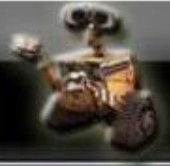


Artificial Intelligence

Continuous Latent Variables

Donghui Wang
AI Institute@ZJU
2015.4

Contents

- Principal Component Analysis (PCA)
- Probabilistic PCA
- Kernel PCA
- Nonlinear Latent Variable Models

References:

1. *Bishop. "Pattern Recognition and Machine Learning", Chapter 12. 2006.*

Principal Component Analysis (PCA)

What's PCA?

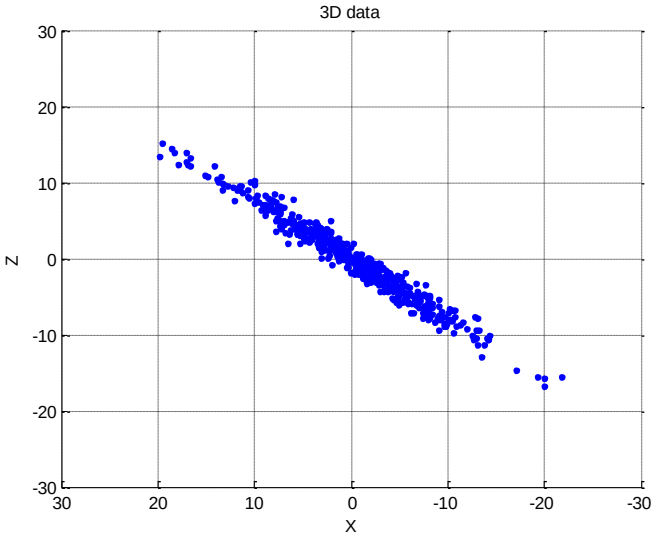
- PCA - Principal Component Analysis
  - *Karhunen-Loève* transform

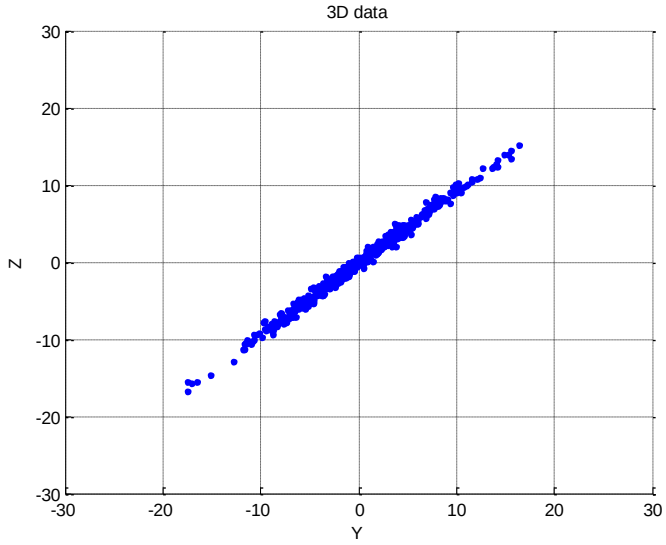
- Two commonly used definitions:
  - The orthogonal projection of the data onto a lower dimensional linear space, known as the *principal subspace*, such that the variance of the projected data is maximized.
  - The linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections.

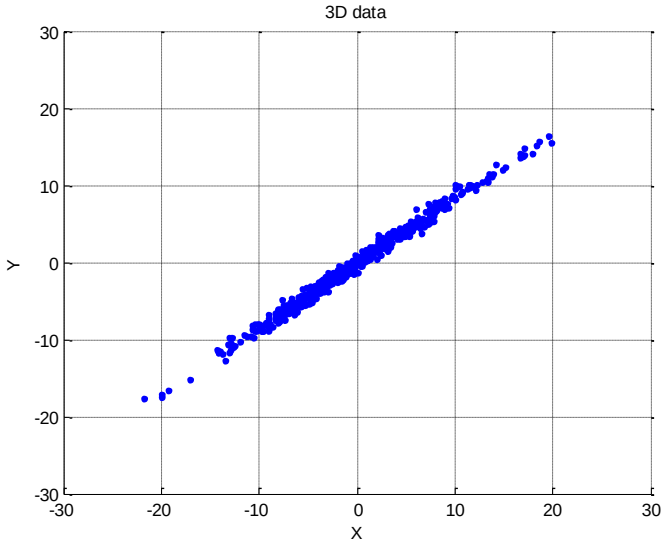
Why PCA?

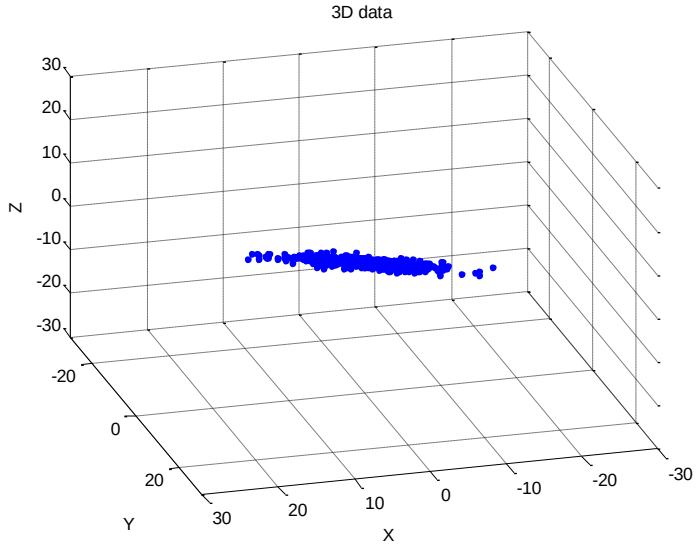
- Dimensionality reduction
  - Avoid the curse of dimensionality
- Lossy data compression
- Feature extraction
- Data visualization
  - How to visualize high-dimensional data?
- ...

- Dir
  - –
- Lo
- Fe
- Da
  - –
- …









Maximum variance formulation

- Consider a data set of observations $\{\mathbf{x}_n\}$ where n=1,…,N, and $\mathbf{x}_n$ is a D-dimensional vector.
- Goal: *project the data onto a M-dimensional subspace (M<D) while maximizing the variance of the projected data.*

- *M=1:*
  - *define the direction of this space using a D-dimensional unit vector* $\mathbf{u}_1$ and $\mathbf{u}^{\mathrm{T}}_1\mathbf{u}_1 = 1$.

Maximizing it.

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \qquad \mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^{\mathrm{T}} \Rightarrow \frac{1}{N}\sum_{n=1}^{N}\left\{\mathbf{u}_1^{\mathrm{T}}\mathbf{x}_n - \mathbf{u}_1^{\mathrm{T}}\bar{\mathbf{x}}\right\}^2 = \mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1$$

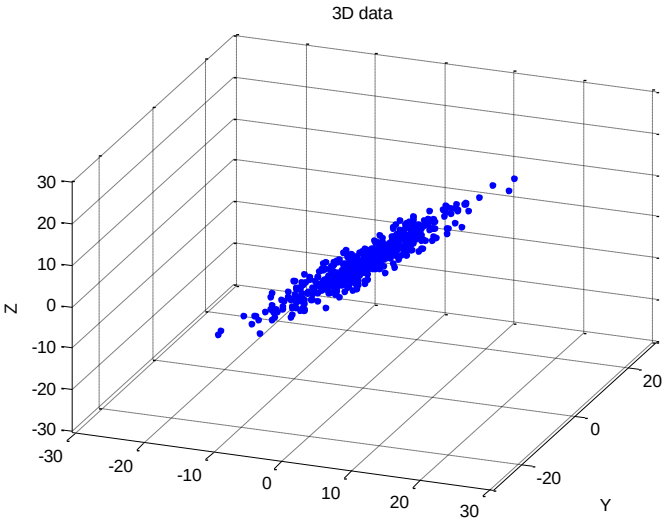
Lagrange multiplier

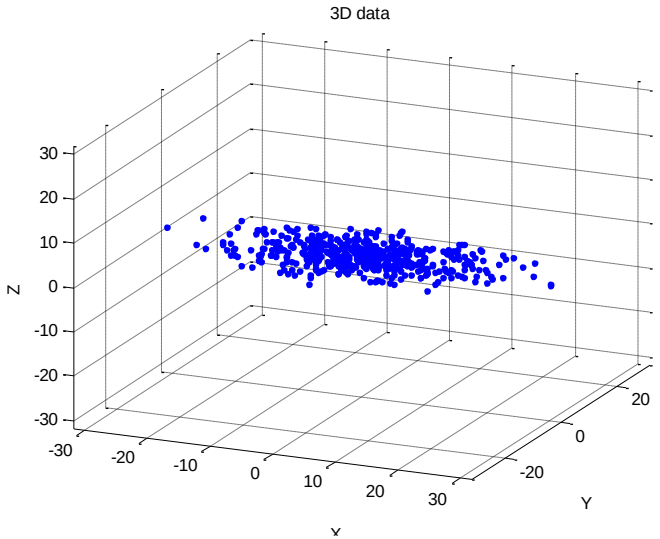
$$\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 = 1$$

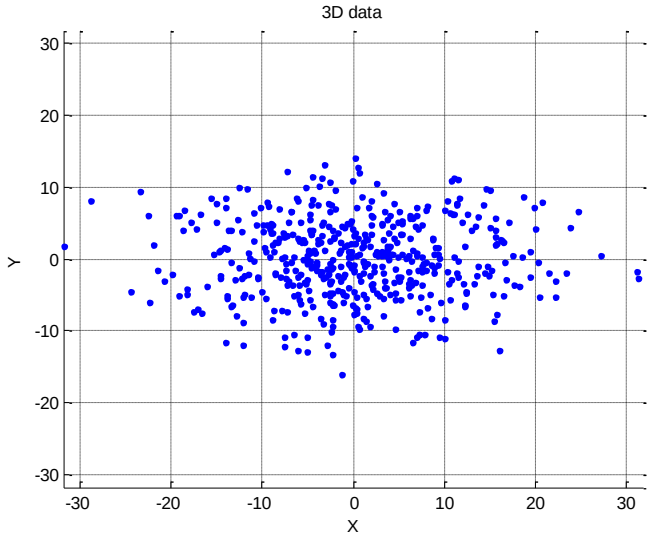
$$\mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1 + \lambda_1\left(1 - \mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1\right) \Rightarrow \mathbf{S}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \Rightarrow \mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1 = \lambda_1$$

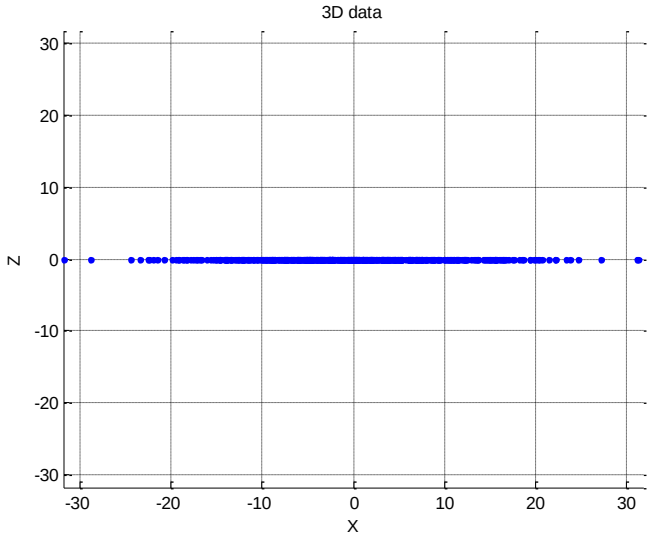
The variance will be a maximum when we set u_1 equal to the eigenvector having the largest eigenvalue λ_1.

- *M>1:*
  - *Optimal linear projection is now defined by the M eigenvectors of the data covariance matrix S corresponding to the M largest eigenvalues.*


3D data
Z
Y


3D data
Z
Y


3D data
Y
X


3D data
Z
X

Minimum-error formulation

- Introduce a complete orthonormal set of D-dimensional basis vector $\{\mathbf{u}_i\}$ where i=1,…,D that satisfy $\mathbf{u}^T_i\mathbf{u}_j = \delta_{ij}$.
- For a data set of observations $\{\mathbf{x}_n\}$, we have $\mathbf{x}_n = \sum_{i=1}^{D} \alpha_{ni}\mathbf{u}_i \Rightarrow \mathbf{x}_n = \sum_{i=1}^{D} \left(\mathbf{x}_n^T\mathbf{u}_i\right)\mathbf{u}_i$

- Goal: *approximate* $\mathbf{x}_n$ *using a representation involving a restricted number M<D of variables corresponding to a projection onto a lower-dimensional.*

$$\widetilde{\mathbf{x}}_n = \sum_{i=1}^{M} z_{ni}\mathbf{u}_i + \sum_{i=M+1}^{D} b_i\mathbf{u}_i$$

Minimizing it.

