Locally Linear Embedding

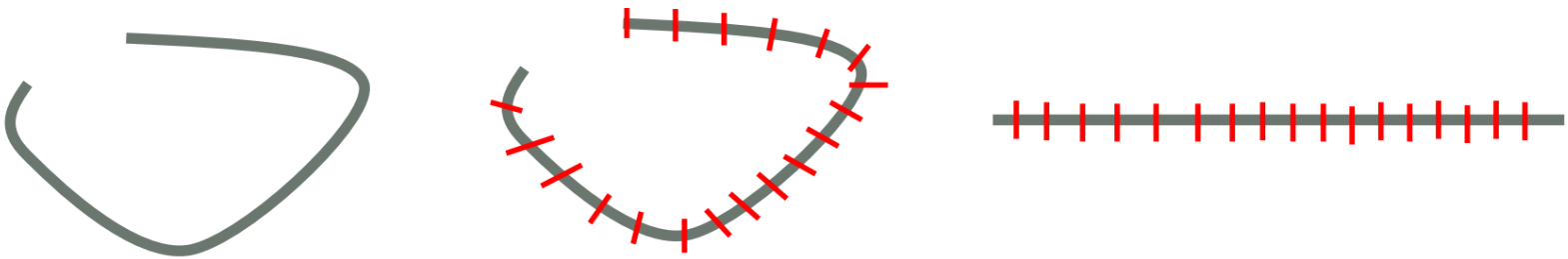
Statistical Machine Learning (ENGG*6600*02)

