Name: Kushal Arora UFID: 46998412

**Advanced Machine Learning**
**Project Title**
**Fall 2014**

## I. Algorithms

### 1. Locally Linear Embedding

Locally Linear Embedding[1] tries to reconstruct the data point in lower dimension using the neighborhood in higher dimension. The Locally Linear Embedding algorithm can be summarized in following steps:

1. Find neighborhood for each point
2. For each point compute the weight matrix that minimizes the reconstruction error for the point using neighborhood
3. Using this weight vector find the embedding that best reconstructs the point.

**Notes:**
This was a part of compulsory requirement. I used Scipy + Python for implementation.

**Issues:**
Algorithm crashes for any number of neighbor for face dataset. So the analysis for faces missing.

### 2. ISOMAP:

Isomap[2] works on the premise the in lower dimension the geodesic distances between the data are conserved. So the algorithm can be split into two parts.

1. First we calculate the geodesic distance for each point. Data space is discretized and metric is enforced(L2 norm) and all pair shortest path distance is evaluated using Floyd Warshall[3]
2. Once we can affinity matrix or pairwise distances between all point, Classical MDS[4] is applied to find the embedding in lower dimensional space

**Notes:**
This again was implemented using Scipy + Python

**3. Laplacian Eigenmaps**

Laplacian eigenmaps[5] uses the spectral techniques to conserves the neighborhood in lower dimension. The data is discretized and the graph generated can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. Minimization of a cost function based on the graph ensures that points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances. The algorithm can be summarized as follows:

1. Calculate the pairwise distances and form an undirected graph.
2. Calculate the Laplacian Operator on the the graph.
3. Solve the following generalized eigenvalue problem
   $$Lv = \lambda Dv$$
4. Ignoring the first eigenvector corresponding to $\lambda = 0$, next d eigenvectors give embedding in lower d dimensional space.

This can be viewed as approximate solution to N graph cut problem where attempt is that to cluster together the similar data points in lower dimension.

**4. Hessian Eigenmaps**

The method Hessian-based Locally Linear Embedding[6][7][8] (HLLE) works on the principle that Local Isometry and attempts to find the coordinates in lower domain without evaluating the isometric mapping. The key observation is that is Manifold is locally isometric to an open connected subset of $R^d$ , then the quadratic form of hessian function H(f ) has a (d + 1)-dimensional null space, consisting of the constant functions and a d-dimensional space of functions spanned by the original isometric coordinates.

> **Notes:**
> The results for Hessian Eigenmaps for Racespace is bad. Due to time constraint further digging on the issue couldn’t be done.

**5. Local Tangent Space Analysis**

Local Tangent spaces Analysis[9][10] tries to learn the embedding by aligning the local tangent spaces at each point. It begins by computing the *k*-nearest neighbors of every point. It computes the tangent space at every point by computing the *d*-first principal components in each local neighborhood. It then optimizes to find an embedding that aligns the tangent spaces.

**Notes:**

This algorithm was picked because of two reasons:

1. It is simple to implement and is very similar to LLE
2. Conception beauty of finding the embedding by straightening out the fold by aligning tangent spaces.

**6. T Stochastic Neighbor Embedding (Used Off the shelf)**

T Stochastic Neighbor Embedding[11] is among the best unsupervised algorithm for Dimensionality reduction. This takes the probabilistic view of the problem putting a t student kernel at each point and assigning a probability distribution for neighborhood of each point. Then it tries to conserve the distribution in lower dimension while learning the embedding.

**Notes:**

TSNE generates best embedding for Digits data set.

**II. Datasets Used:**

1. Swiss Roll

This dataset was used using make_swiss_roll function of scikit learn library.

2. Racespace

Only registered images were used from this data set. Due to lesser number of samples, classes Indo-Arier and Indianer were dropped and only Caucasian, Negroid and Mongolian were used for analysis.

3. Faces

This is Olivetti Faces dataset. This dataset too is available in scikit learn.

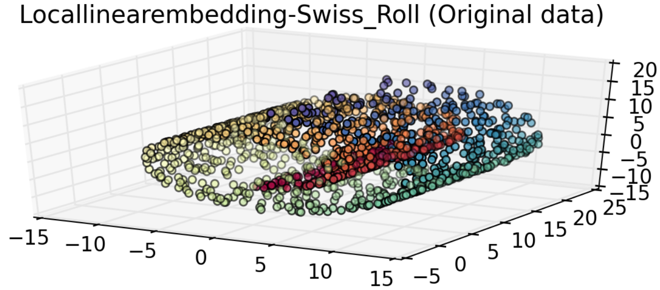
4. Digits

This is MNIST digit dataset. This dataset too is available from scikit-learn.

