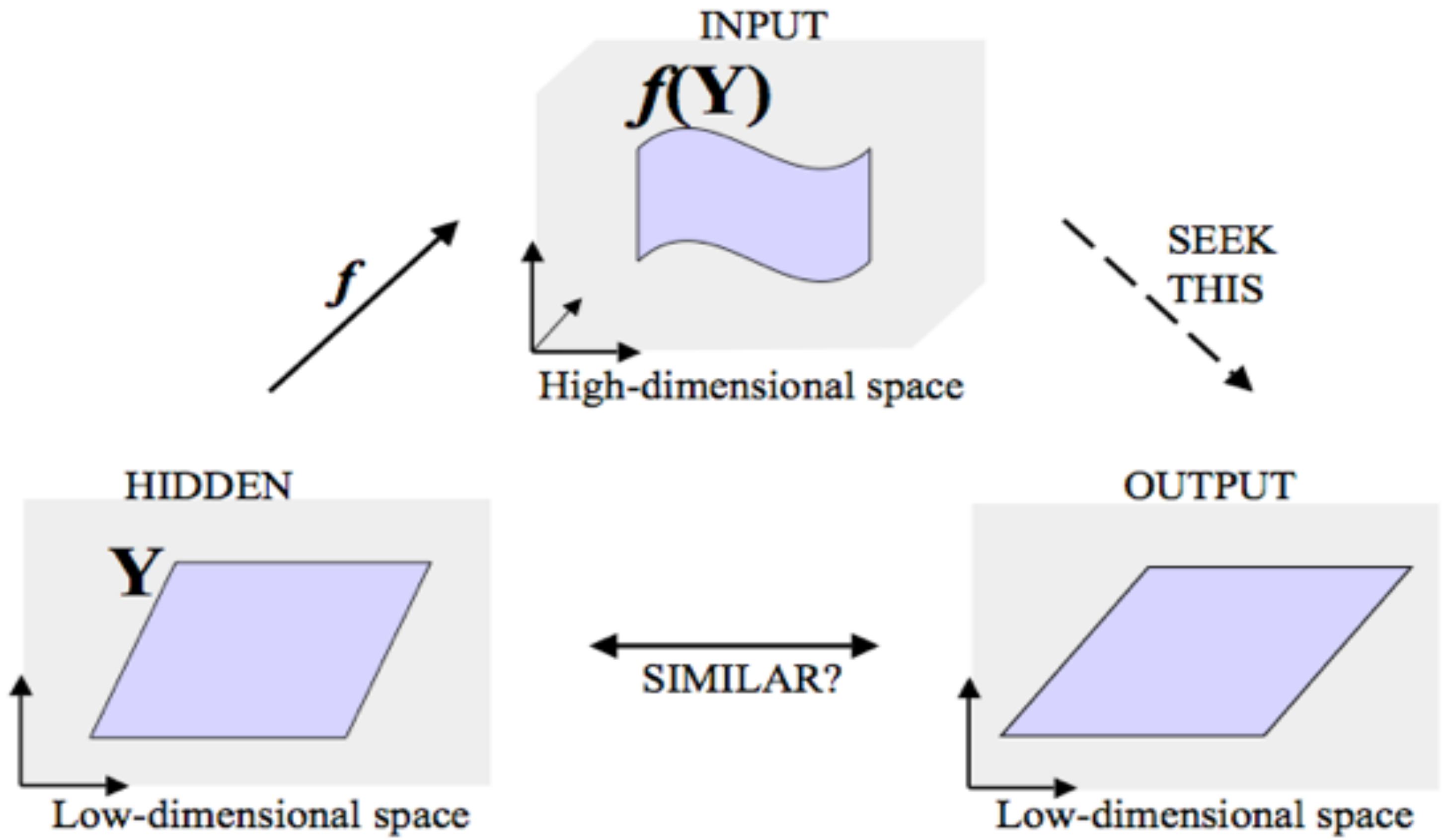
Manifold Learning II: Extended Locally Linear Embedding

姚 遠

2017

Generative Models in Manifold Learning



Spectral Geometric Embedding

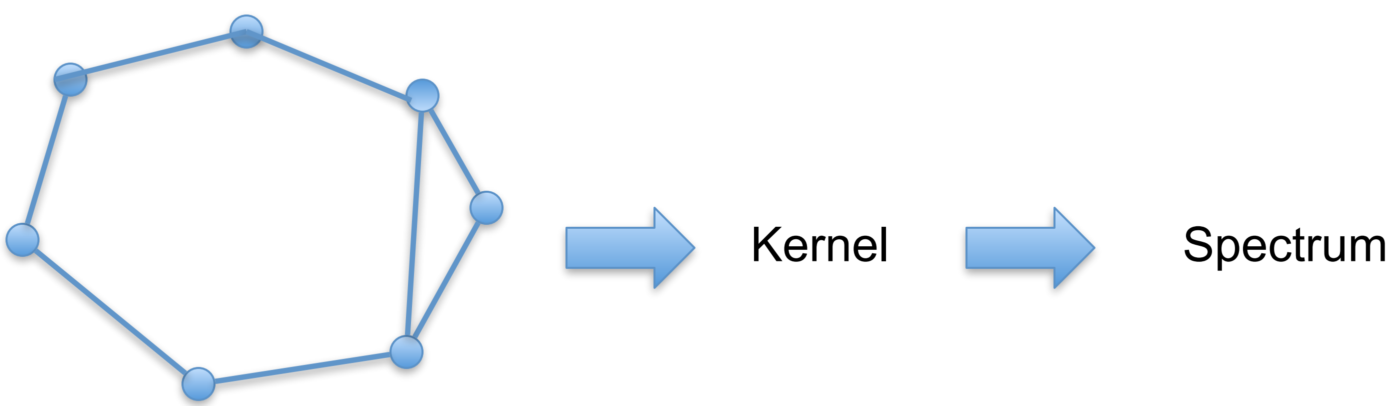
Given $x_1, \ldots, x_n \in \mathcal{M} \subset \mathbb{R}^N$,
Find $y_1, \ldots, y_n \in \mathbb{R}^d$ where $d << N$

- ISOMAP (Tenenbaum, et al, 00)
- LLE (Roweis, Saul, 00)
- Laplacian Eigenmaps (Belkin, Niyogi, 01)
- Local Tangent Space Alignment (Zhang, Zha, 02)
- Hessian Eigenmaps (Donoho, Grimes, 02)
- Diffusion Maps (Coifman, Lafon, et al, 04)

