SLT coding exercise #1

Locally Linear Embedding

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

Due on Monday, March 6th, 2017

Linus Handschin
12-922-621

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

Given sample $\mathbf{X}_i$ and its neighbors $\mathbf{X}_j$ in a high dimensional space LLE computes the barycentric coordinates. The sample is reconstructed by a linear combination, given by $\mathbf{W}_ij$ (the same weight matrix is used to reconstruct the point in the low dimensional space). The reconstruction error is then given by:
$E(W) = \sum_i |\mathbf{X}_i - \sum_j \mathbf{W}_{ij}\mathbf{X}_j|^2$
Each data point is reconstructed only is reconstructed from its neighbor and the contribtuion must be normalized, e.g. $\sum_j \mathbf{W}_{ij} = 1$.

Each sample $\mathbf{X}_i$ is mapped to a low dimensional representation $\mathbf{Y}_i$ by minimizing the the following cost function for a fixed $\mathbf{W}_{ij}$:
$C(Y) = \sum_i |\mathbf{Y}_i - \sum_j \mathbf{W}_{ij}\mathbf{Y}_j|^2$

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?

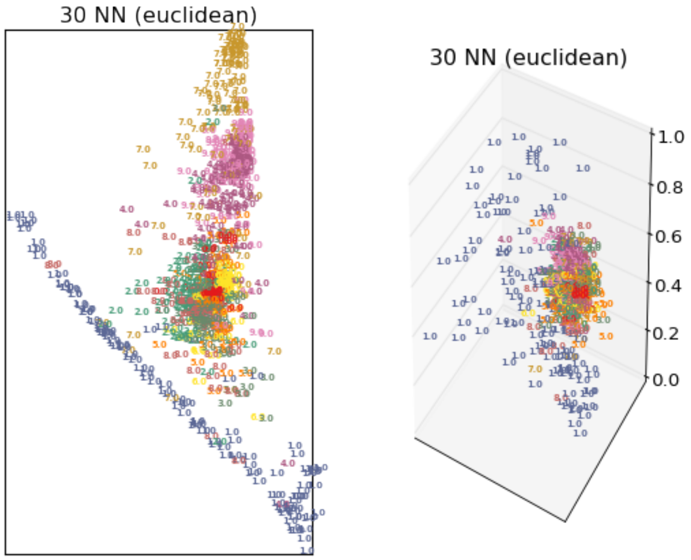

Figure 1: 2D and 3D LLE of 1000 MNIST samples using the 30 nearest neighbor (according their eucledian distance) of each sample

The above plots show distinct clusters only using the 30 nearest neighbors in a 2dimensional and 3 dimensional embedding space.
(All the following expermements are performed using 1000 random selected MNIST samples and eucledian distance metric for the 30 nearest neighbors if not stated otherwise.)

(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?

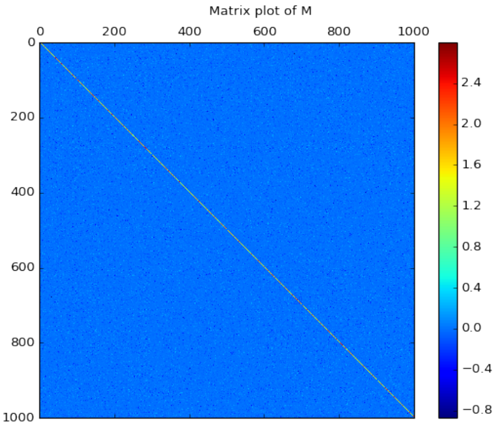

Figure 2: Matrix plot of M with large values on the diagonal

The matrix plot shows large values on the diagonal and all other values are all close to zero.

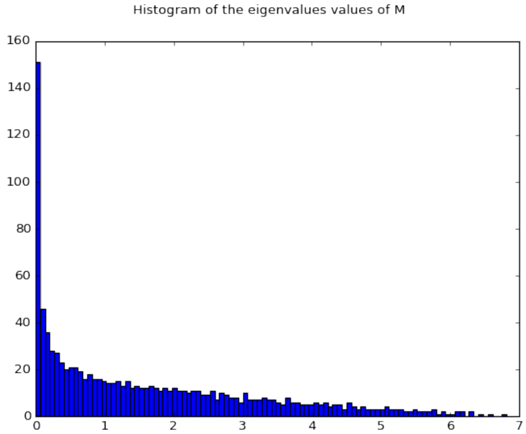

Figure 3: The histogram shows that most eigenvalues are close to zero.

The distribution of the eigenvalues shows that most of the eigenvalues are close to zero. To minimize the reconstruction error we are looking for the the eigenvectors with low corresponding eigenvalues. For this model a knee can seen around the first 200 to 250 dimensions.

(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!).

