful tool for manifold learning and non-linear dimensionality reduction approach, Laplacian Embedding has been intensively investigated in many studies [15, 25–30].

1.5 Nonnegative Laplacian Embedding

However, both positive and negative values in the solution of multi-way clustering tasks make the results hard to interpret directly, because the clustering indicator vectors require nonnegative results. In two-way clustering, this is not a problem because a linear Ψ-transformation [31] of the eigenvectors leads to two genuine indicator vectors (each row has only one nonzero positive entry). Thus, mixed-sign solution is a generic difficulty for multi-way spectral clustering. To tackle this problem, a clustering task is performed after the embedding. That is, in traditional way, the clustering indicator vectors approximated by the eigenvectors of the Laplacian matrix will be grouped by using K-means clustering [32] in the eigenvector space. Thus, the traditional clustering solution provided from this process is neither stable nor intuitive. It is also very sensitive to the data outliers. To tackle this difficulty and for easier decoding the clustering membership from X, Luo et al. (2009) further developed Laplacian embedding by additionally imposing the nonnegative constraint on the embedding matrix X, which minimizes the following objective:

$$\min_{X} \ \mathbf{tr}\left(X^T\left(D-U\right)X\right), \quad s.t.\ X \geq 0, X^TX = I \ . \tag{1.4}$$

1.6 Orthogonality is Not Guaranteed by Auxillary Function

Despite the fact that the nonnegativity can be achieved in the NLE method, there are still some difficulties of this model that are not well handled. It has been noted that the NLE method imposes the nonnegative constraint at the cost of relaxing the orthogonality on the approximation solution [17], while the orthogonality constraint ($\mathbf{X}^T\mathbf{X} = \mathbf{I}$) is of significant importance to guarantee a good

performance. The true meaning of the orthogonality constraint is to prevent degenerate solution ($\mathbf{X} \rightarrow 0$). For one dimensional problem, the orthogonality can avoid that the embedded data collapse onto a point. For multi-dimensional problem, the orthogonality can prevent data points from collapsing onto a subspace with dimension less than desired.

1.7 Strictly Orthogonally Constrained Nonnegative Laplacian embedding Provides a Solution

In this proposal, we propose a new approach to perform the Laplacian embedding with strictly guaranteed orthogonality and nonnegativity in the solution. Unlike the auxiliary function method [33] used in NLE [15], the orthogonality of our solution is rigorously achieved by using the Alternating Direction Method of Multipliers (ADMM) [34, 35], leading to a more stable solution and a better performance in the spectral clustering problem. We also keep the non-negativity in the constraint. As a result, the clustering membership can be easily read off from the embedded data due to the nonnegative constraint, *i.e.*, we can consider each row of the solution $\mathbf{X}$ as the posterior clustering probability. In other words, the values in i-th row of the solution can be viewed as the likelihoods that the i-th data point belongs to different clusters, which gives our new approach soft clustering capability that is crucial in many real world applications.

we can easily identify the cluster membership from the solution:

$$s = \arg\max_{1 \leq j \leq s} X_{ij}. \tag{1.5}$$

where s represents the largest component in the i-th row of solution $\mathbf{X}$ while s is the number of total classes among the embedded data. due to the more desired cluster indicators, this approach more clearly exposes the intrinsic structure properties of the data in nonliear embedding problem.

1.8 A More Robust Approach with p-order

Finally, we recognize that the squared ℓ_2-norm distance used in traditional Laplacian embedding objectives does not guarantee the optimal embedding and is notoriously known to be sensitive to the outliers. With strict orthogonality and nonnegativity guaranteed simultaneously in the solution, we are also interested in promote the robustness of our new NLE model by using the p-th

order ($0 < p \leq 2$) of the ℓ_2-norm distance in the objective. Optimization problem for quadratic function with both orthonormal and nonnegative constraints is highly nonlinear and nonconvex in feasible domain. The p-th order term further makes the objective nonsmooth and difficult to optimize. To solve this challenging optimization problem, we propose a novel ***smoothed iterative reweighted method***. Compared to the iterative reweighted method proposed in [36] to solve the $\ell_{2,1}$-norm minimization, our new optimization framework explicitly adds a smoothness term for improved numerical stability. Most importantly, as an important theoretical contribution, we rigorously prove the convergence of the new iterative algorithm, which, though, was not present in [36] and its following works.

To summarize, we propose a new nonnegative Laplacian embedding model that is interesting from a number of perspectives as follows:

We derive an algorithm using iterative reweighted method and formally introduce a smoothness term to solve the general nonsmooth problem in optimization, as one of the most important novelty of this paper. We also rigorously prove the convergence of iterative reweighted method with explicit smoothness terms, which plays a critical role in solving the future problem of this type.

Beside imposing the nonnegative constraint to guarantee that the cluster membership can be read off directly, our algorithm can also achieve absolute strict orthogonality simultaneously, which is crucial to avoid degenerate solutions and thus generates more promising results.

We propose to use the p-th order ($0 < p \leq 2$) of the ℓ_2-norm distances in our objective to learn the robust embeddings against noises and outliers.

Soft clustering capability of our method gives different views to interpret the solution. For example, the solution can be considered as posterior probability of different clusters.

CHAPTER 2

STRICTLY ORTHOGONAL p-ORDER NONNEGATIVE LAPLACIAN EMBEDDING

The squared ℓ_2-norm distances used in the both objectives in Eq. (1.3) and Eq. (1.4) do not tolerate large value of distance, thus making the distances in the embedded space tend to be even, *i.e.*, not too large but also not too small. Therefore, solving the objective in Eq. (1.3) or Eq. (1.4) may not find the optimal embedding such that most of the distances of local data pairs are minimized but a few of them are large [37]. Motivated by recent papers that use not-squared ℓ_2-norm distances [38–41] or the p-th order of the ℓ_2-norm distances [37, 42, 43] to promote the robustness of learning models, in this proposal we propose to solve the following problem to find the optimal spectral embedding from an input graph:

$$\min_{X} \sum_{i,j} w_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^p \ , \quad s.t.\ X \geq 0, X^T X = I \ , \tag{2.1}$$

where $0 < p \leq 2$. Obviously, the NLE method in Eq. (1.4) proposed by [15] is a special case of our proposed new method when $p = 2$. More importantly, by setting $p \leq 1$, the method will focus on minimizing most of the distances of local data pairs. We call Eq. (2.1) as the proposed strictly orthogonal p-Order Nonnegative Laplacian Embedding (PO-NLE) method.

2.1 Smoothed Iterative Reweighted Method and Its Convergence

Although the motivation of our new method in Eq. (2.1) is clear, it is a nonsmooth objective and difficult to efficiently solve in general. Thus, in this section, we first introduce a novel smoothed iterative reweighted method to solve this challenging optimization problem.

First, let us consider a general problem as follows:

$$\min_{x \in \mathcal{C}} f(x) + \sum_i tr\left((g_i^T(x) g_i(x))^{\frac{p}{2}} \right) \ . \tag{2.2}$$

When $g_i(x)$ is a vector output function, $tr\left((g_i^T(x) g_i(x))^{\frac{p}{2}} \right)$ becomes the following term:

$$tr\left((g_i^T(x) g_i(x))^{\frac{p}{2}} \right) = \left\| g_i(x) \right\|_2^p \ . \tag{2.3}$$

Equation (2.2) is nonsmooth, thus we turn to solve the following smooth problem:

$$\min_{x\in C} f(x) + \sum_i tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) \ , \tag{2.4}$$

where $\delta > 0$ is a small constant. When $\delta \to 0$, Eq. (2.4) is reduced to Eq. (2.2) since the following equation holds:

$$\lim_{\delta\to 0} tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) = \|g_i(x)\|_2^p \ . \tag{2.5}$$

Before deriving the algorithm for optimizing the problem in Eq. (2.4), we need the following lemmas. First, according to the chain rule in calculus, we have:

Lemma 1 *Suppose $g(x)$ is a matrix output function, $h(x)$ is a scalar output function, x is a scalar, vector or matrix variable, then we have*

$$\frac{\partial h(g(x))}{\partial x} = \frac{\sum_{i,j} \frac{\partial h(g(x))}{\partial g_{ij}(x)} \partial g_{ij}(x)}{\partial x} = \left(\frac{\partial h(g(x))}{\partial g(x)}\right)^T \frac{\partial g(x)}{\partial x} \ . \tag{2.6}$$

According to the chain rule in Lemma 1, we can easily derive the following two lemmas:

Lemma 2 *Suppose $g(x)$ is a scalar, vector or matrix output function, x is a scalar, vector or matrix variable, then we have*

$$\begin{aligned} & \frac{\partial tr((g^T(x)g(x) + \delta I)^{\frac{p}{2}})}{\partial x} \\ = \ & 2\,\frac{p}{2}(g^T(x)g(x) + \delta I)^{\frac{p-2}{2}} g^T(x)\frac{\partial g(x)}{\partial x} \\ = \ & p\,(g^T(x)g(x) + \delta I)^{\frac{p-2}{2}} g^T(x)\frac{\partial g(x)}{\partial x} \ . \end{aligned} \tag{2.7}$$

Lemma 3 *Suppose $g(x)$ is a scalar, vector or matrix output function, x is a scalar, vector or matrix variable, D is a constant and D is symmetrical if D is a matrix, then we have*

$$\frac{\partial tr(g^T(x)g(x)D)}{\partial x} = 2Dg^T(x)\frac{\partial g(x)}{\partial x} \ . \tag{2.8}$$

Now we derive the algorithm to optimize the problem in Eq. (2.4). The Lagrangian function of the problem in Eq. (2.4) is

$$\mathcal{L}(x,\lambda) = f(x) + \sum_i tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) - r(x,\lambda) \ , \tag{2.9}$$

where $r(x, \lambda)$ is a Lagrangian term for the constraint $x \in \mathcal{C}$. By setting the derivative of Eq.(2.9) w.r.t. x to zero, we have

$$\frac{\partial L(x,\lambda)}{\partial x} = f'(x) + \sum_i \frac{\partial tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}})}{\partial x} - \frac{\partial r(x,\lambda)}{\partial x} = 0 \ . \tag{2.10}$$

