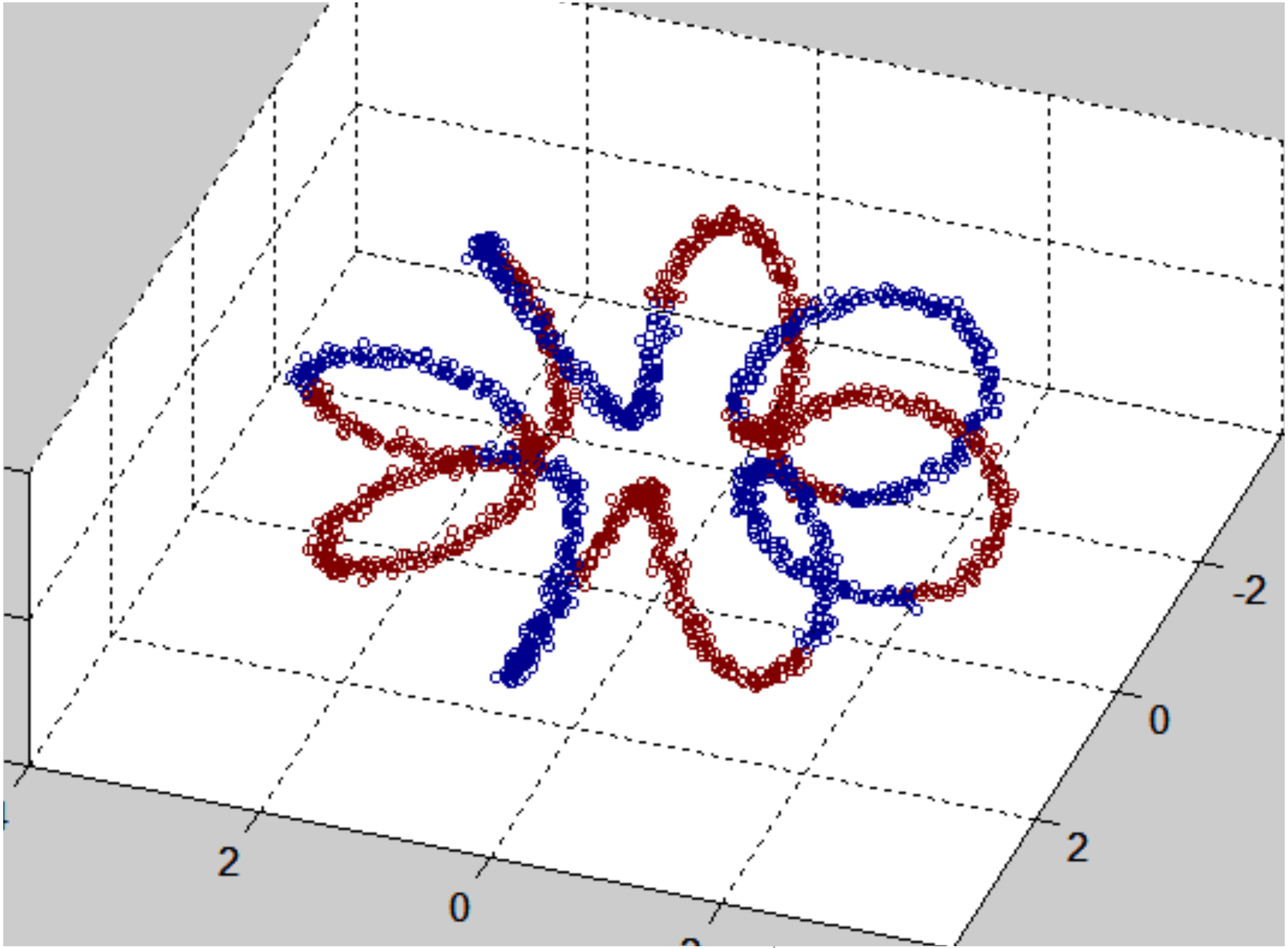
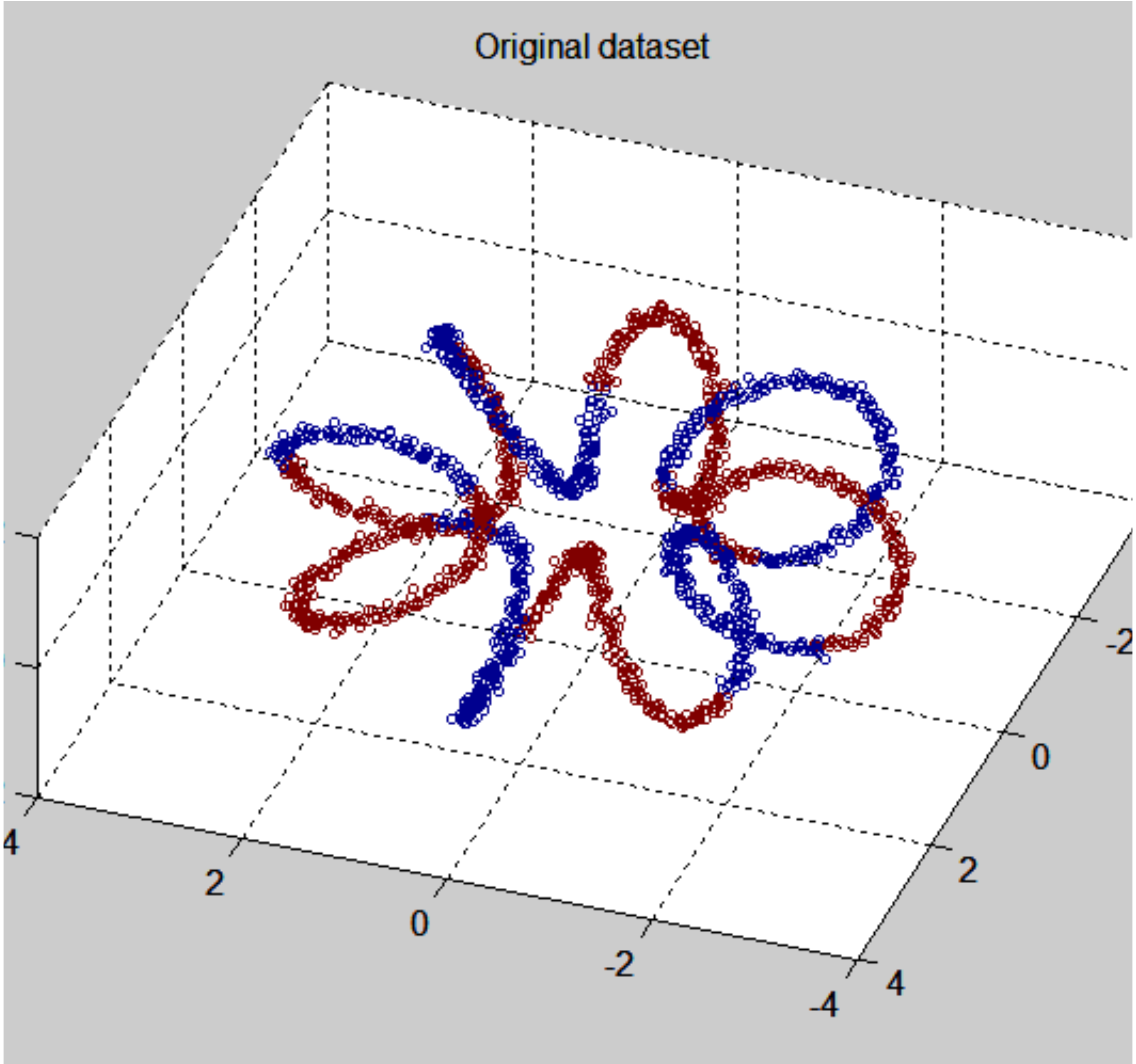
