

Graph-Laplacian PCA: Closed-form Solution and Robustness

Bo Jiang^a, Chris Ding^{b,a}, Bin Luo^a, Jin Tang^a

^aSchool of Computer Science and Technology, Anhui University, Hefei, 230601, China

^bCSE Department, University of Texas at Arlington, Arlington, TX 76019, USA

zeyiabc@163.com, chqding@uta.edu, luobin@ahu.edu.cn, ahhftang@gmail.com

Abstract

Principal Component Analysis (PCA) is a widely used to learn a low-dimensional representation. In many applications, both vector data X and graph data W are available. Laplacian embedding is widely used for embedding graph data. We propose a graph-Laplacian PCA (gLPCA) to learn a low dimensional representation of X that incorporates graph structures encoded in W . This model has several advantages: (1) It is a data representation model. (2) It has a compact closed-form solution and can be efficiently computed. (3) It is capable to remove corruptions. Extensive experiments on 8 datasets show promising results on image reconstruction and significant improvement on clustering and classification.

1. Introduction

In many computer vision applications, the dimensionality of data are usually very high. An effective approach is dimensionality reduction: in low-dimensional space the class distribution becomes more apparent which significantly improves the machine learning results. There are several dimension reduction methods suitable for different data types. If the input data are vector data with feature vectors, Principal Component Analysis (PCA) and Linear Discriminative Analysis (LDA) are the two most widely used algorithms because of their relative simplicity and effectiveness. These methods generally deal with the case where data mainly lie in a linear data manifold. Other methods include Locally Linear Embedding[12], Local Tangent Space Alignment[15], etc., which can deal with the case with data lying in nonlinear manifolds.

When the input data are graph data in the form of pairwise similarities (or distances) as the graph edge weights, Laplacian Embedding is a classical method[8, 1]. In addition, Manifold learning is also one important class of popular approaches such as Isomap[14], Locality Preserving Projections[9], etc. These methods generally deal with the case where data mainly lie in a nonlinear data manifold.

Of course, given a vector data, we can use kernel to build similarity matrix and thus the graph data. And from a graph data, we can produce low dimensional embedding that results in vector data. Thus the distinction between vector data and graph is sometimes blurred. A recent trend is to use multiple data sources to achieve better results. Another direction is multi-view learning that learn multiple clusterings from multiple sources and then combine them together. In these approaches, the input data types are assumed to be same, either all graph data, or all vector data. However, as mentioned above, one may transform data from one type to another type to match the requirement that data be of same type in these learning approaches.

In this paper, we assume the input data contains vector data X and graph data W . Our task is to learn a low dimensional data representation of X that incorporates the cluster information encoded in graph data W . We propose to use Laplacian embedding coordinates directly into the data representation for X . The resulting graph-Laplacian PCA (gLPCA) model has three aspects: (1) data representation, (2) data embedding, (3) a closed form solution which can be efficiently computed. We run the model on 8 datasets and demonstrate various capabilities.

2. PCA vs. Laplacian Embedding

We start with a brief introduction of principal component analysis and Laplacian embedding.

2.1. Principal Component Analysis

Let the input data matrix $X = (x_1, \dots, x_n) \in R^{p \times n}$ contains the collection of n data column vectors in p dimension space. In image processing, each column x_i is linearized array of pixels gray levels. PCA [10, 3] finds the optimal low-dimensional (k -dim) subspace defined by the principal directions $U = (u_1, \dots, u_k) \in R^{p \times k}$. The projected data points in the new subspace are $V^T = (v_1, \dots, v_n) \in R^{k \times n}$. PCA finds U and V by minimizing

$$\min_{U, V} \|X - UV^T\|_F^2 \text{ s.t. } V^T V = I \quad (1)$$

In addition, PCA relates closely to K-means clustering naturally [4]. The principal components V are actually the continuous solution of the cluster membership indicators in the K-means clustering method. This provides a motivation to relate PCA to Laplacian embedding whose primary purpose is clustering.

2.2. Manifold Embedding using Graph Laplacian

PCA provides an embedding for the data lying on a linear manifold. However, in many applications, data lie in a non-linear manifold. One popular method is to use graph Laplacian based embedding. Given the pairwise similarity data matrix $W \in R^{n \times n}$ containing the edge weights on a graph with n nodes, Laplacian embedding [8, 1] preserves the local geometrical relationships and maximizes the smoothness with respect to the intrinsic manifold of the data set in the low embedding space. Let $Q^T = (q_1, q_2, \dots, q_n) \in R^{k \times n}$ be the embedding coordinates of the n data points. They are obtained by minimizing

$$\min_Q \sum_{i,j=1}^n \|q_i - q_j\|^2 W_{ij} = \text{tr}(Q^T(D - W)Q) \quad (2)$$

s.t. $Q^T Q = I$

where $D = \text{diag}(d_1, \dots, d_n)$ and $d_i = \sum_j W_{ij}$. The solution eigenvectors of $D - W$.