According to Lemma 2, Eq.(2.10) can be rewritten as

$$\begin{aligned} f'(x) + \sum_i p(g_i^T(x)g_i(x) + \delta I)^{\frac{p-2}{2}} g_i^T(x)\frac{\partial g_i(x)}{\partial x} \\ - \frac{\partial r(x,\lambda)}{\partial x} = 0 \ . \end{aligned} \tag{2.11}$$

If we can find a solution x that satisfies the Eq.(2.11), we usually find a local or global optimal solution to the problem in Eq. (2.4) according to the Karush-Kuhn-Tucker (KKT) conditions. However, directly finding a solution x that satisfies Eq.(2.11) is not a easy task. In this proposal, we propose an iterative algorithm to find it. A basic observation is that, if $D_i = \frac{p}{2}(g_i^T(x)g_i(x) + \delta I)^{\frac{p-2}{2}}$ is a constant, Eq.(2.11) is reduced to

$$f'(x) + \sum_i 2D_i g_i^T(x)\frac{\partial g_i(x)}{\partial x} - \frac{\partial r(x,\lambda)}{\partial x} = 0 \ , \tag{2.12}$$

which is equivalent to solving the following problem:

$$\min_{x\in\mathcal{C}} f(x) + \sum_i tr(g_i^T(x)g_i(x)D_i) \ . \tag{2.13}$$

Based on the observation above, we first guess a solution x, then we calculate D_i based on the current solution x and then update the current solution x by the optimal solution of the problem (2.13) based on the calculated D_i. We iteratively perform this procedure until it converges. This algorithm is summarized in Algorithm 1.

Algorithm 1: The algorithm to solve the smooth problem.

Initialize $x \in \mathcal{C}$;

while *not converge* **do**

1. For each i, calculate $D_i = \frac{p}{2}(g_i^T(x)g_i(x) + \delta I)^{\frac{p-2}{2}}$;
2. Update x by solving the problem $\min_{x\in\mathcal{C}} f(x) + \sum_i tr(g_i^T(x)g_i(x)D_i)$;

Output: x.

The convergence of Algorithm 1 is guaranteed by the following theorem.

Theorem 1 *The Algorithm 1 will monotonically decrease the objective of the problem (2.4) in each iteration until the algorithm converges.*

Before proving the convergence of the Algorithm 1, we first introduce several lemmas:

Lemma 4 *For any $\sigma > 0$, the following inequality holds when $0 < p \leq 2$:*

$$\frac{p}{2}\sigma - \sigma^{\frac{p}{2}} + \frac{2-p}{2} \geq 0 \ . \tag{2.14}$$

Proof. Denote $f(\sigma) = p\sigma - 2\sigma^{\frac{p}{2}} + 2 - p$, we have the following derivatives:

$$f'(\sigma) = p(1 - \sigma^{\frac{p-2}{2}}), \quad \text{and} \quad f''(\sigma) = \frac{p(2-p)}{2}\sigma_i^{\frac{p-4}{2}}.$$

Obviously, when $\sigma > 0$ and $0 < p \leq 2$, then $f''(\sigma) \geq 0$ and $\sigma = 1$ is the only point that $f'(\sigma) = 0$. Note that $f(1) = 0$, thus when $\sigma > 0$ and $0 < p \leq 2$, then $f(\sigma) \geq 0$, which indicates Eq.(2.14). □

Lemma 5 ([44]) *For any positive definite matrices $\tilde{M}, M$ with the same size, suppose the eigen-decomposition $\tilde{M} = U\Sigma U^T$, $M = V\Lambda V^T$, where the eigenvalues in Σ is in increasing order and the eigenvalues in Λ is in decreasing order. Then the following inequality holds when $0 < p \leq 2$:*

$$Tr(\tilde{M}M) \geq Tr(\Sigma\Lambda) \ . \tag{2.15}$$

Lemma 6 *For any positive definite matrices $\tilde{M}, M$ with the same size, the following inequality holds when $0 < p \leq 2$:*

$$tr(\tilde{M}^{\frac{p}{2}}) - \frac{p}{2}tr(\tilde{M}M^{\frac{p-2}{2}}) \leq tr(M^{\frac{p}{2}}) - \frac{p}{2}tr(MM^{\frac{p-2}{2}}). \tag{2.16}$$

Proof. For any $\sigma > 0$, $\lambda > 0$ and $0 < p \leq 2$, according to Lemma 4 we have $\frac{p}{2}(\frac{\sigma}{\lambda}) - (\frac{\sigma}{\lambda})^{\frac{p}{2}} + \frac{2-p}{2} \geq 0$, which indicates

$$\frac{p}{2}\sigma\lambda^{\frac{p-2}{2}} - \sigma^{\frac{p}{2}} + \frac{2-p}{2}\lambda^{\frac{p}{2}} \geq 0 \ . \tag{2.17}$$

Suppose the eigen-decomposition $\tilde{M} = U\Sigma U^T$, $M = V\Lambda V^T$, where the eigenvalues in Σ is in increasing order and the eigenvalues in Λ is in decreasing order. Then according to Eq.(2.17), we

have

$$\frac{p}{2}tr(\Sigma\Lambda^{\frac{p-2}{2}}) - tr(\Sigma^{\frac{p}{2}}) + \frac{2-p}{2}tr(\Lambda^{\frac{p}{2}}) \geq 0 \ , \tag{2.18}$$

and according to Lemma 5 we have

$$\frac{p}{2}tr(\tilde{M}M^{\frac{p-2}{2}}) - \frac{p}{2}tr(\Sigma\Lambda^{\frac{p-2}{2}}) \geq 0 \ , \tag{2.19}$$

$$\frac{p}{2}tr(\tilde{M}M^{\frac{p-2}{2}}) - tr(\Sigma^{\frac{p}{2}}) + \frac{2-p}{2}tr(\Lambda^{\frac{p}{2}}) \geq 0 \ . \tag{2.20}$$

Note that $tr(\tilde{M}^{\frac{p}{2}}) = tr(\Sigma^{\frac{p}{2}})$ and $tr(M^{\frac{p}{2}}) = tr(\Lambda^{\frac{p}{2}})$, so we have

$$\begin{aligned}
&\tfrac{p}{2}tr(\tilde{M}M^{\frac{p-2}{2}}) - tr(\tilde{M}^{\frac{p}{2}}) + \tfrac{2-p}{2}tr(M^{\frac{p}{2}}) \geq 0 \\
&\Rightarrow tr(\tilde{M}^{\frac{p}{2}}) - \tfrac{p}{2}tr(\tilde{M}M^{\frac{p-2}{2}}) \leq \tfrac{2-p}{2}tr(M^{\frac{p}{2}}) \\
&\Rightarrow tr(\tilde{M}^{\frac{p}{2}}) - \tfrac{p}{2}tr(\tilde{M}M^{\frac{p-2}{2}}) \leq tr(M^{\frac{p}{2}}) - \tfrac{p}{2}tr(MM^{\frac{p-2}{2}}) \ ,
\end{aligned}$$

which completes the proof. □

Lemma 7 *For any matrices $\tilde{A}, A$ with the same size and $\delta > 0$, the following inequality holds when $0 < p \leq 2$:*

$$\begin{aligned}
&tr((\tilde{A}^T\tilde{A} + \delta I)^{\frac{p}{2}}) - \tfrac{p}{2}tr(\tilde{A}^T\tilde{A}(A^TA + \delta I)^{\frac{p-2}{2}}) \\
&\leq tr((A^TA + \delta I)^{\frac{p}{2}}) - \tfrac{p}{2}tr(A^TA(A^TA + \delta I)^{\frac{p-2}{2}}) \ .
\end{aligned} \tag{2.21}$$

Proof. Note that $\tilde{A}^T\tilde{A} + \delta I$ and $A^TA + \delta I$ are positive definite matrices since $\delta > 0$. Then according to Lemma 6 we have

$$\begin{aligned}
&tr((\tilde{A}^T\tilde{A} + \delta I)^{\frac{p}{2}}) - \tfrac{p}{2}tr((\tilde{A}^T\tilde{A} + \delta I)(A^TA + \delta I)^{\frac{p-2}{2}}) \\
&\leq tr((A^TA + \delta I)^{\frac{p}{2}}) - \tfrac{p}{2}tr((A^TA + \delta I)(A^TA + \delta I)^{\frac{p-2}{2}}) \ ,
\end{aligned} \tag{2.22}$$

which indicates Eq.(2.21). □

As a result, we can prove Theorem 1 as follows now.

Proof of Theorem 1: In step 2 of Algorithm 1, suppose the updated x is $\tilde{x}$. According to step 2, we know

$$f(\tilde{x}) + \sum_i tr(g_i^T(\tilde{x})g_i(\tilde{x})D_i) \leq f(x) + \sum_i tr(g_i^T(x)g_i(x)D_i) \ , \tag{2.23}$$

where the equality holds when and only when the algorithm converges.

For each i, according to Lemma 7, we have

$$\begin{aligned} &tr((g_i^T(\tilde{x})g_i(\tilde{x}) + \delta I)^{\frac{p}{2}}) - \frac{p}{2}tr(g_i^T(\tilde{x})g_i(\tilde{x})(g_i^T(x)g_i(x) + \delta I)^{\frac{p-2}{2}}) \\ &\leq tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) - \frac{p}{2}tr(g_i^T(x)g_i(x)(g_i^T(x)g_i(x) + \delta I)^{\frac{p-2}{2}}) \ . \end{aligned} \tag{2.24}$$

Note that $D_i = \frac{p}{2}(g_i^T(x)g_i(x) + \delta I)^{\frac{p-2}{2}}$, so for each i we have

$$\begin{aligned} &tr((g_i^T(\tilde{x})g_i(\tilde{x}) + \delta I)^{\frac{p}{2}}) - tr(g_i^T(\tilde{x})g_i(\tilde{x})D_i) \\ &\leq tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) - tr(g_i^T(x)g_i(x)D_i) \ . \end{aligned} \tag{2.25}$$

Then we have

$$\begin{aligned} &\sum_i tr((g_i^T(\tilde{x})g_i(\tilde{x}) + \delta I)^{\frac{p}{2}}) - \sum_i tr(g_i^T(\tilde{x})g_i(\tilde{x})D_i) \\ &\leq \sum_i tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) - \sum_i tr(g_i^T(x)g_i(x)D_i) \ . \end{aligned} \tag{2.26}$$

Summing Eq. (2.23) and Eq. (2.26) in the two sides, we arrive at

$$\begin{aligned} &f(\tilde{x}) + \sum_i tr((g_i^T(\tilde{x})g_i(\tilde{x}) + \delta I)^{\frac{p}{2}}) \leq \\ &f(x) + \sum_i tr((g_i^T(x)g_i(x) + \delta I)^{\frac{p}{2}}) \ . \end{aligned} \tag{2.27}$$

Note that the equality in Eq. (2.27) holds only when the algorithm converges. Thus the Algorithm 1 will monotonically decrease the objective of the problem in Eq. (2.4) in each iteration until the algorithm converges. □

In the convergence, the equality in Eq. (2.11) will hold, thus the KKT condition of problem (2.4) is satisfied. Therefore, the Algorithm 1 will usually converge to a local optimum solution to the problem (2.4). If the problem (2.4) is convex, the Algorithm 1 will converge to a global optimum solution.

