Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data

STAT 6390

The University of Texas at Dallas

April 17, 2024

Rozina Akter
Ibrahim Hossain Sajal
John Kenney

Motivation

There are several concrete situations related to image analysis and acoustics. Let's discuss the following image and acoustic analysis:

- **Scene variation:** Pose variation and facial gesturing.
- **Image variation:** Changes in lighting sources and in lighting colors.
- **Acoustic articulation:** Changes in sound source, direction of the sound.

Motivation continued...

In all such situations, there is an underlying parameter controlling the articulation of the scene; here are two examples:

- **Facial expression:** The tonus of several facial muscles control facial expression. A parameter vector θ records the contradiction of each of those muscles.
- **Pose variations:** Several joint angles control the combined pose of the elbow-wrist-finger system in combination.

We are interested in recovering the underlying parameters θ_i from the observed points m_i on the articulation manifold M.

Parametrization

Parametrization Recovery Problem: Given a collection of data points (m_i) on an articulation manifold M, recover the mapping ψ and the parameter points θ_i.

- If ψ is one solution, and $\psi : \mathbb{R}^d \mapsto \mathbb{R}^d$, the combined mapping $\psi \circ \phi$ is another solution.
- We need some extra assumptions to solve this problem and determine unique solutions.

Assumptions

- **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 traveled by a bug walking along the manifold M according to the shortest path between m and m'. Then the isometry assumption says that

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

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

Under, these assumptions, ISOMAP, introduced by Dr. Tenenbaum et al.[1], which recovered Θ.

Validity of assumptions

- The above-stated assumptions lead to two associated questions:

Q1 **Do interesting articulation manifolds have isometric structure?**
YES. For example, The author considered images of a ball on a white background, where the underlying articulation parameter is the position of the ball's center. Images m has been modeled as continuous function $m(x, y)$ on the plane $(x, y) \in \mathbb{R}^2$. Let, B_θ denote the ball of radius 1 centered at $\theta \in \mathbb{R}^2$, and define

$$m_\theta(x, y) = 1_{B_\theta}(x, y)$$

If θ is a convex subset of $\mathbb{R}^2$, then isometry holds.

Validity of assumptions

Q2 **Are interesting parameter spaces truly convex?**
NO.
A simple example occurs with images showing two balls that articulate by translation. The two balls never overlap. In this case, the parameter space $\Theta \subset \mathbb{R}^4$ becomes nonconvex.

Weaker assumptions

- **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'
- **Connectedness:** The parameter space Θ is an open, connected subset of $\mathbb{R}^d$

In such a setting, the original assumptions of **ISOMAP** are violated, and the method fails to recover the parameter space up to a linear mapping. In this article, Dr. Donoho proposed a procedure to solve this problem.

1. Introduction
2. Methodology
3. Our Implementation and Results
4. HLLE Extensions
5. Conclusion
6. References

Overview of the Method

- Dr. Donoho describes a method for recovering a low dimensional parametrization of high dimensional data by assuming that the data lie on a manifold M.
- The method, Hessian Locally Linear Embedding (HLLE) accomplishes linear embedding by minimizing the Hessian functional on the manifold that is locally isometric to an open, connected subset Θ of Euclidean space $\mathbb{R}^d$. This subset need not to be convex.

Notation

- Θ is a parameter space and $\Theta \subset \mathbb{R}^d$
- A smooth mapping $\psi : \Theta \mapsto \mathbb{R}^n$
- $\mathbb{R}^n$ is the embedding space obeys $d < n$
- $m_i, i = 1, 2, ..., N$ are the data examples with different choices of control vectors $\theta_i, i = 1, 2, ..., N$. We also enumerate $m = \psi(\theta)$ of all possible measurements as the parameters vary.
- M is the data manifold and we speak of the image $M = \psi(\Theta)$. All the data points m_i lie exactly in the manifold M.