School of Engineering,
University of Guelph, ON, Canada

Course Instructor: Benyamin Ghojogh
Summer 2023

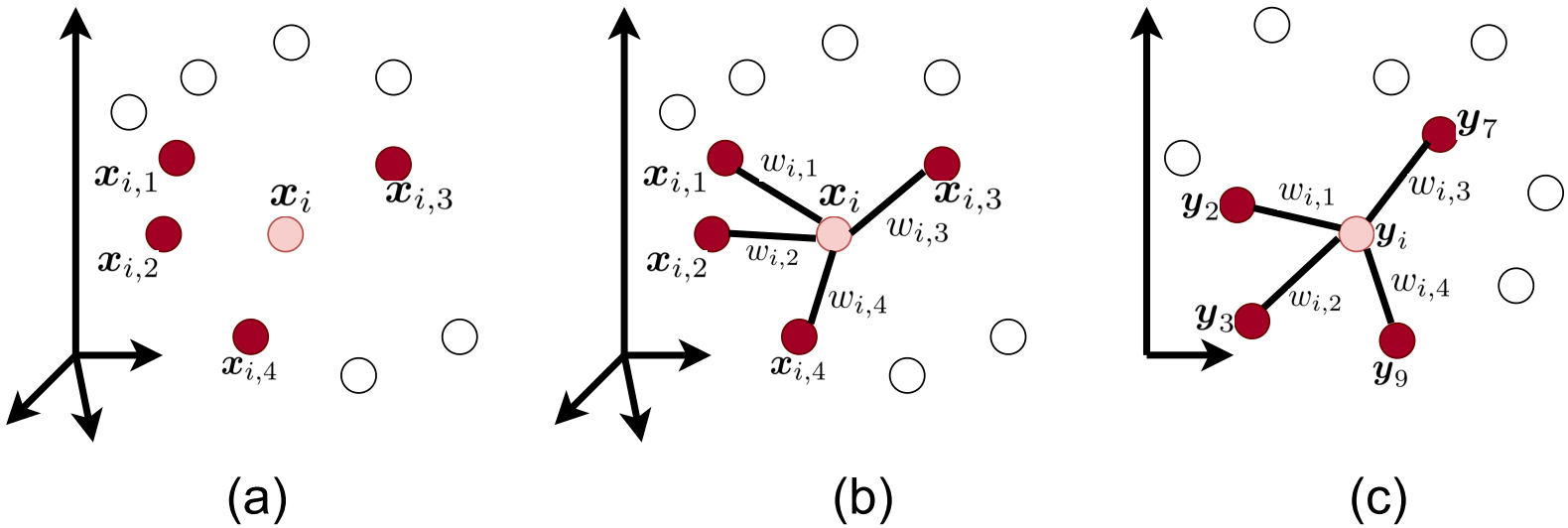
Locally Linear Embedding

- **Locally Linear Embedding (LLE)** (2000) [1, 2] is a **nonlinear spectral dimensionality reduction** method [3] which can be used for manifold embedding and feature extraction [4].
- LLE tries to **preserve the local structure of data in the embedding space**. In other words, the close points in the high-dimensional input space should also be close to each other in the low-dimensional embedding space. By this **local fitting**, hopefully the far points in the input space also fall far away from each other in the embedding space. This idea of **fitting locally and thinking globally** is the main idea of LLE [5, 6, 7, 8].
- In another perspective, the idea of local fitting by LLE is similar to idea of **piece-wise spline regression** [9]. LLE unfolds the nonlinear manifold by locally unfolding of manifold piece by piece and it hopes that these local unfoldings result in a suitable total manifold unfolding.



Locally Linear Embedding

- LLE, first proposed in [1] and developed in [10, 6], has three steps [11].
  1. First, it finds k-Nearest Neighbors (kNN) graph of all training points.
  2. Then, it tries to find weights for reconstructing every point by its neighbors, using linear combination.
  3. Using the same found weights, it embeds every point by a linear combination of its embedded neighbors.
- The main idea of LLE is using the same reconstruction weights in the lower dimensional embedding space as in the high dimensional input space.



k-Nearest Neighbors

k-Nearest Neighbors

- A kNN graph is formed using pairwise Euclidean distance between the data points. Therefore, every data point has k neighbors.
- Let $\boldsymbol{x}_{ij} \in \mathbb{R}^d$ denote the j-th neighbor of $\boldsymbol{x}_i$ and let the matrix $\mathbb{R}^{d \times k} \ni \boldsymbol{X}_i := [\boldsymbol{x}_{i1}, \ldots, \boldsymbol{x}_{ik}]$ include the k neighbors of $\boldsymbol{x}_i$.
- We assume that k is large enough so that the kNN graph is connected.

Linear Reconstruction by the Neighbors

Linear Reconstruction by the Neighbors

- In the second step, we find the weights for linear reconstruction of every point by its kNN. The optimization for this linear reconstruction in the high dimensional input space is formulated as:

$$\begin{aligned} \underset{\widetilde{\boldsymbol{W}}}{\text{minimize}} \quad & \varepsilon(\widetilde{\boldsymbol{W}}) := \sum_{i=1}^{n} \left|\left| \boldsymbol{x}_i - \sum_{j=1}^{k} \widetilde{w}_{ij} \boldsymbol{x}_{ij} \right|\right|_2^2, \\ \text{subject to} \quad & \sum_{j=1}^{k} \widetilde{w}_{ij} = 1, \quad \forall i \in \{1, \dots, n\}, \end{aligned} \tag{1}$$

  where $\mathbb{R}^{n \times k} \ni \widetilde{\boldsymbol{W}} := [\widetilde{\boldsymbol{w}}_1, \dots, \widetilde{\boldsymbol{w}}_n]^\top$ includes the weights, $\mathbb{R}^k \ni \widetilde{\boldsymbol{w}}_i := [\widetilde{w}_{i1}, \dots, \widetilde{w}_{ik}]^\top$ includes the weights of linear reconstruction of the i-th data point using its k neighbors, and $\boldsymbol{x}_{ij} \in \mathbb{R}^d$ is the j-th neighbor of the i-th data point.

- The constraint $\sum_{j=1}^{k} \widetilde{w}_{ij} = 1$ means that the weights of linear reconstruction sum to one for every point. Note that the fact that some weights may be **negative** causes the problem of **explosion** of some weights because very large positive and negative weights can cancel each other to have a total sum of one. However, this problem does not occur because, as we will see, the solution to this optimization problem has a **closed form**; thus, weights do not explode. If the solution was found **iteratively**, the weights would grow and explode gradually [12].

