Dimensionality Reduction: A comparative Review[1]

Ying Fu

School of Business Administration
South China University of Technology

Sept 11,2019

Outline

1. Introduction
2. Nonlinear Techniques for dimensionality reduction
   - Preserve global properties
   - Preserve local properties
   - Perform global alignment of a mixture of linear models
3. Characterization of the techniques
   - Relations
   - General properties
   - Out-of-sample extension
4. Experiments and discussion
   - Experimental setup
   - Experiments on datasets
   - Discussion

Aims of this paper

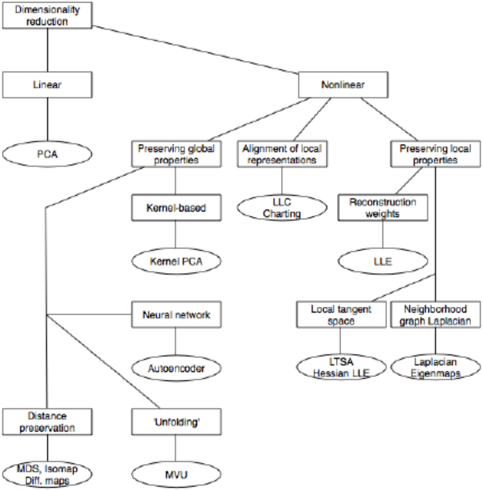
1. To investigate to **what extent** novel nonlinear dimensionality reduction techniques outperform the traditional PCA.
   - Both a theoretical and an empirical evaluation
2. To identify the inherent **weakness** of the twelve nonlinear techniques for dimensionality reduction.
   - A careful analysis of the empirical results on **specifically designed** artifical datasets and on real-world datasets.

Formal definition of dimensionality reduction

definition

- Assume we have a dataset represented in a $n \times D$ matrix $\mathbf{X}$ consisting of n datavectors $x_i (i \in 1, 2, ..., n)$ with dimensionality D.
- Assume further that the dataset has intrinsic dimensionality d (where $d < D$, and often $d \ll D$)
- Dimensionality reduction transform dataset $\mathbf{X}$ with dimensionality D into a new dataset $\mathbf{Y}$ with dimensionality d, while preserve the geometry of the data as much as possible.

Taxonomy of techniques



Outline

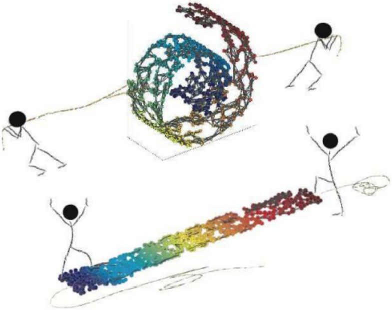
1. Introduction
2. Nonlinear Techniques for dimensionality reduction
   - Preserve global properties
   - Preserve local properties
   - Perform global alignment of a mixture of linear models
3. Characterization of the techniques
   - Relations
   - General properties
   - Out-of-sample extension
4. Experiments and discussion
   - Experimental setup
   - Experiments on datasets
   - Discussion

Global techniques

1. **MDS**: Preserve the Euclidean distance.
2. **Isomap**: Preserve pairwise geodesic distance
3. **MVU**: Unfolding
4. **Kernel PCA**: Kernel based
5. **Diffusion maps**: Preserve the diffusion distance
6. **Multilayer autoencoders**: Neural network

Maximum Variance Unfolding:MVU(Ying Fu)

Unfold neighbourhood graph while preserving local structure



Intuitive explanation:

Imagine the inputs as a swiss roll that is coiled up in three dimensions. By pulling the swiss roll taut, the roll is arranged in a line.

Maximum Variance Unfolding(MVU)

objective

- Maximum the sum of the squared Euclidean distance between all datapoints, under the constraint that the distance inside the neighborhood graph G are preserved.
- Maximize $\sum_{ij} \|y_i - y_j\|^2$ **with subject to:** $\|y_i - y_j\|^2 = \|x_i - x_j\|^2$ for $\forall (i,j) \in G$
- **MVU reduces to MDS if G contains all pairs of points**(Ying Fu)

MVU reformulates the optimization problem as a SDP

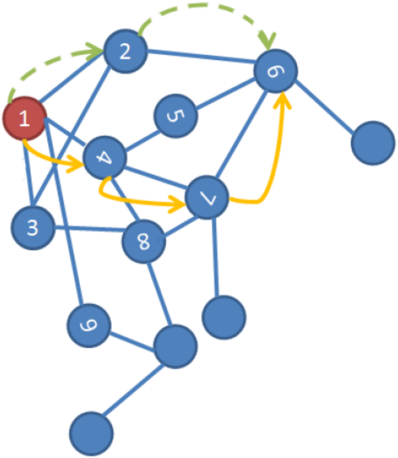
Defining a matrix **K** that is the inner product of the low-dimensional data representation **Y**.

