Pattern Recognition

Dimensionality Reduction

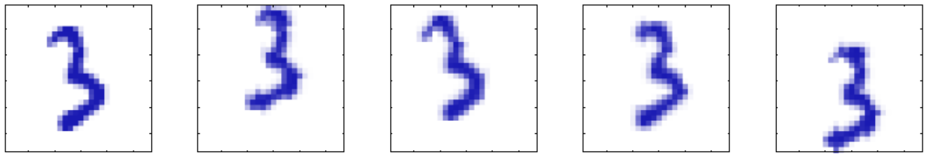
林彥宇 教授
Yen-Yu Lin, Professor

國立陽明交通大學 資訊工程學系
Computer Science, National Yang Ming Chiao Tung University

Some slides are modified from S.-J. Wang and T.-L. Liu

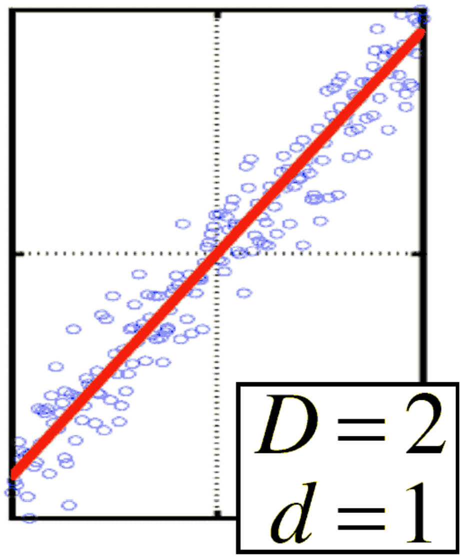
Dimensionality reduction

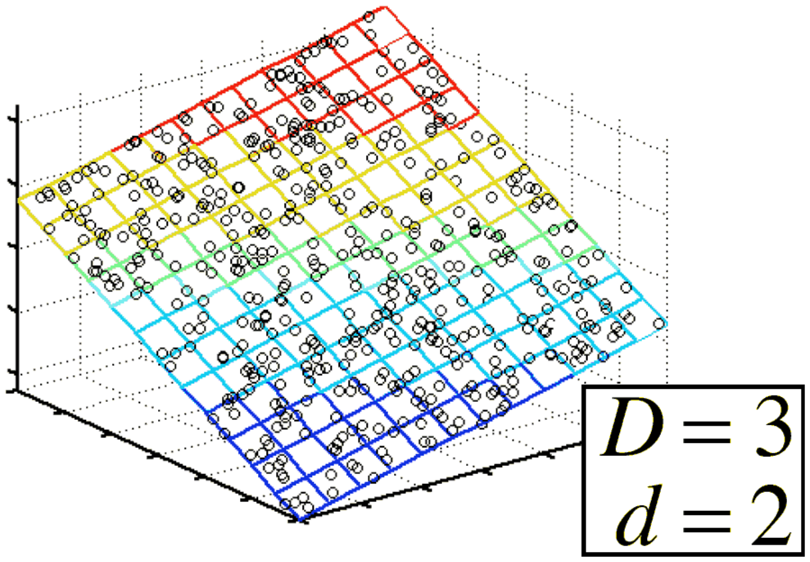
- Many data sets have the property that the data points all lie close to a manifold of much lower dimensionality than that of the original data space
- A synthetic data set obtained by taking one of the off-line digit images and creating multiple copies via random displacements and rotations



  - Dimension of the data space: 100 x 100 = 10,000
  - Degrees of freedom: vertical translation, horizontal translation, rotation
  - Intrinsic dimension: 3

Data dimension vs. intrinsic dimension





Principal component analysis (PCA)

- Principal component analysis (PCA) is one of the most widely used dimensionality reduction techniques
- PCA is used for various applications, such as data compression, feature extraction, and data visualization
- PCA linearly projects data from a high-dimensional input space to a low-dimensional feature space
  - Principal subspace: the lower dimensional space obtained by PCA
  - Principal component: a projection direction/vector found by PCA
  - A projection is composed of one or a few principal components

