Nonlinear Dimensionality Reduction

Yunpeng Shi

October 9, 2017
What is Dimension Reduction?
Figure: High dimensional data usually have low dimensional structure
Assume that there are \( n \) data points \( x_i \in \mathbb{R}^p \), we want to find a map \( \Phi(x) \mapsto y \) such that \( y \in \mathbb{R}^d \) where \( d < p \) or even \( d \ll p \). Namely, we are finding a map, either linear or nonlinear, that projects the high dimensional data points into lower dimensional one.
Principal Component Analysis (PCA)

- Seeks an optimal low dimensional vector space that gives smallest projection distance for the input data
Principal Component Analysis (PCA)

- Seeks an optimal low dimensional vector space that gives smallest projection distance for the input data
- The mapping is linear
Principal Component Analysis (PCA)

- Seeks an optimal low dimensional vector space that gives smallest projection distance for the input data
- The mapping is linear

The problem is formulated as follows:
Given a set of high dimensional data \( \{x_i\}_{i=1}^n \) where \( x_i \in \mathbb{R}^p \), we want to find a \( d \)-dimensional linear subspace \( L \) such that:

\[
L = \arg\min_L \sum_{i=1}^n \text{dist}^2(x_i, L)
\]  \( (1) \)
PCA 2D case

Figure: Illustration of PCA
PCA

1. Center the data set to origin.
1 Center the data set to origin.

2 Do eigenvalue decomposition for Gram matrix $X^T X$ where

$$X = [x_1, x_2, ..., x_n]$$

Namely, $K = X^T X = U \Lambda U^T$ where $U$ is orthogonal matrix. Equivalently $U$ can also be obtained by directly applying SVD to $X$. 

Yunpeng Shi

Nonlinear Dimensionality Reduction
1. Center the data set to origin.

2. Do eigenvalue decomposition for Gram matrix $X^T X$ where $X = [x_1, x_2, ..., x_n]$
   Namely, $K = X^T X = U \Lambda U^T$ where $U$ is orthogonal matrix. Equivalently $U$ can also be obtained by directly applying SVD to $X$.

3. Let $U = [U_1, U_2, ... U_n]$. then $L = \text{span}\{U_1, U_2..., U_d\}$
1. Center the data set to origin.
2. Do eigenvalue decomposition for Gram matrix $X^TX$ where $X = [x_1, x_2, ..., x_n]$
    Namely, $K = X^TX = U\Lambda U^T$ where $U$ is orthogonal matrix.
    Equivalently $U$ can also be obtained by directly applying SVD to $X$.
3. Let $U = [U_1, U_2, ...U_n]$. then $L = \text{span}\{U_1, U_2..., U_d\}$
4. Project the centered data to $L$, we have $y_i = [U_1, U_2..., U_d]^T x_i$. 

Yunpeng Shi
Nonlinear Dimensionality Reduction
1. Center the data set to origin.

2. Do eigenvalue decomposition for Gram matrix $X^T X$ where $X = [x_1, x_2, ..., x_n]$ 
   Namely, $K = X^T X = UΛU^T$ where $U$ is orthogonal matrix. Equivalently $U$ can also be obtained by directly applying SVD to $X$.

3. Let $U = [U_1, U_2, ... U_n]$. then $L = \text{span}\{U_1, U_2, ..., U_d\}$

4. Project the centered data to $L$, we have $y_i = [U_1, U_2, ..., U_d]^T x_i$. 
Good-of-Fit Measure

\[
\frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{n} \lambda_i}
\]
Nonlinear Dimension Reduction and Manifold Learning

What if data structure is intrinsically nonlinear?

Figure: swiss roll data
PCA does not work well in nonlinear case

What if data structure is intrinsically nonlinear?
Capture nonlinearity

**Kernel matrix**
Each element of kernel matrix can be viewed as inner product in feature space. namely, $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$, where $\Phi(\cdot)$ is a mapping to feature space. $K(x, x')$ can be viewed as similarity between $x$ and $x'$. Usually, we use kernel:
1. linear kernel $k(x, x') = \langle x, x' \rangle$
2. polynomial kernel $k(x, x') = (\langle x, x' \rangle + c)^d$
3. Gaussian kernel $k(x, x') = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$
Capture nonlinearity

**Distance Matrix**

For graph-based methods, usually we use pairwise distance information to obtain adjacency matrix. The linear kernel matrix (gram matrix $X^TX$) can also be obtained from Euclidean distance matrix by using double centering [4]:

$$K = -\frac{1}{2}HD^{(2)}H$$

where $D^{(2)}$ is the matrix of distance square, and $H$ is centering matrix $I - \frac{1}{n}11^T$.

