Artificial Intelligence: Assignment 4

November 14, 2016

Prof. Andrea Torsello

Francesco Pelosin

Dimensionality Reduction or Manifold Learning

One of the problems with high dimensional data is that usually not all the features (dimensions) of a particular phenomenon are really meaningful and descriptive. Sometimes these dimensions are not relevant and they could introduce noise, we should get rid of these exceeding dimensions.

In order to analyze, visualize and represent data in a meaningful way, we need some tools which are able to capture the important facts of a phenomenon. This problem is called dimensionality reduction or manifold learning. It is also important to address this problem to reduce memory complexity and time complexity when we have to perform other operations with data.

Two of the classical tools that have been studied to address such problems are Principal Component Analysis (PCA) and Multidimensional Scaling (MDS). These two algorithms perform linear mappings from the original representation to a new one, in practice they project data into a linear subspace. These algorithms do not have good performances when the manifold which we are studying has a non-linear shape. Two famous examples are the Swiss Roll (Figure 2) and the S-shape manifolds, they are two-dimensional non-linear shapes embedded in a three-dimensional space. Given the fact that PCA and MDS are linear mappers, they usually fail to capture the structure of data, resulting in a poor descriptive embedding.

Several methods have been proposed to address such problems, and some of them are extensions of linear methods like: Kernel PCA and Isomap. Other methods have their own approach like Locally Linear Embedding (LLE) or Laplacian Eigenmaps (LEM). For this assignment we briefly discuss two of them: Isomap and Laplacian Eigenmaps, in particular we were asked to apply these two techniques to a subset of the Triesch Dataset.

Dataset

The dataset consists in 240 greyscale images of hand postures each of which has been labeled from 1 to 10, each image is 128×128 wide and so in total each one is composed by 16384 pixels whose intensity value varies between $[0, 255]$ (from black to white). To work with our algorithms we have to convert each image in a feature vector, in particular the i^{th} feature corresponds to the i^{th} pixel, and at the end we will append the classification column so our dataset matrix is $D^{240 \times 16385}$.

The ten different hand postures try to emulate some letters of the alphabet and, for us humans, the images are pretty different from each other and simple to recognize. Hand postures can be seen in Figure 1 whose background should be white, but it has been colored consistently to the scatter plots we will use during the report in order to simplify the visual interpretation.

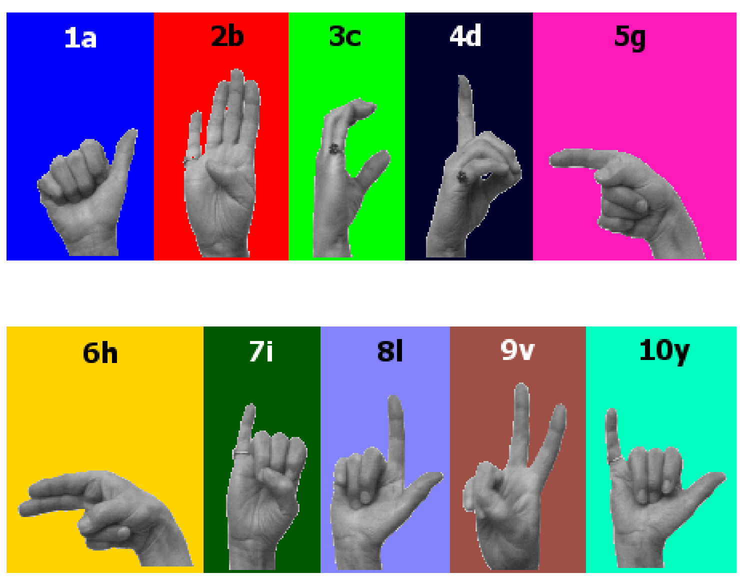

Figure 1: Hand postures with colored background to visual interpretation for scatter plots, real images have white background

Isomap

Isomap can be interpreted as a non-linear extension of MDS. As we said MDS is a linear technique, and his main characteristic is that it tries to preserve distances between each observation while embedding data in a subspace of lower dimensionality. It is good for simple phenomena but it can fail especially if the phenomenon is non-linear.