Definitions of PCA

- Two commonly used definitions:
  - Maximum variance of the projected data (Hotelling 1933): The orthogonal projection of the data onto a lower dimensional linear space such that the variance of the projected data is maximized
  - Minimum projection error (Pearson 1901): The linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections
- The two different definitions lead to the same algorithm, PCA

Definitions of PCA

- Maximum variance vs. minimum projection error

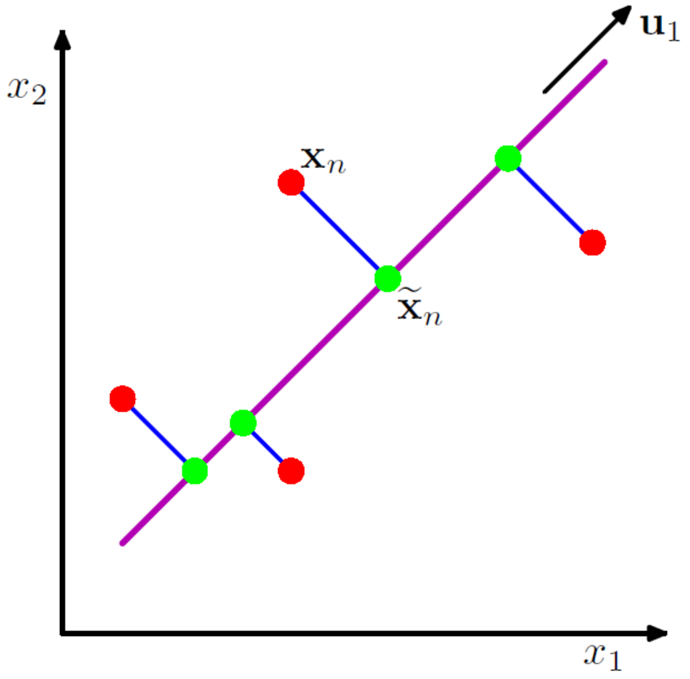
Points: data

Points: projected data

Line: 1D principal space

Definition 1: Maximize the variance of green points

Definition 2: Minimize the sum of squared projection errors, indicated by blue lines



Maximum variance formulation

- Consider a data set of observations $\{\mathbf{x}_n\}$ where $n = 1, 2, \dots, N$ and $\mathbf{x}_n$ in an Euclidean space of dimensionality D
- The goal of PCA is to project the data onto a space having dimensionality $M < D$ while maximizing the variance of the projected data
- Consider the projection onto a one-dimensional space ($M = 1$)
- We can define the direction of this space using a D-dimensional unit vector $\mathbf{u}_1$ and $\mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 = 1$
  - Data point: $\mathbf{x}_n$
  - Projected data point: $\mathbf{u}_1^{\mathrm{T}}\mathbf{x}_n$

國立交通大學 National Chiao Tung University

Maximum variance formulation

- Data mean and projected data mean

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n \quad \text{and} \quad \mathbf{u}_1^{\mathrm{T}}\overline{\mathbf{x}}$$

- The variance of the projected data is

$$\frac{1}{N}\sum_{n=1}^{N}\left\{\mathbf{u}_1^{\mathrm{T}}\mathbf{x}_n - \mathbf{u}_1^{\mathrm{T}}\overline{\mathbf{x}}\right\}^2 = \mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1$$

where S is the covariance matrix

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

Maximum variance formulation: Optimization

- Constrained optimization problem

$$\text{maximize} \quad \mathbf{u}_1^{\mathrm{T}} S \mathbf{u}_1$$
$$\text{subject to} \quad \mathbf{u}_1^{\mathrm{T}} \mathbf{u}_1 = 1$$

- Introduce a Lagrange multiplier to convert the constrained optimization problem to an unconstrained one, where the Lagrangian function is