$$k_{ij} = -\frac{1}{2} \left( d_{ij}^2 - \frac{1}{n} \sum_l d_{il}^2 - \frac{1}{n} \sum_l d_{jl}^2 + \frac{1}{n^2} \sum_{lm} d_{lm}^2 \right).$$

Conversely, we can also get $D$ from Gram matrix $K$. 
Classical Scaling

\[ \phi(Y) = \sum_{ij} (d_{ij}^2 - \| y_i - y_j \|^2) \]

\( y_i \) is restricted to be \( x_iM \), and \( \| m_j \|^2 = 1 \) for \( \forall j \).
Isomap

- Define Adjacency Matrix from Distance.

Yunpeng Shi
Nonlinear Dimensionality Reduction
Isomap

- Define Adjacency Matrix from Distance.
- Compute geodesic distance between two nodes.
Isomap

- Define Adjacency Matrix from Distance.
- Compute geodesic distance between two nodes.
Isomap

Weakness

- short-circuiting
- suffer from 'holes' in the manifold
- suffer from nonconvex manifold

Applications

- wood inspection
- visualization of biomedical data
- head pose estimation
Kernel PCA

\[ y_i = \left\{ \sum_{j=1}^{n} a_{1}^{(j)} \kappa(x_j, x_i), \ldots, \sum_{j=1}^{n} a_{d}^{(j)} \kappa(x_j, x_i) \right\} \]

Weakness: focus too much on global distances
Applications: face recognition, speech recognition, and novelty detection
Maximum Variance Unfolding (MVU) [4]
Assume that the data are centered at the origin. The Gram matrix $X^TX$. Then MVU learn the kernel matrix $K$ in the following way:

Maximize $\sum_{i,j} \| \tilde{y}_i - \tilde{y}_j \|^2$ subject to:
1. $\| \tilde{y}_i - \tilde{y}_j \|^2 = \| \tilde{x}_i - \tilde{x}_j \|^2$ for all $(i, j)$ with $\eta_{ij} = 1$.
2. $\sum_i \tilde{y}_i = 0$

Maximize $\text{trace}(K)$ subject to:
1. $K_{ii} - 2K_{ij} + K_{jj} = \| \tilde{x}_i - \tilde{x}_j \|^2$ for all $(i, j)$ with $\eta_{ij} = 1$.
2. $\sum_{i,j} K_{ij} = 0$.
3. $K \succeq 0$. 
SDP problem.
SDP problem.
Preserve local distance.
MVU

- SDP problem.
- Preserve local distance.
- Maximize global variance (unfolding).

Weakness: short-circuiting
Applications: sensor localization, DNA microarray data analysis
Diffusion maps

- Construct graph of data using Gaussian kernel
- Use the weights as transition probability to form a Markov chain
- Choose eigenvectors of transition matrix as low dimensional representation of data
Diffusion maps

\[
\Pr(X_{t+1} = j | X_t = i) = M_{ij} = \frac{w_{ij}}{\sum_i w_{ij}}, \text{ or } M = D^{-1}W
\]

From SVD,

\[
M = \Phi \Lambda \Psi^T
\]

(Diffusion Map) Given a graph \( G = (V, E, W) \) construct \( M \) and its decomposition described above. The Diffusion Map is a map \( \phi_t : V \to \mathbb{R}^{n-1} \) given by

\[
\phi_t (v_i) = \begin{bmatrix}
\lambda_2^t \varphi_2(i) \\
\lambda_2^t \varphi_3(i) \\
\vdots \\
\lambda_n^t \varphi_n(i)
\end{bmatrix}
\]
Truncated diffusion maps

\[ \phi_t^{(d)} (v_i) = \begin{bmatrix} \lambda_2^t \varphi_2(i) \\ \lambda_3^t \varphi_3(i) \\ \vdots \\ \lambda_{d+1}^t \varphi_{d+1}(i) \end{bmatrix}. \]
\[ \| \phi_t(v_{i_1}) - \phi_t(v_{i_2}) \|^2 = \sum_{j=1}^{n} \frac{1}{\deg(j)} \left[ \text{Prob} \{ X(t) = j | X(0) = i_1 \} - \text{Prob} \{ X(t) = j | X(0) = i_2 \} \right]^2. \]