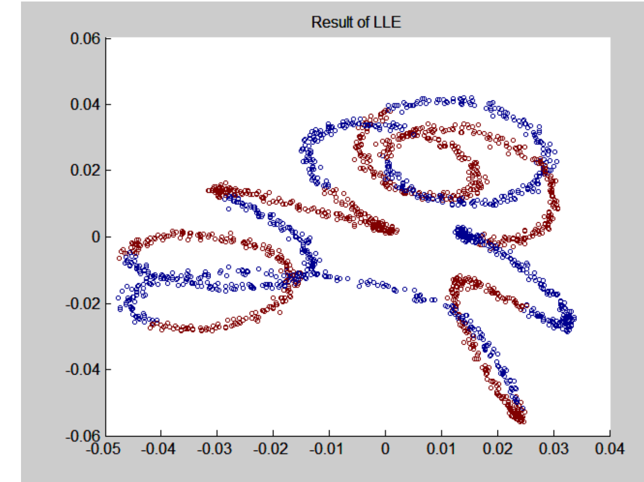
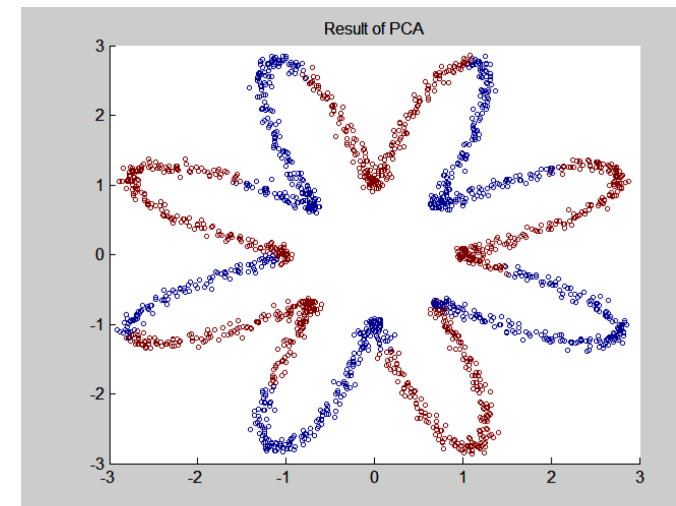