Here we **note** that the iterative reweighted method introduced in [36] solves the $\ell_{2,1}$-norms minimization problem, which is nonsmooth. However, the method in [36] does not explicitly use the smoothness constant (*i.e.*, δ in Eq. (2.4)). Without the smoothness term, the algorithm is heavily impacted by the singularity problem due to inverted matrices that divide 0s, which routinely leads to inferior learning performances. To improve the numerical stability, in [36] and the following works by the same group of authors [39, 45], a smoothness term was informally added for em-

pirical purpose. But they only theoretically proved the convergence of the algorithm without the smoothness term and did not provide any theoretical analysis on the objectives using the smoothness term. As an important theoretical contribution, we formally introduce the smoothness term (*i.e.*, δI in Eq. (2.4)) into our algorithm and theoretically prove the convergence of our algorithm in which the smoothness term leads to much more stable solutions. Thus, we call Algorithm 1 as the proposed *Smoothed Iterative Reweighted Method*.

2.2 Algorithm to Solve the Optimization Problem

Equipped with Algorithm 1, we can derive the solution algorithm to the problem in Eq. (2.1) now. According to Step 2 of Algorithm 1 (*i.e.*, Eq. (2.13)), the key step to solve Eq. (2.1) is to solve the following problem:

$$\min_X \sum_{i,j} w_{ij} d_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2 \ , \quad s.t.\ X \geq 0, X^T X = I \ , \tag{2.28}$$

where $d_{ij} = \frac{p}{2} \left(\left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2 + \delta \right)^{\frac{p-2}{2}}$ and $\delta \to 0$.

Denote $\widetilde{\mathbf{W}}_{ij} = w_{ij} d_{ij}$ and let $\widetilde{\mathbf{D}}$ be the diagonal matrix with the i-th diagonal entry as $\sum_j \tilde{w}_{ij} = w_{ij} d_{ij}$. The problem in Eq. (2.28) can be written as following:

$$\begin{aligned} \min_X \sum_{i,j} \tilde{w}_{ij} \left\| \mathbf{x}_i - \mathbf{x}_j \right\|_2^2 &= \mathbf{tr} \left(X^T \left(\widetilde{\mathbf{D}} - \widetilde{\mathbf{W}} \right) X \right) \ , \\ s.t.\ X \geq 0, X^T X &= I \ . \end{aligned} \tag{2.29}$$

Obviously, Eq. (2.29) is identical to the NLE objective in Eq. (1.4) proposed in [15]. In [15], a solution algorithm was derived using the auxiliary function method [33]. However, as analyzed in [17, 46] the orthogonal constraint indeed are not guaranteed, which, though, is very important to avoid degenerate solutions [17]. Thus, instead of using the solution algorithm provided in [15], we derive the solution algorithm to solve Eq. (2.29) using the Alternating Direction Method of Multipliers (ADMM) [34, 35].

The ADMM method solves convex optimization problems by breaking them into smaller pieces

that are easier to handle. Specifically, given the following objective with the equality constraint:

$$\min_{x,z} f(x) + g(z), \qquad s.t. \quad h(x,z) = 0, \tag{2.30}$$

Algorithm 2 solves the problem by decoupling it into subproblems and optimizing each variable while fixing others [34, 35], where y is the Lagrangian multiplier to the constraint h. It is worth noting that Algorithm 2 was proved to converge Q-linearly to the optimal solution [34].

Algorithm 2: The ADMM algorithm.

Set $1 < \rho < 2$ and initialize $\mu > 0$ and y. ;
while not converge **do**
1. Update x by solving $x^{k+1} = \arg\min_x (f(x) + \frac{\mu}{2}\|h(x, z^k) + \frac{y^k}{\mu}\|^2)$.;
2. Update z by solving $z^{k+1} = \arg\min_z (g(z) + \frac{\mu}{2}\|h(x^{k+1}, z) + \frac{y^k}{\mu}\|^2)$;
3. Update y by $y^{k+1} = y^k + \mu h(x^{k+1}, z^{k+1})$;
4. Update μ by $\mu = \rho\mu$.
end while

Denoting $L = \widetilde{\mathbf{D}} - \widetilde{\mathbf{W}}$ for brevity[1], we can write the objective in Eq. (2.29) as following:

$$\min_X \ \mathbf{tr}\left(X^T L X\right), \quad s.t.\ X^T X = I, X \geq 0 \ . \tag{2.31}$$

We can solve Eq. (2.31) by solving the following equivalent optimization problem:

$$\min_{X,Y} \ \mathbf{tr}\left(Y^T L X\right), \quad s.t.\ Y = X, Y^T Y = I, X \geq 0 \ , \tag{2.32}$$

where the constraint of $X^T X = I$ in Eq. (2.31) is implicitly enforced by the constraints of $Y = X$ and $Y^T Y = I$.

According to Step 1 and Step 2 of Algorithm 2, we need to solve the following optimization problem:

$$\begin{aligned} \min_{X,Y,\mathbf{\Lambda}} \ & \mathbf{tr}\left(Y^T L X\right) + \frac{\mu}{2}\left\|Y - X + \frac{1}{\mu}\mathbf{\Lambda}\right\|_{\mathrm{F}}^2, \\ s.t. \quad & Y^T Y = I, X \geq 0 \ , \end{aligned} \tag{2.33}$$

[1]In practice, due to the zero mode of the Laplacian matrix of a graph [47], we compute $L = \widetilde{\mathbf{D}} - \widetilde{\mathbf{W}} + \frac{\widetilde{\mathbf{W}}_{++}}{n^2}\mathbf{e}^T\mathbf{e}$ to ensure that L is positive definite, where $\widetilde{\mathbf{W}}_{++} = \sum_{i,j} \widetilde{\mathbf{W}}$ and $\mathbf{e}$ is the vector with all entries to be 1.

in which we introduced the Lagrangian multiplier $\mathbf{\Lambda}$ for the constraint of $Y = X$. The detailed procedures to solve Eq. (2.33) using the ADMM method is provided in the following steps:

Step 1. Initialization.

Step 2. Solving Y, when we fix the other variable X and the Lagrangian multiplier variable $\mathbf{\Lambda}$:

$$\min_{Y} \ \mathbf{tr}\left(Y^T L X\right) + \frac{\mu}{2}\left\|Y - X + \frac{1}{\mu}\mathbf{\Lambda}\right\|_{\mathrm{F}}^2, \quad s.t. \quad Y^T Y = I \ . \tag{2.34}$$

Denoting $M = \mu X - \mathbf{\Lambda} - LX$, we can write the optimization problem in Eq. (2.34) as following:

$$\max_{Y} \ \mathbf{tr}\left(Y^T M\right) \quad s.t. \quad Y^T Y = I \ . \tag{2.35}$$

According to [45, Theorem 1], the problem in Eq. (2.35) can be solved by computing the SVD of M: if $svd\,(M) = UAV^T$, the solution of Eq. (2.35) is given by UV^T.

Step 3. Solving X, when we fix the other variable Y and the Lagrangian multiplier variable $\mathbf{\Lambda}$:

$$\min_{X} \ \mathbf{tr}\left(Y^T L X\right) + \frac{\mu}{2}\left\|Y - X + \frac{1}{\mu}\mathbf{\Lambda}\right\|_{\mathrm{F}}^2, \quad s.t. \quad X \geq 0 \ . \tag{2.36}$$

Denoting $M = Y + \frac{1}{\mu}\mathbf{\Lambda} - \frac{1}{\mu}L^T Y$, we can write the optimization problem in Eq. (2.36) as following:

$$\min_{X} \ \|X - M\|_{\mathrm{F}}^2, \quad s.t. \quad X \geq 0 \ , \tag{2.37}$$

which can be decoupled to solve the following problem for every entry of X:

$$\min_{x_{ij}} \ (x_{ij} - n_{ij})^2 \ , \quad s.t. \quad x_{ij} \geq 0 \ , \tag{2.38}$$

which can be easily solved as follows: $x_{ij} = \max\,(n_{ij}, 0)$.

Step 4. Update $\mathbf{\Lambda}$ by $\mathbf{\Lambda} = \mathbf{\Lambda} + \mu\,(Y - X)$.

Step 5. Update μ by $\mu = \rho\mu$.

CHAPTER 3

EXPERIMENT AND RESULTS

In this section we empirically evaluate our new method on one synthetic data set, four data sets from the UCI Machine Learning Data Repository, and three image data sets. We will compare our new method against its counterparts: NLE, Normalized Cut (NCut) [10] and Laplacian Embedding (LE). The seven standard data sets are summarized in Table 3.1.

In our evaluations, we use *clustering accuracy* and *clustering purity* to measure the performance of the compared methods. We also study the robustness of our method on the real world data sets when they are contaminated with noise. The performance variations when we increase the value of p will be shown to validate our hypothesis that the optimal solution is usually obtained when p is less than 2 and close to 1 (it depends on data sets), given that the data is noisy. Orthogonality of the solution will be illustrated and compared against the NLE method in [15].

