
Cauchy Graph Embedding

Dijun Luo
Chris Ding
Feiping Nie
Heng Huang

DIJUN.LUO@GMAIL.COM
CHQDING@UTA.EDU
FEIPINGNIE@GMAIL.COM
HENG@UTA.EDU

The University of Texas at Arlington, 701 S. Nedderman Drive, Arlington, TX 76019

Abstract

Laplacian embedding provides a low-dimensional representation for the nodes of a graph where the edge weights denote pairwise similarity among the node objects. It is commonly assumed that the Laplacian embedding results preserve the local topology of the original data on the low-dimensional projected subspaces, *i.e.*, for any pair of graph nodes with large similarity, they should be embedded closely in the embedded space. However, in this paper, we will show that the Laplacian embedding often cannot preserve local topology well as we expected. To enhance the local topology preserving property in graph embedding, we propose a novel Cauchy graph embedding which preserves the similarity relationships of the original data in the embedded space via a new objective. Consequentially the machine learning tasks (such as k Nearest Neighbor type classifications) can be easily conducted on the embedded data with better performance. The experimental results on both synthetic and real world benchmark data sets demonstrate the usefulness of this new type of embedding.

1. Introduction

Unsupervised dimensionality reduction is an important procedure in various machine learning applications which range from image classification (Turk & Pentland, 1991) to genome-wide expression modeling (Alter et al., 2000). Many high-dimensional real world data often intrinsically lie in a low-dimensional space, hence the dimensionality of the data can be reduced without significant loss of information. From the data embedding point of view, we can

classify unsupervised embedding approaches into two categories. Approaches in the first category are to embed data into a linear space via linear transformations, such as principle component analysis (PCA) (Jolliffe, 2002) and multidimensional scaling (MDS) (Cox & Cox, 2001). Both PCA and MDS are eigenvector methods and can model linear variabilities in high-dimensional data. They have been long known and widely used in many machine learning applications.

However, the underlying structure of real data is often highly nonlinear and hence cannot be accurately approximated by linear manifolds. The second category approaches embed data in a nonlinear manner based on different purposes. Recently several promising nonlinear methods have been proposed, including IsoMAP (Tenenbaum et al., 2000), Local Linear Embedding (LLE) (Roweis & Saul, 2000), Local Tangent Space Alignment (Zhang & Zha, 2004), Laplacian Embedding/Eigenmap (Hall, 1971; Belkin & Niyogi, 2003; Luo et al., 2009), and Local Spline Embedding (Xiang et al., 2009) *etc.* Typically, they set up a quadratic objective derived from a neighborhood graph and solve for its leading eigenvectors: Isomap takes eigenvectors associated with the largest eigenvalues; LLE and Laplacian embedding use eigenvectors associated with the smallest eigenvalues. Isomap tries to preserve the global pairwise distances of the input data as measured along the low-dimensional manifold; LLE and Laplacian embedding try to preserve local geometric relationships of the data.

As one of the most successful methods in transductive inference (Belkin & Niyogi, 2004), spectral clustering (Shi & Malik, 2000; Simon, 1991; Ng et al., 2002; Ding et al., 2001), and dimensionality reduction (Belkin & Niyogi, 2003), Laplacian embedding seeks a low-dimensional representation of a set of data points with a matrix of pairwise similarities (*i.e.* a graph data) Laplacian embedding and the related usage of eigenvectors of graph Laplace matrix was first developed in 1970s. It was called *quadratic placement* (Hall, 1971) of graph nodes in a space. The eigenvectors of graph Laplace matrix were used for *graph partitioning* and

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011. Copyright 2011 by the author(s)/owner(s).

connectivity analysis (Fiedler, 1973). This approach became popular in 1990s for circuit layout in VLSI community (please see the review (Alpert & Kahng, 1995)), and graph partitioning (Pothen et al., 1990) for domain decomposition, a key problem in distributed-memory computing. A generalized version of graph Laplacian (p -Laplacian) was also developed for other graph partitioning (Bühler & Hein, 2009; Luo et al., 2010).

It is generally considered that Laplacian embedding has the *local topology preserving* property: a pair of graph nodes with high mutual similarities are embedded nearby in the embedding space, whereas a pair of graph nodes with small mutual similarities are embedded far-way in the embedding space. Local topology preserving property provides a basis for utilizing the quadratic embedding objective function as **regularization** in many applications (Zhou et al., 2003; Zhu et al., 2003; Nie et al., 2010). Such assumption was considered as a desirable property of Laplacian embedding, and many previous research work used it as the regularization term to embed the graph data with preserving local topology (Weinberger et al.; Ando & Zhang).

In this paper, we point out the perceived local topology preserving property of Laplacian embedding does not hold in many applications. More precisely, we first give a precise definition of the local topology preserving property, and then show Laplacian embedding often gives an embedding without preserving local topology in the sense that node pairs with large mutual similarities are not embedded nearby in the embedding space. After that, we will propose a novel Cauchy embedding method that not only has nice nonlinear embedding properties as Laplacian embedding, but also successively preserves the local topology existing in original data. Moreover, we introduce Exponential and Gaussian embedding approaches that further emphasize the data points with large similarities to have small distances in embedding space. Our empirical studies on both synthetic data and real world benchmark data sets demonstrate the promising results of our proposed methods.