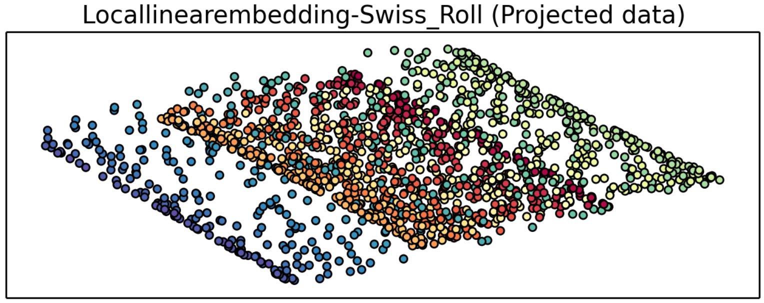
## III. Results:

### Results::Swiss Roll

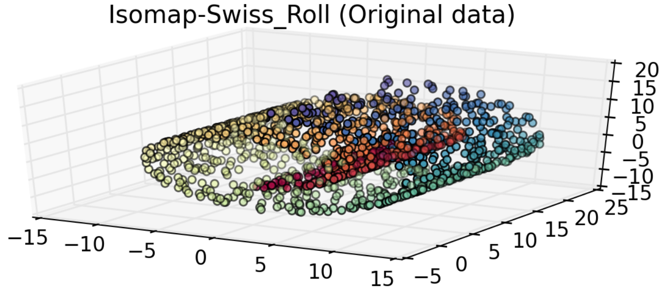
Locallinearembedding-Swiss_Roll (Original data)



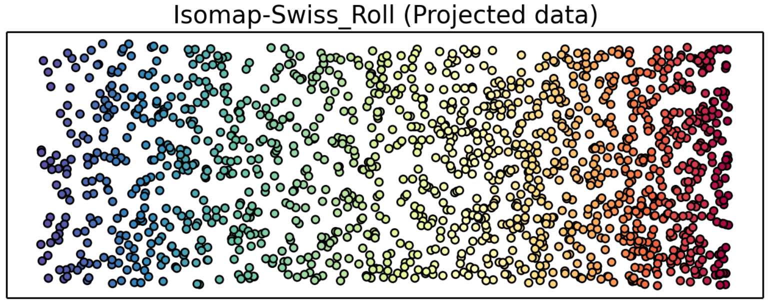

Locallinearembedding-Swiss_Roll (Projected data)



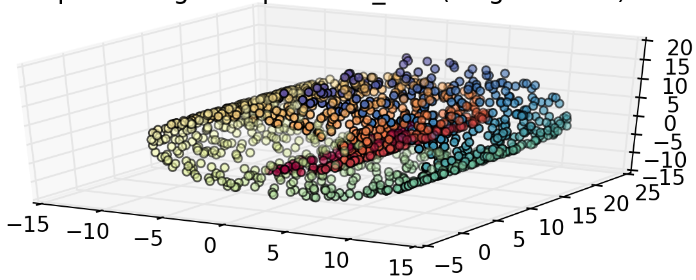
Isomap-Swiss_Roll (Original data)



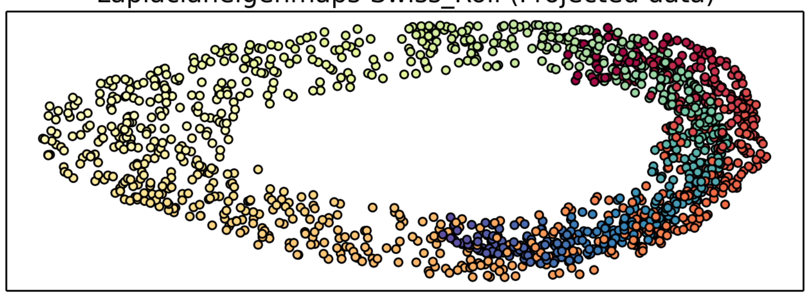

Isomap-Swiss_Roll (Projected data)



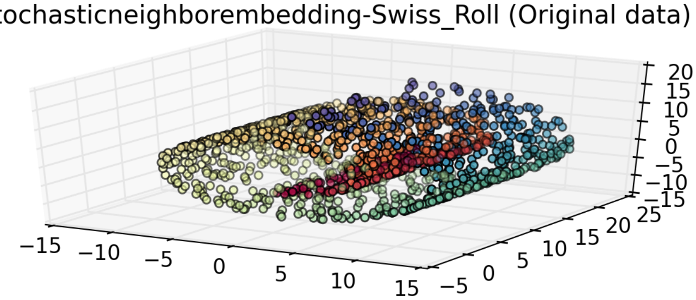
Laplacianeigenmaps-Swiss_Roll (Original data)



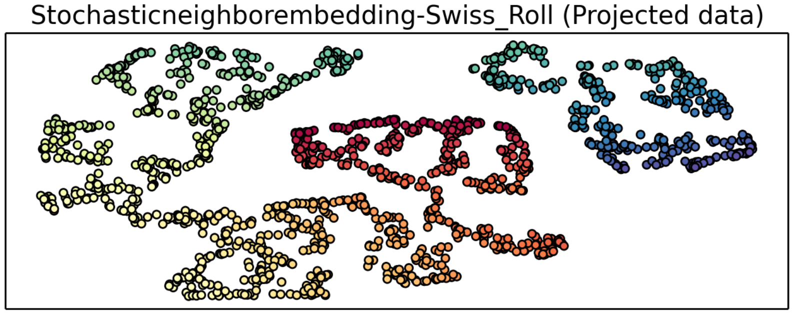

Laplacianeigenmaps-Swiss_Roll (Projected data)