Maximize trace(K) subject to:

1. $k_{ii} - 2k_{ij} + k_{jj} = \|x_i - x_j\|^2$ for $\forall (i,j) \in G$ (locally metric)
2. $\sum_{ij} k_{ij} = 0$ (centered)
3. $\mathbf{K} \geq 0$ (positive definite)

This is a **semi-definite program**:convex optimization with unique solution. From the solution K of the SDP, the low-dimensional data representation Y can be obtained by performing a singular value decomposition.

Diffusion maps[2](De la Porte)



A random walk on a dataset

- Each "jump" has a probability associated with it.
- The dashed path between nodes 1 and 6 requires two jumps (i.e., two time units) with the probability along the path being p(node 1; node 2) and p(node 2; node 6)

Diffusion maps:Constructing a graph of the data

- Using the Gaussian kernel function to compute the weights of the edges in the graph.

$$w_{ij} = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

- Normalize the matrix **W**:

$$p_{ij} = \frac{w_{ij}}{\sum_k w_{ik}}$$

- Matrix **P** represents the probability of a transition from one datapoint to another datapoint in a single timestep.
- The forward probability matrix for t timesteps $\mathbf{P^t}$

Diffusion maps: Preserve the diffusion distance

Using the random walk forward probabilities p_{ij}^t, the diffusion distance is defined by:

$$D^{(t)}(x_i, x_j) = \sqrt{\sum_k \frac{(p_{ik}^t - p_{jk}^t)^2}{\psi(x_k)}}$$

where:

$$\psi(x_i) = \frac{\sum_j p_{ij}}{\sum_k \sum_j p_{kj}}$$

Note: $\psi(x_i)$ attributes more weight to part of the graph with high density.

Intuitive explanation of diffusion distance(Ying Fu)

- Given a datapoint x_i in a graph of the data , we let it diffuse for a period of time t.
- Given another datapoint x_j, we also let it diffuse for a period time t.
- At the end, we look at the difference between the two distributions. And that is our Diffusion Distance.

Local techniques

1. **LLE: Reconstruction weights**
2. **Laplacian Eigenmaps**
3. **Hessian LLE**: Preserve the local tangent space
4. **Local Tangent Space Alignment**: Preserve the local tangent space.

Laplacian Eigenmaps

The distance in the low-dimensional data representation between a datapoint and its first nearest neighbor contributes more to the cost function.

The cost function is minimized:

$$\phi(Y) = \sum_{ij} (y_i - y_j)^2 w_{ij}$$

where:

$$w_{ij} = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

Intuitive explations::Nearby points in the highdimensional space are brought closer together in the low-dimensional representation.

Laplacian Eigenmaps: Formulating as an eigenproblem[3]

W: Graph matrix
D: Degree matrix (a diagonal matrix of which the entries are the row sums of W): $d_{ii} = \sum_j w_{ij}$
Graph Laplacian **L**: $\mathbf{L} = \mathbf{D} - \mathbf{W}$
Cost function is minimized:

$$\phi(Y) = \sum_{ij} (y_i - y_j)^2 w_{ij} = \mathbf{2Y^T LY}$$

With the subject to:

- $\mathbf{Y^T DY} = \mathbf{1}$ (Removing an arbitrary scaling factor in the embedding)
- $\mathbf{Y^T D1} = \mathbf{0}$ (Cause $\mathbf{1}$ is an eigenvector with eigenvalue $\mathbf{0}$)

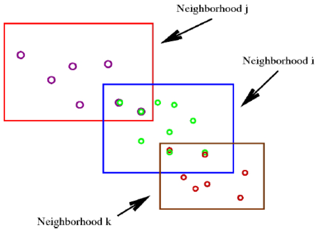
Find d smallest nonzero eigenvalues.