Isomap assumes that the data manifold lives in an higher dimensionality with respect to his intrinsic complexity. It extends MDS by trying to capture the shape of the manifold through approximations of geodesic distances between the observations. As we said, Isomap tries to avoid "invalid" approximations which linear algorithms otherwise capture through standard metrics like the Euclidean distance, Figure 2 may visualize this concept. The intuition behind Isomap relies on the fact that by concatenating a series of small linear steps Isomap approximates geodesic distance between distant points, this makes it an appealing algorithm over non-linear phenomena.

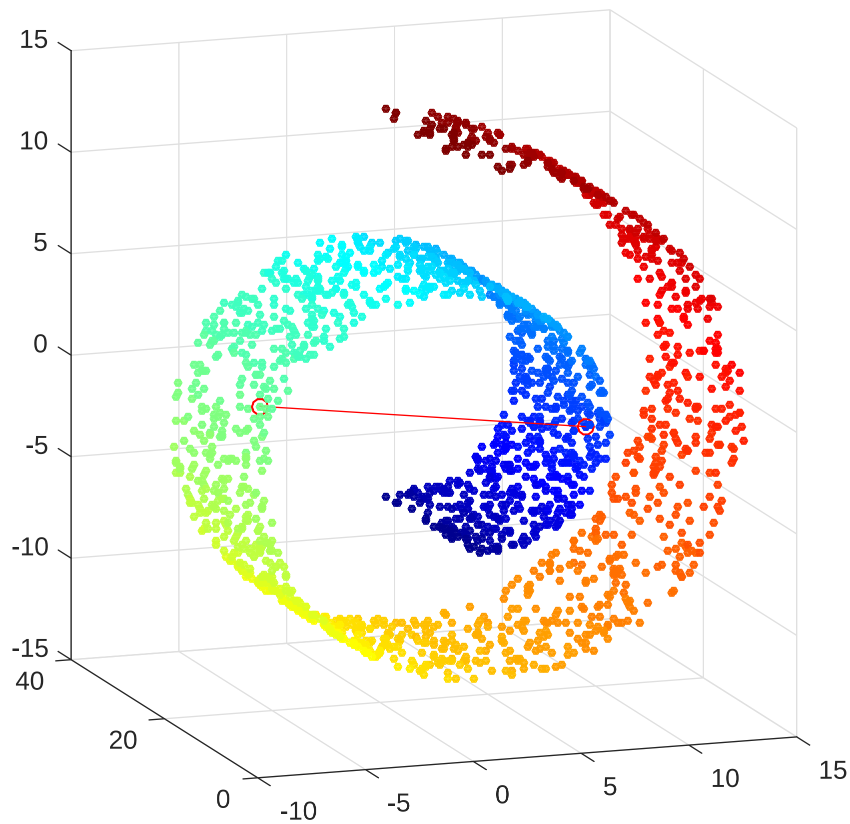

Figure 2: Swiss roll Manifold, the Euclidean distance between these two points fails to capture the real distance which should follow the shape of the manifold (geodesic distance).

