SLT coding exercise #1

# Locally Linear Embedding

https://gitlab.vis.ethz.ch/vwegmayr/slt-coding-exercises

Due on Monday, March 6th, 2017

IDIL KANPOLAT
16-931-107

# Contents

# The Model

The model section is intended to allow you to recapitulate the essential ingredients used in Locally Linear Embedding. Write down the *necessary* equations to specify Locally Linear Embedding and and shortly explain the variables that are involved. This section should only introduce the equations, their solution should be outlined in the implementation section.

With LLE we are calculating the low dimensional embedding of our dataset considering the neighbours of the data points. The aim is to find the weights of the mapping by minimizing the error function defined and after fixing the weights finding the low dimensional vectors by minimizing the related error function.

$V := \{X_1, X_2, ...X_N\}$ is a finite subset of $R^D$ and $N(X_i) := \{X \in V | X \text{ is a neighbour of } X_i\}$.

Find $W_{ij}$ that minimizes
$E(W) = \sum_i (X_i - \sum_j W_{ij} X_j)^2$ subject to $\sum_j W_{ij} = 1$ and $for X_j \notin N(X_i), W_{ij} = 0$

For fixed $W$ find $U := \{Y_1, Y_2, ...Y_N\} \in R^d, d \ll D$ that minimizes
$E(Y) = \sum_i (Y_i - \sum_j W_{ij} Y_j)^2$ subject to $\sum_i Y_i = 0$ and $\sum_i Y_i^T Y_i = \mathbb{I}$

# The Questions

This is the core section of your report, which contains the tasks for this exercise and your respective solutions. Make sure you present your results in an illustrative way by making use of graphics, plots, tables, etc. so that a reader can understand the results with a single glance. Check that your graphics have enough resolution or are vector graphics. Consider the use of GIFs when appropriate.
Hard limit: Two pages

## (a) Get the data

For this exercise we will work with the MNIST data set. In order to learn more about it and download it, go to http://yann.lecun.com/exdb/mnist/.

## (b) Locally linear embedding

Implement the LLE algorithm and apply it to the MNIST data set. Provide descriptive visualizations for 2D & 3D embedding spaces. Is it possible to see clusters?

## (c) Cluster structure

Investigate the cluster structure of the data. Can you observe block structures in the $M$ matrix (use matrix plots)? Also plot the singular values of $M$. Do you notice something? Can you think of ways to determine the optimal embedding dimension?

## (d) Nearest Neighbors

Investigate the influence of the choice of how many nearest neighbors you take into account. Additionally, try different metrics to find the nearest neighbors (we are dealing with images!).

## (e) Linear manifold interpolation

Assume you pick some point in the embedding space. How can you map it back to the original (high dimensional) space? Investigate how well this works for points within and outside the manifold (does it depend on the dimensionality of the embedding space?) Try things like linearly interpolating between two embedding vectors and plot the sequence of images along that line. What happens if you do that in the original space?

### (b) Locally linear embedding

I have applied LLE to 3000 points of MNIST test set due to time constraint. The results for k=5 and k=30 nearest neighbours and for 2 dim and 3 dim cases can be seen below. In the graphs bellow the numbers represent the class labels. In both cases it is not really possible to observe a distinguishable cluster structure but overlapping clusters. There are labels with distinguishable behaviour such as class 0 for 'k=5, 2D, 3000 points' case.