Related: Kernel PCA (Schoelkopf, et al, 98)

Meta-Algorithm

- Construct a neighborhood graph
- Construct a positive semi-definite kernel
- Find the spectrum decomposition



Recall: ISOMAP

1. Construct Neighborhood Graph.
2. Find shortest path (geodesic) distances.

D_{ij} is $n \times n$

3. Embed using Multidimensional Scaling.

Recall: LLE

- Construct a neighborhood Graph G=(V,E)
- Solve weights

$$\min_{\sum_{j \in \mathbb{N}_i} w_{ij} = 1} \|x_i - \sum_{j \in \mathcal{N}_i} w_{ij} x_j\|^2,$$

- Compute Embedding

$$\min_Y \sum_{i=1}^n \|Y_i - \sum_{j=1}^n W_{ij} Y_j\|^2 = \mathrm{trace}((I - W) Y^T Y (I - W)^T).$$

$$W_{ij}^{n \times n} = \begin{cases} w_{ij} & j \in \mathcal{N}(i), \\ 0 & \text{other's.} \end{cases}$$

This is equivalent to find smallest eigenvectors of $K = (I - W)^T (I - W)$.

Issues of LLE

Pick up a point x_i and its neighbors $\mathcal{N}_i$. Compute the local fitting weights

$$\min_{\sum_{j\in\mathbb{N}_i} w_{ij}=1} \|x_i - \sum_{j\in\mathcal{N}_i} w_{ij}x_j\|^2,$$

which is equivalent to

$$\min_{\sum_{j\in\mathbb{N}_i} w_{ij}=1} \|\sum_{j\in\mathcal{N}_i} w_{ij}(x_j - x_i)\|^2,$$

$$\min_{w_{ij}} \frac{1}{2}\|\sum_{j\in\mathcal{N}_i} w_{ij}(x_j - x_i)\|^2 + \lambda(1 - \sum_{j\in\mathcal{N}_i} w_{ij}).$$

$$w_i = \lambda C_i^\dagger \mathbf{1},$$

$$\lambda = \frac{1}{\mathbf{1}^T C_i^\dagger \mathbf{1}}, \qquad C_i(j,k) = \langle x_j - x_i, x_k - x_i\rangle$$

ill-posed or ill-conditioned?

Issues of LLE

$$w_i(\mu) = \lambda(C_i + \mu I)^{-1}\mathbf{1} = \sum_j \frac{1}{\lambda_j^{(i)} + \mu} v_j v_j^T \mathbf{1} \tag{82}$$

where the local PCA $C_i = V\Lambda V^T$ ($\Lambda = \mathrm{diag}(\lambda_j^{(i)})$, $V = [v_j]$).

- Low-pass filter of constant 1-vector
  - preserve projections on bottom eigenvectors associated with small eigenvalues $\lambda_j^{(i)} \ll \mu$
  - suppress projections on top eigenvectors associated with large eigenvalues
- If 1-vector is not so well-spread over null eigenspace, instability and missing directions as mu goes down!

Modified LLE (MLLE)

- Use all the null eigenspace!

MLLE replace the weight vector above by a weight matrix $W_i \in \mathbb{R}^{k_i \times s_i}$, a family of s_i weight vectors using bottom s_i eigenvectors of C_i, $V_i = [v_{k_i - s_i + 1}, \ldots, v_{k_i}] \in \mathbb{R}^{k_i \times s_i}$, such that

$$W_i = (1 - \alpha_i) w_i(\mu) \mathbf{1}_{s_i}^T + V_i H_i^T, \tag{83}$$

where $\alpha_i = \|V_i^T \mathbf{1}_{k_i}\|_2 / \sqrt{s_i}$ and $H_i = I_{s_i} - 2uu^T$ ($\|u\|_2 = 1$ or 0) is a Householder matrix ($H_i := I_{s_i}$ if $u = 0$) such that $HV_i^T \mathbf{1}_{k_i} = \alpha_i \mathbf{1}_{s_i}$ (hence $W_i^T \mathbf{1}_{k_i} = \mathbf{1}_{s_i}$, every column of W_i being a legal weight vector). In fact, one can choose u in the direction of $V_i^T \mathbf{1}_{k_i} - \alpha_i \mathbf{1}_{s_i}$. An adaptive choice of s_i is given in [**ZW**] using the

$$\min_Y \sum_i \sum_{l=1}^{s_i} \|y_i - \sum_{j \in \mathcal{N}_i} W_i(j,l) y_j\|^2 := \sum_i \|Y \widehat{W}_i\|_F^2 = \mathrm{trace}[Y (\sum_i \widehat{W}_i \widehat{W}_i^T) Y^T]$$

where $\widehat{W}_i$ is the embedding of $W_i \in \mathbb{R}^{k_i \times s_i}$ into $\mathbb{R}^{n \times s_i}$,

$$\widehat{W}_i(j,:) = \begin{cases} -1_{s_i}^T, & j = i, \\ W_i, & j \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases}$$

MLLE Algorithm (II)

Step 2 (local residue PCA): for each x_i and its neighbors $\mathcal{N}_i$ ($k_i = |\mathcal{N}_i|$), let $C_i = V\Lambda V^T$ be its eigenvalue decomposition where $\Lambda = (\lambda_1, \ldots, \lambda_{k_i})$ with $\lambda_1 \geq \cdots \geq \lambda_{k_i}$. Find the size of almost normal subspace s_i as the maximal size that the ratio of residue eigenvalue sum over principle eigenvalue sum is below a threshold, *i.e.*

$$s_i = \max_l \left\{ l \leq k_i - d, \frac{\sum_{j=k_i-l+1}^{k_i} \lambda_j}{\sum_{j=1}^{k_i-l} \lambda_j} \leq \eta \right\}$$

where η is a parameter, such as the median of ratios of residue eigenvalue sum over principle eigenvalue sum. Construct the normal subspace basis matrix as s_i-bottom eigenvector matrix of C_i, $V_i = [v_{k_i-s_i+1}, \ldots, v_{k_i}] \in \mathbb{R}^{k_i \times s_i}$, define the weight matrix

$$W_i = (1 - \alpha_i) w_i(\mu) \mathbf{1}_{s_i}^T + V_i H_i^T \in \mathbb{R}^{k_i \times s_i},$$

where $\alpha_i = \|V_i^T \mathbf{1}_{k_i}\|_2 / \sqrt{s_i}$ and $H_i = I_{s_i} - 2uu^T/\|u\|^2$ with $u = V_i^T \mathbf{1}_{k_i} - \alpha_i \mathbf{1}_{s_i}$ (or $u = 0$ if it is small).