Hessian Matrix

Let's $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a function with input vector $s \in \mathbb{R}^n$. If $f(x) \in \mathbb{R}$ and all second-order partial derivatives of f exist, then Hessian H of f is a square $n \times n$ matrix defined as

$$\mathrm{H}=\begin{bmatrix} \frac{\delta^2 f}{\delta x_1^2} & \frac{\delta^2 f}{\delta x_1 \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_1 \delta x_n} \\ \frac{\delta^2 f}{\delta x_2 \delta x_1} & \frac{\delta^2 f}{\delta x_2^2} & \cdots & \frac{\delta^2 f}{\delta x_2 \delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2 f}{\delta x_n \delta x_1} & \frac{\delta^2 f}{\delta x_n \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_n^2} \end{bmatrix}$$

That is, the Hessian matrix is
$(H_f)_{i,j} = \frac{\delta^2 f}{\delta x_i \delta x_j}$ where i and j range from 1 to n.

The H Function

- The method follows from properties of a quadratic form

$$\mathscr{H}(f) = \int_M ||H_f(m)||_m^2 dm$$

defined on functions

$$f : M \mapsto \mathbb{R}$$

- $\mathscr{H}(f)$ measures the average over the manifold M of the Frobenius norm of the Hessian of f.

- **What is Frobenius norm of the Hessian of f?**

  The Frobenius norm of a matrix belongs to the group of entry-wise matrix norms. The general p-norm of matrix **A** reads:

$$\|A\|_{p,p} = \left(\sum_{i=1}^{n}\sum_{j=1}^{m}|a_{ij}|^p\right)^{1/p}$$

The Frobenius norm is then obtained by setting p=2:

$$\|A\|_{2,2} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}|a_{ij}|^2}$$

Alternatively,

$$\|A\|_{2,2} = \sqrt{tr(A^T A)} = \sqrt{\sum_{i=1}^{min(n,m)} \sigma_i^2(A)}$$

where A^T represents the conjugate transpose of matrix A.

Tangent space and Neighborhood

- Suppose that $M \subset \mathbb{R}^n$ is a smooth manifold, and thus the tangent space $T_m(M)$ is well defined at each point $m \in M$.
- Tangent space as an affine subspace, associate an orthonormal coordinate system to each such tangent space $T_m(M) \subset \mathbb{R}^n$.
- There is a neighborhood N_m of m such that each point $m' \in N_m$ has a unique closest point $v' \in T_m(M)$ and such that the implied mapping $m' \to v'$ is smooth.
- We obtain local coordinates for a neighborhood N_m of $m \in M$, $x_1^{(tan,m)}, \ldots, x_d^{(tan,m)}$.
- We use the local coordinates to define the Hessian of $f : M \to \mathbb{R}$ that is C^2 near m.

Tangent Hessian

- Suppose $m' \in N_m$ has local coordinates $x = x^{(tan,m)}$.
- The rule $g(x) = f(m')$ defines a function $g : U \to \mathbb{R}$, where U is a neighborhood of 0 in $\mathbb{R}^d$.
- As the mapping $m' \to g$ is smooth, the function g is C^2.
- Hessian of f at m in tangent coordinates as the Hessian of g

$$(H_f^{(tan)}(m))_{i,j} = \frac{\delta}{\delta x_i}\frac{\delta}{\delta x_j} g(x)|_{x=0} \tag{1}$$

- Consider a quadratic form defined on C^2 functions by

$$\mathscr{H}(f) = \int_M ||(H_f^{(tan)}(m))||_F^2 dm \tag{2}$$

where dm stands for a probability measure on M.
- $\mathscr{H}(f)$ measures the average curviness of f over manifold M.

Theorem and Corollary

Theorem: 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 $\mathscr{H}(f)$ has a (d+1)-dimensional null space consisting of the constant function and a d-dimensional space of functions spanned by the original isometric coordinates.

Corollary: Under the same assumptions as Theorem, the original isometric coordinates θ can be recovered, up to a rigid motion, by identifying a suitable basis for the null space of $\mathscr{H}$.