Linear Reconstruction by the Neighbors

- We can restate the objective $\varepsilon(\widetilde{\boldsymbol{W}}) = \sum_{i=1}^{n} \left|\left| \boldsymbol{x}_i - \sum_{j=1}^{k} \widetilde{w}_{ij} \boldsymbol{x}_{ij} \right|\right|_2^2$ as:

$$\varepsilon(\widetilde{\boldsymbol{W}}) = \sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{X}_i \widetilde{\boldsymbol{w}}_i||_2^2. \tag{2}$$

- The constraint $\sum_{j=1}^{k} \widetilde{w}_{ij} = 1$ implies that $\mathbf{1}^\top \widetilde{\boldsymbol{w}}_i = 1$; therefore, $\boldsymbol{x}_i = \boldsymbol{x}_i \mathbf{1}^\top \widetilde{\boldsymbol{w}}_i$.
- We can simplify the term in $\varepsilon(\widetilde{\boldsymbol{W}})$ as:

$$\begin{aligned} ||\boldsymbol{x}_i - \boldsymbol{X}_i \widetilde{\boldsymbol{w}}_i||_2^2 &= ||\boldsymbol{x}_i \mathbf{1}^\top \widetilde{\boldsymbol{w}}_i - \boldsymbol{X}_i \widetilde{\boldsymbol{w}}_i||_2^2 = ||(\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)\, \widetilde{\boldsymbol{w}}_i||_2^2 \\ &= \widetilde{\boldsymbol{w}}_i^\top (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)^\top (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)\, \widetilde{\boldsymbol{w}}_i = \widetilde{\boldsymbol{w}}_i^\top \boldsymbol{G}_i\, \widetilde{\boldsymbol{w}}_i, \end{aligned}$$

  where $\boldsymbol{G}_i$ is a Gram matrix defined as:

$$\mathbb{R}^{k \times k} \ni \boldsymbol{G}_i := (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)^\top (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i). \tag{3}$$

- Finally, Eq. (1) can be rewritten as:

$$\begin{aligned} \underset{\{\widetilde{\boldsymbol{w}}_i\}_{i=1}^n}{\text{minimize}} \quad & \sum_{i=1}^{n} \widetilde{\boldsymbol{w}}_i^\top \boldsymbol{G}_i\, \widetilde{\boldsymbol{w}}_i, \\ \text{subject to} \quad & \mathbf{1}^\top \widetilde{\boldsymbol{w}}_i = 1, \quad \forall i \in \{1, \ldots, n\}. \end{aligned} \tag{4}$$

Linear Reconstruction by the Neighbors

- We had:

$$\underset{\{\widetilde{\boldsymbol{w}}_i\}_{i=1}^n}{\text{minimize}} \quad \sum_{i=1}^{n} \widetilde{\boldsymbol{w}}_i^\top \boldsymbol{G}_i \, \widetilde{\boldsymbol{w}}_i,$$
$$\text{subject to} \quad \boldsymbol{1}^\top \widetilde{\boldsymbol{w}}_i = 1, \quad \forall i \in \{1, \dots, n\}.$$

- The Lagrangian for Eq. (4) is [13]:

$$\mathcal{L} = \sum_{i=1}^{n} \widetilde{\boldsymbol{w}}_i^\top \boldsymbol{G}_i \, \widetilde{\boldsymbol{w}}_i - \sum_{i=1}^{n} \lambda_i \, (\boldsymbol{1}^\top \widetilde{\boldsymbol{w}}_i - 1).$$