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

- By setting the derivative w.r.t. $\mathbf{u}_1$ to zero, we have

$$\mathbf{S}\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

- $\mathbf{u}_1$ is an eigenvector of S

Maximum variance formulation: Optimization

- The objective in the constrained optimization problem of PCA is

$$\mathbf{u}_1^{\mathrm{T}}\mathbf{S}\mathbf{u}_1 = \lambda_1$$

- The maximum variance of the projected data is equal to the largest eigenvalue λ_1 of the covariance matrix S

- The projection vector $\mathbf{u}_1$ is the eigenvector of S corresponding to the largest eigenvalue λ_1

- The eigenvector $\mathbf{u}_1$ is known as the first principal component, which is used to project a data point $\mathbf{x}_n$ via $\mathbf{u}_1^{\mathrm{T}}\mathbf{x}_n$

國立交通大學
National Chiao Tung University

Maximum variance formulation: Multi-dimensional extension

- Consider the general case of an M-dimensional projection space
- The optimal linear projection is defined by the M eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots,$ and $\mathbf{u}_M$ of the data covariance matrix S corresponding to the M largest eigenvalues $\lambda_1, \lambda_2, \ldots,$ and λ_M.
- Computational cost of PCA
  - ➢ Full eigenvector decomposition: $O(D^3)$
  - ➢ Find the first M eigenvectors and eigenvalues via the power method: $O(MD^2)$

Minimum error formulation

- We introduce a complete orthonormal set of D-dimensional basis vectors $\{\mathbf{u}_i\}$ where $i = 1, 2, \dots, D$

- Each data point can be represented exactly by a linear combination of the basis vectors

$$\mathbf{x}_n = \sum_{i=1}^{D} \alpha_{ni} \mathbf{u}_i$$

  - where $\alpha_{ni} = \mathbf{x}_n^{\mathrm{T}} \mathbf{u}_i$

- We have

$$\mathbf{x}_n = \sum_{i=1}^{D} \left(\mathbf{x}_n^{\mathrm{T}} \mathbf{u}_i \right) \mathbf{u}_i .$$

國立交通大學
National Chiao Tung University

Minimum error formulation

- We aim to approximate this data point using a representation involving a restricted number $M < D$ of variables, which corresponds to a projection onto a lower-dimensional subspace
- Without loss of generality, we approximate each data point $\mathbf{x}_n$ by

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

  - Each point $\mathbf{x}_n$ has its own coefficients $\{z_{ni}\}$ for the first M basis vectors
  - $\{b_i\}$ are constants that are the same for all data points
- In minimum error formulation, we optimize $\{\mathbf{u}_i\}, \{z_{ni}\}$, and $\{b_i\}$

Minimum error formulation: Optimization

- In this formulation, we minimize the squared projection error, the squared distance, i.e.,

$$J = \frac{1}{N}\sum_{n=1}^{N}\|\mathbf{x}_n - \widetilde{\mathbf{x}}_n\|^2.$$

- Setting the derivative of J with respect to $\{z_{nj}\}$ to zero, we get

$$z_{nj} = \mathbf{x}_n^{\mathrm{T}}\mathbf{u}_j \quad \text{where } j = 1, 2, \dots, M$$

- Setting the derivative of J with respect to $\{b_j\}$ to zero, we get

$$b_j = \overline{\mathbf{x}}^{\mathrm{T}}\mathbf{u}_j \quad \text{where } j = M+1, M+2, \dots, D$$

國立交通大學
National Chiao Tung University

Minimum error formulation: Optimization

- If we substitute for $\{z_{ni}\}$ and $\{b_i\}$, the displacement vector between $\mathbf{x}_n$ and its approximated point $\tilde{\mathbf{x}}_n$

