SLT coding exercise #1

Locally Linear Embedding

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

Due on Monday, March 6th, 2017

Alberto Montes
16-932-451

March 1, 2017

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

Considering a data set of N points D-dimensional $\mathbf{x}_i \in \mathbb{R}^D$. The goal of Locally Linear Embedding is to embed this points into a lower dimensional space $d \ll D$, so to represent each point $\mathbf{x}_i$ with a point $\mathbf{y}_i \in \mathbb{R}^d$. Define a reconstruction error:

$$E(\mathbf{W}) = \sum_i \|\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_j\|^2 \tag{1}$$

Being j the index of the K nearest neighbors of $\mathbf{x}_i$ being K a parameter to tune. As the error want to be invariant to uniform translations, $\mathbf{W}$ has the following restriction: $\sum_j w_{ij} = 1$. $\mathbf{W}$ is computed with the following equation:

$$w_{ij} = \frac{\sum_k C_{jk}^{(i)-1}}{\sum_{jk} C_{lk}^{(i)-1}} \text{ and } C_{lk}^{(i)-1} = (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_k) \tag{2}$$

Then fixing $\mathbf{W}$ and minimizing the embedding error:

$$E(\mathbf{y}_1 \dots \mathbf{y}_N) = \sum_i \|\mathbf{y}_i - \sum_j w_{ij}\mathbf{y}_j\|^2 = \sum_k \mathbf{u}_k^T \mathbf{M} \mathbf{u}_k \tag{3}$$

with $\mathbf{u}_k = (y_{ik}, \dots, y_{Nk})$ for $k = 1, \dots, d$ and $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T(\mathbf{I} - \mathbf{W})$.
The solution of this system correspond to the eigenvectors corresponding to the lowest eigenvalues discarding the first one with eigenvalue 0.

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. . .

The implementation of the algorithm is at `locally_linear_embedding.py` file which computes the embedding for the given data. Lets compute the 2D and 3D embeddings for 1000 samples of the MNIST test dataset. The algorithm computes with 8 nearest neighbor and a euclidean distance

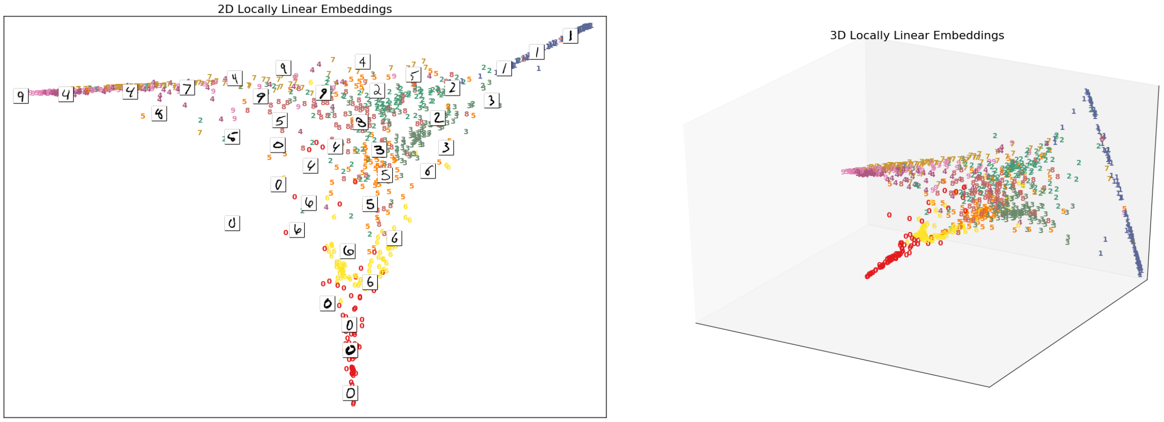
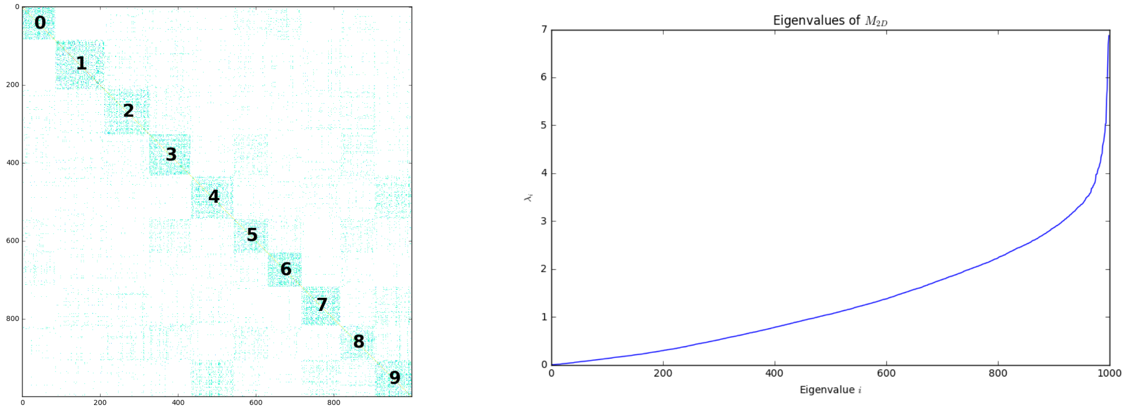