Laplacian embedding is closely connected with graph clustering. In fact, the embedding vectors (q_1, q_2, \dots, q_n) of Eq.(2) provide an approximation solution for the Ratio Cut spectral clustering[2], i.e., they can be seen as the relaxation solution of the cluster indicators (q_i for data i) in the spectral clustering objective function. This is similar to PCA being the spectral relaxation of K-means clustering [4].

3. Graph-Laplacian PCA

Suppose we are given vector data X and pairwise similarity W . We wish to learn a low dimensional data representation of X that incorporates data cluster structures inherent in W , i.e., a representation regularized by the data manifold encoded in W . Because v_i in PCA is exactly the same role of q_i in Laplacian embedding, we set them equal and thus propose to combine Eqs.(1,2) into the following model

$$\min_{U,Q} J = \|X - UQ^T\|_F^2 + \alpha \text{Tr}(Q^T(D - W)Q) \quad (3)$$

s.t. $Q^T Q = I$

where $\alpha \geq 0$ is a parameter balance the contribution from the two parts.

This model has several aspects. (1) It is a data representation, i.e., $X \simeq UQ^T$. (2) It is a manifold embedding using Q . (3) Although the model is not a convex problem, it has a closed-form solution and can be efficiently computed. We call this model graph-Laplacian PCA (gLPCA).

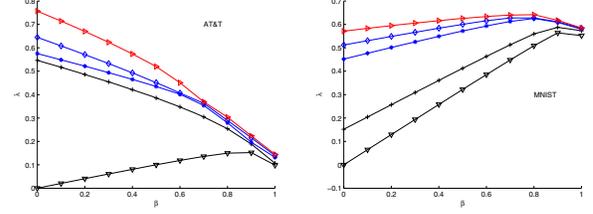


Figure 1. Lowest 5 eigenvalues of G_β for the AT&T and MNIST datasets.

3.1. Closed-form Solution

gLPCA has the following compact closed-form solution:

Theorem 3.1 *The optimal solution (U^*, Q^*) of gLPCA are given by*

$$Q^* = (v_1, v_2, \dots, v_k) \quad (4)$$

$$U^* = XQ^*, \quad (5)$$

where v_1, v_2, \dots, v_k are the eigenvectors corresponding to the first k smallest eigenvalues of the matrix

$$G_\alpha = -X^T X + \alpha L, \quad L \equiv D - W. \quad (6)$$

Note that data X is centered as in standard PCA.

Proof.

1. Solving for the optimal U^* while fixing Q . This is given by

$$\frac{\partial J}{\partial U} = -2XQ + 2U = 0. \quad (7)$$

Thus $U^* = XQ$.

2. Solve the optimal Q^* . Set $U = XQ$, Eq.(3) becomes

$$\min_Q J_2(Q) = \|X - XQQ^T\|_F^2 + \alpha \text{tr} Q^T L Q \quad (8)$$

s.t. $Q^T Q = I$

By some algebra, Eq.(8) is equivalent to the following

$$\min_Q \text{tr} Q^T (-X^T X + \alpha L) Q \quad (9)$$

s.t. $Q^T Q = I$

Thus, the optimal Q^* can be obtained by computing eigenvectors corresponding to the first k smallest eigenvalues of the matrix G_α . \square

3.2. Properties of gLPCA

We use λ_n , the largest eigenvalue of kernel matrix $X^T X$ to normalize $X^T X$. Similar, we use ξ_n , the largest eigenvalue of Laplacian matrix L to normalize L . In this way, the two terms in G of Eq.(6) (used in Eq.(9)) have similar scale. We set

$$\alpha = \frac{\beta}{1 - \beta} \frac{\lambda_n}{\xi_n}, \quad (10)$$

where β is the alternative model parameter instead of α . Substituting α into Eq.(9), it is equivalent to

$$\begin{aligned} \min_Q \quad & \text{Tr } Q^T \left[(1 - \beta) \left(I - \frac{X^T X}{\lambda_n} \right) + \beta \frac{L}{\xi_n} \right] Q \quad (11) \\ \text{s.t.} \quad & Q^T Q = I \end{aligned}$$

Therefore, the solution of Q are given by the eigenvectors of G_β :

$$G_\beta = (1 - \beta) \left(I - \frac{X^T X}{\lambda_n} \right) + \beta \frac{L}{\xi_n} \quad (12)$$

Clearly, parameter β should be in the range $0 \leq \beta \leq 1$. **PCA limit:** When $\beta = 0$, gLPCA reduces to standard PCA. **LE limit:** When $\beta = 1$, gLPCA reduces to Laplacian embedding. In this case, however, the data representation gLPCA is still valid because $U = XQ$ is well-defined and $x_i \simeq Uq_i$.

