EX3 - Advanced Practical Course In Machine Learning
Manifold Learning

Daniel Afrimi
203865837

December 8, 2020

To view the plots in the document that are appropriate for each section, click on the **Figure X**

1 Theoretical Questions

1.1 PCA

In PCA we diagonalize the empirical covariance matrix of our data, $S = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (x_i - \hat{x})(x_i - \hat{x})^T$ Assume the data is centered, meaning $\hat{x} = 0$

1.1.1 PSD

For a nonzero vector $y \in \mathbb{R}^k$ we have:

$$\begin{aligned} y^\top S y &= y^\top \left(\tfrac{1}{n-1} \textstyle\sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^\top \right) y \\ &= \tfrac{1}{n-1} \textstyle\sum_{i=1}^{n} y^\top (x_i - \bar{x})(x_i - \bar{x})^\top y \\ &= \tfrac{1}{n-1} \textstyle\sum_{i=1}^{n} \left((x_i - \bar{x})^\top y \right)^2 \geq 0 \,. \quad (*) \end{aligned}$$

Therefore, S is always positive semi-definite. The additional condition for S to be positive definite is: Define $z_i = (x_i - \bar{x})$ for all i. For any nonzero$y \in \mathbb{R}^k$ () is zero if and only if $z_i^\top y = 0$ for each i. suppose the set $\{z_1, \ldots, z_n\}$ spans $\mathbb{R}^k$ Then, there are scalars $\alpha_1, \ldots, \alpha_n$ that y is linear combination of them $y = \alpha_1 z_1 + \cdots + \alpha_n z_n$ we have $y^\top y = \alpha_1 z_1^\top y + \cdots + \alpha_n z_n^\top, and therefore y = 0$ a contradiction.
Its stand for $\hat{x} = 0$ too

1.1.2 Rank

Remember that for a matrix X $Rank(X) = Rank(XX^T)$, The claim stems from the SVD, which after the decomposition the eigenvalue of XX^T is squared.

We will look on this vector x = $\begin{bmatrix} x_1 - \hat{x} \\ x_2 - \hat{x} \\ \vdots \\ x_n - \hat{x} \end{bmatrix}$

$Rank(x) = Rank(xx^T) = Rank(S)$. so if the rank of x is d, we will get that the rank of the scatter matrix id d. and this is happens only when the data (x) is on the subspace with dimension d.

1.1.3 Isometric

The new data points defined as: $y_i = x_i \cdot U_d$, when taking the first d eigen values (in order) and the first d vectors (columns of U). Because U is orthogonal matrix the $\langle u_i, u_j \rangle = 0 for j \neq i$. in addition the rank of X is d (b).

for each $i \in [n], x_i = \sum_j^d \langle x_i, u_j \rangle u_j$ this is one way to represent each vector by oronormal vectors. without assuming generality the distance between two different vector x_i, x_k:

$$\left\|x_i^T - x_k^T\right\|_2^2 = \langle x_i^T - x_k^T, x_i^T - x_k^T \rangle = \langle \sum_j^d \langle x_i, u_j \rangle u_j, \sum_j^d \langle x_k, u_j \rangle u_j = \sum_j^d \langle x_i - x_k, u_j \rangle^2$$

For y_i, y_k

$$\left\|y_i^T - y_k^T\right\|_2^2 = \left\|U_d^T \sum_j^d x_i^T - U_d^T \sum_j^d x_k^T\right\|_2^2 = \left\|U_d^T \cdot \left(\sum_j^d x_i^T - x_k^T\right)\right\|_2^2 = \left\|\sum_i^d \sum_j^d U_d^T \cdot \langle x_i - x_k, u_j \rangle u_j\right\|_2^2$$

U is orthogonal matrix

$$\left\|\sum_i^d \langle x_i - x_k, u_j \rangle\right\|_2^2 = \langle \sum_i^d \langle x_i - x_k, u_j \rangle, \sum_i^d \langle x_i - x_k, u_j \rangle \rangle = \sum_i^d \langle x_i - x_k, u_j \rangle^2$$

