Laplacian Eigenmaps for Dimensionality Reduction and Data Representation

M. Belkin and P. Niyogi,

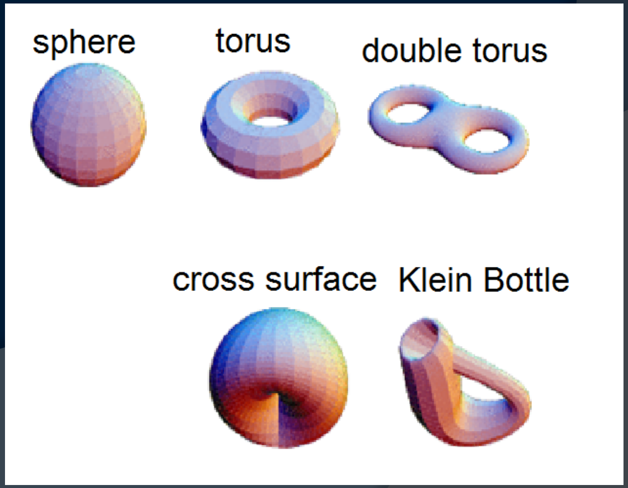
Neural Computation, pp. 1373–1396, 2003

Outline

- Introduction
- Algorithm
- Experimental Results
- Applications
- Conclusions

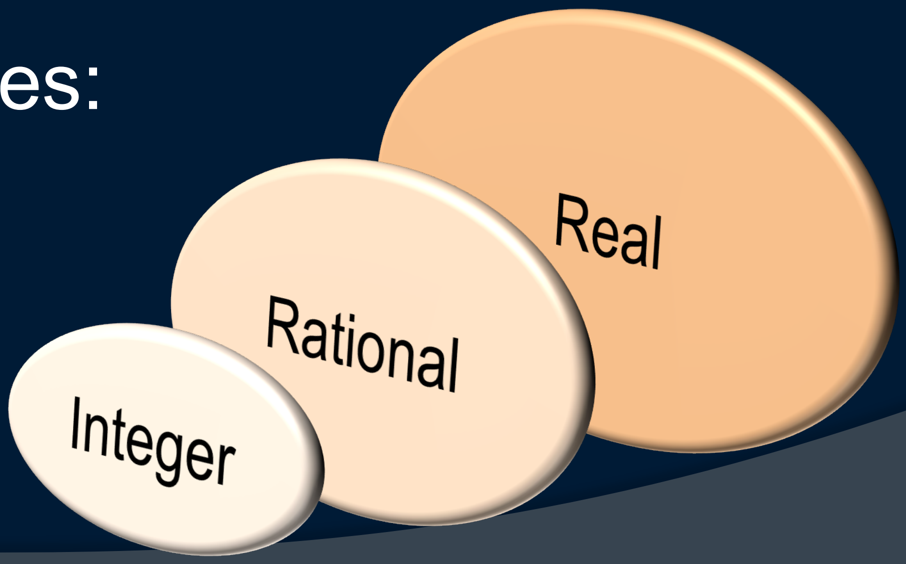
Manifold

- A manifold is a topological space which is locally Euclidean. In general, any object which is nearly "flat" on small scales is a manifold.
- Examples of 1-D manifolds include a line, a circle, and two separate circles.



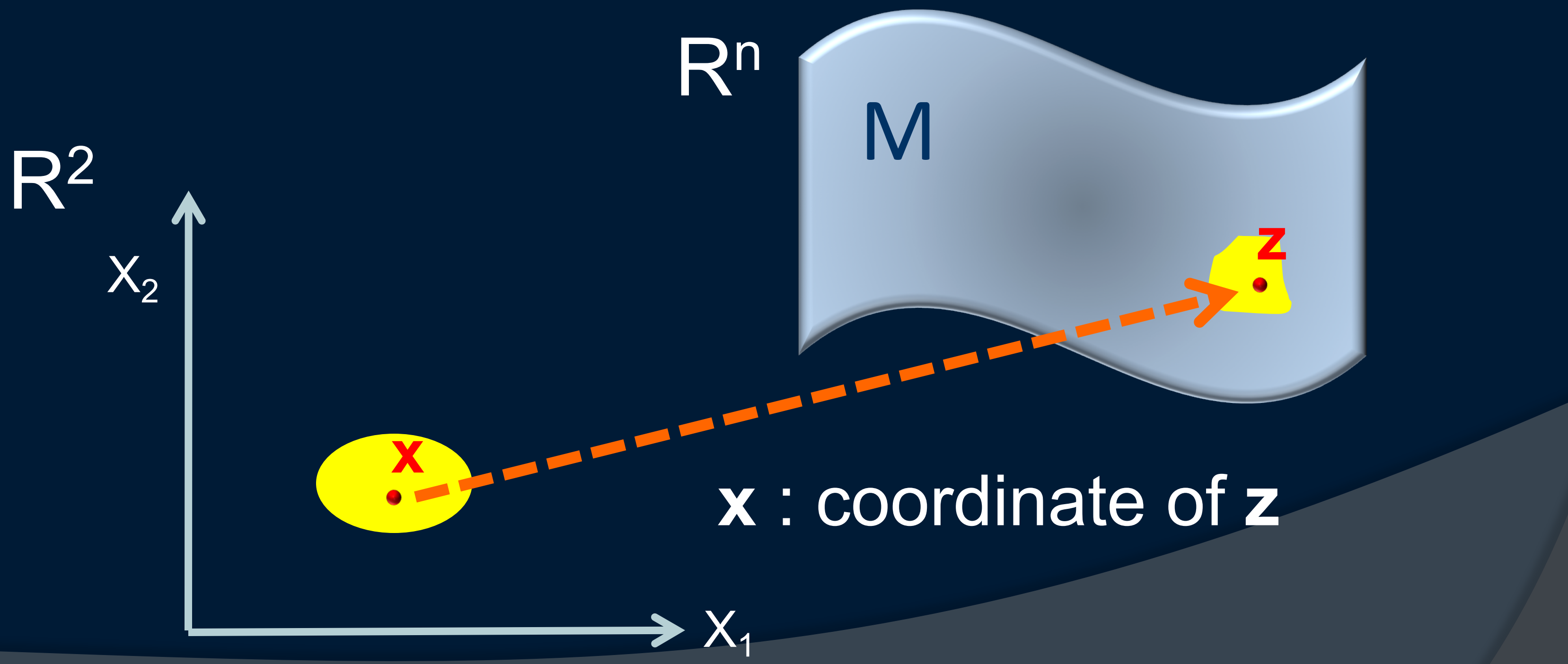
Embedding

- An embedding is a representation of a topological object, manifold, graph, field, etc. in a certain space in such a way that its connectivity or algebraic properties are preserved.
- Examples:



Manifold and Dimensionality Reduction (1)

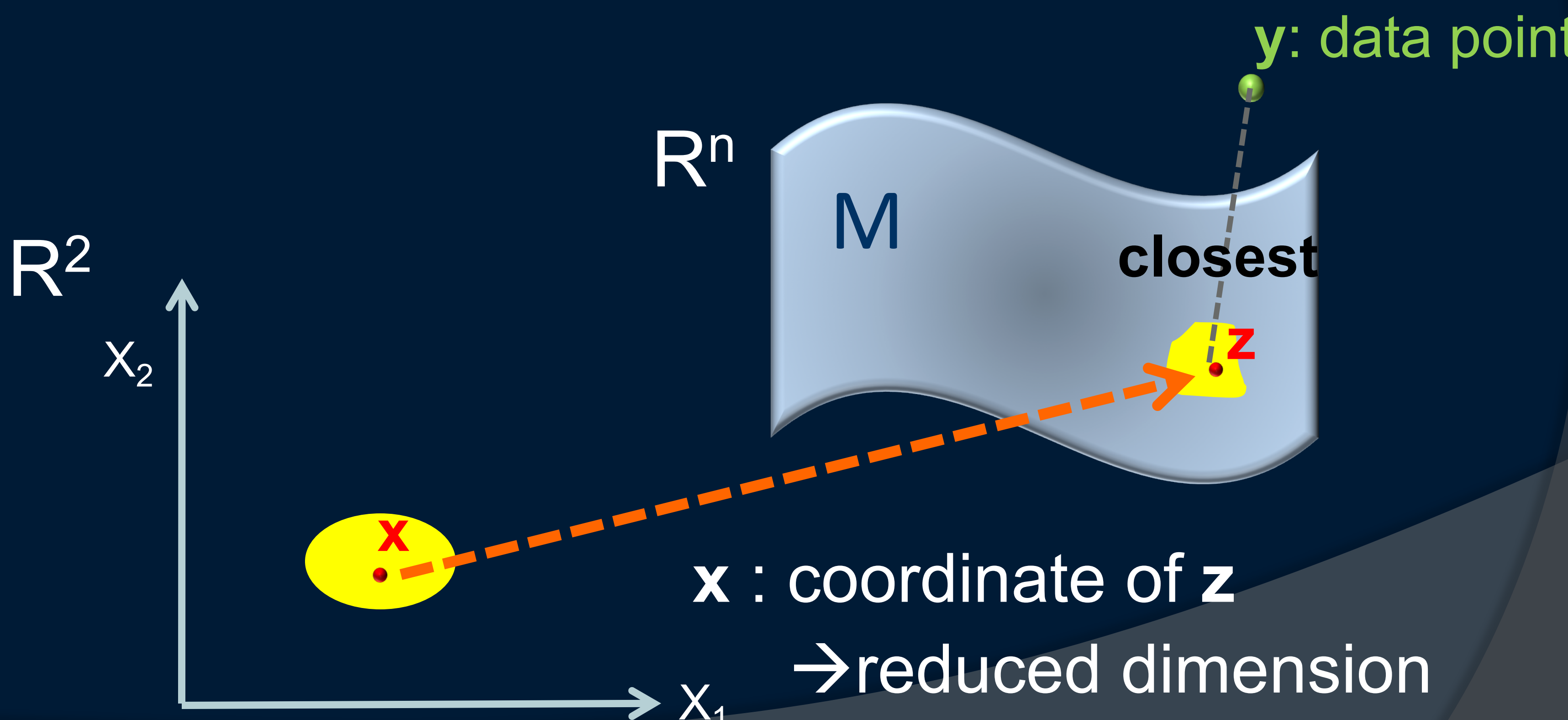
- Manifold: generalized “subspace” in R^n
- Points in a local region on a manifold can be indexed by a subset of R^k ($k \ll n$)



x : coordinate of **z**

Manifold and Dimensionality Reduction (2)

- If there is a global indexing scheme for M…



x : coordinate of **z**

→reduced dimension representation of **y**

Introduction (1)

- We consider the problem of constructing a representation for data lying on a low dimensional manifold embedded in a high dimensional space

Introduction (2)

- Linear methods
  - PCA (Principal Component Analysis) *1901*
  - MDS (Multidimensional Scaling) *1952*
- Nonlinear methods
  - ISOMAP *2000*
  - LLE (Locally Linear Embedding) *2000*
  - LE (Laplacian Eigenmap) *2003*

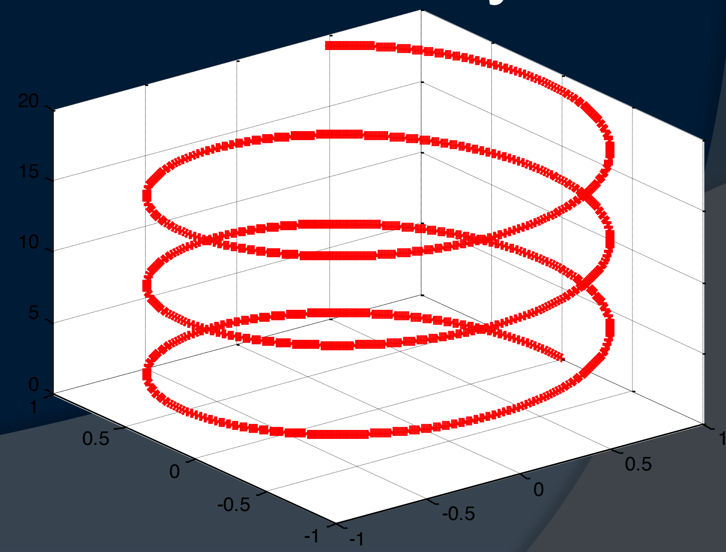
Linear Methods (1)

- What are “linear” methods?