Proposition 3.2 *The matrix G_β has the following properties: (1) G_β is semi-positive definite; (2) $\mathbf{e} = (1 \cdots 1)^T$ is an eigenvector of G_β ;*

$$G_\beta \mathbf{e} = (1 - \beta) \mathbf{e}. \quad (13)$$

(3) Any other eigenvector v is orthogonal to \mathbf{e} .

Proof. First, because λ_n is the largest eigenvalue of $X^T X$, thus $I - X^T X / \lambda_n$ is s.p.d. It is well-known that L is s.p.d. Thus G_β is s.p.d. Second, because X is centered, we have $X\mathbf{e} = 0$. Thus \mathbf{e} is an eigenvector of $X^T X$ with zero eigenvalue. Also, it is well-known that \mathbf{e} is an eigenvector of L with zero eigenvalue. This proves (2). Third, for any symmetric real matrix G_β , its non-degenerate eigenvectors are mutually orthogonal. Since \mathbf{e} is an eigenvector, thus all other eigenvectors are orthogonal to \mathbf{e} . \square

By Proposition 3.2, we write the matrix G_β as

$$G_\beta = (1 - \beta) \left(I - \frac{X^T X}{\lambda_n} \right) + \beta \left(\frac{L}{\xi_n} + \frac{\mathbf{e}\mathbf{e}^T}{n} \right) \quad (14)$$

Here, we added $\mathbf{e}\mathbf{e}^T/n$ in the Laplacian matrix part. By Proposition 3.2, this does not change any eigenvectors and eigenvalues, except it shifts the trivial eigenvector \mathbf{e} up to have eigenvalue 1, the largest eigenvalue of G_β . This is useful because it guarantees that the computed lowest k eigenvectors does not contain \mathbf{e} . Two examples of eigenvalue distributions are shown in Fig. 1.

3.3. Illustrative examples

Image reconstruction. To illustrate the data representation aspect of gLPCA, we run gLPCA on the original and occluded images from AT&T face dataset (more details given in the Experiments section). Figure 2, 3 show the



Figure 2. Reconstructed images from PCA and gLPCA. In each panel, 10 images of one person are: 1st line: original images; 2nd line: PCA; 3rd - 5th lines: gLPCA at $\beta = 0.3, 0.5, 1$ respectively.

original image, reconstructed images of standard PCA and gLPCA. In each panel, the top row contains 10 original images of one person, 2nd - 5th rows contain reconstructed images (columns of UQ^T) at $\beta = 0, 0.3, 0.5, 1$, respectively. $\beta=0$ results in standard PCA. Here we did two experiments, one using the original images, another using occluded images. Only images of 2 persons (out of total 40 persons) in each experiment are shown due to space limitation.

Here we can observe that (1) Large errors (occlusions) are suppressed in gLPCA, but retained in standard PCA. (2) gLPCA reconstructed images of same person tend to be more similar than those in standard PCA. These features indicate that the class structures are generally more apparent in gLPCA representation, and motivate us to use it for data clustering and semi-supervised learning tasks.

Data approximation. In Figure 4, we show the percentage residual $\|X - UQ^T\|/\|X\|$ on five datasets: AT&T, MNIST, USPS, BinAlpha, and COIL20 datasets (details see §5). In general, residual errors remain close to PCA at small β , and go slightly higher at the Laplacian embedding (LE) limit $\beta = 1$. This is a bit surprising because in this LE limit, Q is entirely determined by Laplacian embedding and does not involve the data X (although the graph similarity W captures some relationship of the data). To understand this



Figure 3. PCA and gLPCA reconstruction on *partially occluded* AT&T face dataset. Details are same as Fig.2.

