Maximum Variance Unfolding (MVU)

Raviteja Vemulapalli

May 2, 2013

Overview

- What is maximum variance unfolding?
- Results from the original paper
- Experiments on swiss roll data

MVU of Swiss roll

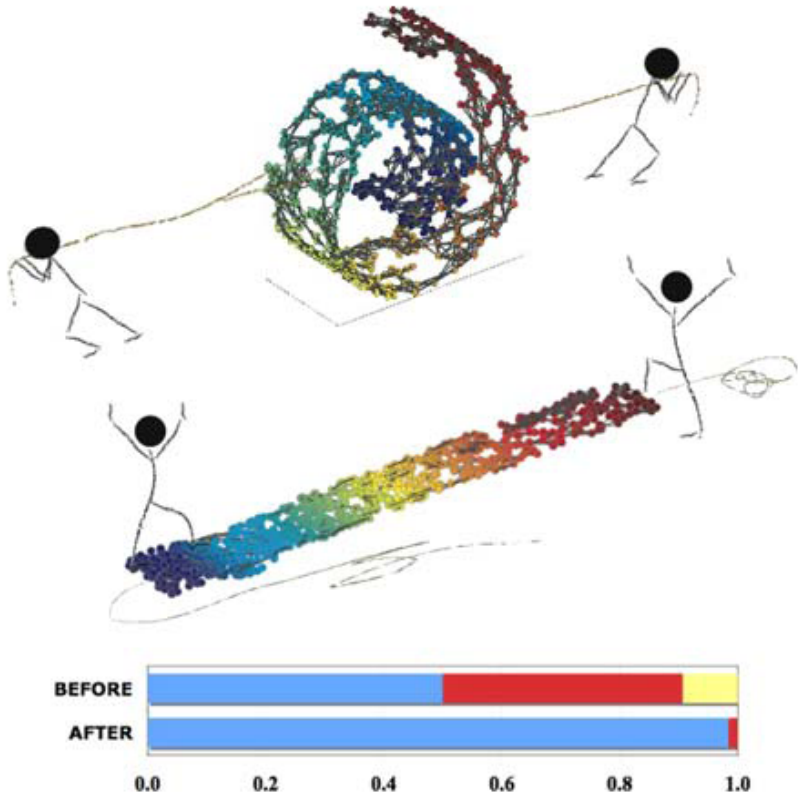


Figure: Maximum variance unfolding of swiss roll(figure taken from [1]).

MVU formulation

Input: $\{x_i \in \mathcal{R}^D\}_{i=1}^N$ Output: $\{y_i \in \mathcal{R}^d\}_{i=1}^N$

MVU formulation

Input: $\{x_i \in \mathcal{R}^D\}_{i=1}^N$ Output: $\{y_i \in \mathcal{R}^d\}_{i=1}^N$

Assumption: The manifold is isometric to a connected subset of Euclidean space.

MVU formulation

Input: $\{x_i \in \mathcal{R}^D\}_{i=1}^N$ Output: $\{y_i \in \mathcal{R}^d\}_{i=1}^N$

Assumption: The manifold is isometric to a connected subset of Euclidean space.

Isometry (informally) is a smooth invertible mapping that looks locally like a rotation plus translation.

MVU formulation

Input: $\{x_i \in \mathcal{R}^D\}_{i=1}^N$ Output: $\{y_i \in \mathcal{R}^d\}_{i=1}^N$

Assumption: The manifold is isometric to a connected subset of Euclidean space.

Isometry (informally) is a smooth invertible mapping that looks locally like a rotation plus translation.

Two data sets $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_N\}$ are said to be k-locally isometric if for every point x_i, there exists a rotation and translation that maps x_i and its k nearest neighbors $\{x_1^i, \ldots, x_k^i\}$ onto the point y_i and its neighbors $\{y_1^i, \ldots, y_k^i\}$.

MVU formulation: Constraints

Centering: Translational degree of freedom in the output can be removed by enforcing

$$\sum_{i=1}^{N} y_i = 0. \tag{1}$$

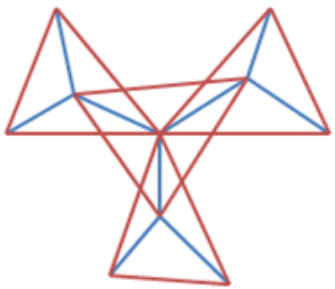
MVU formulation: Constraints

Centering: Translational degree of freedom in the output can be removed by enforcing

$$\sum_{i=1}^{N} y_i = 0. \tag{1}$$

Local isometry: The data sets $\{x_1, \ldots, x_N\}$ and $\{y_1, \ldots, y_N\}$ are locally isometric if whenever x_i and x_j are themselves neighbors or common neighbors of another point x_k in the data set, we have