2. Laplacian Embedding

We start with a brief introduction to Laplacian embedding/Eigenmap. The input data is a matrix W of pairwise similarities among n data objects. We view W as the edge weights on a graph with n nodes. The task is to embed the nodes of the graph into 1-D space with coordinates (x_1, \dots, x_n) . The objective is that if i, j 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 minimizing (Hall, 1971; Belkin & Niyogi, 2003)

$$\min_{\mathbf{x}} J(\mathbf{x}) = \sum_{ij} (x_i - x_j)^2 w_{ij}. \quad (1)$$

The minimization of $\sum_{ij} (x_i - x_j)^2 w_{ij}$ would get $x_i = 0$ if there is no constraint on the magnitude of the vector \mathbf{x} . Therefore, the normalization $\sum_i x_i^2 = 1$ is imposed. The original objective function is invariant if we replace x_i by $x_i + a$ where a is a constant. Thus the solution is not unique. To fix this uncertainty, we can adjust the constant such that $\sum x_i = 0$ (x_i is centered around 0). With the centering constraint, x_i have mixed signs. With these two constraints, the embedding problem becomes:

$$\min_{\mathbf{x}} \sum_{ij} (x_i - x_j)^2 w_{ij}, \text{ s.t. } \sum_i x_i^2 = 1, \sum_i x_i = 0. \quad (2)$$

The solution of this embedding problem can be easily obtained, because

$$J(\mathbf{x}) = 2 \sum_{ij} x_i (D - W)_{ij} x_j = 2\mathbf{x}^T (D - W)\mathbf{x}, \quad (3)$$

where $D = \text{diag}(d_1, \dots, d_n)$, $d_i = \sum_j W_{ij}$. The matrix $(D - W)$ is called as the graph Laplacian and the embedding solution of minimizing the embedding objective is given by the eigenvectors of

$$(D - W)\mathbf{x} = \lambda\mathbf{x}. \quad (4)$$

Laplacian embedding has been widely used in machine learning, and often as *regularization* for embedding the graph nodes with preserving local topology.

3. The Local Topology Preserving Property of Graph Embedding

In this paper, we study the local topology preserving property of the graph embedding. We first provide definition of local topology preserving, and show that in contrary to the widely accepted conception, Laplacian embedding may not preserve the local topology of original data in the embedded space for many cases.

3.1. Local Topology Preserving

We first provide a definition of *local topology preserving*. Given a symmetric (undirected) graph with edge weights $W = (w_{ij})$, and a corresponding embedding $(\mathbf{x}_1, \dots, \mathbf{x}_n)$ for the n nodes of the graph. We say that the embedding preserves local topology if the following condition holds

$$\text{if } w_{ij} \geq w_{pq}, \text{ then } (x_i - x_j)^2 \leq (x_p - x_q)^2, \forall i, j, p, q. \quad (5)$$

Roughly speaking, this definition says that for any pair of nodes (i, j) , the more similar they are (the bigger the edge weight w_{ij} is), the closer they should be embedded together (the smaller $|x_i - x_j|$ should be).

Laplacian embedding has been widely used in machine learning with a perceived notion of preserving local topology. As a contribution of this paper, we point out here that

this perceived notion of local topology preserving is in fact false in many cases.

Our finding has two aspects. First, at large distance (small similarity):

The quadratic function of the Laplacian embedding emphasizes the large distance pairs, which enforces node pair (i, j) with small w_{ij} to be separated far-away.

Second, at small distance (large similarity):

The quadratic function of the Laplacian embedding de-emphasizes the small distance pairs, leading to many violations of local topology preserving at small distance pairs.

In the following, we will show examples to support our finding. One consequence of the finding is that k Nearest Neighbor (k NN) type classification approach will perform poorly because they rely on the local topology property. After that, we propose a new type of graph embedding method which emphasizes the small distance (large similarity) data pairs and thus enforces the local topology preserving in embedded space.

3.2. Experimental Evidences

Following the work of Isomap (Tenenbaum et al., 2000) and LLE (Roweis & Saul, 2000), we do experiments on two simple “manifold” data (C-shape and S-shape). For the manifold data in Figure 1, if the embedding preserves *local topology*, we expect the 1D embedding¹ results will simply flatten (unroll) the manifold. In the Laplacian embedding results, the flattened manifold consists of data points ordered as $x_1, x_2, x_3, \dots, x_{56}, x_{57}$. This is not a simple unrolling of the original manifold. The Cauchy embedding (see §4) results are a simple unrolling of the original manifold — indicating a *local topology preserving* embedding. For the visualization purpose, we use the blue lines to connect the data points which are neighboring in the embedded space.