Stochasticneighborembedding-Swiss_Roll (Original data)

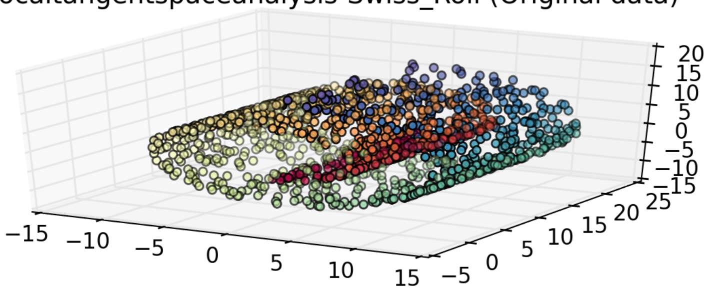



Stochasticneighborembedding-Swiss_Roll (Projected data)

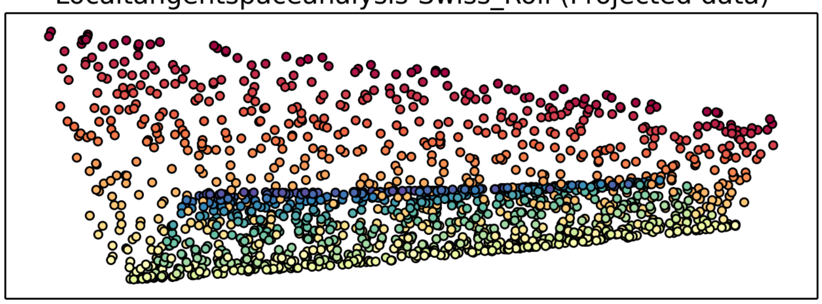


Name: Kushal Arora UFID: 46998412

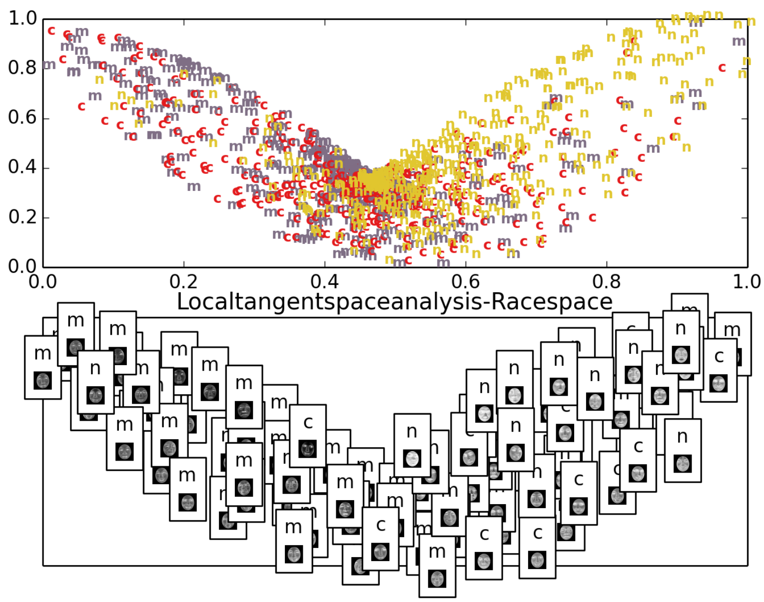
Localtangentspaceanalysis-Swiss_Roll (Original data)

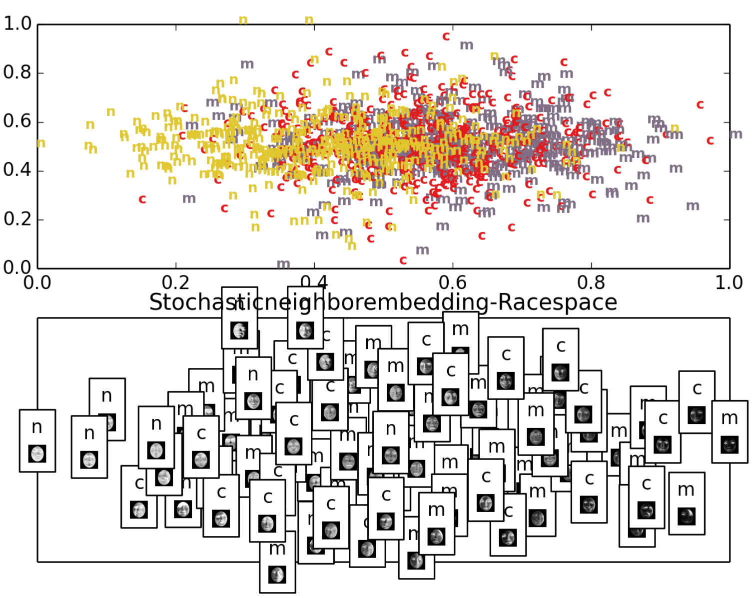



Localtangentspaceanalysis-Swiss_Roll (Projected data)