3.1 Experiments on A Synthetic Data Set

To illustrate the effectiveness of our new PO-NLE method, we create a synthetic data set as follows. We first randomly generate 3 data points as centroids in the 30-dimensional space. Then we generate 3 groups of data points and each group consists of 39 data points which are randomly distributed around one of the three centroids. A threshold is set to make the distance of groups large enough. As shown in Figure 3.1, different colors (red, black and blue) and shapes are used to represent different groups of data points. We randomly initialize $\mathbf{X}$ ($0 \leq \mathbf{X} \leq 1$) and set $\rho = 1.02$, $\mu = 0.1$ and $p = 0.8$ in our algorithm. we use K-Nearest Neighbors with heat kernel to construct our adjacency matrix $\mathbf{W}$. The value variations when our algorithm iterates are shown as the red curve in Figure 3.1. For visualization purpose, we set $r = 3$, *i.e.*, we embed the original data into the 3-dimensional space using our new PO-NLE algorithm. The x, y and z axes of 3D plots in the figure correspond to the first, second and third row in matrix $\mathbf{X}$, respectively.

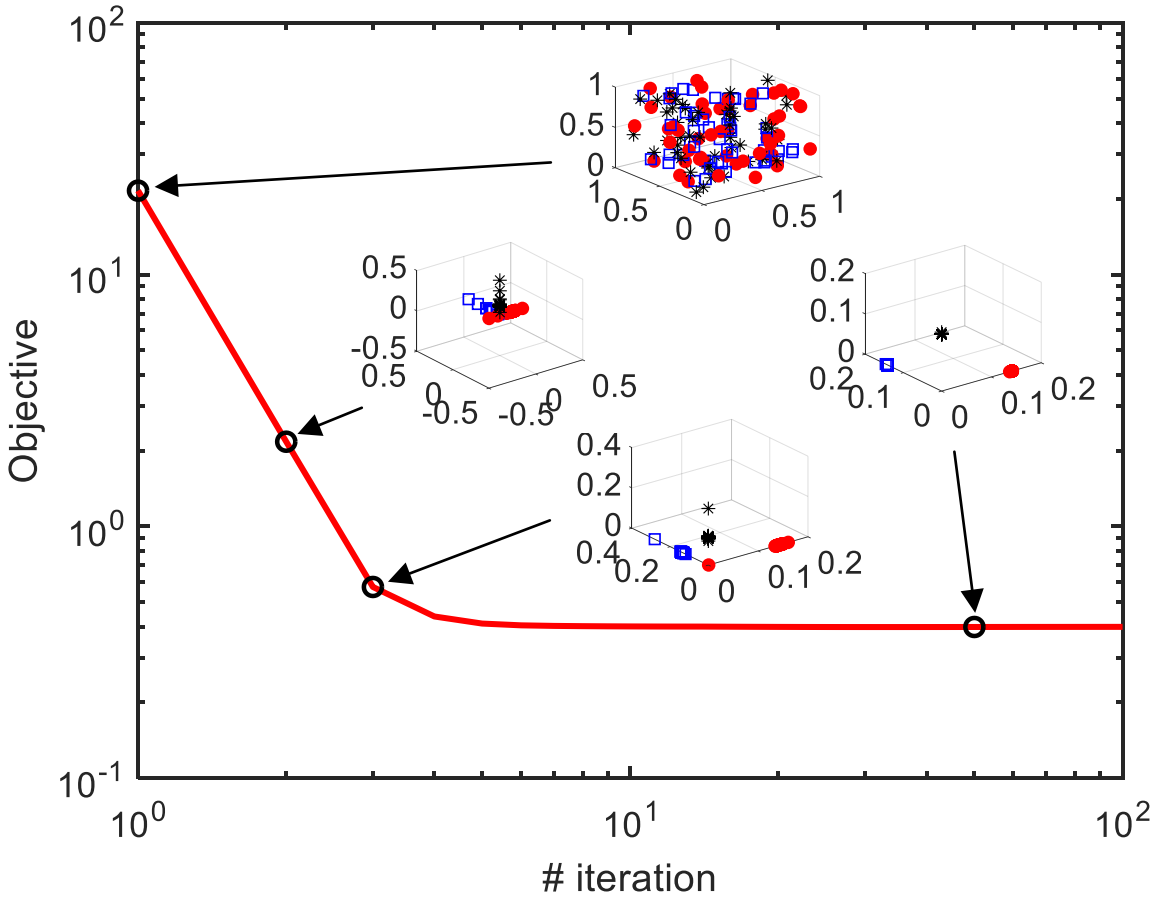


Figure 3.1: The objective function of our method on synthetic data with the result of 3D plots illustrating the clustering structure on checkpoints. The x, y and z axis of 3D plots correspond to the first, second and third column in matrix $\mathbf{X}$, respectively.

Table 3.1: Dataset Discriptions.

Dataset	# Size	# Dimension	# Class
MINIST	5000	784	10
AT&T	400	10304	40
Caltech101	332	900	5
Ionosphere	351	34	2
Wine	178	13	3
Iris	150	4	3
Glass	214	9	6

From Figure 3.1 we can see the following interesting results. First, we can observe that the objective function monotonically decreases in each iteration, which empirically confirms the convergence of the solution algorithm to solve PO-NLE derived by our new smoothed iterative reweighted method. Second, for each checkpoint shown by the black circles on the objective curve, the clustering structure of the experimental data becomes more and more clear in the 3D plots when the algorithm iterates. The three clusters of data points gradually find a solution to separate themselves apart and fall on different axes. Note that, due to the nonnegative constraints on $\mathbf{X}$, data points will finally converge on the positive part of each axis. This observation clearly demonstrate the effectiveness of the proposed new method.

3.2 Studies of the Orthogonality of the Solutions of Our New Method

An important improvement of our new method over the NLE method is that the orthogonality of our solution is rigorously guaranteed, which, as analyzed in [17, 46] is very important to avoid degenerate solutions. Thus, in this subsection we empirically study the orthogonality of the solutions of our new method and compare them against the solutions from the NLE method. Figure 3.2, Figure 3.3, Figure 3.4, Figure 3.5, Figure 3.6 and Figure 3.7 show the heatmap visualizations of $\mathbf{X}^T\mathbf{X}$ learned from our method while Figure 3.8, Figure 3.9, Figure 3.10, Figure 3.11, Figure 3.12 and Figure 3.13 show the heatmap visualizations of $\mathbf{X}^T\mathbf{X}$ learned from NLE method for comparison. The result on AT&T data set can not be shown due to large number of classes. In all 7 data sets, the solution of NLE method are very loosely constrained by $\mathbf{X}^T\mathbf{X} = \mathbf{I}$.

The experiment results show that the learned embeddings from our method are strictly orthogonal on all the experimental data sets, such as shown in Figure 3.2, which will in return lead to better clustering performances and robustness after embedding. In contrast, the NLE method failed to guarantee the orthogonality, as can be seen in Figure 3.8.

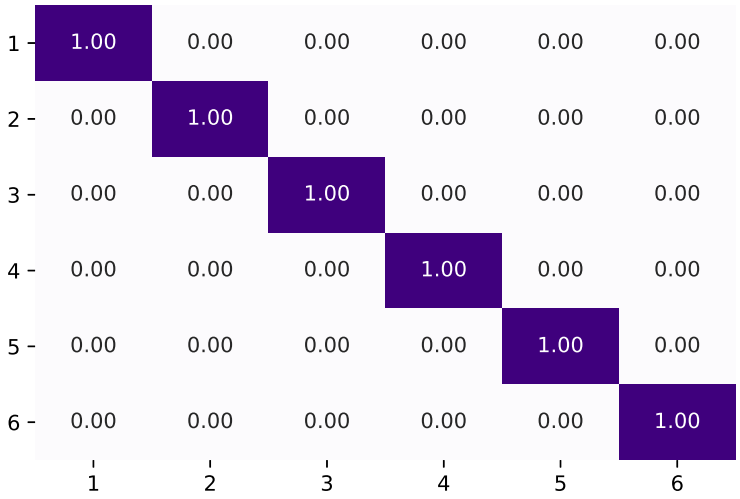


Figure 3.2: Visualization of $X^T X$ learned by our method.

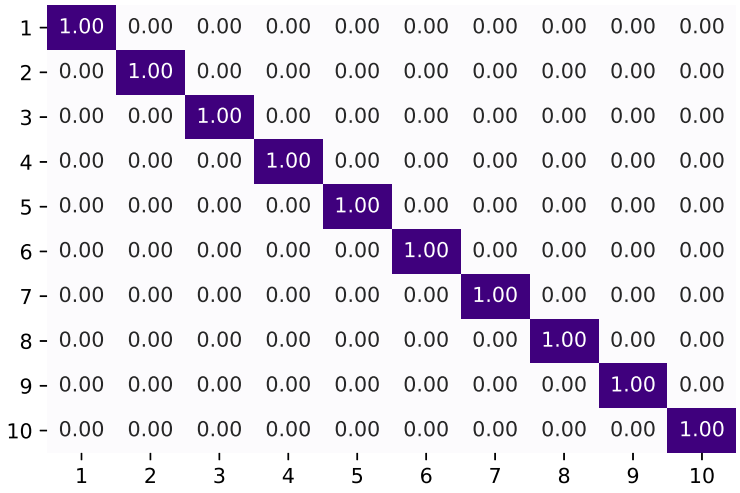


Figure 3.3: Visualization of $X^T X$ learned by our method.

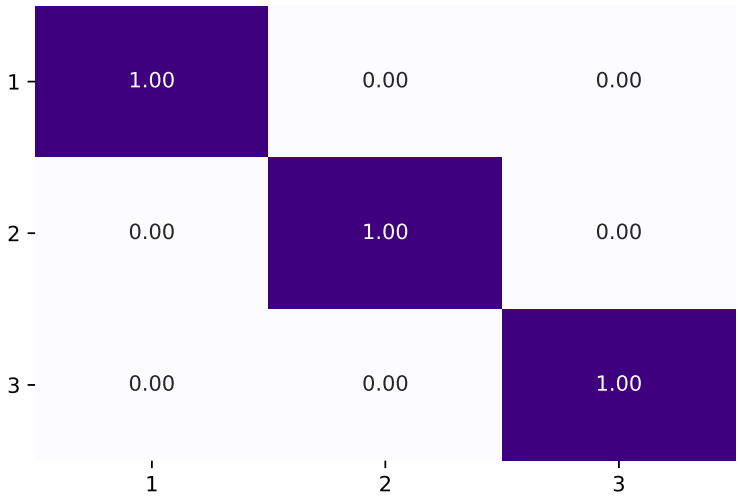


Figure 3.4: Visualization of $X^T X$ learned by our method.