1.2 LLE

1.2.1

The vector we get from the sum is $w_i^T \cdot z$ and the squared magnitude of any vector y is yy^t therefore:

The zz^t is matrix consisting of all the inner products of the neighbors - Gram matrix. and than we will get:

$$RSS_i = w_i^T G w_i$$

$$RSS = \langle \sum_j w_j z_j, \sum_k w_k z_k \rangle = \sum_{k,j} w_j^T z_j^T w_k z_k = w^T G w$$

1.2.2

We want to minimize RSS_i, with the constraint that the sum of the weights is eqaul to 1.

We will use the Lagrange multiplier - λ. S0 the constraints in matrix way is $1^T w_i = 1$ where $\mathbf{1}$ is a matrix of k rows and one columns.

$$\mathbb{L}(w_i, \lambda) = w_i^T G w_i - \lambda(1^T w_i - 1) = 0$$

We will take the derivatives (G is gram matrix and therefor the matrix is symmetric):

$\frac{\partial \mathbb{L}}{w_i} = 2G_i w_i - \lambda \mathbf{1} = 0$

$\frac{\partial \mathbb{L}}{\lambda} = 1^T w_i \cdot \lambda \mathbf{1} = 0$

If the Gram matrix is invertible we will get: $w_i = \frac{\lambda}{2} G_i^{-1} \mathbf{1}$

2 Scree Plot

Scree plot is a method for determining the optimal number of components useful to describe the data in MDS. The aim is to evaluate the number of components required to capture most information contained in the data.

We can see that when applied noise to the data, more eigenvalues vanished. The assumption mostly used is that the significant eigenvalues correspond to the useful data and the remaining smaller ones are contributed by noise, but when increasing the noise there are less significant eigenvalues. In **Figures 1** we can see how noise affect the eigenvalues of the decomposition of distance matrix.

3 Demonstration Algorithms on Data sets

In **Figures 2** and in **Figures 4** we can see the differences between the algorithms on the datasets. The explantion about each movie is above.