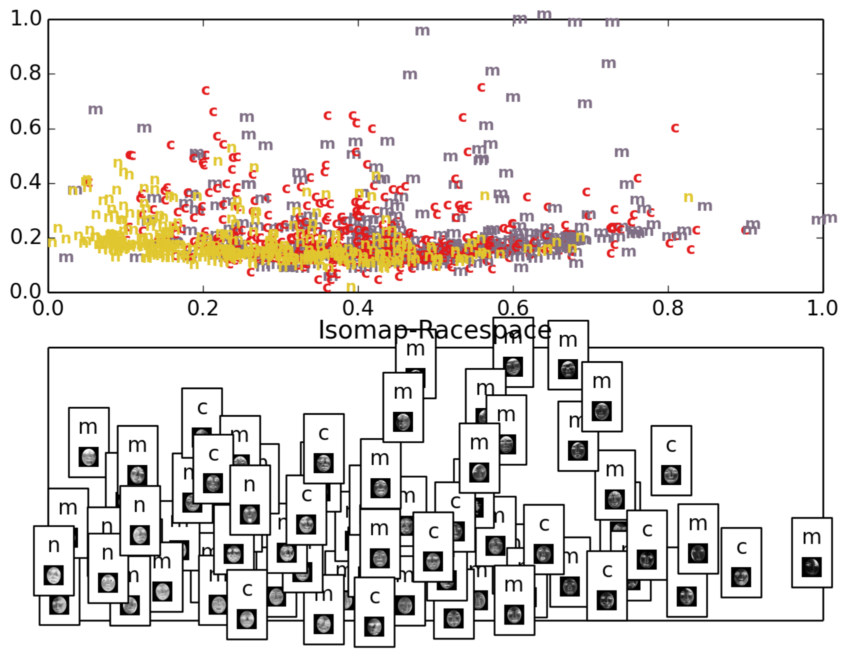


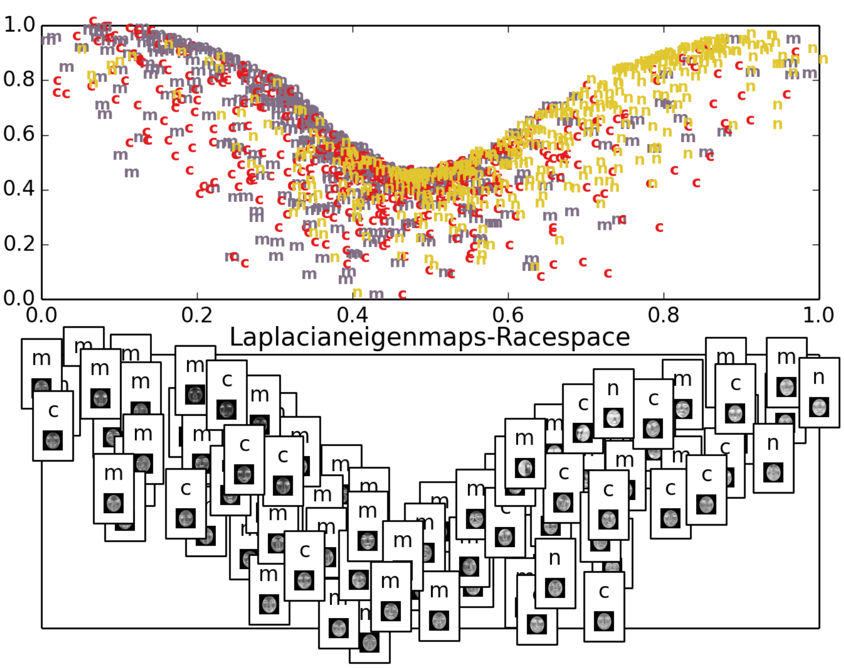
## Results: Racespace



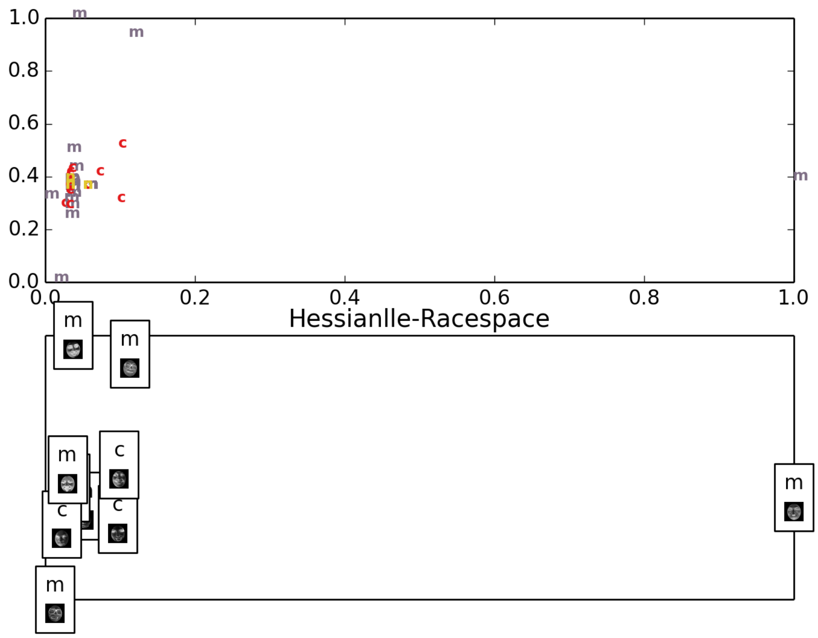



Name: Kushal Arora UFID: 46998412



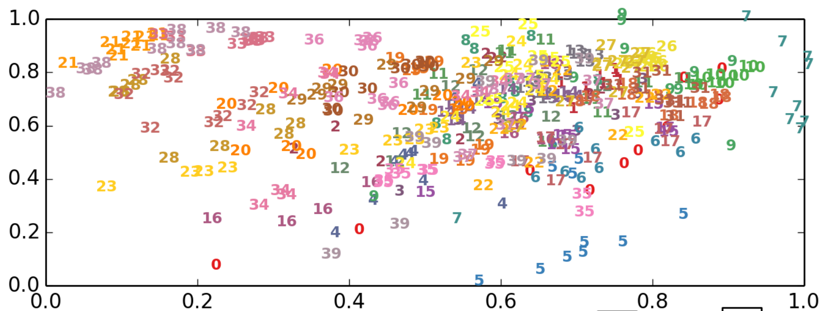



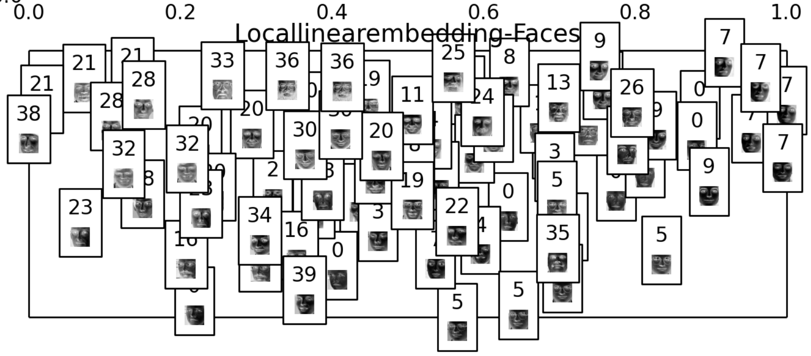
Name: Kushal Arora UFID: 46998412



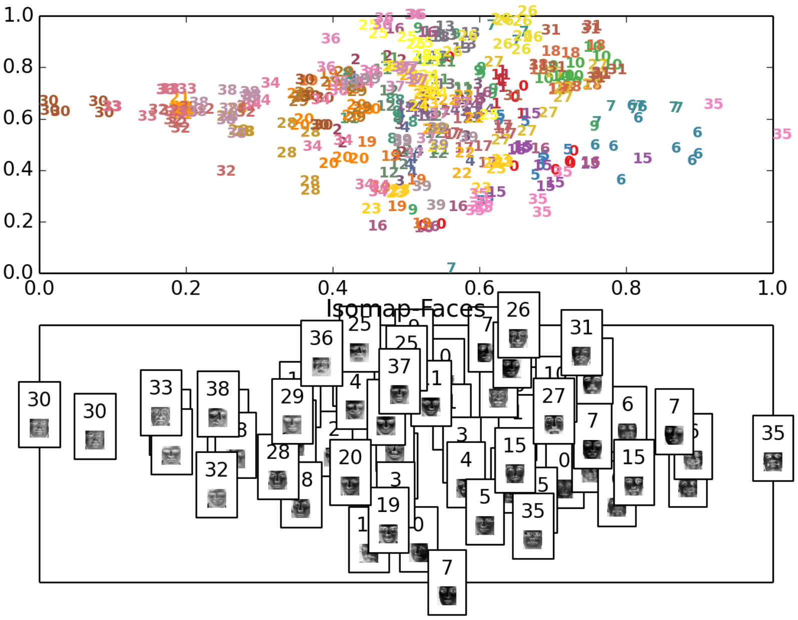

**(3D projection of the data is available in output directory in attached code)**

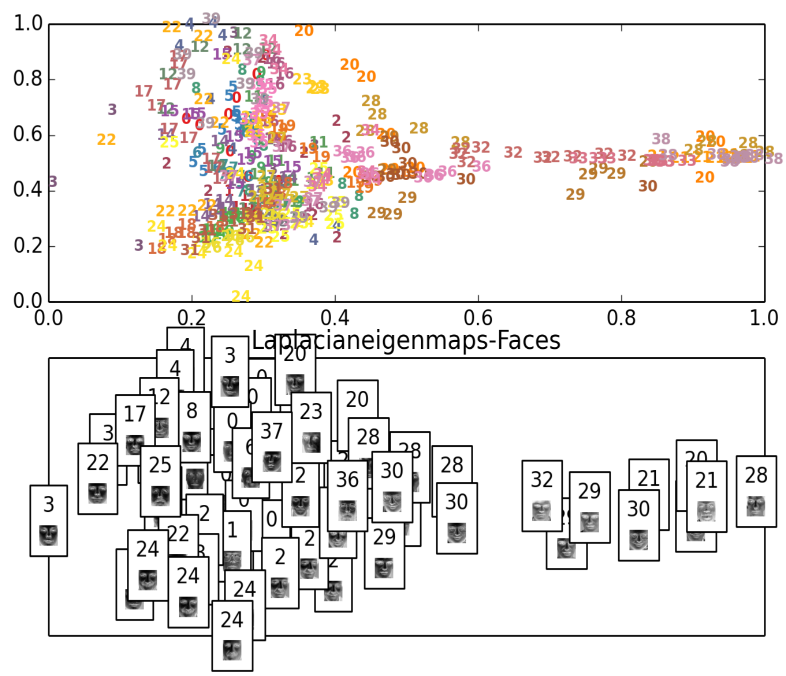
## Results::Faces

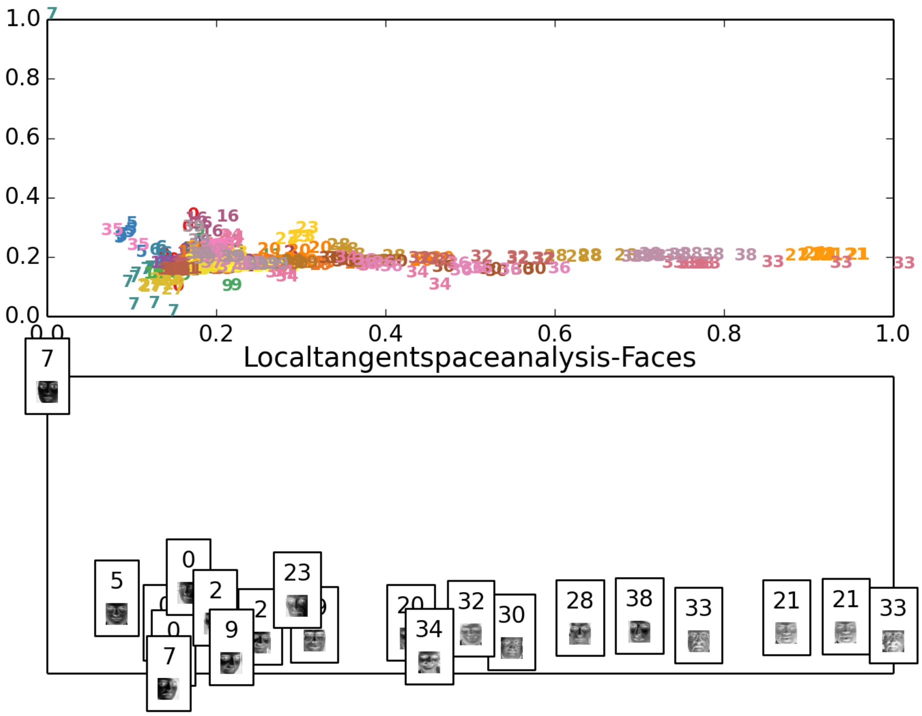