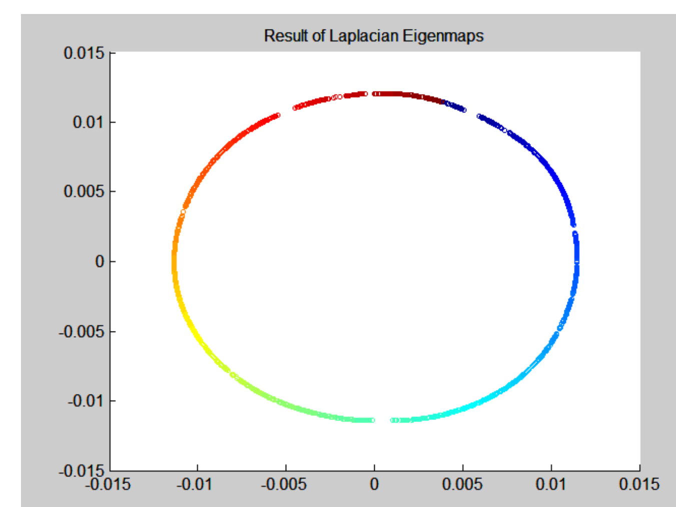
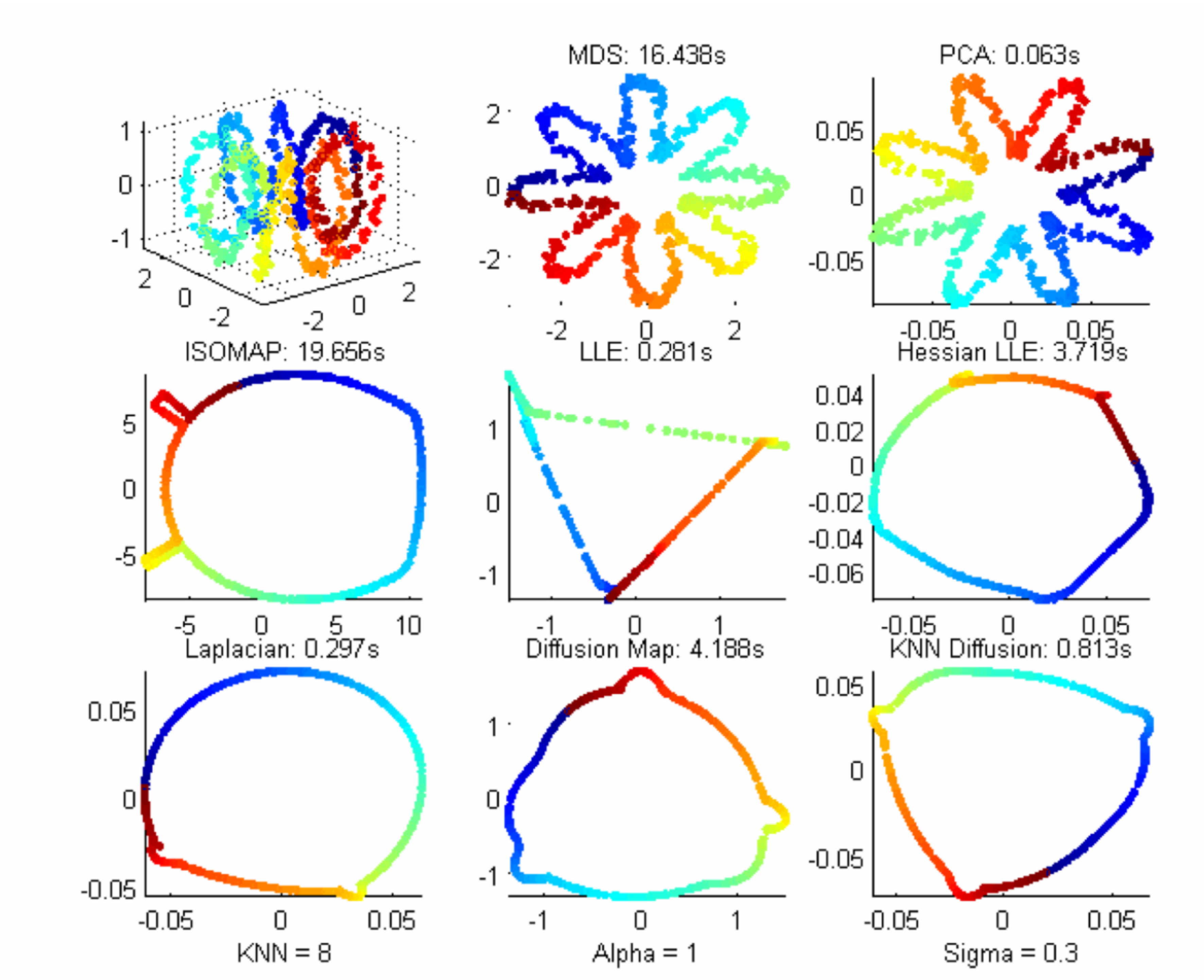