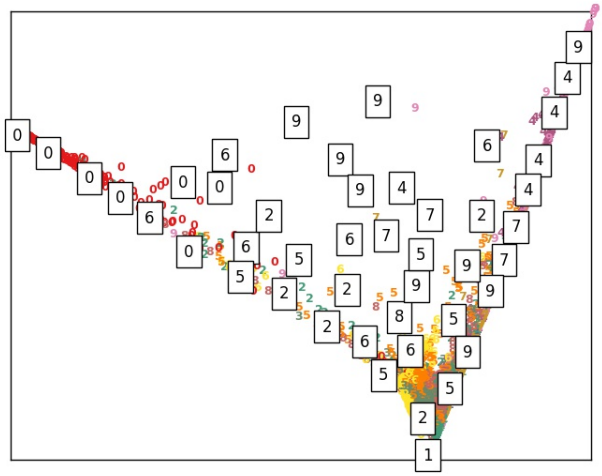

Figure 1: Embedded vectors for k=5, 2D, 3000 points

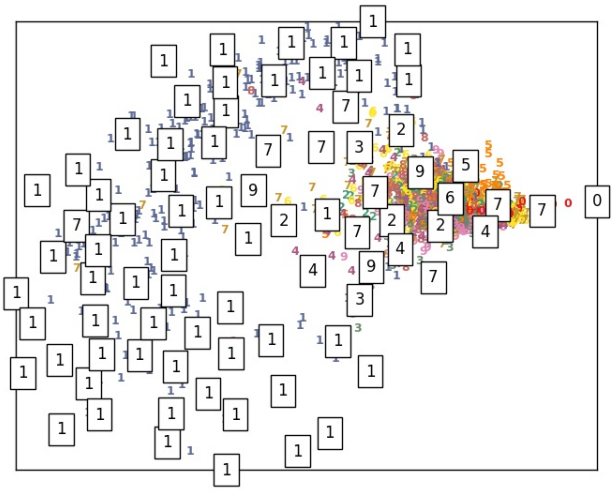

Figure 2: Embedded vectors for k=30, 2D, 3000 points

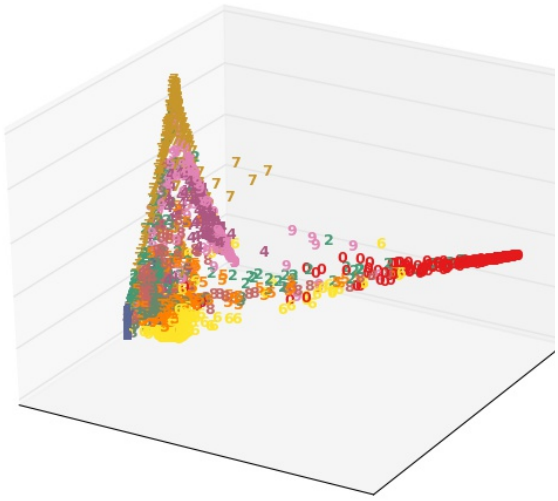

Figure 3: Embedded vectors for k=5, 3D, 3000 points

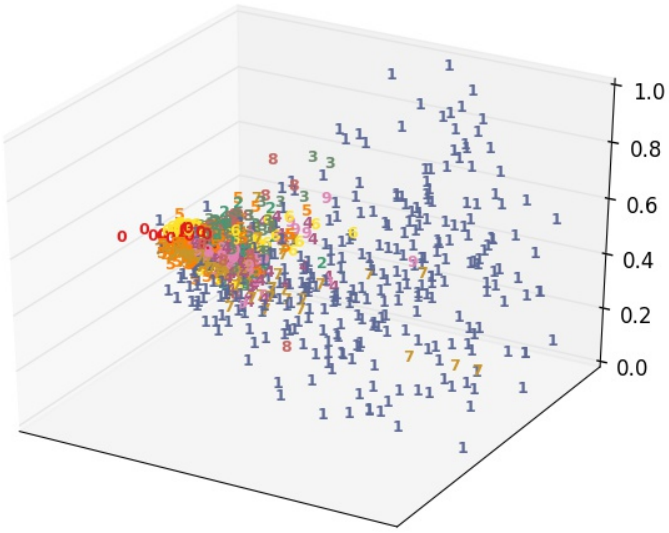


Figure 4: Embedded vectors for k=30, 3D, 3000 points

The Questions continued on next page...

## (c) Cluster structure

In the exercise we have seen that minimizing the error corresponds to finding the d eigenvectors of $M = (\mathbb{I} - W)^T(\mathbb{I} - W)$ with the smallest eigenvalues that are not equal to zero. We can decide on the number d with respect to the eigenvalues that are small enough. The figure below represents the $M$ matrix, where the diagonal elements contain the information and the rest of the elements are mostly sparse. The second figure is a plot of the eigenvalues which might help deciding on the dimension of the mapped space.