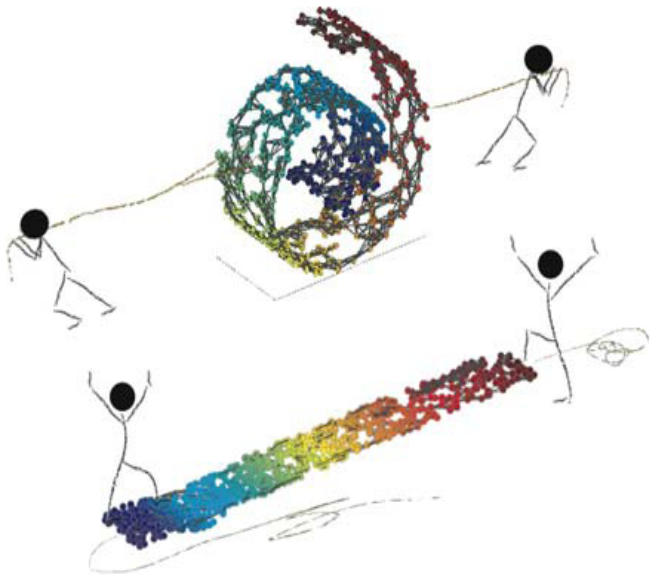
$$\|y_i - y_j\|_2^2 = \|x_i - x_j\|_2^2. \tag{2}$$

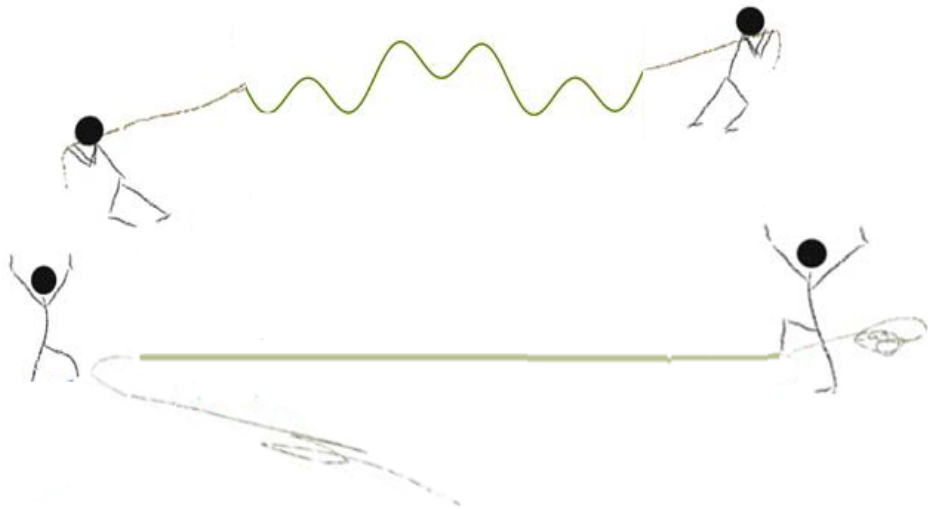


MVU formulation: Cost function

Pull the data samples as far apart as possible (subject to the centering and local isometry constraints) to unfold the manifold.

$$\text{maximize} \sum_{i,j=1}^{N} \|y_i - y_j\|_2^2 \tag{3}$$





MVU formulation: Optimization problem

$$\begin{aligned} &\text{maximize} \sum_{i,j=1}^{N} \|y_i - y_j\|_2^2 \\ &\text{subject to} \sum_{i=1}^{N} y_i = 0 \\ &\qquad \|y_i - y_j\|_2^2 = d_{ij} \text{ for all } (i,j) \text{ with } \eta_{ij} = 1. \end{aligned} \tag{4}$$

d_{ij} is the squared distance between x_i and x_j.

$\eta_{ij} = 1$ if x_i and x_j are themselves neighbors or common neighbors of another point x_k in the data set.

MVU formulation: Optimization problem

Let $K = [K_{ij}]$, where $K_{ij} = \langle y_i, y_j \rangle$. Then we have

$$d_{ij} = \|y_i - y_j\|_2^2 = K_{ii} + K_{jj} - 2K_{ij}$$

$$0 = \left\| \sum_{i=1}^{N} y_i \right\|_2^2 = \sum_{i,j=1}^{N} \langle y_i, y_j \rangle = \sum_{i,j=1}^{N} K_{ij}$$

$$\sum_{i,j=1}^{N} \|y_i - y_j\|_2^2 = \sum_{i,j=1}^{N} K_{ii} + K_{jj} - 2K_{ij} = 2Tr(K) = 2NVar\left(\{y_i\}_{i=1}^{N}\right) \tag{5}$$

MVU formulation: Optimization problem

$$\begin{aligned} &\text{maximize } Tr(K) \\ &\text{subject to } \sum_{i,j=1}^{N} K_{ij} = 0 \\ &\qquad K_{ii} + K_{jj} - 2K_{ij} = d_{ij} \text{ for all } (i,j) \text{ with } \eta_{ij} = 1 \\ &\qquad K \succeq 0 \end{aligned} \tag{6}$$

Note that the gram matrix K determines the outputs y_i uniquely up to rotation.

This optimization problem is a semi-definite program(SDP) and can be solved using standard SDP solvers.

MVU output: Kernel PCA

The final lower dimensional outputs y_i are obtained from the gram matrix K through kernel PCA.

Let $U\Lambda U^\top$ be the Eigen-decomposition of gram matrix K.
Then $[y_1, y_2, \ldots, y_N] = I_{d\times D}\ \Lambda^{1/2} U^\top$.

The dimensionality of the manifold is given by the number of non-zero Eigenvalues.

Maximum Variance Unfolding Algorithm

Assumption: The manifold is isometric to a connected subset of Euclidean space.

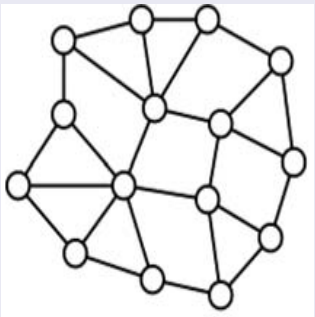
Maximum Variance Unfolding Algorithm

Assumption: The manifold is isometric to a connected subset of Euclidean space.