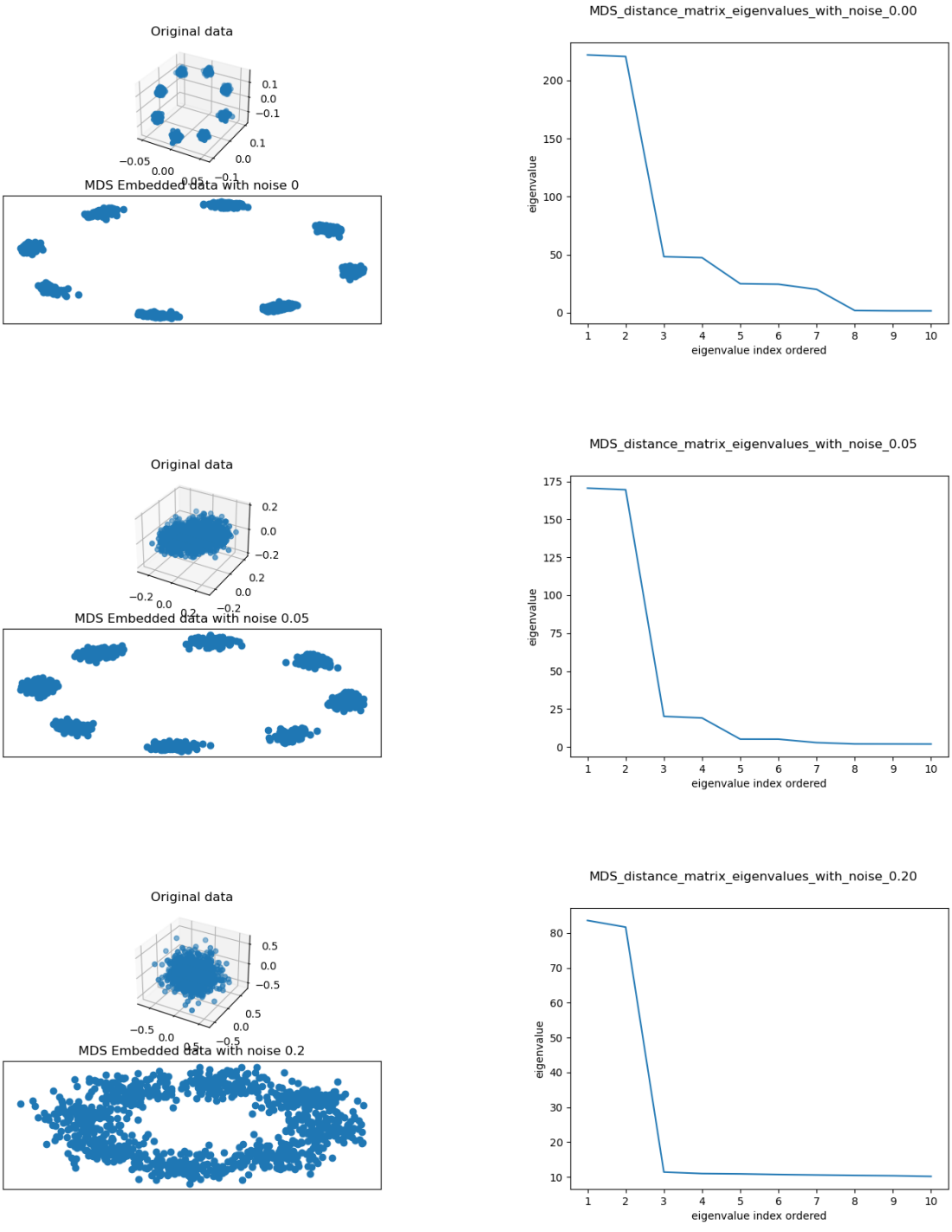
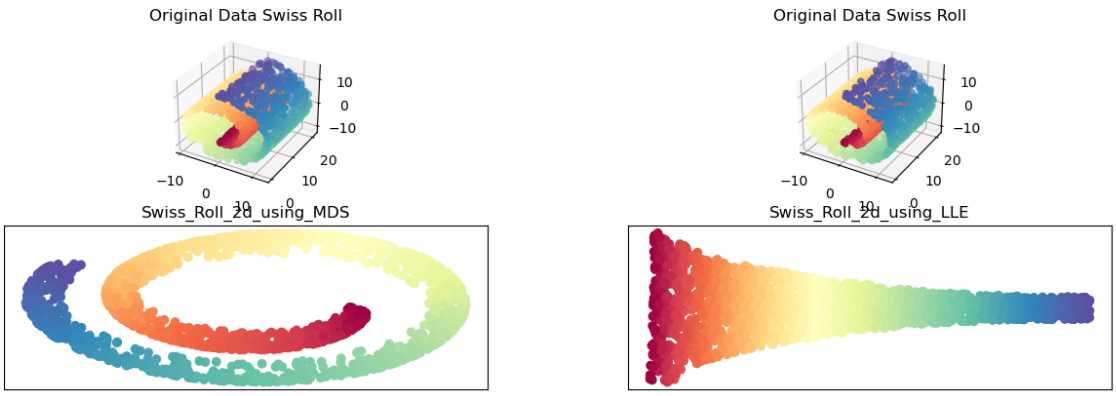


Figure 1: Scree plot - MDS with noise



(a) Swiss Roll MDS 3D to 2D

(b) Swiss Roll LLE 3D to 2D

Figure 2: Embedded Data on Swiss Roll datasets using LLE, MDS Algorithms

3.1 MDS

MDS is an algorithm that use the target metric space from information about inter-point distances measured in some other metric space. MDS problems often occur in data analysis, representation and visualization.