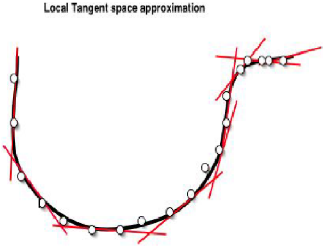
Hessian LLE(TODO)

- Minimizes the curviness of the high-dimensional space.
- Assuming that the low-dimensional data representation is locally isometric.
- Done by an **eigenanalysis** of a matrix $\mathcal{H}$ that describes the curviness of the manifold.
- **Find d smallest nonzero eigenvalues**

Local tangent space alignment:LTSA(TODO)

- Assuming the manifold is local linearity.
- Similar to HLLE, with the only difference of describe local properties of the high-dimensional data using the **local tangent space**.
- LTSA align these linear mappings.
- **Find d smallest nonzero eigenvalues**





Global alignment of linear models(TODO)

1. **Locally Linear Coordination(LLC)**: Alignment of local representations(Find d smallest nonzero eigenvalues).
2. **Manifold charting**: Alignment of local representations(Find d smallest nonzero eigenvalues).

Outline

1 Introduction

2 Nonlinear Techniques for dimensionality reduction
- Preserve global properties
- Preserve local properties
- Perform global alignment of a mixture of linear models

3 Characterization of the techniques
- Relations
- General properties
- Out-of-sample extension

4 Experiments and discussion
- Experimental setup
- Experiments on datasets
- Discussion

Interrelated techniques

1. Kernel PCA with **a linear kernel** is identical to performing traditional PCA.
2. Autoencoders with **a linear activation functions** is identical to performing traditional PCA.
3. MDS with **Euclidean distance** is identical to PCA.
4. Perfroming MDS using geodesic distance is identical to performing Isomap.
5. Isomap with the number of nearest neighbors k set to n-1 is identical to traditional MDS, as well as to PCA.

Interrelated techniques

1. Isomap retains **pure geodesic distance**, while diffusion maps retain a **weighted sum of distance of all paths through a graph**.
2. Diffusion maps in which $\mathbf{t} = \mathbf{1}$ are fairly similar to Kernel PCA with the **Gaussian kernel function.**
3. Isomap,LLE,Laplacian Eigenmaps can be considered as special cases of Kernel PCA(using a specific kernel function)
4. MVU can be viewed upon as a special case of Kernel PCA in which the SDP is the kernel function.

What not included in empirical comparative evaluation

1. **MDS**(equivalent to PCA)
2. **Kernel PCA using a linear kernel** (equivalent to PCA)
3. **Autoencoders using linear activation functions**(equivalent to PCA)
4. **Kernel PCA using a Gaussian kernel**(resemble to diffusion maps)
5. **Kernel PCA using a polynomial kernel, instead**

General properties

Technique	*Convex*	*Parameters*	*Computational*	*Memory*
PCA	yes	none	$O(D^3)$	$O(D^2)$
MDS	yes	none	$O(n^3)$	$O(n^2)$
Isomap	yes	k	$O(n^3)$	$O(n^2)$
MVU	yes	k	$O((nk)^3)$	$O((nk)^3)$
Kernel PCA	yes	$\kappa(\cdot,\cdot)$	$O(n^3)$	$O(n^2)$
Diffusion maps	yes	σ, t	$O(n^3)$	$O(n^2)$
Autoencoders	no	net size	$O(inw)$	$O(w)$
LLE	yes	k	$O(pn^2)$	$O(pn^2)$
Laplacian Eigenmaps	yes	k, σ	$O(pn^2)$	$O(pn^2)$
Hessian LLE	yes	k	$O(pn^2)$	$O(pn^2)$
LTSA	yes	k	$O(pn^2)$	$O(pn^2)$
LLC	no	m, k	$O(imd^3)$	$O(nmd)$
Manifold charting	no	m	$O(imd^3)$	$O(nmd)$

Observations:

- Some **nonlinear techniques** for dimensionality reduction may suffer from **getting stuck in local optima** (e.g. Autoencoders, LLC and manifold charting).
- All nonlinear techniques requires **free parameters**.
- A number of **nonlinear techniques** have **computational disadvantages** and may suffer from a **memory complexity** compared to PCA.

Out-of-sample extension:embedding of new datapoints

- **PCA**:Multiplying the new datapoint with the linear mapping matrix **M**, the same with Kernel PCA.
- **Autoencoders**:The trained network defines the transformation.
- **For a number of nonlinear techniques**: Using an estimation technique.(e.g.Isomap,LLE,LE are using Nystrom approximation which approximates the eigenvectors)

Outline

1. Introduction
2. Nonlinear Techniques for dimensionality reduction
   - Preserve global properties
   - Preserve local properties
   - Perform global alignment of a mixture of linear models
3. Characterization of the techniques
   - Relations
   - General properties
   - Out-of-sample extension
4. Experiments and discussion
   - Experimental setup
   - Experiments on datasets
   - Discussion