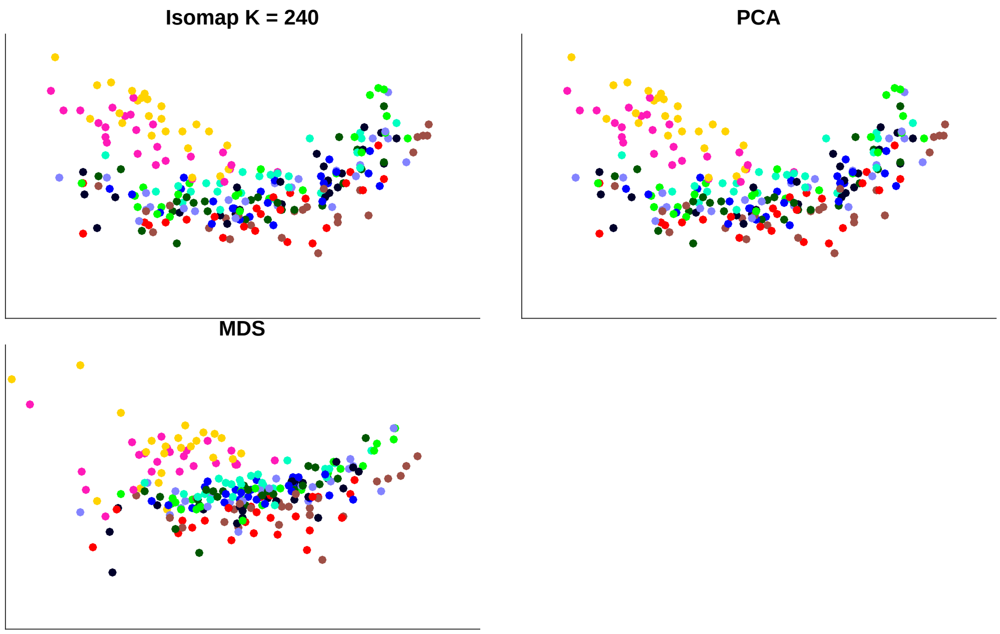

Figure 3: Comparison between 240-Isomap, PCA and MDS.

The algorithm is very simple, given a set of points $x_1 \dots x_n \in \mathbb{R}^D$:

1. Construct an Adjacency Graph W.

   - ϵ-Isomap where we put an edge between two points if one is within a fixed radius from the other
   - k-Isomap where we put an edge between two points if one is in the k nearest neighbour from the other

   As a metric to measure distances $w_{ij} = \left\|x_i - x_j\right\|^2$, we use the standard Euclidean Distance in both cases and when two nodes are not connected we set $w_{ij} = \infty$.

2. Compute approximated Geodesic Distances. We compute the shortest path distances between all pairs of points using Dijkstra or Floyd-Warshall algorithm and before the last step we square those distances in D since MDS requires a squared distance matrix.

3. Compute MDS. We return $Y = MDS(D)$ therefore we map the approximated geodesic distances to Euclidean distances living in a lower space.

Observations In this assignment we were asked to perform k-Isomap and in Figure 4 several plots represent different instances of Isomap as k varies. As we can see, by choosing a low value of k, we have a better distinction between classes, while by increasing the value we can clearly see that the representation becomes less descriptive. The reason relies on the fact that with a small k we have a better geodesic approximation between distant points, in fact if we set $k = 240$ the shortest distance between each point is the Euclidean distance itself and the result it's almost identical to classical MDS and PCA as shown in Figure 3.