Dimensionality reduction

Locally linear embedding
Laplacian Eigenmap

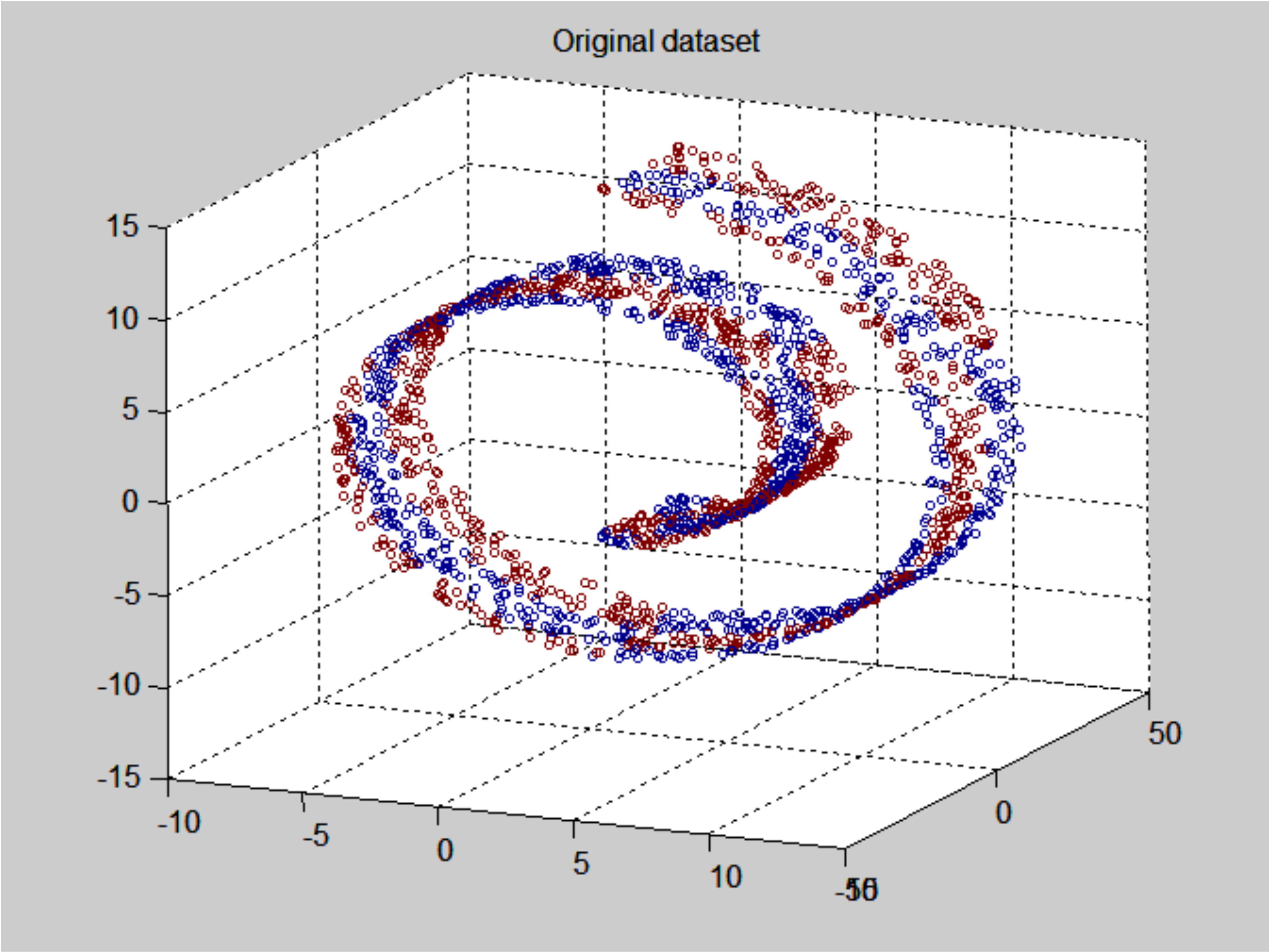

Original dataset
4
2
0
-2
-4
-2
0
2
4

Result of LLE
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.05 -0.04 -0.03 -0.02 -0.01 0 0.01 0.02 0.03 0.04

Result of PCA
3
2
1
0
-1
-2
-3
-3 -2 -1 0 1 2 3

Result of Laplacian Eigenmaps
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.015 -0.01 -0.005 0 0.005 0.01 0.015

Helix

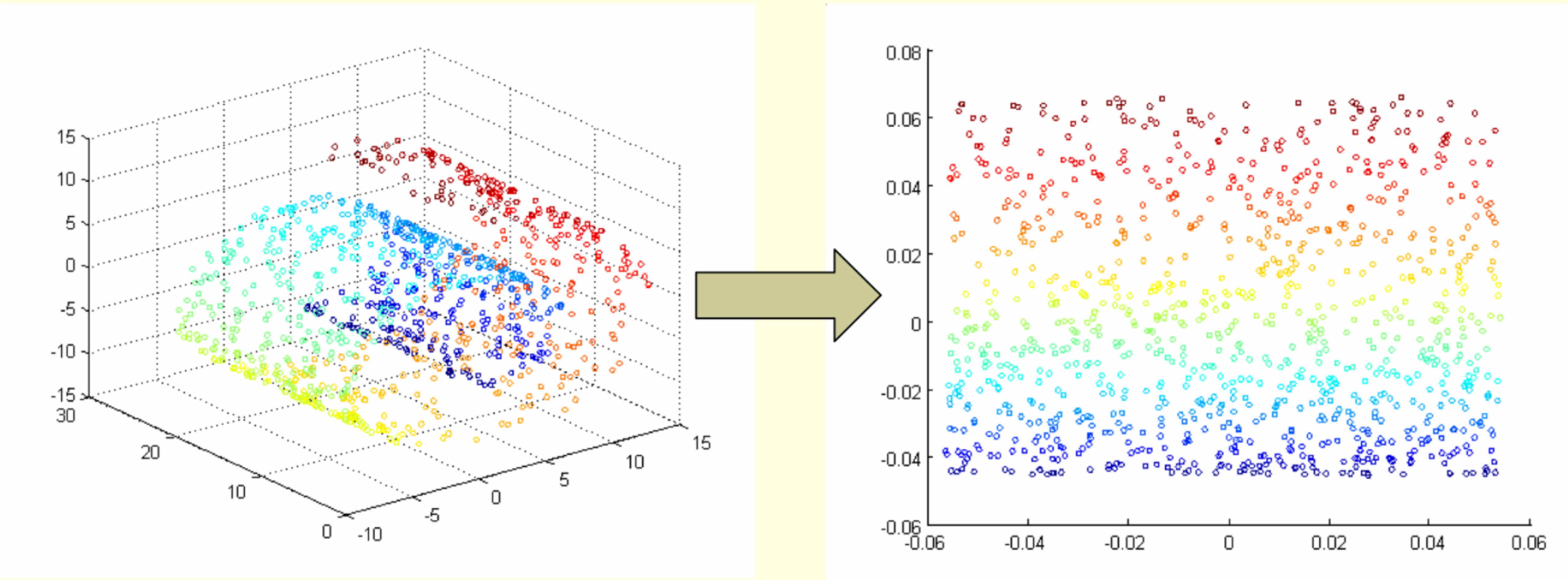


MDS: 16.438s
PCA: 0.063s
ISOMAP: 19.656s
LLE: 0.281s
Hessian LLE: 3.719s
Laplacian: 0.297s
Diffusion Map: 4.188s
KNN Diffusion: 0.813s
KNN = 8
Alpha = 1
Sigma = 0.3