- Setting the derivative of Lagrangian to zero gives:

$$\mathbb{R}^k \ni \frac{\partial \mathcal{L}}{\partial \widetilde{\boldsymbol{w}}_i} = 2\boldsymbol{G}_i \widetilde{\boldsymbol{w}}_i - \lambda_i \boldsymbol{1} \overset{\text{set}}{=} \boldsymbol{0} \implies \widetilde{\boldsymbol{w}}_i = \frac{1}{2} \boldsymbol{G}_i^{-1} \lambda_i \boldsymbol{1} = \frac{\lambda_i}{2} \boldsymbol{G}_i^{-1} \boldsymbol{1}. \tag{5}$$

$$\mathbb{R} \ni \frac{\partial \mathcal{L}}{\partial \lambda} = \boldsymbol{1}^\top \widetilde{\boldsymbol{w}}_i - 1 \overset{\text{set}}{=} 0 \implies \boldsymbol{1}^\top \widetilde{\boldsymbol{w}}_i = 1. \tag{6}$$

- Using Eqs. (5) and (6), we have:

$$\frac{\lambda_i}{2} \boldsymbol{1}^\top \boldsymbol{G}_i^{-1} \boldsymbol{1} = 1 \implies \lambda_i = \frac{2}{\boldsymbol{1}^\top \boldsymbol{G}_i^{-1} \boldsymbol{1}}. \tag{7}$$

- Using Eqs. (5) and (7), we have:

$$\widetilde{\boldsymbol{w}}_i = \frac{\lambda_i}{2} \boldsymbol{G}_i^{-1} \boldsymbol{1} = \frac{\boldsymbol{G}_i^{-1} \boldsymbol{1}}{\boldsymbol{1}^\top \boldsymbol{G}_i^{-1} \boldsymbol{1}}. \tag{8}$$

Linear Reconstruction by the Neighbors

- According to Eq. (3):

$$\mathbb{R}^{k\times k} \ni \boldsymbol{G}_i := (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i)^\top (\boldsymbol{x}_i \mathbf{1}^\top - \boldsymbol{X}_i).$$

  the rank of matrix $\boldsymbol{G}_i \in \mathbb{R}^{k\times k}$ is at most equal to $\min(k, d)$.
- If $d < k$, then $\boldsymbol{G}_i$ is singular and $\boldsymbol{G}_i$ should be replaced by $\boldsymbol{G}_i + \epsilon \boldsymbol{I}$ where ϵ is a small positive number.
- Usually, the data are high dimensional (so $k \ll d$) as in images and thus if $\boldsymbol{G}_i$ is full rank, we will not have any problem with inverting it.
- Strengthening the main diagonal of $\boldsymbol{G}$ is referred to as **regularization in LLE** [14]. This numerical technique is widely used in manifold and subspace learning (e.g., see [15]).

Linear Embedding

Linear Embedding

- In the second step, we found the weights for linear reconstruction in the high dimensional input space. In the third step, we embed data in the low dimensional embedding space using the same weights as in the input space. This linear embedding can be formulated as the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{Y}}{\text{minimize}} \quad & \sum_{i=1}^{n} \left|\left| \boldsymbol{y}_i - \sum_{j=1}^{n} w_{ij} \boldsymbol{y}_j \right|\right|_2^2, \\ \text{subject to} \quad & \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^\top = \boldsymbol{I}, \\ & \sum_{i=1}^{n} \boldsymbol{y}_i = \boldsymbol{0}, \end{aligned} \tag{9}$$

  where $\boldsymbol{I}$ is the identity matrix, the rows of $\mathbb{R}^{n \times p} \ni \boldsymbol{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]^\top$ are the embedded data points (stacked row-wise), $\boldsymbol{y}_i \in \mathbb{R}^p$ is the i-th embedded data point, and w_{ij} is the weight obtained from the linear reconstruction if $\boldsymbol{x}_j$ is a neighbor of $\boldsymbol{x}_i$ and zero otherwise:

$$w_{ij} := \begin{cases} \widetilde{w}_{ij} & \text{if } \boldsymbol{x}_j \in k\text{NN}(\boldsymbol{x}_i) \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

- The second constraint in Eq. (9) ensures the zero mean of embedded data points. The first and second constraints together satisfy having unit covariance for the embedded points.