**Proof.**

Note that \( \sum_{j=1}^{n} \frac{1}{\deg(j)} \left[ \text{Prob} \{ X(t) = j | X(0) = i_1 \} - \text{Prob} \{ X(t) = j | X(0) = i_2 \} \right]^2 \) can be rewritten as

\[
\sum_{j=1}^{n} \frac{1}{\deg(j)} \left[ \sum_{k=1}^{n} \lambda_k^t \varphi_k(i_1) \psi_k(j) - \sum_{k=1}^{n} \lambda_k^t \varphi_k(i_2) \psi_k(j) \right]^2 = \sum_{j=1}^{n} \frac{1}{\deg(j)} \left[ \sum_{k=1}^{n} \lambda_k^t (\varphi_k(i_1) - \varphi_k(i_2)) \psi_k(j) \right]^2
\]

and

\[
\sum_{j=1}^{n} \frac{1}{\deg(j)} \left[ \sum_{k=1}^{n} \lambda_k^t (\varphi_k(i_1) - \varphi_k(i_2)) \psi_k(j) \right]^2 = \sum_{j=1}^{n} \left[ \sum_{k=1}^{n} \lambda_k^t (\varphi_k(i_1) - \varphi_k(i_2)) \frac{\psi_k(j)}{\sqrt{\deg(j)}} \right]^2
\]

\[= \left\| \sum_{k=1}^{n} \lambda_k^t (\varphi_k(i_1) - \varphi_k(i_2)) D^{-\frac{1}{2}} \psi_k \right\|^2. \]
Note that $D^{-\frac{1}{2}}\psi_k = v_k$ which forms an orthonormal basis, meaning that

$$\left\| \sum_{k=1}^{n} \lambda_k^t (\varphi_k(i_1) - \varphi_k(i_2)) D^{-\frac{1}{2}}\psi_k \right\|^2 = \sum_{k=1}^{n} (\lambda_k^t (\varphi_k(i_1) - \varphi_k(i_2)))^2$$

$$= \sum_{k=2}^{n} (\lambda_k^t \varphi_k(i_1) - \lambda_k^t \varphi_k(i_2))^2,$$

where the last inequality follows from the fact that $\varphi_1 = 1$ and concludes the proof of the theorem. $\blacksquare$
Define pairwise probabilities

\[ p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \]

\[ p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \]

Pairwise probabilities in target space

\[ q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq m} (1 + \|y_k - y_m\|^2)^{-1}} \]

minimize Kullback-Leibler divergence

\[ KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \]
Locally linear embedding (LLE)

- Construct graph from distance matrix using KNN
Locally linear embedding (LLE)

- Construct graph from distance matrix using KNN
- Assumes that each node is a convex combination of its neighbors (locally linear assumption)
**Locally linear embedding (LLE)**

- Construct graph from distance matrix using KNN
- Assumes that each node is a convex combination of its neighbors (locally linear assumption)
- Only need to solve two Least squares problem.
solve for eigenvectors corresponding to d-smallest nonzero eigenvalues of \((I - W)^T (I - W)\).
**LLE**

![LLE 2D Graph](image1)

![LLE 3D Graph](image2)
Weakness

- suffers from manifolds that contain holes
- tends to collapse large portions of the data very close together
- covariance constraint may give rise to undesired rescalings

Applications: sound source localization
Laplacian Eigenmaps [1]

- Define similarity via Gaussian kernel $e^{-\frac{||x_i - x_j||^2}{t}}$. 

Laplacian Eigenmaps
Laplacean Eigenmaps

Laplacean Eigenmaps [1]

- Define similarity via Gaussian kernel $e^{-\frac{\|x_i - x_j\|^2}{t}}$.
- Compute geodesic distance between two nodes.
Laplacian Eigenmaps [1]

- Define similarity via Gaussian kernel $e^{-\frac{||x_i - x_j||^2}{t}}$.
- Compute geodesic distance between two nodes.
- For each pair of similar nodes, it minimize the embedded pairwise distance.

It solves the optimization problem:

$$\min_Y \sum_{(i,j)} ||y_i - y_j||^2 W_{ij}$$
Laplacian Eigenmaps

The problem above can be rewritten as:

$$\min_Y \text{Tr}(Y^TLY)$$

where $Y^TDY = I$ is used to fix the scale and avoid collapsed solution. Degree matrix $D$ is diagonal matrix of row sum of $W$ and graph Laplacian $L =: D - W$. The standard solution for the problem above is given by solving eigenvalue problem $Lf = \lambda Df$ and m-dimensional embedding $y_i = [f_1(i), f_2(i), \ldots f_m(i)]$ where $f_i$ is the eigenvector corresponding to ith smallest eigenvalue (except 0).
Laplacian Eigenmaps