Hessian Locally Linear Embedding (HLLE)

Consider the setting with sampled data (m_i) lying on M, we to recover the underlying parameterization ψ and underlying parameter settings θ_i. *HLLE* algorithm:

- **Input**: $(m_i : i = 1, ..., N)$ a collection of N points in $\mathbb{R}^n$.
- **Parameters**: d - the dimension of the parameter space; k - the size of the neighborhoods for fitting.
- **Constraints**: $min(k, n) > d$.
- **Output**: $(w_i : i = 1, ..., N)$ a collection of N points in $\mathbb{R}^d$, the recovered parametrization.

HLLE Procedure

- Identify neighbors
- Obtaining tangent coordinates
- Develop Hessian estimator
- Develop quadratic form
- Find approximate null space
- Find basis for null space

HLLE Procedure (Contd.)

Identify neighbors

- For each m_i, identify the indices corresponding to the k-nearest neighbors in Euclidean distance.
- N_i denotes the collection of those neighbors.
- For each N_i, form a $k \times n$ matrix M^i with rows that consist of the recentered points $m_j - \bar{m}_i$, where $j \in N_i$.

Obtaining tangent coordinates

- Singular value decomposition of M^i, getting matrices U, D, and V; dimension of U is $k \times min(k, n)$.
- The first d columns of U give the tangent coordinates of points in N_i.

HLLE Procedure (Contd.)

Develop Hessian estimator

- Develop matrix H^i with the property that if f is a smooth function $f : M \to \mathbb{R}$, and $\mathbf{f}_j = f(m_i)$.
- The vector v^i with entries that are obtained by extracting entries from $\mathbf{f}$ corresponding to points in neighborhood N_i.
- Then $H^i v^i$ gives a $d(d+1)/2$ vector with entries that approximate the entries of the Hessian matrix, $(\frac{\delta f}{\delta U_i \delta U_j})$.

Develop quadratic form

- Build a symmetric matrix $\mathscr{H}_{i,j}$ having, in coordinate pair ij,

$$\mathscr{H}_{i,j} = \sum_l \sum_r ((H^l)_{r,i} (H^l)_{r,j}) \tag{3}$$

HLLE Procedure (Contd.)

Find approximate null space

- Eigen-analysis of $\mathscr{H}$ to identify the $(d-1)$-dimensional subspace corresponding to the $d+1$ smallest eigenvalues.
- An eigenvalue 0 is associated with the subspace of constant functions.
- Next d eigenvalues will correspond to eigenvectors spanning in space V^d where the embedding coordinates are to be found.

Find basis for null space

- Select a basis for V^d - its restriction to specific neighborhood N_0 provides an orthonormal basis.
- The given basis has basis vectors $w_1, \ldots, w_d$; which are the embedding coordinates.

Remarks

- **Coding requirements**: Easy implementation in MATLAB, MATHEMATICA, S-PLUS, R. R implementation is available from R package *dimRed* [2].
- **Storage requirements**: Although it may seem to require $O(N^2)$ storage as it involves solving a eigenvalue problem for $N \times N$ matrix, the actual storage required is proportional to $k \cdot N$.
- **Computational complexity**: For our sparse matrix, the cost of each product is $2kN$, whereas the cost is $2N^2$ for a full matrix, making the overall cost of the sparse version $O(kN^2)$.

Remarks (Contd.)

Building Hessian estimator

- Create a matrix with $1 + d + \frac{(d+1)d}{2}$ columns
- The first one is a vector of ones, the first d of the remaining columns are columns of U, and finally the last $\frac{d(d+1)}{2}$ consist of the various cross products and squares of those d columns. For example, in case of $d = 2$
  $X^i = [1, U_{.,1}, U_{.,2}, (U^2_{.,1}), (U^2_{.,2}), (U_{.,1} \times U_{.,2})]$
- Perform the usual Gram–Schmidt orthonormalization process on the matrix X^i, yielding a matrix $\tilde{X}^i$ with orthonormal columns.
- Define H^i by extracting the last $\frac{d(d+1)}{2}$ columns and transposing.

$$(H^i)_{r,l} = (\tilde{X}^i)_{l,1+d+r}$$

Remarks (Contd.)