MLLE Algorithm (III)

Step 3 (global alignment): define the weight embedding matrix

$$\widehat{W}_i(j,:) = \begin{cases} -1_{s_i}^T, & j = i, \\ W_i, & j \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases}$$

Compute $K = \widehat{W}^T \widehat{W}$ which is a positive semi-definite kernel matrix;
Step 4 (Eigenmap): Compute Eigenvalue decomposition $K = U \Lambda U^T$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ where $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_{n-1} > \lambda_n = 0$; choose bottom $d+1$ nonzero eigenvalues and corresponding eigenvectors and drop the smallest eigenvalue-eigenvector (0-constant) pair, such that

$$U_d = [u_{n-d}, \ldots, u_{n-1}], \quad u_j \in \mathbb{R}^n,$$

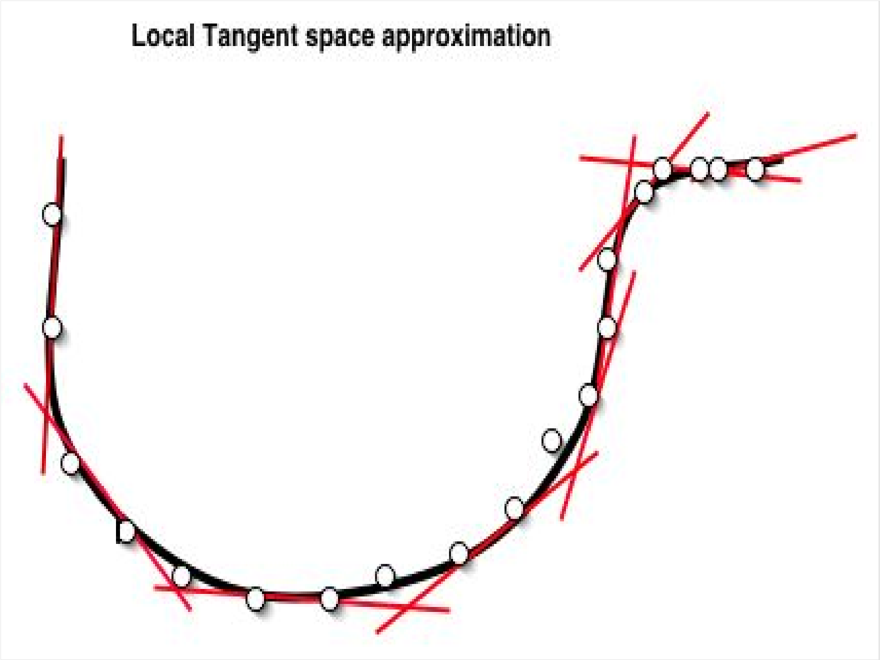
$$\Lambda_d = \mathrm{diag}(\lambda_{n-d}, \ldots, \lambda_{n-1}).$$

Define $Y_d = U_d \Lambda_d^{\frac{1}{2}}$.

Issues of MLLE

- MLLE computes bottom eigenvectors of local Gram (Covariance) matrix, expensive in computation and sensitive to noise
- How about only using top eigenvectors in local PCA?
  - LTSA
  - Hessian LLE

Local Tangent Space Alignment



Find a good approximation of tangent space of curve using discrete samples.
— Principal curve/manifold (Hastie-Stuetzle'89, Zha-Zhang'02)

Local SVD

For each x_i in $\mathbb{R}^d$ with neighbor $\mathcal{N}_i$ of size $|\mathcal{N}_i| = k_i - 1$, let $X^{(i)} = [x_{j_1}, x_{j_2}, \ldots, x_{j_{k_i}}] \in \mathbb{R}^{p \times k_i}$ be the coordinate matrix. Consider the local SVD (PCA)

$$\tilde{X}^{(i)} = [x_{i_1} - \mu_i, ..., x_{i_{k_i}} - \mu_i]^{p \times k_i} = X^{(i)} H = \tilde{U}^{(i)} \tilde{\Sigma} (\tilde{V}^{(i)})^T,$$

where $H = I - \frac{1}{k_i} \mathbf{1}_{k_i} \mathbf{1}_{k_i}^T$. Left singular vectors $\{\tilde{U}_1^{(i)}, ..., \tilde{U}_d^{(i)}\}$ give an orthonormal basis of the approximate d-dimensional tangent space at x_i. Right singular vectors $(\tilde{V}_1^{(i)}, \ldots, \tilde{V}_d^{(i)}) \cdot \tilde{\Sigma} \in \mathbb{R}^{k_i \times d}$ present the d-coordinates of k_i samples with respect to the tangent space basis.

LTSA

Let $Y_i \in \mathbb{R}^{d\times k_i}$ be the embedding coordinates of the samples in $\mathbb{R}^d$ and $L_i : \mathbb{R}^{p\times d}$ be an estimated basis of the tangent space at x_i in $\mathbb{R}^p$. Let $\Theta_i = \tilde{U}_d^{(i)} \tilde{\Sigma}_d (\tilde{V}_d^{(i)})^T \in \mathbb{R}^{p\times k_i}$ be the truncated SVD using top d components. LTSA looks for the minimizer of the following problem

$$\min_{Y,L} \sum_i \|E_i\|^2 = \sum_i \left\| Y_i (I - \frac{1}{n} 11^T) - L_i^T \Theta_i \right\|^2 . \tag{84}$$

One can estimate $L_i^T = Y_i(1 - \frac{1}{n}11^T)\Theta_i^\dagger$. Hence it reduces to

$$\min_{Y} \sum_i \|E_i\|^2 = \sum_i \left\| Y_i (I - \frac{1}{n} 11^T)(I - \Theta_i^\dagger \Theta_i) \right\|^2 \tag{85}$$

where $I - \Theta_i^\dagger \Theta_i$ is the projection to the normal space at x_i. This is equivalent to

LTSA Kernel

$$G_i = [1/\sqrt{k_i}, \tilde{V_1}^{(i)}, ..., \tilde{V_d}^{(i)}]^{k_i \times (d+1)},$$

$$W_i^{k_i \times k_i} = I - G_i G_i^T,$$

$$K^{n \times n} = \Phi = \sum_{i=1}^{n} S_i W_i W_i^T S_i^T$$

where the selection matrix $S_i^{n \times k} : [x_{i_1}, ..., x_{i_k}] = [x_1, ..., x_n] S_i^{n \times k}$.