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

  - The displacement vector lies in the space orthogonal to the principal space

- The squared projection error

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

Minimum error formulation: Optimization

$$J = \sum_{i=M+1}^{D} \mathbf{u}_i^{\mathrm{T}} \mathbf{S} \mathbf{u}_i$$

- Consider the example of a two-dimensional data space with one-dimensional projection space, i.e., $D = 2$ and $M = 1$
- We want to optimize $\mathbf{u}_2$ so as to minimize $J = \mathbf{u}_2^{\mathrm{T}} S \mathbf{u}_2$, subject to $\mathbf{u}_2^{\mathrm{T}} \mathbf{u}_2 = 1$, i.e.,

$$\begin{aligned} \text{minimize} \quad & \mathbf{u}_2^{\mathrm{T}} S \mathbf{u}_2 \\ \text{subject to} \quad & \mathbf{u}_2^{\mathrm{T}} \mathbf{u}_2 = 1 \end{aligned}$$

- The Lagrangian function is

$$\widetilde{J} = \mathbf{u}_2^{\mathrm{T}} \mathbf{S} \mathbf{u}_2 + \lambda_2 \left(1 - \mathbf{u}_2^{\mathrm{T}} \mathbf{u}_2\right).$$

- Setting the derivative with respect to $\mathbf{u}_2$ to zero leads to

$$\mathrm{S}\mathbf{u}_2 = \lambda_2 \mathbf{u}_2$$

- $\mathbf{u}_2$ is an eigenvector of S with eigenvalue λ_2

國立交通大學 National Chiao Tung University

Minimum error formulation: Optimization

- The objective in this formulation is to minimize

$$\mathbf{u}_2^{\mathrm{T}} S \mathbf{u}_2 = \mathbf{u}_2^{\mathrm{T}} \lambda_2 \mathbf{u}_2 = \lambda_2$$

- The basis $\mathbf{u}_2$ is the eigenvector of covariance matrix S with the smallest eigenvalue

- Recall that

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

- The projection vector $\mathbf{u}_1$ is the eigenvector of S corresponding to the largest eigenvalue λ_1

Minimum error formulation: Multi-dimensional extension

- For arbitrary D and arbitrary $M < D$, the optimal linear projection is defined by the M eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, …, and $\mathbf{u}_M$ of the data covariance matrix S corresponding to the M largest eigenvalues λ_1, λ_2, …, and λ_M

- The approximated point

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

Algorithm summary

- 1. We are given a set of N unlabeled data $\{\mathbf{x}_n\}$. Data lie in D-dimensional space. The desired dimension of the projection space is M
- 2. Compute the covariance matrix

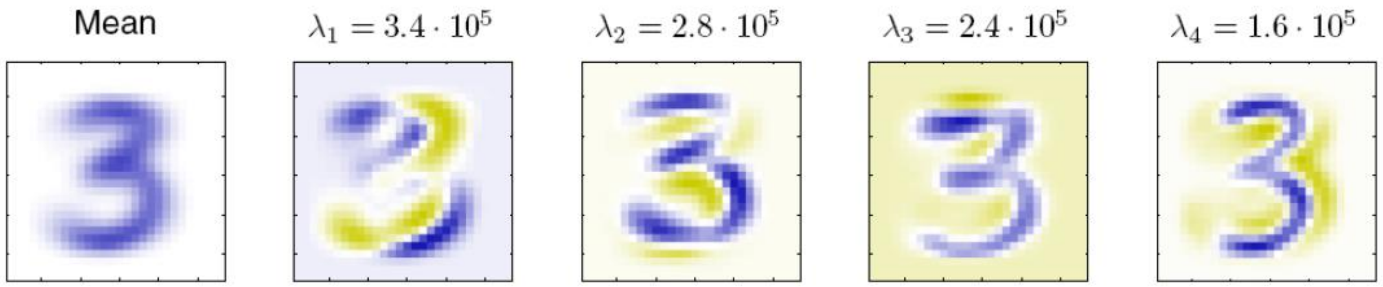
$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}}.$$