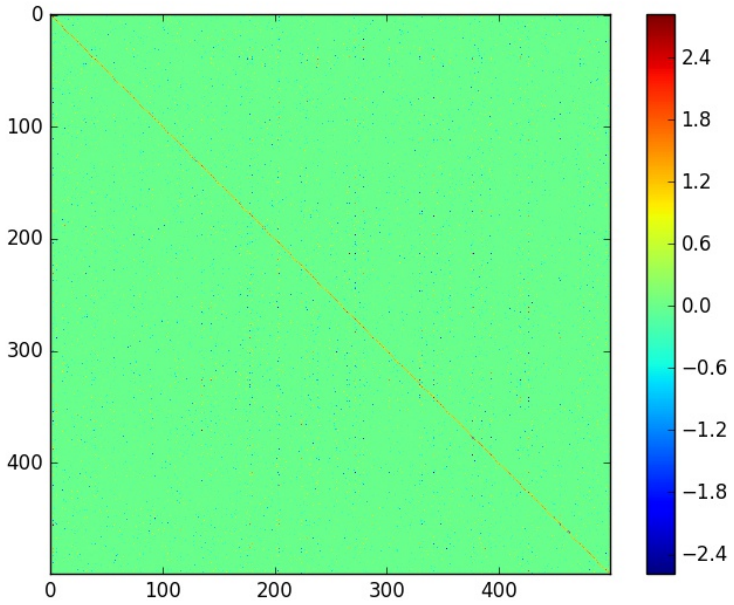


Figure 5: M matrix for k=30, 800 points

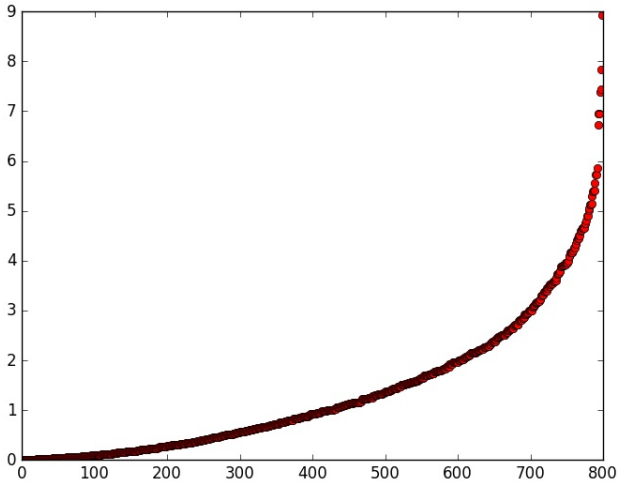


Figure 6: Eigenvalues of M matrix for k=30, 800 points

## (d) Nearest Neighbors

The mapping is repeated for different number of nearest neighbours. From the figures we can see that a small number such as k=2 will not be efficient to reflect the global structure of the dataset, whereas increasing k can cause us to lose its nonlinear character[1].

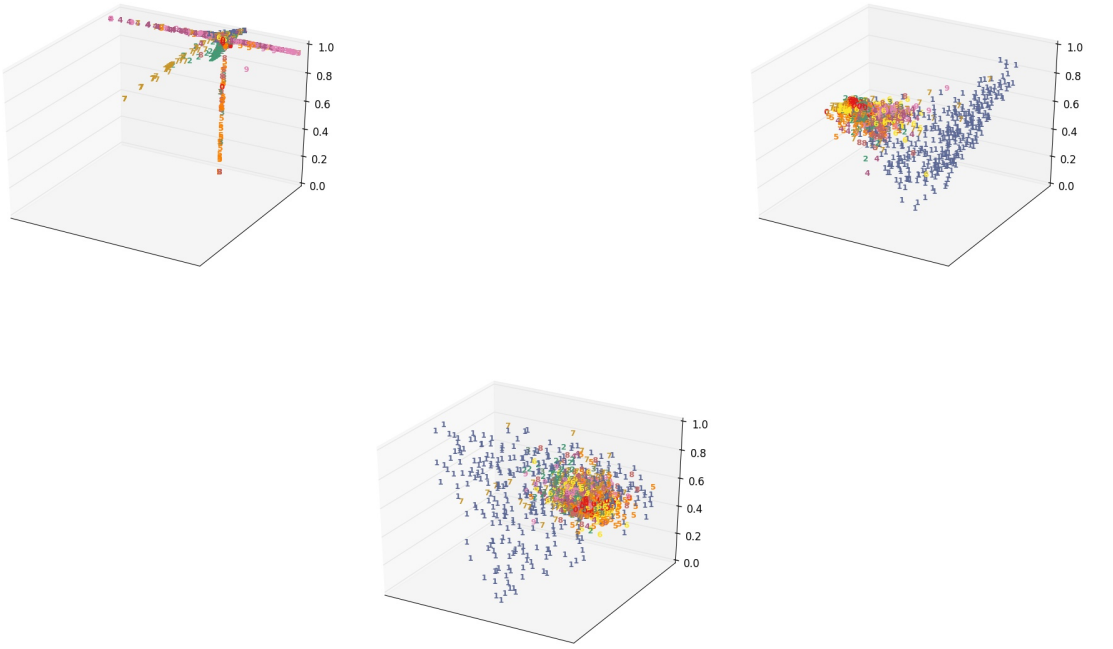

Figure 7: Embedded vectors for k= 2, 50, 100, 3D, 3000 points

## (e) Linear manifold interpolation

## The Implementation

In the implementation section you give a concise insight to the practical aspects of this coding exercise. It mainly mentions the optimization methods used to solve the model equations. Did you encounter numerical or efficiency problems? If yes, how did you solve them? Provide the link to your git branch of this coding exercise.
Hard limit: One page

I have used scikit-learn for implementation mostly. The code is slow for cases with high number of data points or high number of nearest neighbours. To eliminate these problems I have worked with 3000 points in most of the cases. I believe the main bottleneck is the nearest neighbour calculation.
16-931-107/1_locally_linear_embedding

## Your Page

Your page gives you space to include ideas, observations and results which do not fall into the categories provided by us. You can also use it as an appendix to include things which did not have space in the other sections.
No page limit.

16-931-107/1_locally_linear_embedding

## References

[1] Juliana Valencia-Aguirre, Andrs lvarez-Mesa *Automatic Choice of the Number of Nearest Neighbors in Locally Linear Embedding* CIARP, pp 77-84, 2009.