Explicit Non-Linear Dimensionality Reduction

Pierre Visconti

Department of Mathematics, Walla Walla University

Introduction

Real world data is often complex and of high dimension. The high dimensionality of the data poses problems by the high computational cost in analyzing the data, and being difficult to understand. Dimensionality reduction is the idea that when the data is of artificially high dimension, it can be reduced to a lower dimension while still retaining the intrinsic properties of the data. Linear techniques such as Principal Components Analysis (PCA) have been well known for some time, but cannot handle complex non-linear data. Non-linear methods, known as Manifold learning, can handle complex data, however most existing algorithms do not build an explicit model, meaning new incoming data samples must be added to the original data and the dimensionality reduction algorithm run again on the entire dataset, thereby defeating the intended purpose of lowering the overall computational complexity. The goal of this project is to explore a manifold learning algorithm that builds an explicit model and apply it as a pre-processing step in supervised machine learning algorithms to reduce the overall time and storage complexity of the algorithm.

Background

Given a data set $\mathcal{X} := \{x_1, x_2, ..., x_N\}$ in the high dimensional space $\mathbf{R}^n$, assume there exists an explicit polynomial mapping from $\mathcal{X}$ to its low dimensional representation $\mathcal{Y} := \{y_1, y_2, ..., y_N\}$ in $\mathbf{R}^m$. For a given data vector $x_i \in \mathcal{X}$, define the mapping $\phi : \mathbf{R}^n \to \mathbf{R}^{pn}$ as

$$\phi(x_i) = \begin{bmatrix} \overbrace{x_i \odot x_i \odot \cdots \odot x_i}^{p \text{ times}} \\ \vdots \\ x_i \odot x_i \\ x_i \end{bmatrix}.$$

The k-th component of $y_i \in \mathcal{Y}$ is defined as a polynomial of degree p with respect to x_i, such that

$$y_i^k = v_k^T \phi(x_i), \text{ where } v_k \in \mathbf{R}^{pn}$$

The polynomial mapping assumption can then be combined with a popular dimensionality reduction algorithm known as Locally Linear Embedding (LLE). LLE aims to preserve local relationships between data points based on the idea that if the dataset is sampled from a smooth manifold, then the neighbors of each point remain nearby in the low-dimensional space.

LLE first finds a set of linear coefficients that best reconstructs each data point x_i by its k-nearest neighbors. Using Euclidean Distance, the linear reconstruction weights $R_{ij}, i, j = 1, 2, ..., N$ are given by minimizing the sum of the squared distances between all the data points and their reconstructions, subject to the constraints that x_i is only reconstructed from its neighbors, and the weights for x_i must sum to 1.

$$R_{ij} = \underset{\sum_{j=1}^N R_{ij}=1}{\arg\min} \sum_{i=1}^N \left\| x_i - \sum_{j=1}^N R_{ij} x_j \right\|_2^2 \tag{i}$$

LLE then constructs a neighborhood preserving mapping by fixing the weights R_{ij}, while optimizing the coordinates y_i, subject to constraints that make the problem well-posed.

$$\min \sum_{i=1}^N \left\| y_i - \sum_{j=1}^N R_{ij} y_j \right\|_2^2 \tag{ii}$$

$$\text{s.t.} \quad \frac{1}{N} \sum_{i=1}^N y_i y_i^T = I_m$$

Applying the polynomial assumption to these concepts leads to the following algorithm.

Neighborhood Preserving Polynomial Embedding Algorithm

Given a data set $\mathcal{X} := \{x_1, x_2, ..., x_N\}$ in the high dimensional space $\mathbf{R}^n$, the NPPE algorithm finds an explicit polynomial mapping from $\mathcal{X}$ to its low dimensional representation $\mathcal{Y} := \{y_1, y_2, ..., y_N\}$ in $\mathbf{R}^m$.

Algorithm Overview

Inputs: Data matrix $X = [x_1 \ \ x_2 \ \cdots \ x_N]$ of size $n \times N$, the number k of nearest neighbors, the polynomial degree p, and the low dimensional space m.

1. Compute the linear weights R.
2. Compute the non-linear weights W.
3. Solve the generalized eigenvalue problem to get the eigenvectors $v_i, i = 1, 2, ..., m$.
4. Map the high-dimensional data to the low-dimensional embedding space.