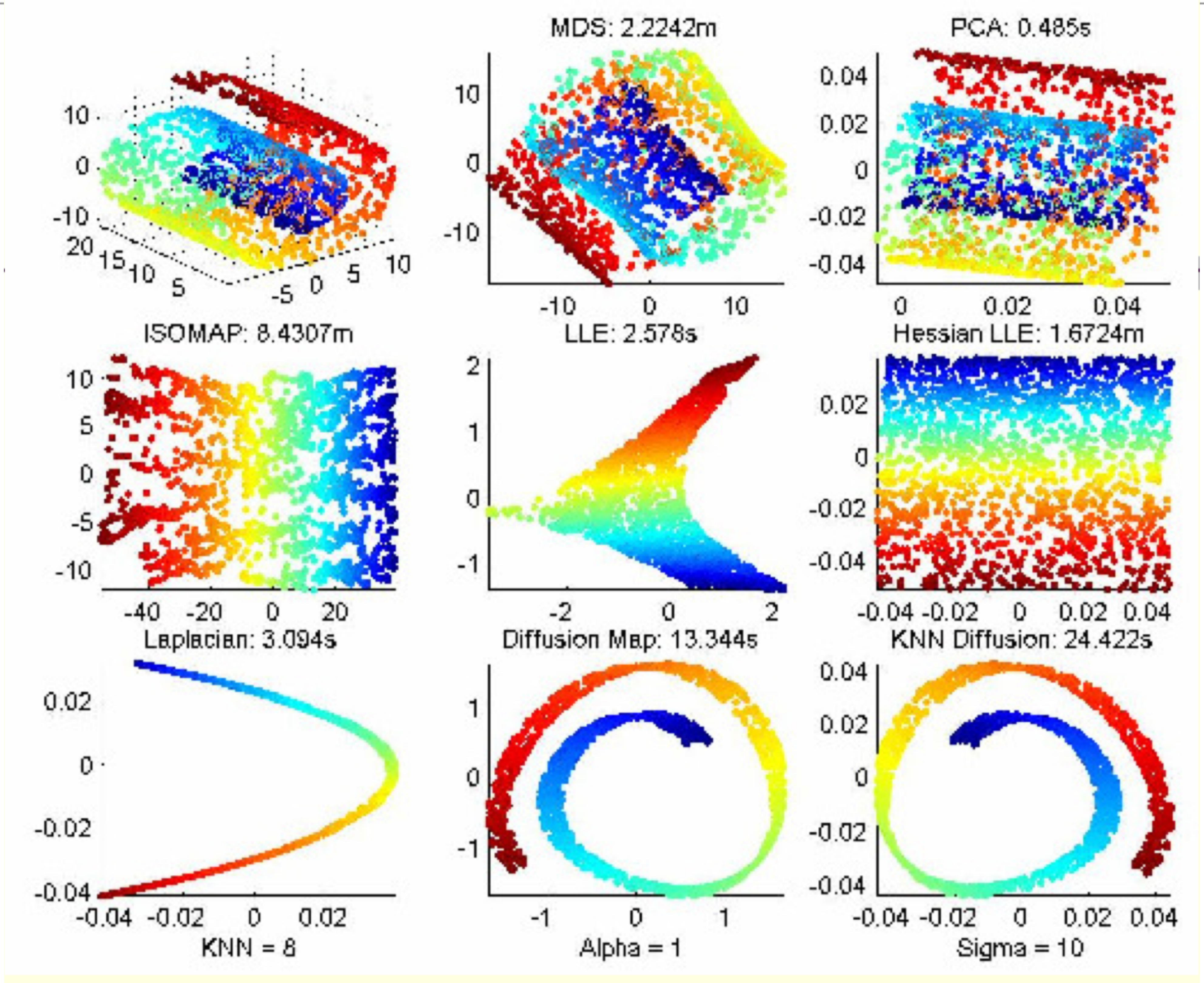
Swiss



Discover embedding



Example



MDS: 2.2242m
PCA: 0.485s
ISOMAP: 8.4307m
LLE: 2.578s
Hessian LLE: 1.6724m
Laplacian: 3.094s
Diffusion Map: 13.344s
KNN Diffusion: 24.422s
KNN = 8
Alpha = 1
Sigma = 10

Science paper

- Nonlinear dimensionality Reductiom by locally linear embedding

Science AAAS

Nonlinear Dimensionality Reduction by Locally Linear Embedding
Author(s): Sam T. Roweis and Lawrence K. Saul
Source: *Science*, New Series, Vol. 290, No. 5500 (Dec. 22, 2000), pp. 2323-2326
Published by: American Association for the Advancement of Science

Nonlinear Dimensionality Reduction by Locally Linear Embedding

Sam T. Roweis[1] and Lawrence K. Saul[2]

Many areas of science depend on exploratory data analysis and visualization. The need to analyze large amounts of multivariate data raises the fundamental problem of dimensionality reduction: how to discover compact representations of high-dimensional data. Here, we introduce locally linear embedding (LLE), an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs. Unlike clustering methods for local dimensionality reduction, LLE maps its inputs into a single global coordinate system of lower dimensionality, and its optimizations do not involve local minima. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, such as those generated by images of faces or documents of text.