  - Assume that data is a linear function of the parameters

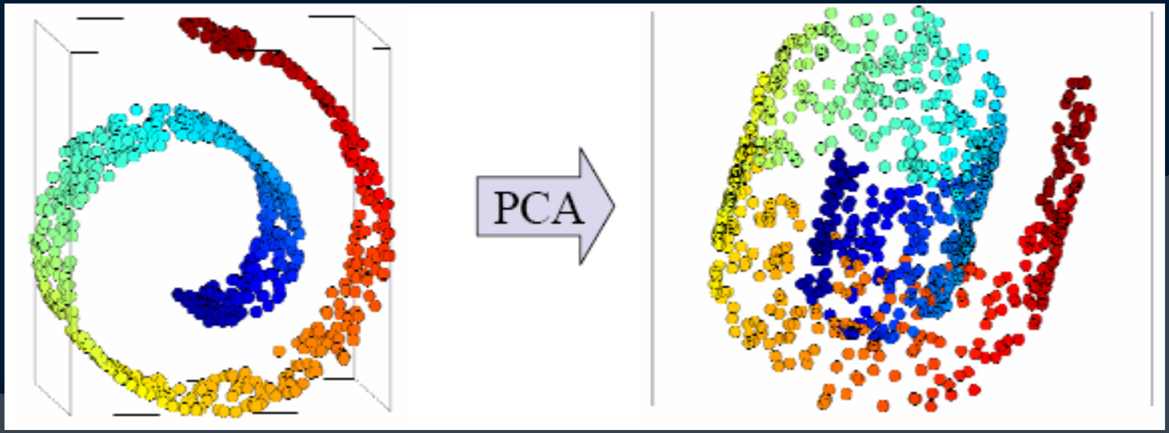
- Deficiencies of linear methods

  - Data may not be best summarized by linear combination of features



Linear Methods (2)

- PCA: rotate data so that principal axes lie in direction of maximum variance
- MDS: find coordinates that best preserve pairwise distances
- Linear methods do nothing more than "globally transform" (rotate/translate/scale) data.

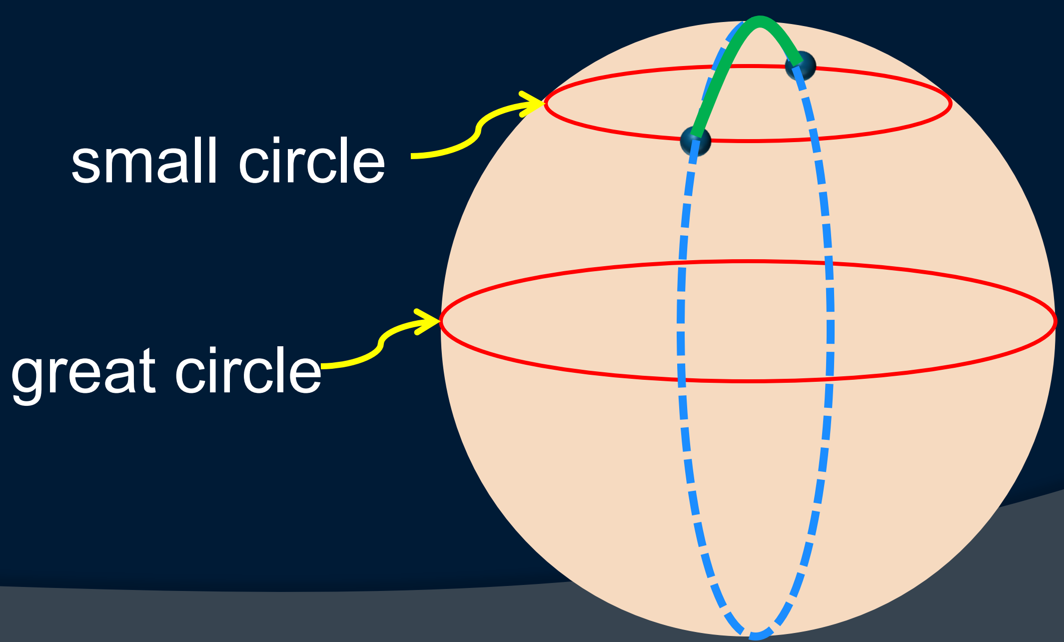


ISOMAP, LLE and Laplacian Eigenmap

- The graph-based algorithms have 3 basic steps.
  - 1. Find K nearest neighbors.
  - 2. Estimate local properties of manifold by looking at neighborhoods found in Step 1.
  - 3. Find a global embedding that preserves the properties found in Step 2.

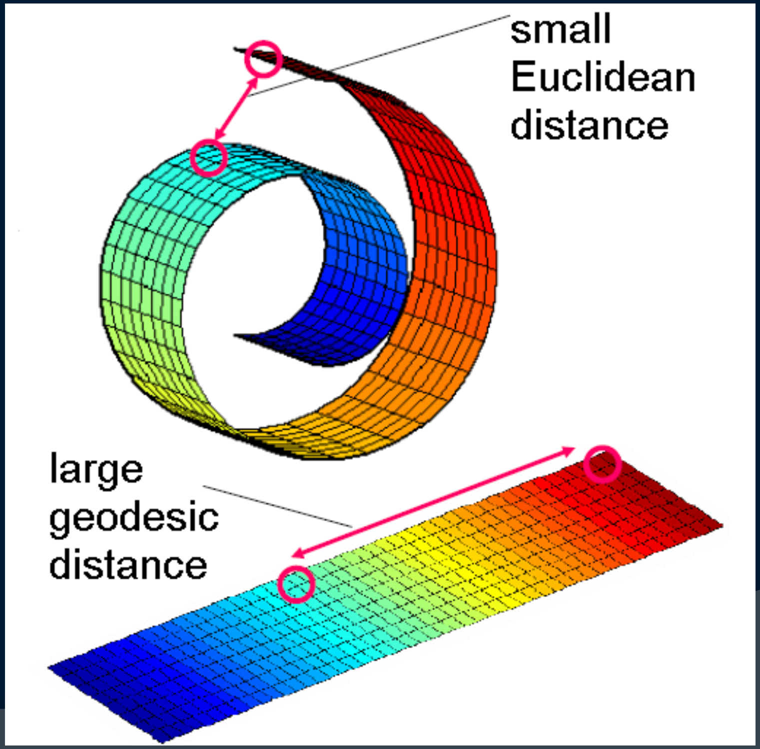
Geodesic Distance (1)

- Geodesic: the shortest curve on a manifold that connects two points on the manifold
  - Example: on a sphere, geodesics are great circles
- Geodesic distance: length of the geodesic



Geodesic Distance (2)

- Euclidean distance needs not be a good measure between two points on a manifold
  - Length of geodesic is more appropriate



ISOMAP

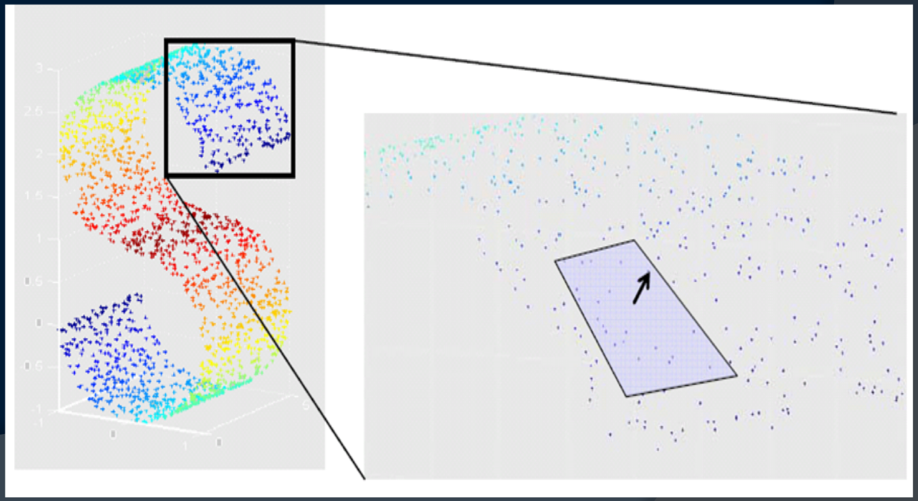
- Comes from Isometric feature mapping

  Step1: Take a distance matrix $\{\gamma_{ij}\}$ as input

  Step2: Estimate geodesic distance between any two points by “a chain of short paths”

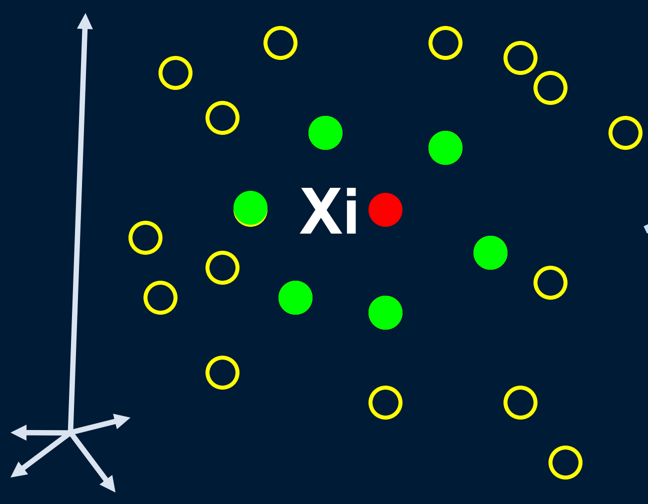
  → Approximate the geodesic distance by Euclidean distance

  Step3: Perform MDS

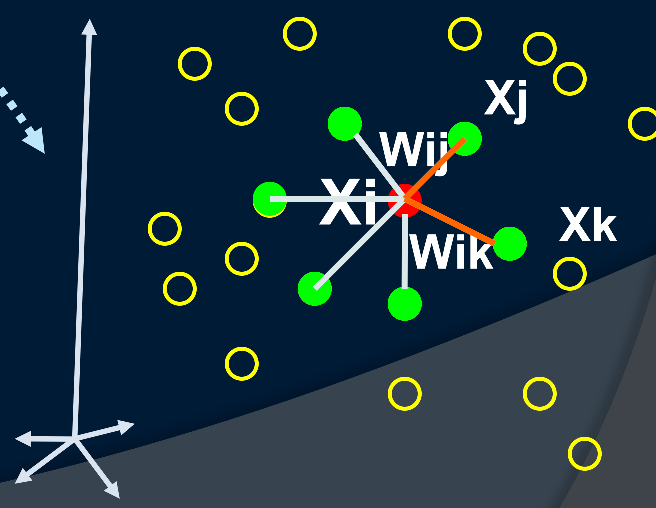


LLE (1)

- Assumption: manifold is approximately "linear" when viewed locally



1. select neighbors



2. reconstruct with linear weights

LLE (2)

- The geometrical property is best preserved if the error below is small

$$\Phi(\mathbf{X}) = \sum_{i=1}^{m} ||\mathbf{x}_i - \sum_{j=1}^{m} W_{ij}\mathbf{x}_j||^2$$

i.e. choose the best W to minimize the cost function

Linear reconstruction of x_i

Outline

- Introduction
- Algorithm
- Experimental Results
- Applications
- Conclusions

Some Aspects of the Algorithm

- It reflects the intrinsic geometric structure of the manifold
- The manifold is approximated by the adjacency graph computed from the data points
- The Laplace Beltrami operator is approximated by the weighted Laplacian of the adjacency graph

Laplace Beltrami Operator (1)

- The Laplace operator is a second order differential operator in the *n*-dimensional Euclidean space:

$$\Delta = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i{}^2}$$

- Laplace Beltrami operator:

  The Laplacian can be extended to functions defined on surfaces, or more generally, on Riemannian and pseudo-Riemannian manifolds.