1) Constant eigenvector is of 0-eigenvalue

2) So choose d+1 smallest eigenvectors for embedding

LTSA Algorithm (Zha-Zhang'02)

Algorithm 6: LTSA Algorithm

Input: A weighted undirected graph $G = (V, E)$ such that

$V = \{x_i \in \mathbb{R}^p : i = 1, \ldots, n\}$

$E = \{(i, j) :$ if j is a neighbor of i, i.e. $j \in \mathcal{N}_i\}$, e.g. k-nearest neighbors

Output: Euclidean d-dimensional coordinates $Y = [y_i] \in \mathbb{R}^{k \times n}$ of data.

Step 1 (local PCA): Compute local SVD on neighborhood of x_i, $x_{i_j} \in \mathcal{N}(x_i)$,

$$\tilde{X}^{(i)} = [x_{i_1} - \mu_i, ..., x_{i_k} - \mu_i]^{p \times k} = \tilde{U}^{(i)} \tilde{\Sigma} (\tilde{V}^{(i)})^T,$$

where $\mu_i = \sum_{j=1}^k x_{i_j}$. Define

$$G_i = [1/\sqrt{k}, \tilde{V_1}^{(i)}, ..., \tilde{V_d}^{(i)}]^{k \times (d+1)};$$

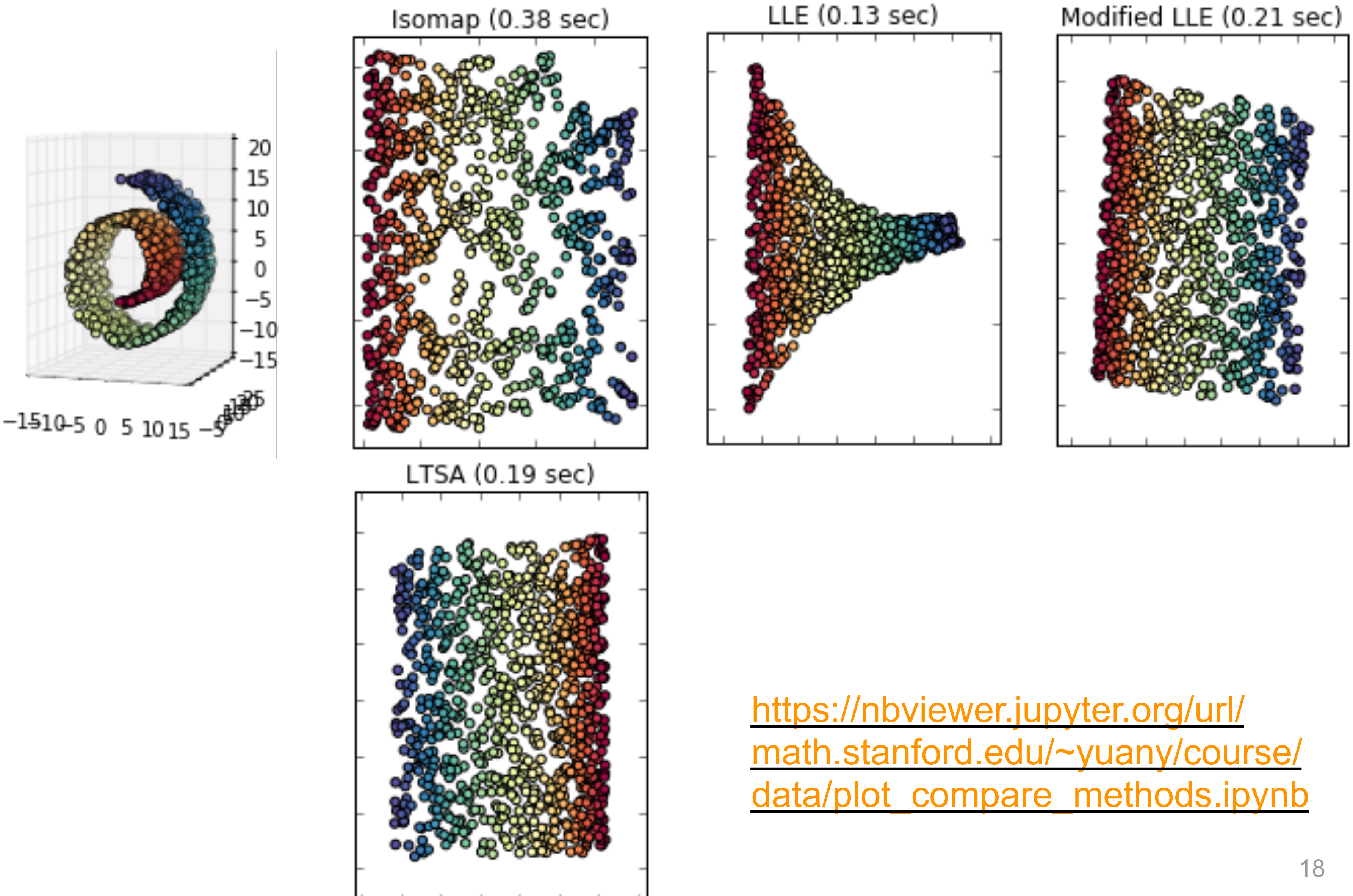
Step 2 (tangent space alignment): Alignment (kernel) matrix

$$K^{n \times n} = \sum_{i=1}^n S_i W_i W_i^T S_i^T, \quad W_i^{k \times k} = I - G_i G_i^T,$$

where selection matrix $S_i^{n \times k} : [x_{i_1}, ..., x_{i_k}] = [x_1, ..., x_n] S_i^{n \times k}$;

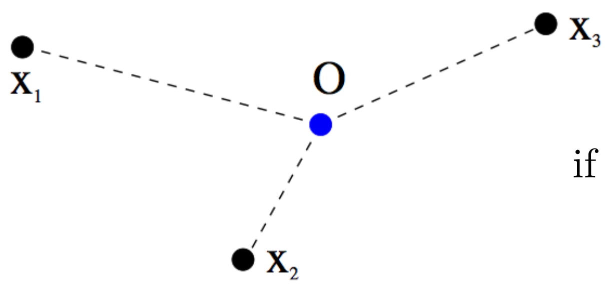
Step 3: Find smallest $d + 1$ eigenvectors of K and drop the smallest eigenvector, the remaining d eigenvectors will give rise to a d-embedding.