We also apply Laplacian embedding and Cauchy embedding to some manifolds which have a little more complicated structures, see Figure 2. The manifolds lie on four letters “I”, “C”, “M”, “L”. Each letter consists of 150 2D data points. One can see that Cauchy embedding results (bottom row of Figure 2) preserve more local topology than Laplacian embedding (top row). Notice that Cauchy embedding and Laplacian embedding give the identical result for the letter “L”. Because the manifold of the “L” shape is originally smooth. However, for

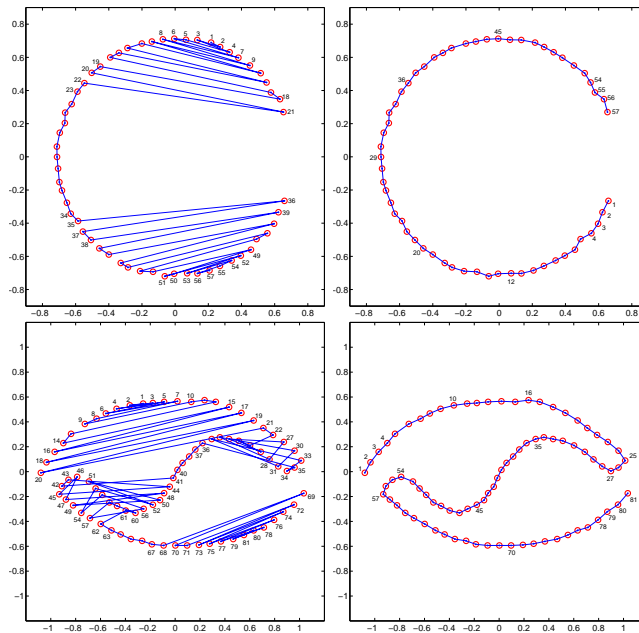


Figure 1. Embedding results comparison. A C-shape and S-shape manifold dataset are visualized in both figures. After performing Laplacian embedding (left) and Cauchy embedding (right), we use the numbers 1, 2, \dots to indicate the ordering of data points on the flattened manifold. For the visualization purpose, we use the blue lines to connect the data points which are neighboring in the embedded space.

other shapes of manifolds, such as, “M”, the Cauchy embedding gives perfectly local topology preserving results, while the Laplacian embedding leads to disordering on the original manifold. For both Figures 1 and 2, w_{ij} is computed as $w_{ij} = \exp(-\|x_i - x_j\|^2 / \bar{d}^2)$ where \bar{d} is the average Euclidean distance among all the data points, *i.e.* $\bar{d} = (\sum_{i \neq j} \|x_i - x_j\|) / (n(n - 1))$.

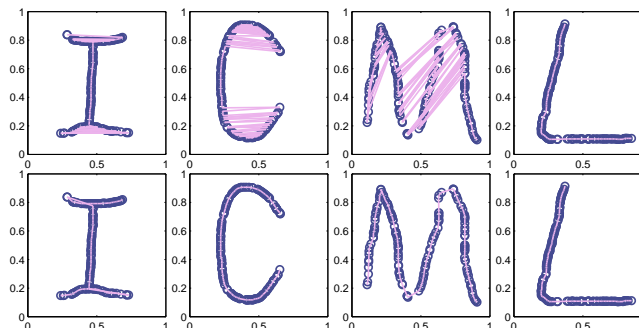


Figure 2. Embedding the “I”, “C”, “M”, “L” manifolds 2D data points using Laplacian embedding (top) and Cauchy embedding (bottom). The ordering scheme is the same as in Figure 1.

¹The 1D embedding is identical to an ordering of graph nodes.

4. Cauchy Embedding

In this paper, we propose a new graph embedding approach that emphasizes short distance, and ensure that locally, the more similar two nodes are, the closer they will be in the embedding space.

We motivate our approach as the following. Starting with the Laplacian embedding. The key idea is that for a pair (i, j) with large w_{ij} , $(x_i - x_j)^2$ should be small such that the objective function is minimized. Now, if $(x_i - x_j)^2 \equiv \Gamma_1(|x_i - x_j|)$ is small, so is

$$\frac{(x_i - x_j)^2}{(x_i - x_j)^2 + \sigma^2} \equiv \Gamma_2(|x_i - x_j|).$$

Furthermore, function $\Gamma_1(\cdot)$ is monotonic as is function $\Gamma_2(\cdot)$. Therefore, instead of minimizing the quadratic function of Eq. (2), we minimize

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{ij} \frac{(x_i - x_j)^2}{(x_i - x_j)^2 + \sigma^2} w_{ij}, \\ \text{s.t.}, \quad & \|\mathbf{x}\|^2 = 1, \mathbf{e}^T \mathbf{x} = 0. \end{aligned} \quad (6)$$

The function involved can be simplified, since

$$\frac{(x_i - x_j)^2}{(x_i - x_j)^2 + \sigma^2} = 1 - \frac{\sigma^2}{(x_i - x_j)^2 + \sigma^2}.$$

Thus the optimization for the new embedding is

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{ij} \frac{w_{ij}}{(x_i - x_j)^2 + \sigma^2}, \\ \text{s.t.} \quad & \sum_i x_i^2 = 1, \sum_i x_i = 0. \end{aligned} \quad (7)$$

We call this new embedding as Cauchy embedding because $f(x) = 1/(x^2 + \sigma^2)$ is the usual Cauchy distribution.