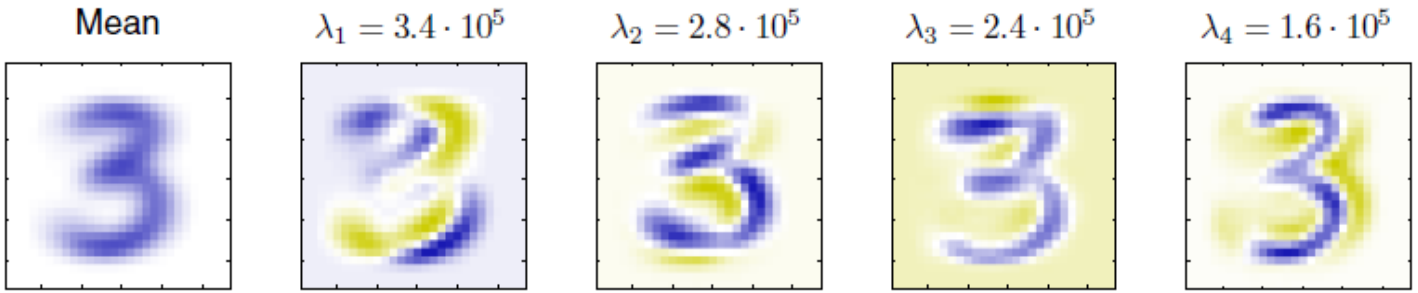
$$\Rightarrow J = \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{x}_n - \widetilde{\mathbf{x}}_n\|^2 \Rightarrow \begin{aligned} z_{nj} &= \mathbf{x}_n^T\mathbf{u}_j \\ b_j &= \overline{\mathbf{x}}^T\mathbf{u}_j \end{aligned}$$

$$\Rightarrow \mathbf{x}_n - \widetilde{\mathbf{x}}_n = \sum_{i=M+1}^{D}\left\{(\mathbf{x}_n - \overline{\mathbf{x}})^T\mathbf{u}_i\right\}\mathbf{u}_i \Rightarrow J = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D}\left(\mathbf{x}_n^T\mathbf{u}_i - \overline{\mathbf{x}}^T\mathbf{u}_i\right)^2 = \sum_{i=M+1}^{D}\mathbf{u}_i^T\mathbf{S}\mathbf{u}_i$$

$$\Rightarrow \widetilde{J} = \mathbf{u}_2^T\mathbf{S}\mathbf{u}_2 + \lambda_2\left(1 - \mathbf{u}_2^T\mathbf{u}_2\right) \Rightarrow \mathbf{S}\mathbf{u}_i = \lambda_i\mathbf{u}_i \Rightarrow J = \sum_{i=M+1}^{D}\lambda_i$$

Applications of PCA

- Data compression (offline digits data set):
  - First four eigenvectors and eigenvalues:



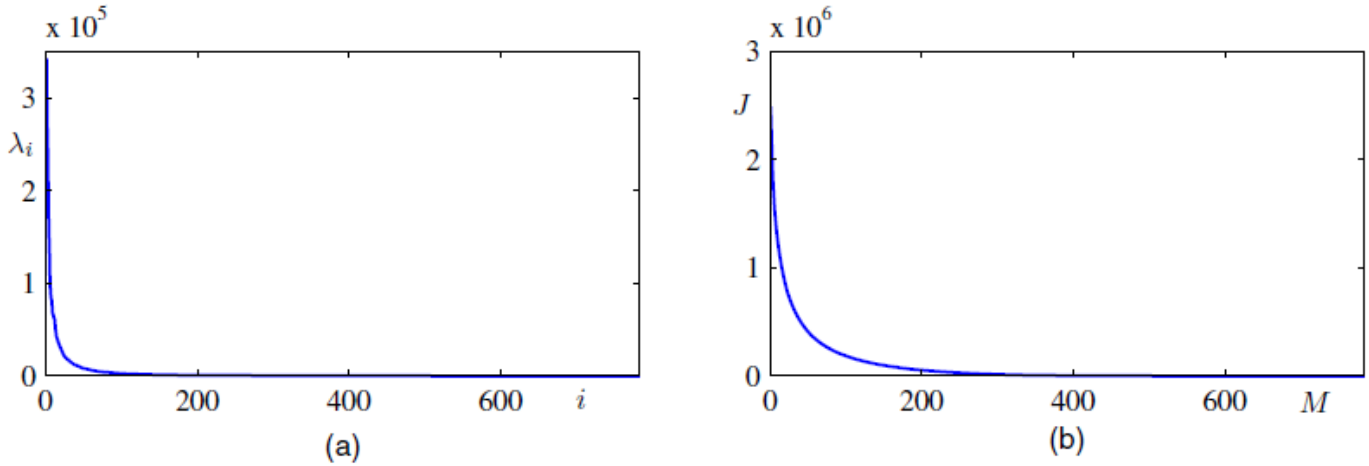
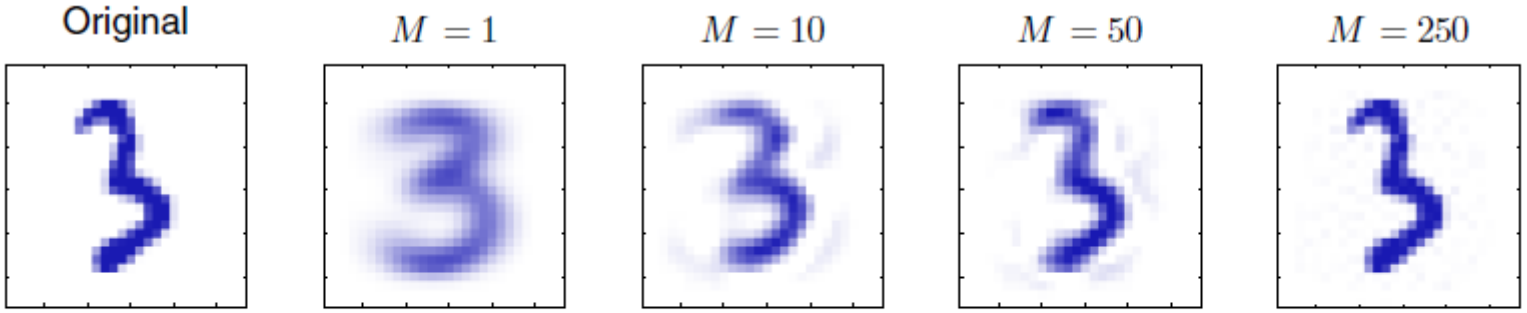


Figure 12.4 (a) Plot of the eigenvalue spectrum for the off-line digits data set. (b) Plot of the sum of the discarded eigenvalues, which represents the sum-of-squares distortion J introduced by projecting the data onto a principal component subspace of dimensionality M.

Applications of PCA

- Data compression (offline digits data set):

$$\widetilde{\mathbf{x}}_n = \sum_{i=1}^{M} (\mathbf{x}_n^{\mathrm{T}}\mathbf{u}_i)\mathbf{u}_i + \sum_{i=M+1}^{D} (\overline{\mathbf{x}}^{\mathrm{T}}\mathbf{u}_i)\mathbf{u}_i = \overline{\mathbf{x}} + \sum_{i=1}^{M} \left(\mathbf{x}_n^{\mathrm{T}}\mathbf{u}_i - \overline{\mathbf{x}}^{\mathrm{T}}\mathbf{u}_i\right)\mathbf{u}_i$$



An original example from the off-line digits data set together with its PCA reconstructions obtained by retaining M principal components for various values of M. As M increases the reconstruction becomes more accurate and would become perfect when $M = D = 28 \times 28 = 784$.

Applications of PCA

- Data normalization (pre-processing):

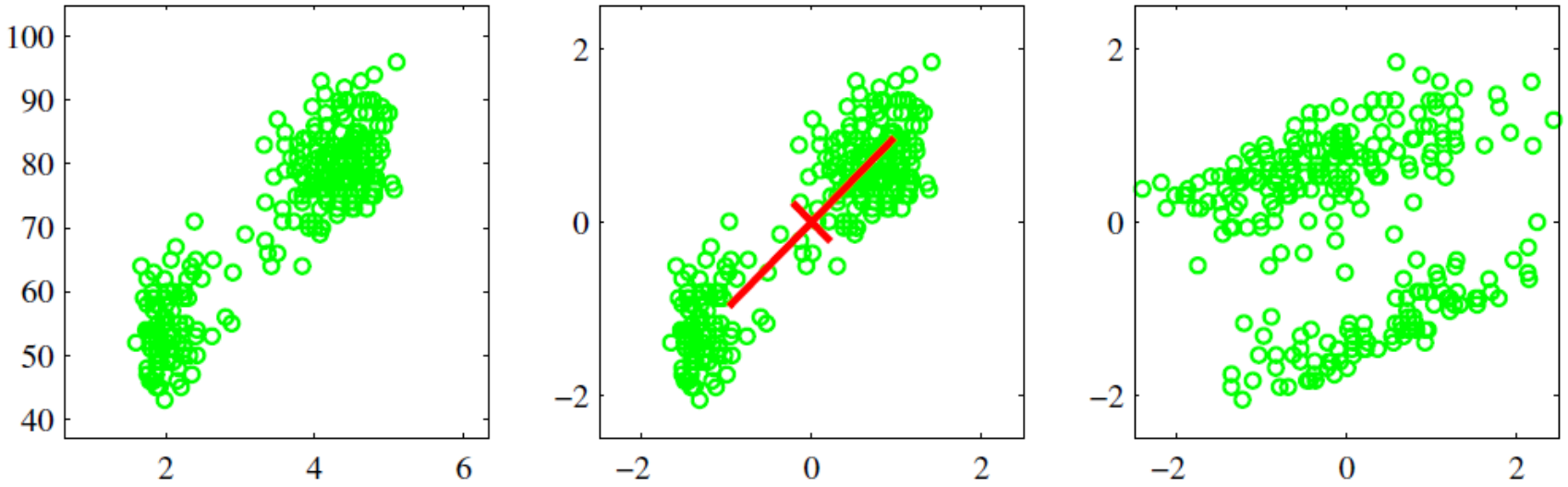
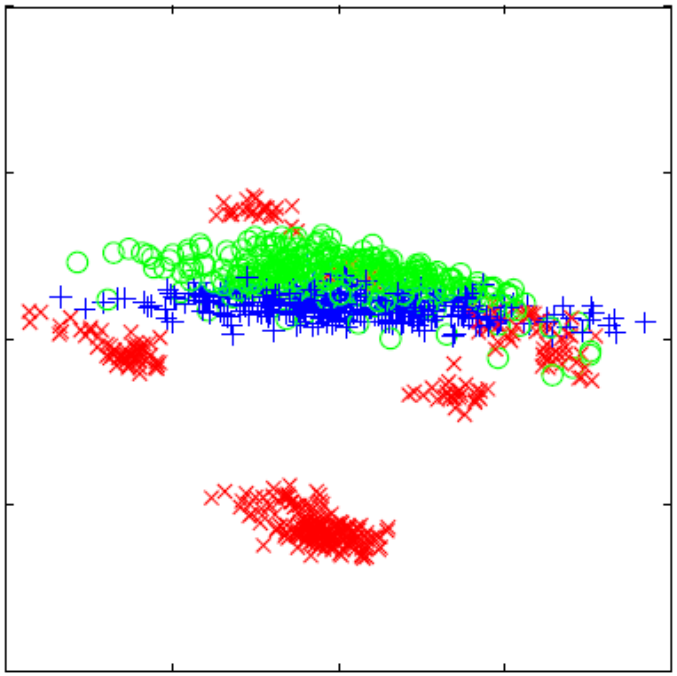


