


MANI fold Learning Matlab Demo

[DOWNLOADS](#)
[DOCUMENTATION](#)
[LINKS](#)

Todd
Wittman
Department
of
Mathematics
University
of
Minnesota



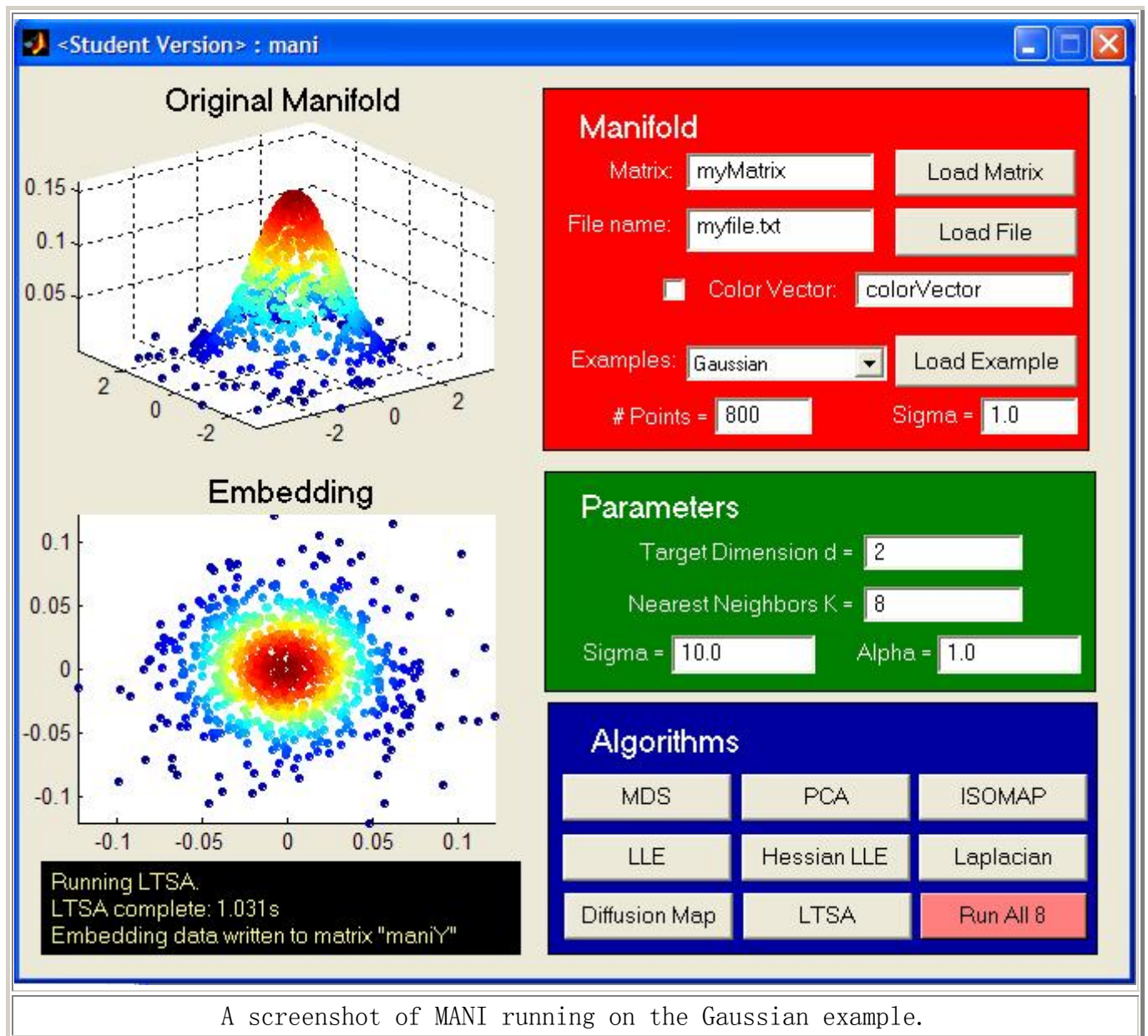
DOWNLOADS

In the last few years, many manifold learning techniques have been developed for Non-Linear Dimensionality Reduction (NLDR). To my knowledge, no one has surveyed the techniques to determine which method performs the best and under which conditions each method is appropriate. To explore the different methods, I put together 8 manifold learning techniques and 9 basic examples into one Matlab GUI. I named it "mani", short for MANIfold Learning Matlab Demo. I hope this simple program can serve as an instruction tool and research aid for the manifold learning community. If you find the GUI useful or have comments or questions, please e-mail [Todd Wittman](mailto:wittman@math.umn.edu) at wittman@math.umn.edu.

mani.m	MANIfold Learning Matlab Demo m-file <i>Just download this file, start Matlab, and type "mani".</i>
mani_presentation.ppt	PowerPoint presentation "Manifold Learning Techniques: So Which is the Best?" <i>This is a presentation I gave in Prof. Gilad Lerman's class on geometric data analysis at University of Minnesota.</i>

DOCUMENTATION

The MANI GUI is pretty straightforward to use, just push buttons and change parameters. To begin using it, simply download the [mani.m](#) file, start up Matlab, and type "mani". Below is some limited documentation on what all the buttons do.