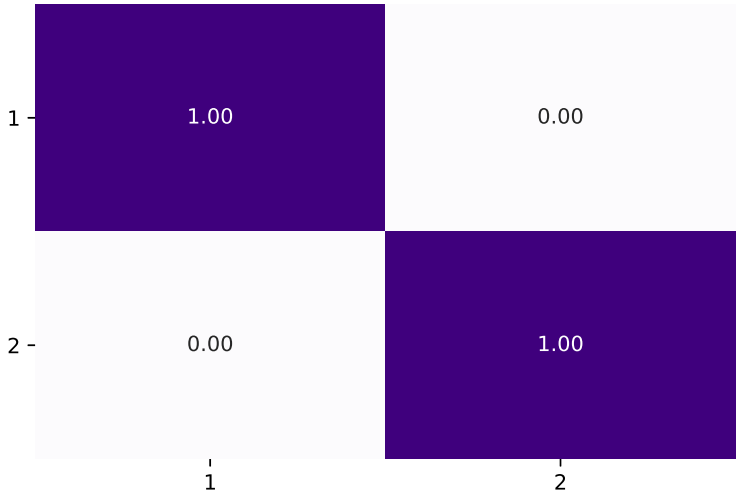


Figure 3.5: Visualization of $X^T X$ learned by our method.

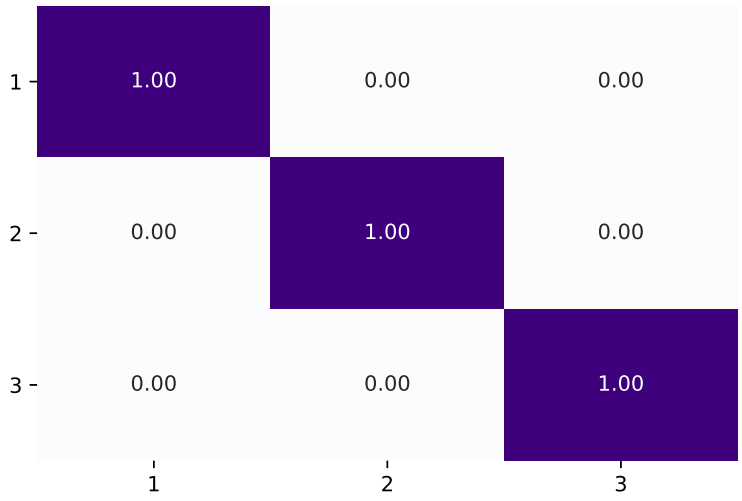


Figure 3.6: Visualization of $X^T X$ learned by our method.

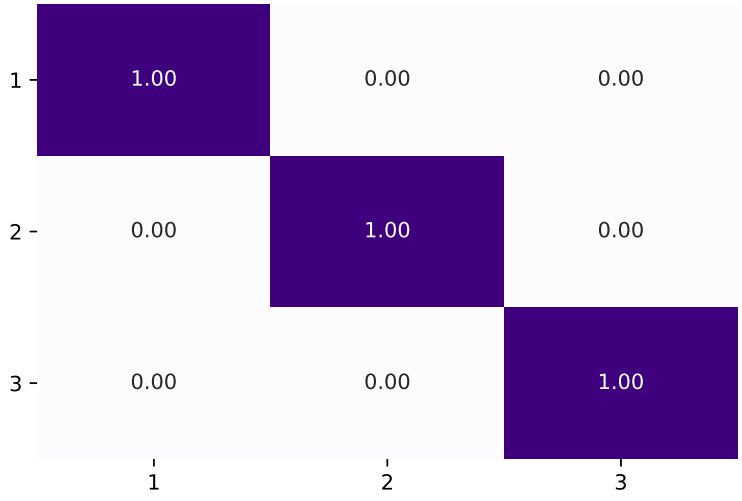


Figure 3.7: Visualization of $X^T X$ learned by our method.

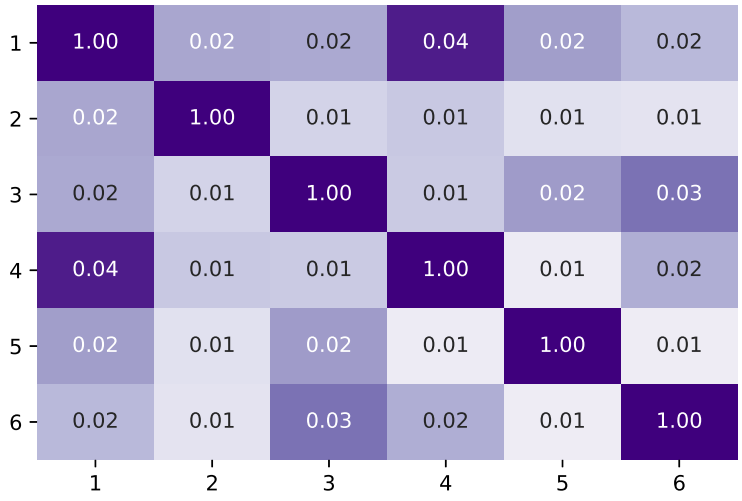


Figure 3.8: Visualization of $X^T X$ learned by NLE method.

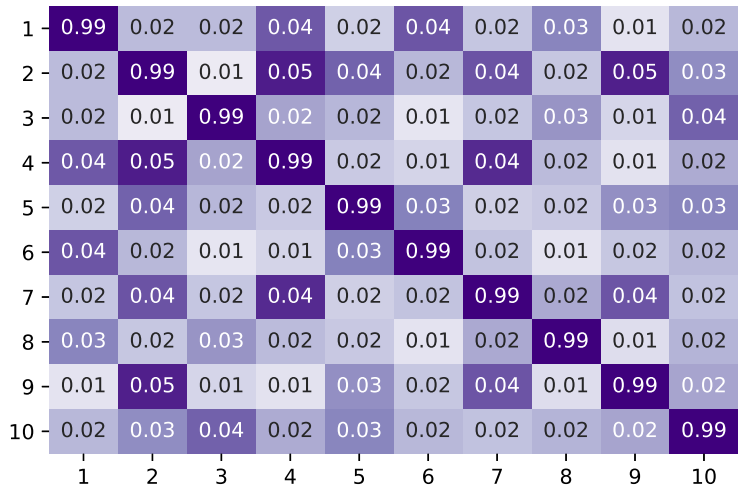


Figure 3.9: Visualization of $X^T X$ learned by NLE method.

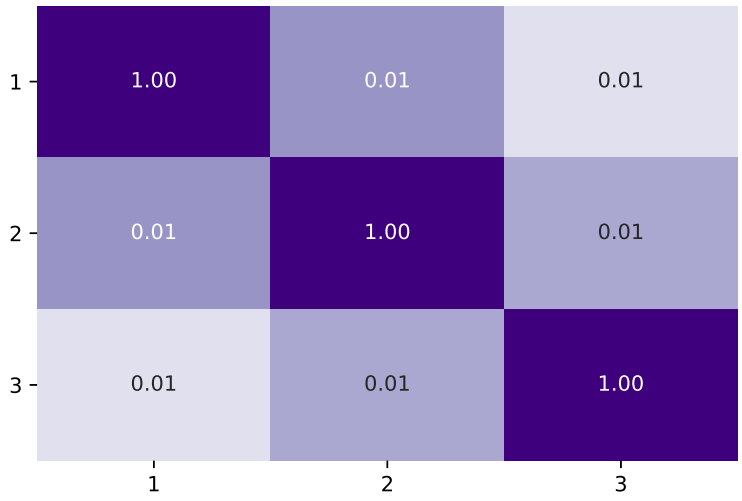


Figure 3.10: Visualization of $X^T X$ learned by NLE method.

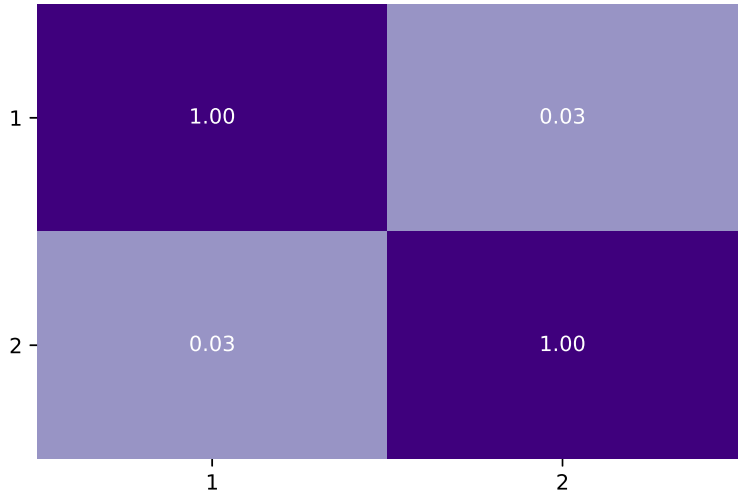


Figure 3.11: Visualization of $X^T X$ learned by NLE method.