Preparations

Datasets

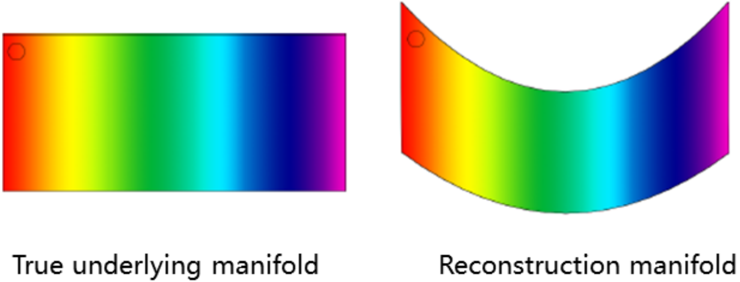
- Artifical datasets
- Natural datasets

Evaluation

- To evaluate the local structure of the data
- **Generalization errors** of k-**neighbor** classifiers trained on low-dimensional data representation instead of **reconstruction error**.
  1. Reconstruction errors measure global structure.
  2. Reconstruction errors cannot be computed on real-world datasets.

Reconstruction errors measure global structure

Although high reconstruction error, the local structure of the two manifolds is nearly identical (as the circles indicate).



Preparations:parameter settings

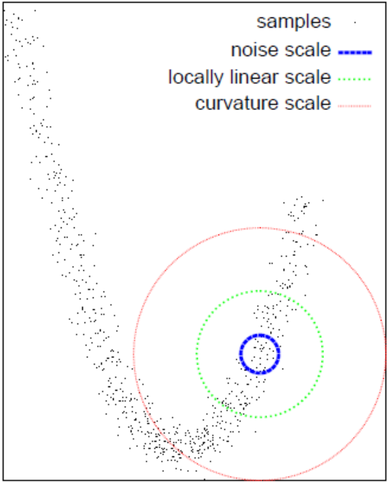
Technique	*Parameter settings*
PCA	None
Isomap	$5 \le k \le 15$
MVU	$5 \le k \le 15$
Kernel PCA	$\kappa = (XX^T + 1)^5$
Diffusion maps	$10 \le t \le 100 \quad \sigma = 1$
Autoencoders	Three hidden layers; sigmoid
LLE	$5 \le k \le 15$
Laplacian Eigenmaps	$5 \le k \le 15 \quad \sigma = 1$
Hessian LLE	$5 \le k \le 15$
LTSA	$5 \le k \le 15$
LLC	$5 \le k \le 15 \quad 5 \le m \le 25$
Manifold charting	$5 \le m \le 25$

Other settings:

- **Grid search** to find best parameters.
- σ is fixed to **1** to restrict computional requirements.
- **k** in the *knn* was set to **1**
- **Maximum likelihood intrinsic dimenisonality estimator** to determine target dimensionality.
- **Leave-one-out validation** to obtain results of experiments.

Estimate intrinsic dimensionality[4](Matthew Brand)

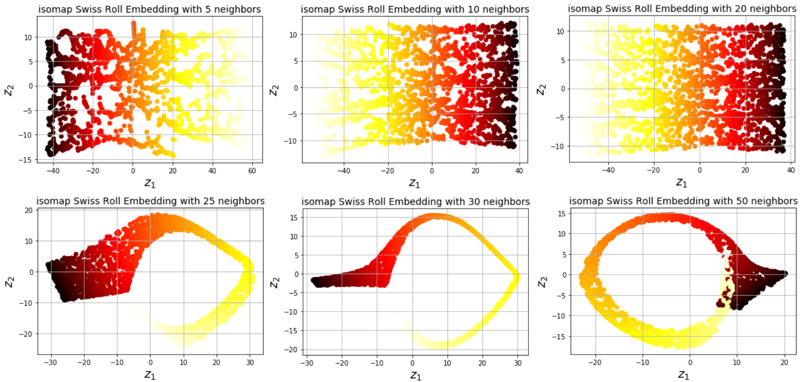
Scale behavior of a 1D manifold in 2-space



Density Estimation:

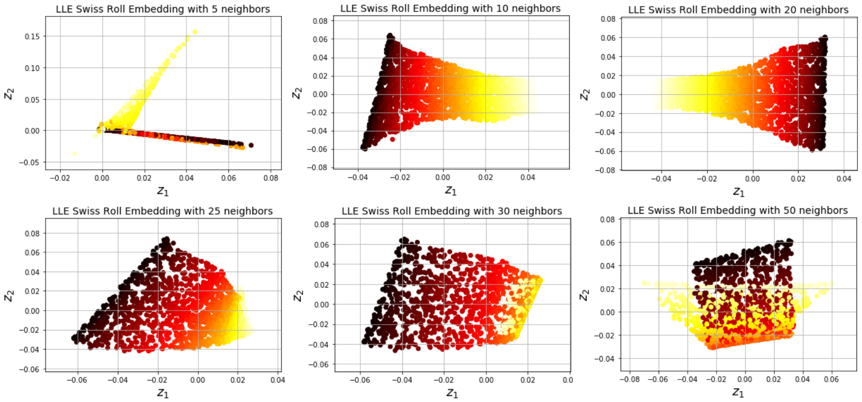
- Considering a boll of radius $\mathbf{r}$ centered on a data point and containing $\mathbf{n}(\mathbf{r})$ data points.
- Defining $\mathbf{c}(\mathbf{r}) = \frac{\log \mathbf{r}}{\log \mathbf{n}(\mathbf{r})}$
- At noise scale,$\mathbf{c}(\mathbf{r}) = \frac{\mathbf{1}}{\mathbf{D}} < \frac{\mathbf{1}}{\mathbf{d}}$
- At locally linear scale, $\mathbf{n}(\mathbf{r}) \propto \mathbf{r}^{\mathbf{d}}$, $\mathbf{c}(\mathbf{r}) = \frac{\mathbf{1}}{\mathbf{d}}$
- $\mathbf{c}(\mathbf{r}) < \frac{\mathbf{1}}{\mathbf{d}}$ whlie Curvaturing at large scales
- The maximum of $\mathbf{c}(\mathbf{r}) = \frac{\mathbf{1}}{\mathbf{d}}$ gives an estimation.

Choose k neighbors:Isomap(Ying Fu)



- If k is too small, it may suffer form “holes“
- If k is too large, short-circuiting may occurs.

Choose k neighbors:LLE(Ying Fu)



- If k is too large, it may suffer from folding.

Artificial datasets

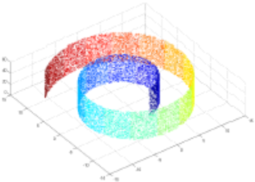
Requirements

1. Data that lies on or near a low-dimensional manifold that **is or is not isometric to Euclidean space**.
2. Data that lies on or near an **discontinuous manifold**.
3. A manifold with a **high intrinsic dimensionality**.

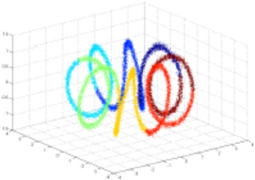
Datasets

1. Swiss roll dataset(1)
2. Helix dataset(1)
3. Twin peaks dataset(1)
4. Broken Swiss roll dataset(2)
5. High-dimensinal(HD) dataset(3)

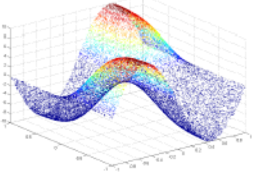
Artificial datasets



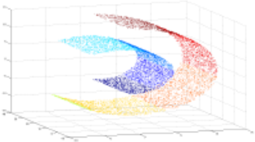
(a) Swiss roll dataset.



(b) Helix dataset.



(c) Twinpeaks dataset.



(d) Broken Swiss roll dataset.

Natural datasets

Datasets representing tasks from a varity of domains

1. MNIST dataset
   - Consisting of 60,000 handwritten digits, only **2500** is randomly selected in our experiments.
   - Each image have **28** $\times$ **28** pixels, considered as **784** dimension.
2. COIL20 dataset
3. NiSIS dataset
4. ORL dataset
   - A face recognition dataset containing **400** graysacle images of **112** $\times$ **92** pixels that depict **40** faces under various conditions.
5. HIVA dataset
   - A **drug discovery** dataset with two classed.
   - Consisting of **3845** datapoints with dimenisonality **1617**.