Figure 1: 2D and 3D Locally Linear Embedding with 8 nearest neighbors

On Figure 1 where the resulting embeddings are shown, it can be easily seen some clusters corresponding to digits, and also how samples with the same digit tent to be close to each other. Also is observed how the digits `9`, `0` and `1` are at the extremes of the manifold as are the most different from the rest. Also can be observed a bunch of other digits close to each other as their shape are very similar.

To see more in deep perspective how clusters are made, in Figure 2a there is represented the values of the matrix with the columns and rows ordered by the digit of the its correspondent sample. It can be observed how a lot of correlation between samples of the same digit, and also some cross correlations between digits such as between 4 and 9 or 3 and 5 for example. This results are consistent with the clusters observed at the embeddings.



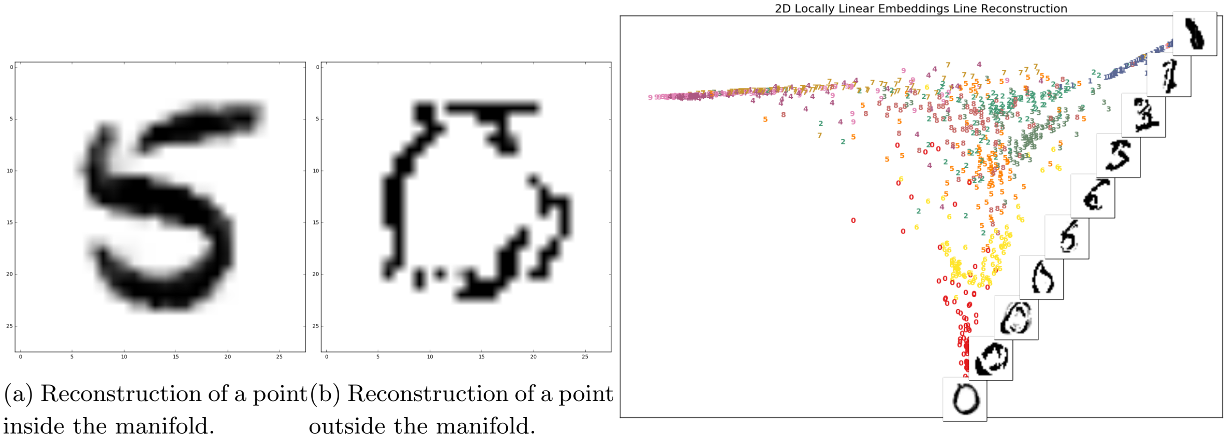
(a) Matrix plot of M.