The most important difference between the objective function of Cauchy embedding [Eq. (6) or Eq. (7)] vs. the objective function of Laplacian embedding [Eq. (1)] is the following. For Laplacian embedding, large distance $(x_i - x_j)^2$ terms contribute more because of the quadratic form, whereas for Cauchy embedding, small distance $(x_i - x_j)^2$ terms contribute more. This key difference ensures Cauchy embedding exhibits stronger local topology preserving property.

4.1. Multi-dimensional Cauchy Embedding

For representation simplicity and clarity, we consider 2D embedding first. For 2D embedding, each node i is embedded in 2D space with coordinates (x_i, y_i) . The usual

Laplacian embedding can be formulated as

$$\min_{\mathbf{x}, \mathbf{y}} \quad \sum_{ij} [(x_i - x_j)^2 + (y_i - y_j)^2] w_{ij}, \quad (8)$$

$$\text{s.t.} \quad \|\mathbf{x}\|^2 = 1, \mathbf{e}^T \mathbf{x} = 0, \quad (9)$$

$$\|\mathbf{y}\|^2 = 1, \mathbf{e}^T \mathbf{y} = 0, \quad (10)$$

$$\mathbf{x}^T \mathbf{y} = 0, \quad (11)$$

where $\mathbf{e} = (1, \dots, 1)^T$. The constraint $\mathbf{x}^T \mathbf{y} = 0$ is important, because without it, the optimization obtains its optimal value at $\mathbf{x} = \mathbf{y}$.

The 2D Cauchy is motivated with the following optimization

$$\min_{\mathbf{x}, \mathbf{y}} \sum_{ij} \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{(x_i - x_j)^2 + (y_i - y_j)^2 + \sigma^2} w_{ij}, \quad (12)$$

with the same constraints Eqs. (8-11). This is simplified to

$$\max_{\mathbf{x}, \mathbf{y}} \sum_{ij} \frac{w_{ij}}{(x_i - x_j)^2 + (y_i - y_j)^2 + \sigma^2}. \quad (13)$$

In general, the p -dimensional Cauchy embedding to $R = (\mathbf{r}_1, \dots, \mathbf{r}_n) \in \mathfrak{R}^{p \times n}$ is obtained by optimizing

$$\max_R \quad J(R) = \sum_{ij} \frac{w_{ij}}{\|\mathbf{r}_i - \mathbf{r}_j\|^2 + \sigma^2}, \quad (14)$$

$$\text{s.t.} \quad RR^T = I, R\mathbf{e} = \mathbf{0}. \quad (15)$$

4.2. Exponential and Gaussian Embedding

In Cauchy embedding the short distance pairs are emphasized more than large distance pairs, in comparison to Laplacian embedding. We can further emphasize the short distance pairs and **de-emphasize** large distance pairs by the following **Gaussian embedding**:

$$\max_{\mathbf{x}} \sum_{ij} \exp \left[-\frac{(x_i - x_j)^2}{\sigma^2} \right] w_{ij}, \quad (16)$$

$$\text{s.t.}, \quad \|\mathbf{x}\|^2 = 1, \mathbf{e}^T \mathbf{x} = 0, \quad (17)$$

or the **exponential embedding**

$$\max_{\mathbf{x}} \sum_{ij} \exp \left[-\frac{|x_i - x_j|}{\sigma} \right] w_{ij}, \quad (18)$$

$$\text{s.t.}, \quad \|\mathbf{x}\|^2 = 1, \mathbf{e}^T \mathbf{x} = 0, \quad (19)$$

In general, we may introduce the **decay** function $\Gamma(d_{ij})$ and write the three embedding objective as

$$\max_{\mathbf{x}} \sum_{ij} \Gamma(|x_i - x_j|) w_{ij}, \quad \text{s.t.}, \quad \|\mathbf{x}\|^2 = 1, \mathbf{e}^T \mathbf{x} = 0, \quad (20)$$

Here is a list of decay functions:

$$\text{Laplacian embed: } \Gamma_{\text{Laplace}}(d_{ij}) = -d_{ij}^2 \quad (21)$$

$$\text{Cauchy embed: } \Gamma_{\text{Cauchy}}(d_{ij}) = \frac{1}{d_{ij}^2 + \sigma^2} \quad (22)$$

$$\text{Gaussian embed: } \Gamma_{\text{Gaussian}}(d_{ij}) = e^{-d_{ij}^2/\sigma^2} \quad (23)$$

$$\text{Exponential embed: } \Gamma_{\text{exp}}(d_{ij}) = e^{-d_{ij}/\sigma} \quad (24)$$

$$\text{Linear embed: } \Gamma_{\text{linear}}(d_{ij}) = -d_{ij} \quad (25)$$

Notice that linear embed here is equivalent to the p -Laplacian when $p \rightarrow 1$ (Bühler & Hein, 2009). It is easy to generalize to other decay functions. We discuss two properties for decay functions.