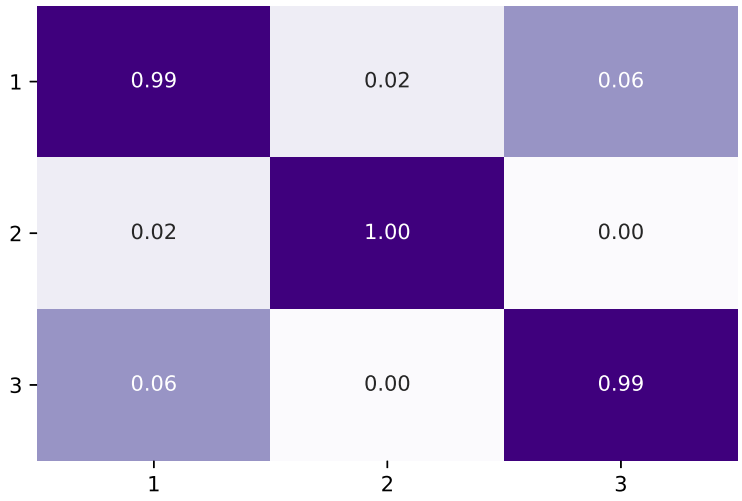


Figure 3.12: Visualization of $X^T X$ learned by NLE method.

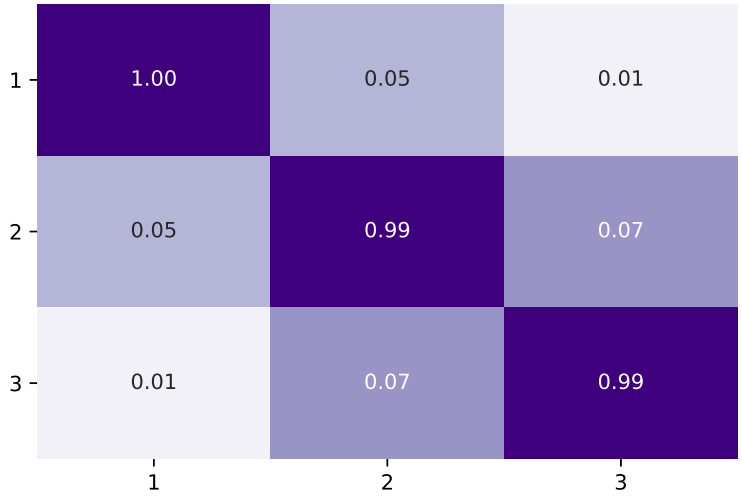


Figure 3.13: Visualization of $X^T X$ learned by NLE method.

Table 3.2: Best and average (Ave) clustering accuracy and purity by our method, NLE, NCut and LE over 200 trials. "↑" means that the bigger number are the better. **Top:** the results on noiseless data (Section 3.3); **bottom:** the results on noisy data (Section 3.4).

Data sets	Clustering accuracy↑								Purity↑							
	Ours		NLE		NCut		LE		Ours		NLE		NCut		LE	
	Best	Ave	Best	Ave	Best	Ave	Best	Ave	Best	Ave	Best	Ave	Best	Ave	Best	Ave
MINIST	**0.7450**	**0.5946**	0.6700	0.5013	0.6540	0.5909	0.6630	0.5874	**0.7880**	**0.6811**	0.7140	0.5553	0.6597	0.4933	0.6438	0.5121
AT&T	**0.8250**	**0.7719**	0.7825	0.6885	0.7525	0.6383	0.7325	0.6882	**0.8675**	**0.8304**	0.8325	0.7476	0.7325	0.6782	0.7250	0.6487
Caltech101	**0.8342**	**0.7719**	0.7131	0.5131	0.5972	0.4965	0.6663	0.5957	**0.9644**	**0.8719**	0.8383	0.5512	0.5943	0.5625	0.6612	0.5768
Ionosphere	**0.8604**	**0.8065**	0.8120	0.6267	0.7236	0.6449	0.7493	0.6475	**0.9658**	**0.8129**	0.8462	0.7017	0.7175	0.6255	0.7413	0.6580
Wine	**0.7303**	**0.7088**	0.7022	0.6464	0.6910	0.6686	0.6685	0.6585	**0.8034**	**0.7092**	0.7753	0.6698	0.7058	0.6497	0.7702	0.6259
Iris	**0.9667**	**0.8945**	0.9600	0.7591	0.9067	0.7824	0.9000	0.7144	**0.9600**	**0.9045**	0.9600	0.7962	0.9067	0.8080	0.9000	0.7340
Glass	**0.5888**	**0.4646**	0.5748	0.4451	0.4439	0.3703	0.5093	0.4102	**0.7710**	**0.6384**	0.5935	0.4832	0.6176	0.5037	0.5335	0.4512
MINIST	**0.5460**	**0.4458**	0.4590	0.3625	0.4210	0.3795	0.4430	0.3928	**0.6520**	**0.5879**	0.5550	0.4369	0.5230	0.4736	0.5070	0.4661
AT&T	**0.7275**	**0.6630**	0.6800	0.5797	0.6575	0.5718	0.6550	0.5606	**0.7800**	**0.7408**	0.7175	0.6529	0.7075	0.6408	0.6975	0.6324
Caltech101	**0.8199**	**0.7681**	0.5839	0.4291	0.5524	0.4177	0.5025	0.4160	**0.8993**	**0.8260**	0.5839	0.4465	0.5573	0.4312	0.5036	0.4285
Ionosphere	**0.7692**	**0.5923**	0.6211	0.5250	0.5755	0.5249	0.5783	0.5270	**0.8889**	**0.7123**	0.6279	0.5305	0.5795	0.5311	0.5848	0.5318
Wine	**0.6292**	**0.5077**	0.5506	0.4318	0.5787	0.4308	0.5730	0.4237	**0.6461**	**0.5537**	0.5506	0.4450	0.5347	0.4400	0.6067	0.4346
Iris	**0.7867**	**0.6679**	0.6733	0.4958	0.6200	0.4689	0.6467	0.4755	**0.8667**	**0.7078**	0.6733	0.5183	0.6800	0.4934	0.6333	0.4953
Glass	**0.5421**	**0.4586**	0.4159	0.3249	0.4299	0.3305	0.3738	0.2914	**0.7383**	**0.6165**	0.4486	0.3565	0.4626	0.3592	0.3832	0.3171

3.3 Experiments on Noiseless Real Data Sets

Now we compare our new method, NLE, NCut and LE on the seven standard data sets as summarized in Table 3.1. Each data set will be tested by different algorithms independently for 200 times. For NCut and LE algorithms, we run K-means clustering with random initialization for 50 times and report the best results.

The performances of the compared methods evaluated by clustering accuracy and clustering purity are reported in the top half of Table 3.2, from which we can see that our method clearly outperforms other competing methods, especially on those comparatively noisier data sets. Due to the nonnegative solutions of our new method, we do not need additional clustering step. Instead, the clustering membership can be read off directly from the learned embeddings. The strictly guaranteed orthogonality constraint avoids degenerate solution and helps improve the performance compared with loosely constrained NLE method which does not have such desired property.

To illustrate the convergence of objective function, Figure 3.14 and Figure 3.15 show a typical run of our algorithm on two UCI benchmark data sets. As the algorithm iterates and the objective value decreases, the accuracy shows a relatively smoothly increasing line.

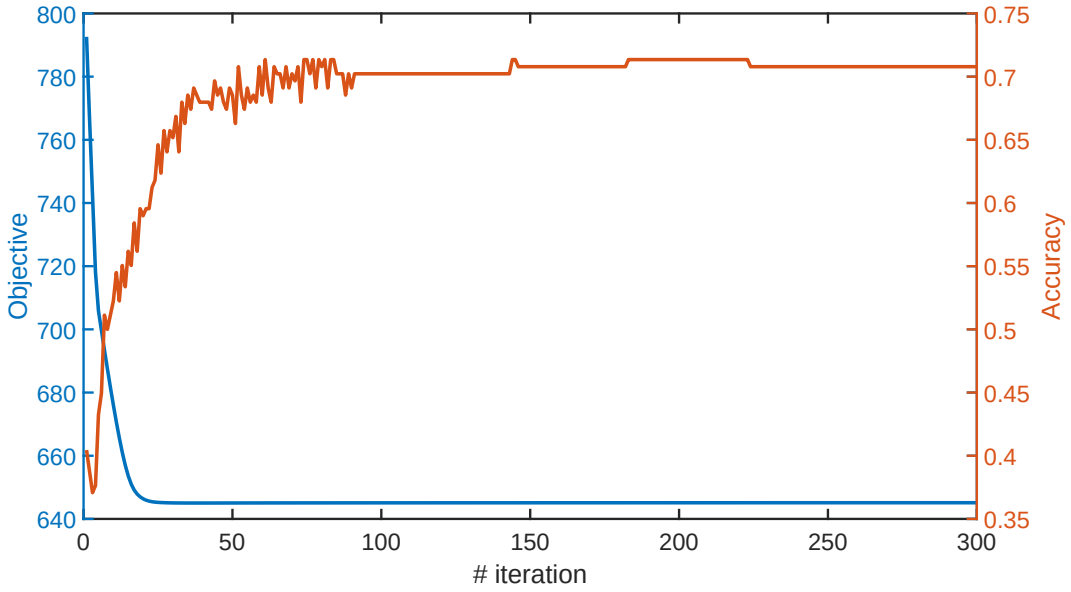


Figure 3.14: A typical run of our algorithm on wine data set with iteration ranging from 1 to 300 to illustrate the convergence of objective function and accuracy of the clustering result.