Comparisons on Swiss Roll



https://nbviewer.jupyter.org/url/math.stanford.edu/~yuany/course/data/plot_compare_methods.ipynb

Hessian LLE



In LLE, one chooses the weights w_{ij} to minimize the following energy

$$\min_{\sum_{j\in\mathbb{N}_i} w_{ij}=1} \| \sum_{j\in\mathcal{N}_i} w_{ij}(x_j - x_i)\|^2.$$

if the points $\tilde{x}_j = x_j - x_i$ are linearly dependent

$$0 = \sum_{j\in\mathcal{N}_i} w_{ij}\tilde{x}_j, \quad \text{and} \quad 1 = \sum_{j\in\mathcal{N}_i} w_{ij}.$$

For any smooth function $y(x)$, consider its Taylor expansion up to the second order

$$y(x) = y(0) + x^T\nabla y(0) + \frac{1}{2}x^T(\mathcal{H}y)(0)x + o(\|x\|^2).$$

$$\begin{aligned}
(I - W)y(0) &:= y(0) - \sum_{j\in\mathcal{N}_i} w_{ij}y(\tilde{x}_i) \\
&\approx y(0) - \sum_{j\in\mathcal{N}_i} w_{ij}y(0) - \sum_{j\in\mathcal{N}_i} w_{ij}\tilde{x}_i^T\nabla y(0) - \frac{1}{2}\sum_{j\in\mathcal{N}_i} \tilde{x}_i^T(\mathcal{H}y)(0)\tilde{x}_i \\
&= -\frac{1}{2}\sum_{j\in\mathcal{N}_i} \tilde{x}_i^T(\mathcal{H}y)(0)\tilde{x}_i.
\end{aligned}$$

Hessian Null

The Hessian matrix

$$(\mathcal{H}y)(0) := \left[\frac{\partial^2 y(x)}{\partial x(i)\partial x(j)}\right]_{x=0} = 0,$$

if function $y(x)$ is a linear transform of the coordinates $x \in \mathbb{R}^p$ in the tangent space at x_i. In this case $(I - W)y(0) = 0$ and y reaches a minimizer.

In other words, the kernel of $(\mathcal{H}y)$ has dimension $d+1$, consisting the constant function and d linearly independent coordinates. Inspired by such an observation, Donoho and Grimes [**DG03b**] proposed Hessian LLE (Eigenmap) in search of

$$\min_{y \perp \mathbf{1}} \int \|\mathcal{H}y\|^2, \quad \|y\| = 1.$$

Hessian LLE Algorithm (I)

Algorithm 7: Hessian LLE Algorithm

Input: A weighted undirected graph $G = (V, E, d)$ such that

$V = \{x_i \in \mathbb{R}^p : i = 1, \ldots, n\}$

$E = \{(i, j) :$ if j is a neighbor of i, i.e. $j \in \mathcal{N}_i\}$, e.g. k-nearest neighbors

Output: Euclidean d-dimensional coordinates $Y = [y_i] \in \mathbb{R}^{d \times n}$ of data.

Step 1: Compute local PCA on neighborhood of x_i, for,

$$\tilde{X}^{(i)} = [x_{i_1} - \mu_i, ..., x_{i_k} - \mu_i]^{p \times k} = \tilde{U}^{(i)} \tilde{\Sigma} (\tilde{V}^{(i)})^T, \quad x_{i_j} \in \mathcal{N}(x_i),$$

where $\mu_i = \sum_{j=1}^{k} x_{i_j} = \frac{1}{k} X_i \mathbf{1}$;

- Left top singular vectors $\{\tilde{U}_1^{(i)}, ..., \tilde{U}_d^{(i)}\}$ give an orthonormal basis of the approximate tangent space at x_i,
- Right top singular vectors $[\tilde{V}_1^{(i)}, ..., \tilde{V}_d^{(i)}]$ are representation coordinates in the tangent space of local sample points around x_i.

Continued...

Hessian LLE Algorithm (II)

4 ***Step 2***: Null Hessian estimation: define

$$M = [1, \tilde{V}_1, ..., \tilde{V}_k, \tilde{V}_1\tilde{V}_2, ..., \tilde{V}_{d-1}\tilde{V}_d] \in \mathbb{R}^{k\times(1+d+\binom{d}{2})}$$

where $\tilde{V}_i\tilde{V}_j = [\tilde{V}_{ik}\tilde{V}_{jk}]^T \in \mathbb{R}^k$ denotes the elementwise product (Hadamard product) between vector $\tilde{V}_i$ and $\tilde{V}_j$. Now we perform a Gram-Schmidt Orthogonalization procedure on M, get

$$\tilde{M} = [1, \hat{v}_1, ..., \hat{v}_k, \hat{w}_1, \hat{w}_2, ..., \hat{w}_{\binom{d}{2}}] \in \mathbb{R}^{k\times(1+d+\binom{d}{2})}$$

Define

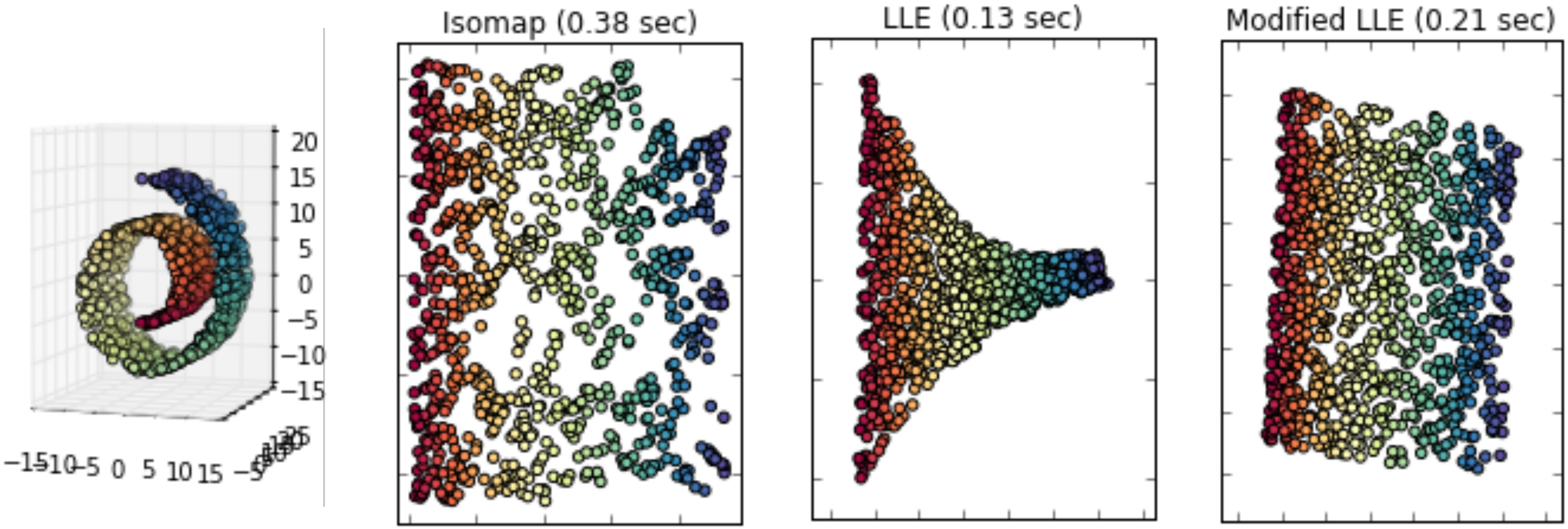
$$[H^{(i)}]^T = [last \quad \binom{d}{2} \quad columns \quad of \quad \tilde{M}]_{k\times\binom{d}{2}}.$$