- 3. Perform eigen-decomposition
  - Get M eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, …, and $\mathbf{u}_M$ of S corresponding the largest M eigenvalues
- 4. Approximate the data $\{\mathbf{x}_n\}$ via

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

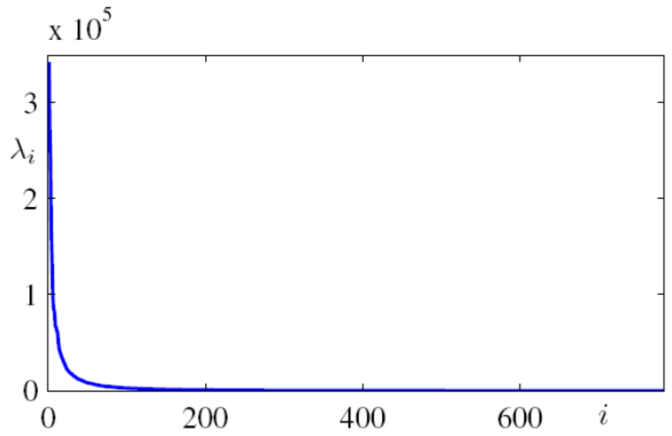
Applications of PCA: Data compression

- Off-line digit data set: handwritten digit images, each of which is of resolution 28 x 28 (784)
- Because each eigenvector of the covariance matrix is a vector in the original D-dimensional space, it can be displayed as an image
- The mean vector (image) and the first four PCA eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, and $\mathbf{u}_4$, together with the corresponding eigenvalues


Mean
$\lambda_1 = 3.4 \cdot 10^5$
$\lambda_2 = 2.8 \cdot 10^5$
$\lambda_3 = 2.4 \cdot 10^5$
$\lambda_4 = 1.6 \cdot 10^5$

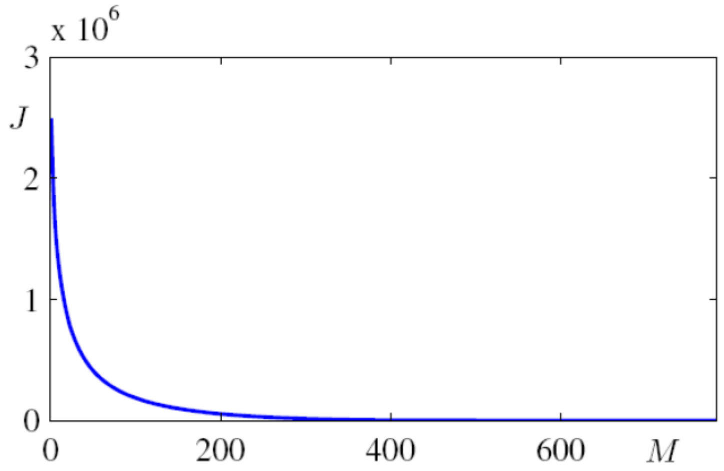
Applications of PCA: Data compression

- The spectrum of eigenvalues, sorted into decreasing order



- The distortion measure (squared projection error) J associated with a particular value of M

$$J = \sum_{i=M+1}^{D} \lambda_i$$



國立交通大學
National Chiao Tung University

Applications of PCA: Data compression

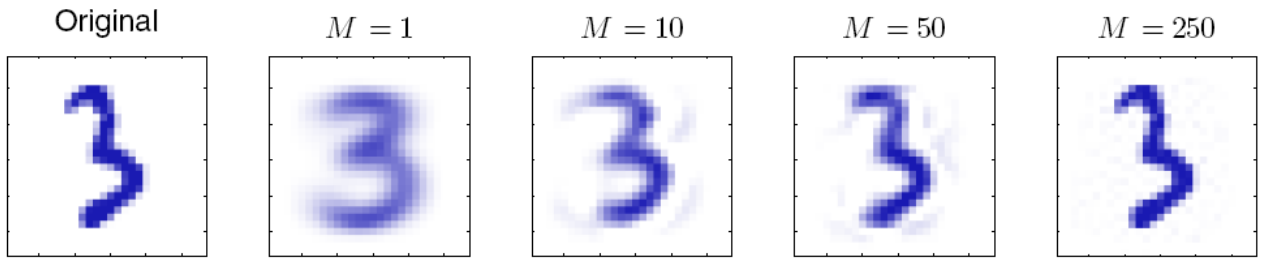
- PCA is used to approximate each input data point $\mathbf{x}_n$ via