Laplace Beltrami Operator (2)

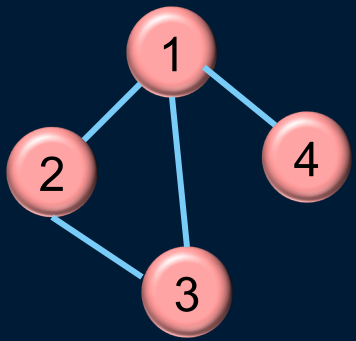
- We can justify that the eigenfunctions of the Laplace Beltrami operator have properties desirable for embedding… 

Lapalcian of a Graph (1)

- Let G(V,E) be a undirected graph without graph loops. The Laplacian of the graph is

$$L_{ij} = \begin{cases} d_{ij} & \text{if } i=j \text{ (degree of node i)} \\ -1 & \text{if } i\neq j \text{ and (i,j) belongs to E} \\ 0 & \text{otherwise} \end{cases}$$

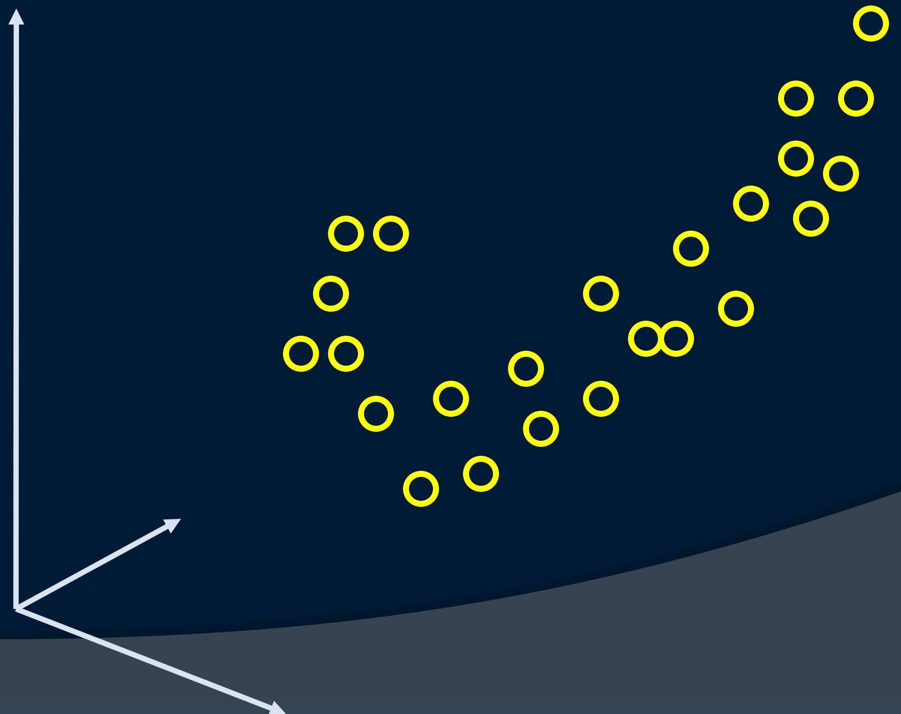
Lapalcian of a Graph (2)



$$L=\begin{bmatrix}3 & -1 & -1 & -1\\ -1 & 2 & -1 & 0\\ -1 & -1 & 2 & 0\\ -1 & 0 & 0 & 1\end{bmatrix}=\underbrace{\begin{bmatrix}3 & 0 & 0 & 0\\ 0 & 2 & 0 & 0\\ 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}}_{D}-\underbrace{\begin{bmatrix}0 & 1 & 1 & 1\\ 1 & 0 & 1 & 0\\ 1 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\end{bmatrix}}_{W\text{(weight matrix)}}$$

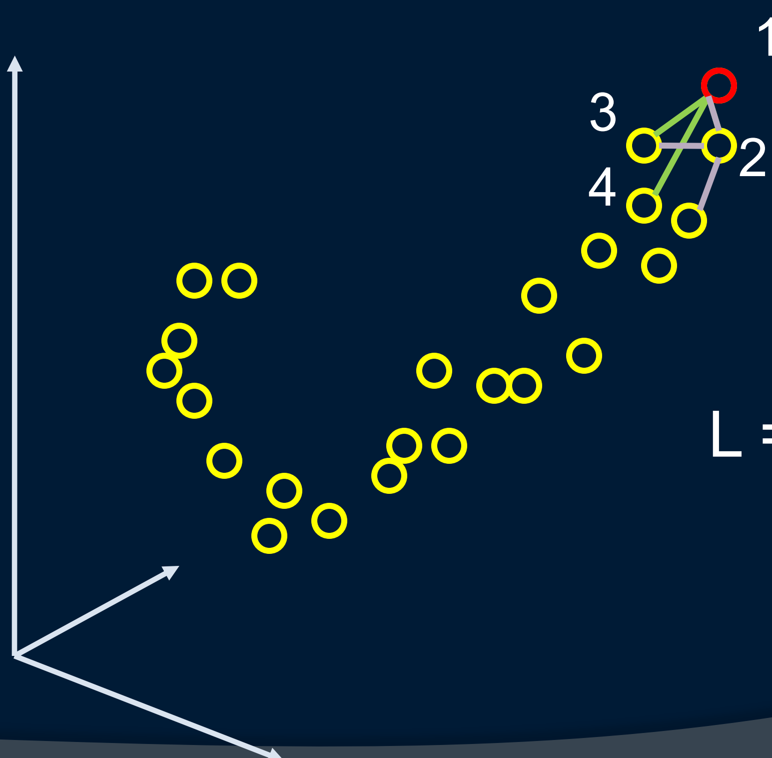
Laplacian Eigenmap (1)

- Consider that $\underline{\underline{x}}_1, \ldots, \underline{\underline{x}}_n \in \mathrm{M}$, and M is a manifold embedded in R^{ℓ}. Find $\underline{\underline{y}}_1, \ldots, \underline{\underline{y}}_n$ in R^m such that $\underline{\underline{y}}_i$ represents $\underline{\underline{x}}_i$ $(m \ll \ell)$



Laplacian Eigenmap (2)

- Construct the adjacency graph to approximate the manifold



$$L = \begin{bmatrix} 3 & -1 & & \\ -1 & 3 & & \\ -1 & -1 & & \\ -1 & 0 & & \\ 0 & -1 & & \\ \vdots & 0 & & \end{bmatrix} = D - W$$

$\vdots$

Laplacian Eigenmap (3)

- There are two variations for W (weight matrix)

  - simple-minded (1 if connected, 0 o.w.)

  - heat kernel (t is real)

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{t}}$$

Laplacian Eigenmap (4)

- Consider the problem of mapping the graph G to a *line* so that connected points stay as close together as possible
- To choose a good “map”, we have to minimize the objective function

$$\sum_{ij}(y_i - y_j)^2 W_{ij}$$

➡ $W_{ij}\uparrow$, $(y_i-y_j)\downarrow$

➡ $\underline{\mathbf{y}}^T L \underline{\mathbf{y}}$ where $\underline{\mathbf{y}} = [y_1 \ldots y_n]^T$

Laplacian Eigenmap (5)

- Therefore, this problem reduces to find $\boxed{\text{argmin } \underline{\mathbf{y}}^T L \underline{\mathbf{y}} \text{ subjects to } \underline{\mathbf{y}}^T D \underline{\mathbf{y}} = 1}$ (removes an arbitrary scaling factor in the embedding)
- The solution $\underline{\mathbf{y}}$ is the eigenvector corresponding to the minimum eigenvalue of the generalized eigenvalue problem

$$L\underline{\mathbf{y}} = \lambda D\underline{\mathbf{y}}$$

Laplacian Eigenmap (6)

- Now we consider the more general problem of embedding the graph into m-dimensional Euclidean space
- Let Y be such a n*m map

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{bmatrix} \Rightarrow \sum_{i,j} \left\| \underline{\underline{y}}^{(i)} - \underline{\underline{y}}^{(j)} \right\|^2 W_{ij} = tr(Y^T L Y)$$

$$where\ \underline{\underline{y}}^{(i)} = [y_{11} \quad y_{12} \quad \cdots \quad y_{1m}]^T$$

$$\underset{Y^T D Y = I}{\arg\min}\ tr(Y^T L Y)$$

Laplacian Eigenmap (7)

- To sum up:

  Step1: Construct adjacency graph

  Step2: Choosing the weights

  Step3: Eigenmaps $L\underline{y} = \lambda D\underline{y}$

  ➡ $L\underline{y}_0 = \lambda_0 D\underline{y}_0, \quad L\underline{y}_1 = \lambda_1 D\underline{y}_1 \ldots$

  $0 = \lambda_0 \leqq \lambda_1 \leqq \ldots \leqq \lambda_{n-1}$

  ➡ $\underline{\underline{x}}_i \rightarrow (\underline{y}_0(i), \underline{y}_1(i), \ldots, \underline{y}_m(i))$

Recall that we have n data points, so L and D is n×n and $\underline{y}$ is a n×1 vector

ISOMAP, LLE and Laplacian Eigenmap

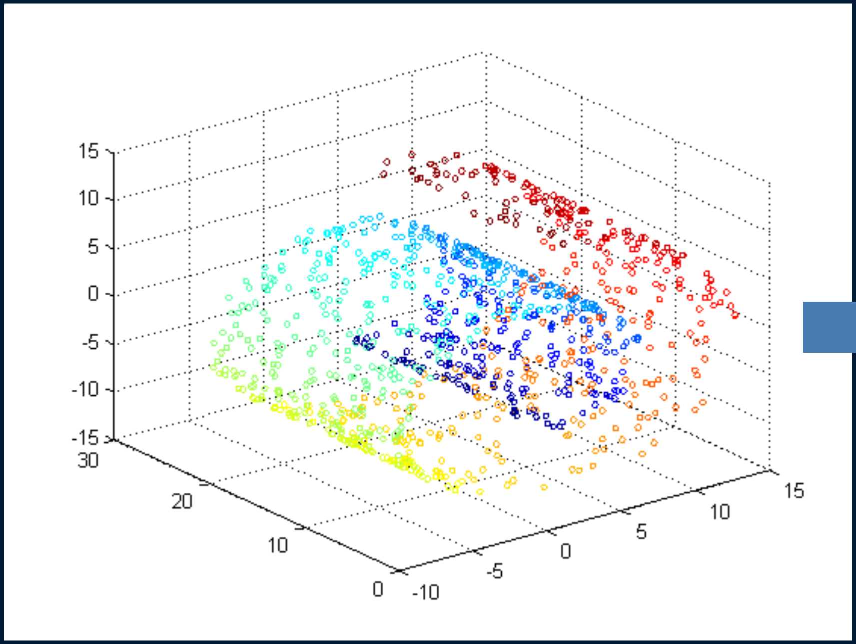
- The graph-based algorithms have 3 basic steps.
  - 1. Find K nearest neighbors.
  - 2. Estimate local properties of manifold by looking at neighborhoods found in Step 1.
  - 3. Find a global embedding that preserves the properties found in Step 2.