Experiments on artificial datasets:Observations

Dataset (d)	*None*	*PCA*	*Isomap*	*MVU*	*KPCA*	*DM*	*Autoenc.*	*LLE*	*LEM*	*HLLE*	*LTSA*	*LLC*	*MC*
Swiss roll (2D)	3.68%	30.56%	3.28%	5.12%	29.30%	28.06%	30.58%	7.44%	10.16%	3.10%	**3.06%**	27.74%	42.74%
Helix (1D)	1.24%	38.56%	**1.22%**	3.76%	44.54%	36.18%	32.50%	20.38%	10.34%	failed	1.68%	26.68%	28.16%
Twinpeaks (2D)	0.40%	0.18%	0.30%	0.58%	0.08%	**0.06%**	0.12%	0.54%	0.52%	0.10%	0.36%	12.96%	**0.06%**
Broken Swiss (2D)	2.14%	27.62%	14.24%	36.28%	27.06%	23.92%	26.32%	37.06%	26.08%	**4.78%**	16.30%	26.96%	23.92%
HD (5D)	24.19%	22.14%	20.45%	23.62%	29.25%	34.75%	**16.29%**	35.81%	41.70%	47.97%	40.22%	38.69%	31.46%

1. Nonlinear techniques employing **neighborhood graph**(Isomap,LLE,LE,MVU,LTSA,LLC) outperform other techniques on **standard manifold learning**(e.g.swiss roll).
2. Local nonlinear dimensionality reduction(LLE,HLLE) perform less well on **manifolds that are not isometric to Euclidean space**.(e.g. Helix)
3. Most nonlinear techniques cannot deal with **discontinuous manifold**.(e.g. broken swiss roll,except HLLE)

Experiments on artificial datasets:Observations(cont.)

Dataset (d)	*None*	*PCA*	*Isomap*	*MVU*	*KPCA*	*DM*	*Autoenc.*	*LLE*	*LEM*	*HLLE*	*LTSA*	*LLC*	*MC*
Swiss roll (2D)	3.68%	30.56%	3.28%	5.12%	29.30%	28.06%	30.58%	7.44%	10.16%	3.10%	**3.06%**	27.74%	42.74%
Helix (1D)	1.24%	38.56%	**1.22%**	3.76%	44.54%	36.18%	32.50%	20.38%	10.34%	failed	1.68%	26.68%	28.16%
Twinpeaks (2D)	0.40%	0.18%	0.30%	0.58%	0.08%	**0.06%**	0.12%	0.54%	0.52%	0.10%	0.36%	12.96%	**0.06%**
Broken Swiss (2D)	2.14%	27.62%	14.24%	36.28%	27.06%	23.92%	26.32%	37.06%	26.08%	**4.78%**	16.30%	26.96%	23.92%
HD (5D)	24.19%	22.14%	20.45%	23.62%	29.25%	34.75%	**16.29%**	35.81%	41.70%	47.97%	40.22%	38.69%	31.46%

1. Most nonlinear techniques perform **poorly** on dataset with **high intrinsic dimensionality**.(e.g. HD dataset)
2. Hessian LLE **fails** to find a solution on the helix dataset. The failure is the result of the inability of the eigensolver to solve the eigenproblem up to sufficient precision.

Experiments on natural datasets

Dataset (d)	*None*	*PCA*	*Isomap*	*MVU*	*KPCA*	*DM*	*Autoenc.*	*LLE*	*LEM*	*HLLE*	*LTSA*	*LLC*	*MC*
MNIST (20D)	5.11%	**5.06%**	28.54%	18.35%	65.48%	59.79%	14.10%	19.21%	19.45%	89.55%	32.52%	36.29%	38.22%
COIL20 (5D)	0.14%	3.82%	14.86%	21.88%	7.78%	4.51%	**1.39%**	9.86%	14.79%	43.40%	12.36%	6.74%	18.61%
ORL (8D)	2.50%	**4.75%**	44.20%	39.50%	5.50%	49.00%	69.00%	9.00%	12.50%	56.00%	12.75%	50.00%	62.25%
NiSIS (15D)	8.24%	**8.73%**	20.57%	19.40%	11.70%	22.94%	9.82%	28.71%	43.08%	45.00%	failed	26.86%	20.41%
HIVA (15D)	4.63%	5.05%	4.97%	4.89%	5.07%	3.51%	4.84%	5.23%	5.23%	failed	6.09%	**3.43%**	5.20%

1. **PCA, Kernel PCA and autoencoders** perform strongly on almost all datasets.
2. The **failures of Hessian LLE and LTSA** are the result of the inability of the eigensolver to **identify eigenvalues up to a sufficient precision**.
3. The classification performance of our classifiers was not improved by performing dimensionality reduction.