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

  - The coordinate of $\mathbf{x}_n$ in the low-dimensional space is $[a_i]$= ($\mathbf{x}_n^{\mathrm{T}}\mathbf{u}_i$- $\overline{\mathbf{x}}^{\mathrm{T}}\ \mathbf{u}_i$), for $1 \leq i \leq M$
  - The reconstructed data point in the original space is $\tilde{\mathbf{x}}_n$
  - Storage space before PCA: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$
  - Storage space using PCA: $\overline{\mathbf{x}}, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M$, the coordinate of each data point

Applications of PCA: Data compression

- An original example and its PCA reconstructions with different values of M



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 $M = D = 28 \times 28 = 784$.

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

Applications of PCA: Data normalization

- PCA can make a more substantial normalization of data to give it zero mean and unit covariance
  - Different input variables become decorrelated
- This processing is called whitening or sphering
- 1. Solve the eigenvalues and eigenvectors of the covariance matrix S

$$\mathbf{SU} = \mathbf{UL}$$

  - where L is a $D \times D$ diagonal matrix with elements $\{\lambda_i\}$
  - U is a $D \times D$ orthogonal matrix with columns given by $\{\mathbf{u}_i\}$

Applications of PCA: Data normalization

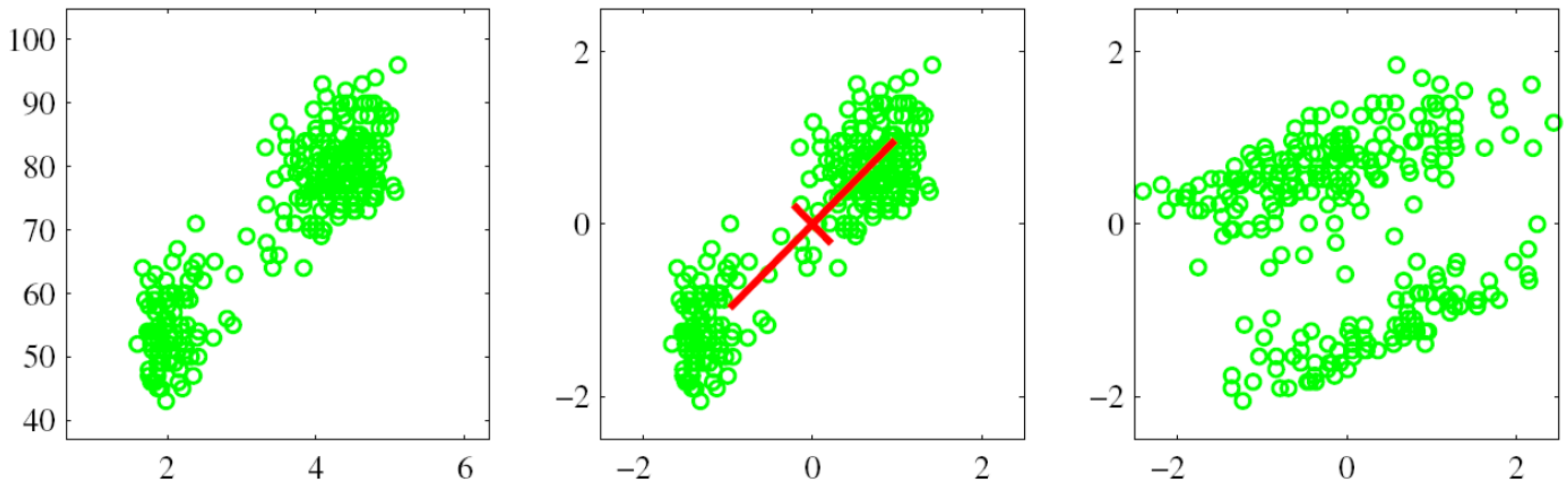
- 2. Transform each data point $\mathbf{x}_n$ to