(b) M eigenvalues.

Figure 2: Matrix M representations obtained for a 2D Locally Linear Embedding.

In Figure 2b there is plot the eigenvalues of the matrix M which constantly increases until a point after the 900th eigenvalue that the values start increasing abruptly. The best option to choose the optimal embedding dimension will be to choose the ith position where the knee is observed at the eigenvalues plot. Other possible study to be made about the Locally Linear Embedding is see the effect of the number nearest neighbors used at the algorithm where the Nearest Neighbors are applied.

Regarding the embedding reconstruction, running the LLE algorithm with 8 nearest neighbors and embedding in 2D, I have tried the reconstruction of a sample inside the manifold, an another sample outside the manifold. On Figure 3a can be observed how a digit number 5 has been reconstructed by the 8 nearest neighbors of the manifold. On the other hand, on Figure 3b can be observed how a point close the 0 but placed outside the manifold is reconstructed but showing some malformation in the shape of the digit and also some discontinuities on the trace.



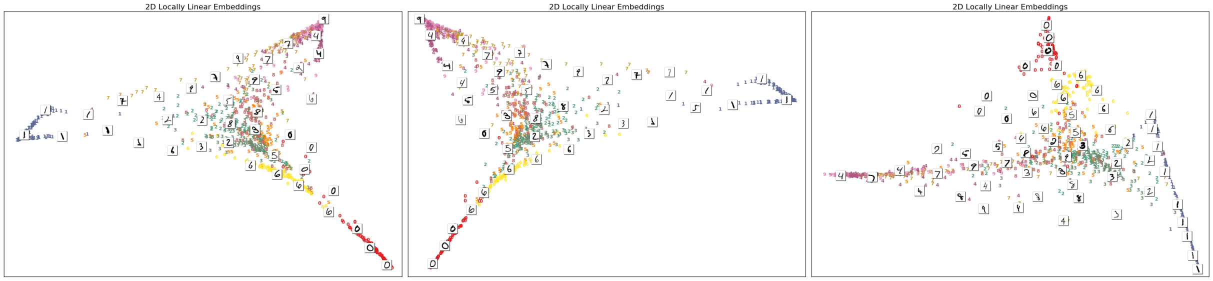
(a) Reconstruction of a point inside the manifold.

(b) Reconstruction of a point outside the manifold.

(c) Reconstruction of a line of points outside of the manifold.

Figure 3: Reconstruction of points in the 2D embedding to the original space.

In addition, it has been drawn a line from two points crossing outside the manifold and reconstructing the middle points. In Figure 3c can be observed how the points reconstructed outside the manifold show the same characteristics as the one in Figure 3b but in addition it can be seen the evolution of the digit from 0 to 1 passing through 6, 5 and 3.
Finally, different distance measures have been tried to run the LLE algorithm. In Figure 4 can be found the 2D embedding for different metrics used: correlation between images, cosine distance and Manhattan or L1 distance respectively.



(a) Correlation metric.

(b) Cosine distance.

(c) L1 distance.

Figure 4: 2D embedding for different metric/distance functions.

For this three metrics, it can be observed how the clusters keep forming for each of the digits of the dataset, but change the shape of the embedded manifold.

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

My current implementation is found on the git branch `16-932-451/1_locally_linear_embedding`.
The implementation has not required further optimizations of the code. Working with 1000 samples it has been possible to work with dense arrays in a very precise and fast way even when W matrix is very sparse. For larger datasets, it would be necessary to work with sparse representation of this matrix in addition to use approximate methods to compute the eigenvalues and eigenvectors. In addition to this, I have not encounter this problem but in case when the weights are computed, we found a C singular matrix, it would be necessary to add a little regularization much lower than the trace as the original paper specify in the annex.

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.