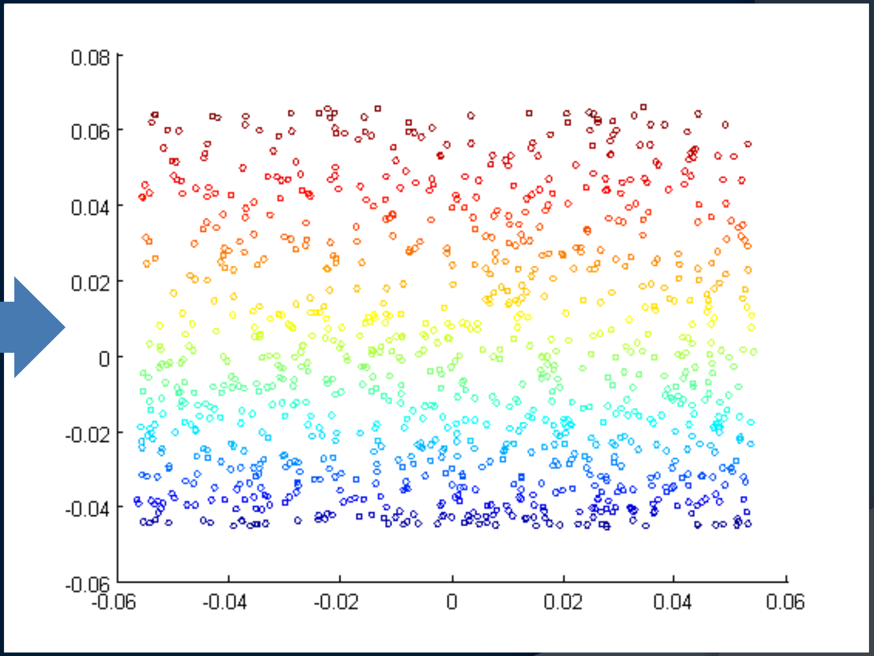
Outline

- Introduction
- Algorithm
- Experimental Results
- Applications
- Conclusions

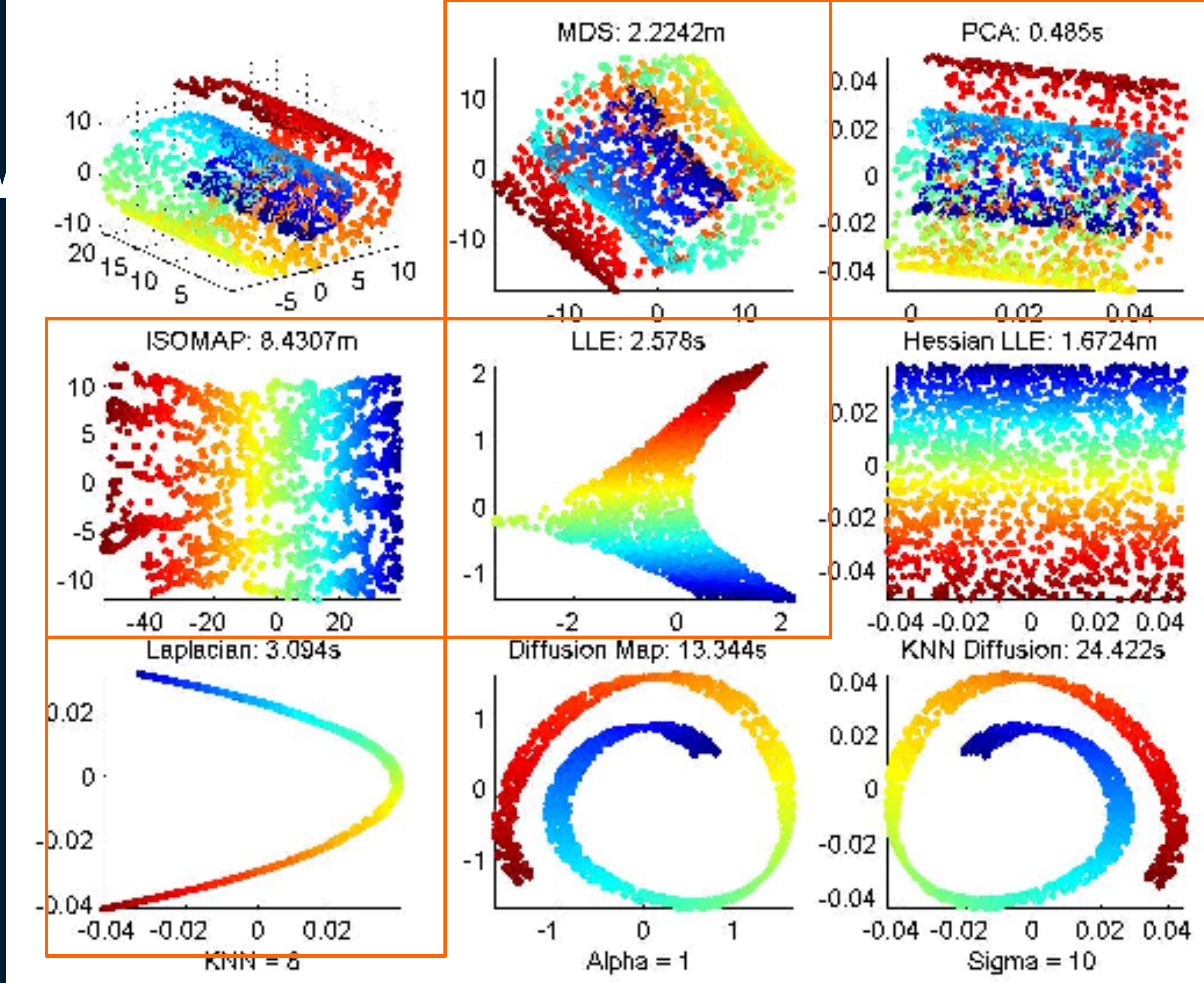
- The following material is from http://www.math.umn.edu/~wittman/mani/

Swiss Roll (1)


15
10
5
0
-5
-10
-15
30
20
10
0
-10
-5
0
5
10
15


0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.06
-0.04
-0.02
0
0.02
0.04
0.06

Sw



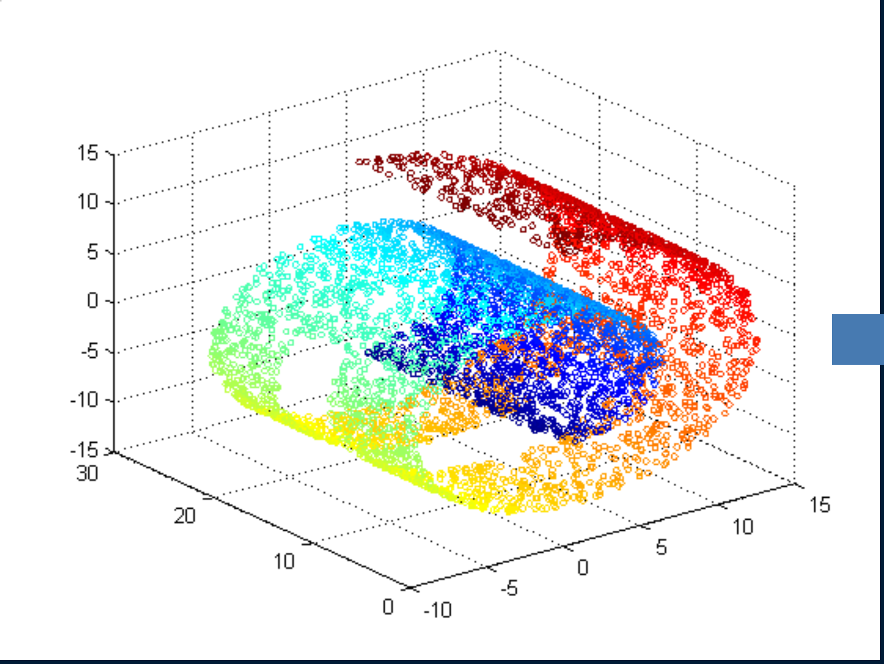
MDS is very slow, and ISOMAP is extremely slow.
MDS and PCA don’t can’t unroll Swiss Roll, use no manifold information.
LLE and Laplacian can’t handle this data.

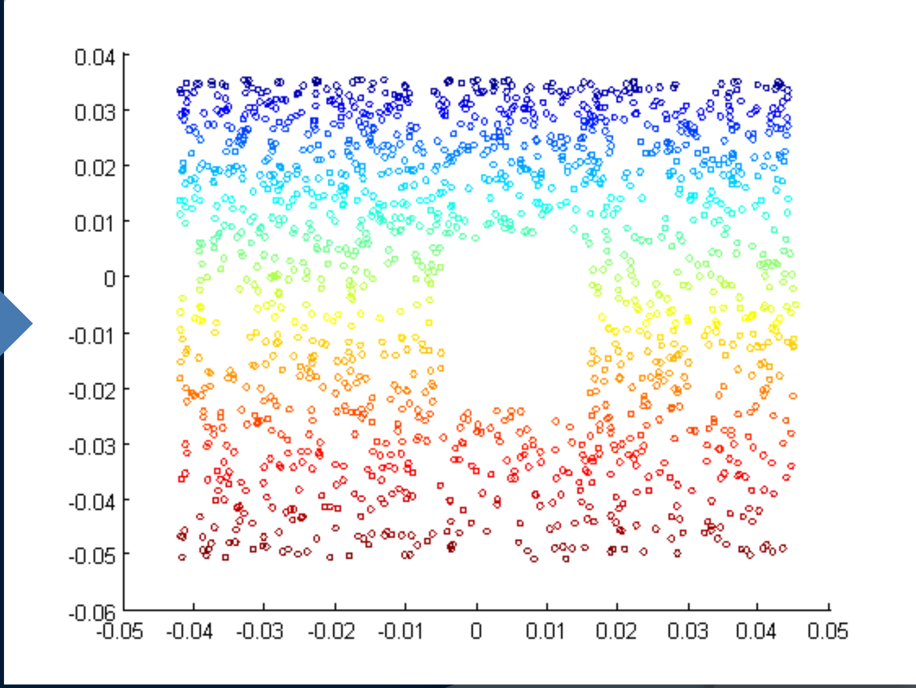
Swiss Roll (3)

- Isomap provides a isometric embedding that preserves global geodesic distances

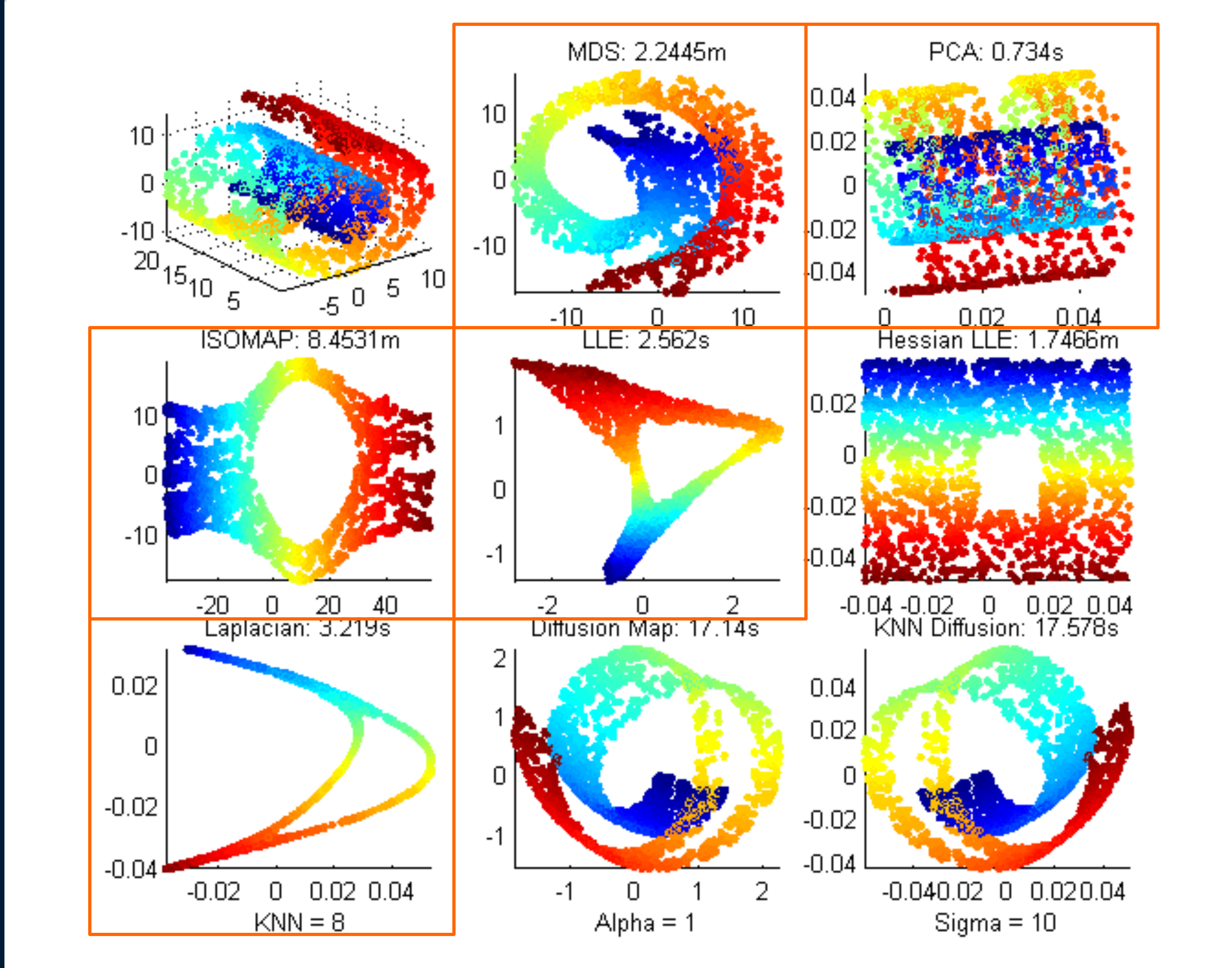
  → It works only when the surface is flat
- Laplacian eigenmap tries to preserve the geometric characteristics of the surface