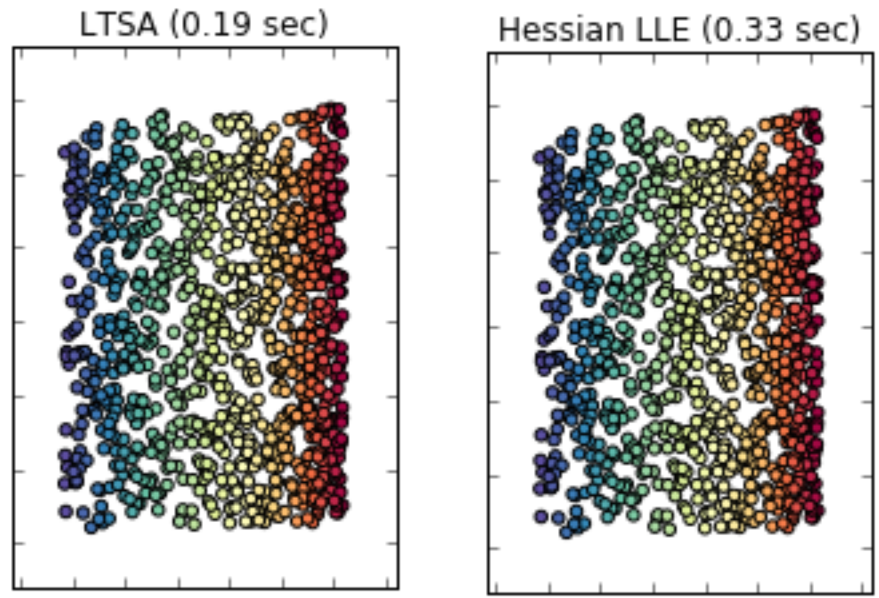
Step 3: Define

$$K = \sum_{i=1}^{n} S^{(i)}H^{(i)T}H^{(i)}S^{(i)T} \in \mathbb{R}^{n\times n}, \quad [x_1, .., x_n]S^{(i)} = [x_{i_1}, ..., x_{i_k}],$$

find smallest $d+1$ eigenvectors of K and drop the smallest eigenvector, and the remaining d eigenvectors will give rise to a d-embedding.

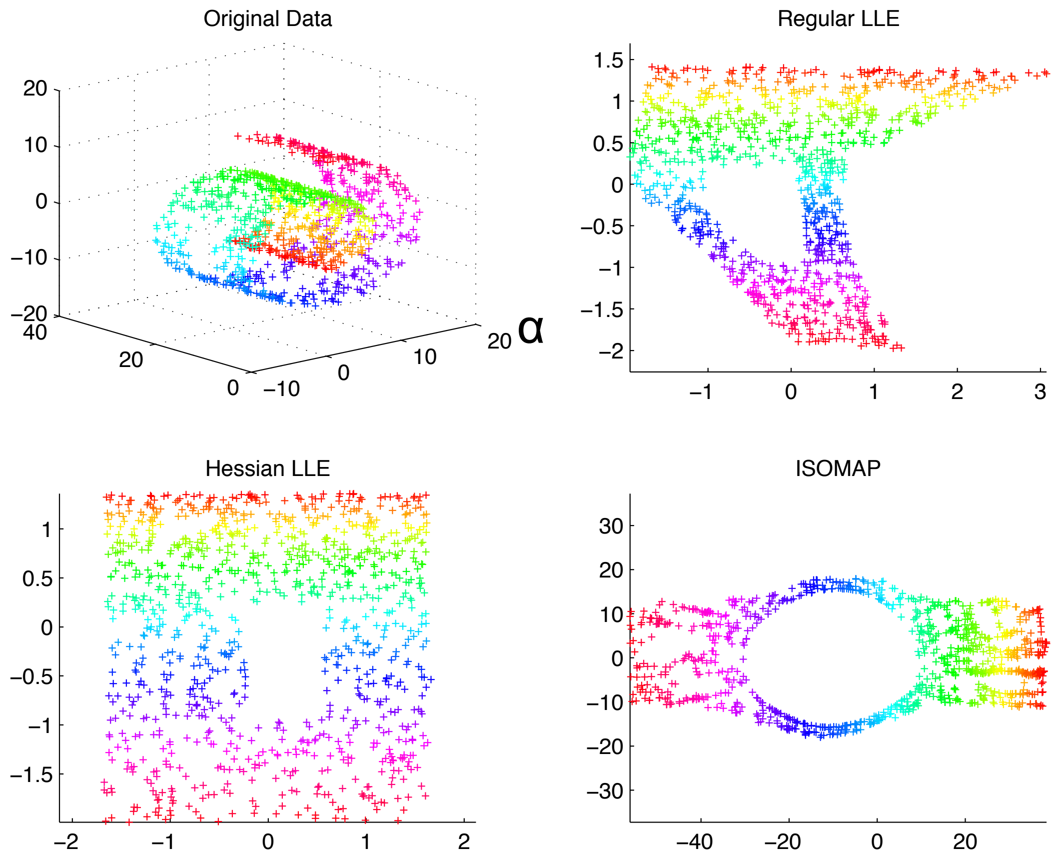
Comparisons on Swiss Roll





https://nbviewer.jupyter.org/url/math.stanford.edu/~yuany/course/data/plot_compare_methods.ipynb

Comparisons on Swiss Roll with holes



Two Assumptions on ISOMAP

(ISO1) *Isometry.* The mapping ψ preserves geodesic distances. That is, define a distance between two points m and m' on the manifold according to the distance travelled by a bug walking along the manifold M according to the shortest path between m and m'. Then the isometry assumption says that

$$G(m, m') = |\theta - \theta'|, \qquad \forall m \leftrightarrow \theta, m' \leftrightarrow \theta',$$

where $|\cdot|$ denotes Euclidean distance in $\mathbb{R}^d$.