Laplacian Eigenmap Sigma=1

Laplacian Eigenmap 3D Sigma=1
Laplacian Eigenmaps

Weakness: tends to collapse
Applications: face recognition, analysis of fMRI data, semi-supervised learning
Hessian LLE

- build a graph via KNN
- apply PCA for each $N(x_i)$ to find tangent spaces $S_i$
- estimate tangent Hessian $H_i$

$$\mathcal{H}_{lm} = \sum_i \sum_j ((H_i)_{jl} \times (H_i)_{jm}).$$
Hessian LLE

Weakness: similar to Laplacian eigenmaps and LLE
Applications: sensor localization
Local Tangent Space Analysis (LTSA)

- build a graph via KNN
- apply PCA for each \( N(x_i) \) to find tangent spaces \( \Theta_i \)
- there exists a linear mapping \( L_i \) from the local tangent space coordinates \( \Theta_{ij} \) to the low-dimensional representations \( y_{ij} \)

\[
\min_{Y_i, L_i} \sum_i \|Y_i J_k - L_i \Theta_i \|^2
\]
Local Tangent Space Analysis (LTSA)

\[ \mathbf{B}_{N_i N_i} = \mathbf{B}_{N_{i-1} N_{i-1}} + \mathbf{J}_k \left( \mathbf{I} - \mathbf{V}_i \mathbf{V}_i^T \right) \mathbf{J}_k \]

find eigenvectors corresponding to \( d \) smallest nonzero eigenvalues of the symmetric matrix \( \frac{1}{2}(\mathbf{B} + \mathbf{B}^T) \)
Local Tangent Space Analysis (LTSA)

Weakness: trivial solutions
Applications: microarray data
Nonconvex Techniques: Sammon Mapping

\[
\phi(Y) = \frac{1}{\sum_{ij} d_{ij}} \sum_{i \neq j} \left( d_{ij} - \|y_i - y_j\| \right)^2
\]

Weakness: scales too much when \(d_{ij}\) small
Applications: gene and geospatial data
Nonconvex Techniques: Multilayer Autoencoders

- Weakness: tedious training
- Applications: data imputation, HIV data analysis
Nonconvex Techniques: Locally Linear Coordination (LLC)

- construct mixture of $m$ factor analyzers using EM algorithms
- construct $m$ data representations $z_{ij}$ and their corresponding responsibilities $r_{ij}$ for every datapoint $x_i$.
- build $n \times mD$ matrix $U$ that contains $u_{ij} = r_{ij} z_{ij}$
construct mixture of $m$ factor analyzers using EM algorithms
construct $m$ data representations $z_{ij}$ and their corresponding responsibilities $r_{ij}$ for every datapoint $x_i$.
build $n \times mD$ matrix $U$ that contains $u_{ij} = r_{ij}z_{ij}$

**Model alignment:**
solve $Av = \lambda Bv$, where $A$ is inproduct of $M^TU$, $B = U^TU$,
$M = (I - W)^T(I - W)$ from LLE.
$L = [v_1, v_2, \cdots v_d]$ and $Y = UL$. 

Nonconvex Techniques: Locally Linear Coordination (LLC)
**Weakness**: presence of local maxima in the log-likelihood function

**Applications**: images of a single person with variable pose and expression, handwritten digits
1. find $z_{ij}$ and $r_{ij}$ as before.
2. find a linear mapping $M$ from the data representations $z_{ij}$ to the global coordinates $y_i$ that minimizes the cost function

$$\phi(Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} r_{ij} \|y_i - y_{ij}\|^2$$

where $y_i = \sum_{k} y_{ik}$, $y_{ij} = z_{ij} M$.}

$$\phi(Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{m} r_{ij} r_{ik} \|y_{ij} - y_{ik}\|^2$$
Can be rewritten as

$$\phi(Y) = L^T (D - U^T U) L$$

where $D = \text{diag}(D_j) = \text{diag}(\sum_i r_{ij} \text{cov}(Z_j, 1))$, $u_{ij} = [r_{ij} z_{ij}, 1]$

$$\phi(Y) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^m r_{ij} r_{ik} \|y_{ij} - y_{ik}\|^2$$
## Computation Complexity