Algorithm:

- Form a graph that connects each point to its k neighbors.

Neighborhood graph



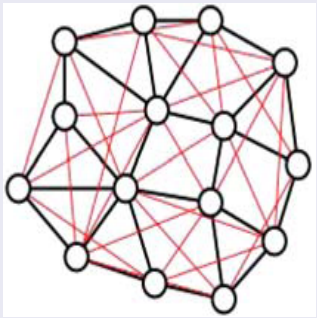
Maximum Variance Unfolding Algorithm

Assumption: The manifold is isometric to a connected subset of Euclidean space.

Algorithm:

- Form a graph that connects each point to its k neighbors.
- Add additional edges by connecting points that are common neighbors of another point in the data set.

Neighborhood graph



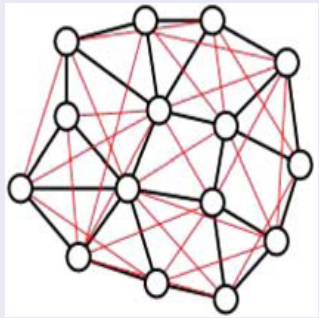
Maximum Variance Unfolding Algorithm

Assumption: The manifold is isometric to a connected subset of Euclidean space.

Algorithm:

- Form a graph that connects each point to its k neighbors.
- Add additional edges by connecting points that are common neighbors of another point in the data set.
- Compute the Gram matrix (centered on the origin) that corresponds to the maximum data variance and also preserves the distances between all connected points.

Neighborhood graph



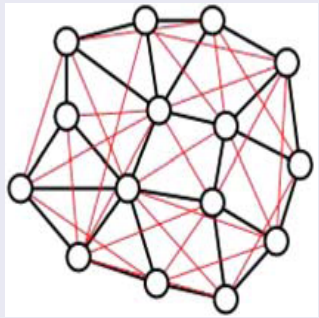
Maximum Variance Unfolding Algorithm

Assumption: The manifold is isometric to a connected subset of Euclidean space.

Algorithm:

- Form a graph that connects each point to its k neighbors.
- Add additional edges by connecting points that are common neighbors of another point in the data set.
- Compute the Gram matrix (centered on the origin) that corresponds to the maximum data variance and also preserves the distances between all connected points.
- Find the lower dimensional embedding using kernel PCA.

Neighborhood graph



MVU formulation: Relaxations

$$\begin{aligned} &\text{maximize } Tr(K) \\ &\text{subject to } \sum_{i,j=1}^{N} K_{ij} = 0,\ K \succeq 0, \\ &\qquad K_{ii} + K_{jj} - 2K_{ij} \leq d_{ij} \text{ for all } (i,j) \text{ with } \eta_{ij} = 1. \end{aligned} \tag{7}$$

$$\begin{aligned} &\text{maximize } Tr(K) - \gamma C(\zeta) \\ &\text{subject to } \sum_{i,j=1}^{N} K_{ij} = 0,\ K \succeq 0, \\ &\qquad K_{ii} + K_{jj} - 2K_{ij} = d_{ij} + \zeta_{ij} \text{ for all } (i,j) \text{ with } \eta_{ij} = 1, \end{aligned} \tag{8}$$

where $C(\zeta)$ is a penalty function.

Results from the paper: Swiss roll

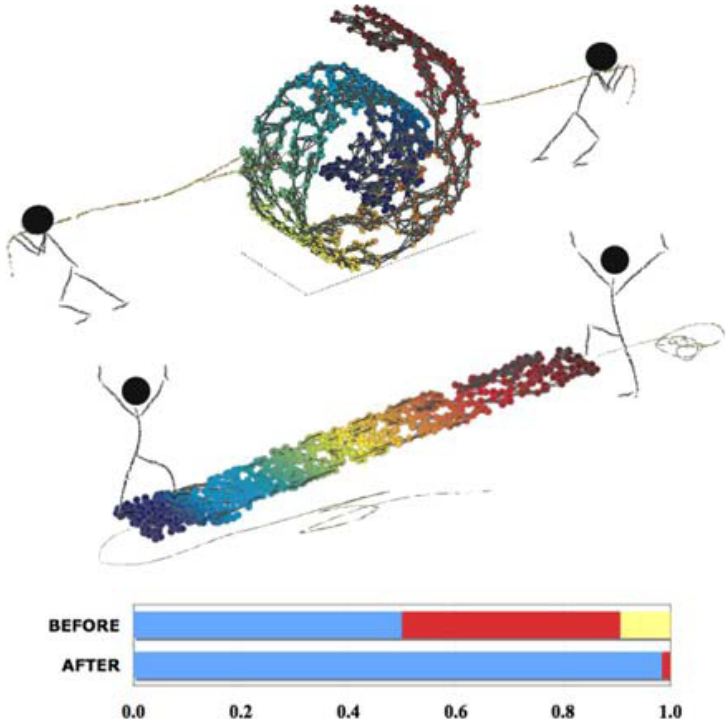


Figure: Maximum variance unfolding of swiss roll using 800 data points and 6 neighbors.