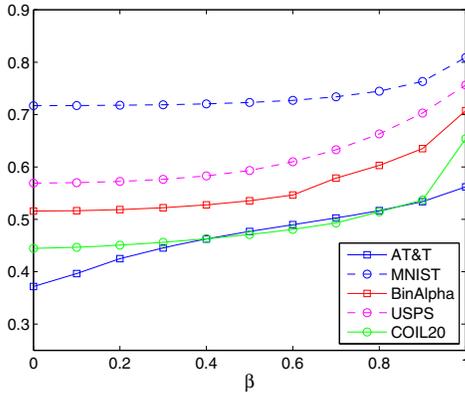


Figure 4. Residual error $\|X - UQ^T\|/\|X\|$ at different β .

better, we compute a baseline by replacing $D - W$ in G_β by a random symmetric matrix. The obtained Q satisfies $Q^T Q = I$, but it contains no relevant information. This baseline random- Q results are indeed much poorer compared to gLPCA at LE limit: the residuals are shown in Table 1.

Demonstration of embedding results. We demonstrate the embedding aspect of gLPCA using 2D visualization. We first randomly select 30 images of three persons (out

Table 1. Residual errors for gLPCA at Laplacian Embedding limit.

Dataset	AT&T	MNIST	BinAlpha	USPS	COIL20
gLPCA	0.5620	0.8091	0.7565	0.7075	0.6539
Random	0.9479	0.9897	0.9832	0.9898	0.9854

of 40 persons) from AT&T face dataset as the input, and then run PCA, Laplacian embedding and gLPCA, respectively (more details are given in the Experiments Section). The embedding results are shown in Figure 5. Clearly, gLPCA results are in-between PCA and LE results. However, in gLPCA embedding, class structures are more apparent than in PCA or LE.

4. Robust gLPCA

We can formulate a robust version of the above gLPCA using $L_{2,1}$ norm as following

$$\begin{aligned} \min_{U, Q} \|X - UQ^T\|_{2,1} + \alpha \text{Tr} Q^T (D - W) Q \quad (15) \\ \text{s.t. } Q^T Q = I \end{aligned}$$

where the matrix $L_{2,1}$ norm is first introduced in [6] and is defined as $\|A\|_{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^p A_{ij}^2}$. We call Eq.(15) as Robust graph Laplacian PCA (RgLPCA).

In the following, we present an efficient updating algorithm to solve this model. Firstly, Eq.(15) can be rewritten equivalently as

$$\begin{aligned} \min_{U, Q, E} \|E\|_{2,1} + \alpha \text{Tr} Q^T (D - W) Q \quad (16) \\ \text{s.t. } E = X - UQ^T, Q^T Q = I \end{aligned}$$

We use the Augmented Lagrange Multiplier (ALM) method to solve this problem. ALM solves a sequences of sub-problems

$$\begin{aligned} \min_{U, Q, E} \|E\|_{2,1} + \text{Tr} C^T (E - X + UQ^T) \\ + \frac{\mu}{2} \|E - X + UQ^T\|_F^2 + \alpha \text{Tr} Q^T L Q \quad (17) \\ \text{s.t. } Q^T Q = I \end{aligned}$$

where C is Lagrange multipliers and μ is the penalty parameter. There are two major parts of this algorithm: solving the sub-problem and updating parameters (C, μ) .

Solving the sub-problems

The objective of Eq.(17) can be written as

$$\min_{U, Q, E} \|E\|_{2,1} + \frac{\mu}{2} \|E - X + UQ^T\|_F^2 + \alpha \text{Tr} (Q^T L Q)$$

First, we solve U, Q while fixing E . This is

$$\min_{U, Q^T Q = I} \frac{\mu}{2} \|E - X + UQ^T\|_F^2 + \alpha \text{tr} (Q^T L Q) \quad (18)$$

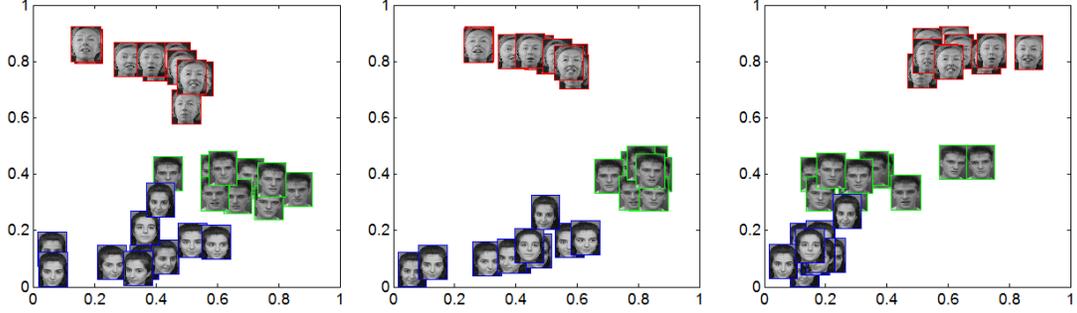


Figure 5. 2D Embedding results on three faces on AT&T dataset using PCA (left), gLPCA (middle) and Laplacian embedding (right)