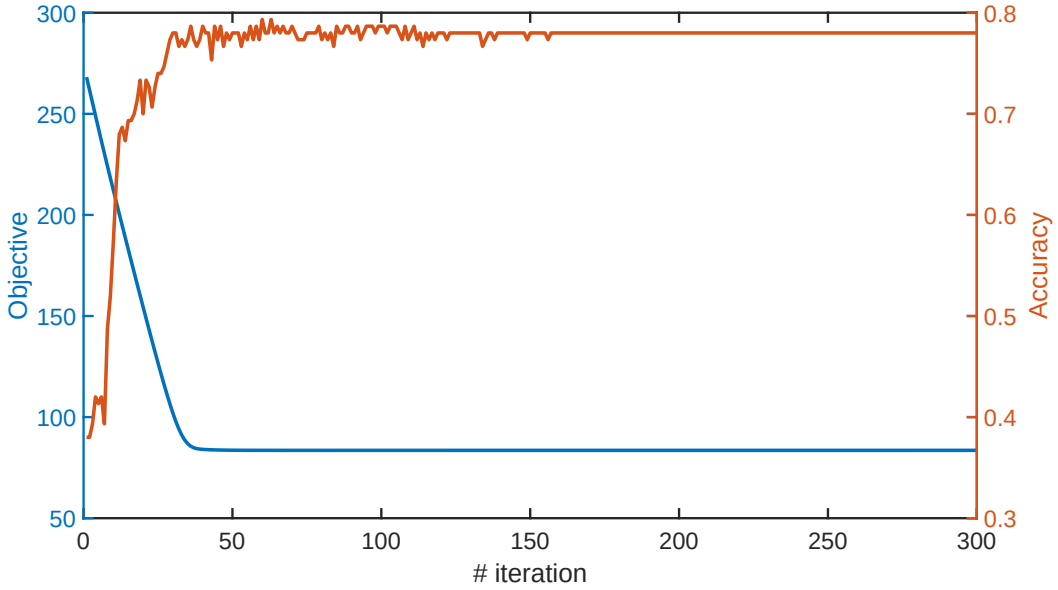
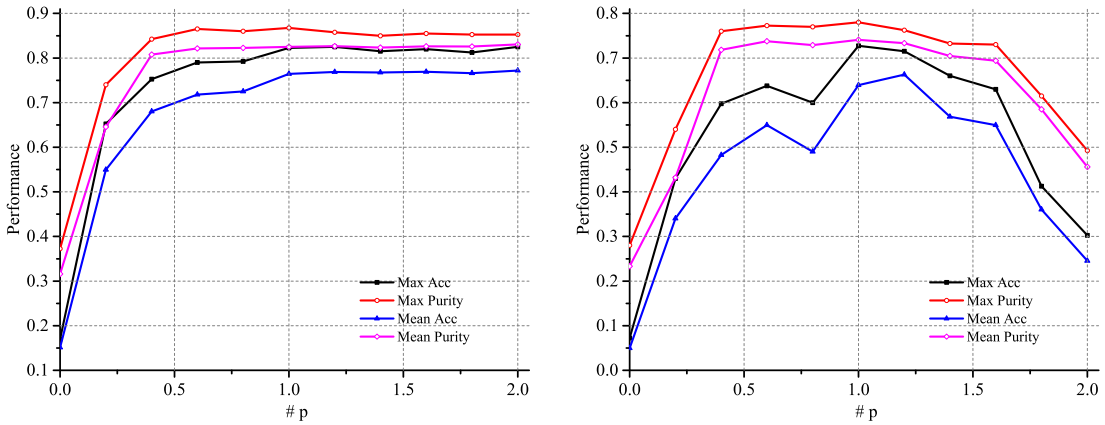


Figure 3.15: A typical run of our algorithm on iris data set with iteration ranging from 1 to 300 to illustrate the convergence of objective function and accuracy of the clustering result.

3.4 Experiments on Noisy Real Data Sets

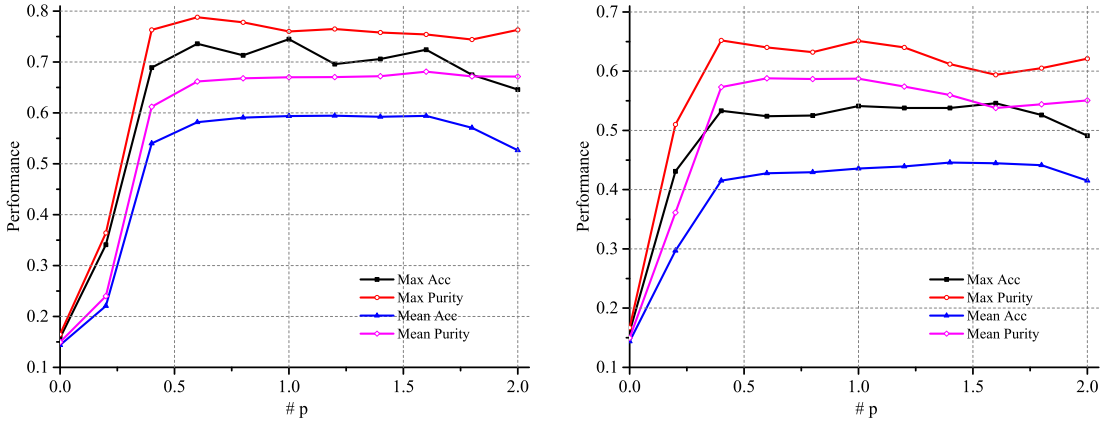
To study the impacts of the value of p in our new embedding model, we randomly contaminate 20% of the data points in all 7 data sets and we run our method with increasing p on those data sets. For each p, we run 200 times for the same contaminated data and original data respectively. The results are shown in Figure 3.16, Figure 3.17, Figure 3.18, Figure 3.19, Figure 3.20, Figure 3.21 and Figure 3.22.



(a) On original data.

(b) On contaminated data.

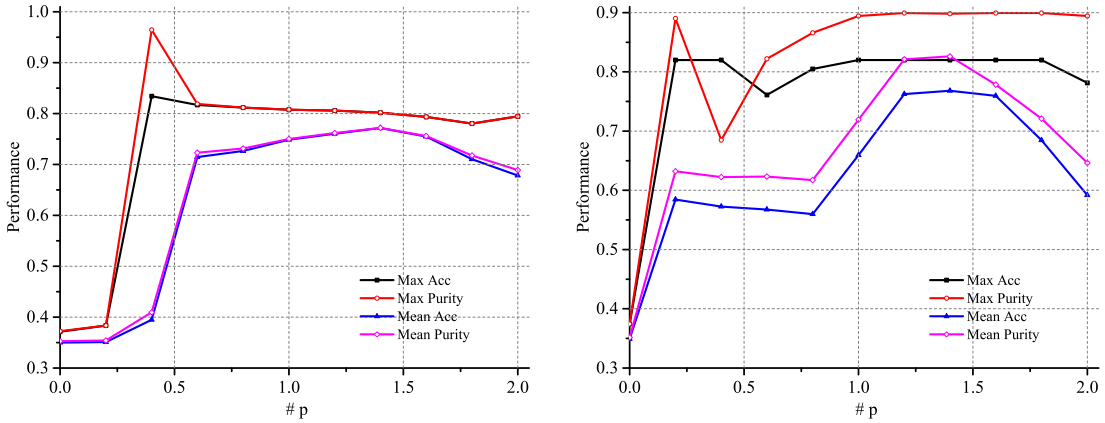
Figure 3.16: The comparison of performance on original and contaminated AT&T data set.



(a) Performance with original data

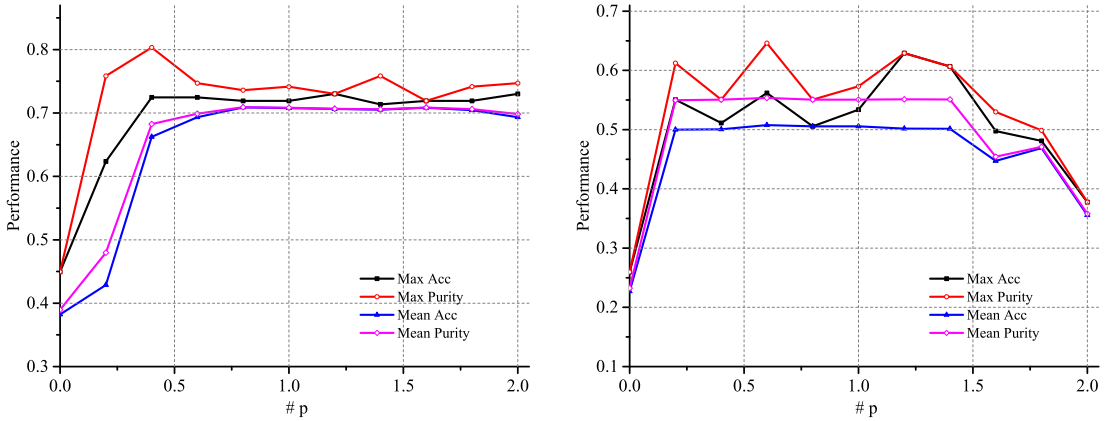
(b) Performance with contaminated data

Figure 3.17: The comparison of performance on original and contaminated minist data set.



(a) Performance with original data (b) Performance with contaminated data

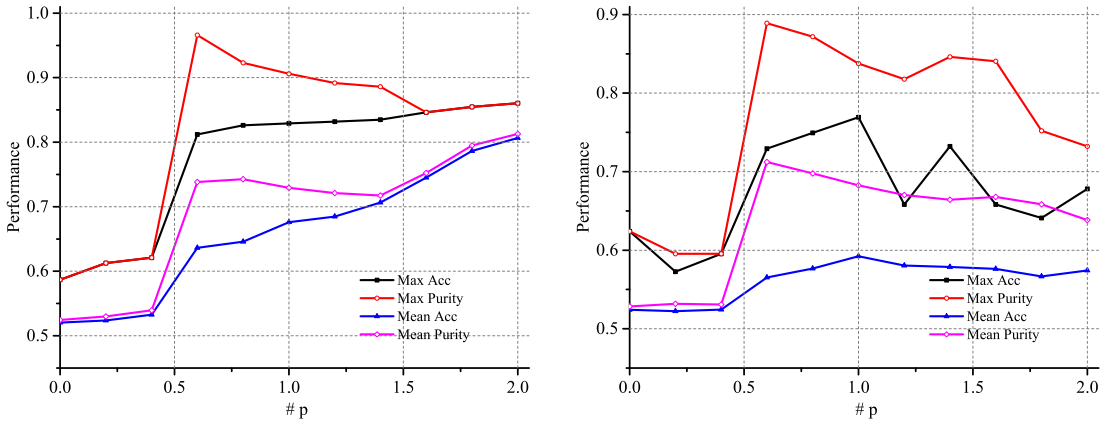
Figure 3.18: The comparison of performance on original and contaminated caltech101 data set.



