SLT coding exercise #1

Locally Linear Embedding

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

Due on Monday, March 6th, 2017

Robin Vaaler
12-927-463

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

$E(W) = \sum_i |\mathbf{X}_i - \sum_j \mathbf{W}_{ij}\mathbf{X}_j|^2$
The weights W_{ij} refer to the amount of contribution the point X_j has while reconstructing the point X_i. The cost function is minimized under two constraints: (a) Each data point 'X_i is reconstructed only from its neighbors, thus enforcing W_{ij} to be zero for all points i and j that aren't neighbors and (b) The sum of every row of the weight matrix equals 1.
$\sum_j \mathbf{W}_{ij} = 1$
The original data points are collected in a D dimensional space and the goal of the algorithm is to reduce the dimensionality to d such that $D >> d$. The same weights W_{ij} that reconstructs the ith data point in the D dimensional space will be used to reconstruct the same point in the lower d dimensional space. A neighborhood preserving map is created based on this idea. Each point X_i in the D dimensional space is mapped onto a pointY_i in the d dimensional space by minimizing the cost function
$C(Y) = \sum_i |\mathbf{Y}_i - \sum_j \mathbf{W}_{ij}\mathbf{Y}_j|^2$
In this cost function, unlike the previous one, the weights W_{ij} are kept fixed and the minimization is done on the points Y_i to optimize the coordinates.

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

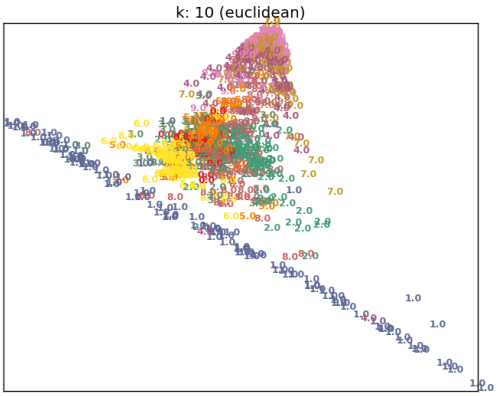
Investigate the influence of the choice of how many nearest neighbors you take into account. Additionally, try different metrics to find the nearest neighbors (we are dealing with images!).

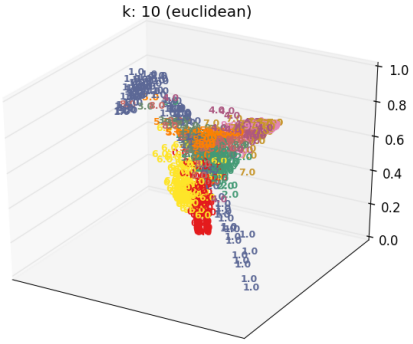
(e) Linear manifold interpolation

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?

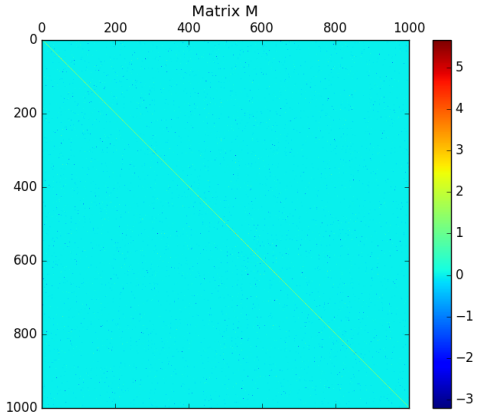
The Questions continued on next page...

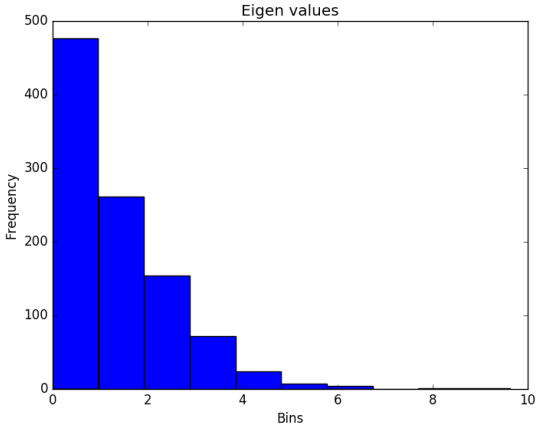
(a) I used the sklearn library to download the dataset. To allow for faster iteration I only used a sample of 1000 images for my analyis.
(b) The figure on the left (right) visualizes the resulting embedding space for two (three) dimensions. Images corresponding to similar digits tend to be close in the resulting embedding space.



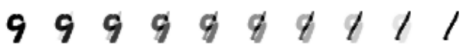


(c) The M matrix is very sparse and is almost diagonal. Most eigen values of M (equal to singular values because M is semi-positive-definite) are small.





(d)
(e) When one just linearly combines two images the result is just a superposition of the two original images (left). By following the same procedure in the embedding space and re-transforming the resulting embedding to the original space one can see a nonlinear transformation from one digit to another one (right).



The Questions continued on next page

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

In a first step I make use of the k-nearest neighbor implementation provided by the sklearn library. Given the k neighrest neighbors of every data point, the C matrices get constructed since they are needed to compute the W matrix. Instead of inverting the C matrices to get W, one can solve a least-squares problem (using e.g. numpy). To finally get the low dimensional embeddings Y one can use the eigendecomposition implementation provieded by the scipy library to decompose the matrix $M = (I - W)^T(I - W)$. To make sure that we don't encounter any numerical instabilities, the paper recomends to add ϵI to all the C matrices, where ϵ should depend on the traces of the C matrices and the number of neighbors k.

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
12-927-463/1_locally_linear_embedding