(ISO2) *Convexity.* The parameter space Θ is a convex subset of $\mathbb{R}^d$. That is, if θ, θ' is a pair of points in Θ, then the entire line segment $\{(1-t)\theta + t\theta' : t \in (0,1)\}$ lies in Θ.

Convexity is hard to meet: consider two balls in an image which never intersect, whose center coordinate space (x_1, y_1, x_2, y_2) must have a hole.

Relaxations (Donoho-Grimes'2003)

(LocISO1) *Local Isometry.* In a small enough neighborhood of each point m, geodesic distances to nearby points m' in M are identical to Euclidean distances between the corresponding parameter points θ and θ'.

(LocISO2) *Connectedness.* The parameter space Θ is a open connected subset of $\mathbb{R}^d$.

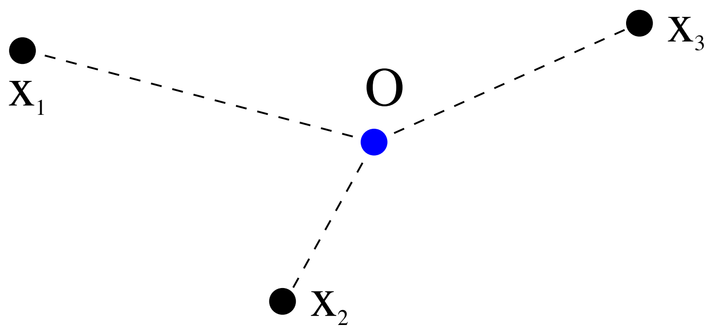
Convergence of Hessian LLE (Donoho-Grimes)

Theorem 1 *Suppose $M = \psi(\Theta)$ where Θ is an open connected subset of $\mathbb{R}^d$, and ψ is a locally isometric embedding of Θ into $\mathbb{R}^n$. Then $\mathcal{H}(f)$ has a $d+1$ dimensional nullspace, consisting of the constant function and a d-dimensional space of functions spanned by the original isometric coordinates.*

We give the proof in Appendix A.

Corollary 2 *Under the same assumptions as Theorem 1, the original isometric coordinates θ can be recovered, up to a rigid motion, by identifying a suitable basis for the null space of $\mathcal{H}(f)$.*

Laplacian and LLE



$$\sum w_i x_i = 0$$

$$\sum w_i = 1$$

Hessian H. Taylor expansion :

$$f(x_i) = f(0) + x_i^t \nabla f + \frac{1}{2} x_i^t H x_i + o(\|x_i\|^2)$$

$$(I-W)f(0) = f(0) - \sum w_i f(x_i) \approx f(0) - \sum w_i f(0) - \sum_i w_i x_i^t \nabla f - \frac{1}{2} \sum_i x_i^t H x_i =$$

$$= -\frac{1}{2} \sum_i x_i^t H x_i \approx -trH = \Delta f$$

when x_i becomes an orthonormal basis…

Discrete Laplacian

Find $y_1, \ldots, y_n \in R$

$$\min \sum_{i,j} (y_i - y_j)^2 W_{ij}$$

Tries to preserve locality

A Fundamental Identity

But $L = D - W, \quad D = \mathrm{diag}(\sum_{j \in \mathcal{N}_i} w_{ij})$

$$\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 W_{ij} = \mathbf{y}^T L \mathbf{y}$$

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = \sum_{i,j}(y_i^2 + y_j^2 - 2y_i y_j) W_{ij}$$

$$= \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{i,j} y_i y_j W_{ij}$$

$$= 2\mathbf{y}^T L \mathbf{y}$$

Embedding of Unnormalized Laplacian Eigenmap

$$\lambda = 0 \rightarrow \mathbf{y} = \mathbf{1}$$

Uniform sampling: $$\min_{\mathbf{y}^T\mathbf{1}=0} \mathbf{y}^T L \mathbf{y}$$

Let $Y = [\mathbf{y}_1 \mathbf{y}_2 \ldots \mathbf{y}_m]$

$$\sum_{i,j} ||Y_i - Y_j||^2 W_{ij} = \text{trace}(Y^T L Y)$$

subject to $Y^T Y = I$.

Use eigenvectors of L to embed.

Nonuniform (GEV): $$\underset{\substack{\mathbf{y} \\ \mathbf{y}^T D \mathbf{y}=1}}{\operatorname{argmin}} \mathbf{y}^T L \mathbf{y}.$$

How to find weights? Heat kernels...

- $H_t(x,y) = \sum_i e^{-\lambda_i t} \phi_i(x) \phi_i(y)$
- in $\mathbb{R}^d$, closed form expression

$$H_t(x,y) = \frac{1}{(4\pi t)^{d/2}} e^{-\frac{\|x-y\|^2}{4t}}$$

- Goodness of approximation depends on the gap

$$\left| H_t(x,y) - \frac{1}{(4\pi t)^{d/2}} e^{-\frac{\|x-y\|^2}{4t}} \right|$$

good for small *t*

- H_t is a Mercer kernel intrinsically defined on manifold. Leads to SVMs on manifolds.

Laplacian Eigenmaps (I) [Belkin-Niyogi]

Algorithm 8: Laplacian Eigenmap

Input: An adjacency graph $G = (V, E, d)$ such that

$V = \{x_i : i = 1, \ldots, n\}$

$E = \{(i, j) :$ if j is a neighbor of i, i.e. $j \in \mathcal{N}_i\}$, e.g. k-nearest neighbors, ϵ-neighbors

$d_{ij} = d(x_i, x_j)$, e.g. Euclidean distance for $x_i \sim x_j$ are in neighbor

Output: Euclidean d-dimensional coordinates $Y = [y_i] \in \mathbb{R}^{k \times n}$ of data.

Step 1: Choose weights

(a) Heat kernel weights (parameter t):

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & i \sim j, \\ 0, & \text{otherwise.} \end{cases}$$

(b) Simple-minded ($t \to \infty$), $W_{ij} = 1$ if i and j are connected by an edge and $W_{ij} = 0$ otherwise.