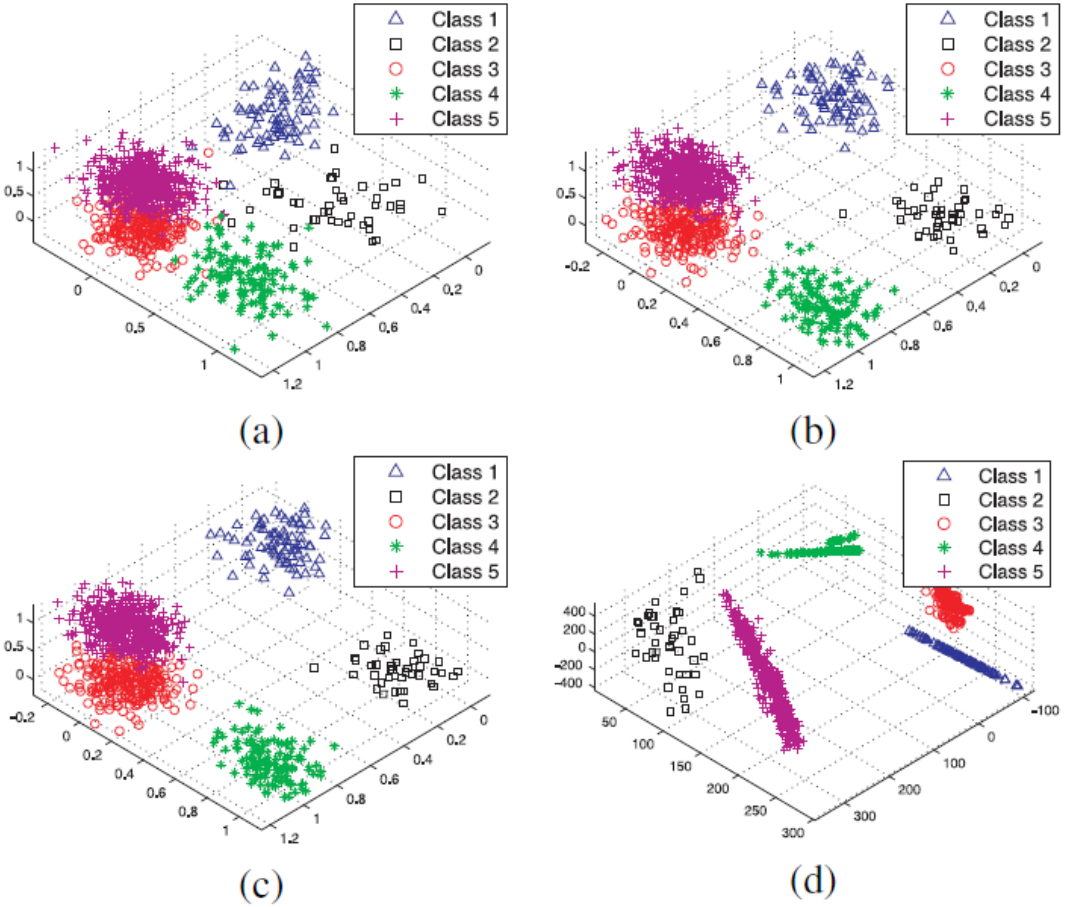
Figure 12.6 Illustration of the effects of linear pre-processing applied to the Old Faithful data set. The plot on the left shows the original data. The centre plot shows the result of standardizing the individual variables to zero mean and unit variance. Also shown are the principal axes of this normalized data set, plotted over the range $\pm\lambda_i^{1/2}$. The plot on the right shows the result of whitening of the data to give it zero mean and unit covariance.

Applications of PCA

- Data visualization:

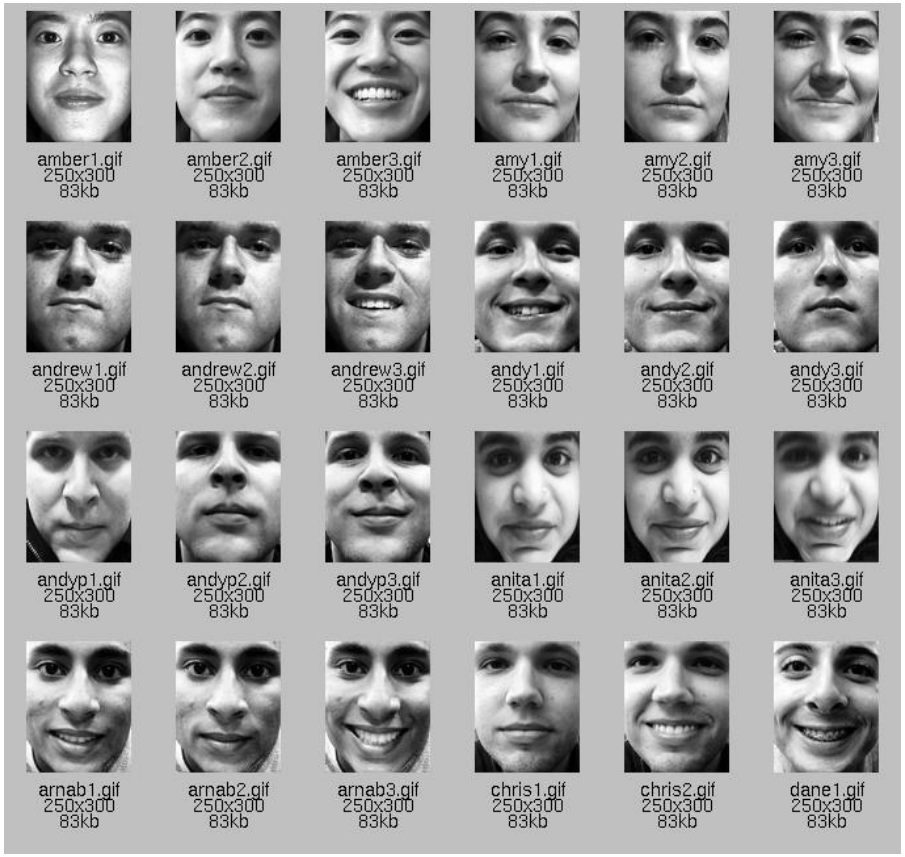


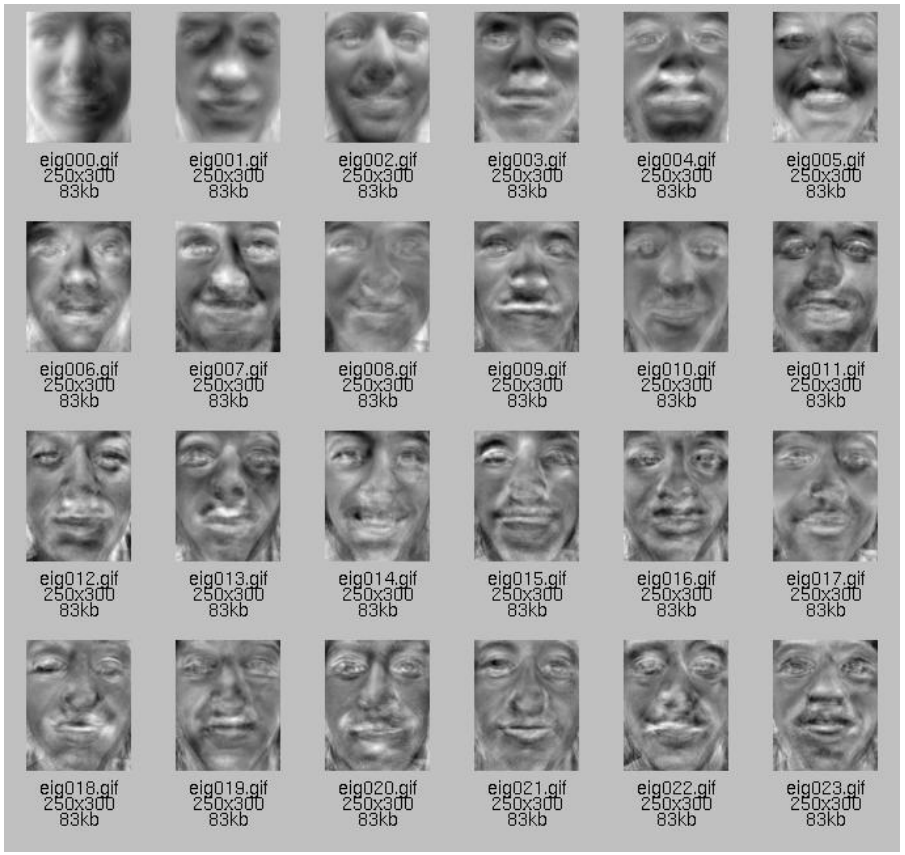
Visualization of the oil flow data set obtained by projecting the data onto the first two principal components. The red, blue, and green points correspond to the 'laminar', 'homogeneous', and 'annular' flow configurations respectively.



Applications of PCA

- Face recognition (eigenface):



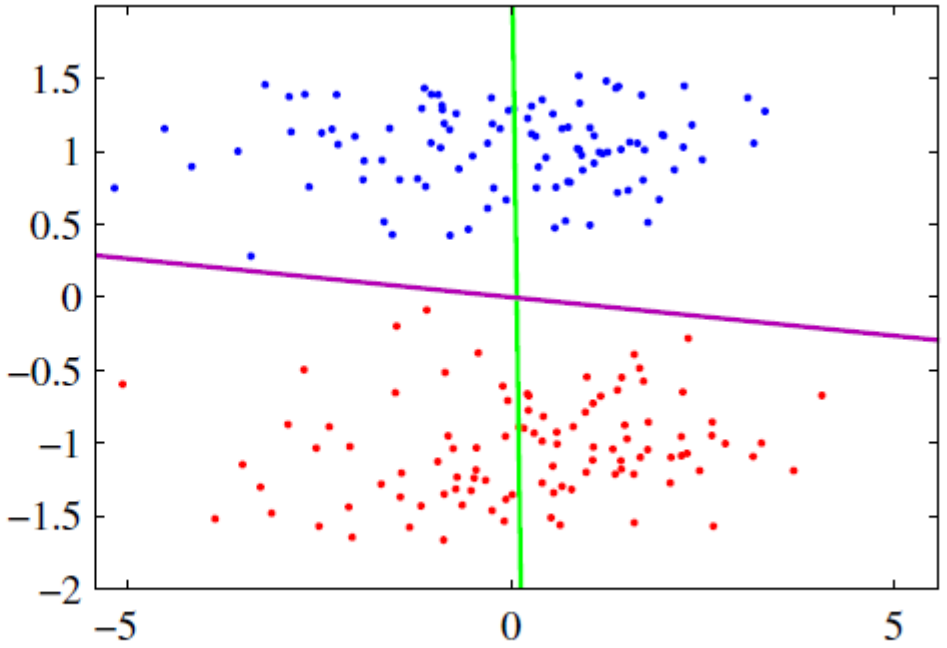


http://www.clear.rice.edu/elec301/Projects99/faces/images.html

Applications of PCA

- PCA vs. Fisher's LDA:

A comparison of principal component analysis with Fisher's linear discriminant for linear dimensionality reduction. Here the data in two dimensions, belonging to two classes shown in red and blue, is to be projected onto a single dimension. PCA chooses the direction of maximum variance, shown by the magenta curve, which leads to strong class overlap, whereas the Fisher linear discriminant takes account of the class labels and leads to a projection onto the green curve giving much better class separation.



http://www.face-rec.org/algorithms/PCA/jcn.pdf
http://www.cs.jhu.edu/~hager/Public/teaching/cs461/pami97-eigenfaces.pdf

Probabilistic PCA

Why Probabilistic PCA?

- Probabilistic PCA vs. conventional PCA:
  - We can derive an EM algorithm for PCA that is computationally efficient in situations where only a few leading eigenvectors are required and that avoids having to evaluate the data covariance matrix as an intermediate step.
  - The combination of a probabilistic model and EM allows us to deal with missing values in the data set.
  - Probabilistic PCA forms the basis for a Bayesian treatment of PCA in which the dimensionality of the principal subspace can be found automatically from the data.
  - The existence of a likelihood function allows direct comparison with other probabilistic density models.
  - Probabilistic PCA can be used to model class-conditional densities and hence be applied to classification problems.
  - The probabilistic PCA model can be run generatively to provide samples from the distribution.

Probabilistic PCA

- PCA can also be expressed as the maximum likelihood solution of a probabilistic latent variable model.