Basis for the null space

- Let V be the $N \times d$ matrix of eigenvectors built from the nonconstant eigenvectors associated with the $(d+1)$ smallest eigenvalues.
- let $V_{l,r}$ denote the lth entry in the rth eigenvector of $\mathscr{H}$.
- Define the matrix $(R)rs = \sum_{j \in N_1} V_{j,r} V_{j,s}$.
- Desired $N \times d$ matrix of embedding coordinates is obtained from $W = V \cdot R^{-1/2}$.

Hessian Locally Linear Embedding Algorithm Donoho-Grimes [3]

Algorithm 1: Hessian Locally Linear Embedding (HLLE)

Data: Dataset $X \in \mathbb{R}^{n \times D}$
Result: Low-dimensional embedding coordinates $Y \in \mathbb{R}^{n \times d}$
Parameters: k: the size of Neighborhoods and d: the dimension of the parameter space

Step 1: Find Nearest Neighbors for each data point so you have a collection of N_i from (i = 1,...,n). This can be done via KNN or the ϵ-Neighborhood algorithm

For (i = 1 to n)

- **Step 2:** Based on N_i construct a matrix M^i of size $(k+1) \times D$ and then center the matrix by subtracting avg of columns in M^i
- **Step 3:** Obtain the tangent coordinates: perform SVD on M^i then the first d columns of U give you the tangent coordinates of the points in N_i
- **Step 4:** Construct the local Hessian estimator: construct a new matrix $A^i = [1 \quad U \quad U^2 \quad (U_{:,j} \odot U_{:,m})]$ of size $(k+1) \times (1 + d + d(d+1)/2)$ where $j \neq m$ and $1 \leq j \leq m \leq d$. Then perform Gram-Schmidt process and extract Q and save the last $d(d+1)/2$ columns as a H^i of size $(k+1) \times d(d+1)/2$

Step 5: Construct the sparse matrix S of size $n \times (nd(d+1)/2)$ from the list of H^i and then calculate $H = S \times S^\top$ of size $n \times n$

Step 6: Perform Eigen Decomposition on H and extract the last $(d+1)$ eigenvectors corresponding to the smallest $(d+1)$ eigenvalues then drop the constant vector and then you have the d dimensional Low-dimensional embedding $Y \in \mathbb{R}^{n \times d}$