Step 2 (Eigenmap): Let $D = \mathrm{diag}(\sum_j W_{ij})$ and $L = D - W$. Compute smallest $d + 1$ generalized eigenvectors

$$L y_l = \lambda_l D y_l, \quad l = 0, 1, \ldots, d,$$

such that $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_d$. Drop the zero eigenvalue λ_0 and constant eigenvector y_0, and construct $Y_d = [y_1, \ldots, y_d] \in \mathbb{R}^{n \times d}$.

Hessian vs. Laplacian

- Laplacian LLE

$$f^T L f = \sum_{i \geq j} w_{ij}(f_i - f_j)^2 \geq 0 \sim \int \|\nabla_M f\|^2 = \int (\mathrm{trace}(f^T \mathcal{H} f))^2$$

where $\mathcal{H} = [\partial^2 / \partial_i \partial_j] \in \mathbb{R}^{d \times d}$ is the Hessian matrix.

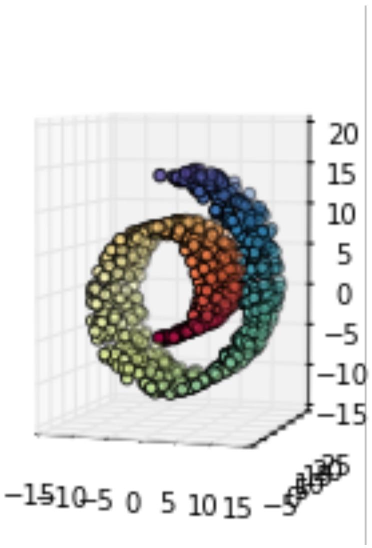
- Hessian LLE

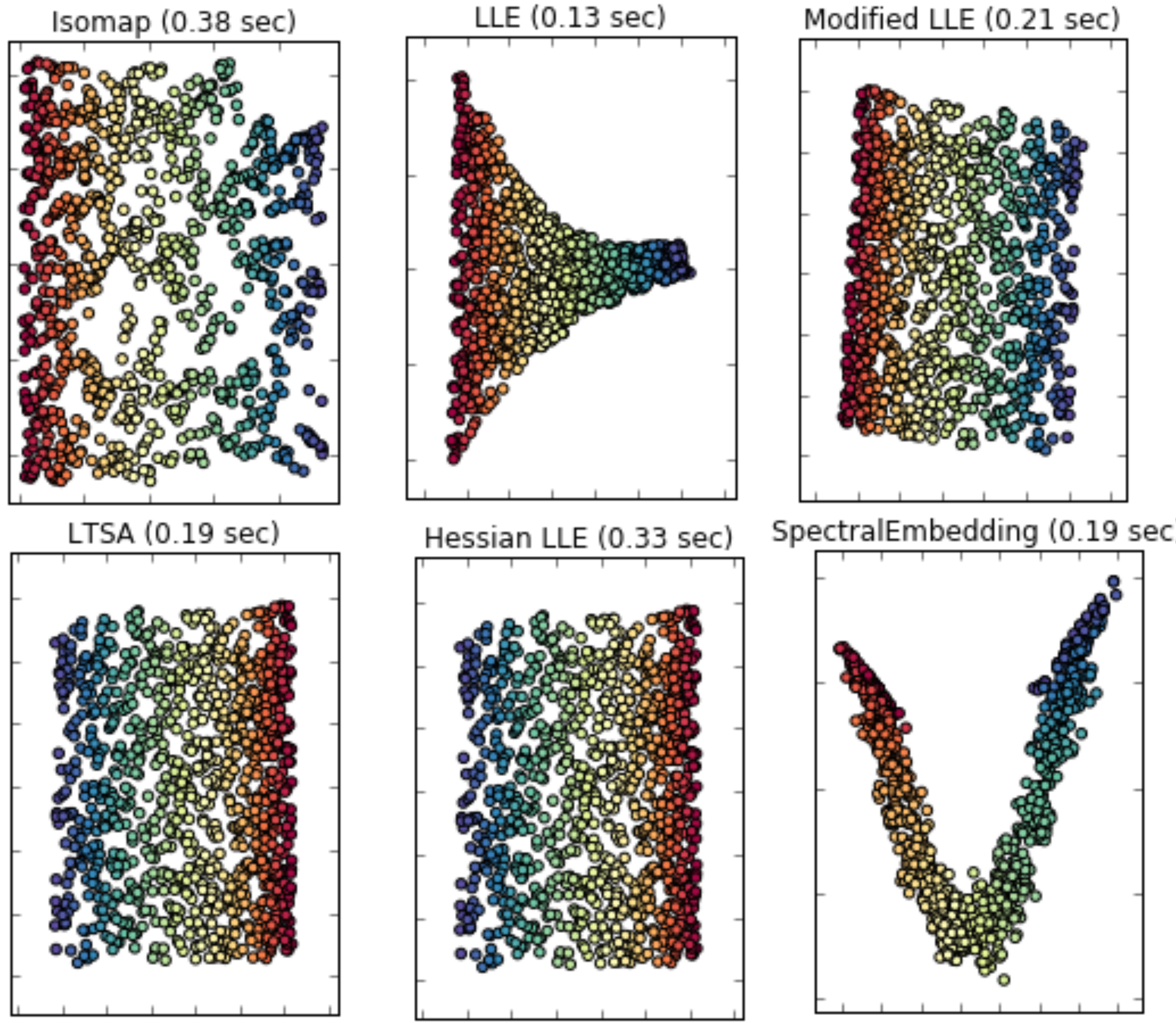
$$\min \int \|\mathcal{H} f\|^2, \quad \|f\| = 1$$

- Laplacian kernel: const + linear + bilinear
- Hessian kernel: const + linear functions

Note that: $\Delta(f) = trace\big(H(f)\big)$

Comparisons on Swiss Roll





https://nbviewer.jupyter.org/url/math.stanford.edu/~yuany/course/data/plot_compare_methods.ipynb

Connection to Markov Chain

- $L = D - W$
- $P = I - D^{-1}L = D^{-1}W$ is a markov matrix
- v is generalized eigenvector of L: $L v = \lambda D v$
- v is also a right eigenvector of P with eigenvalue $1-\lambda$
- P is lumpable iff v is piece-wise constant
- So Laplacian eigenmaps have Markov Chain interpretations (Diffusion Map), with more connection to topology …