(1) There is one requirement on decay functions: $\Gamma(d)$ must be a monotonically decreasing as d increases. If this monotonicity is violated, the embedding is not meaningful.

(2) A decay function is undefined up to a constant, *i.e.*, $\Gamma'(d_{ij}) = \Gamma(d_{ij}) + c$ leads to the same embedding for any constant c .

One can see the different behaviors of these decay functions as $|\Gamma(d)|$ vs d , which are shown in Figure 3. We see that in $\Gamma_{\text{Laplace}}(d)$ and $\Gamma_{\text{linear}}(d)$, large distance pairs dominate, whereas in $\Gamma_{\text{exp}}(d)$, Γ_{Gaussian} , and $\Gamma_{\text{Cauchy}}(d)$, small distance pairs dominate.

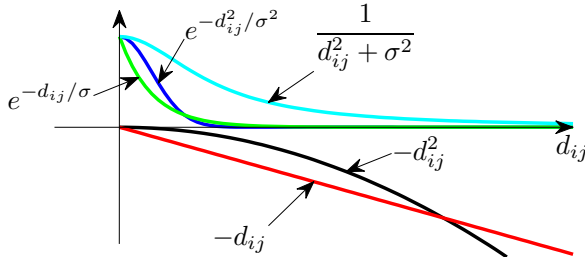


Figure 3. Decay functions for 5 different embedding approaches: $-d_{ij}^2$ for Laplacian embedding, $1/(d_{ij}^2 + \sigma^2)$ for Cauchy embedding, $e^{-d_{ij}^2/\sigma^2}$ for Gaussian embedding, $e^{-d_{ij}/\sigma}$ for Exponential embedding, and $-d_{ij}$ for linear embedding.

4.3. Algorithms to Compute Cauchy Embedding

Our algorithm is based on the following theorem.

Theorem 1 If $J(R)$ defined in Eq. (14) is Lipschitz continuous with constant $L \geq 0$, and

$$R^* = \arg \min_R \left\| R - \left(\tilde{R} + \frac{1}{L} \nabla J(\tilde{R}) \right) \right\|_F^2, \quad (26)$$

$$\text{s.t. } RR^T = I, Re = \mathbf{0},$$

then $J(R^*) \geq J(\tilde{R})$.

Proof. Since $J(R)$ is Lipschitz continuous with constant L , from (Nesterov, 2003), we have

$$J(X) \leq J(Y) + \langle X - Y, \nabla J(X) \rangle + \frac{L}{2} \|X - Y\|_F^2, \forall X, Y$$

By apply this inequality, we further obtain

$$J(\tilde{R}) \leq J(R^*) + \langle \tilde{R} - R^*, \nabla J(\tilde{R}) \rangle + \frac{L}{2} \|\tilde{R} - R^*\|_F^2, \quad (27)$$

By definition of R^* , we have

$$\begin{aligned} & \left\| R^* - \left(\tilde{R} + \frac{1}{L} \nabla J(\tilde{R}) \right) \right\|_F^2 \\ & \leq \left\| \tilde{R} - \left(\tilde{R} + \frac{1}{L} \nabla J(\tilde{R}) \right) \right\|_F^2 = \frac{1}{L^2} \|\nabla J(\tilde{R})\|_F^2, \end{aligned}$$

or

$$\begin{aligned} & \|R^* - \tilde{R}\|_F^2 - 2\langle R^* - \tilde{R}, \frac{1}{L} \nabla J(\tilde{R}) \rangle + \frac{1}{L^2} \|\nabla J(\tilde{R})\|_F^2 \\ & \leq \frac{1}{L^2} \|\nabla J(\tilde{R})\|_F^2, \end{aligned}$$

$$\|R^* - \tilde{R}\|_F^2 + 2\langle \tilde{R} - R^*, \frac{1}{L} \nabla J(\tilde{R}) \rangle \leq 0, \quad (28)$$

By combining Eq. (27) and Eq. (28) and notice that $L \geq 0$, we have

$$J(R^*) \geq J(\tilde{R}),$$

which completes the proof.

Further more, for Eq. (26), we have the following solution,

Theorem 2 $R^* = V^T$ is the optimal solution of Eq. (26), where $USV^T = M(I - \mathbf{e}\mathbf{e}^T/n)$, is the Singular Value Decomposition (SVD) of $M(I - \mathbf{e}\mathbf{e}^T/n)$ and $M = \tilde{R} + \frac{1}{L} \nabla J(\tilde{R})$.