Swiss Roll with hole and 1 turn Dimension Reduction (DR) using HLLE

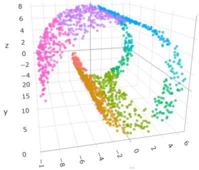

Figure 1: 3D Swiss Roll

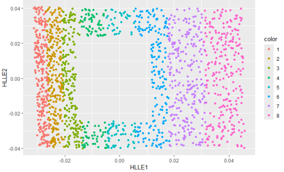

Figure 2: HLLE DR with knn $= 10$

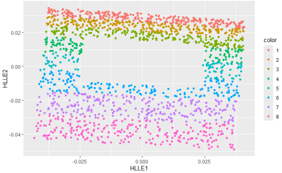

Figure 3: HLLE DR with ϵ neighborhood $\epsilon = 1.5$, adj $= 0.5$

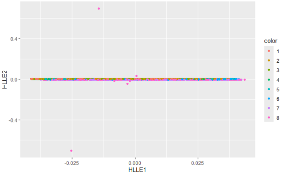

Figure 4: HLLE DR with knn $= 50$

Swiss Roll with hole and 1 turn Dimension Reduction (DR) comparisons

Figure 5: HLLE DR with knn = 10

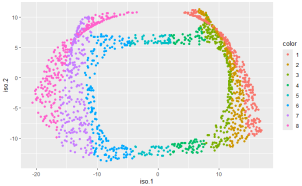

Figure 6: Isomap DR

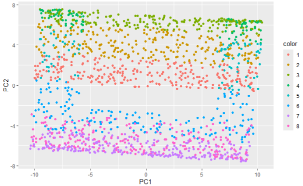

Figure 7: PCA DR

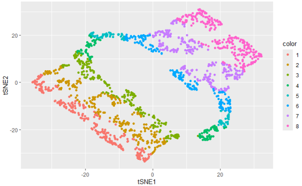

Figure 8: tNSE DR

Swiss Roll with hole and 2 turns Dimension Reduction using HLLE

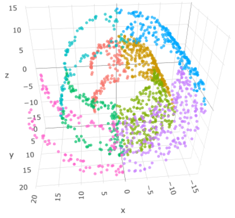

Figure 9: 3D Swiss Roll

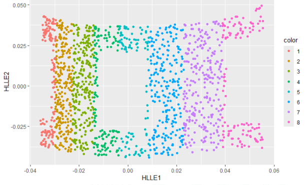

Figure 10: HLLE DR with knn $= 9$

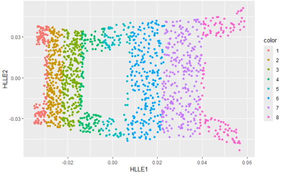

Figure 11: HLLE DR with ϵ neighborhood $\epsilon = 2.9$, adj $= 0.2$

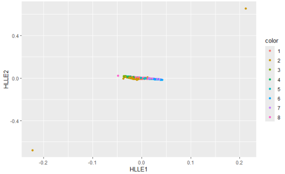

Figure 12: HLLE DR with knn $= 20$

Swiss Roll with hole and 2 turns Dimension Reduction comparisons

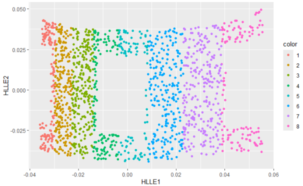

Figure 13: HLLE DR with knn = 9

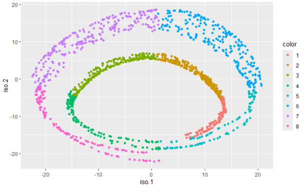

Figure 14: Isomap DR

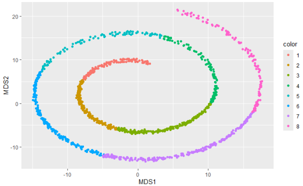

Figure 15: MDS DR

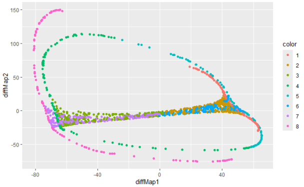

Figure 16: Diffusion Map DR

3D face mesh Dimension Reduction (DR) comparisons

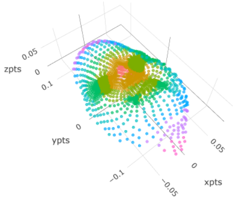

Figure 17: 3D Mesh of a face

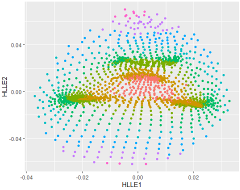

Figure 18: HLLE DR with knn = 14

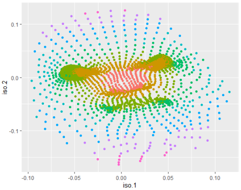

Figure 19: Isomap DR

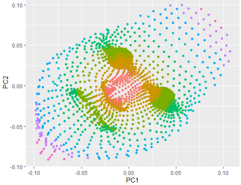

Figure 20: PCA DR

309 Chest X-Rays of Covid, Normal, Pneumonia of 28x28 images [4]

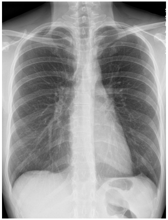

Figure 21: Xray of Covid Chest

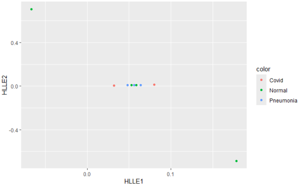

Figure 22: HLLE DR with knn $= 10$

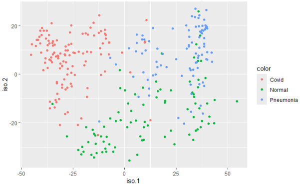

Figure 23: Isomap DR

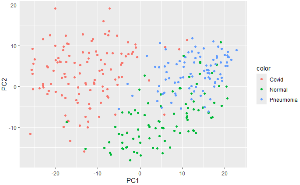

Figure 24: PCA DR

1. Introduction
2. Methodology
3. Our Implementation and Results
4. **HLLE Extensions**
5. Conclusion
6. References

Robust Hessian Locally Linear Embedding by Xing, Du, Wang [5]