How do we judge similarity? Our mental representations of the world are formed by processing large numbers of sensory inputs—including, for example, the pixel intensities of images, the power spectra of sounds, and the joint angles of articulated bodies. While complex stimuli of this form can be represented by points in a high-dimensional vector space, they typically have a much more compact description. Coherent structure in the world leads to strong correlations between inputs (such as between neighboring pixels in images), generating observations that lie on or close to a smooth low-dimensional manifold. To compare and classify such observations—in effect, to reason about the world—depends crucially on modeling the nonlinear geometry of these low-dimensional manifolds.

Scientists interested in exploratory analysis or visualization of multivariate data (*1*) face a similar problem in dimensionality reduction. The problem, as illustrated in Fig. 1, involves mapping high-dimensional inputs into a low-dimensional "description" space with as many coordinates as observed modes of variability. Previous approaches to this problem, based on multidimensional scaling (MDS) (*2*), have computed embeddings that attempt to preserve pairwise distances [or generalized disparities (*3*)] between data points; these distances are measured along straight lines or, in more sophisticated usages of MDS such as Isomap (*4*), along shortest paths confined to the manifold of observed inputs. Here, we take a different approach, called locally linear embedding (LLE), that eliminates the need to estimate pairwise distances between widely separated data points. Unlike previous methods, LLE recovers global nonlinear structure from locally linear fits.

The LLE algorithm, summarized in Fig. 2, is based on simple geometric intuitions. Suppose the data consist of N real-valued vectors $\vec{X}_i$, each of dimensionality D, sampled from some underlying manifold. Provided there is sufficient data (such that the manifold is well-sampled), we expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold. We characterize the local geometry of these patches by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are measured by the cost function

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2 \qquad (1)$$

which adds up the squared distances between all the data points and their reconstructions. The weights W_{ij} summarize the contribution of the jth data point to the ith reconstruction. To compute the weights W_{ij}, we minimize the cost

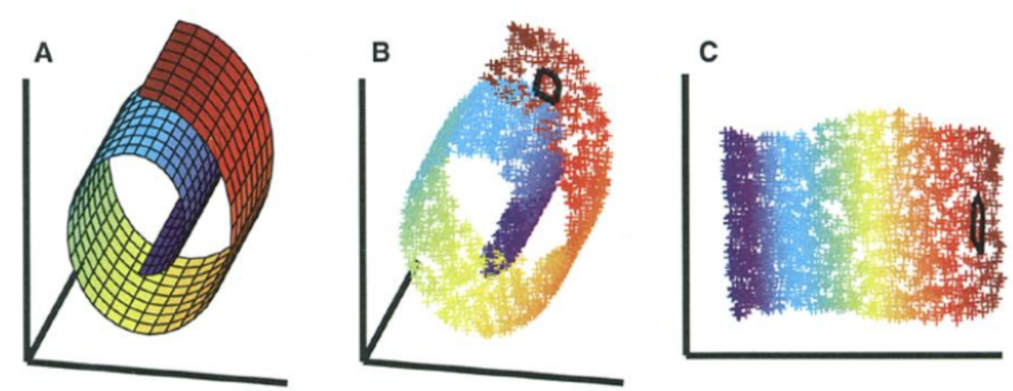


Fig. 1. The problem of nonlinear dimensionality reduction, as illustrated (*10*) for three-dimensional data (**B**) sampled from a two-dimensional manifold (**A**). An unsupervised learning algorithm must discover the global internal coordinates of the manifold without signals that explicitly indicate how the data should be embedded in two dimensions. The color coding illustrates the neighborhood-preserving mapping discovered by LLE; black outlines in (**B**) and (**C**) show the neighborhood of a single point. Unlike LLE, projections of the data by principal component analysis (PCA) (*28*) or classical MDS (*2*) map faraway data points to nearby points in the plane, failing to identify the underlying structure of the manifold. Note that mixture models for local dimensionality reduction (*29*), which cluster the data and perform PCA within each cluster, do not address the problem considered here: namely, how to map high-dimensional data into a single global coordinate system of lower dimensionality.

[1]Gatsby Computational Neuroscience Unit, University College London, 17 Queen Square, London WC1N 3AR, UK. [2]AT&T Lab—Research, 180 Park Avenue, Florham Park, NJ 07932, USA.

E-mail: roweis@gatsby.ucl.ac.uk (S.T.R.); lsaul@research.att.com (L.K.S.)