Experiments on artificial datasets(Ying Fu)

Results given by the author:

Dataset (d)	*None*	*PCA*	*Isomap*	*MVU*	*KPCA*	*DM*	*Autoenc.*	*LLE*	*LEM*	*HLLE*	*LTSA*	*LLC*	*MC*
Swiss roll (2D)	3.68%	30.56%	3.28%	5.12%	29.30%	28.06%	30.58%	7.44%	10.16%	3.10%	**3.06%**	27.74%	42.74%
Helix (1D)	1.24%	38.56%	**1.22%**	3.76%	44.54%	36.18%	32.50%	20.38%	10.34%	failed	1.68%	26.68%	28.16%
Twinpeaks (2D)	0.40%	0.18%	0.30%	0.58%	0.08%	**0.06%**	0.12%	0.54%	0.52%	0.10%	0.36%	12.96%	**0.06%**
Broken Swiss (2D)	2.14%	27.62%	14.24%	36.28%	27.06%	23.92%	26.32%	37.06%	26.08%	**4.78%**	16.30%	26.96%	23.92%
HD (5D)	24.19%	22.14%	20.45%	23.62%	29.25%	34.75%	**16.29%**	35.81%	41.70%	47.97%	40.22%	38.69%	31.46%

Repeated results:

Dataset	None	PCA	Isomap	MVU	KPCA	DM	Autoenc.	LLE	LEM	HLLE	LTSA	LLC	MC
Swiss roll	5.55%	30.50%	5.45%	0.00%	0.00%	30.60%	48.20%	31.10%	25.55%	6.10%	6.15%	21.80%	16.15 %
Helix	1.70%	3.00%	1.45%	0.00%	0.00%	3.00%	2.80%	20.55%	1.40%	1.25%	1.40%	3.80%	4.00%
Twinpeaks	0.60%	0.40%	0.20%	0.00%	0.00%	0.15%	0.30%	2.65%	0.55%	0.35%	0.20%	0.85%	0.50%

- Done by the **drtoolbox** provided by **van der Maaten**
- The **broken swiss** and **HD datasets** are not provided.
- All the parameters are **default** in the drtoolbox.
- Generalization errors of **1-NN classifiers**.

Experiments on artificial datasets(Ying Fu)

Results given by the author:

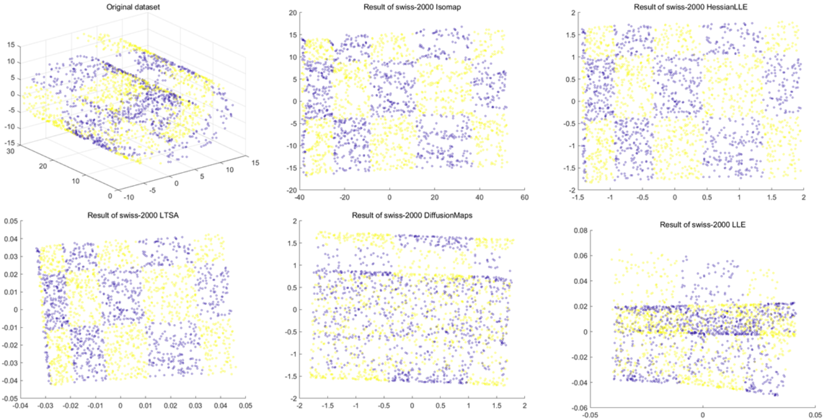
Dataset (d)	*None*	*PCA*	*Isomap*	*MVU*	*KPCA*	*DM*	*Autoenc.*	*LLE*	*LEM*	*HLLE*	*LTSA*	*LLC*	*MC*
Swiss roll (2D)	3.68%	30.56%	3.28%	5.12%	29.30%	28.06%	30.58%	7.44%	10.16%	3.10%	**3.06%**	27.74%	42.74%
Helix (1D)	1.24%	38.56%	**1.22%**	3.76%	44.54%	36.18%	32.50%	20.38%	10.34%	failed	1.68%	26.68%	28.16%
Twinpeaks (2D)	0.40%	0.18%	0.30%	0.58%	0.08%	**0.06%**	0.12%	0.54%	0.52%	0.10%	0.36%	12.96%	**0.06%**
Broken Swiss (2D)	2.14%	27.62%	14.24%	36.28%	27.06%	23.92%	26.32%	37.06%	26.08%	**4.78%**	16.30%	26.96%	23.92%
HD (5D)	24.19%	22.14%	20.45%	23.62%	29.25%	34.75%	**16.29%**	35.81%	41.70%	47.97%	40.22%	38.69%	31.46%

Repeated results:

Dataset	None	PCA	Isomap	MVU	KPCA	DM	Autoenc.	LLE	LEM	HLLE	LTSA	LLC	MC
Swiss roll	5.55%	30.50%	5.45%	0.00%	0.00%	30.60%	48.20%	31.10%	25.55%	6.10%	6.15%	21.80%	16.15 %
Helix	1.70%	3.00%	1.45%	0.00%	0.00%	3.00%	2.80%	20.55%	1.40%	1.25%	1.40%	3.80%	4.00%
Twinpeaks	0.60%	0.40%	0.20%	0.00%	0.00%	0.15%	0.30%	2.65%	0.55%	0.35%	0.20%	0.85%	0.50%