Proof. Let $M = \tilde{R} + \frac{1}{L} \nabla J(\tilde{R})$, by applying the Lagrangian multipliers Λ and μ , we get following Lagrangian function,

$$\mathcal{L} = \|R - M\|_F^2 + \langle RR^T - I, \Lambda \rangle + \mu^T Re, \quad (29)$$

By taking the derivative *w.r.t.* R , and setting it to zero, we have

$$2R - 2M + \Lambda R + \mu \mathbf{e}^T = 0, \quad (30)$$

Since $Re = 0$, and $\mathbf{e}^T \mathbf{e} = n$, we have $\mu = 2Me/n$, and

$$(I + \Lambda)R = M(I - \mathbf{e}\mathbf{e}^T/n), \quad (31)$$

Since $USV^T = M(I - \mathbf{e}\mathbf{e}^T/n)$, we let $R^* = V^T$ and $\Lambda = US - I$, then the KKT condition of the Lagrangian function is satisfied. Notice that the objective function of

Eq. (26) is convex *w.r.t* R . Thus $R^* = V^T$ is the optimal solution of Eq. (26).

From the above theorem, we use the following algorithm to solve the Cauchy embedding problem.

Algorithm. Starting from an initial solution and an initial guess of Lipschitz continuous constant L , we iteratively update the current solution until convergence. Each iteration consists of the following steps:

(1) Compute M ,

$$M \leftarrow R + \frac{1}{L} \nabla J(R) \quad (32)$$

(2) Compute the SVD of $M(I - \mathbf{e}\mathbf{e}^T/n)$: $USV^T = M(I - \mathbf{e}\mathbf{e}^T/n)$, and set $R \leftarrow V^T$,

(3) If Eq. (28) does not hold, increase L by $L \leftarrow \gamma L$.

We use the Laplacian embedding results as the initial solution for the gradient algorithm.

5. Experimental Results

5.1. Experiments on Image Embedding

We are going to demonstrate the advantages of Cauchy Embedding using two-dimensional visualization. We select four letters in BinAlpha data set², (“C”, “P”, “X”, “Z”) and four digits from MNIST (Cun et al., 1998) (“0”, “3”, “6”, “9”) and scale the data such that the average pairwise distance is 1. Algorithm in §4.3 is run with the default settings mentioned above. The embedding results are drawn in Figure 4(a) and 4(b).

In Laplacian embedding results, all images from different groups collapse together, except some outliers. *E.g.* in left panel of Figure 4(a), letters “C” and “P”, and “Z” are visually difficult to be distinguished from each other. Thus, one image of letter “P” is far way from all other images. However, in Cauchy embedding results, the distance among images are balanced out with much more clear visualizations. By employing the minimum distance penalty, the objects are distributed more evenly.

5.2. Embedding on US Map

In the previous sections, we provide insight analysis of the ordering capacity of Cauchy Embedding. here we employ our algorithm on the 49 cities in United State (capitals of 48 states in US mainland plus Washington DC). In this experiment, we seek a path through all cities using the first embedding directions of both Laplacian embedding and Cauchy embedding. For both method, we construct the map using the spherical distances among cities: $w_{ij} = \exp(-d_{ij}^2/\bar{d}^2)$ where d_{ij} is the spherical distance

between city i and j , and \bar{d} is the average pairwise distance of all the selected 49 cities. Then standard Laplacian and Cauchy embedding with default settings are employed.

For Laplacian Embedding, the cities are sorted by the second eigenvector of graph Laplacian. Here we assume that all cities lie in a 1-D manifold. The results are shown in Figure 5(a) and 5(b). In Figure 5(a), the terminal cities are *Olympia* from west and *Augusta* from east. Thus the path has to be go though all other cities in the middle. Thus the total path is long. However, Cauchy Embedding result captures the tight 1-D manifold structure, see Figure 5(b). The terminal cities are *Phoenix* in Arizona and *Sacramento* in California. And the resulting path goes through all cities in an efficient order.

5.3. Classification And Smoothness Comparisons

We use five data sets to demonstrate the classification performance in the embedded space of Exponential embedding and Cauchy embedding algorithms, and compare them to Laplacian embedding. The data sets include 9 UCI data sets, AMLALL, CAR, Auto, Cars, Dermatology, Ecoli, Iris, Prostate, and Zoo³, and two public image data set: JAFFE⁴, AT&T⁵.

The classification accuracy is computed by the nearest neighbor classifier on the embedded space, *i.e.* using the Euclidean distance on the embedding space to establish the nearest neighbor classifier. The embedded dimension and σ in Laplacian embedding are tuned such that the Laplacian embedding method reaches its best classification accuracy where σ is the Gaussian similarity parameter to compute $W : w_{ij} = \exp(-\|x_i - x_j\|^2/\sigma^2)$. Then we use the Laplacian embedding result as initialization to run Exponential embedding and Cauchy embedding, respectively. The results of classification accuracy on the embedded space of Laplacian, Gaussian, Exponential and Cauchy methods are recorded respectively, and are reported in Table 1. From the results we can see that Exponential embedding and Cauchy embedding tend to outperform Laplacian embedding in terms of classification accuracy.