1. Find K nearest neighbors of each vector, X_i, in R^D as measured by Euclidean distance.

2. Compute the weights W_{ij} that best reconstruct X_i from its neighbors.

$$X_i \approx \sum_j W_{ij} X_j$$

3. Compute vectors Y_i in R^d reconstructed by the weights W_{ij}. Solve for all Y_i simultaneously.

$$Y_i \approx \sum_j W_{ij} Y_j$$

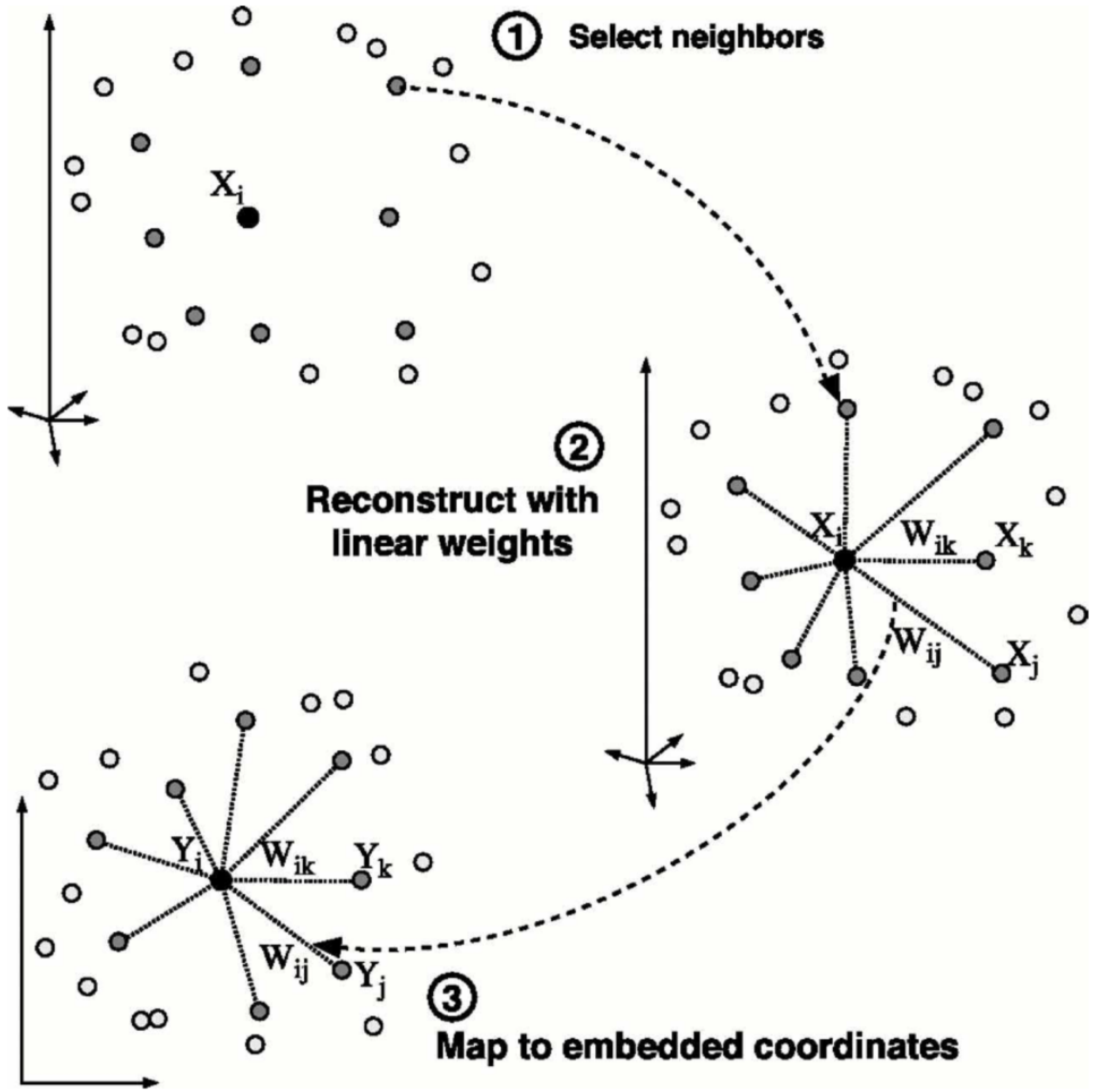


Figure from Roweis and Saul, 2001

- To compute the N x N weight matrix W we want to minimize the following cost function:

$$\varepsilon(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

where $W_{ij} = 0$ if X_j is not one of the K nearest neighbors of X_i and where the rows of W sum to 1

$$\sum_j W_{ij} = 1 \longrightarrow$$

N

W = (**.5 .2 .3 … 0 0 0**
… W sparse
…
…) N

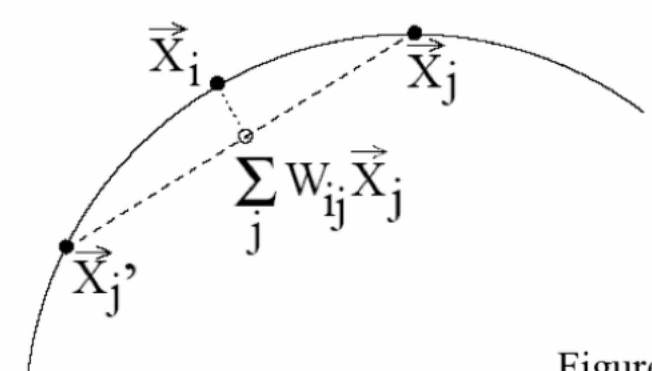


Figure from Roweis and Saul, 2003

Solving for one row of W

- Consider a particular data point $X_i = z$ with K nearest neighbors $X_j = n_j$ and reconstruction weights $W_{ij} = w_j$ that sum to one. Then,

$$\varepsilon = \left| z - \sum_j w_j n_j \right|^2$$

$$= \left| \sum_j w_j (z - n_j) \right|^2 \qquad \text{since } \sum_j w_j = 1$$

$$= \sum_j \sum_k w_j w_k C_{jk} \qquad \text{where } C_{jk} = (z - n_j)\cdot(z - n_k),$$

the local covariance matrix

- Now using Lagrange multipliers to enforce the sum to one constraint on the w_j, the optimal weights are given by

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{l,m} C_{lm}^{-1}}$$

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{l,m} C_{lm}^{-1}}$$

- Inversion of local covariance matrix can be avoided by solving the linear system of equations below and rescaling so the weights sum to one.

$$\sum_j C_{jk} w_k = 1$$

- Note: If the covariance matrix is singular or nearly singular regularization techniques must be used to solve this problem (this typically arises if K > D).