Name: Kushal Arora UFID: 46998412



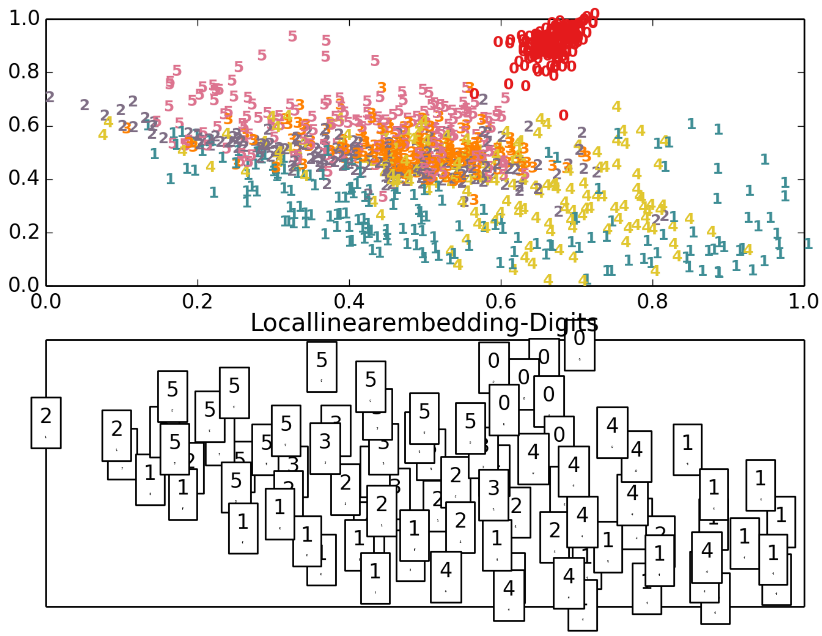



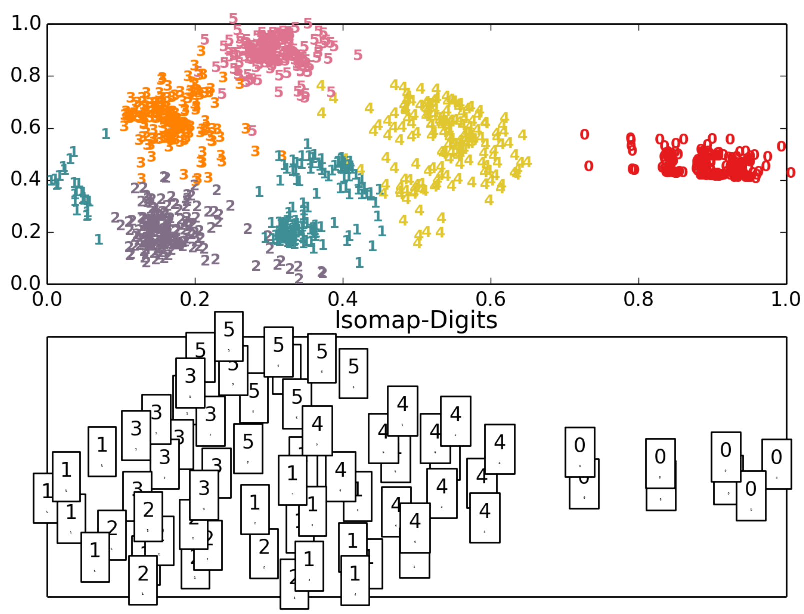



‘

**(3D projection of the data is available in output directory in attached code)**

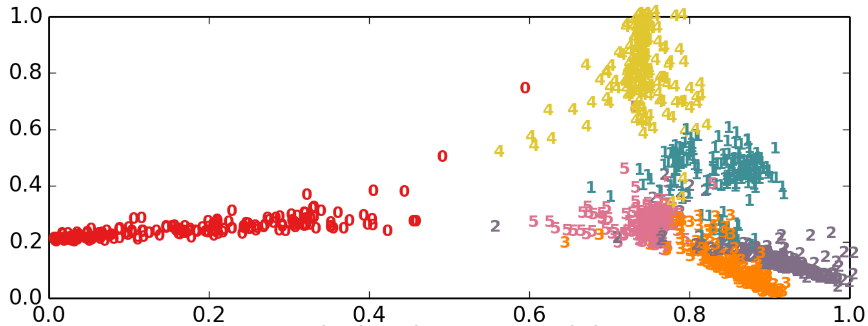
## Results::Digits



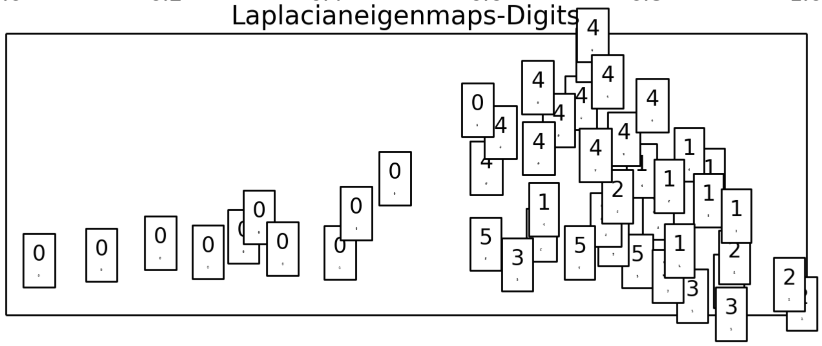



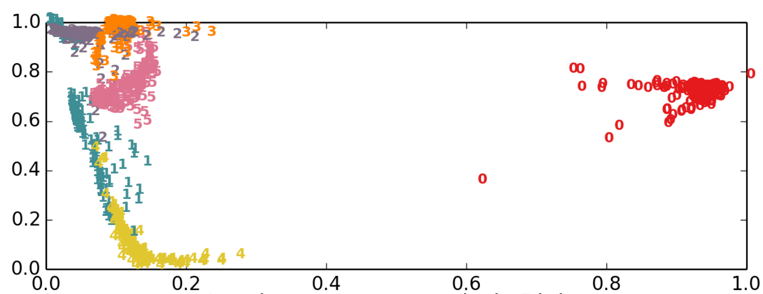
Name: Kushal Arora

UFID: 46998412

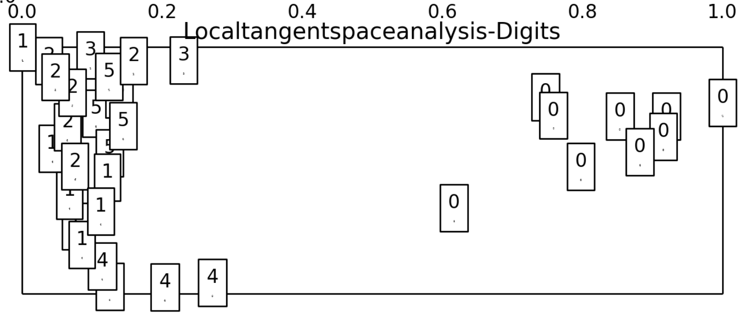



Laplacianeigenmaps-Digits

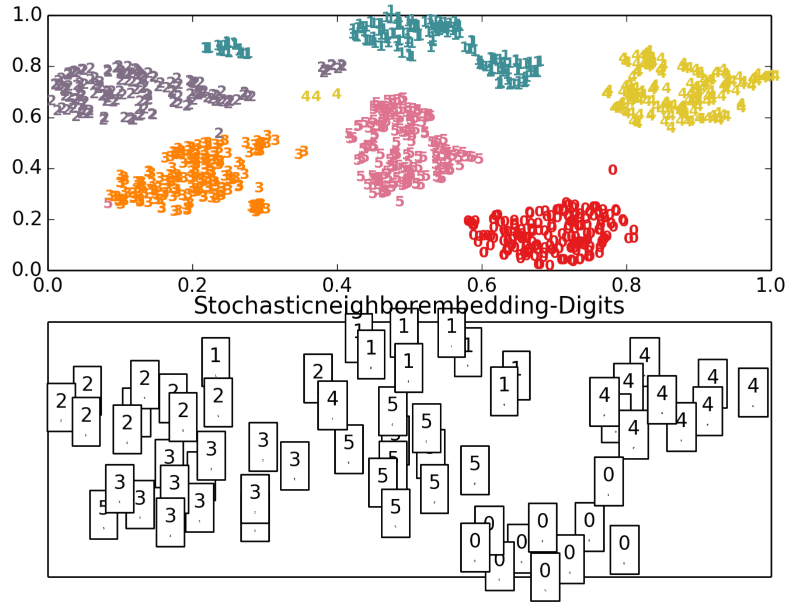


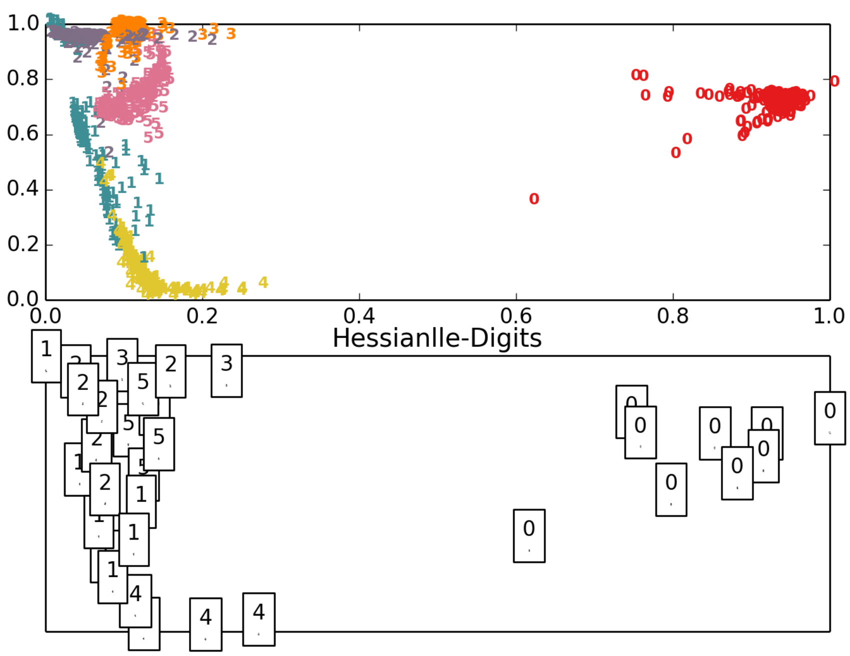