We also evaluate the improvement of Eq.(5) as follows. We find 50 pairs of data points with the greatest W_{ij} and pick up all possible pairs of (i, j) , 2500 combinations in total, to check how many of them conflict with Eq.(5). The numbers of violations for Laplacian, Gaussian, Exponential and Cauchy methods are also reported in Table 1.

³<http://www.ics.uci.edu/mlearn/MLRepository.html>

⁴<http://kasrl.org/jaffe.html>

⁵http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.tar.Z

²<http://www.cs.toronto.edu/~roweis/data.html>

Cauchy Graph Embedding

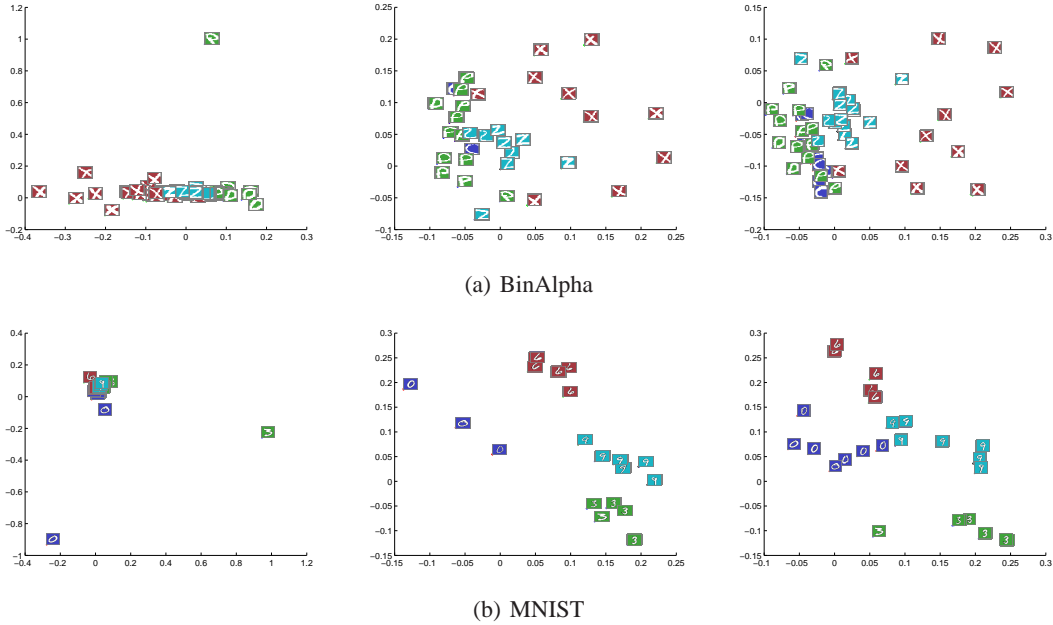


Figure 4. 2D visualizations on two data sets for Laplacian Embedding (left), Cauchy Embedding (middle) and Cauchy Embedding with minimum distance penalty (right) using Eq. 24.

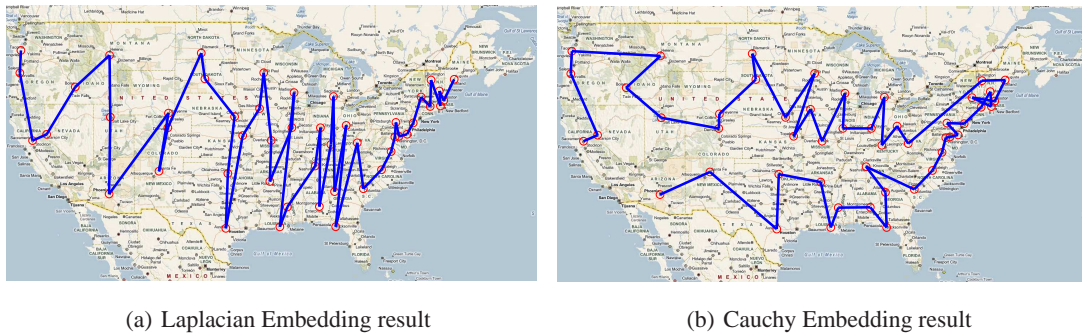


Figure 5. Cities ordering by Laplacian Embedding and Cauchy Embedding. For Laplacian Embedding, the cities are sorted by the second eigenvector of graph Laplacian.

6. Conclusion

Although many previous research work used Laplacian embedding as the regularization item to embed the graph data with preserving local topology, in this paper, we point out such perceived local topology preserving property of Laplacian embedding does not hold in many applications. In order to preserve the local topology in original data on low-dimensional manifold, we proposed a novel Cauchy Graph Embedding that emphasizes short distance and ensures that the more similar two nodes are, the closer they will be in the embedding space. Experimental results on both synthetic data and real world data demonstrate Cauchy embedding can successfully preserve local topology on projected space. Classification results on five data

sets show Cauchy embedding achieve higher accuracy than Laplacian embedding.