Results from the paper: Trefoil knot

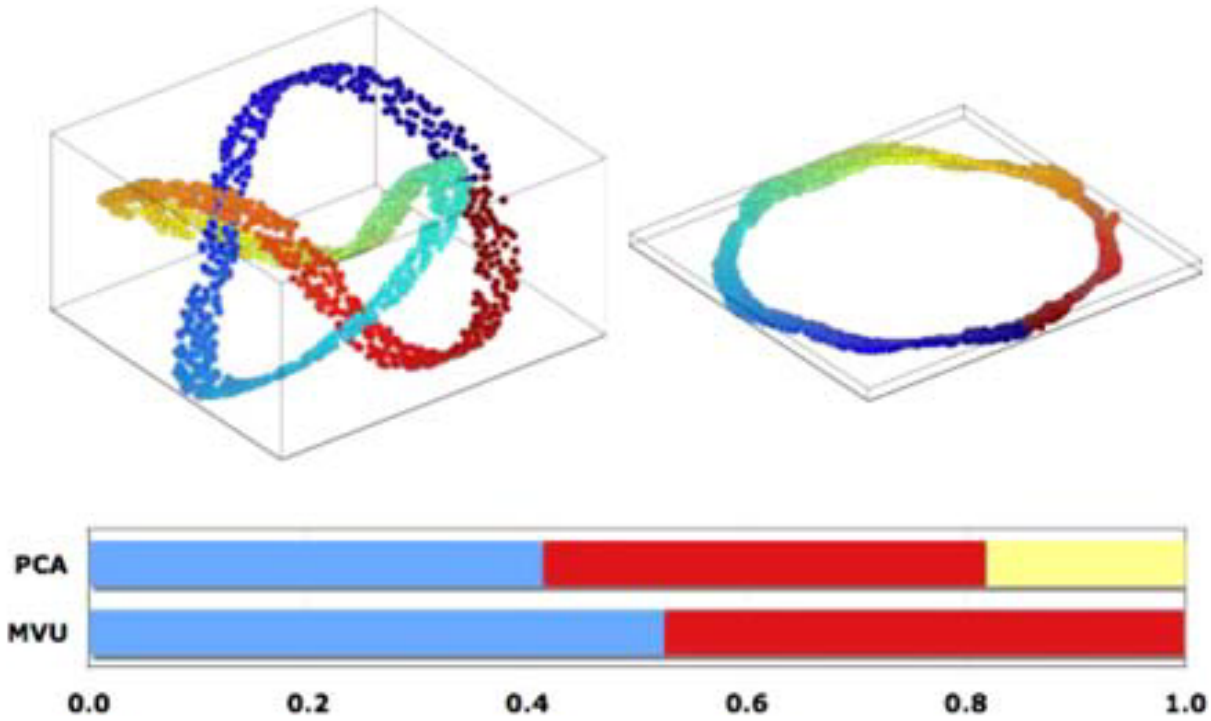


Figure: Maximum variance unfolding of trefoil knot using 1617 data points and 5 neighbors.

Results from the paper: Teapot 360 degrees

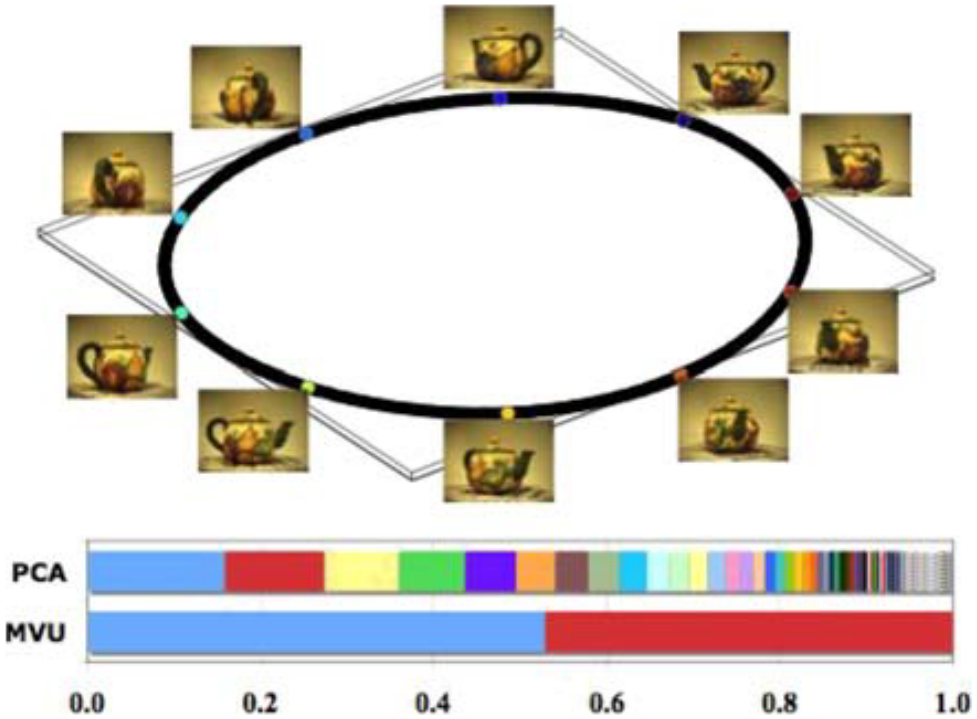


Figure: Maximum variance unfolding of images of a rotating teapot (360 degree rotation) using 400 images and 4 neighbors.