which is the form of gLPCA and is solved by Theorem 3.1. Second, we solve E while fixing U, Q . This is

$$\min_E \|E\|_{2,1} + \frac{\mu}{2} \|E - A\|_F^2 \quad (19)$$

where $A = X - UQ^T - C/\mu$. Let e_i be the i -th column of matrix E . Similarly for a_i . Eq.(19) is decomposed into n independent problems

$$\min_{e_i} \|e_i\| + \frac{\mu}{2} \|e_i - a_i\|^2 \quad (20)$$

The solution of this proximal operator is known [11] to be

$$e_i = \max\left(1 - \frac{1}{\mu \|a_i\|}, 0\right) a_i. \quad (21)$$

Updating parameters

At the end of each ALM iteration, C, μ are updated as

$$C = C + \mu(E - X + UQ^T) \quad (22)$$

$$\mu = \rho\mu \quad (23)$$

where $\rho > 1$. In practice, $\rho = 1.1 - 1.5$ are good choices.

5. Experiments

To validate the effectiveness of our models, we run gLPCA and RgLPCA on eight datasets, including AT&T, Bin-alpha¹, MNIST, USPS, COIL20² and three UCI datasets³ (ISOLET1, Libras Movement (LMove) and Multiple Feature Data Set (MFeat)). Table 2 summarizes the characteristics of them. We do both **clustering** and **semi-supervised learning** on these datasets.

5.1. Results for gLPCA

The primary goal of the evaluation is to compare gLPCA results with standard PCA and Laplacian Embedding (LE)

¹<http://olivier.chapelle.cc/ssl-book/benchmarks.html>

²<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multi-class.html>

³<http://archive.ics.uci.edu/ml/datasets.html>

Table 2. Dataset descriptions.

Dataset	# Size	# Dimension	# Class
AT&T	400	644	40
MNIST	500	784	10
Bin-alpha	1014	320	36
USPS	1000	256	10
COIL 20	1440	784	72
ISOLET1	1560	617	26
LMove	360	91	15
MFeat	2000	216	10

results because gLPCA is a combination of the two models. To be complete, we also compare to some other embedding methods including the Locality Preserving Projections (LPP)[9] and Normalized Cut (Ncut)[13, 7]⁴ and original data.

First we perform clustering task on PCA, LE, LPP, Ncut and original data representations to evaluate them. We use K-means and semi-NMF clustering [5] algorithms for this evaluation. We use accuracy (ACC), normalized mutual information (NMI) and purity (PUR) as the measurements of the clustering qualities. We run K-means with random initialization 50 times and use the average clustering results. To see how gLPCA model performs at different regularization parameter β , we show in Figure 6 the K-means clustering accuracy on five datasets: AT&T, MNIST, USPS, BinAlpha, and COIL20 datasets. Here we can see that (1) gLPCA results are fairly stable in the entire β range. (2) gLPCA results are generally in-between PCA (at $\beta = 0$) results and Laplacian embedding (at $\beta = 1$) results. (3) There are always some range (mostly $\beta = 0.4 \sim 0.9$) of β where gLPCA is better than PCA and LE.

The complete clustering results are shown in Table 3. From Table 3, we observe that (1) both PCA and LE usually provide better clustering results than original data does, demonstrating the usefulness of PCA and LE. (2) gLPCA consistently performs better than classic PCA, Laplacian embedding and other methods. This is consistent with the

⁴Ncut uses eigenvectors Q of $(D - W)$ with $Q^T D Q = I$ orthonormalization.

Table 3. Clustering Results using K-means and semi-NMF on original data, standard PCA, LE, LPP, Ncut and gLPCA.