Linear Embedding

- Suppose $\mathbb{R}^n \ni \boldsymbol{w}_i := [w_{i1}, \ldots, w_{in}]^\top$ and let $\mathbb{R}^n \ni \mathbf{1}_i := [0, \ldots, 1, \ldots, 0]^\top$ be the vector whose i-th element is one and other elements are zero. The objective function in Eq. (9) can be restated as:

$$\sum_{i=1}^{n} \left|\left| \boldsymbol{y}_i - \sum_{j=1}^{n} w_{ij} \boldsymbol{y}_j \right|\right|_2^2 = \sum_{i=1}^{n} ||\boldsymbol{Y}^\top \mathbf{1}_i - \boldsymbol{Y}^\top \boldsymbol{w}_i||_2^2,$$

  which can be stated in matrix form:

$$\sum_{i=1}^{n} ||\boldsymbol{Y}^\top \mathbf{1}_i - \boldsymbol{Y}^\top \boldsymbol{w}_i||_2^2 = ||\boldsymbol{Y}^\top \boldsymbol{I} - \boldsymbol{Y}^\top \boldsymbol{W}^\top||_F^2 = ||\boldsymbol{Y}^\top (\boldsymbol{I} - \boldsymbol{W})^\top||_F^2, \tag{11}$$

  where the i-th row of $\mathbb{R}^{n \times n} \ni \boldsymbol{W} := [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n]^\top$ includes the weights for the i-th data point and $||.||_F$ denotes the Frobenius norm of matrix.
- The Eq. (11) is simplified as:

$$\begin{aligned} ||\boldsymbol{Y}^\top (\boldsymbol{I} - \boldsymbol{W})^\top||_F^2 &= \mathbf{tr}\big((\boldsymbol{I} - \boldsymbol{W}) \boldsymbol{Y} \boldsymbol{Y}^\top (\boldsymbol{I} - \boldsymbol{W})^\top\big) = \mathbf{tr}\big(\boldsymbol{Y}^\top (\boldsymbol{I} - \boldsymbol{W})^\top (\boldsymbol{I} - \boldsymbol{W}) \boldsymbol{Y}\big) \\ &= \mathbf{tr}(\boldsymbol{Y}^\top \boldsymbol{M} \boldsymbol{Y}), \end{aligned} \tag{12}$$

  where $\mathbf{tr}(.)$ denotes the trace of matrix and:

$$\mathbb{R}^{n \times n} \ni \boldsymbol{M} := (\boldsymbol{I} - \boldsymbol{W})^\top (\boldsymbol{I} - \boldsymbol{W}). \tag{13}$$

- Note that $(\boldsymbol{I} - \boldsymbol{W})$ is the Laplacian of matrix $\boldsymbol{W}$ because the columns of $\boldsymbol{W}$, which are $\boldsymbol{w}_i$'s, add to one (for the constraint used in Eq. (1)). Hence, according to Eq. (13), the matrix $\boldsymbol{M}$ can be considered as the gram matrix over the Laplacian of weight matrix.

Linear Embedding

- Finally, Eq. (9) can be rewritten as:

$$\begin{aligned} \underset{\boldsymbol{Y}}{\text{minimize}} \quad & \mathbf{tr}(\boldsymbol{Y}^\top \boldsymbol{M} \boldsymbol{Y}), \\ \text{subject to} \quad & \frac{1}{n} \boldsymbol{Y}^\top \boldsymbol{Y} = \boldsymbol{I}, \\ & \boldsymbol{Y}^\top \mathbf{1} = \mathbf{0}, \end{aligned} \tag{14}$$

  where the dimensionality of $\mathbf{1}$ and $\mathbf{0}$ are $\mathbb{R}^n$ and $\mathbb{R}^p$, respectively.
- The second constraint will be satisfied implicitly (see our tutorial paper "Locally linear embedding and its variants: Tutorial and survey" [16] for proof). Therefore, if we ignore the second constraint, the Lagrangian for Eq. (14) is [13]:

$$\mathcal{L} = \mathbf{tr}(\boldsymbol{Y}^\top \boldsymbol{M} \boldsymbol{Y}) - \mathbf{tr}\big(\boldsymbol{\Lambda}^\top (\frac{1}{n} \boldsymbol{Y}^\top \boldsymbol{Y} - \boldsymbol{I})\big),$$