Acknowledgments This research was supported by NSF-CCF 0830780, NSF-CCF 0917274, NSF-DMS 0915228.

References

- Alpert, C.J. and Kahng, A.B. Recent directions in netlist partitioning: a survey. *Integration, the VLSI Journal*, 19:1–81, 1995.
- Alter, O., Brown, P., and Botstein, D. Singular value decomposition for genome-wide expression data processing and modeling. *PNAS*, 97(18):10101–10106, 2000.
- Ando, R. K. and Zhang, T. Learning on graph with laplacian regularization. In *Advances in Neural Information Processing Systems 19*, pp. 25–32.

Cauchy Graph Embedding

Table 1. Classification accuracy and number of violations of Eq.(5) (inside the parenthesis) in the embedded space of Laplacian, Gaussian, Cauchy, and Exponential methods.

DATA	LAPLACIAN	CAUCHY	GAUSSIAN	EXPONENTIAL
AMLALL	0.819 (496)	0.822 (416)	0.819 (440)	0.819 (426)
CAR	0.639 (536)	0.657 (536)	0.656 (532)	0.633 (530)
AUTO	0.392 (388)	0.427 (312)	0.425 (324)	0.372 (328)
CARS	0.635 (284)	0.705 (272)	0.651 (284)	0.608 (372)
DERMATOLOGY	0.870 (540)	0.881 (360)	0.880 (376)	0.880 (396)
ECOLI	0.788 (564)	0.821 (432)	0.821 (396)	0.765 (436)
IRIS	0.888 (428)	0.887 (276)	0.889 (192)	0.884 (404)
JAFFE	0.897 (552)	0.899 (452)	0.899 (468)	0.885 (372)
AT&T	0.728 (596)	0.791 (480)	0.788 (516)	0.728 (500)
PROSTATE	0.875 (684)	0.882 (580)	0.877 (500)	0.845 (472)
ZOO	0.908 (436)	0.916 (336)	0.912 (352)	0.914 (420)

- Belkin, M. and Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comp.*, 15(6): 1373–1396, 2003.
- Belkin, M. and Niyogi, P. Semi-supervised learning on riemannian manifolds. *Machine Learning*, pp. 209–239, 2004.
- Bühler, T. and Hein, M. Spectral Clustering based on the graph p -Laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 81–88. ACM, 2009.
- Cox, T. F. and Cox, M. A. A. Multidimensional scaling. *Chapman and Hall*, 2001.
- Cun, Y. L. Le, Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998.
- Ding, C., He, X., Zha, H., Gu, M., and Simon, H. A min-max cut algorithm for graph partitioning and data clustering. *Proc. IEEE Int’l Conf. Data Mining (ICDM)*, pp. 107–114, 2001.
- Fiedler, M. Algebraic connectivity of graphs. *Czech. Math. J.*, 23:298–305, 1973.
- Hall, K. M. R-dimensional quadratic placement algorithm. *Management Science*, 17:219–229, 1971.
- Jolliffe, I.T. *Principal Component Analysis*. Springer, 2nd edition, 2002.
- Luo, D., Ding, C., Huang, H., and Li, T. Non-negative Laplacian Embedding. In *2009 Ninth IEEE International Conference on Data Mining*, pp. 337–346. IEEE, 2009.
- Luo, D., Huang, H., Ding, C., and Nie, F. On the eigenvectors of p -Laplacian. *Machine learning*, 81(1):37–51, 2010. ISSN 0885-6125.
- Nesterov, Y. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2003.
- Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
- Nie, F., Xu, D., Tsang, I., and Zhang, C. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932, 2010.
- Pothen, A., Simon, H. D., and Liou, K. P. Partitioning sparse matrices with egeenvectors of graph. *SIAM Journal of Matrix Anal. Appl.*, 11:430–452, 1990.
- Roweis, S. and Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- Shi, Jianbo and Malik, Jitendra. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- Simon, H. D. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2(2/3):135–148, 1991.
- Tenenbaum, J.B., de Silva, V., and Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
- Turk, M. A. and Pentland, A. P. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.
- Weinberger, K. Q., Sha, F., Zhu, Q., and Saul, L. K. Graph laplacian regularization for large-scale semidefinite programming. In *NIPS*, pp. 1489–1496.
- Xiang, S., Nie, F., Zhang, C., and Zhang, C. Nonlinear dimensionality reduction with local spline embedding. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1285–1298, 2009.
- Zhang, Z. and Zha, Z. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Scientific Computing*, 26:313–338, 2004.
- Zhou, D., Bousquet, O., Lal, T.N., Weston, J., and Schölkopf, B. Learning with local and global consistency. *Proc. Neural Info. Processing Systems*, 2003.
- Zhu, X., Ghahramani, Z., and Lafferty, J. Semi-supervised learning using gaussian fields and harmonic functions. *Proc. Int’l Conf. Machine Learning*, 2003.