Dataset	Metric	K-means						semi-NMF					
		Original	PCA	LE	Ncut	LPP	gLPCA	Original	PCA	LE	Ncut	LPP	gLPCA
AT & T	ACC	0.6701	0.6755	0.7193	0.6856	0.7160	0.7236	0.6145	0.6721	0.7355	0.7136	0.7013	0.7521
	NMI	0.8426	0.8477	0.8592	0.8482	0.8594	0.8665	0.7723	0.8386	0.8628	0.8568	0.8573	0.8635
	PUR	0.6935	0.7091	0.7568	0.7365	0.7550	0.7692	0.6415	0.7188	0.7633	0.7491	0.7457	0.7658
MNIST	ACC	0.5495	0.5624	0.5849	0.5760	0.5757	0.6028	0.4949	0.5562	0.5918	0.5742	0.5626	0.6124
	NMI	0.5137	0.5077	0.6114	0.6163	0.5584	0.6254	0.4022	0.4867	0.6151	0.6185	0.5450	0.6412
	PUR	0.5842	0.5895	0.6310	0.6339	0.6227	0.6573	0.5036	0.5777	0.6426	0.6372	0.6159	0.6779
BinAlf	ACC	0.5982	0.6032	0.6108	0.6004	0.6167	0.6284	0.4498	0.5836	0.6115	0.6147	0.6134	0.6321
	NMI	0.7196	0.6641	0.7401	0.7353	0.7338	0.7448	0.5365	0.6978	0.7390	0.7438	0.7145	0.7445
	PUR	0.6269	0.6351	0.6524	0.6400	0.6528	0.6650	0.4607	0.6177	0.6514	0.6474	0.6432	0.6666
USPS	ACC	0.6138	0.6215	0.6667	0.6153	0.6371	0.6868	0.6051	0.5911	0.6854	0.6491	0.6509	0.6958
	NMI	0.5517	0.5800	0.6208	0.6282	0.5779	0.6294	0.4910	0.4840	0.6344	0.6501	0.5900	0.6420
	PUR	0.6211	0.6057	0.6407	0.6561	0.6567	0.7054	0.5942	0.5874	0.7151	0.6769	0.6728	0.7161
COIL20	ACC	0.5695	0.5907	0.5804	0.5739	0.5823	0.6315	0.4771	0.5976	0.6788	0.6457	0.6143	0.6542
	NMI	0.7238	0.7336	0.7633	0.7518	0.7492	0.7789	0.5939	0.7294	0.7951	0.7757	0.7365	0.7852
	PUR	0.6046	0.6361	0.6559	0.6540	0.6539	0.6647	0.4844	0.6272	0.6781	0.6750	0.6463	0.6819
ISOLET1	ACC	0.6006	0.5860	0.5545	0.5981	0.6318	0.6431	0.5744	0.5653	0.5513	0.5558	0.6165	0.6472
	NMI	0.7551	0.7248	0.7218	0.7306	0.7598	0.7669	0.6715	0.6602	0.7329	0.7224	0.7239	0.7618
	PUR	0.6474	0.6295	0.5707	0.6330	0.6797	0.6638	0.6002	0.5895	0.5782	0.5804	0.6446	0.6686
LMove	ACC	0.4414	0.4454	0.4461	0.4521	0.4524	0.4687	0.4378	0.4372	0.4709	0.4637	0.4719	0.4861
	NMI	0.5746	0.5830	0.5964	0.5961	0.6082	0.6178	0.5456	0.5448	0.6056	0.6098	0.6046	0.6154
	PUR	0.4723	0.4797	0.4948	0.4963	0.4921	0.5052	0.4630	0.4621	0.4979	0.4956	0.5092	0.5131
MFeat	ACC	0.5928	0.5803	0.6326	0.6402	0.6341	0.6477	0.5927	0.5912	0.6458	0.6628	0.6439	0.7223
	NMI	0.6009	0.5909	0.6987	0.7006	0.6965	0.7043	0.6010	0.5980	0.7042	0.7103	0.6969	0.7346
	PUR	0.6323	0.6183	0.6819	0.6907	0.6936	0.6899	0.6303	0.6289	0.6867	0.7030	0.6994	0.7465

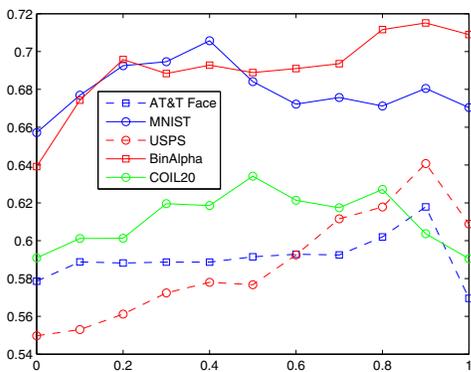


Figure 6. Clustering accuracy using gLPCA representation.

observations in §3.3 that class structures are more apparent in gLPCA representation.