A screenshot of MANI running on the Gaussian example.

MANIFOLD: Reading the input data.

The input data can be read from a matrix in the workspace, from a text file, or selected from one of 8 built-in examples. Each row of the input data should be a data element. After input is read, it is automatically written to the workspace variable "maniX". If the example is 2D or 3D, the data will be displayed in the Manifold plot in the top left corner. The user can specify a color vector to color the scatterplot and its accompanying embedding.

- Load Matrix: Enter the name of a matrix in the Matlab

workspace and press this button. The matrix should be $N \times D$, where N is the number of data items and D is the dimension of the manifold.

- Load File: Enter the name of a text file and press this button. The text file should contain numbers separated by spaces with each data element as a row.
- Color Vector: Checking this box indicates that there is a N -dimensional column vector in the workspace that specifies the colors of the $N \times D$ input matrix. The colors can be specified by any list of numbers, with the colors varying linearly from red to blue in Matlab's "jet" colormap. The plot in the output embedding will also be colored by this vector. As the examples illustrate, color is very useful in understanding how the points are mapped under NLDR.
- Load Example: To explore performance under different situations, I created a set of 9 built-in examples. Several of these examples are taken from the papers describing NLDR, such as the Swiss Roll in the article describing ISOMAP. Just select an example from the pop-up menu and set the parameters. In each example, the user can specify the number of data points N . Plus, each example has a second parameter that can be set.
 - Swiss Roll: A randomly sampled plane is rolled up into a spiral. A good NLDR technique should unroll this example into a plane. If the number of nearest neighbors K is set too large, the embedding will jump across folds. The parameter "Z Scaling" sets the height of the spiral. A smaller value will make the folding more compact and thus harder to unravel. This example was proposed by Tennenbaum et. al. to show that ISOMAP preserves local geometry.
 - Swiss Hole: A hole is punched in the Swiss Roll example. The user can adjust the "Z Scaling" parameter to change the height of the roll. This example was used by Donoho & Grimes to demonstrate that Hessian LLE can handle non-convex data sets.
 - Corner Planes: A plane is bent in the middle with a "Lift Angle" in degrees specified by the user. The NLDR technique should flatten the plane out into a perfect rectangle. As the lift angle increases, it becomes harder to flatten. If the angle is greater than 90 degrees, the half-planes

may be folded onto each other and data points may be written on top of each other.

- Punctured Sphere: The bottom 3/4 of a sphere is sampled non-uniformly. The sampling is densest along the top rim and sparsest on the bottom of the sphere. The 2D projection should display concentric circles. The user can set the "Z Scaling" to control the height of the sphere. This example was written by Saul & Roweis for LLE.
- Twin Peaks: Two corners of a rectangular plane are bent up. The 2D projection should show a roughly rectangular shape with blue and red in opposite corners. The user can control "Z Scaling" to change the degree of the bend. This example was written by Saul & Roweis for LLE.
- 3D Clusters: Random 3D points are assigned to tight non-overlapping data clusters. Since most NLDR techniques require a connected data set, the clusters are randomly connected with blue 1D line segments. The user can specify the number of clusters desired. A good NLDR technique should preserve clusters, as shown by the color groupings.
- Toroidal Helix: A 1D curve is coiled around a helix. There is a small amount of noise in the sampling along the helix. The user can set the "Sample Rate", which controls the uniformity of the sampling by changing the exponent on the curve parameter t . A value of 1.0 has uniform sampling. The sampling becomes more non-uniform as the value gets larger, with the denser portion at the back of the helix in the blue points. The NLDR method should unravel the coil into a perfect circle. This example was written by Coifman & Lafon to demonstrate Diffusion Maps.
- Gaussian: Points are drawn from a Gaussian distribution with standard deviation "Sigma" specified by the user. The NLDR method should form concentric circles, although this becomes more difficult as the value of Sigma gets smaller.
- Occluded Disks: A set of 20x20 binary images are generated with a disk of radius 3 at a random center. The manifold plot actually shows the centers of the disks in a 20x20 grid. The user can

specify the radius of an occluding disk that will be present in every image. Setting radius $r=0$ will result in no occluding disk. This example tests the NLDR methods on high-dimensional image data and the occluder creates a non-convex data set. The 2D embedding should reveal the centers of the circles, just like in the manifold plot. Since this is 400-dimensional data, the embedding could take a very long time to compute. This example was suggested independently by Carrie Grimes and Joe Kenney.