<table>
<thead>
<tr>
<th>Technique</th>
<th>Parametric</th>
<th>Parameters</th>
<th>Computational</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>yes</td>
<td>none</td>
<td>$O(D^3)$</td>
<td>$O(D^2)$</td>
</tr>
<tr>
<td>Class. scaling</td>
<td>no</td>
<td>none</td>
<td>$O(n^3)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Isomap</td>
<td>no</td>
<td>$k$</td>
<td>$O(n^3)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Kernel PCA</td>
<td>no</td>
<td>$\kappa(\cdot, \cdot)$</td>
<td>$O(n^3)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>MVU</td>
<td>no</td>
<td>$k$</td>
<td>$O((nk)^3)$</td>
<td>$O((nk)^3)$</td>
</tr>
<tr>
<td>Diffusion maps</td>
<td>no</td>
<td>$\sigma,t$</td>
<td>$O(n^3)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>LLE</td>
<td>no</td>
<td>$k$</td>
<td>$O(pn^2)$</td>
<td>$O(pn^2)$</td>
</tr>
<tr>
<td>Laplacian Eigenmaps</td>
<td>no</td>
<td>$k,\sigma$</td>
<td>$O(pn^2)$</td>
<td>$O(pn^2)$</td>
</tr>
<tr>
<td>Hessian LLE</td>
<td>no</td>
<td>$k$</td>
<td>$O(pn^2)$</td>
<td>$O(pn^2)$</td>
</tr>
<tr>
<td>LTSA</td>
<td>no</td>
<td>$k$</td>
<td>$O(pn^2)$</td>
<td>$O(pn^2)$</td>
</tr>
<tr>
<td>Sammon mapping</td>
<td>no</td>
<td>none</td>
<td>$O(in^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Autoencoders</td>
<td>yes</td>
<td>net size</td>
<td>$O(inw)$</td>
<td>$O(w)$</td>
</tr>
<tr>
<td>LLC</td>
<td>yes</td>
<td>$m,k$</td>
<td>$O(imd^3)$</td>
<td>$O(nmd)$</td>
</tr>
<tr>
<td>Manifold charting</td>
<td>yes</td>
<td>$m$</td>
<td>$O(imd^3)$</td>
<td>$O(nmd)$</td>
</tr>
</tbody>
</table>
Artificial Data

(a) Swiss roll dataset.

(b) Helix dataset.

(c) Twinpeaks dataset.

(d) Broken Swiss roll dataset.
Artificial Data

Evaluate to what extent the local structure of the data is retained:

1. **Generalization errors** of 1-nearest neighbor classifiers that are trained on the low-dimensional data representation.
2. **Trustworthiness**: if low-dim points are close to each other, does high-dim ones have the same pattern?
3. **Continuity**: if high dimensional points are close to each other, does the low-dim pts close to each other?
Artificial Data

\[ T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_{i}^{(k)}} (r(i, j) - k) \]

\[ C(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in V_{i}^{(k)}} (\hat{r}(i, j) - k) \]
Artificial Data

(a) True underlying manifold.

(b) Reconstructed manifold up to a non-linear warping.
### Artificial Data

<table>
<thead>
<tr>
<th>Technique</th>
<th>Parameter settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>None</td>
</tr>
<tr>
<td>Isomap</td>
<td>$5 \leq k \leq 15$</td>
</tr>
<tr>
<td>Kernel PCA</td>
<td>$\kappa = (X X^T + 1)^5$</td>
</tr>
<tr>
<td>MVU</td>
<td>$5 \leq k \leq 15$</td>
</tr>
<tr>
<td>Diffusion maps</td>
<td>$10 \leq t \leq 100$ $\sigma = 1$</td>
</tr>
<tr>
<td>LLE</td>
<td>$5 \leq k \leq 15$</td>
</tr>
<tr>
<td>Laplacian Eigenmaps</td>
<td>$5 \leq k \leq 15$ $\sigma = 1$</td>
</tr>
<tr>
<td>Hessian LLE</td>
<td>$5 \leq k \leq 15$</td>
</tr>
<tr>
<td>LTSA</td>
<td>$5 \leq k \leq 15$</td>
</tr>
<tr>
<td>Sammon mapping</td>
<td>None</td>
</tr>
<tr>
<td>Autoencoders</td>
<td>Three hidden layers</td>
</tr>
<tr>
<td>LLC</td>
<td>$5 \leq k \leq 15$ $5 \leq m \leq 25$</td>
</tr>
<tr>
<td>Manifold charting</td>
<td>$5 \leq m \leq 25$</td>
</tr>
</tbody>
</table>
### Artificial Data