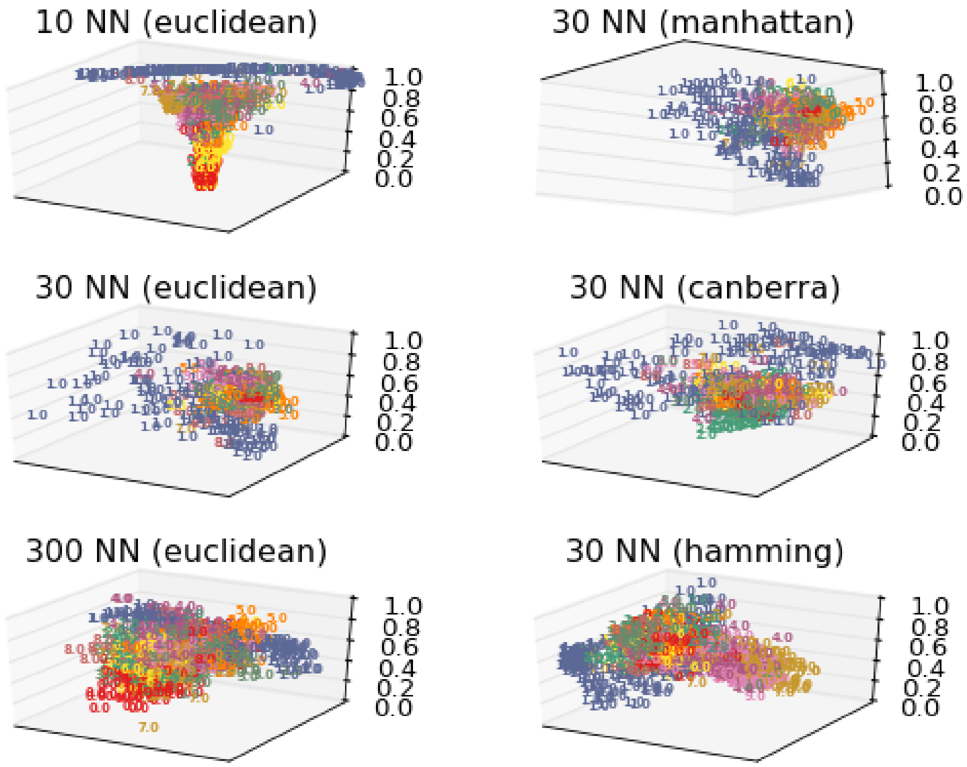

Figure 4: 3D embeddings for different distance metrics and nearest neighbor configurations.

Experiments with different distance metrics (manhattan, euclidiean, canberra and hamming distance) and different number of neighbors (10, 30, 300 nearest neighbors) showed relative low influence on the clusters structure. For all tested metrics and number of nearest neighbors the samples of the digit one can easily be separated. This makes sense since ones have the least numbers of dark pixels. For all tested configurations the cluster were very stable (4). Since the digits are black white with only few gray scale values and centered in the image the distance metric has low influence and good results can be achived using the euclidean distance. Increasing the number of neighbors results in dense (and overlapping) clusters with only a few outliers.

(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?

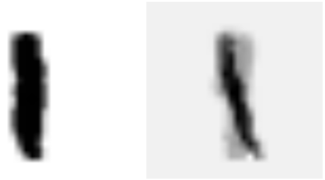

Figure 5: The reconstruction from 2d space back to the 784 dimensional space (28x28 images). Left: original, right: reconstruction

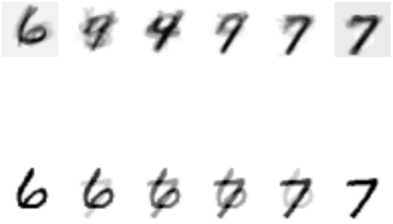

Figure 6: Top: interpolation between embedded digit 6 and 7. Bottom: Same interpolation in the original high dimensional space.

The reconstruction of a one digit from a two dimensional embedding space is shown in figure 5. The reconstruction requires the following three steps:

1. Find the set of the k-nearest neighbors of $\mathbf{y}$ in the low dimensional space ($\mathcal{N}(\mathbf{y})$)
2. Compute the weight matrix $\mathbf{w}$ as in the original LLE algorithm
3. Approximate the original point as the linear combination found in step 2 by neighbors found in step 1 in the original space e.g.: $\hat{\mathbf{x}} = \sum_{x_j \in \mathcal{N}(x) = \mathcal{N}(y)} \mathbf{w}_j \mathbf{x}_j$

The linear interpolation between embedding vectors and points in the original space can be seeen in figure 6. The interpolation in the original space is just a linear transition between the interpolating digits. Much more interesting is the interpolation of embedding vectors. This shows corresponding digit of the cluster which the line traverses.

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 encountered numerical instabilities when inverting the matrix $\mathbf{C}$ for reconstructing a point. To solve this I followed the adivice of the original LLE paper and found the weights by solving the linear system of equations $\sum_j \mathbf{C}_{jk}\mathbf{w}_k = 1$.

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.

Your Answer
12-922-621/1_locally_linear_embedding