3.1.1 Swiss Roll

MDS not so good on this dataset. this is because the data are non linear, and MDS assume it is linear subspace for the data. MDS preserve the euclidean space, therefore we can see that two points the are far in 3D is closer on 2D (after the projection) - for example a point on the purple grid and a point on the orange grid.

3.1.2 Faces

We can see that MDS scatter plot is different for each execution of the algorithm and there are several face images are in the wrong position in the face order. This means that MDS performs poorly in nonlinear dimensionality reduction because it uses euclidean distance to calculate the points distance or similarity. Although the dark images is on the left, while the white ones on the right.

3.2 LLE

The algorithm assume that any date point in a high dimensional ambient space can be a linear combination of data points in its neighborhood. In other words, a data point has its neighborhood deciding its sufficient statistics. Alignment of such local linear structures can lead to a global unfolding of data manifolds, often described as fit locally and think globally.

3.2.1 Swiss Roll

The color coding illustrates the neighborhood-preserving mapping discovered by this algorithm. Unlike LLE, projections of the data by PCA or MDS map faraway data points to nearby points in the plane, failing to identify the underlying structure of the manifold.

3.2.2 Faces

Images of faces mapped into the embedding space described by the first two coordinates of LLE. Illustrating one particular mode of variability in pose and expression.

3.3 Diffusion Maps

Diffusion maps is a dimension reduction technique that can be used to discover low dimensional structure in high dimensional data. It assumes that the data points, which are given as points in a high dimensional metric space, actually live on a lower dimensional structure. To uncover this structure, diffusion maps builds a neighborhood graph on the data based on the distances between nearby points.

3.3.1 Swiss Roll

In **Figures 3** we can see the results of diffusion map on this data set. the result not look quite good, this is because the optimal hyper parameters. I tried several parameters in order to get good results, and those on the figure are the best.

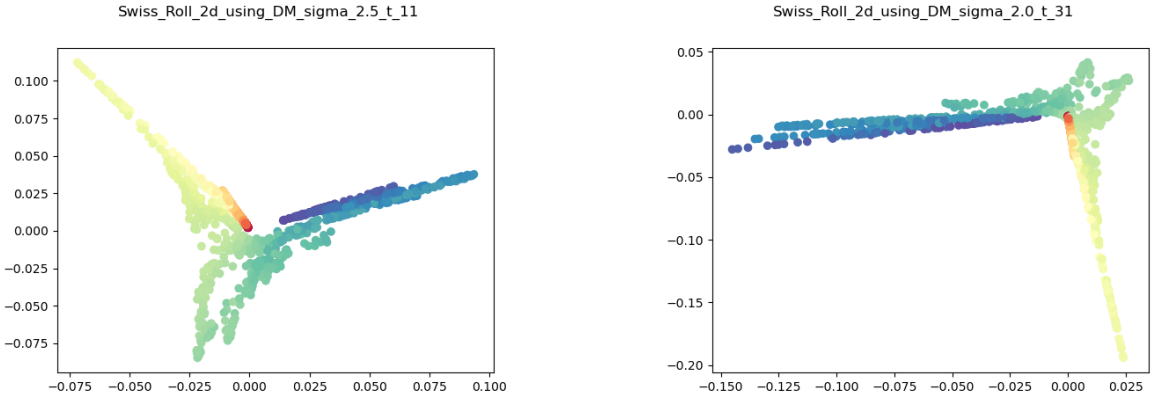


Figure 3: Dimension Reduction on swissroll data set by Diffusion Maps

3.3.2 Faces

In **Figures 5**, it can been seen that the Diffusion map face order is right and it can successfully show the gradually changed angles of captures faces and the illumines.