Computing the Linear Weights

From Eq. (i), the linear weight matrix $R = [r_1 \ \ r_2 \ \cdots \ r_N]$ in $\mathbf{R}^{N \times N}$, is given by computing the following closed formed solution for r_i, where r_i is a column vector in the i-th row of R, e is a column vector of all ones, and $G_{jl} = (x_j - x_i)^T (x_l - x_i)$ where x_j, x_l are in the k-nearest neighbors of x_i.

$$r_i = \frac{G^{-1} e}{e^T G^{-1} e} \tag{1}$$

Computing the Non-linear Weights

The non-linear weight matrix $W \in \mathbf{R}^{N \times N}$, is computed by the following equation.

$$W_{ij} = R_{ij} + R_{ji} - \sum_{k=1}^N R_{ik} R_{kj}, \text{ and } D_i = \sum_{j=1}^N W_{ij} = 1 \tag{2}$$

Solving the Generalized Eigenvalue Problem

Define $\phi = [\phi(x_1) \ \ \phi(x_2) \ \cdots \ \phi(x_N)]$ to be a $(pn) \times N$ matrix and W as the non-linear reconstruction weight matrix of size $N \times N$. Casting LLE into the framework of spectral embedding and applying the polynomial assumption turns Eq. (ii) into the following, where D is a diagonal matrix with i-th entry D_i.

$$\min_{v_k} \sum_k v_k^T \phi (D - W) \phi v_k$$

$$\text{s.t.} \quad v_j^T \phi D \phi v_k = \delta_{jk}$$

By the Rayleigh-Ritz Theorem, the optimal solutions can be found by solving the following generalized eigenvalue problem to obtain the eigenvectors $v_i, i = 1, 2, ..., m$ corresponding to the m smallest eigenvalues.

$$\phi(D - W)\phi^T v_i = \lambda \phi D \phi^T v_i, \quad v_i^T \phi D \phi^T v_j = \delta_{ij} \tag{3}$$

Mapping to the Low Dimensional Space

For a data sample $x_i \in \mathcal{X}$, its low dimensional representation $y_i \in \mathcal{Y}$ is given by

$$y_i = \left[v_1^T \phi(x_i) \ \ v_2^T \phi(x_i) \ \cdots \ v_m^T \phi(x_i) \right]^T \tag{4}$$

Importantly, the mapping function above holds true for a new data sample $x_t \notin \mathcal{X}$, allowing for its low dimensional representation y_t to be computed efficiently.

Results and Conclusions

The NPPE algorithm is applied to the SwissRoll dataset comprised of points in $\mathbf{R}^3$, shown in **Figure 1**, to find an explicit polynomial mapping to $\mathbf{R}^2$. The data is split into a training set, the lower part, and a testing set, the upper part. The parameter of k-nearest neighbors is set to be 1% of the training samples N and the polynomial degree $p = 2$.