  where $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix including the Lagrange multipliers.
- Equating derivative of $\mathcal{L}$ to zero gives us:

$$\mathbb{R}^{n \times p} \ni \frac{\partial \mathcal{L}}{\partial \boldsymbol{Y}} = 2\boldsymbol{M}\boldsymbol{Y} - \frac{2}{n} \boldsymbol{Y} \boldsymbol{\Lambda} \overset{\text{set}}{=} \mathbf{0} \implies \boldsymbol{M}\boldsymbol{Y} = \boldsymbol{Y}(\frac{1}{n}\boldsymbol{\Lambda}), \tag{15}$$

  which is the eigenvalue problem for $\boldsymbol{M}$ [17]. Therefore, the columns of $\boldsymbol{Y}$ are the eigenvectors of $\boldsymbol{M}$ where eigenvalues are the diagonal elements of $(1/n)\boldsymbol{\Lambda}$.
- As Eq. (14) is a minimization problem, the columns of $\boldsymbol{Y}$ should be sorted from the smallest to largest eigenvalues.

Linear Embedding

- Recall:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{M} := (\boldsymbol{I} - \boldsymbol{W})^\top (\boldsymbol{I} - \boldsymbol{W}).$$

- Recall that we explained $(\boldsymbol{I} - \boldsymbol{W})$ in $\boldsymbol{M}$ is the Laplacian matrix for the weights $\boldsymbol{W}$. It is well-known in linear algebra and graph theory that if a graph has k disjoint connected parts, its Laplacian matrix has k zero eigenvalues (see [18, Theorem 3.10] and [19, 20]).
- As the kNN graph, or $\boldsymbol{W}$, is a connected graph, $(\boldsymbol{I} - \boldsymbol{W})$ has one zero eigenvalue whose eigenvector is $\boldsymbol{1} = [1, 1, \ldots, 1]^\top$.
- After sorting the eigenvectors from smallest to largest eigenvalues, we ignore the first eigenvector having zero eigenvalue and take the p smallest eigenvectors of $\boldsymbol{M}$ with non-zero eigenvalues as the columns of $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$.

Lemma

The fact that we have the eigenvector $\boldsymbol{1}$ *with zero eigenvalue implicitly ensures that* $\sum_{i=1}^{n} \boldsymbol{y}_i = \boldsymbol{Y}^\top \boldsymbol{1} = \boldsymbol{0}$ *which was the second constraint.*

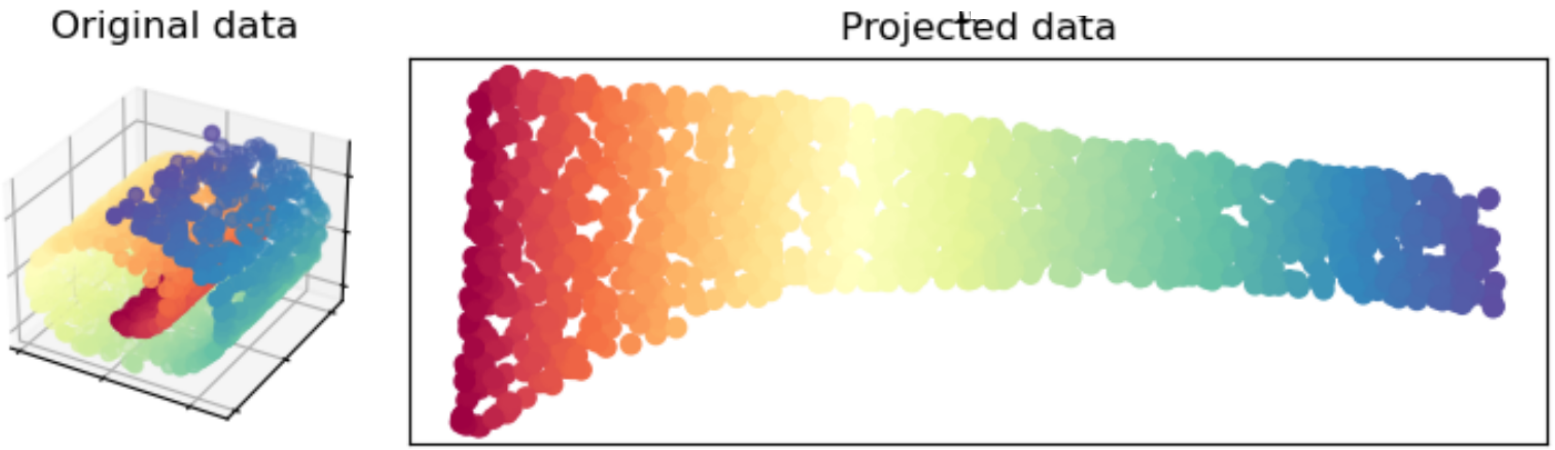
Proof.