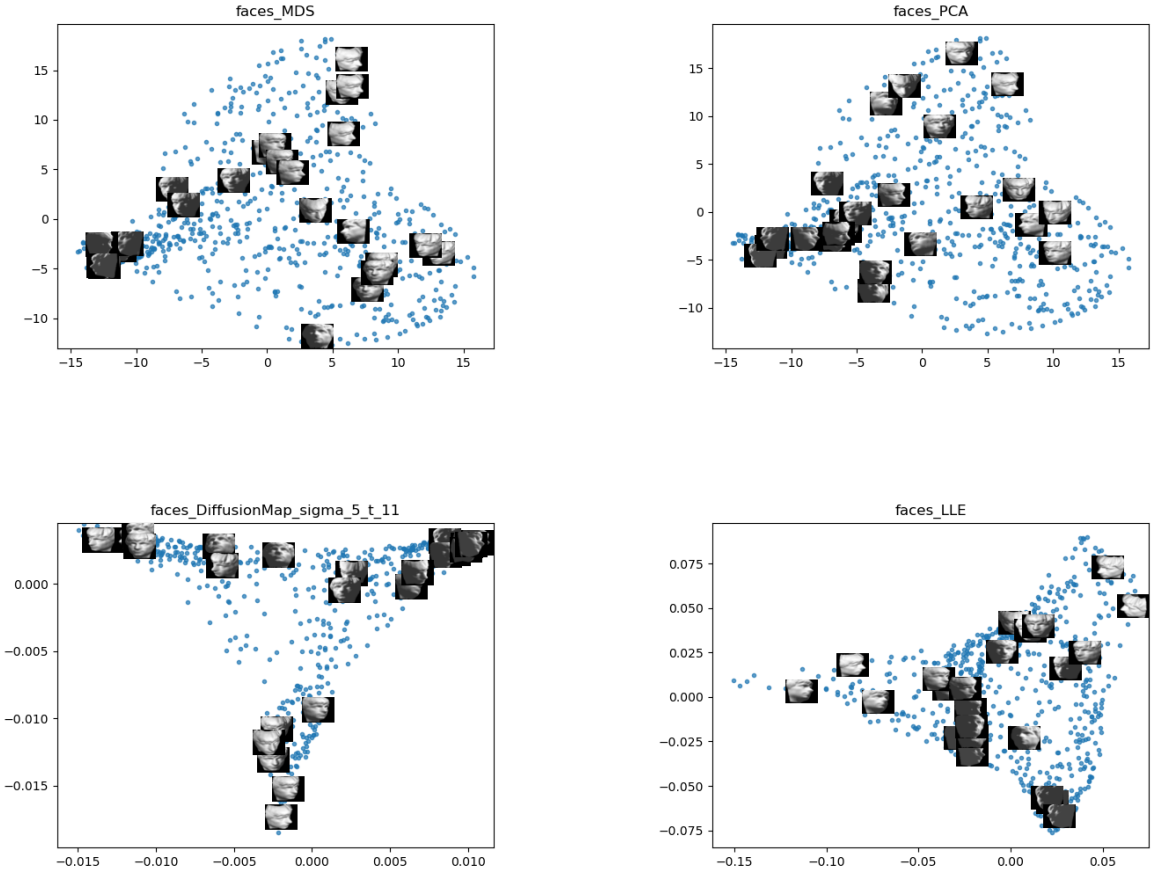


Figure 4: Dimension Reduction on faces data set with different algorithms

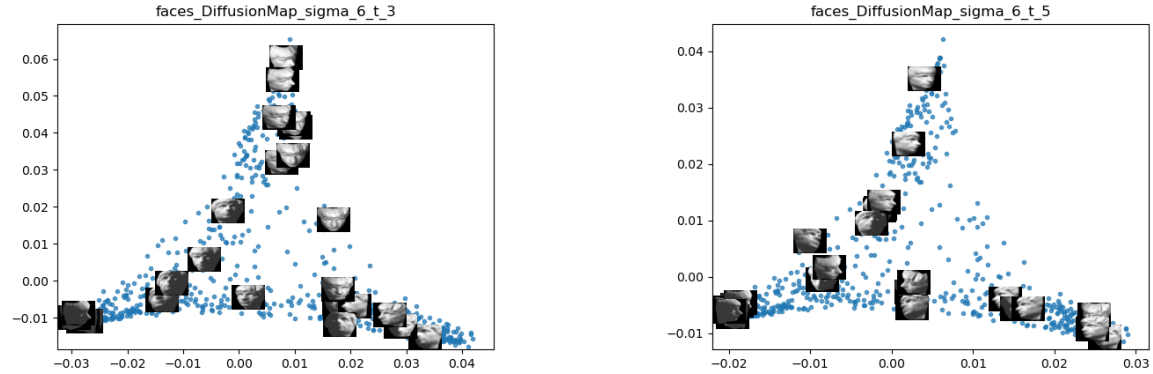


Figure 5: Dimension Reduction on faces data set by Diffusion Maps

4 Netflix Prize Dataset

The dataset consists of Ratings for 17,770 movies by 2,649,430 users.
The matrix (moviesID as columns and userID as rows) is highly sparse: around %1 of the entries are non-zero (there is no rating for all movies by all the users). Also, we are provided with an array that connects movie ID to tile and year of production, which ranges from 1896 to 2005.

4.1 Pre-Processing

The folder contains files named 'combinedDataX', where in each text file there's info about ratings of users to different movies. Every time a new movie starts, there's a line with its id and ':', and then after this line. the lines in the format of: "'customerId, rating, date"'.

I took a code that some member of the course published and obtained a matrix that its rows are the moviesID and the columns are the userID, and a matrix of the movies with there publication date. because the matrix that we got is huge, and sparse (see code).

The problem that any movie recommendation algorithm (like the netflix prize dataset) faces is to predict how well a user would like a movie he hasn't seen. This can be thought of as predicting the missing values in the user-movie matrix. the matrix also contains missing values as every user hasn't seen every movie.

I have taken the most 3000 movies (which more users rated them), and for the userID, in order to work in efficent way, and to avoid sparse matrix as long as i can (the new matrix is less sparse.