Non-Convexity (1)





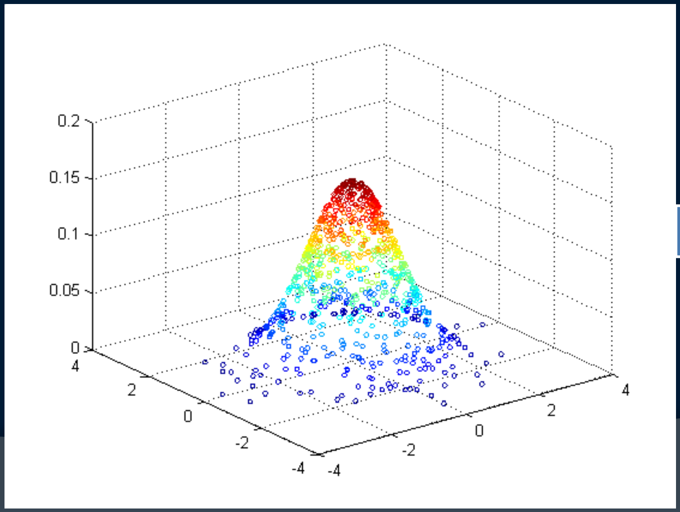
N

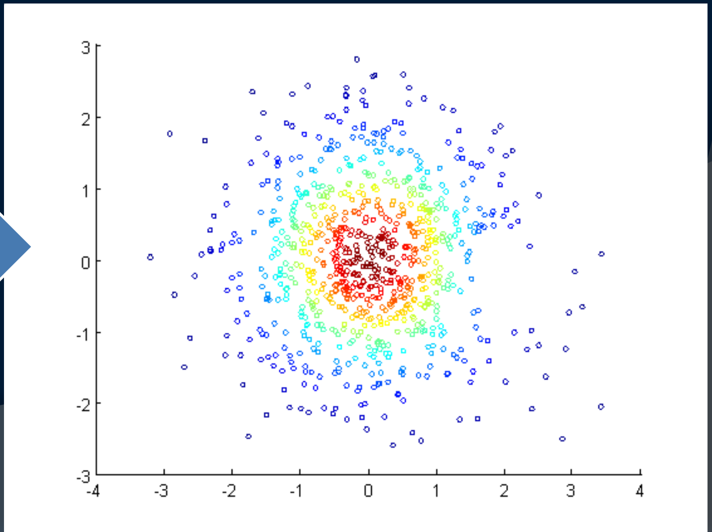


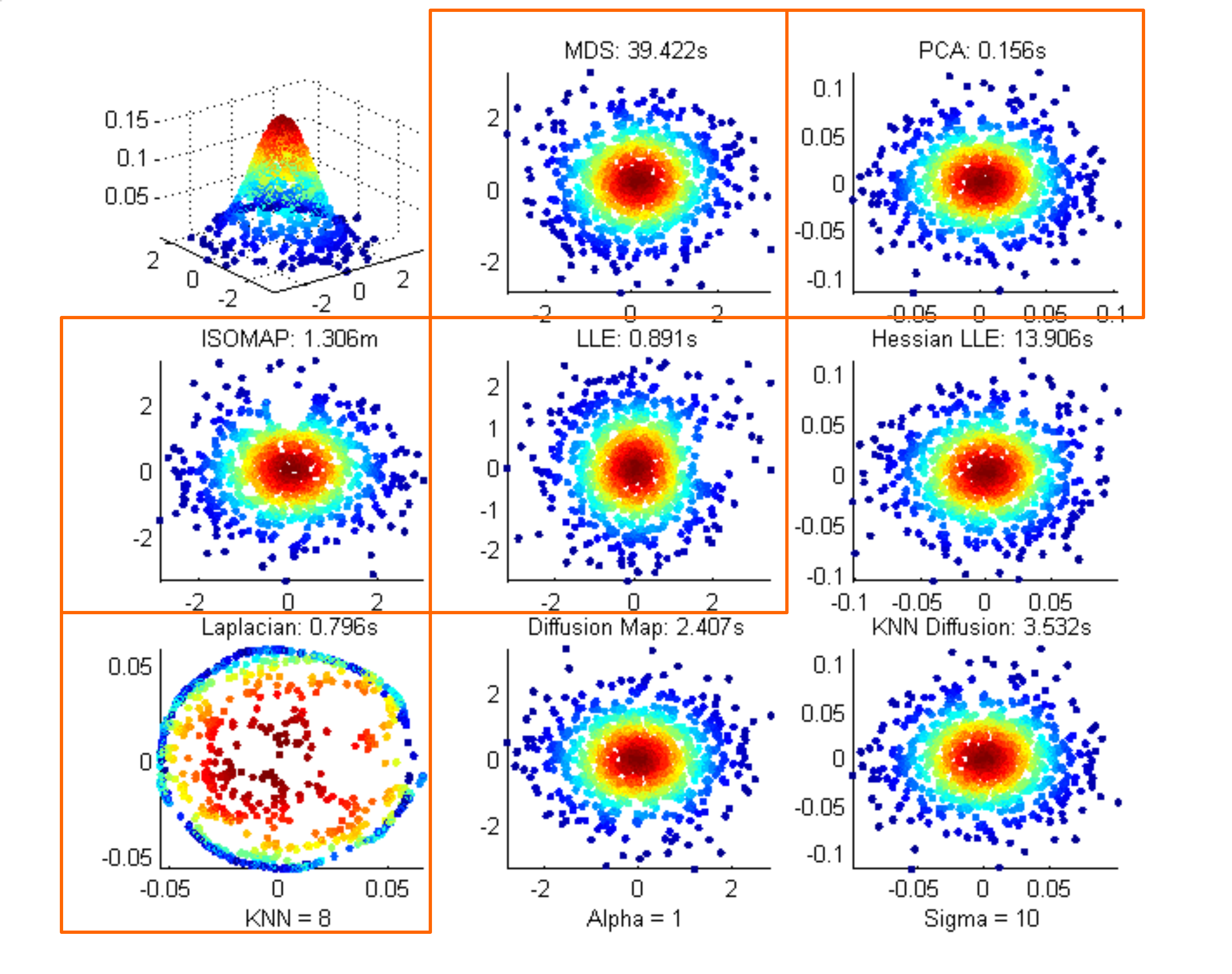
Only Hessian LLE can handle non-convexity.
ISOMAP, LLE, and Laplacian find the hole but the set is distorted.

Curvature & Non-uniform Sampling

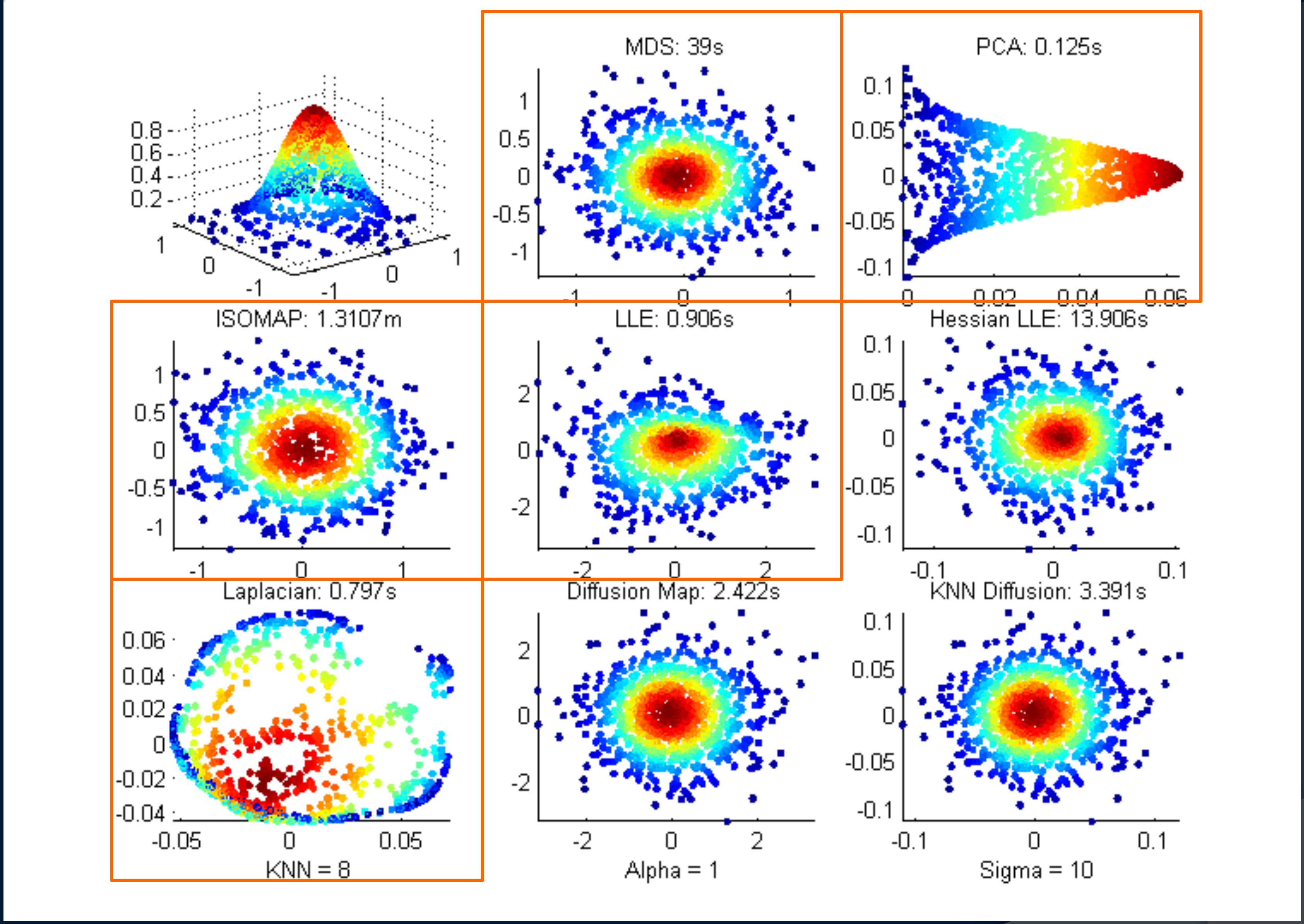
- Gaussian: We can randomly sample a Gaussian distribution.
- We increase the curvature by decreasing the standard deviation.
- Coloring on the z-axis, we should map to concentric circles



