SLT coding exercise #1

Locally Linear Embedding

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

Due on Monday, March 6th, 2017

Matthew Thornton
16-948-507

Contents

The Model 3

The Questions 4

(b) Locally linear embedding 4

(c) Cluster structure 4

(d) Nearest Neighbors 5

(e) Linear manifold interpolation 5

The Implementation 6

Your Page 7

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

$$\epsilon(W) = \sum_i |X_i - \sum_j W_{ij} X_j|^2$$

Where X_i is the i^{th} training data sample and where ϵ is the error from approximating X_i from X_j with weights W_{ij}.
X_j are the K nearest neighbors to X_i which are calculated by finding the K smallest distances between X_i and X_j for each X_i.

$$dist(x_i, x_j) = (x_j - x_i)^2$$

$$\Phi(Y) = \sum_i |Y_i - \sum_j W_{ij} Y_j|^2$$

Φ is a cost function that is to be minimized by choosing embedding dimensions Y_i.

The Questions

(b) Locally linear embedding

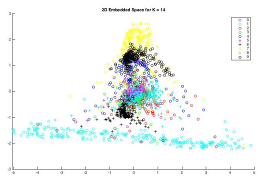
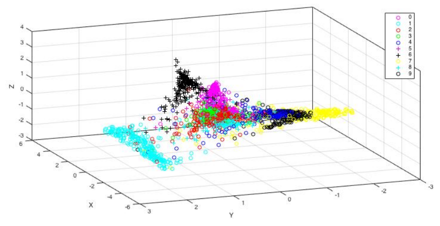
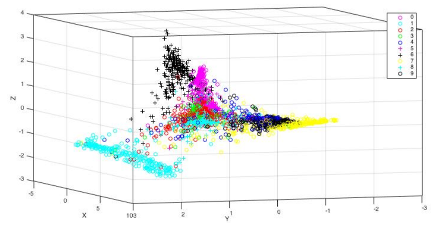


Figure 1: 2D Embedding Space with K = 14



(a) View 1

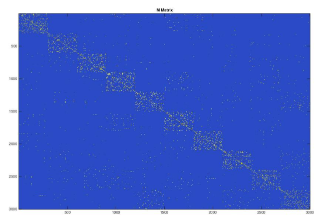


(b) View 2

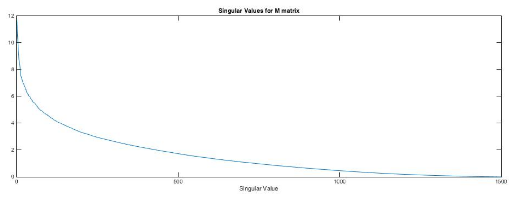
Figure 2: 3D Embedding Space with K = 14

The clusters are evident by labeling the known points with their known training image labels. However, some clusters of numbers are more distinct than others. For example in both the 2D and 3D spaces, number 7 (yellow) and number 0 (cyan circles) are far apart from each other and in mostly unique regions. This would allow for a point near these regions to be mapped successfully back to the higher dimensional space.

(c) Cluster structure



(a) M Matrix



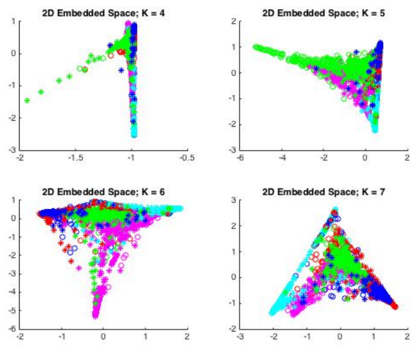
(b) Singular Values of M Matrix

Figure 3

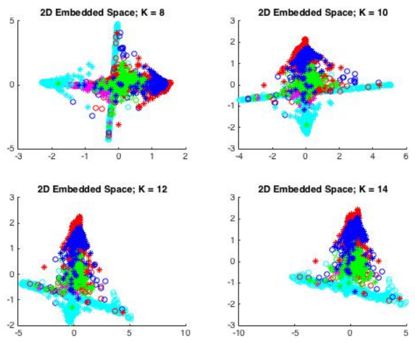
In Figure 3 a) one can see the clusters of data, squares around the diagonal. This occurs because the training data is arranged by known handwritten numbers and the K nearest neighbors are most likely to be other realizations of the same number. From plot b) one can see the singular values; they decrease exponentially. Choosing d embedding dimensions means choosing d singular values. To determine the optimal embedding dimension one could choose a singular value threshold (based on the exponential decay parameter) and then set d to the number of singular values which are under this threshold.