<table>
<thead>
<tr>
<th>Dataset (d)</th>
<th>None</th>
<th>PCA</th>
<th>Isomap</th>
<th>KPCA</th>
<th>MVU</th>
<th>DM</th>
<th>LLE</th>
<th>LEM</th>
<th>HILLE</th>
<th>LTSA</th>
<th>Sammon</th>
<th>Autoenc</th>
<th>LLC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swiss roll (2D)</td>
<td>3.68%</td>
<td>29.76%</td>
<td>3.40%</td>
<td>30.24%</td>
<td>4.12%</td>
<td>33.50%</td>
<td>3.74%</td>
<td>22.06%</td>
<td>3.56%</td>
<td>3.90%</td>
<td>22.34%</td>
<td>49.00%</td>
<td>26.72%</td>
<td>22.66%</td>
</tr>
<tr>
<td>Helix (1D)</td>
<td>1.24%</td>
<td>35.50%</td>
<td>13.18%</td>
<td>38.04%</td>
<td>7.48%</td>
<td>35.44%</td>
<td>32.12%</td>
<td>15.24%</td>
<td>52.22%</td>
<td>0.92%</td>
<td>52.22%</td>
<td>52.22%</td>
<td>27.44%</td>
<td>25.94%</td>
</tr>
<tr>
<td>Twin peaks (2D)</td>
<td>0.40%</td>
<td>0.26%</td>
<td>0.22%</td>
<td>0.12%</td>
<td>0.56%</td>
<td>0.26%</td>
<td>0.94%</td>
<td>0.88%</td>
<td>0.14%</td>
<td>0.18%</td>
<td>0.32%</td>
<td>49.06%</td>
<td>11.04%</td>
<td>0.30%</td>
</tr>
<tr>
<td>Broken Swiss (2D)</td>
<td>2.14%</td>
<td>25.96%</td>
<td>14.48%</td>
<td>32.06%</td>
<td>32.06%</td>
<td>58.26%</td>
<td>20.74%</td>
<td>24.70%</td>
<td>42.62%</td>
<td>6.48%</td>
<td>27.40%</td>
<td>87.86%</td>
<td>37.06%</td>
<td>32.24%</td>
</tr>
<tr>
<td>HD (5D)</td>
<td>24.19%</td>
<td>22.18%</td>
<td>23.26%</td>
<td>27.46%</td>
<td>25.38%</td>
<td>23.14%</td>
<td>36.94%</td>
<td>10.66%</td>
<td>15.86%</td>
<td>37.06%</td>
<td>20.70%</td>
<td>49.18%</td>
<td>34.14%</td>
<td>21.34%</td>
</tr>
</tbody>
</table>

Table 3: Generalization errors of 1-NN classifiers trained on artificial datasets (smaller numbers are better).

<table>
<thead>
<tr>
<th>Dataset (d)</th>
<th>None</th>
<th>PCA</th>
<th>Isomap</th>
<th>KPCA</th>
<th>MVU</th>
<th>DM</th>
<th>LLE</th>
<th>LEM</th>
<th>HILLE</th>
<th>LTSA</th>
<th>Sammon</th>
<th>Autoenc</th>
<th>LLC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swiss roll (2D)</td>
<td>0.88</td>
<td>0.78</td>
<td>0.72</td>
<td>0.96</td>
<td>0.73</td>
<td>0.83</td>
<td>0.88</td>
<td>0.87</td>
<td>0.35</td>
<td>1.00</td>
<td>0.89</td>
<td>0.46</td>
<td>0.81</td>
<td>0.88</td>
</tr>
<tr>
<td>Helix (1D)</td>
<td>0.98</td>
<td>0.96</td>
<td>0.98</td>
<td>0.97</td>
<td>0.78</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
<td>0.89</td>
<td>1.00</td>
<td>0.97</td>
<td>0.76</td>
<td>0.76</td>
<td>0.83</td>
</tr>
<tr>
<td>Twin peaks (2D)</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>0.97</td>
<td>0.78</td>
<td>0.94</td>
<td>0.97</td>
<td>0.97</td>
<td>0.89</td>
<td>1.00</td>
<td>0.97</td>
<td>0.52</td>
<td>0.86</td>
<td>1.00</td>
</tr>
<tr>
<td>Broken Swiss (2D)</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>0.68</td>
<td>0.89</td>
<td>1.00</td>
</tr>
<tr>
<td>HD (5D)</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>0.89</td>
<td>0.91</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 4: Trustworthinesses $T(12)$ on the artificial datasets (larger numbers are better).