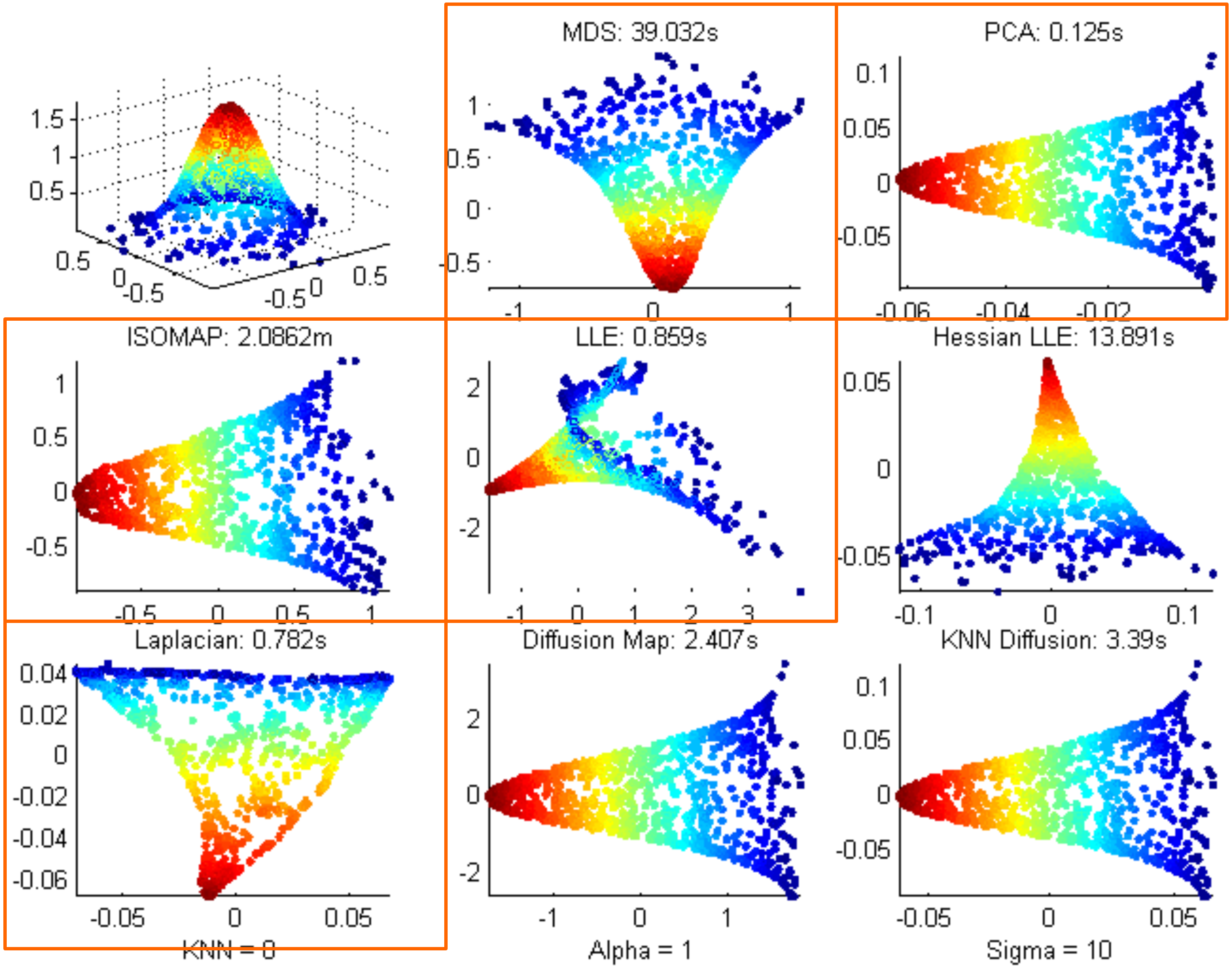




For std = 1 (low curvature), MDS and PCA can project accurately.
Laplacian Eigenmap cannot handle the change in sampling.



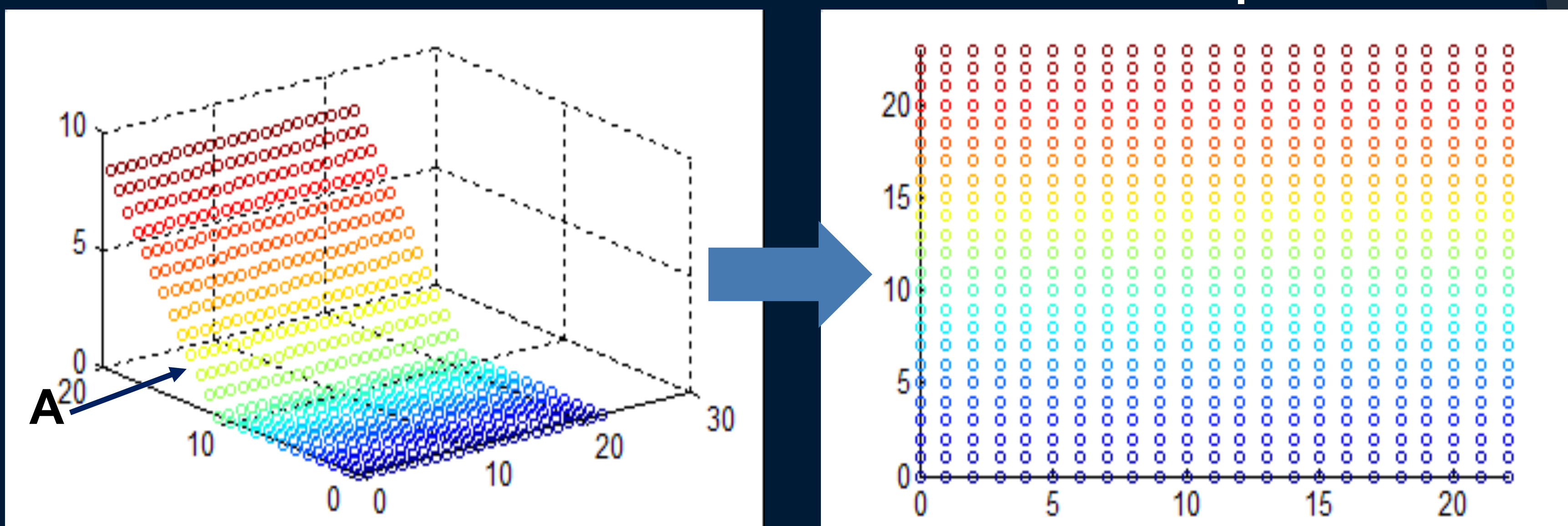
For std = 0.4 (higher curvature), PCA projects from the side rather than top-down. Laplacian looks even worse.

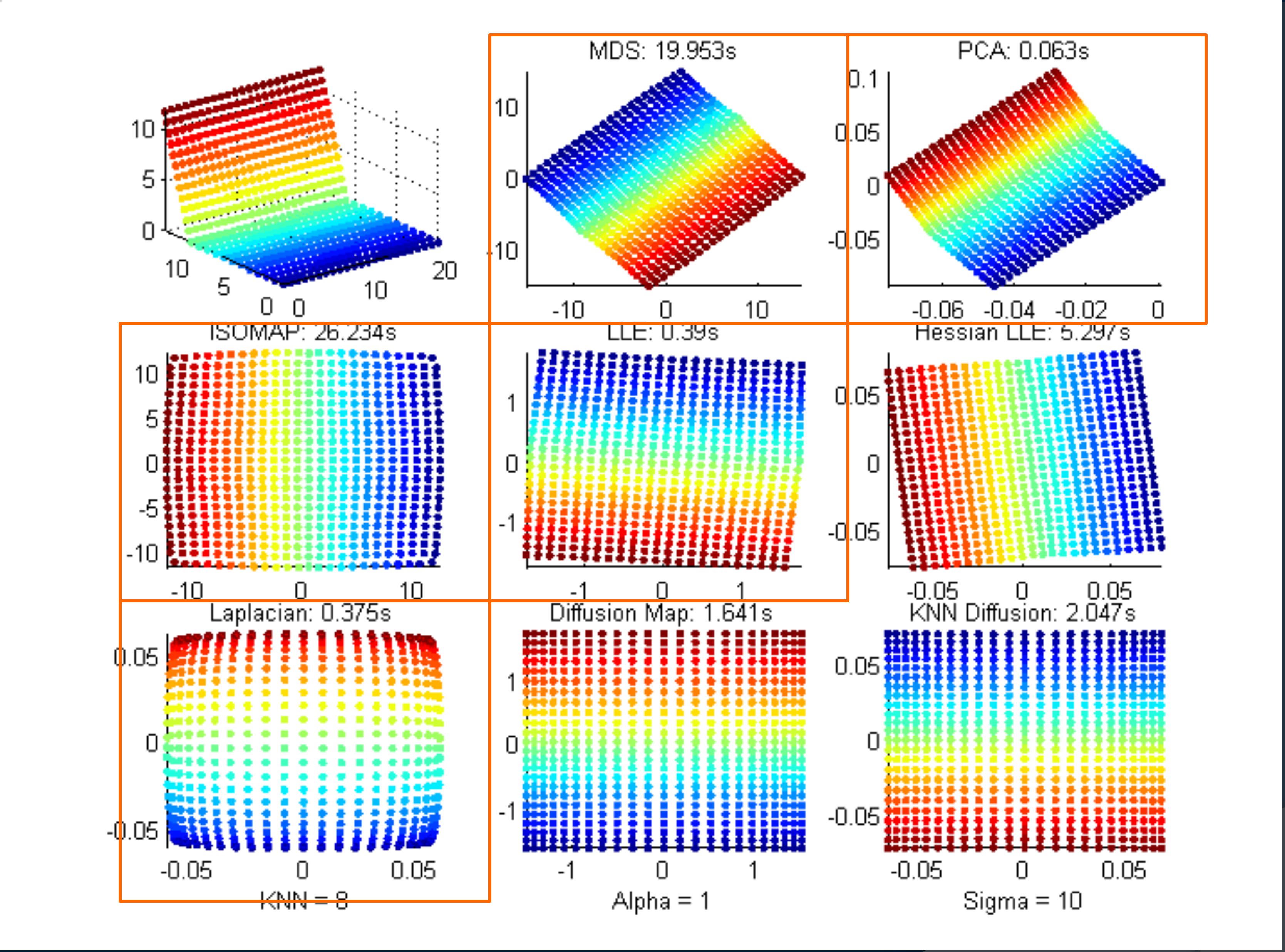


For std = 0.3 (high curvature), none of the methods can project correctly.

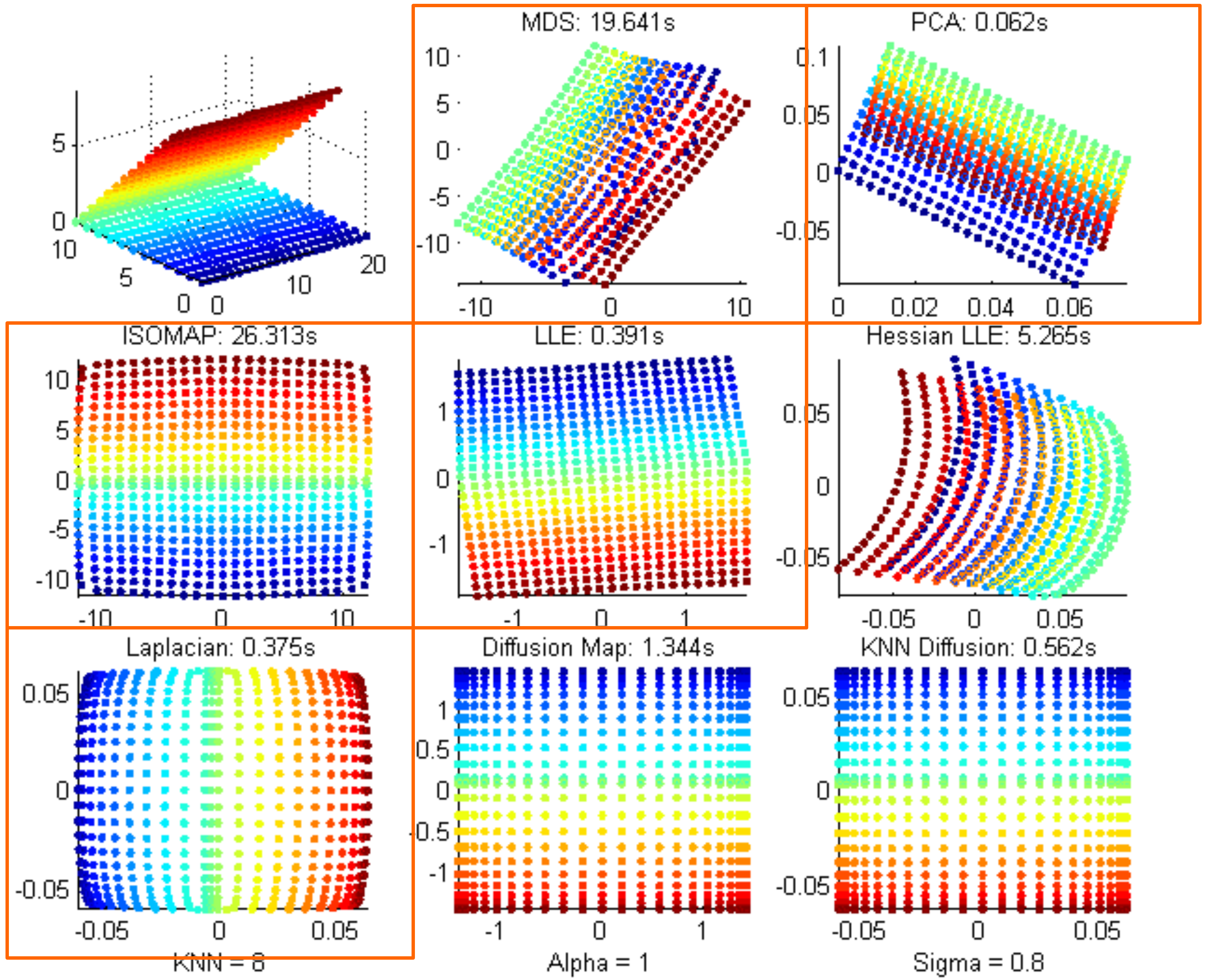
Corner

- Corner Planes: We bend a plane with a lift angle A.
- We want to bend it back down to a plane.