- One thing that was found during implementation of the HLLE algorithm is that the space collapses fairly easily depending on the neighborhood size or if the data itself contains a lot of noise and/or outliers
- One way to counter this was discussed in the paper on Robust HLLE by Xing[5]
- In this paper the authors introduce a slightly different framework for calculating the Hessian estimator that is claimed to be faster than Donoho's implementation of the Hessian estimator
- They also introduce the use of robust PCA for obtaining the local tangent coordinates
- They introduce 3 steps for the Robust HLLE by first employing an algorithm to identify outliers, second locally smooth the data, and lastly implement RHLLE

Fast Outlier Identifying Algorithm

Algorithm 2: Fast Outlier Identifying Algorithm

Data: Data points x_j^i, Number of iterations L, Threshold ϵ

Step 1: Update $\bar{x}_i$ with the weighted sample mean vector:

Initialization: $\bar{x}_i^{(0)} \leftarrow \frac{1}{k}\sum_{j=1}^{k} x_j^i \quad l \leftarrow 0$;

while $\|\bar{x}_i^{(l+1)} - \bar{x}_i^{(l)}\|_2^2 > \epsilon$ *and* $l < L$ **do**

Compute weights $(w_j^i)^l = \frac{\exp\left(-\|x_j^i - \bar{x}_i\|_{2/\sigma}\right)}{\sum_{j=1}^{k}\exp\left(-\|x_j^i - \bar{x}_i\|_2^2/\sigma\right)}$ (j = 1, . . . , k) ;

Update mean $\bar{x}_i^{(l+1)} \leftarrow \sum_{j=1}^{k}(w_j^i)^l(\bar{x}_i^j)^{(l)}$;

Step 2: Update weights based on Huber function: for $j = 1$ *to* k **do**

Compute projection error ϵ_j^i via weighted PCA ;

Compute weight w_j^i:

$$w_j^i = w(\epsilon_j^i) = \frac{\rho'(\epsilon_j^i)}{\epsilon_j^i} = \begin{cases} 1, & \text{if } \epsilon_j^i \leq \frac{1}{2}c \\ \frac{c}{2\epsilon_j^i}, & \text{if } \epsilon_j^i > \frac{1}{2}c \end{cases}$$

Note: σ represents the mean squared Euclidean distance of k neighbors.
c > 0 and user-defined, and

$$\rho(\epsilon) = \begin{cases} \frac{1}{2}\epsilon^2, & \text{if } |\epsilon| \leq c \\ c(|\epsilon| - \frac{1}{2}c), & \text{if } |\epsilon| > c \end{cases}$$

is the Huber function

Robust HLLE Algorithm [5]

Algorithm 3: Robust Hessian Locally Linear Embedding (RHLLE)

Data: Dataset $X^{on} = [x_1^{on}, x_2^{on}, ..., x_{Nall}^{on}] \in \mathbb{R}^{D \times N}$
Result: Low-dimensional embedding coordinates Y