Computing Embedded Vectors Y_i

- Now that we have our weight matrix W, we would like to compute each of our embedding vectors Y_i. Minimize the following cost functions for fixed weights W_{ij}

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$

- To make the problem well posed we add two constraints: (1) centered at the origin and (2) unit covariance:

$$\sum_i Y_i = 0 \qquad\qquad \frac{1}{N} \sum_i Y_i Y_i^T = I$$

- The first constraint removes the degree of freedom that Y be translated by a constant amount. The second expresses an assumption that reconstruction errors for different coordinates in the embedding space should be measured on the same scale.

Solving for matrix Y

- Let Y be the matrix that contains Y_i as each of it's columns

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2$$
$$= \left|(I - W)Y\right|^2$$
$$= Y^T M Y$$

Where $M = (I - W)^T (I - W)$ is N x N

- Using Lagrange multipliers and setting the derivative to zero gives

$$(M - \Lambda)Y^T = 0$$

- Λ here is the diagonal Lagrange multiplier matrix. This is an eigenvalue problem where all eigenvectors of M are solutions. The eigenvectors with the smallest eigenvalues minimize our cost. We discard the first (i.e. smallest) eigenvector which corresponds to the mean of Y to enforce constraint (1). The next d eigenvectors then give the Y that minimizes our cost subject to the constraints (see K. Fan for more information on the proof).

Laplacian Eigenmaps for Dimensionality Reduction and Data Representation

Mikhail Belkin
misha@math.uchicago.edu
Department of Mathematics, University of Chicago, Chicago, IL 60637, U.S.A.

Partha Niyogi
niyogi@cs.uchicago.edu
Department of Computer Science and Statistics, University of Chicago, Chicago, IL 60637 U.S.A.

One of the central problems in machine learning and pattern recognition is to develop appropriate representations for complex data. We consider the problem of constructing a representation for data lying on a low-dimensional manifold embedded in a high-dimensional space. Drawing on the correspondence between the graph Laplacian, the Laplace Beltrami operator on the manifold, and the connections to the heat equation, we propose a geometrically motivated algorithm for representing the high-dimensional data. The algorithm provides a computationally efficient approach to nonlinear dimensionality reduction that has locality-preserving properties and a natural connection to clustering. Some potential applications and illustrative examples are discussed.

Laplacian Eigenmap

The graph Laplacian eigenmap algorithm [2] also incorporates directed or undirected graph structure describing the local neighborhood relations between data points. As in Isomap, these neighbor relations can be defined in terms of symmetric nearest neighbors or a small distance criterion. The neighborhood relations are summarized by the adjacency matrix W where $W_{ij} > 0$ if the ith and jth data points are neighbors ($i \sim j$), assumed to be symmetric, otherwise $W_{ij} = 0$. The non-zero weights in W can be chosen from $\{0, 1\}$, or according to $W_{ij} = e^{-|x_i - x_j|^2/2\sigma^2}$ (a Gaussian kernel) where σ is an adjustable parameter. The generalized graph Laplacian L is defined in terms of the adjacency matrix W as:

$$L_{ij} := \begin{cases} d_i, & \text{if } i = j, \\ -W_{ij}, & \text{if } i \sim j, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where $d_i = \sum_{j \sim i} W_{ij}$ is the degree of the ith vertex. The normalized graph Laplacian $\mathcal{L}$ is a symmetric matrix related to L by $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ with the diagonal matrix $D_{ij} = \delta_{ij} d_i$. We assume that the graph is connected, so that L will have a single zero eigenvalue associated with the uniform vector e.

Belkin and Niyogi [2] motivate the role of the graph Laplacian for dimensionality reduction by showing that a plausible cost for a one-dimensional embedding of the nodes of the graph $\psi : V \mapsto \mathcal{R}$ is given by:

$$\boldsymbol{\psi}^{\boldsymbol{T}} L \boldsymbol{\psi} = \frac{1}{2} \sum_{i,j} (\psi_i - \psi_j)^2 W_{ij} \tag{10}$$

which also shows that L is positive definite. Minimizing the quadratic form (10) involves finding the eigenvectors with the smallest eigenvalues of either the graph Laplacian L or $\mathcal{L}$, depending upon the constraints used in the optimization.

Matlab toolbox

Matlab Toolbox for Dimensionality Reduction (v0.8.1 - March 2013)

The Matlab Toolbox for Dimensionality Reduction contains Matlab implementations of 34 techniques for dimensionality reduction and metric learning. A large number of implementations was developed from scratch, whereas other implementations are improved versions of software that was already available on the Web. The implementations in the toolbox are conservative in their use of memory. The toolbox is available for download here.

Currently, the Matlab Toolbox for Dimensionality Reduction contains the following techniques:

- Principal Component Analysis (PCA)
- Probabilistic PCA
- Factor Analysis (FA)
- Classical multidimensional scaling (MDS)
- Sammon mapping
- Linear Discriminant Analysis (LDA)
- Isomap
- Landmark Isomap
- Local Linear Embedding (LLE)
- Laplacian Eigenmaps
- Hessian LLE
- Local Tangent Space Alignment (LTSA)
- Conformal Eigenmaps **(extension of LLE)**