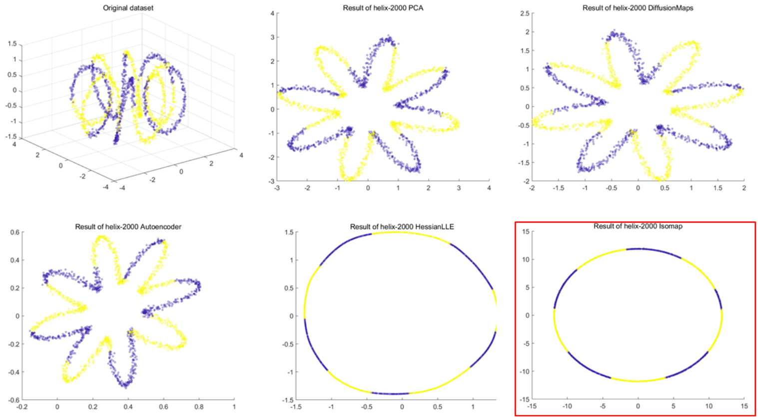
- Not know author's **clear parameters** and **repeated times**.
- **Variance** of all the methods should be measured.(TODO)

Experiments on swiss roll (Ying Fu)



LLE performs poorly maybe due to the parameters of k (default=12) is too large, cause it folds.

Experiments on helix (Ying Fu)



PCA, Diffusion map, and autoencoders perform poorly compared to Isomap if measured only on generalization errors of 1-NN classifiers?

Eigenproblems while performing experiments(Ying Fu)

MVU

```
Compute embedding (solve eigenproblem)...
警告: 将忽略 options 结构体中的 issym 字段，因为第一个输入不是函数句柄。
警告: 第一个输入矩阵接近奇异，或者缩放错误。RCOND =  1.602647e-17。结果可能不准确。
Running Maximum Variance Unfolding...
CSDP OUTPUT ========================================================================
fx 错误使用 cca (line 61)
```

Kernel PCA

```
Eigenanalysis of kernel matrix (using slower but memory-conservative implementation)...
.警告: 矩阵接近奇异值，或者缩放错误。结果可能不准确。RCOND =  8.496938e-24。
.错误使用  *
用于矩阵乘法的维度不正确。请检查并确保第一个矩阵中的列数与第二个矩阵中的行数匹配。要执行按元素相乘，请使用 '.*'。
```

Weaknesses of local techniques

weaknesses:

1. Local techniques may suffer from the **curse of dimensionality** of the embedded manifold.
2. **Eigenproblems**.
3. Local properties of a manifold do not necessarily follow the global structure of the manifold.(**overfitting**)
4. Local techniques assume that the manifold **contains no discontinuities**.
5. Cannot deal with manifolds that are **not isometric to Euclidean space**.
6. Local techniques may suffer from **folding**.

Global techniques

Weaknesses:

- global techniques for dimensionality reduction based on neighborhood graphs are often outperformed by PCA on artificial datasets.(Isomap and MVU)
- Kernel-based techniques are incapable of modelling certain complex manifolds.(Kernel PCA and diffusion maps)
- Techniques that optimize nonconvex objective functions may suffer from local optima in the objective functions.(e.g. autoencoders,LLC, and manifold charting.)

Summary

1. On **selected datasets**, nonlinear techniques for dimensionality reduction **outperform** linear techniques , but nonlinear techniques **perform poorly** on various other **natural datasets**.
2. **Two main weaknesses of local techniques**:
    1. The susceptibility to the curse of dimensionality.(TODO:Design techniques in which the global structure of the data manifold is represented in a number of linear models.)
    2. The problems in finding the smallest eigenvalues in an eigenproblem.(TODO: Design techniques with objective functions that can be optimized well in practice.)

Bibliography

[1] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, p. 13, 2009.

[2] J. De la Porte, B. Herbst, W. Hereman, and S. Van Der Walt, "An introduction to diffusion maps," in *Proceedings of the 19th Symposium of the Pattern Recognition Association of South Africa (PRASA 2008), Cape Town, South Africa*, 2008, pp. 15–25.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in neural information processing systems*, 2002, pp. 585–591.

[4] M. Brand, "Charting a manifold," in *Advances in neural information processing systems*, 2003, pp. 985–992.