So I left with much smaller matrix that I could work with (and less sparsity). after this we have learned in the lecture than even if the data is non-linear, its good to preform PCA over the data, and that exactly what I have done, tried several components for PCA, and i have worked with TruncatedSVD. for getting the best dimension d (smaller that the real dimension), I used a scree plot to check which features are more important that the other.

4.2 Statistics on the Data

I did histogram on the rating, and In **Figures 6** we can see that the mean rating for all movies was around 3.5 (out of 5), movies releases date (we can see that most of the movies in the data are from 2000+).

4.3 MDS

I preformed data reduction by this algorithm, after the PCA stage, I've reduce the dimension to 2D and preform K-means on this data(for clustering).I chose the clustering number when using scree plot (and i saw which eigenvalue are more meaningful. so the numbers of cluster seems like the number of geners in the matrix. In **Figures 7** we can see the result.

4.4 LLE

In **Figures 9** we can see the result of preforming TruncatedSVD and then LLE to 2D dimension (for visualize). For choosing the number of neighbors I've plot the data with different k. we can see that when k grows there are more density in the data (which mean we look for a larger structure), we can visualize connected spaces in the data when k is big.

4.5 Compression

It seems that the LLE algorithm captured more information about the structure of the deceleration. It seems that the realtion of the points in the data is non-linear so we get for different numbers of neighbors for each point, the data appears to be approaching to MDS.

4.6 Spectral Clustering

Like K-means, the number of cluster the i decided to use is 7. In **Figures 8** we can see the clustering of MDS and LLE.