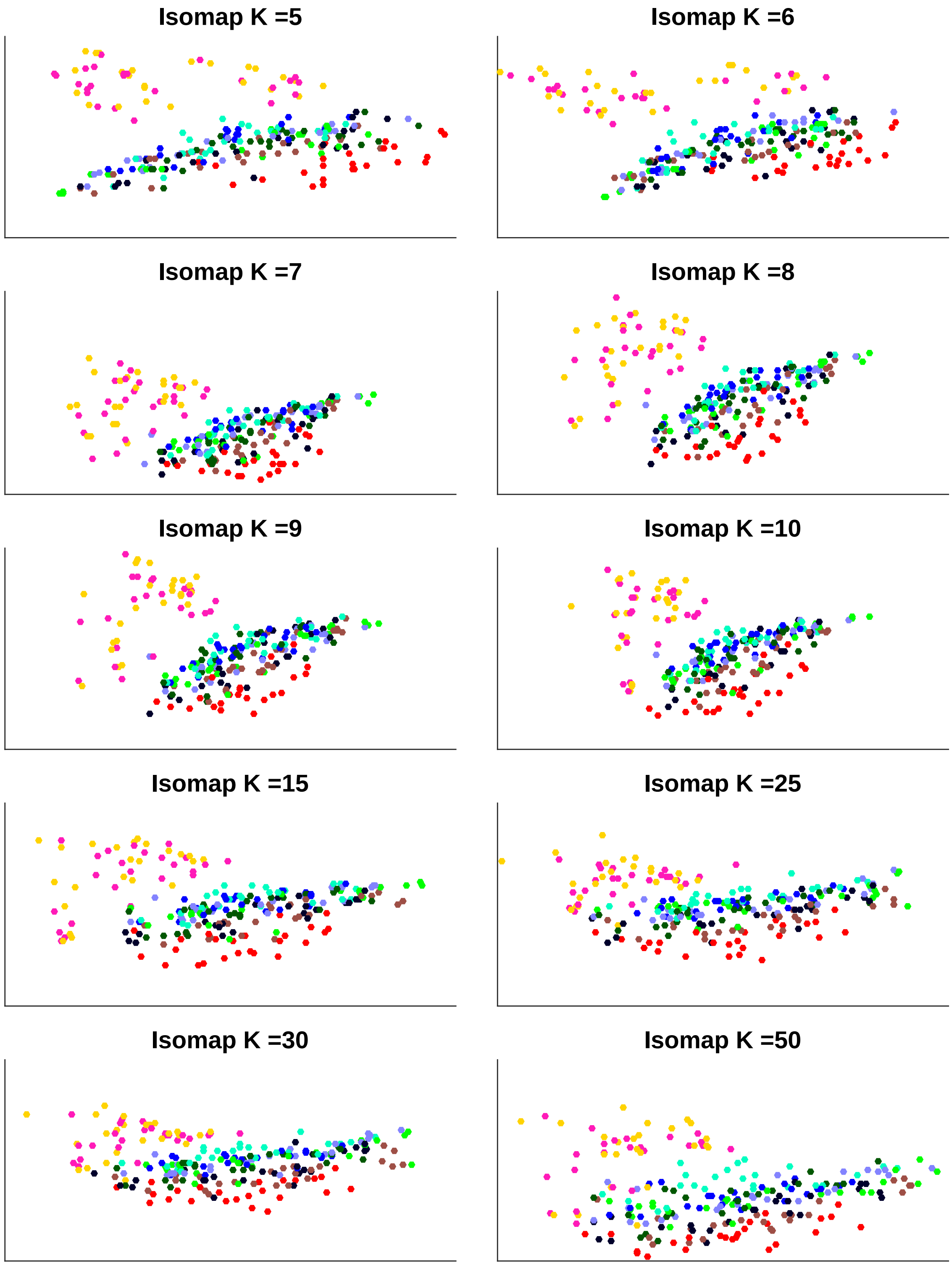

Figure 4: K-Isomap scatter plots of a subset of the Triesch Dataset, with different k.

Laplacian Eigenmaps (LEM)

While Isomap is an extension of MDS, this technique exploits spectral graph theory as the main justification. Although it is different on the approach, the algorithm follow a similar structure.

The steps are very simple, given a set of points $x_1 \dots x_n \in \mathbb{R}^D$:

1. Construct an Adjacency Graph W. We have two variations as we had in Isomap:

   - ϵ-LEM where we put an edge between two points if one is within a fixed radius from the other
   - k-LEM where we put an edge between two points if one is in the k nearest neighbour from the other

   As a metric to measure distances, we use the standard Euclidean Distance in both cases.

2. Choose weights for the edges W_{ij}. In this case we have two possibilities:

   - Gaussian Kernel, where if two points have an edge we set the weight to be $W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$ otherwise 0
   - Simple minded, where if two points have an edge we simply set the weight to $W_{ij} = 1$ otherwise 0. Which is the same as the previous with parameter $2\sigma^2 = \infty$

3. Compute Eigenmaps for each connected component. That's where we use spectral graph theory, we solve the generalized eigenvalue problem for:

   $$Lf = \lambda Df$$

   Where :

   - D is a diagonal matrix defined as $D_{ii} = \sum_j W_{ij} = \sum_i W_{ij}$ since W is symmetric.
   - L is the Unormalized Laplacian Matrix, and it is defined as $L = D - W$

   The justification behind solving the generalized eigenvalue problem is that it can be shown that by doing so, we are actually finding an optimal embedding for the points which is penalizing distant points and rewarding neighboring points.

   After finding the eigenvectors we should get:

   $$\begin{gathered} Lf_0 = \lambda_0 Df_0 \\ Lf_1 = \lambda_1 Df_1 \\ \vdots \\ Lf_{k-1} = \lambda_{k-1} Df_{k-1} \\ 0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_1 \end{gathered}$$

   The first eigenvalue of the Laplacian matrix can be shown to be 0 with corresponding unitary eigenvector, so the optimal m-dimensional embedding for the i^{th} point is then given by the first m eigenvectors (f_0 excluded) accessed in the i^{th} position:

   $$x_i \rightarrow (f_1(i), f_2(i), \dots, f_m(i))$$

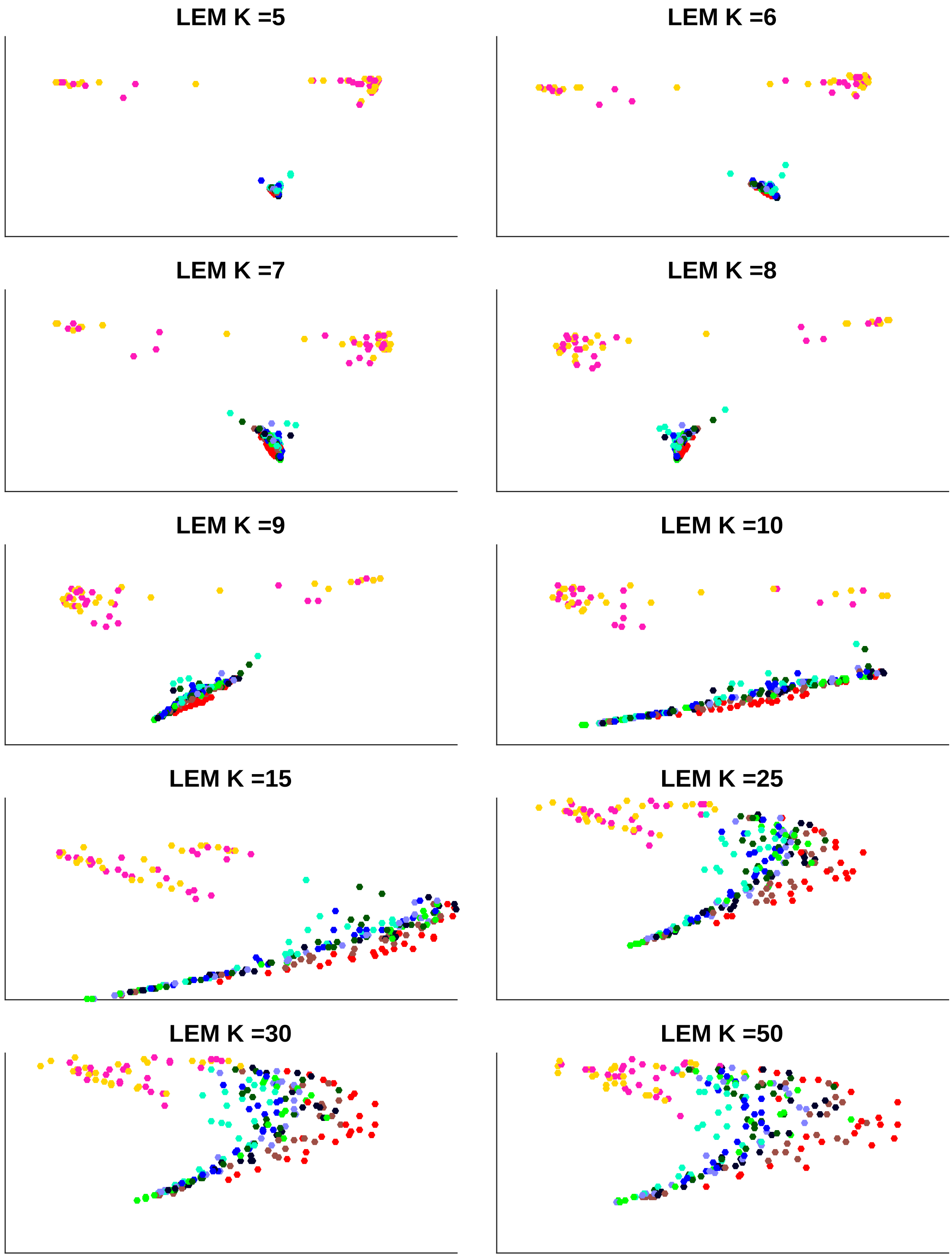

Figure 5: K-LEM scatter plots of a subset of the Triesch Dataset, with different k.

Observations Several scatter plots of the algorithm are shown in Figure 5, in the computation of LEM the simple minded approach has been used. We can clearly see that the results are pretty much the same as the ones obtained through Isomap. By increasing k we are considering new relations so the scatter plots will be less partitioned, in fact with $k = 5$ we can clearly see that hand postures (6) and (5), which are very close each other and very distant to the rest, have been put together that's why they form a cluster. As we increase k we can see that points starts to get closer to each other, in particular with 50-LEM we can see that postures starts to mix. So even if Laplacian Eigenmaps is different from Isomap, in both cases we can "almost" get the same informations.