$$\mathbf{y}_n = \mathbf{L}^{-1/2}\mathbf{U}^{\mathrm{T}}(\mathbf{x}_n - \overline{\mathbf{x}})$$

- The set $\{\mathbf{y}_n\}$ has zero mean
- The set $\{\mathbf{y}_n\}$ has an identity covariance matrix

$$\begin{aligned}\frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n\mathbf{y}_n^{\mathrm{T}} &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{L}^{-1/2}\mathbf{U}^{\mathrm{T}}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^{\mathrm{T}}\mathbf{U}\mathbf{L}^{-1/2} \\ &= \mathbf{L}^{-1/2}\mathbf{U}^{\mathrm{T}}\mathbf{S}\mathbf{U}\mathbf{L}^{-1/2} = \mathbf{L}^{-1/2}\mathbf{L}\mathbf{L}^{-1/2} = \mathbf{I}.\end{aligned}$$

Applications of PCA: Data normalization



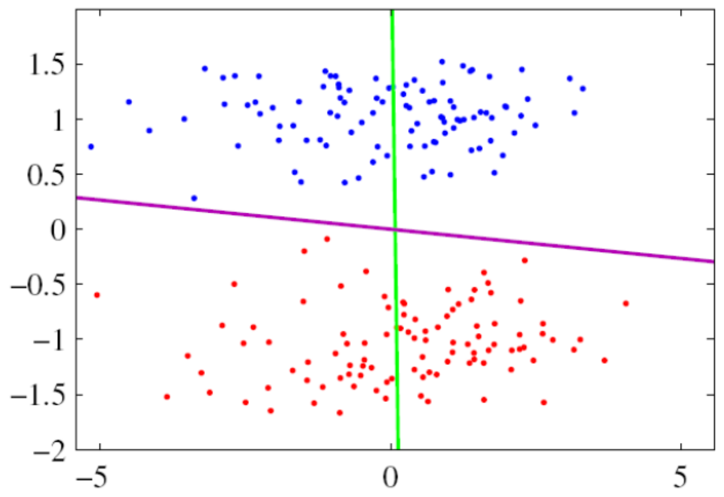
- Left: Original data in a two-dimensional space
- Center: Two principle components and the corresponding eigenvalues
- Right: The whitening results
  - Zero mean and unit covariance

國立交通大學
National Chiao Tung University

PCA vs. FLD

- Comparisons between PCA and the Fisher linear discriminant
  - Both methods can be viewed as techniques for linear dimensionality reduction
  - PCA is unsupervised and depends only on the input data $\{\mathbf{x}_n\}$
  - FLD depends on both data $\{\mathbf{x}_n\}$ and class-label information $\{t_n\}$

- Line: 1D subspace by PCA
  - Maximum variance
  - Minimum error
- Line: 1D subspace by FLD
  - Maximum data separation

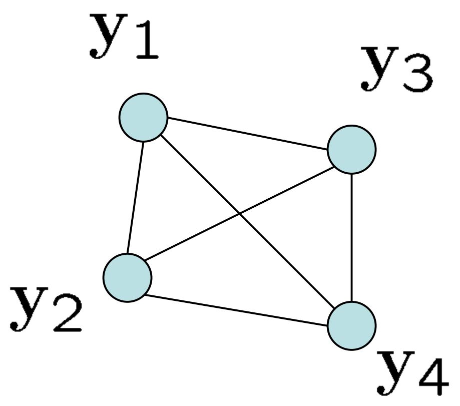


Multidimensional scaling (MDS)