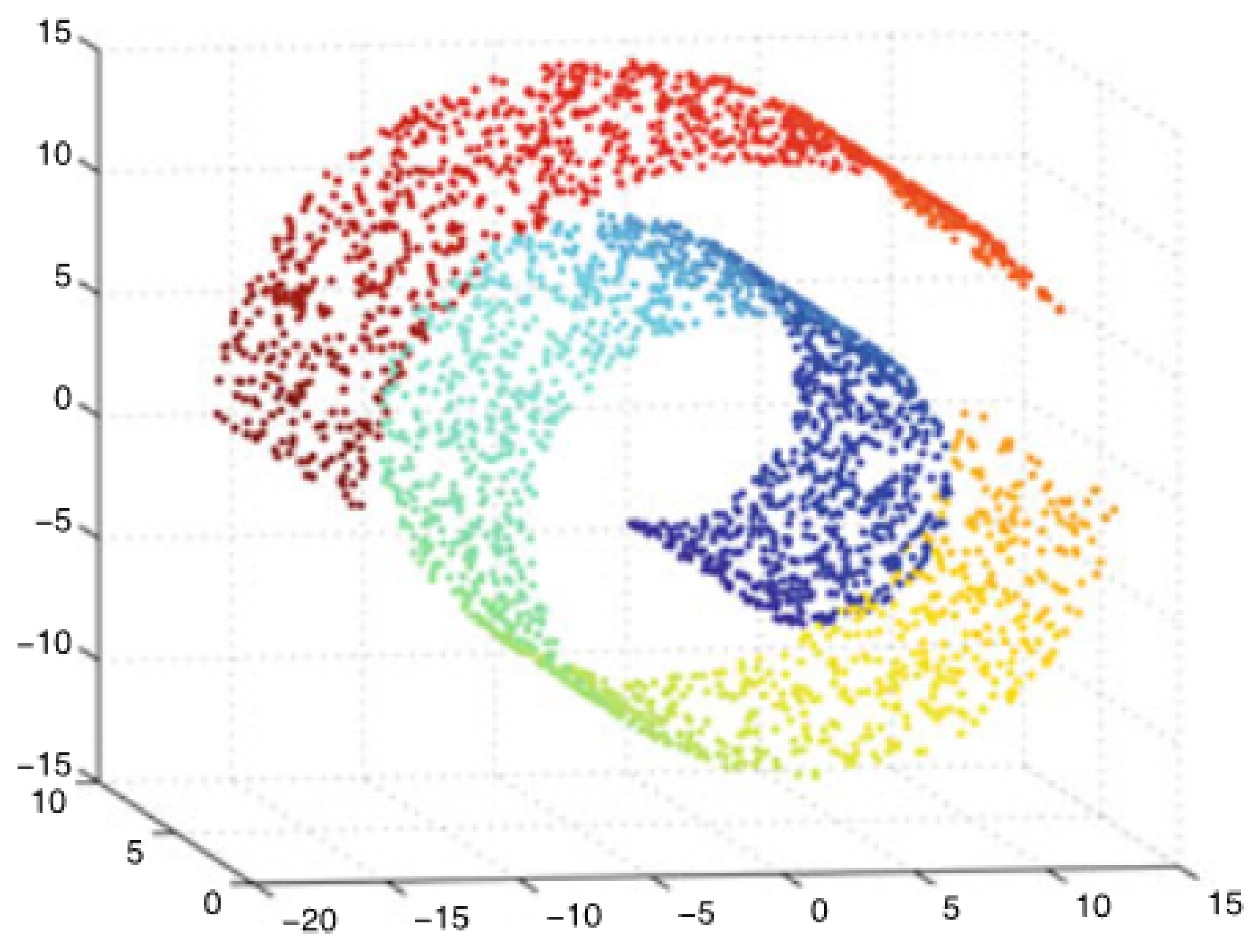

Figure 1: SwissRoll dataset in the high dimensional space $\mathbf{R}^3$

The mapping relationship from $\mathbf{R}^3 \to \mathbf{R}^2$, computed by the NPPE algorithm, is performed only on the training set to test the performance on out of sample data. The learned mapping from the training set is used on both the training and testing sets to find their low dimensional representations, shown in **Figure 2**.