Step 1: Fast Outlier Identifying Algorithm
Remove outliers from X^{on} using the fast outlier identifying algorithm
Obtain dataset $X^{or} = [x_1^{or}, x_2^{or}, ..., x_N^{or}]$ without outliers

Step 2: Noise Reduction Using Local Linear Smoothing
Reduce noise in X^{or} using local linear smoothing via weighted PCA and then project x_i to an approximate tangent subspace
Obtain noise-reduced dataset $X^{nr} = [x_1, x_2, ..., x_N]$

Step 3: Robust Hessian Locally Linear Embedding

- Find k nearest neighbors for each point $x_i \in X^{nr}$
- Compute total reliability scores W_i^p for each local patch $N(x_i)$
  Determine reliable local patch subset RLP
- Compute local tangent coordinates and local Hessian operator for each local patch $N(x_i) \in RLP$
- Compute weighted global functional $\bar{H}_w$
- Perform eigenanalysis of $\bar{H}_w$ to obtain low-dimensional embedding coordinates

Example of Robust HLLE from Xing Paper[5]

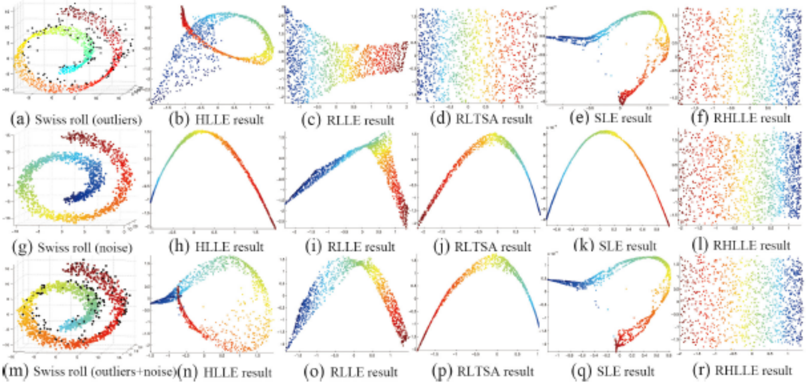

Figure 25: The first row shows the Swiss roll dataset with outliers and the embedding results of different algorithms;The second row shows the Swiss roll dataset with noise and the embedding results of different algorithms; The third row shows the Swiss roll dataset with both outliers and noise and the embedding results of different algorithms.

Conclusion

- HLLE is great for representing smooth and non-noisy data to lower dimensions but is sensitive with the presence of outliers and/or noise in the data.
- A key takeaway is that the HLLE approach assumes local isometry compared to ISOMAP that assumes global isometry.
- Further a useful advantage of HLLE is that it can make use of sparse eigenproblems.
- Another remark is that calculation of a local hessian for each data point using a for loop can be further sped up by parallel computing.
- A drawback of the HLLE approach is that it requires estimation of the second derivative, which is known to be numerically noisy or difficult in very high-dimensional data samples.

References

[1] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.290.5500.2319

[2] G. Kraemer, M. Reichstein, and M. D. Mahecha, "dimRed and coRanking—unifying dimensionality reduction in r," *The R Journal*, vol. 10, no. 1, pp. 342–358, 2018, coRanking version 0.2.6. [Online]. Available: https://journal.r-project.org/archive/2018/RJ-2018-039/index.html

[3] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, 2003. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1031596100

[4] P. Raikote, "Covid-19 image dataset," https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset, 2020, accessed: 15 April 2024.

[5] X. Xing, S. Du, and K. Wang, "Robust hessian locally linear embedding techniques for high-dimensional data," *Algorithms*, vol. 9, no. 2, 2016. [Online]. Available: https://www.mdpi.com/1999-4893/9/2/36

[6] J. Wang, *Geometric Structure of High-Dimensional Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. [Online]. Available: https://doi.org/10.1007/978-3-642-27497-8_3

[7] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.290.5500.2323