SLT coding exercise #1

Locally Linear Embedding

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

Due on Monday, March 6th, 2017

Michael Giegrich
16-715-187

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.
Hard limit: One page

The data set contains N vectors X_i with dimensionality D.First, we need to identify the K nearest neighbors per data point using the appropriate metric. The reconstruction error which is minimized over the optimal weights W is given by

$$E(W) = \sum_i \mid X_i - \sum_j W_{ij} X_j \mid^2 \tag{1}$$

where $W_{ij} = 0$ if X_j is not a nearest neighbor and $\sum_j W_{ij} = 1$. Further, we choose the lower dimensional representations Y_i by minimizing the embedding cost function:

$$\Phi(Y) = \sum_i \mid Y_i - \sum_j W_{ij} Yj \mid^2 \tag{2}$$

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

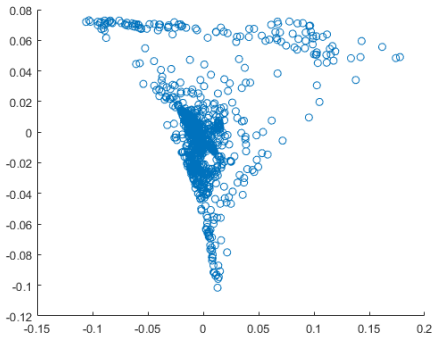
Investigate the influence of the choice of how many nearest neighbors you take into account. Additionally, try different metrics to find the nearest neighbors (we are dealing with images!).

(e) Linear manifold interpolation

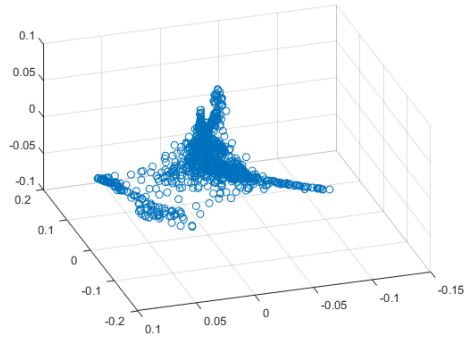
Assume you pick some point in the embedding space. How can you map it back to the original (high dimensional) space? Investigate how well this works for points within and outside the manifold (does it depend on the dimensionality of the embedding space?) Try things like linearly interpolating between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

The Questions continued on next page...

(b) In the visualization of the Local Linear Embedding, we can distinguish several clusters in both the 2D case and the 3D case. The 3D embedding, however, allows us to identify more clusters than the 2D embedding.

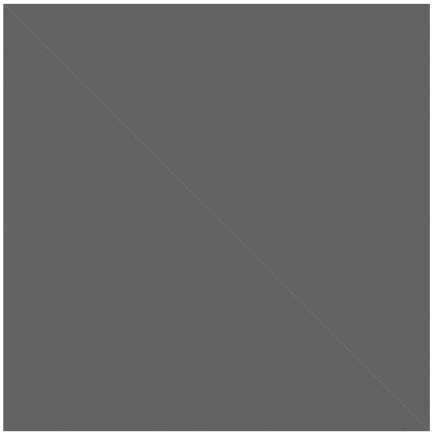


(a) 2D Embedding with 1000 digits and 15 neighbors

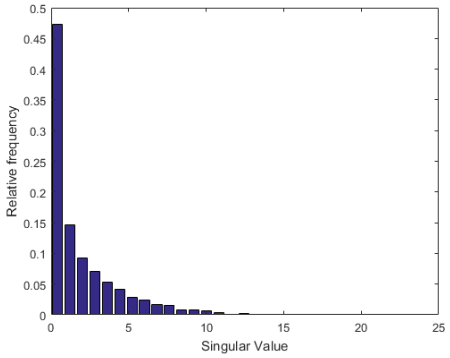


(b) 3D Embedding with 1000 digits and 15 neighbors

(c) The matrix is relatively sparse and close to being a diagonal matrix. I cannot observe any block structures. Most singular values are close to 0 and the relative frequency of singular values is sharply decreasing in the values. The optimal embedding dimension can be found by examining the d+1 smallest eigenvalues not equal to zero. Using them, we can introduce a threshold on the embedding error.