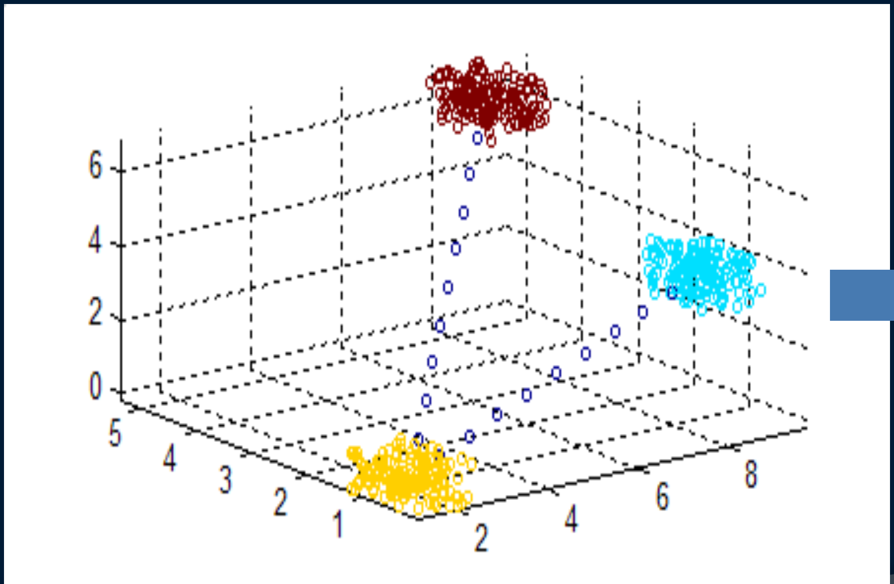
For angle A=75, we see some disortions in PCA and Laplacian.

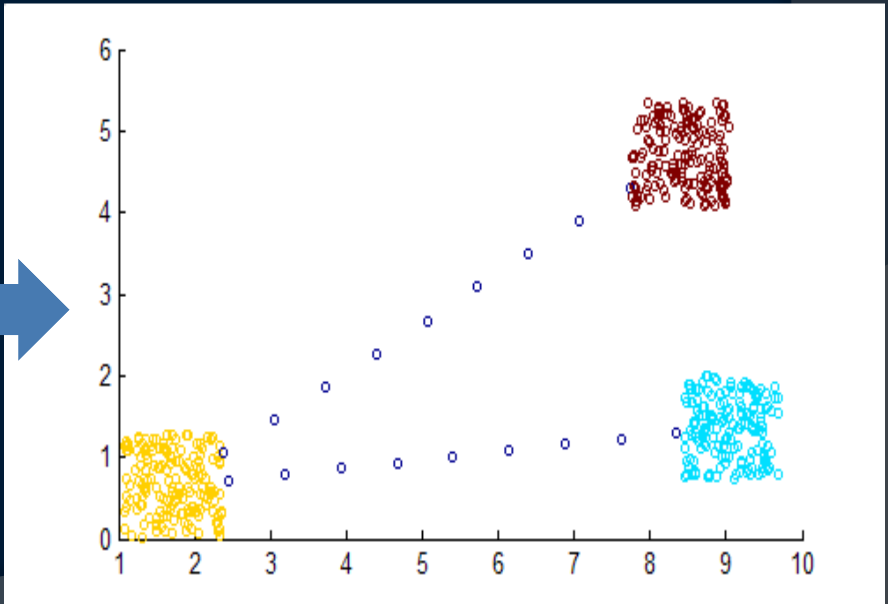


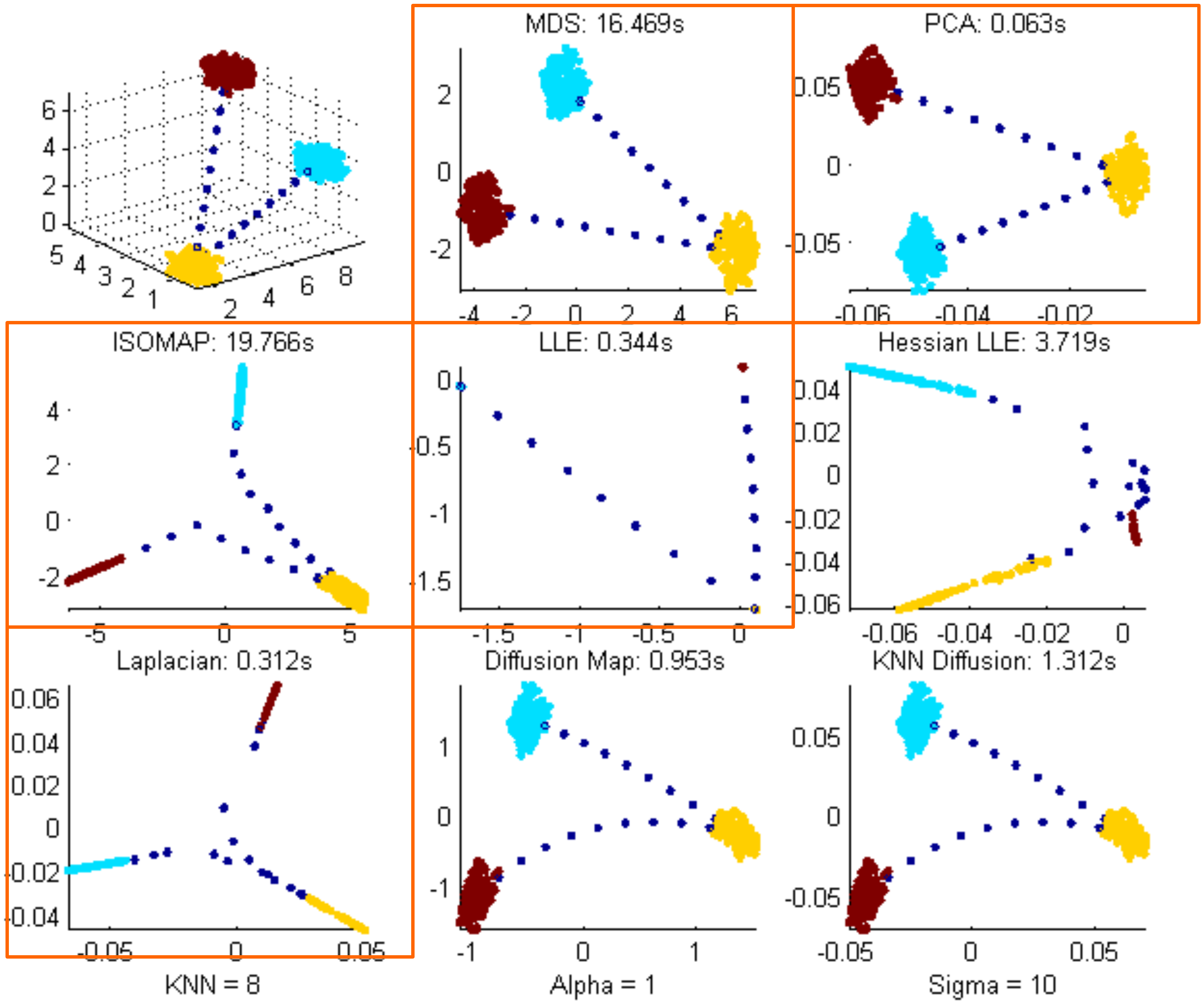
For A = 135, MDS, PCA, and Hessian LLE overwrite the data points.
Diffusion Maps work very well for Sigma < 1.
LLE handles corners surprisingly well.

Clustering

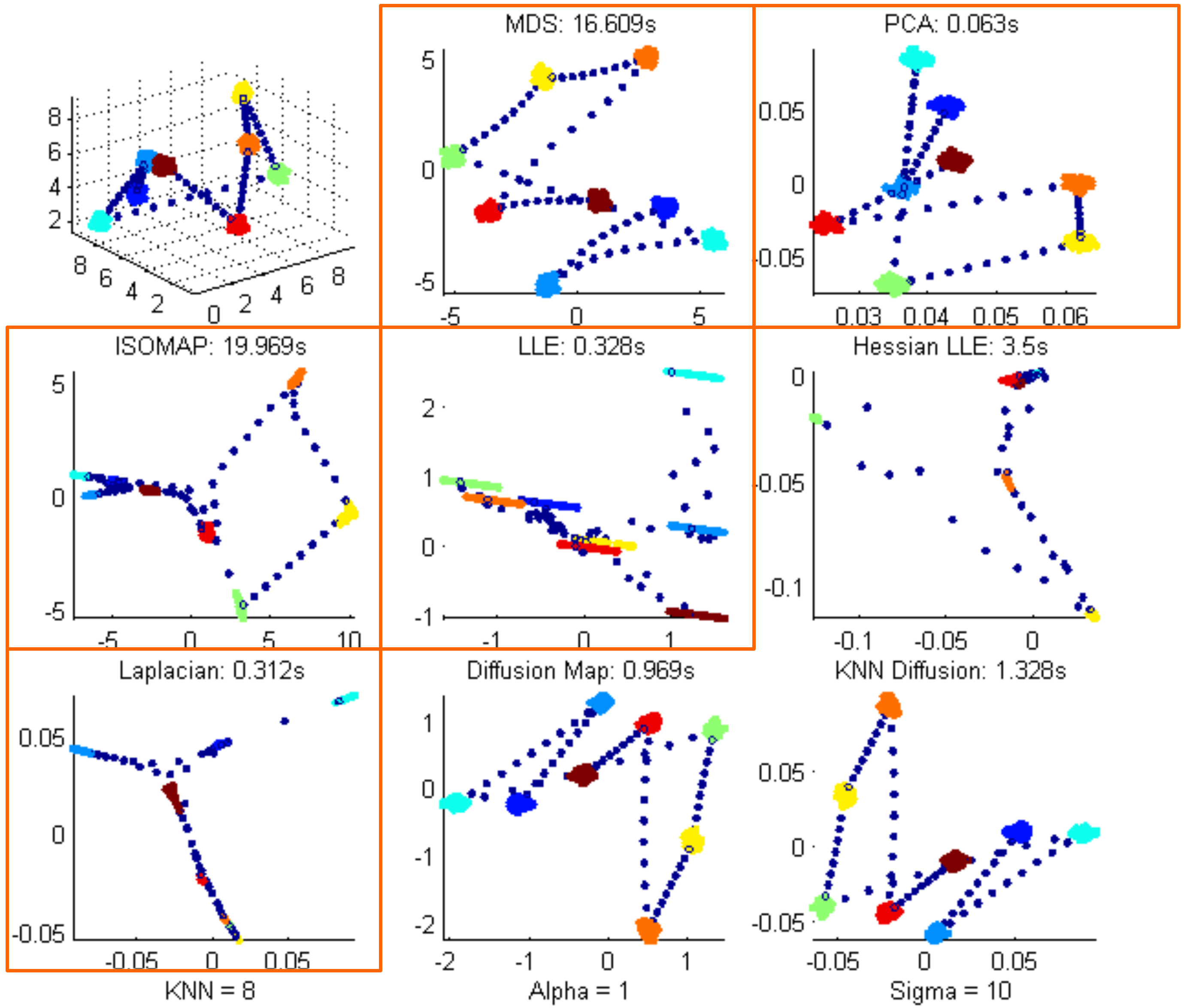
- 3D Clusters: Generate M non-overlapping clusters with random centers. Connect the clusters with a line.