Localtangentspaceanalysis-Digits



Name: Kushal Arora UFID: 46998412




**(3D projection of the data is available in output directory in attached code)**

**IV. References:**

[1] Roweis, Sam T., and Lawrence K. Saul. "Nonlinear dimensionality reduction by locally linear embedding." *Science* 290.5500 (2000): 2323-2326.

[2] Tenenbaum, Joshua B., Vin De Silva, and John C. Langford. "A global geometric framework for nonlinear dimensionality reduction." *Science* 290.5500 (2000): 2319-2323.

[3] Cormen, T. H., Ch E. Leiserson, and R. L. Rivest. "The floyd-warshall algorithm." *Introduction to Algorithms* (1990): 558-565.

[4] MULTIDIMENSIONAL SCALING, http://forrest.psych.unc.edu/teaching/p208a/mds/mds.html

[5] Belkin, Mikhail, and Partha Niyogi. "Laplacian eigenmaps and spectral techniques for embedding and clustering." *NIPS*. Vol. 14. 2001.

[6] Donoho, David L., and Carrie Grimes. "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data." *Proceedings of the National Academy of Sciences* 100.10 (2003): 5591-5596.

[7] Ye, Qiang, and Weifeng Zhi. Discrete hessian eigenmaps method for dimensionality reduction. Technical report, Department of Mathematics, University of Kentucky, 2012.

[8] Nonlinear Manifold Learning II
http://web.mit.edu/6.454/www/www_fall_2003/esuddert/manifold2_talk.pdf

[9] Zhang, Zhen-yue, and Hong-yuan Zha. "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment." Journal of Shanghai University (English Edition) 8.4 (2004): 406-424.

[10] Sui, Yuelei. "Local Tangent Space Alignment." (2012).

[11] "Visualizing High-Dimensional Data Using t-SNE" van der Maaten, L.J.P.; Hinton, G. Journal of Machine Learning Research (2008)