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) \qquad p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \qquad \mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

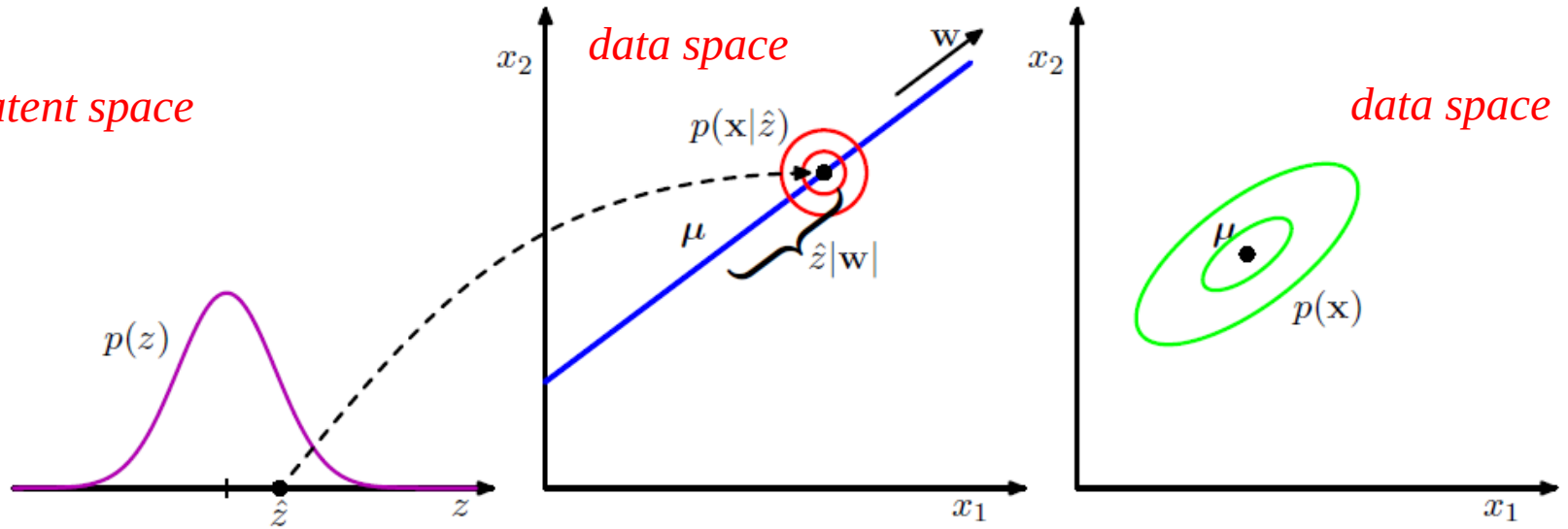


Figure 12.9 An illustration of the generative view of the probabilistic PCA model for a two-dimensional data space and a one-dimensional latent space. An observed data point $\mathbf{x}$ is generated by first drawing a value $\hat{z}$ for the latent variable from its prior distribution $p(z)$ and then drawing a value for $\mathbf{x}$ from an isotropic Gaussian distribution (illustrated by the red circles) having mean $\mathbf{w}\hat{z} + \boldsymbol{\mu}$ and covariance $\sigma^2\mathbf{I}$. The green ellipses show the density contours for the marginal distribution $p(\mathbf{x})$.

Probabilistic PCA

- PCA can also be expressed as the maximum likelihood solution of a probabilistic latent variable model.

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) \qquad p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \qquad \mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z} \Rightarrow p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) \qquad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$$

$$\mathbf{C}^{-1} = \sigma^{-1}\mathbf{I} - \sigma^{-2}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}$$

$$\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}$$

$$\Rightarrow \begin{aligned} \mathbb{E}[\mathbf{x}] &= \mathbb{E}[\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu} \\ \mathrm{cov}[\mathbf{x}] &= \mathbb{E}\left[(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})^{\mathrm{T}}\right] \\ &= \mathbb{E}\left[\mathbf{W}\mathbf{z}\mathbf{z}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\right] + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}] = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I} \end{aligned}$$

$$\Rightarrow p(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}), \sigma^{-2}\mathbf{M}\right)$$

Maximum likelihood PCA

- We have: $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{C}) \qquad \mathbf{C} = \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I}$
- Then,

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\mathbf{W}, \boldsymbol{\mu}, \sigma^2)$$

$$= -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\mathbf{C}| - \frac{1}{2}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$$

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}}$$

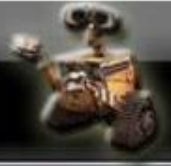
$$\boldsymbol{\mu} = \overline{\mathbf{x}}$$

$$\ln p(\mathbf{X}|\mathbf{W}, \boldsymbol{\mu}, \sigma^2) = -\frac{N}{2}\left\{D\ln(2\pi) + \ln|\mathbf{C}| + \mathrm{Tr}\left(\mathbf{C}^{-1}\mathbf{S}\right)\right\}$$

- *Solving* W_{ML}: $\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_M(\mathbf{L}_M - \sigma^2\mathbf{I})^{1/2}\mathbf{R}$

R *is an arbitrary M×M orthogonal matrix*

- *Solving* σ_{ML}: $\sigma^2_{\mathrm{ML}} = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i$
- *For M = D:* $\mathbf{C} = \mathbf{U}(\mathbf{L} - \sigma^2\mathbf{I})^{1/2}\mathbf{R}\mathbf{R}^{\mathrm{T}}(\mathbf{L} - \sigma^2\mathbf{I})^{1/2}\mathbf{U}^{\mathrm{T}} + \sigma^2\mathbf{I} = \mathbf{U}\mathbf{L}\mathbf{U}^{\mathrm{T}} = \mathbf{S}$



EM algorithm for PCA

- We have:

$$\ln p\left(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2\right) = \sum_{n=1}^{N} \{\ln p(\mathbf{x}_n|\mathbf{z}_n) + \ln p(\mathbf{z}_n)\}$$

$$\mathbb{E}[\ln p\left(\mathbf{X}, \mathbf{Z}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2\right)] = -\sum_{n=1}^{N} \left\{ \frac{D}{2} \ln(2\pi\sigma^2) + \frac{1}{2} \mathrm{Tr}\left(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^{\mathrm{T}}]\right) + \frac{1}{2\sigma^2} \|\mathbf{x}_n - \boldsymbol{\mu}\|^2 - \frac{1}{\sigma^2} \mathbb{E}[\mathbf{z}_n]^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} (\mathbf{x}_n - \boldsymbol{\mu}) + \frac{1}{2\sigma^2} \mathrm{Tr}\left(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^{\mathrm{T}}] \mathbf{W}^{\mathrm{T}} \mathbf{W}\right) \right\}$$

- E Step:

$$\begin{aligned} \mathbb{E}[\mathbf{z}_n] &= \mathbf{M}^{-1} \mathbf{W}^{\mathrm{T}} (\mathbf{x}_n - \overline{\mathbf{x}}) \\ \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^{\mathrm{T}}] &= \sigma^2 \mathbf{M}^{-1} + \mathbb{E}[\mathbf{z}_n] \mathbb{E}[\mathbf{z}_n]^{\mathrm{T}} \end{aligned}$$

- M Step:

$$\mathbf{W}_{\mathrm{new}} = \left[\sum_{n=1}^{N} (\mathbf{x}_n - \overline{\mathbf{x}}) \mathbb{E}[\mathbf{z}_n]^{\mathrm{T}} \right] \left[\sum_{n=1}^{N} \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^{\mathrm{T}}] \right]^{-1}$$

$$\sigma^2_{\mathrm{new}} = \frac{1}{ND} \sum_{n=1}^{N} \left\{ \|\mathbf{x}_n - \overline{\mathbf{x}}\|^2 - 2\mathbb{E}[\mathbf{z}_n]^{\mathrm{T}} \mathbf{W}_{\mathrm{new}}^{\mathrm{T}} (\mathbf{x}_n - \overline{\mathbf{x}}) + \mathrm{Tr}\left(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^{\mathrm{T}}] \mathbf{W}_{\mathrm{new}}^{\mathrm{T}} \mathbf{W}_{\mathrm{new}}\right) \right\}$$

Kernel PCA

Kernel PCA

- Consider a data set $\{\mathbf{x}_n\}$ of observations, where $n = 1, \ldots, N$, in a space of dimensionality D.

$$\sum_n \mathbf{x}_n = 0 \quad \Longrightarrow \quad \mathbf{S}\mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad \mathbf{S} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \qquad \mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i = 1$$

- Consider a nonlinear transformation $\varphi(\mathbf{x})$ into an M-dimensional feature space, so that each data point $\mathbf{x}_n$ is thereby projected onto a point $\varphi(\mathbf{x}_n)$.

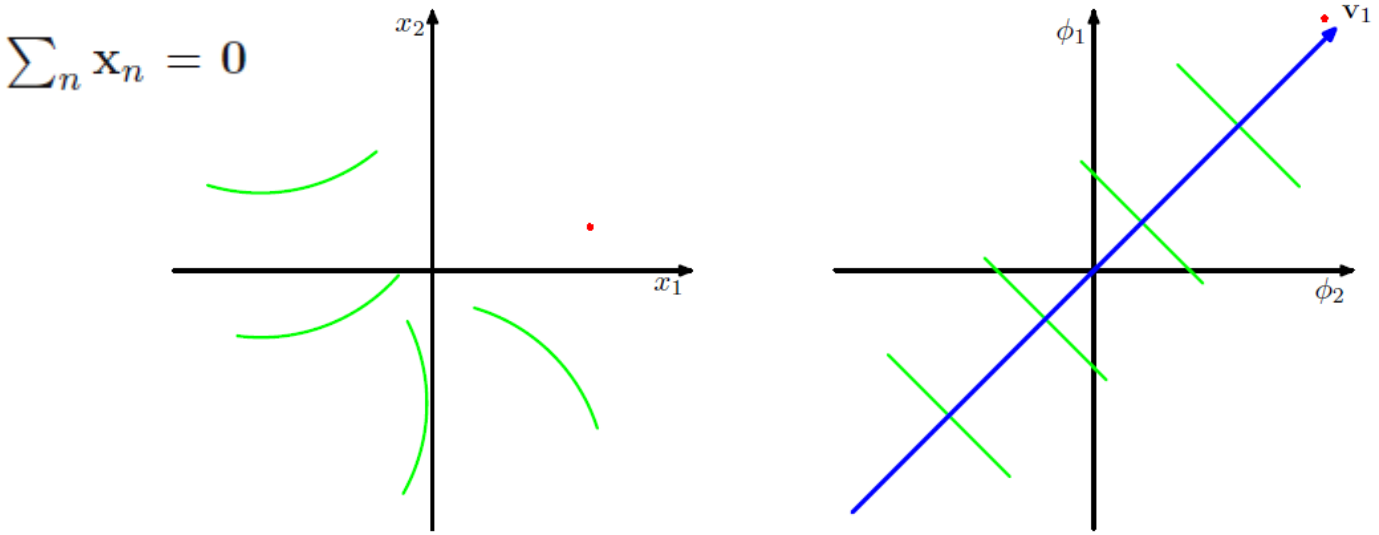


Figure 12.16 Schematic illustration of kernel PCA. A data set in the original data space (left-hand plot) is projected by a nonlinear transformation $\phi(\mathbf{x})$ into a feature space (right-hand plot). By performing PCA in the feature space, we obtain the principal components, of which the first is shown in blue and is denoted by the

Kernel PCA

- In the M-dimensional feature space:

$$\sum_n \phi(\mathbf{x}_n) = \mathbf{0} \qquad \mathbf{C} = \frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}} \qquad \mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i \qquad i = 1, \ldots, M$$

$$\rightarrow \quad \frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{x}_n)\left\{\phi(\mathbf{x}_n)^{\mathrm{T}}\mathbf{v}_i\right\} = \lambda_i \mathbf{v}_i \quad \rightarrow \quad \mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n)$$

- Substituting $\mathbf{v}_i$ back into $\mathbf{C}$:

$$\frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{x}_n)\phi(\mathbf{x}_n)^{\mathrm{T}} \sum_{m=1}^{N} a_{im}\phi(\mathbf{x}_m) = \lambda_i \sum_{n=1}^{N} a_{in}\phi(\mathbf{x}_n)$$