The Questions continued on next page...

(d) Nearest Neighbors



(a) 2D Embedding Space with K = 4, 5, 6, 7 Nearest Neighbors



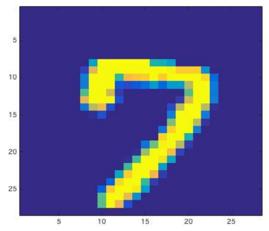
(b) 2D Embedding Space with K = 8, 10, 12, 14 Nearest Neighbors.

Figure 4

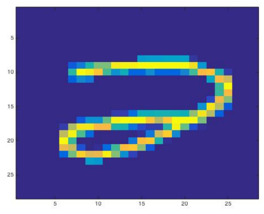
Increasing K for the K nearest neighbors spreads out the clusters in the embedding space. For small K the mapped points fall closer to each other and for larger K the mapped points are more spread out. Somewhere in between, around $K = 14$, the points are mapped in slightly more discretized regions than in the other cases.

(e) Linear manifold interpolation

By picking a point in the embedding space one can again find the K nearest neighbors. By choosing some weighting scheme it is possible to map this point back into the original space with the known higher dimensions (training images). Using this KNN method for a point that is either within or outside of the manifold this "back projection" will always produce an image in the higher dimension. If the clusters are distinct and the random point lies within a cluster then the point unambiguously falls into this cluster and will have a back projection that is a weighted sum of the higher dimensions. A higher dimensional space will allow for more separability of the clusters, which in turn will give a better back projection for some point in the embedded space. Figure 5 shows a) a point that lies near or on the manifold region that corresponds to the handwritten digit 7; whereas, b) shows the mapping of a point within the manifold, but it is not able to be mapped back to a discernible digit in the high dimensional space.



(a) Map From Embedded to Original with Unambiguous Point



(b) Map From Embedded to Original with Ambiguous Point

Figure 5

The Implementation

The algorithm is as follows:

1. K Nearest Neighbors
We are given a matrix where each row is a training image and each column is a data point of the training image. The first goal of the algorithm is, for each row i, to find the K rows that are closest to it. This closeness measure is computed by the Euclidean distance.

This is a computationally complex search function. In order to improve the speed I vectorized the distance equation in order to avoid a large loop.

2. Reconstruction Weights
Then each row of the matrix is modeled by a linear combination of weights from its previously computed nearest neighbors. This is done with a least squares solution to the covariance matrix of the nearest neighbor matrix.

If the number of nearest neighbors, K, is larger than the dimensions of each training image than a regularization constraint is necessary to maintain a unique, well-posed solution.

3. Embedded space
A basis for this weighting matrix is desired. We choose d dimensions in this basis to reduce the space size. The solution is to find the eigendecomposition of this matrix and take the use the first d eigenvectors, corresponding to the smallest, non-zero eigenvalues. These eigenvectors then span the embedded space $\epsilon \mathbb{R}^d$ and show the manifold in this dimension.

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.

The clustering of the labeled images in the embedded space has some dependence on the number of training images that is chosen. It was determined that for this dataset about 200 training images per unique image label was sufficient to provide a defined position in the embedded space for each label.

By choosing fewer training images there is less available information for the algorithm. Therefore there may not be ”sufficient” nearest neighbors for the KNN algorithm to find K ”good” neighbors, which would yield a poor embedded space. However, choosing too many images can lead to an over fitting.

https://gitlab.vis.ethz.ch/vwegmayr/slt-coding-exercises/tree/16-948-507/1_locally_linear_embedding