Results from the paper: Teapot 180 degrees

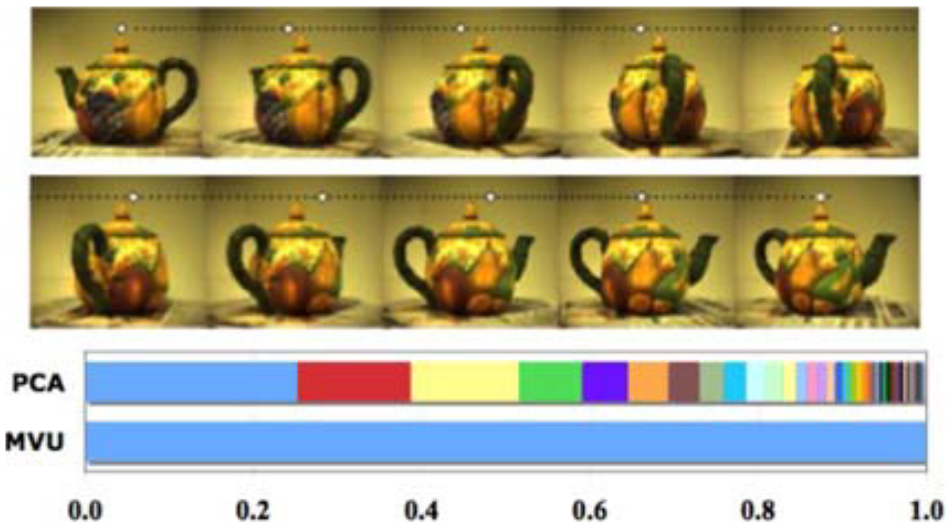


Figure: Maximum variance unfolding of images of a rotating teapot (180 degree rotation) using 200 images and 4 neighbors.

Factors that affect the performance of MVU

- Neighborhood: Number of nearest neighbors used
- Sampling: Number of data samples used
- Noise

MVU of swiss roll using 1000 points and 9 neighbors

Optimization: 500500 variables, 11939 distance constraints.

Computational time: 4.16 hours on a PC with 2.4 GHz processor and 12 GB RAM

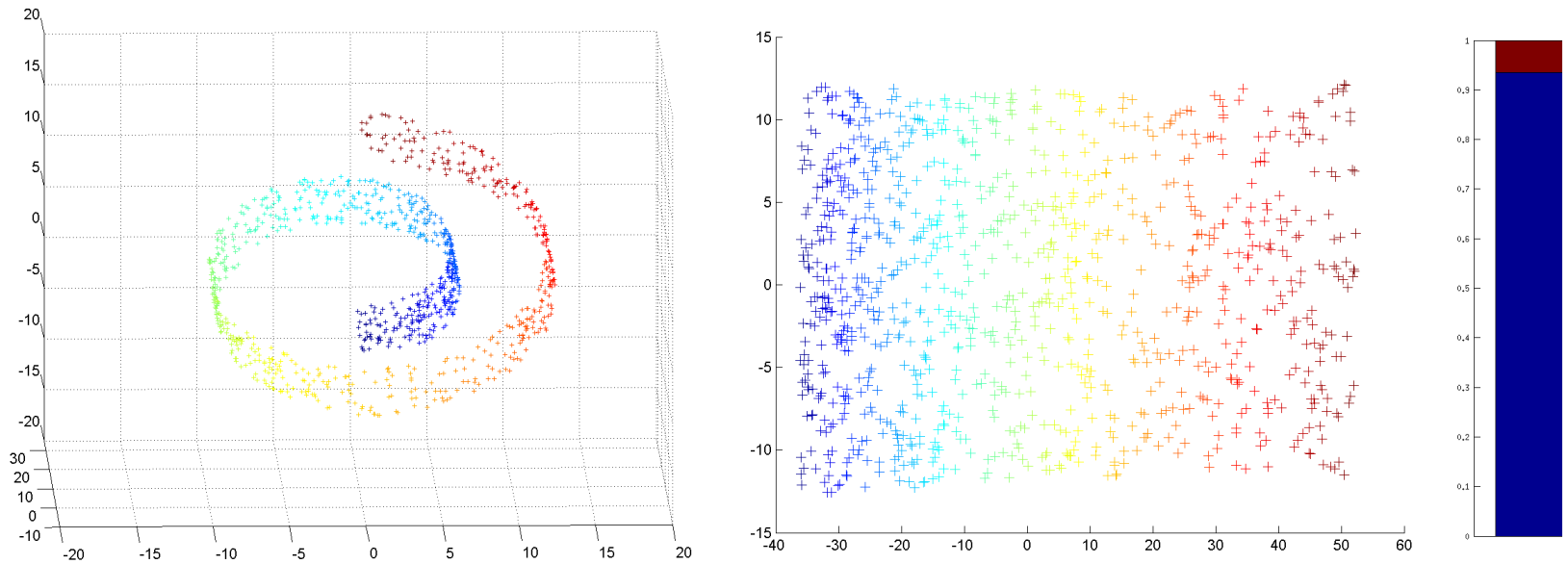
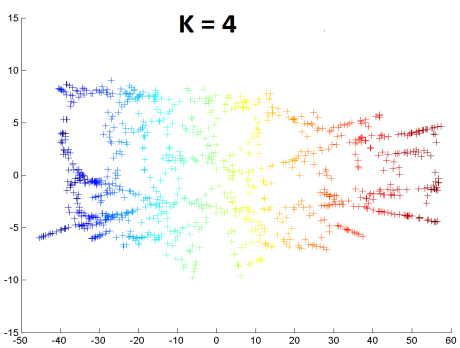
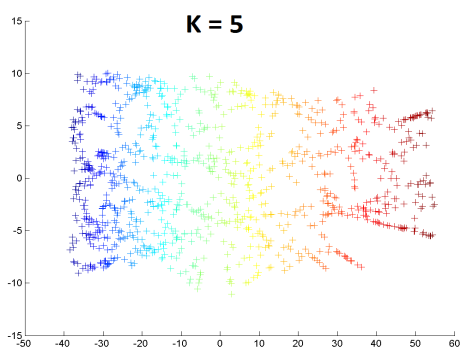