- Multiplying both sides by $\varphi(\mathbf{x}_l)^{\mathrm{T}}$:

$$\xrightarrow{k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m)} \quad \frac{1}{N}\sum_{n=1}^{N} k(\mathbf{x}_l, \mathbf{x}_n) \sum_{m=1}^{m} a_{im}k(\mathbf{x}_n, \mathbf{x}_m) = \lambda_i \sum_{n=1}^{N} a_{in}k(\mathbf{x}_l, \mathbf{x}_n)$$

$$\rightarrow \quad \mathbf{K}^2\mathbf{a}_i = \lambda_i N \mathbf{K}\mathbf{a}_i \quad \rightarrow \quad \mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i$$

$$\rightarrow \quad 1 = \mathbf{v}_i^{\mathrm{T}}\mathbf{v}_i = \sum_{n=1}^{N}\sum_{m=1}^{N} a_{in}a_{im}\phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m) = \mathbf{a}_i^{\mathrm{T}}\mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i^{\mathrm{T}}\mathbf{a}_i$$

$$\rightarrow \quad y_i(\mathbf{x}) = \phi(\mathbf{x})^{\mathrm{T}}\mathbf{v}_i = \sum_{n=1}^{N} a_{in}\phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{x}_n) = \sum_{n=1}^{N} a_{in}k(\mathbf{x}, \mathbf{x}_n)$$

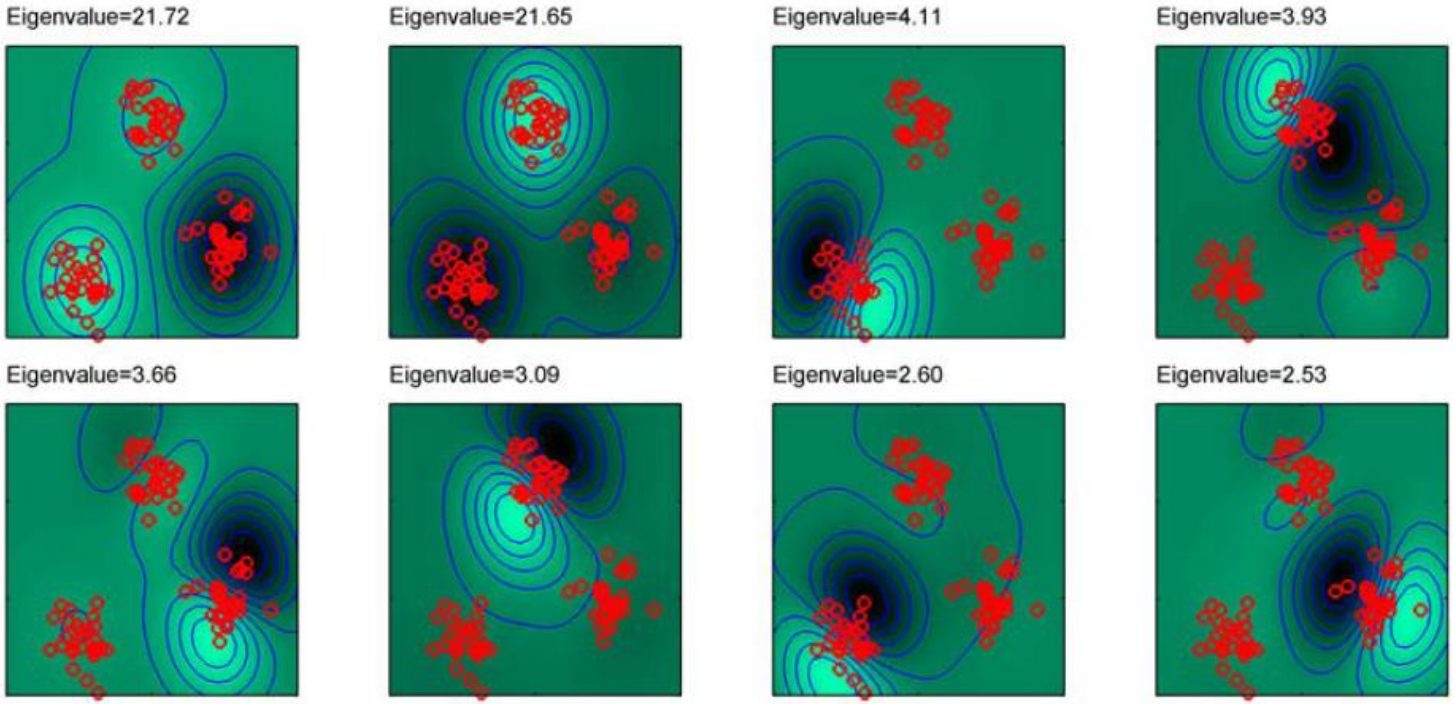
Kernel PCA

- When has not zero mean in the feature space:

$$\widetilde{\phi}(\mathbf{x}_n) = \phi(\mathbf{x}_n) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathbf{x}_l)$$

$$\begin{aligned}\widetilde{K}_{nm} &= \widetilde{\phi}(\mathbf{x}_n)^{\mathrm{T}}\widetilde{\phi}(\mathbf{x}_m) \\ &= \phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_l) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathbf{x}_l)^{\mathrm{T}}\phi(\mathbf{x}_m) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}\phi(\mathbf{x}_j)^{\mathrm{T}}\phi(\mathbf{x}_l) \\ &= k(\mathbf{x}_n,\mathbf{x}_m) - \frac{1}{N}\sum_{l=1}^{N}k(\mathbf{x}_l,\mathbf{x}_m) - \frac{1}{N}\sum_{l=1}^{N}k(\mathbf{x}_n,\mathbf{x}_l) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}k(\mathbf{x}_j,\mathbf{x}_l).\end{aligned}$$

$$\widetilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N \qquad k(\mathbf{x},\mathbf{x}') = \exp(-\|\mathbf{x}-\mathbf{x}'\|^2/0.1)$$



Nonlinear Latent Variable Models

(*Modeling nonlinear manifolds / manifold learning*)

Modeling nonlinear manifolds

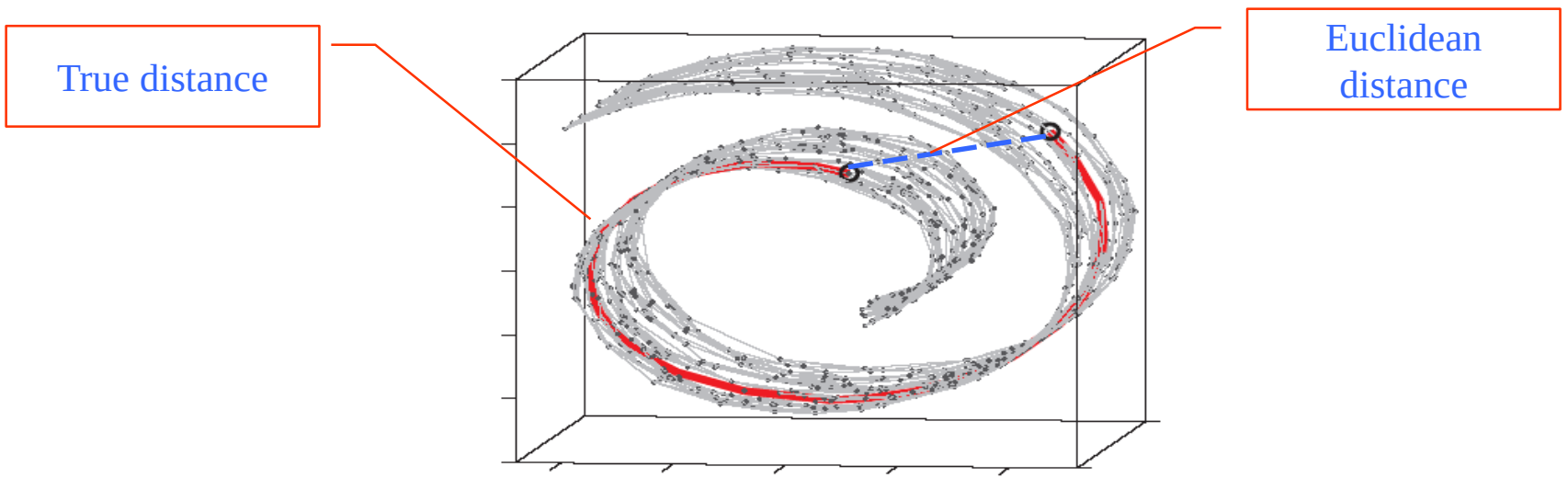
- Two nonprobabilistic methods for dimensionality reduction and data visualization:
  - *Isometric feature mapping (ISOMAP): global method*
    - project the data to a lower-dimensional space using MDS, but where the dissimilarities are defined in terms of the *geodesic distances* measured along the manifold.
  - *Locally linear embedding (LLE):local method*
    - Map the high-dimensional data points down to a lower dimensional space while preserving coefficients.

---

1. *Tenenbaum J.B., Silva V. De , Langford J. C., A global geometric framework for nonlinear dimensionality reduction, Science, 2000, 290 (5500): 2219-2323*
2. *Sam Roweis,Lawrence Saul,Nonlinear dimensionality reduction by locally linear embedding, Science, 2000,290(5500):2323-2326*

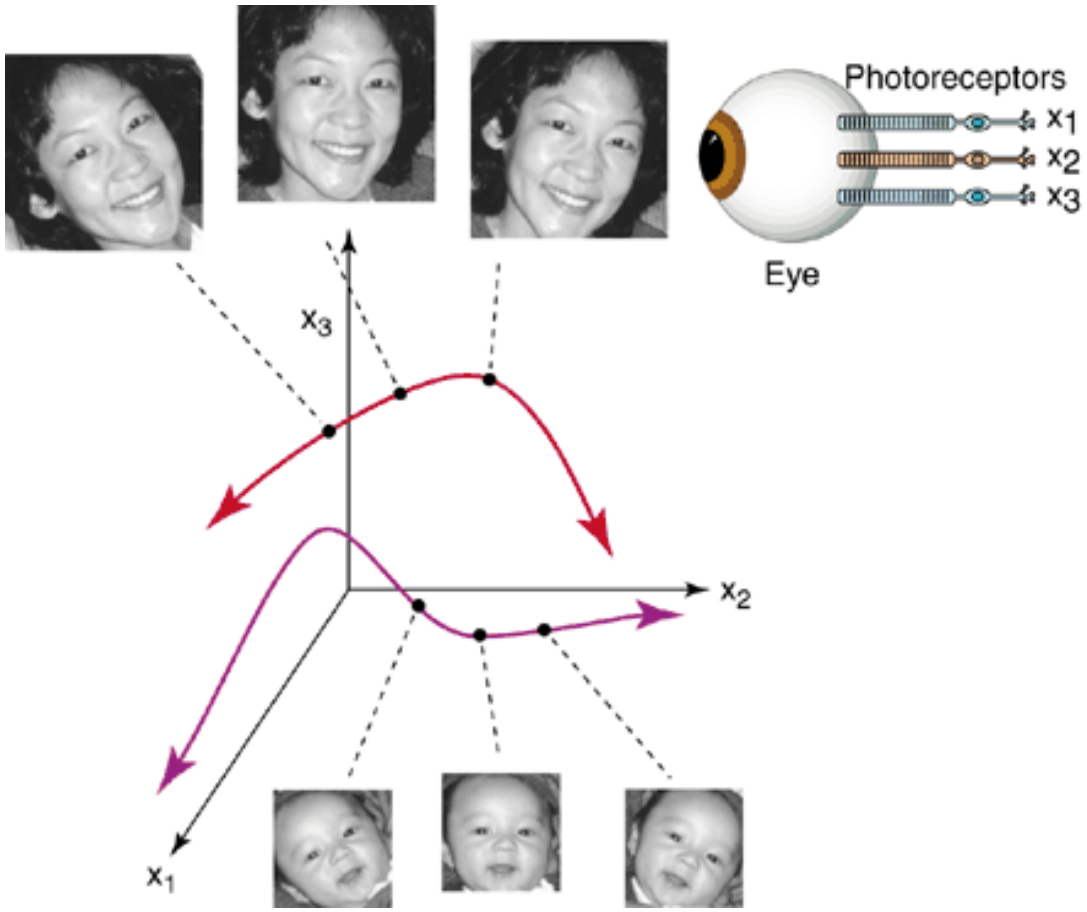
Problem description

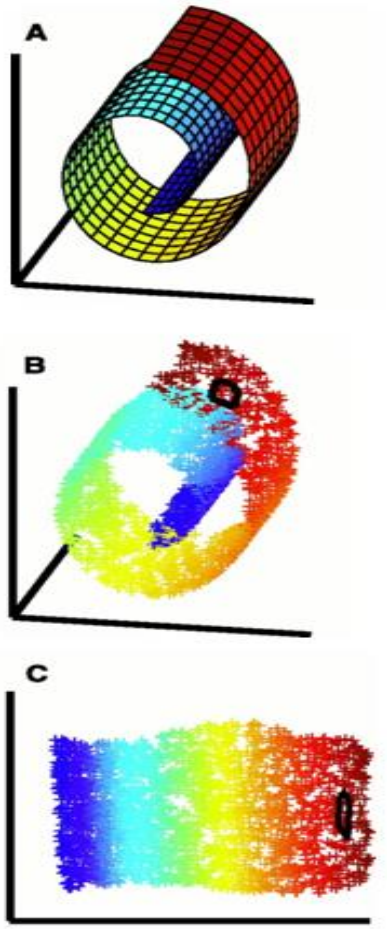
- Given a set $\mathbf{x}_1, \ldots, \mathbf{x}_k$ of *k* points in $\mathbf{R}^n$, find a set of points $\mathbf{y}_1, \ldots, \mathbf{y}_k$ in $\mathbf{R}^m$ (m << n) such that $\mathbf{y}_i$ "represents" $\mathbf{x}_i$ as accurately as possible.
- If the data $\mathbf{x}_i$ is placed in a *super plane* in high dimensional space, the traditional algorithms, such as PCA, work well.
- However, when the data $\mathbf{x}_i$ is placed in a *nonlinear manifold* in high dimensional space, then the linear models can not work any more.
  - A nonlinear manifold can be roughly understood as a distorted super plane, which may be twisted, folded, or curved.