I would like to add some results I get but I didn't found any reasonable explanation. Computing the reconstruction error agains the number of nearest neighbors used in the algorithm, I found that increasing the number of nearest neighbors, the reconstruction error tent to continuously decreasing such as shown in Figure 5. This does not make any sense because I found some equivalent computations in the literature[a] where they find a minimum corresponding an optimal K for the nearest neighbor where the reconstruction error of the embedding was minimum. Also I haven't found the sense to thus results because if I execute the code with 50 nearest neighbors for examples, the resulting 2D embedding consist in a uniform cloud where no clusters can be found such as in Figure 6.

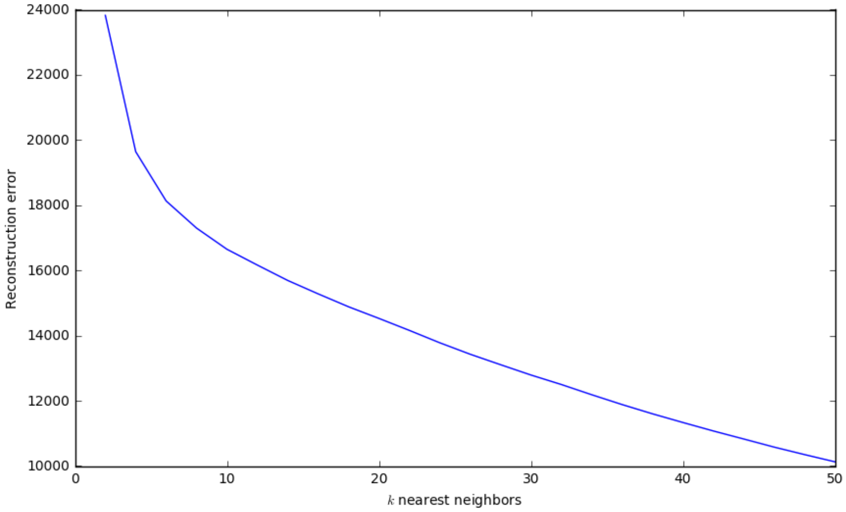


Figure 5: Reconstruction error against the number of Nearest Neighbors.

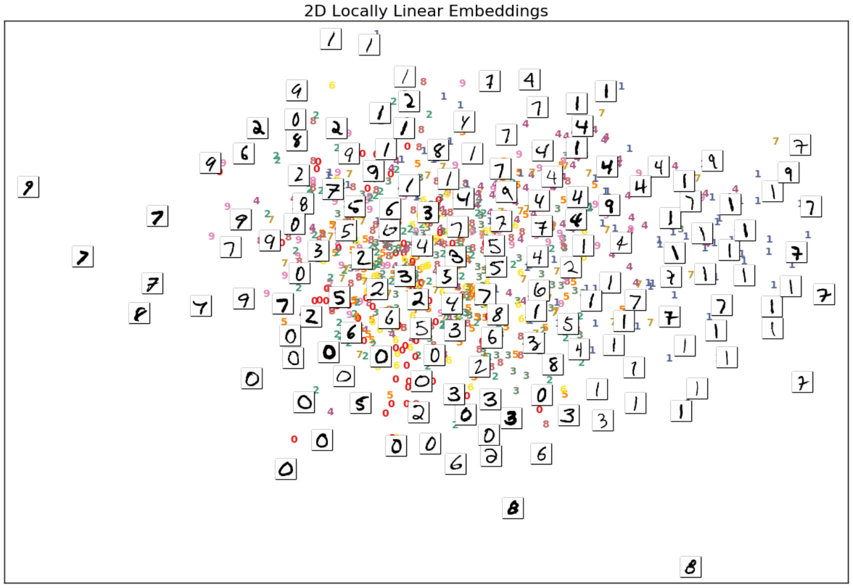


Figure 6: 2D Embedding space using 50 nearest neighbors.

[a] http://jultika.oulu.fi/files/isbn9514280415.pdf