<table>
<thead>
<tr>
<th>Dataset (d)</th>
<th>None</th>
<th>PCA</th>
<th>Isomap</th>
<th>KPCA</th>
<th>MVU</th>
<th>DM</th>
<th>LLE</th>
<th>LEM</th>
<th>HILLE</th>
<th>LTSA</th>
<th>Sammon</th>
<th>Autoenc</th>
<th>LLC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swiss roll (2D)</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>0.91</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Helix (1D)</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.75</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>Twin peaks (2D)</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.98</td>
<td>1.00</td>
</tr>
<tr>
<td>Broken Swiss (2D)</td>
<td>1.00</td>
<td>0.98</td>
<td>0.99</td>
<td>1.00</td>
<td>0.90</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>0.73</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>HD (5D)</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.89</td>
<td>0.91</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5: Continuity $C(12)$ on the artificial datasets (larger numbers are better).
Artificial Data

- graph-based methods in general performs well
- LLE/HLLE may perform less well on manifolds that are not isometric to Euclidean space.
- high generalization errors on the broken Swiss roll dataset
- nonlinear techniques may have problems when they are faced with a dataset with a high intrinsic dimensionality
- strong performance on the Swiss roll dataset does not always generalize to other dataset
Dataset: MNIST, COIL20, NiSIS, ORL, HIVA
## Natural Data

<table>
<thead>
<tr>
<th>Dataset (d)</th>
<th>None</th>
<th>PCA</th>
<th>Isomap</th>
<th>KPCA</th>
<th>MVU</th>
<th>DM</th>
<th>LLE</th>
<th>LEM</th>
<th>HLLE</th>
<th>LTS</th>
<th>Sammon</th>
<th>Autoenc.</th>
<th>LLC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST (20D)</td>
<td>5.11%</td>
<td>6.74%</td>
<td>12.64%</td>
<td>13.86%</td>
<td>13.58%</td>
<td>25.00%</td>
<td>10.02%</td>
<td>13.27%</td>
<td>91.66%</td>
<td>90.32%</td>
<td>6.90%</td>
<td>7.18%</td>
<td>16.12%</td>
<td>14.84%</td>
</tr>
<tr>
<td>COIL20 (5D)</td>
<td>0.14%</td>
<td>3.82%</td>
<td>15.69%</td>
<td>7.78%</td>
<td>25.14%</td>
<td>11.18%</td>
<td>22.29%</td>
<td>11.00%</td>
<td>95.00%</td>
<td>50.35%</td>
<td>0.83%</td>
<td>51.11%</td>
<td>4.31%</td>
<td>27.36%</td>
</tr>
<tr>
<td>ORL (8D)</td>
<td>2.50%</td>
<td>4.75%</td>
<td>27.50%</td>
<td>6.25%</td>
<td>24.25%</td>
<td>90.00%</td>
<td>15.48%</td>
<td>47.59%</td>
<td>48.98%</td>
<td>24.68%</td>
<td>2.75%</td>
<td>6.25%</td>
<td>11.25%</td>
<td>22.50%</td>
</tr>
<tr>
<td>NiSIS (15D)</td>
<td>8.24%</td>
<td>7.95%</td>
<td>13.36%</td>
<td>9.55%</td>
<td>15.67%</td>
<td>48.98%</td>
<td>4.97%</td>
<td>4.81%</td>
<td>3.51%</td>
<td>3.51%</td>
<td>48.98%</td>
<td>9.22%</td>
<td>26.86%</td>
<td>18.91%</td>
</tr>
<tr>
<td>HIVA (15D)</td>
<td>4.63%</td>
<td>5.05%</td>
<td>4.92%</td>
<td>5.07%</td>
<td>4.94%</td>
<td>5.46%</td>
<td>4.97%</td>
<td>4.81%</td>
<td>3.51%</td>
<td>3.51%</td>
<td>48.98%</td>
<td>9.22%</td>
<td>26.86%</td>
<td>18.91%</td>
</tr>
</tbody>
</table>

Table 6: Generalization errors of 1-NN classifiers trained on natural datasets (smaller numbers are better).