Manifold examples

- Embed data in a higher dimensional space to a lower dimensional manifold





Isometric feature mapping (ISOMAP)

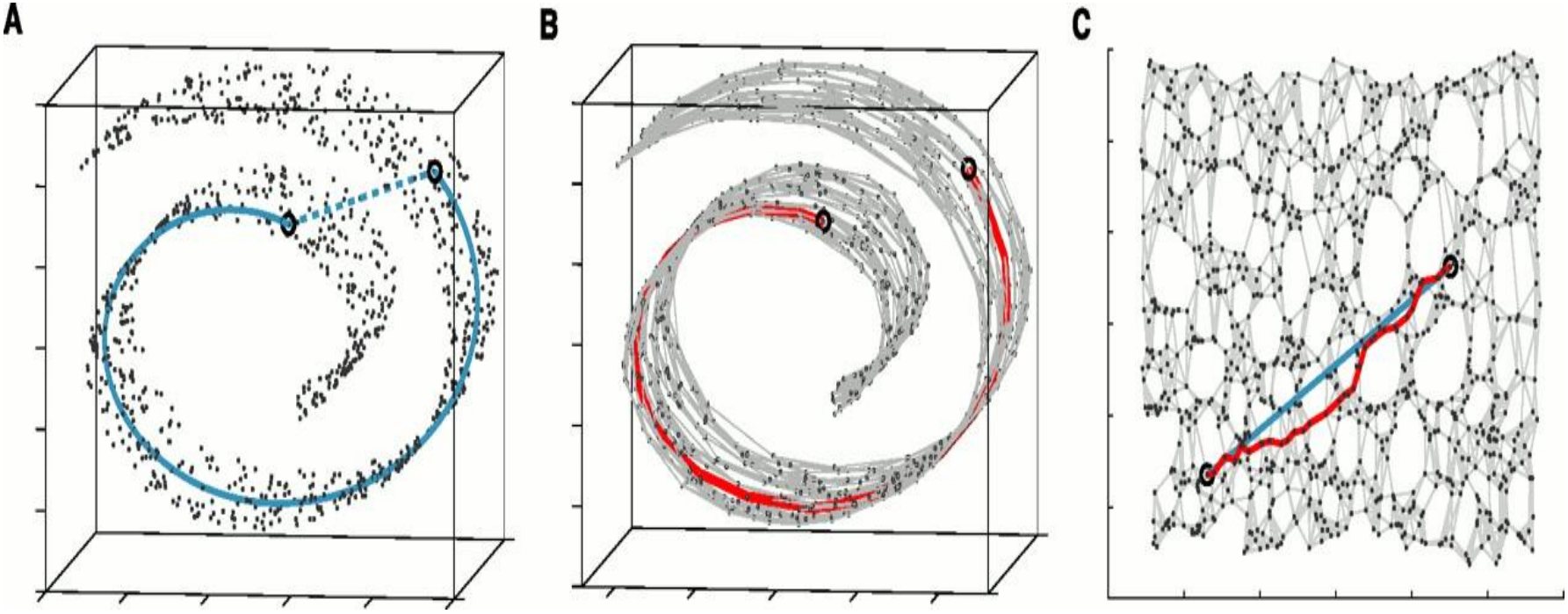
- Idea:
  - Estimate the geodesic distance between faraway points.
  - For neighboring points Euclidean distance is a good approximation to the geodesic distance.
  - For farway points estimate the distance by a series of short hops between neighboring points.
    - Find shortest paths in a graph with edges connecting neighboring data points

Isometric feature mapping (ISOMAP)

- Algorithm:
  - ① Determine the neighbors.
    - All points in a fixed radius.
    - K nearest neighbors
  - ② Construct a neighborhood graph.
    - Each point is connected to the other if it is a K nearest neighbor.
    - Edge Length equals the Euclidean distance
  - ③ Compute the shortest paths between two nodes
    - Floyd's Algorithm
    - Djkastra's ALgorithm
  - ④ Construct a lower dimensional embedding.
    - Classical MDS

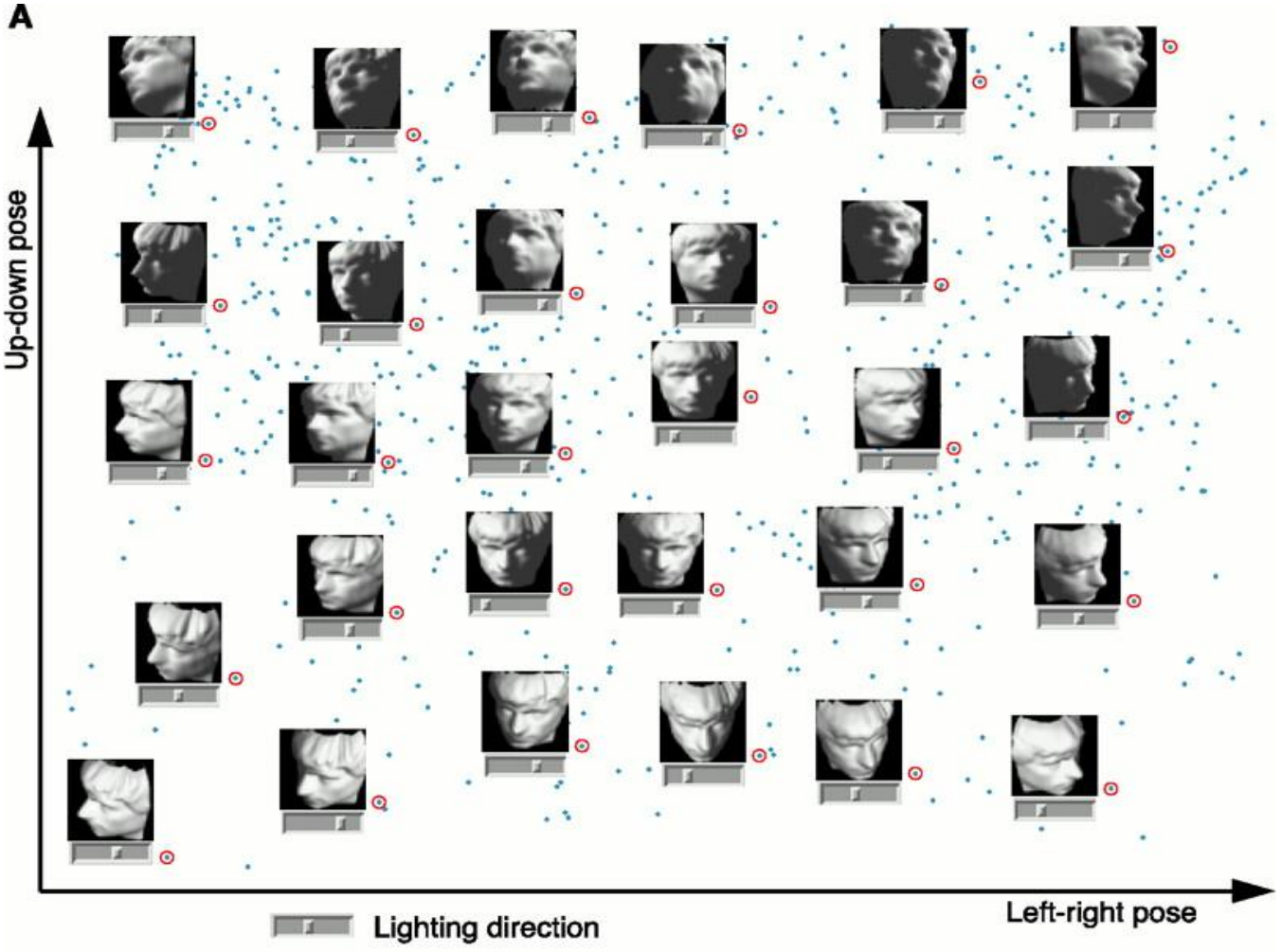
Isometric feature mapping (ISOMAP)

- Algorithm:



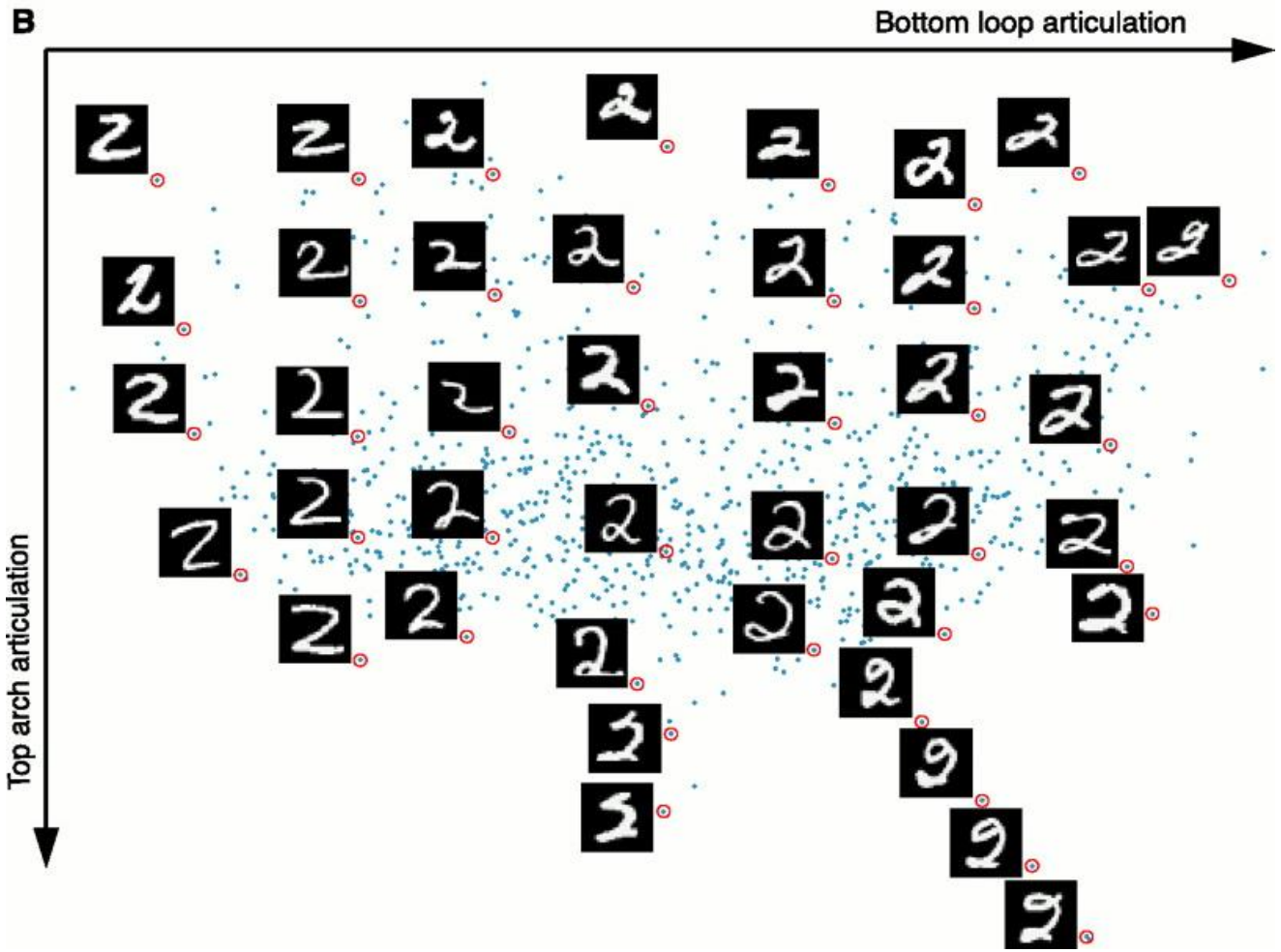
Isometric feature mapping (ISOMAP)