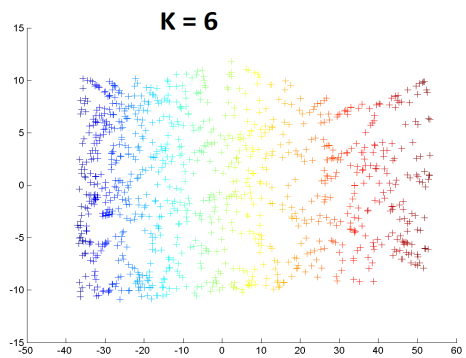
Figure: Left: Original swiss roll data - 1000 points without noise; Center: 2-d MVU embedding; Right: Eigenvalues of the MVU kernel matrix

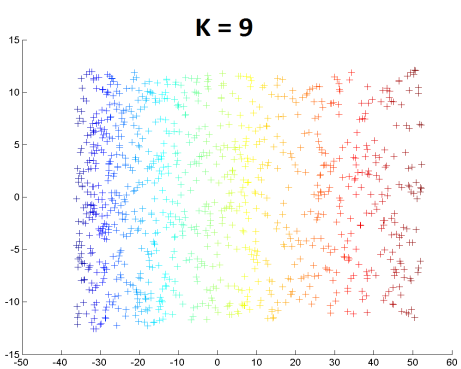
Swiss roll data: 1000 points without noise

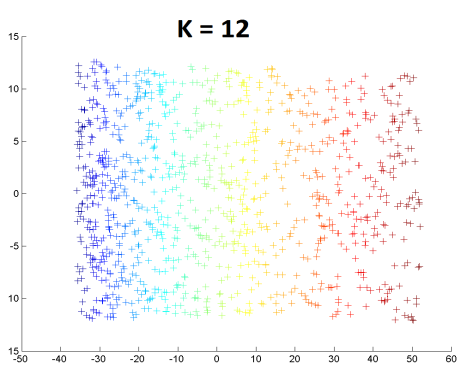
Neighbors	K = 4	K = 5	K = 6	K= 9	K = 12	K = 15
Eigen Spectrum	0.9757 0.0243	0.9638 0.0362	0.9517 0.0483	0.9353 0.0647	0.9342 0.0658	0.7463 0.1983 0.0554
Time (hrs)	0.482	0.636	1.452	4.162	11.016	22.445