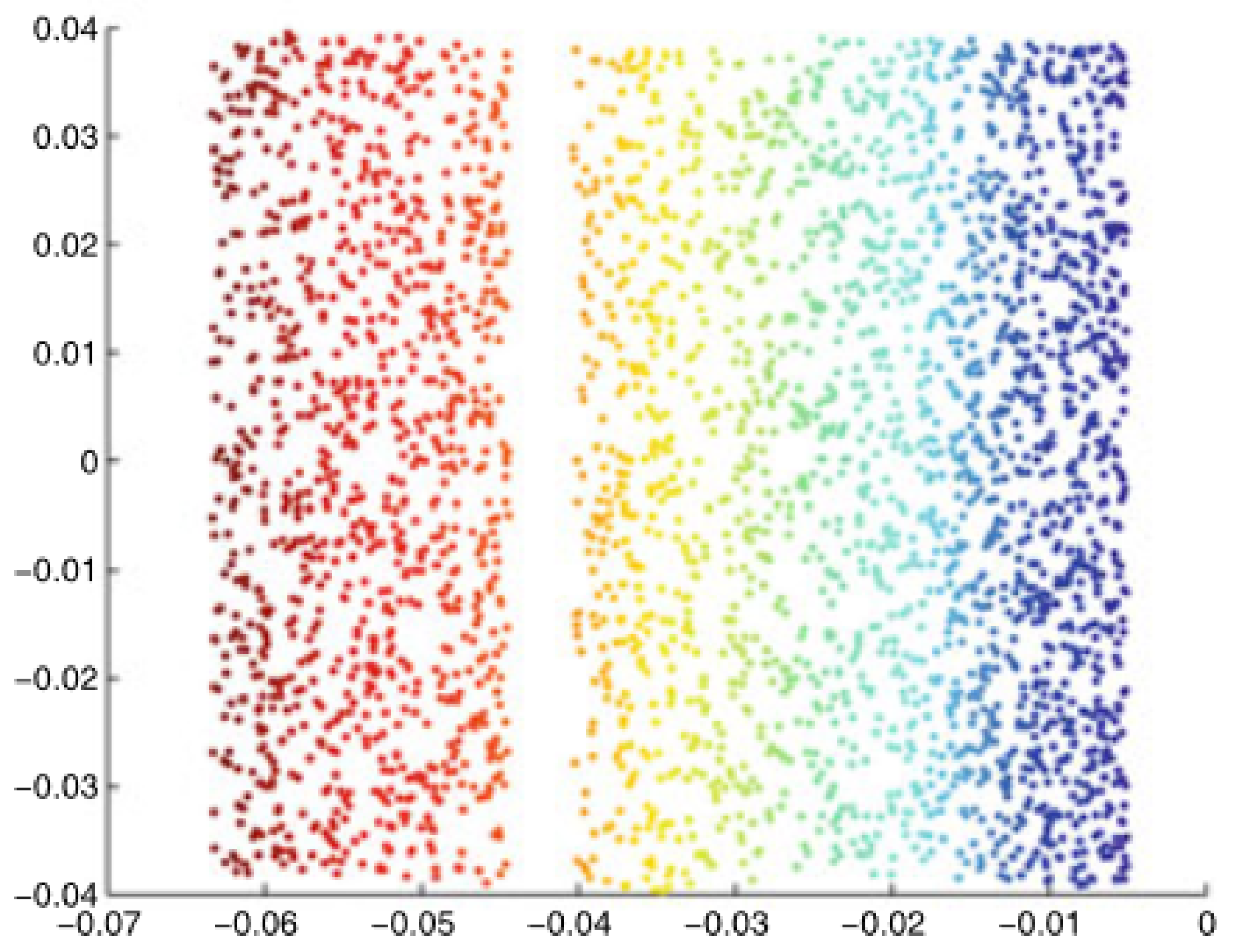

Figure 2: Low dimensional representation of the SwissRoll dataset in $\mathbf{R}^2$

Results show that the NPPE algorithm is effective at representing higher dimensional data in a low dimensional space. Importantly, the learned mapping function remains effective for samples not included in the original training set.

Forthcoming Research

The implementation of the outlined NPPE algorithm, in C++ using Eigen3 for matrix operations and OpenMP for parallel computing, will be finalized. From there, the algorithm will be used as a pre-processing step to reduce the complexity of datasets for supervised machine learning algorithms and the lower overall computation time and hardware storage requirements will be measured.

References

[1] Hong Qiao, Chao Ma, and Rui Li. *The "Hand-eye-brain" System of Intelligent Robot: From Interdisciplinary Perspective of Information Science and Neuroscience*, chapter Explicit Nonlinear Mapping for Manifold Learning with Neighborhood Preserving Polynomial Embedding, pages 81–106. Springer Singapore, 2022.

[2] Hong Qiao, Peng Zhang, Di Wang, and Bo Zhang. An Explicit Nonlinear Mapping for Manifold Learning. *IEEE Transactions on Cybernetics*, 43(1):51–63, 2013.

[3] Lawrence Saul and Sam Roweis. An Introduction to Locally Linear Embedding. *Journal of Machine Learning Research*, 7, 2001.

[4] Laurens van der Maaten, Eric Postma, and H. Herik. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research - JMLR*, 10, 01 2007.

[5] Jianzhong Wang. *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*, chapter Locally Linear Embedding, pages 203–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

Acknowledgements:
Research Advisor: Jonathan Duncan, PhD