Second, we perform classification on PCA, LE and original data representations using regression and KNN classification methods. We randomly select 20%, 30% of labeled data for each class, and use the rest as unlabeled data. We repeat this 5 times and the average classification accuracy are summarized in Table 4. Here, we observe that, in all cases, we obtain higher classification accuracy in gLPCA representation.

5.2. Results for Robust gLPCA (RgLPCA)

We run the proposed RgLPCA model on several image datasets, including AT&T, Bin-alpha, MNIST, USPS and COIL 20. For each class, we corrupt 20% images manually. The size of corruption is 25% of the image. We

perform clustering and classification tasks on these datasets and compare with several other models, including the original data, standard PCA and L21PCA[6]. We use K-means and regression algorithms for clustering and classification, respectively. For clustering, the accuracy (ACC), normalized mutual information (NMI) and purity (PUR) are used as the measurements of the clustering qualities. We run K-means with random initialization 50 times and use the average clustering results. The results are shown in Table 5. For classification, we randomly select 20%, 30% of labeled data for each class, and use the rest as unlabeled data. We repeat 5 times and the average classification accuracy are summarized in Table 6. From table 5, 6 we observe that (1) PCA performs poorly on these occluded data, indicating that PCA is sensitive to outlying observations. (2) L21PCA can generally return better performance than the original data and standard PCA on all the datasets. This suggests that L21PCA are robust to the gross noise. (3) RgLPCA generally performs better than other data representation models.

6. Summary

Graph-Laplacian PCA incorporates PCA and Laplacian embedding (LE) simultaneously to provide a low-dimensional representation that incorporates graph structures. It has a closed-form solution and can be efficiently computed. The robustness of gLPCA against data corruptions as shown in Figures 2 and 3 are probably the most significant new findings of this work. The robustness originates from that Q contains the cluster structure (manifold) information due to LE. The minor residual increase at the LE limit ($\beta = 1$) in Fig.4 shows that the embedding Q ob-

Table 4. Accuracy results of semi-supervised learning using Regression and KNN on original data, PCA, LE, LPP, Ncut and gLPCA.

Dataset	Labeling	Regression						kNN(k = 1)						kNN(k = 5)					
		Original	PCA	LE	Ncut	LPP	gLPCA	Original	PCA	LE	Ncut	LPP	gLPCA	Original	PCA	LE	Ncut	LPP	gLPCA
AT & T	20%	79.0	77.7	78.5	80.5	82.3	85.8	80.1	76.1	82.4	83.1	82.0	84.8	70.4	68.5	79.1	79.1	75.2	81.8
	30%	87.4	89.3	81.4	82.9	88.5	90.7	90.7	88.6	88.3	89.3	91.2	92.3	82.4	82.2	82.4	82.9	85.0	85.8
MNIST	20%	63.1	66.6	67.1	66.4	68.2	71.6	74.9	71.7	73.7	71.6	71.7	80.3	73.3	72.5	74.0	72.9	71.8	78.8
	30%	62.9	68.1	67.5	66.4	68.1	72.3	81.1	75.7	75.0	73.9	75.1	83.2	78.8	77.1	75.4	74.7	76.2	82.8
BinAlf	20%	42.8	69.5	72.7	73.2	71.1	76.9	73.4	69.6	77.3	75.5	71.6	79.0	71.0	67.5	76.2	75.8	68.8	77.3
	30%	24.0	73.7	74.6	73.7	75.0	78.3	78.3	76.8	79.3	78.4	77.0	82.8	77.8	75.5	78.5	79.0	75.4	81.3
USPS	20%	52.8	75.0	79.9	77.0	77.2	81.8	78.5	77.5	80.2	78.9	76.8	81.6	74.3	76.7	80.2	80.5	77.3	81.6
	30%	44.7	75.9	78.3	77.0	77.7	82.5	83.1	81.3	82.6	81.4	79.4	84.1	79.4	80.4	81.7	82.5	79.9	83.7
COIL20	20%	82.8	81.7	84.7	80.5	79.0	87.5	91.3	92.0	84.8	86.5	88.9	94.5	81.4	83.6	82.9	83.5	83.1	89.3
	30%	91.2	82.9	84.0	81.5	79.2	88.6	95.1	95.6	85.1	86.6	88.7	96.4	87.3	89.0	83.2	83.2	82.6	93.0
ISOLET1	20%	77.5	80.2	64.9	66.1	80.5	79.7	77.6	76.4	73.5	71.2	81.1	80.0	80.9	79.4	72.5	70.7	81.5	81.2
	30%	80.9	80.2	65.1	66.8	81.0	81.7	80.6	79.8	75.8	74.0	84.0	82.6	82.1	82.2	76.8	73.5	84.5	84.1
LMove	20%	20.4	54.6	52.1	52.7	53.8	56.1	64.5	65.1	63.4	61.0	68.1	66.3	53.8	54.3	54.6	54.8	55.4	55.6
	30%	27.8	56.1	52.4	53.6	56.1	59.9	73.5	72.7	68.3	66.7	75.0	75.9	61.5	61.6	60.1	61.2	64.6	65.9
MFeat	20%	53.6	80.8	81.2	81.4	84.8	84.5	88.4	84.0	87.5	85.4	89.5	89.6	87.1	83.9	86.1	84.9	90.1	89.4
	30%	93.0	81.4	81.2	81.5	83.8	84.8	91.3	86.9	88.9	87.4	90.4	91.1	90.0	86.3	88.2	86.3	90.4	91.1