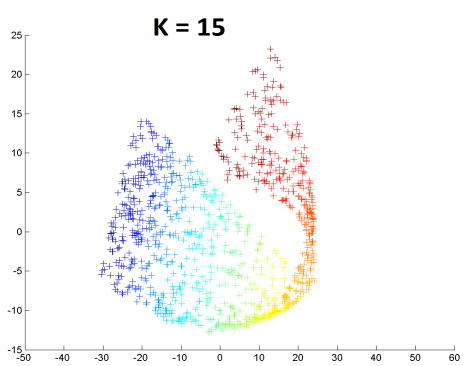






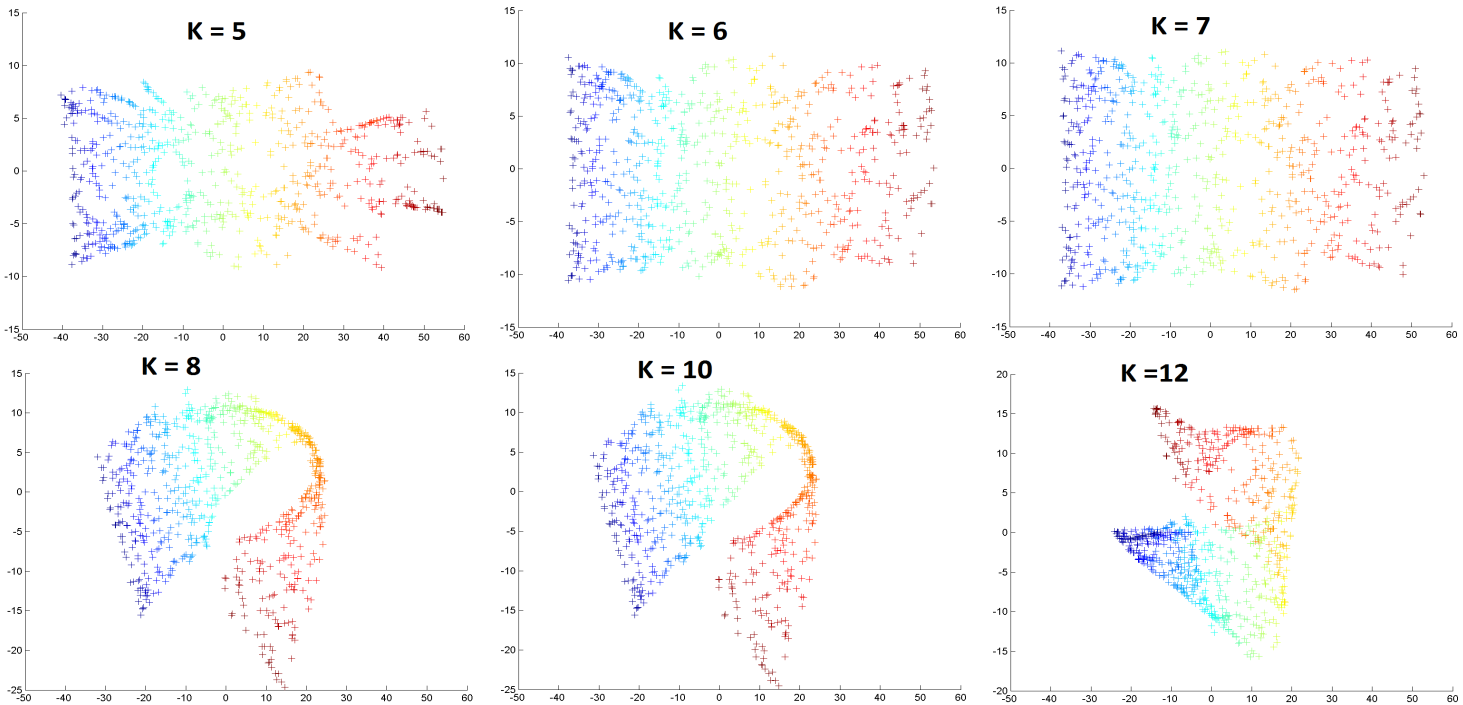






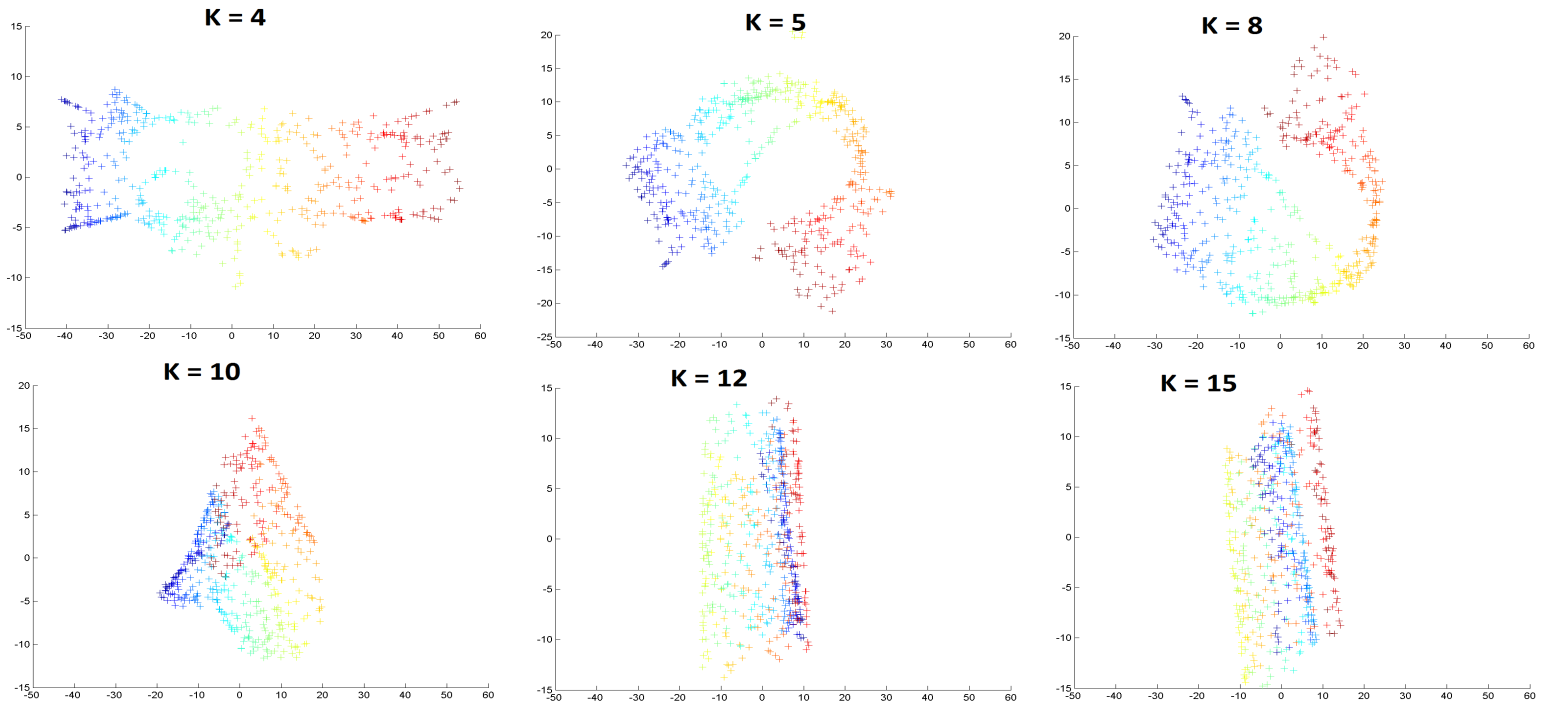
Swiss roll data: 750 points without noise