- Results:



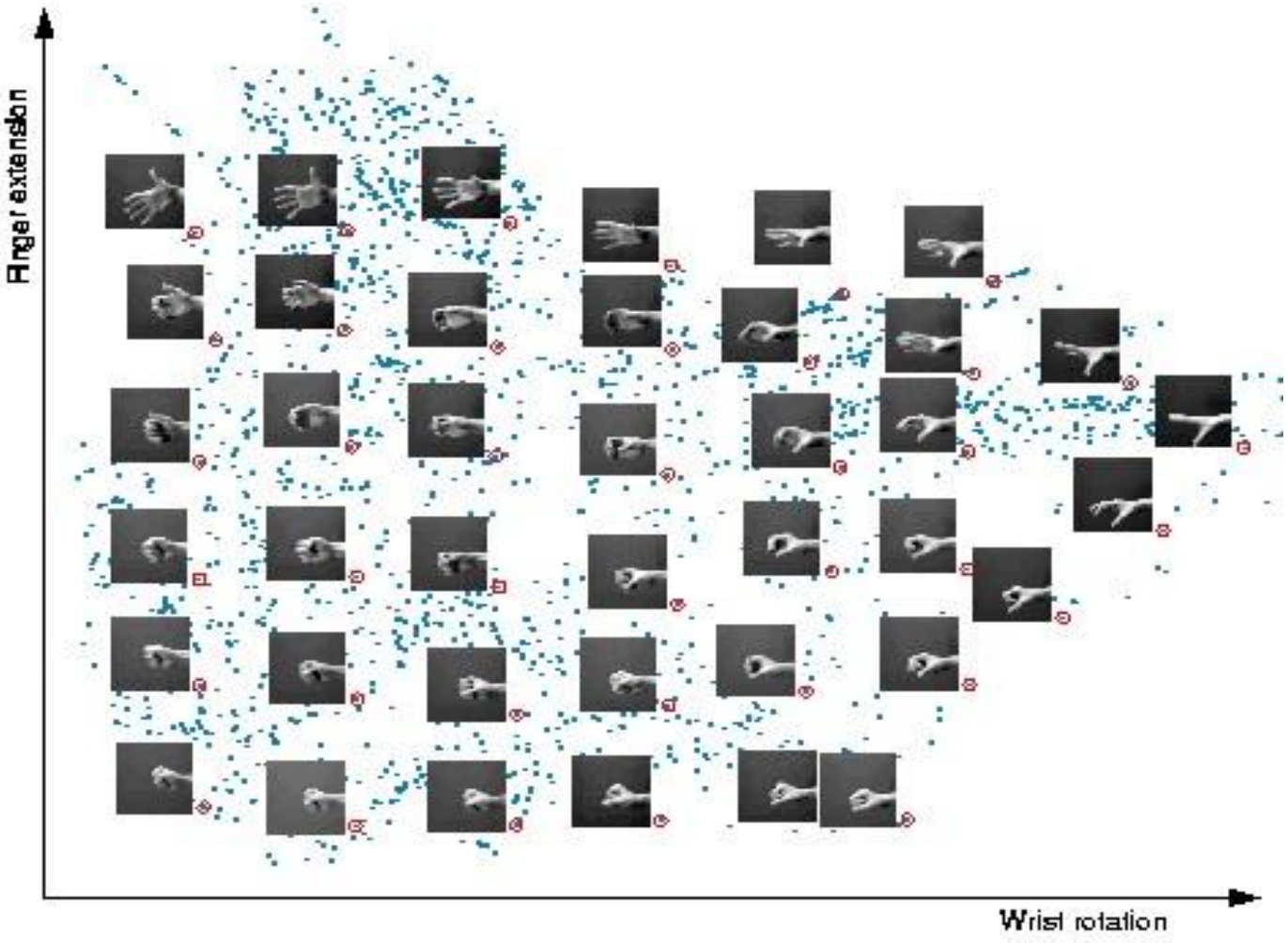
Isometric feature mapping (ISOMAP)

- Results:



Isometric feature mapping (ISOMAP)

- Results:

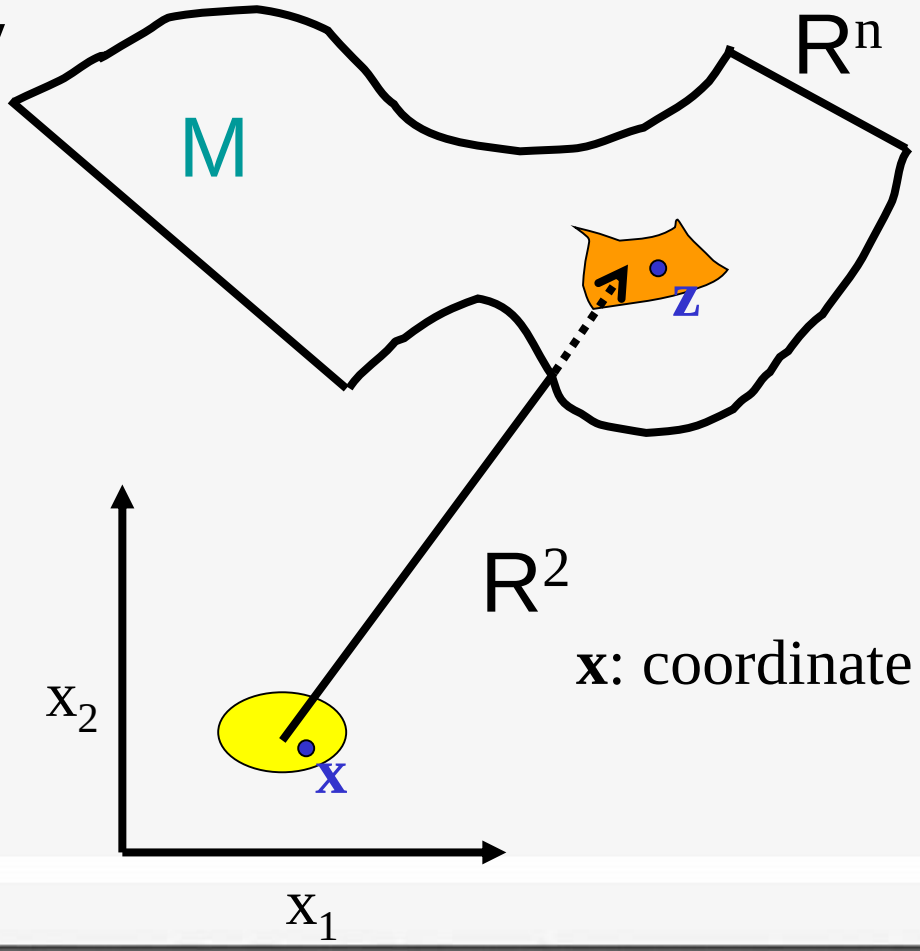


Locally Linear Embedding (LLE)

- Idea:
  - Fit Locally , Think Globally

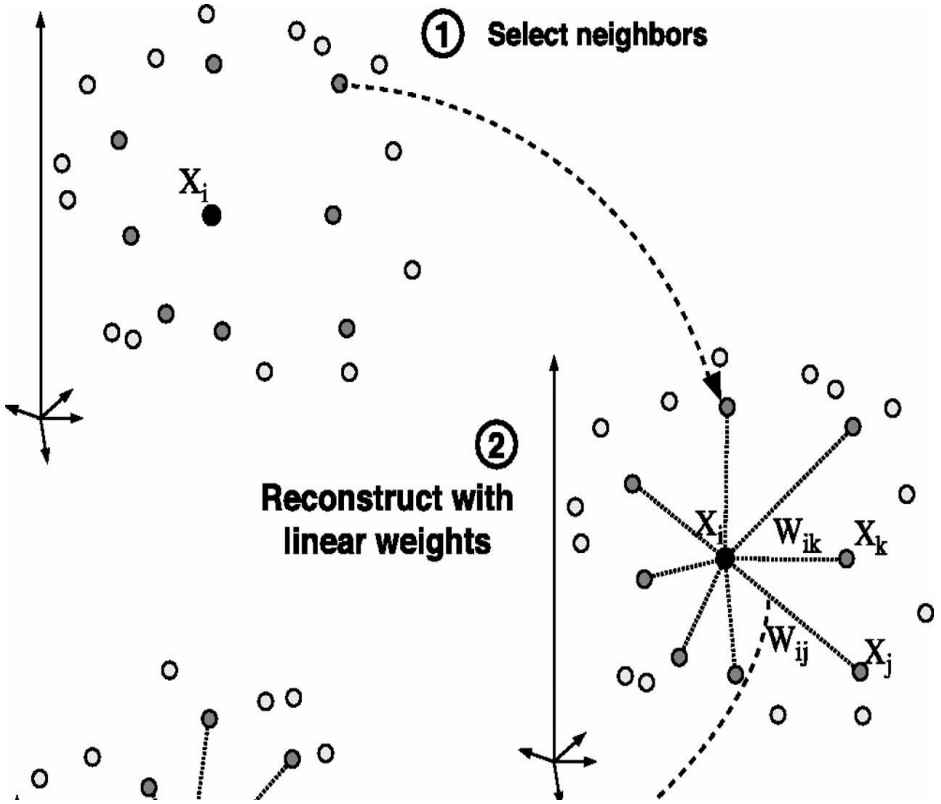
Locally it is a linear patch

Key: how to combine all local patches together?



Locally Linear Embedding (LLE)

- Algorithm:
  - Fit Locally



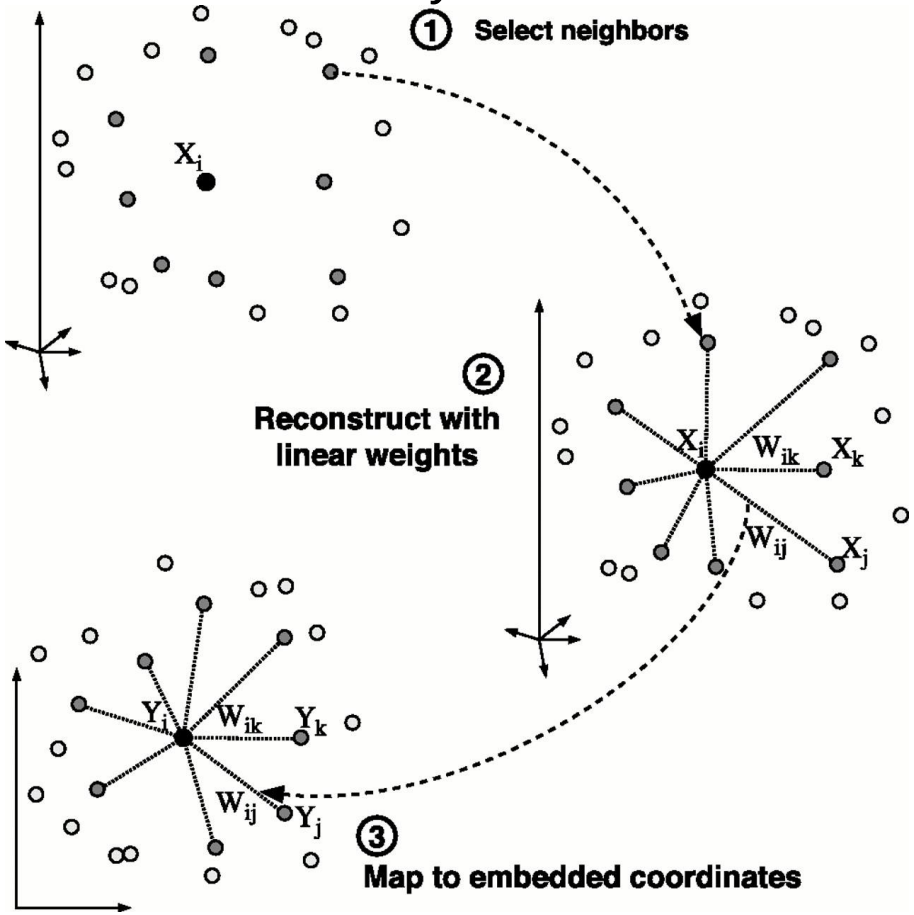
We expect each data point and its neighbors to lie on or close to a locally linear patch of the manifold.