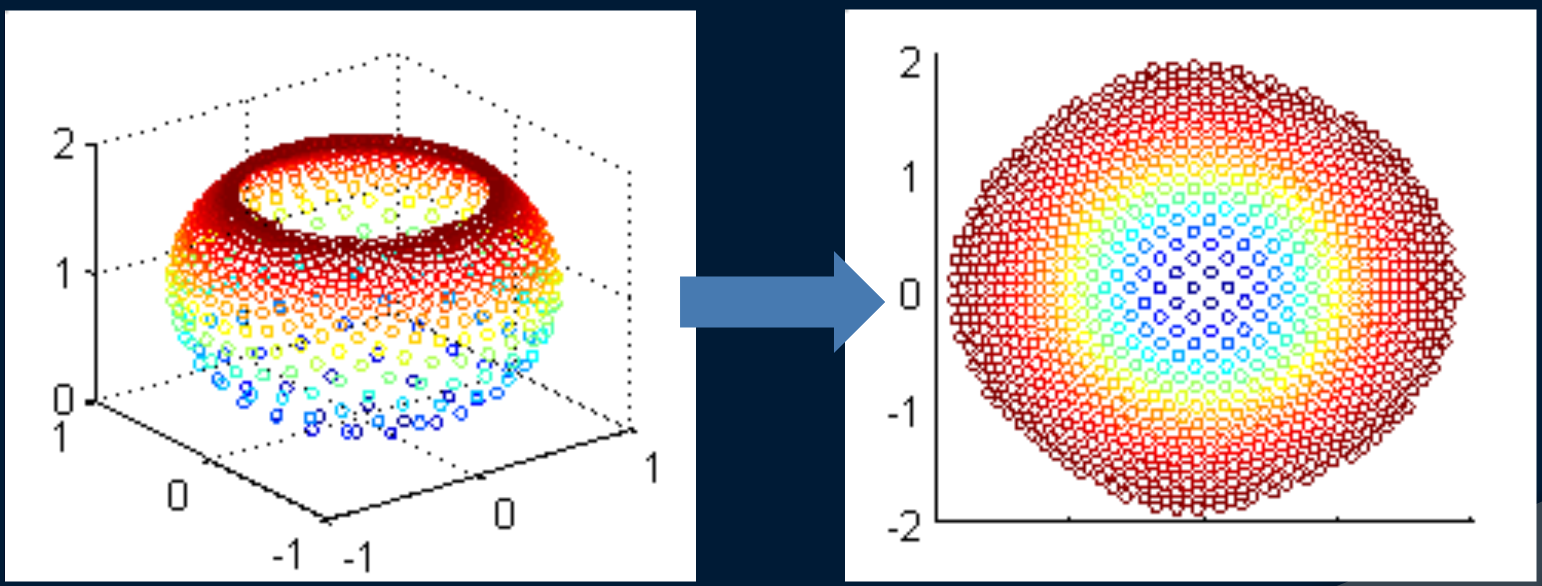
For M = 3 clusters, MDS and PCA can project correctly.
LLE compresses each cluster into a single point.

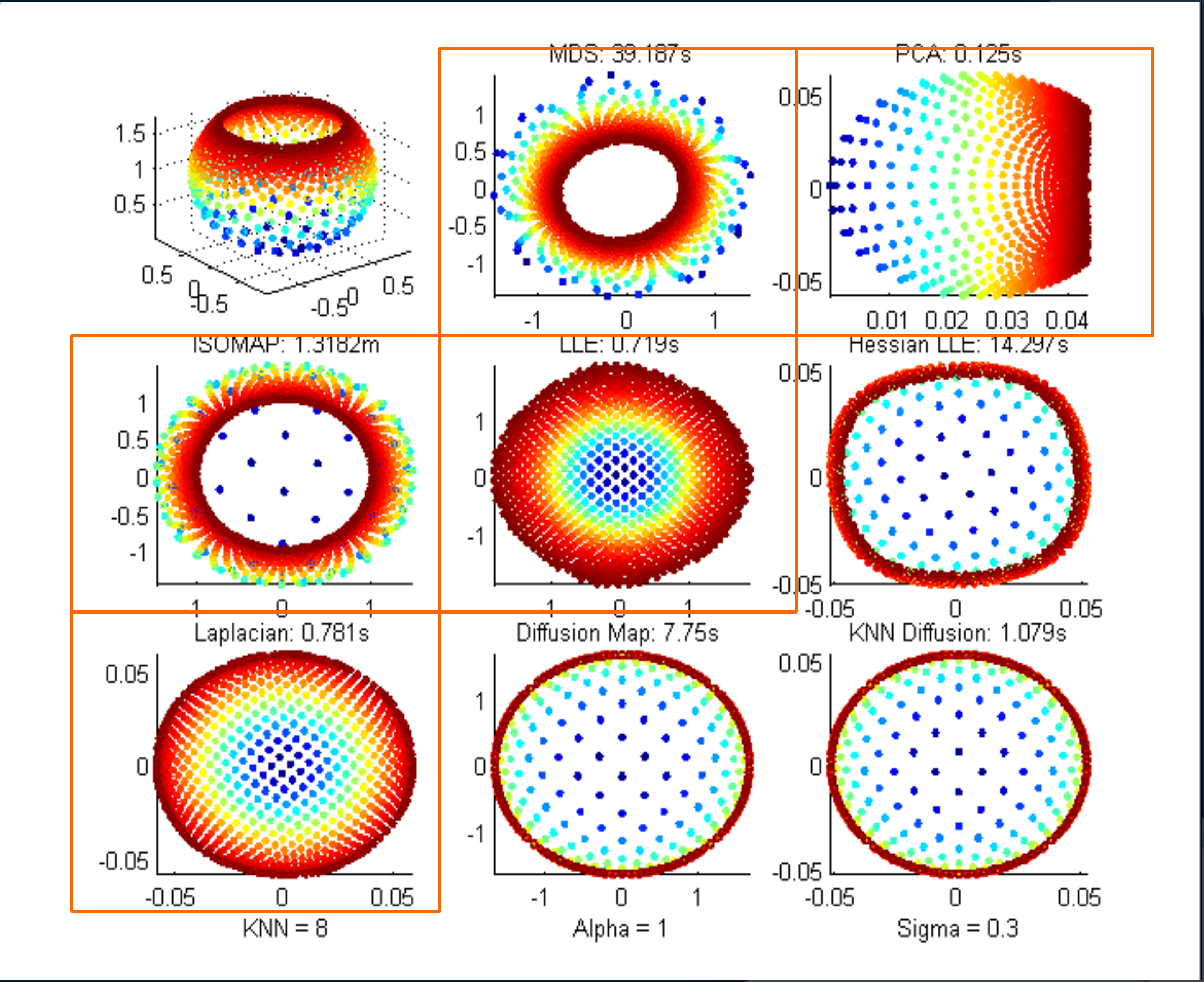


For M=8 clusters, MDS and PCA can still recover.
LLE and ISOMAP are decent, but Hessian and Laplacian fail.

Sparse Data & Non-uniform Sampling

- Punctured Sphere: the sampling is very sparse at the bottom and dense at the top.





Only LLE and Laplacian get decent results.
PCA projects the sphere from the side. MDS turns it inside-out.

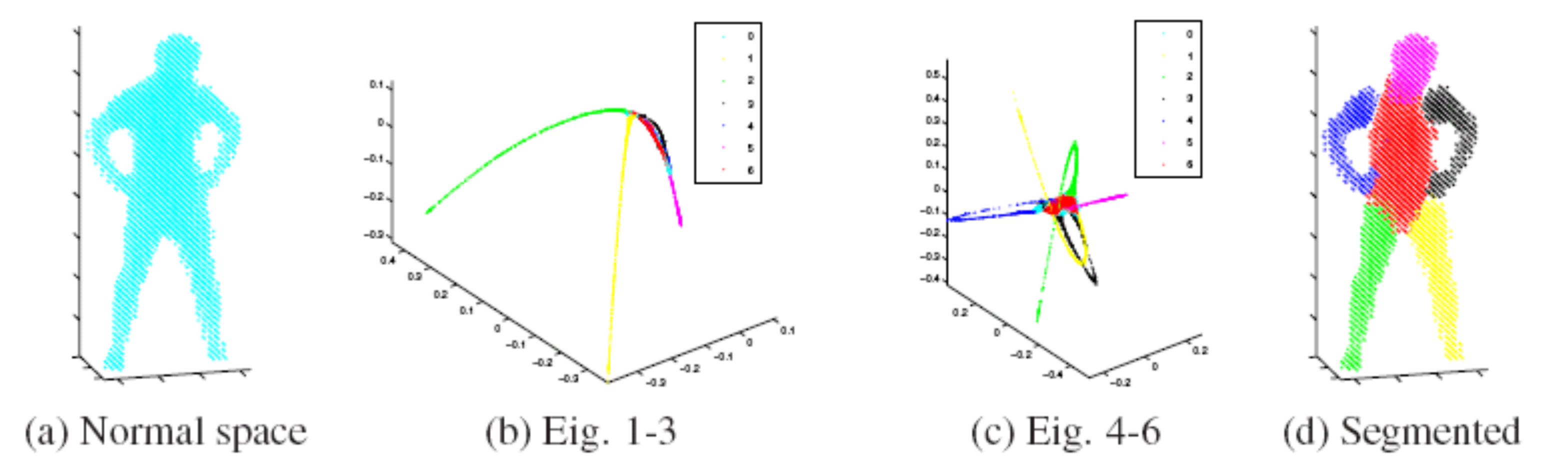
	MDS	PCA	ISOMAP	LLE	Laplacian	Diffusion Map	KNN Diffusion	Hessian
Speed	Very slow	Extremely fast	Extremely slow	Fast	Fast	Fast	Fast	Slow
Infers geometry?	NO	NO	YES	YES	YES	MAYBE	MAYBE	YES
Handles non-convex?	NO	NO	NO	MAYBE	MAYBE	MAYBE	MAYBE	YES
Handles non-uniform sampling?	YES	YES	YES	YES	NO	YES	YES	MAYBE
Handles curvature?	NO	NO	YES	MAYBE	YES	YES	YES	YES
Handles corners?	NO	NO	YES	YES	YES	YES	YES	NO
Clusters?	YES	YES	YES	YES	NO	YES	YES	NO
Handles noise?	YES	YES	MAYBE	NO	YES	YES	YES	YES
Handles sparsity?	YES	YES	YES	YES	YES	NO	NO	NO may crash
Sensitive to parameters?	NO	NO	YES	YES	YES	VERY	VERY	YES

Outline

- Introduction
- Algorithm
- Experimental Results
- Applications
- Conclusions

Applications

- We can apply manifold learning to pattern recognition (face, handwriting etc)
- Recently, ISOMAP and Laplacian eigenmap are used to initialize the human body model.



Outline

- Introduction
- Algorithm
- Experimental Results
- Applications
- Conclusions

Conclusions

- Laplacian eigenmap provides a computationally efficient approach to non-linear dimensionality reduction that has locality preserving properties
- Laplcian and LLE attempts to approximate or preserve neighborhood information, while ISOMAP attempts to faithfully approximate all geodesic distances on the manifold

Reference

- http://www.math.umn.edu/~wittman/mani/
- http://www.cs.unc.edu/Courses/comp290-090-s06/
- ISOMAP http://isomap.stanford.edu

  Joshua B. Tenenbaum, Vin de Silva, and John C. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction,” Science, vol. 290, Dec., 2000.
- LLE http://www.cs.toronto.edu/~roweis/lle/

  Sam T. Roweis, and Lawrence K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” Science, vol. 290, Dec., 2000
- Laplacian eigenmap http://people.cs.uchicago.edu/~misha/ManifoldLearning/index.html

  M. Belkin and P. Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.,*15(6):1373–1396, 2003.