Neighbors	K = 5	K = 6	K = 7	K= 8	K = 10	K = 12
Eigen Spectrum	0.9695 0.0305	0.9491 0.0509	0.9423 0.0577	0.7487 0.1942 0.0572	0.7436 0.1991 0.0573	0.6234 0.2400 0.1366
Time (hrs)	0.284	0.469	0.538	1.45	3.268	3.822



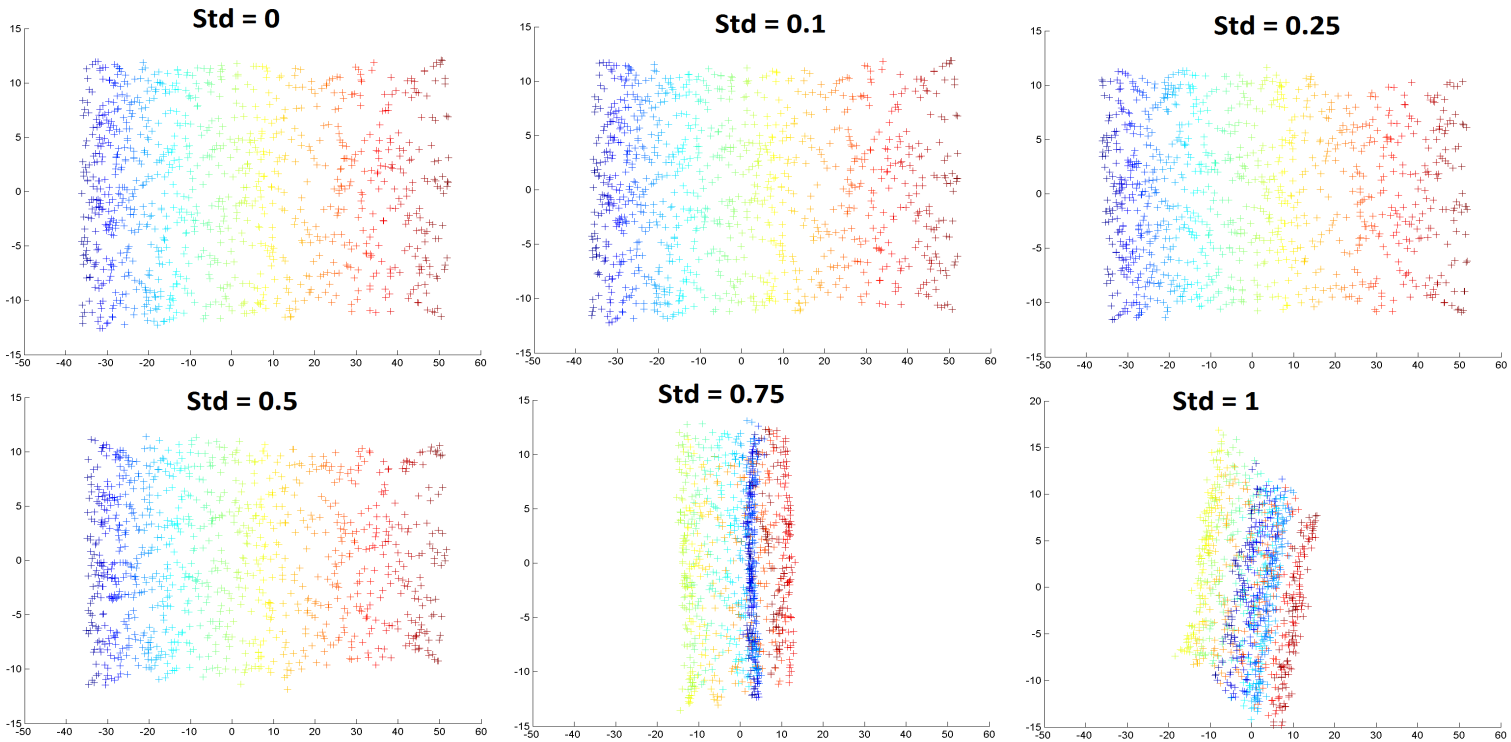
Swiss roll data: 500 points without noise

Neighbors	K = 4	K = 5	K = 8	K= 10	K = 12	K = 15
Eigen Spectrum	0.9751 0.0246 0.0003	0.7863 0.2070 0.0067	0.7610 0.1813 0.0577	0.4814 0.2119 0.1945 0.1122	0.3815 0.2891 0.2272 0.1021	0.3485 0.3376 0.2930 0.0210
Time (hrs)	0.088	0.108	0.482	0.594	1.52	3.126



Swiss roll data: 1000 points with noise and 9 neighbors

Noise	Std = 0	Std = 0.1	Std = 0.25	Std = 0.5	Std = 0.75	Std = 1
Eigen Spectrum	0.9353 0.0647	0.9391 0.0609	0.9428 0.0571 0.0001	0.9431 0.0568 0.0002	0.3736 0.3109 0.2694 0.0460	0.3562 0.3396 0.3013 0.0029



Conclusions from swiss roll experiments

- We need significant number of samples for MVU to work.
- MVU is sensitive to the number of neighbors. The sensitivity depends on the number of data samples.
- MVU is robust to small amounts of noise.

References

[1] Kilian Q. Weinberger and Lawrence K. Saul, “Unsupervised Learning of Image Manifolds by Semidefinite Programming ”, *International Journal of Computer Vision*, 70(1), 77-90, 2006.