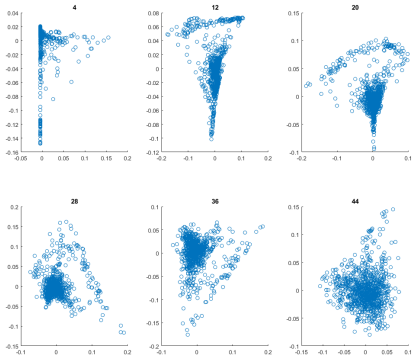
(a) Matrix Plot



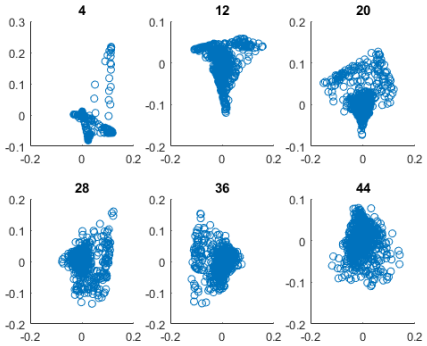
(b) Singular Values

(d) The choice of the number of neighbors has a big effect on embedding. It seems that the visibility of clusters first improves with the number of neighbors but then diminishes for a to large number of neighbors (as can be seen in the Euclidean norm graph). Also the choice of the norm has an impact on the quality of the dimension reduction. The Euclidean Norm seems to generate the best clusters when comparing the norms I used.The Minkowski norm seems to perform also quite well while the City Block and the Chebychev norm perform the worst.

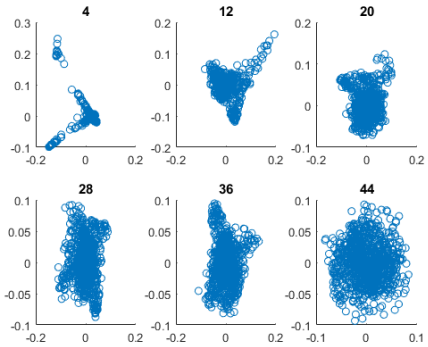
(e) I can map a point back into the high-dimensional space by considering the nearest neighbors in the low dimensional space. By taking into account the distances to the nearest neighbor in the low dimensional space and weighting them accordingly,I am able to interpolate a representation of the high dimensional point. This method seems to work quit well for points on the boundary of the manifold and inside the manifold performance varies. Further, the inversion of the 3D space seems to perform better than the 2D case.



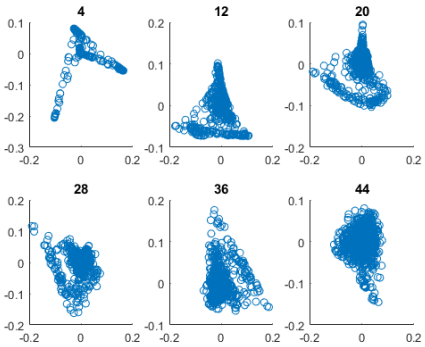
(a) Euclidean Norm



(b) City Block Norm

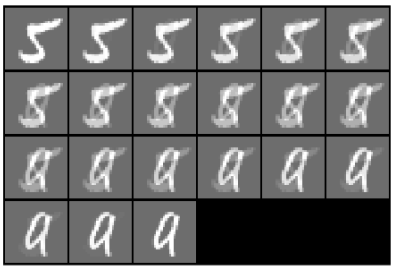


(c) Chebychev Norm



(d) Minkowski Norm

(a) Inversion of a 3D point. The inverted point is on the top left.



(b) Inversion of an Interpolation between two points in the low dimensional

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

For my implementation, I mainly relied on the dimension reduction toolbox for matlab. More specifically the lle function. To employ different norms in part (d), I had to adapt the code accordingly. I did not encounter other problems.

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.

16-715-187/1_locally_linear_embedding