Figure 6: Statistics on the Netflix data set

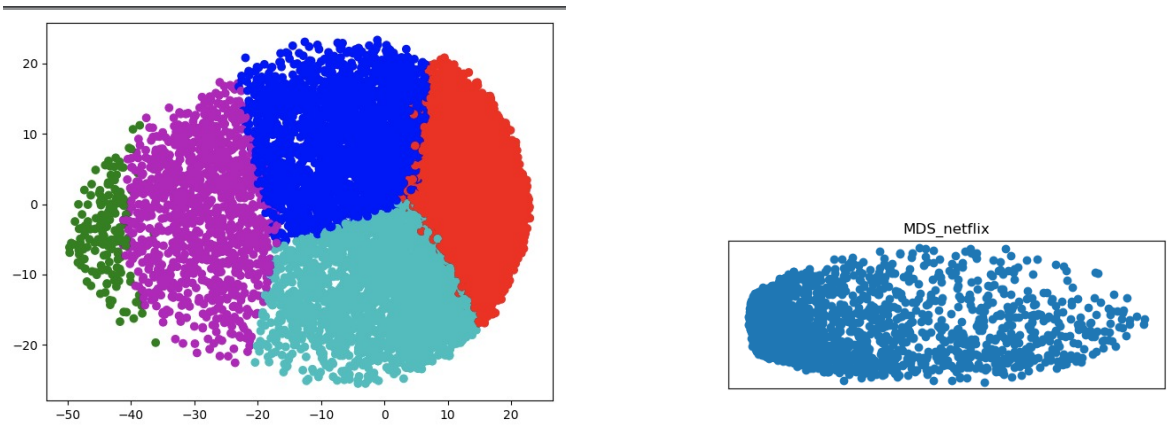


Figure 7: K-mean and MDS

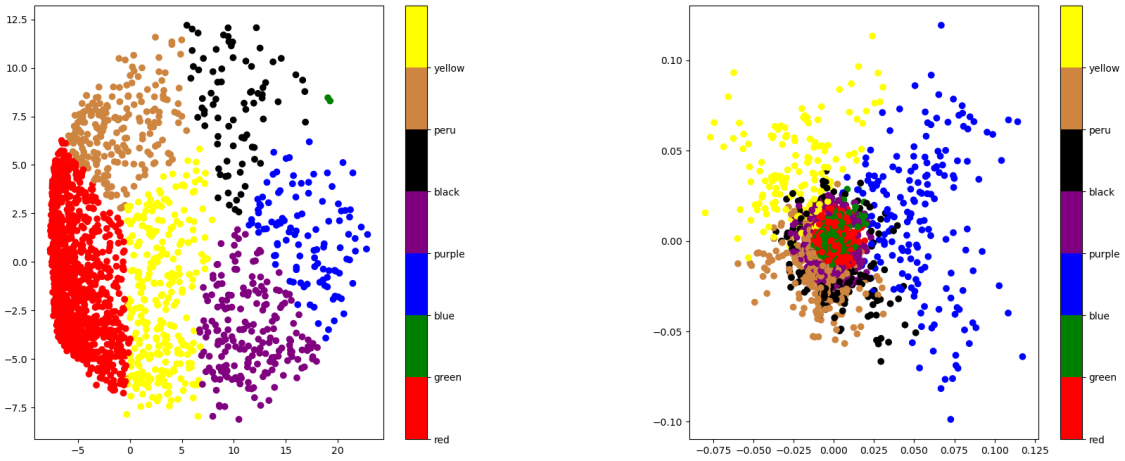


Figure 8: Spectral Clustering On MDS and LLE (left is MDS

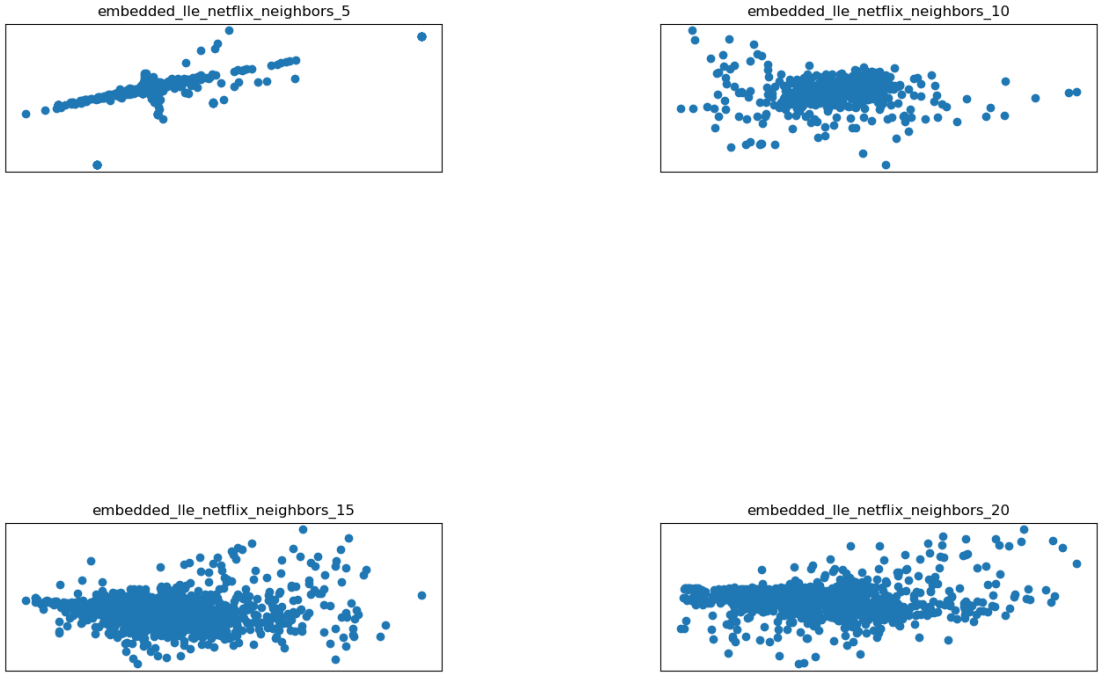


Figure 9: LLE on Netflix data with different number of neighbors

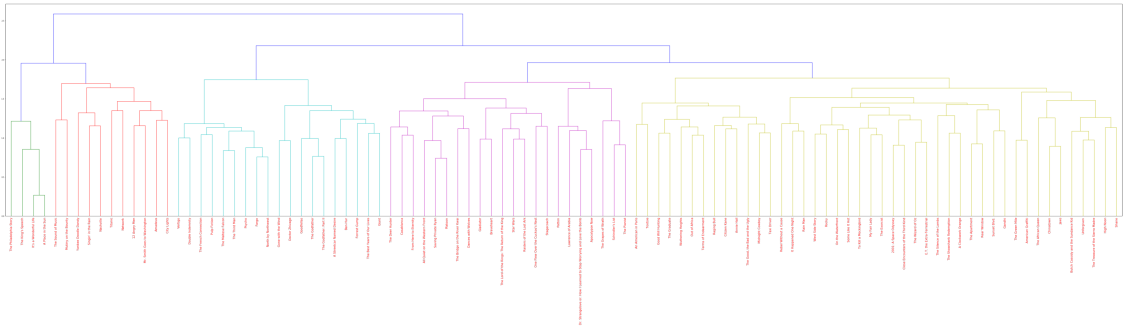


Figure 10: Diagram Movies Names

4.7 Movies Names - Structure

In order find a meaning to the movies names structure, I did some pre proccesing.
Given a sequence of characters, **tokenization** is the process of breaking it in basic semantic units with a useful and basic meaning. This pieces are called tokens.

Computers can't process anything but numbers. In order to realize any operations with text, we need first to transform the text to numbers. This process is called **vectorization**. I used Bag of Words technique.

Since the movies names are converted into vectors, we can compute the cosine similarity between them and represent the "distance" between those vectors.

In **Figures 10** we can see the diagram of the movies names.