See our tutorial paper: "Locally linear embedding and its variants: Tutorial and survey" [16] for proof. □

Examples of LLE Embedding

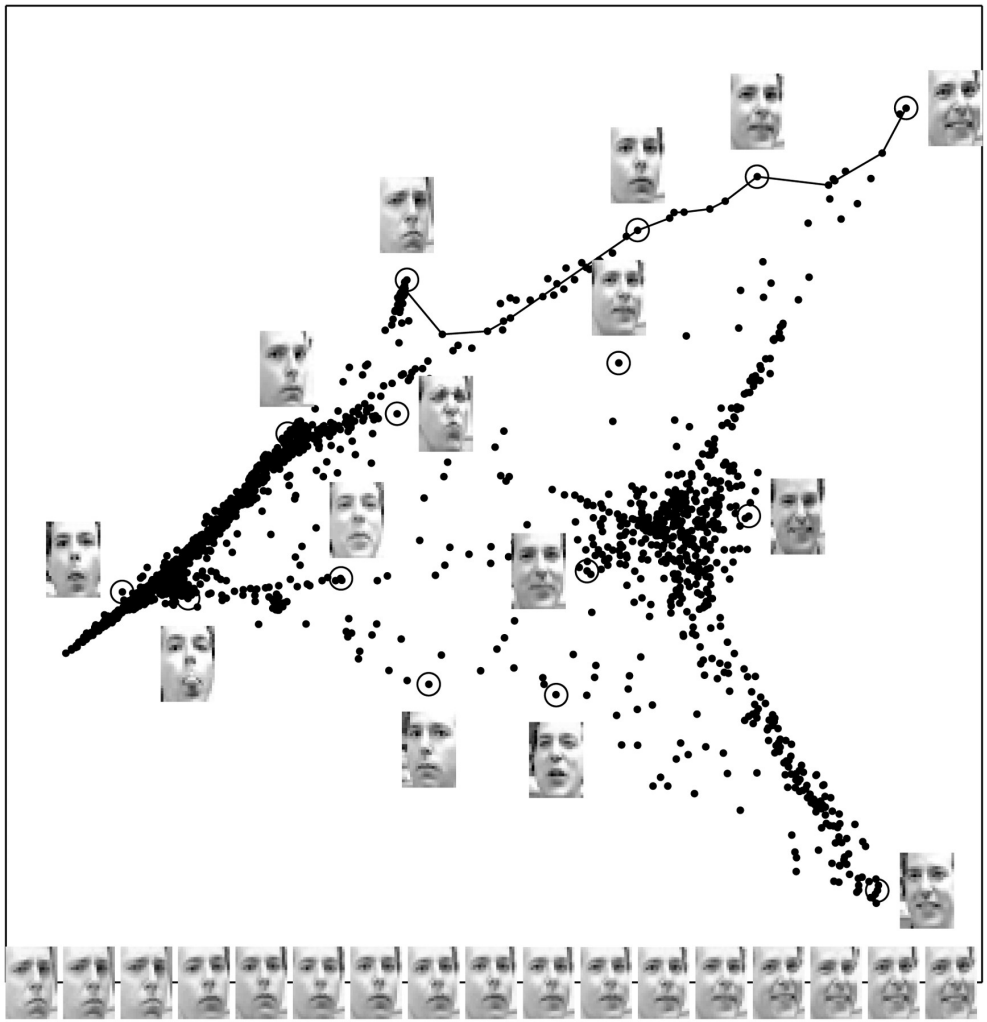
Examples of LLE Embedding

Swiss roll:



Examples of LLE Embedding

Frey facial dataset:



Acknowledgment

- Some slides are based on our tutorial paper: "Locally linear embedding and its variants: Tutorial and survey" [16]
- Some slides of this slide deck are inspired by teachings of Prof. Ali Ghodsi at University of Waterloo, Department of Statistics.
- The code of LLE in my GitHub: `https://github.com/bghojogh/Generative-LLE`
- LLe in sklearn: `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.LocallyLinearEmbedding.html`

References

[1] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[2] J. Chen and Y. Liu, "Locally linear embedding: a survey," *Artificial Intelligence Review*, vol. 36, no. 1, pp. 29–48, 2011.

[3] L. K. Saul, K. Q. Weinberger, F. Sha, J. Ham, and D. D. Lee, "Spectral methods for dimensionality reduction.," *Semi-supervised learning*, vol. 3, 2006.

[4] B. Ghojogh, M. N. Samad, S. A. Mashhadi, T. Kapoor, W. Ali, F. Karray, and M. Crowley, "Feature selection and feature extraction in pattern analysis: A literature review," *arXiv preprint arXiv:1905.02845*, 2019.

[5] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervised learning of nonlinear manifolds," tech. rep., Technical Report CIS-02-18, University of Pennsylvania, 2002.

[6] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of machine learning research*, vol. 4, no. Jun, pp. 119–155, 2003.

[7] K. Yotov, K. Pingali, and P. Stodghill, "Think globally, search locally," in *Proceedings of the 19th annual international conference on Supercomputing*, pp. 141–150, 2005.

[8] H.-T. Wu, N. Wu, *et al.*, "Think globally, fit locally under the manifold setup: Asymptotic analysis of locally linear embedding," *The Annals of Statistics*, vol. 46, no. 6B, pp. 3805–3837, 2018.

References (cont.)

[9] L. C. Marsh and D. R. Cormier, *Spline regression models*. No. 137, Sage, 2001.

[10] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding," tech. rep., 2000.

[11] A. Ghodsi, "Dimensionality reduction a short tutorial," tech. rep., Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada, 2006.

[12] B. Ghojogh, F. Karray, and M. Crowley, "Locally linear image structural embedding for image structure manifold learning," in *International Conference on Image Analysis and Recognition*, pp. 126–138, Springer, 2019.

[13] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[14] G. Daza-Santacoloma, C. D. Acosta-Medina, and G. Castellanos-Domínguez, "Regularization parameter choice in locally linear embedding," *neurocomputing*, vol. 73, no. 10-12, pp. 1595–1605, 2010.

[15] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers, "Fisher discriminant analysis with kernels," in *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pp. 41–48, Ieee, 1999.

[16] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Locally linear embedding and its variants: Tutorial and survey," *arXiv preprint arXiv:2011.10925*, 2020.

References (cont.)

[17] B. Ghojogh, F. Karray, and M. Crowley, "Eigenvalue and generalized eigenvalue problems: Tutorial," *arXiv preprint arXiv:1903.11240*, 2019.

[18] A. Marsden, "Eigenvalues of the Laplacian and their relationship to the connectedness of a graph," *University of Chicago, REU*, 2013.

[19] M. Polito and P. Perona, "Grouping and dimensionality reduction by locally linear embedding," in *Advances in neural information processing systems*, pp. 1255–1262, 2002.

[20] S. Ahmadizadeh, I. Shames, S. Martin, and D. Nešić, "On eigenvalues of Laplacian matrix for a class of directed signed graphs," *Linear Algebra and its Applications*, vol. 523, pp. 281–306, 2017.