Each point can be written as a linear combination of its neighbors.
The weights chosen to minimize the reconstruction Error.

$$min_W \parallel X_i - \sum_{j=1}^{K} W_{ij} X_j \parallel^2 \qquad (1)$$

Locally Linear Embedding (LLE)

- Algorithm:
  - Think Globally



$$min_W \parallel X_i - \sum_{j=1}^{K} W_{ij} X_j \parallel^2 \qquad (1)$$

Low-dimensional embedding:

$$Y_{d \times N} = [Y_1 | Y_2 | ... | Y_N]$$

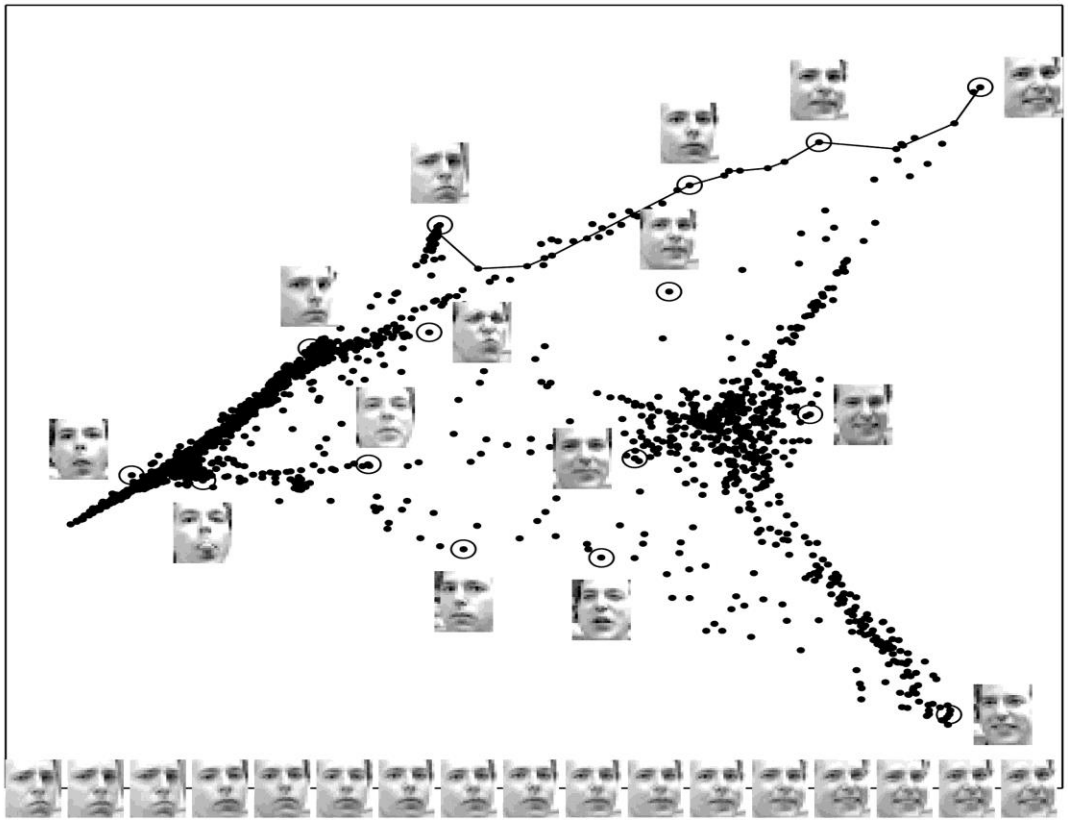
Use the same weights from the original space:

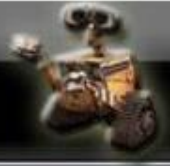
$$min_Y \sum_{i=1}^{N} \parallel Y_i - Y W_i \parallel^2$$

***Y** is given by the eigenvectors of the lowest d non-zero eigenvalues of the matrix* $\mathbf{M}=(\mathbf{I}-\mathbf{W})^{T}(\mathbf{I}-\mathbf{W})$

Locally Linear Embedding (LLE)

- Results: Images of faces mapped into the embedding space described by the first two coordinates of LLE. Representative faces are shown next to circled points. The bottom images correspond to points along the top-right path (linked by solid line) illustrating one particular mode of variability in pose and expression.





Locally Linear Embedding (LLE)

- Limitations:
  - require dense data points on the manifold for good estimation
  - A good neighborhood seems essential to their success
    - How to choose k?
    - Too few neighbors
      - Result in rank deficient tangent space and lead to over-fitting
    - Too many neighbors
      - Tangent space will not match local geometry well

Locally Linear Embedding (LLE)

- Limitations:

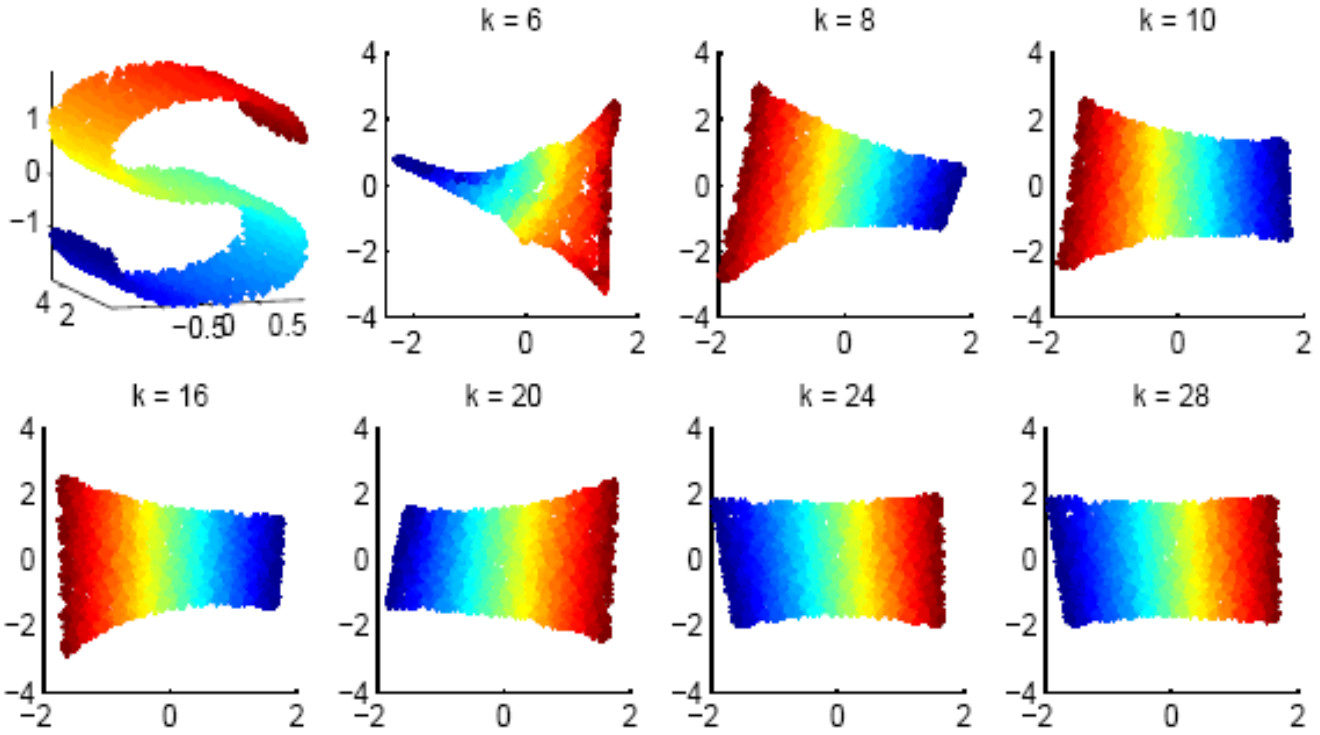
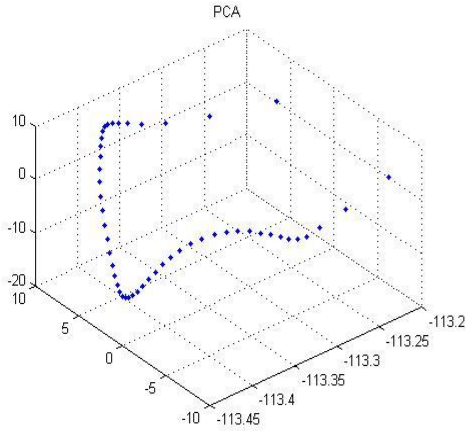


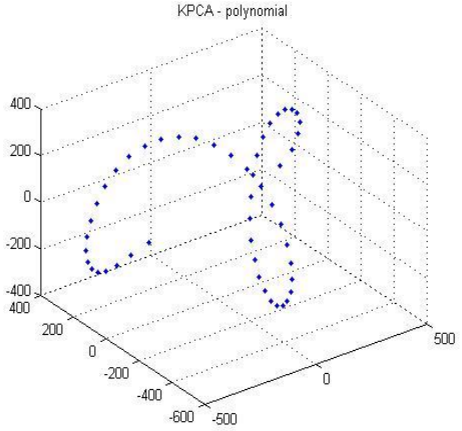
FIG. 5. *S-curve (top left) and computed 2D coordinates by LLE with various neighborhood size k.*

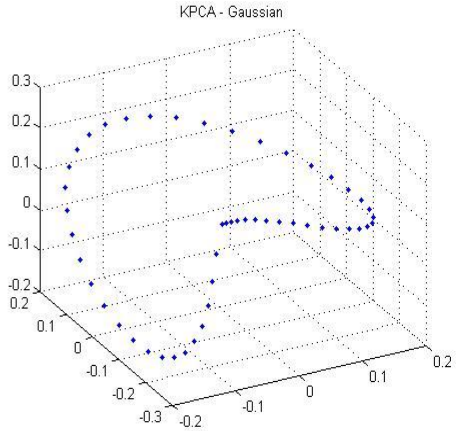
ISOMAP vs. LLE

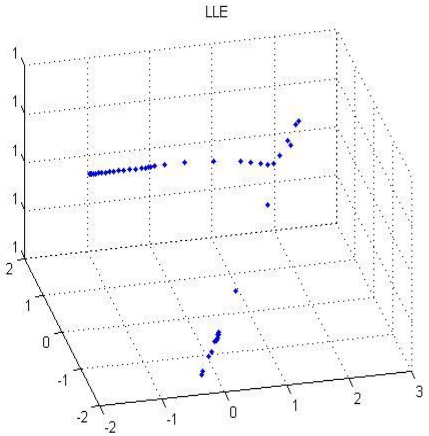
ISOMAP	LLE
Do MDS on the geodesic distance matrix.	Model local neighborhoods as linear a patches and then embed in a lower dimensional manifold.
Global approach	Local aproach
Dynamic programming approaches	Computationally efficient..sparse matrices
Convergence limited by the manifold curvature and number of points.	Good representational capacity

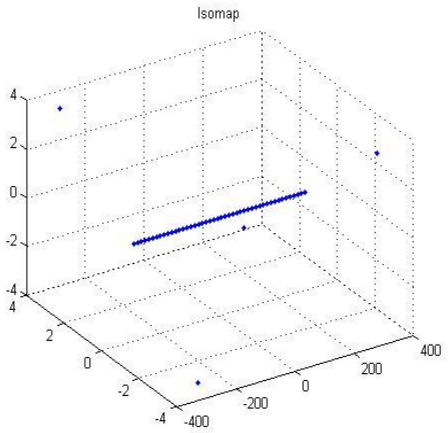
Experimental Results

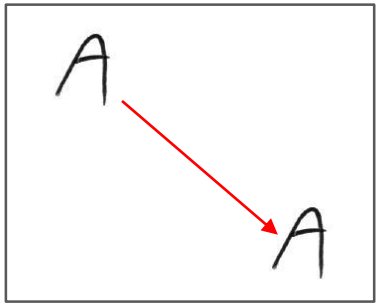












Next: Combining models