Table 5. Clustering results for RgLPCA

Dataset	Metric	Data Representation			
		Original	PCA	L21PCA	RgLPCA
AT & T	ACC	54.55	54.94	61.88	63.85
	NMI	74.52	74.98	77.86	78.67
	PUR	56.12	56.73	64.29	65.73
MNIST	ACC	53.07	55.46	57.63	59.88
	NMI	49.18	47.83	55.83	56.53
	PUR	56.63	56.64	60.83	62.19
BinAlf	ACC	53.56	55.47	57.01	58.07
	NMI	65.49	66.20	67.54	68.84
	PUR	56.68	58.54	59.47	61.55
USPS	ACC	61.21	59.42	67.29	69.05
	NMI	54.21	51.67	60.70	62.05
	PUR	62.31	60.60	68.91	71.03
COIL20	ACC	55.50	57.04	60.10	62.81
	NMI	69.00	69.50	73.17	74.84
	PUR	58.70	60.08	63.82	66.97

Table 6. Classification results for RgLPCA

Dataset	Labeled	Data Representation			
		Original	PCA	L21PCA	RgLPCA
AT & T	20%	71.85	62.81	76.55	80.55
	30%	82.20	82.52	85.08	87.23
MNIST	20%	60.40	62.45	64.54	68.25
	30%	59.80	63.63	65.94	68.89
BinAlf	20%	37.97	63.03	69.41	71.16
	30%	21.24	68.48	70.43	73.40
USPS	20%	49.76	72.18	75.65	79.07
	30%	42.61	73.31	76.71	80.46
COIL20	20%	69.00	69.50	73.17	74.84
	30%	58.70	60.08	63.82	66.97

tained in gLPCA retains the correct manifold information. The poor random baseline residual results corroborate this. Overall, the properties exhibited in illustrative examples of §3.3 are the basic reasons why gLPCA performs better in clustering and semi-supervised learning as compared with PCA, LE, and other methods.

Acknowledgment. This work is supported by National Natural Science Foundation of China (No.61073116, 61211130309, 61202228) and U.S. National Science Foundation NSF-CCF-0917274, NSF-DMS-0915228.

References

- [1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS 2001*.
- [2] P. Chan, M. Schlag, and J. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. CAD-Integrated Circuits and Systems*, 13:1088–1096, 1994.
- [3] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal component analysis to the exponential family. *Neural Info. Processing Systems (NIPS 2001)*, 2001.
- [4] C. Ding and X. He. K-means clustering via principal component analysis. In *ICML 2004*.
- [5] C. Ding, T. Li, and M. I. Jordan. Convex and semi-nonnegative matrix factorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, 2010.
- [6] C. Ding, D. Zhou, X. He, and H. Zha. R1-PCA: Rotational invariant L1-norm principal component analysis for robust subspace factorization. In *ICML 2006*.
- [7] M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. *Penn State Univ Tech Report CSE-01-007*, 2001.
- [8] K. M. Hall. R-dimensional quadratic placement algorithm. *Management Science*, 17:219–229, 1971.
- [9] X. He and P. Niyogi. Locality preserving projection. In *NIPS 2003*.
- [10] I. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.
- [11] D. Luo, C. Ding, and H. Huang. Towards structural sparsity: An explicit L2/L0 approach. In *ICDM 2010*.
- [12] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(22):2323–2326, 2000.
- [13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
- [14] J. B. Tenenbaum, V. de. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality. *Science*, 290(22):2319–2323, 2000.
- [15] Z. Zhang and Z. Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Scientific Computing*, 26:313–338, 2004.