PARAMETERS: Controlling the methods.

The user can specify the parameters used to describe the NLDR method. Not all the parameters control all the methods.

- Target Dimension d : The desired dimension of the embedding. In general, the target dimension d should be less than the input manifold dimension D . In the description of the examples above, it assumed that $d=2$. Note that only data with $d=1, 2$, or 3 will be graphed.
- Nearest Neighbors K : Specifies the number of nearest neighbors (KNN) used to build the graph for the following methods: ISOMAP, LLE, Hessian LLE, Laplacian, and LTSA. If K is too large or too small, the local geometry may not be interpreted correctly by the NLDR technique. The default is $K=8$, which seems to work well for the examples.
- Sigma: This specifies the width of the Gaussian kernel in the Diffusion Map method. The larger Sigma is, the more weight far-away points will exert on the weighted graph.
- Alpha: This parameter controls the normalization used by Diffusion Map.
 - Alpha = 0 is the Graph Laplacian
 - Alpha = $1/2$ is the Fokker-Plank propagator
 - Alpha = 1 is the Laplace-Beltrami operator

ALGORITHMS: Computing the embedding.

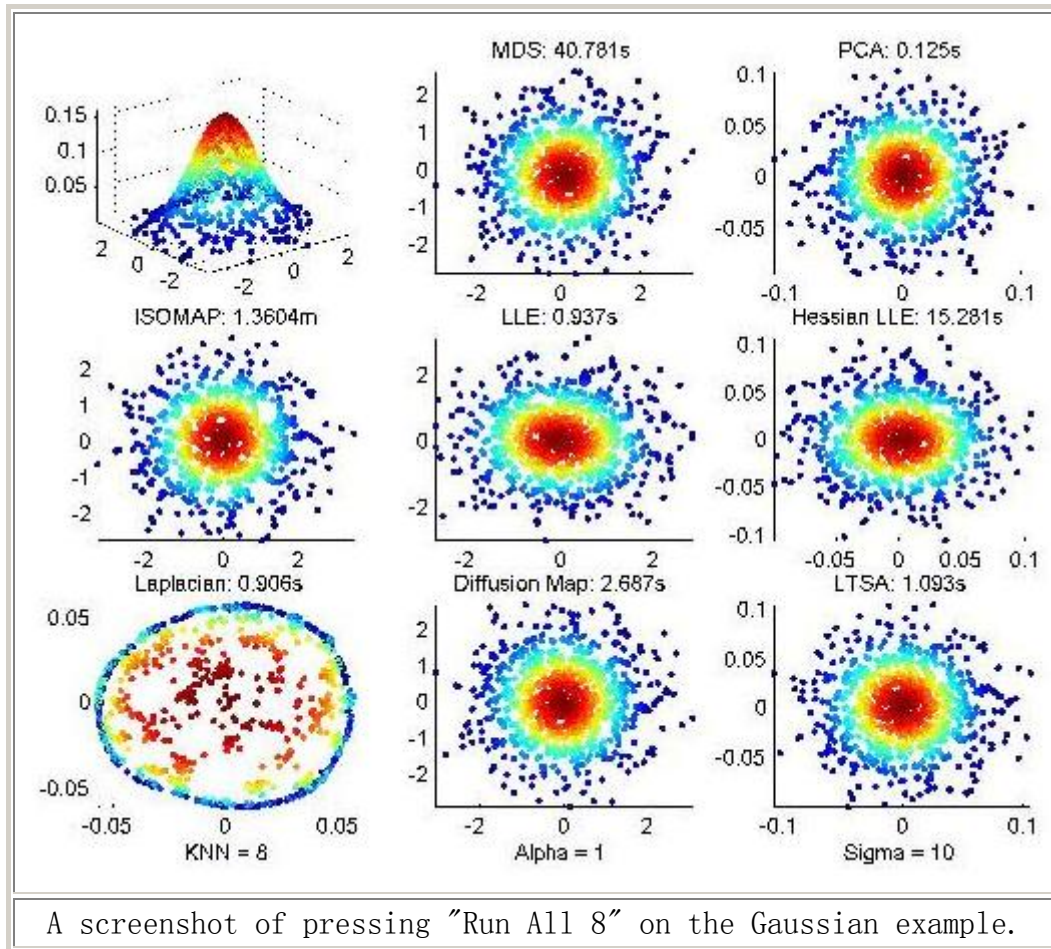
Simply press a button to run the NLDR method on the current input manifold. Some of the methods take a long time to run, particularly MDS and ISOMAP. If the data set is large or high-dimensional, the method may take several minutes to run. When the method is complete, the data matrix is written to the workspace variable "maniY" and the embedding is plotted in the lower left plot if it is 1D, 2D, or 3D. The black box in the lower left corner tells when the method is complete and how long it took to run.