(a) Performance with original data (b) Performance with contaminated data

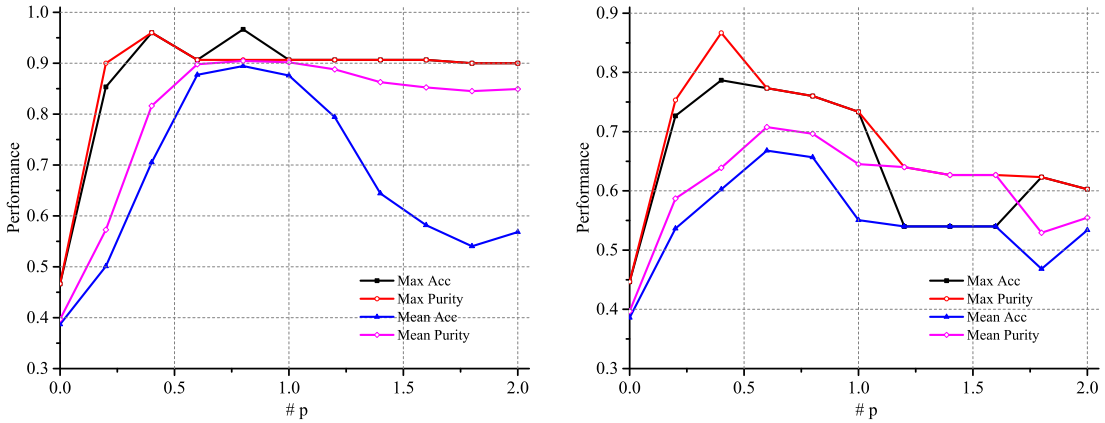
Figure 3.19: The comparison of performance on original and contaminated wine data set.

Interestingly, we can always find the best solution when p is around 1 (sometimes it goes left or right and it depends on the data set). This confirms the correctness of our hypothesis that the the results tends to be worse in both sides when p is too small or too large. In addition, we also observe that the performance of our method is not as good when p is close to 2. This is because when $p = 2$, the quadratic objective function is notoriously known to be very sensitive to the data outliers. Finally, the performance our method drops significantly when p approaches 0. This is because when p is close to 0, there will be no distance any more such that no learning can be



(a) Performance with original data (b) Performance with contaminated data

Figure 3.20: The comparison of performance on original and contaminated ionosphere data set.

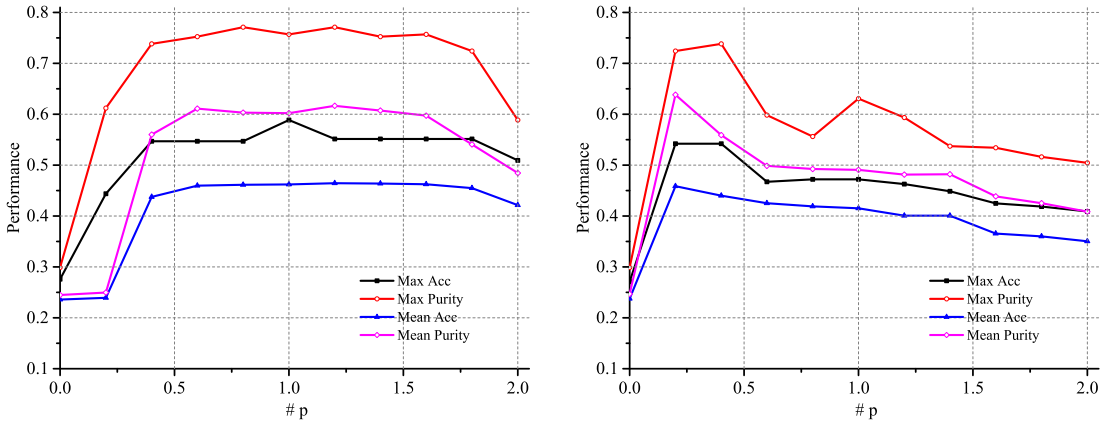


(a) Performance with original data (b) Performance with contaminated data

Figure 3.21: The comparison of performance on original and contaminated iris data set.

performed.

Other algorithms are also tested over 200 times on each data set for comparison. The performances of the clustering methods on contaminated noisy data sets and original data sets respectively are reported in the bottom half of Table 3.2. Among all the best and the average values of *clustering accuracy* and *purity*, our method is consistently better than its counterparts. The results of our approach generally decrease less than other methods on the contaminated data sets, especially for those noisier data sets.



(a) Performance with original data

(b) Performance with contaminated data

Figure 3.22: The comparison of performance on original and contaminated glass data set.

CHAPTER 4

CONCLUSION

Data embedding has been widely used in many machine learning applications due to its efficiency and effectiveness. Laplacian Embedding is a very powerful and unique graph based nonlinear approach that can handle the intrinsic manifold resided in high-dimensional space.

The traditional approach is neither stable nor intuitive and is also sensitive to the data outliers. Although NLE method can achieve non-negative results, the solution is very loosely constrained in terms of the orthogonality. In fact, the orthogonality is very important to avoid degenerate solutions and will also affect the performance. To the best knowledge of us, there is no algorithm that can achieve strict non-negativity and orthogonality in the solution at the same time so far and our method is designed to handle this.

In order to improve the performance and ensure the orthogonality, we proposed a new robust Laplacian embedding approach that uses the p-th order of the ℓ_2-norm distances in the objective and strictly satisfies orthogonality and nonnegativity in the constraints, which results in an objective that is neither convex nor smooth.

We proposed a novel *smoothed iterative reweighted method* to solve the challenging optimization problem, in which a smoothness term is formally and explicitly introduced as an important theoretical contribution of this paper.

Both theoretically and empirically, we proved the convergence of this new algorithm. We have performed extensive experiments, in which the superior performance of our new method has demonstrated its effectiveness and the potential to give a new perspective for nonlinear graph based clustering tasks.

REFERENCES CITED

[1] I. T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. 1986.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

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

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

[5] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM journal on scientific computing*, 26(1):313–338, 2004.

[6] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[7] Blake Shaw and Tony Jebara. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 937–944. ACM, 2009.

[8] Lars Hagen and Andrew B Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11 (9):1074–1085, 1992.

[9] Pak K Chan, Martine DF Schlag, and Jason Y Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on computer-aided design of integrated circuits and systems*, 13(9):1088–1096, 1994.

[10] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[11] Chris HQ Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 107–114. IEEE, 2001.

[12] Kenneth M Hall. An r-dimensional quadratic placement algorithm. *Management science*, 17 (3):219–229, 1970.

[13] Charles J Alpert and Andrew B Kahng. Recent directions in netlist partitioning: a survey. *Integration, the VLSI journal*, 19(1-2):1–81, 1995.

[14] Alex Pothen, Horst D Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM journal on matrix analysis and applications*, 11(3):430–452, 1990.

[15] Dijun Luo, Chris Ding, Heng Huang, and Tao Li. Non-negative laplacian embedding. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 337–346. IEEE, 2009.

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

[17] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[18] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23 (2):298–305, 1973.

[19] William E Donath and Alan J Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[20] C-K Cheng and Y-CA Wei. An improved two-way partitioning algorithm with stable performance (vlsi). *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(12):1502–1511, 1991.

[21] Fan RK Chung. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[22] C. H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 107–114, 2001.

[23] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13 (9):1088–1096, Sep 1994. ISSN 0278-0070. doi: 10.1109/43.310898.

[24] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737, Jun 1997.

[25] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[26] D. Cheng, F. Nie, J. Sun, and Y. Gong. A weight-adaptive laplacian embedding for graph-based clustering. *Neural Computation*, 29(7):1902–1918, July 2017. ISSN 0899-7667. doi: 10.1162/NECO_a_00973.

[27] J. Tang, L. Shao, X. Li, and K. Lu. A local structural descriptor for image matching via normalized graph laplacian embedding. *IEEE Transactions on Cybernetics*, 46(2):410–420, Feb 2016. ISSN 2168-2267. doi: 10.1109/TCYB.2015.2402751.

[28] B. Thirion and O. Faugeras. Nonlinear dimension reduction of fmri data: the laplacian embedding approach. In *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821)*, pages 372–375 Vol. 1, April 2004. doi: 10.1109/ISBI.2004.1398552.

[29] Z. Noorie and F. Afsari. Regularized sparse feature selection with constraints embedded in graph laplacian matrix. In *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, pages 126–130, Dec 2017. doi: 10.1109/ICSPIS.2017.8311602.

[30] K. Stamos, N. A. Laskaris, and A. Vakali. Mani-web: Large-scale web graph embedding via laplacian eigenmap approximation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):879–888, Nov 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2011.2160166.

[31] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.

[32] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[33] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[34] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 1996.

[35] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[36] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

[37] Hua Wang, Feiping Nie, and Heng Huang. Learning robust locality preserving projection via p-order minimization. In *AAAI*, pages 3059–3065, 2015.

[38] Hua Wang, Feiping Nie, and Heng Huang. Robust and discriminative distance for multi-instance learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2919–2924. IEEE, 2012.

[39] Feiping Nie, Hua Wang, Heng Huang, and Chris HQ Ding. Early active learning via robust representation and structured sparsity. In *IJCAI*, pages 1572–1578, 2013.

[40] Feiping Nie, Hua Wang, Cheng Deng, Xinbo Gao, Xuelong Li, Heng Huang, et al. New ℓ_1-norm relaxations and optimizations for graph clustering. In *AAAI*, pages 1962–1968, 2016.

[41] Yun Liu, Yiming Guo, Hua Wang, Feiping Nie, and Heng Huang. Semi-supervised classifications via elastic and robust embedding. In *Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*, 2017.

[42] Hua Wang, Feiping Nie, Weidong Cai, and Heng Huang. Semi-supervised robust dictionary learning via efficient $\ell_{2,0^+}$-norms minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1145–1152, 2013.

[43] Hua Wang, Feiping Nie, and Heng Huang. Robust distance metric learning via simultaneous l1-norm minimization and maximization. In *International Conference on Machine Learning*, pages 1836–1844, 2014.

[44] Axel Ruhe. Perturbation bounds for means of eigenvalues and invariant subspaces. *BIT Numerical Mathematics*, 10(3):343–354, 1970.

[45] Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *International conference on machine learning*, pages 352–360, 2013.

[46] Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005.

[47] Hua Wang, Chris Ding, and Heng Huang. Directed graph learning via high-order co-linkage analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 451–466. Springer, 2010.