<table>
<thead>
<tr>
<th>Dataset (d)</th>
<th>None</th>
<th>PCA</th>
<th>Isomap</th>
<th>KPCA</th>
<th>MVU</th>
<th>DM</th>
<th>LLE</th>
<th>LEM</th>
<th>HLLE</th>
<th>LTS</th>
<th>Sammon</th>
<th>Autoenc.</th>
<th>LLC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST (20D)</td>
<td>---</td>
<td>1.00</td>
<td>0.96</td>
<td>0.99</td>
<td>0.92</td>
<td>0.95</td>
<td>0.96</td>
<td>0.89</td>
<td>0.54</td>
<td>0.54</td>
<td>1.00</td>
<td>1.00</td>
<td>0.93</td>
<td>0.97</td>
</tr>
<tr>
<td>COIL20 (5D)</td>
<td>---</td>
<td>0.99</td>
<td>0.89</td>
<td>0.98</td>
<td>0.92</td>
<td>0.91</td>
<td>0.93</td>
<td>0.27</td>
<td>0.69</td>
<td>0.96</td>
<td>0.99</td>
<td>0.88</td>
<td>0.96</td>
<td>0.92</td>
</tr>
<tr>
<td>ORL (8D)</td>
<td>---</td>
<td>0.99</td>
<td>0.78</td>
<td>0.98</td>
<td>0.95</td>
<td>0.49</td>
<td>0.95</td>
<td>0.29</td>
<td>0.76</td>
<td>0.94</td>
<td>0.99</td>
<td>0.99</td>
<td>0.79</td>
<td>0.82</td>
</tr>
<tr>
<td>NiSIS (15D)</td>
<td>---</td>
<td>0.99</td>
<td>0.89</td>
<td>0.99</td>
<td>0.90</td>
<td>0.40</td>
<td>0.92</td>
<td>0.47</td>
<td>0.47</td>
<td>0.82</td>
<td>0.47</td>
<td>0.99</td>
<td>0.85</td>
<td>0.89</td>
</tr>
<tr>
<td>HIVA (15D)</td>
<td>---</td>
<td>0.97</td>
<td>0.87</td>
<td>0.89</td>
<td>0.89</td>
<td>0.75</td>
<td>0.80</td>
<td>0.78</td>
<td>0.42</td>
<td>0.54</td>
<td>0.42</td>
<td>0.98</td>
<td>0.91</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 7: Trustworthinesses $T(12)$ on the natural datasets (larger numbers are better).

<table>
<thead>
<tr>
<th>Dataset (d)</th>
<th>None</th>
<th>PCA</th>
<th>Isomap</th>
<th>KPCA</th>
<th>MVU</th>
<th>DM</th>
<th>LLE</th>
<th>LEM</th>
<th>HLLE</th>
<th>LTS</th>
<th>Sammon</th>
<th>Autoenc.</th>
<th>LLC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST (20D)</td>
<td>---</td>
<td>1.00</td>
<td>0.94</td>
<td>0.89</td>
<td>0.93</td>
<td>0.95</td>
<td>0.96</td>
<td>0.70</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>COIL20 (5D)</td>
<td>---</td>
<td>1.00</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.92</td>
<td>0.95</td>
<td>0.47</td>
<td>0.71</td>
<td>0.99</td>
<td>1.00</td>
<td>0.92</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td>ORL (8D)</td>
<td>---</td>
<td>0.99</td>
<td>0.76</td>
<td>0.95</td>
<td>0.97</td>
<td>0.57</td>
<td>0.95</td>
<td>0.49</td>
<td>0.76</td>
<td>0.94</td>
<td>0.99</td>
<td>0.98</td>
<td>0.80</td>
<td>0.79</td>
</tr>
<tr>
<td>NiSIS (15D)</td>
<td>---</td>
<td>1.00</td>
<td>0.84</td>
<td>0.98</td>
<td>0.94</td>
<td>0.48</td>
<td>0.91</td>
<td>0.48</td>
<td>0.47</td>
<td>0.64</td>
<td>0.47</td>
<td>1.00</td>
<td>0.84</td>
<td>0.89</td>
</tr>
<tr>
<td>HIVA (15D)</td>
<td>---</td>
<td>0.99</td>
<td>0.84</td>
<td>0.88</td>
<td>0.94</td>
<td>0.80</td>
<td>0.80</td>
<td>0.54</td>
<td>0.51</td>
<td>0.62</td>
<td>0.51</td>
<td>0.99</td>
<td>0.87</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 8: Continuity $C(12)$ on the natural datasets (larger numbers are better).
Full Spectral methods

- **graph-based methods**: may suffer from short-curcuiting issue
- **kernel methods**: choose proper kernel is an issue (suffer from curse of dimensionality)
Sparse spectral methods:
- covariance constraint can be easily cheated
- curse of dimensionality
- difficulty of solving eigen problems
- overfitting (data distribution),
- outliers (use eps-ball instead of KNN)
- real-world data violates smoothness assumption
Nonconvex methods:

- cons: may stuck at local min/max
- pros: more flexibility in designing formulation, may allow higher model complexity and tackle more variations of data.
M. Belkin, and P. Niyogi. ”Laplacian eigenmaps for dimensionality reduction and data representation.” *Neural computation* 15, no. 6 (2003): 1373-1396.


B. Shaw. Graph embedding and nonlinear dimensionality reduction. Columbia University, 2011.


Thanks!