Except for PCA and MDS, all the NLDR methods were written by the original authors of the method. I believe this is an important point, because if we are comparing the performance of the methods, then I can't be blamed for a poor implementation of the technique. The MDS routine was written by Michael Lee.

- MDS: This Multi-Dimensional Scaling (MDS) implementation is the "FastMDS" method that iterates to a solution under a fixed learning rate. MDS was written by Michael Lee.
- PCA: The classical Principal Components Analysis (PCA) uses Matlab's sparse eigen-solver "eigs" to quickly determine the principal vectors.
- ISOMAP: The ISOMAP method builds a graph from the K nearest neighbors and then performs MDS. This ISOMAP routine was written by Tenenbaum, de Silva, and Langford.
- LLE: The Locally Linear Embedding (LLE) technique builds a graph from the K nearest neighbors, computes weights assuming that neighborhoods are locally planar, and takes the top d eigenvectors. LLE was written by Sam Roweis & Lawrence Saul.
- Hessian LLE: The Hessian LLE method adapts the weights in LLE to minimize the Hessian operator. Like LLE, it requires careful setting of the parameter K. The main advantage of Hessian LLE is the only method designed for non-convex data sets, although the eigen-solver used seems to crash on some data sets. This implementation was written by David Donoho and Carrie Grimes.
- Laplacian: The Laplacian eigenmap method builds a graph from the K nearest neighbors and computes weights to minimize the norm of the gradient in the least squares sense. This method was written by Mikhail Belkin and

Partha Niyogi.

- Diffusion Map: The Diffusion Map uses a Gaussian kernel of width Sigma to construct a weighted graph and normalizes the operator with the parameter Alpha. This method was developed by R. Coifman and S. Lafon and the code was written by Stephane Lafon.
- LTSA: Local Tangent Space Alignment uses the K nearest neighbors to form a graph and approximate local tangent spaces for each neighborhood. This method seems to perform as well or better than the other methods on all examples. This code was sent to me by Zhenyue Zhang at Zheijiang University.
- Run All 8: A new figure will open which shows all 8 embeddings that result on the current input manifold. If you change figures while the methods are running, the plots may end up in strange places. It may take a very long time to run to completion. If any method crashes, MANI will simply skip over the method. The time to run each method is shown. But the output data is not available. If you wish to get the data, run the algorithms individually and track the variable "maniY".



LINKS

- [Junping Zhang's Manifold Learning Page](#)
A categorized and comprehensive list of research papers on manifold learning. Has a nice blurb about MANI too!
- [Manifold Learning Resource Page](#)
A list of pointers to papers and websites on manifold learning.
- MDS
[Michael Lee's MDS code](#)
- ISOMAP
J.B. Tenenbaum, V. de Silva and J. C. Langford. [A global geometric framework for nonlinear dimensionality reduction](#). *Science*, vol. 290, pp. 2319--2323, 2000.
- LLE
L. K. Saul and S. T. Roweis. [Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds](#). *Journal of Machine Learning Research*, v4, pp. 119-155, 2003.
- Hessian LLE
D. L. Donoho and C. Grimes. [Hessian Eigenmaps: new locally linear](#)

- [embedding techniques for high-dimensional data](#). Technical Report TR-2003-08, Department of Statistics, Stanford University, 2003.
- Laplacian Eigenmap
M. Belkin, P. Niyogi, [Laplacian Eigenmaps for Dimensionality Reduction and Data Representation](#), *Neural Computation*, June 2003; 15 (6):1373-1396.
 - Diffusion Maps
Nadler, Lafon, Coifman, and Kevrekidis. [Diffusion maps, spectral clustering and reaction coordinates of dynamical systems](#)
 - LTSA
Zhenyue Zhang and Hongyuan Zha. [Principal Manifolds and Nonlinear Dimension Reduction via Tangent Space Alignment](#). *SIAM Journal of Scientific Computing*, 2004, 26 (1): 313-338.

[Todd Wittman](#)

E-mail: wittman@math.umn.edu

Department of Mathematics

University of Minnesota