Conclusions

Both methods examined are not able to fully capture the phenomenon of the dataset, as can be seen in the scatter plots, it is rather difficult to discriminate all the hand postures. It seems that applying these two dimensionality reduction techniques and PCA to this kind of complex problems is actually not so rewarding, at least with a small number of dimensions (two or three).

Even if the overall manifold learning process wasn't so fruitful, one degree of freedom has been captured by all the techniques, and that's the movement of the hand with respect to the wrist. In particular this permits easily to identify two classes (5) and (6) between all the others, in Figure 6 we can see that the most fruitful technique, 5-Isomap, was also able to partially discriminate posture (2). Also with 30-LEM we can detect class (2) but 5-Isomap is far more clear.

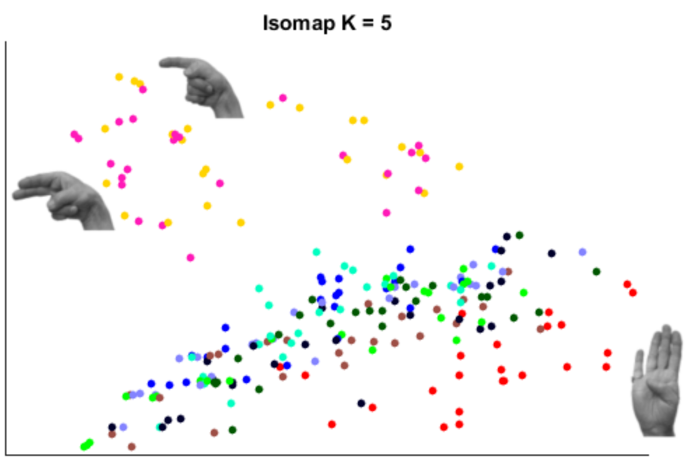

Figure 6: 5-Isomap scatter plot, with (6) (5) and (2) hand postures. We can clearly see that it was able to detect the hand-wrist degree of freedom.

Both methods analyzed have problems and assumptions:

- Isomap:
  - It assumes that not all the dimensions are useful.
  - It does not work if the manifold has some holes inside and several connected components. Since it approximates geodesic distances between all points, the global approach will not work.
  - It is difficult to choose the correct k or ϵ, in case we have labeled data the interpretation is simpler, otherwise the it becomes very difficult.

  - We don't know how many dimensions to use as output. One approach is to analyze the residual variance between the MDS embeddings and the approximated Geodesic distances as dimensions increase. One way to choose the number of dimensions is to stop when we see that variance start decreasing linearly as dimensions increase.

- Laplacian Eigenmaps:
  - Basically it performs spectral clustering, it is good for visualization, clustering data, identifying the degree of freedom of phenomena, but it doesn't preserve much of the relations between points.
  - It is difficult to tune the parameter of k and ϵ it, again if we have labeled data it is far more simple.
  - If we choose the Gaussian Kernel, it is difficult to tune the parameter $2\sigma^2$, that's why the simple minded approach is more appealing.

Even with labeled data the overall performances of the two non-linear techniques doesn't seem to provide a substantial difference from the bad performances of linear classifiers like PCA or MDS. The only improvement is that they manage somehow to partially distinguish class (2), but it is not sufficient since the majority of the classes are mixed together and are not recognizable.

Student: Francesco Pelosin
Email: 839220@stud.unive.it *OR* pelosinfrancesco@live.it
Id: 839220

References

[1] Ali Ghodsi. Dimensionality Reduction A Short Tutorial. *Science*, page 25, 2006.

[2] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[4] Florian Wickelmaier. An introduction to MDS. *Reports from the Sound Quality Research Unit,* {. . . }, page 26, 2003.

[5] Dan Ventura. Manifold Learning Examples PCA , LLE and ISOMAP. *October*, (x):1–9, 2008.

[6] Lawrence Cayton. Algorithms for manifold learning. *Univ of California at San Diego Tech Rep*, 44(CS2008-0923):973–80, 2005.

[7] Lawrence K Saul, Kilian Q Weinberger, Fei Sha, Jihun Ham, and Daniel D Lee. Spectral Methods for Dimensionality Reduction.

[8] J. Triesch and C. von der Malsburg. Robust classification of hand postures against complex backgrounds. In *Proceedings of the 2Nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, FG '96, pages 170–, Washington, DC, USA, 1996. IEEE Computer Society.