- Given the pairwise distances between a set of N data, namely, $\{\Delta_{ij}\}_{i,j=1}^{N}$, we would like to seek N vectors $\{\boldsymbol{y}_i\}_{i=1}^{N}$ so that $||\boldsymbol{y}_i - \boldsymbol{y}_j|| \approx \Delta_{ij}$

- Input Output

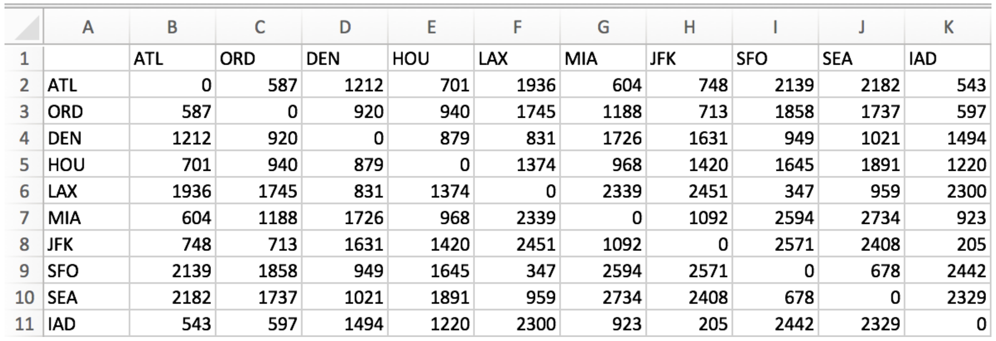
$$\begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{12} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{13} & \Delta_{23} & 0 & \Delta_{34} \\ \Delta_{14} & \Delta_{24} & \Delta_{34} & 0 \end{bmatrix}$$



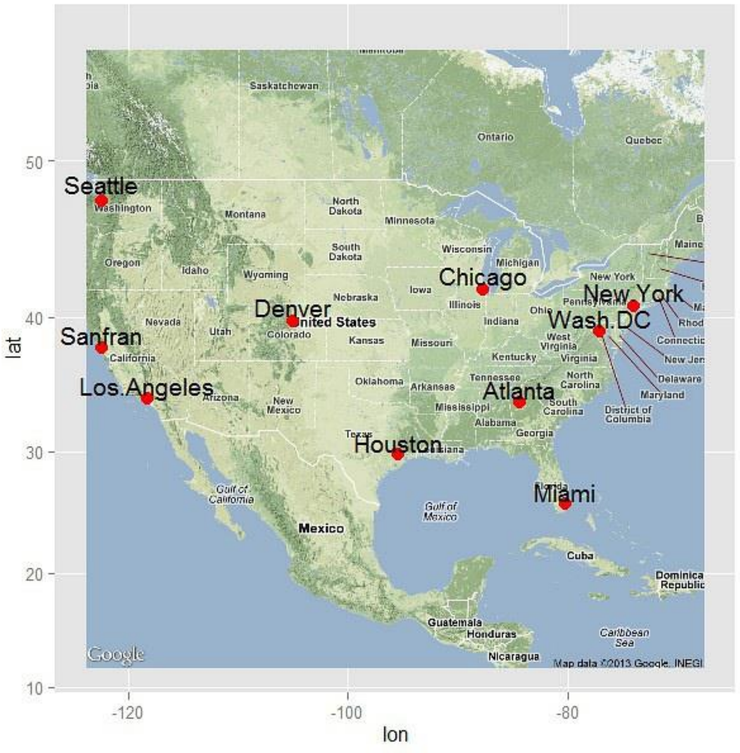
$$\|\mathbf{y}_i - \mathbf{y}_j\| \approx \Delta_{ij}$$

An example of MDS

- The pairwise distances between 10 airports in US
- Airports’ locations in the map

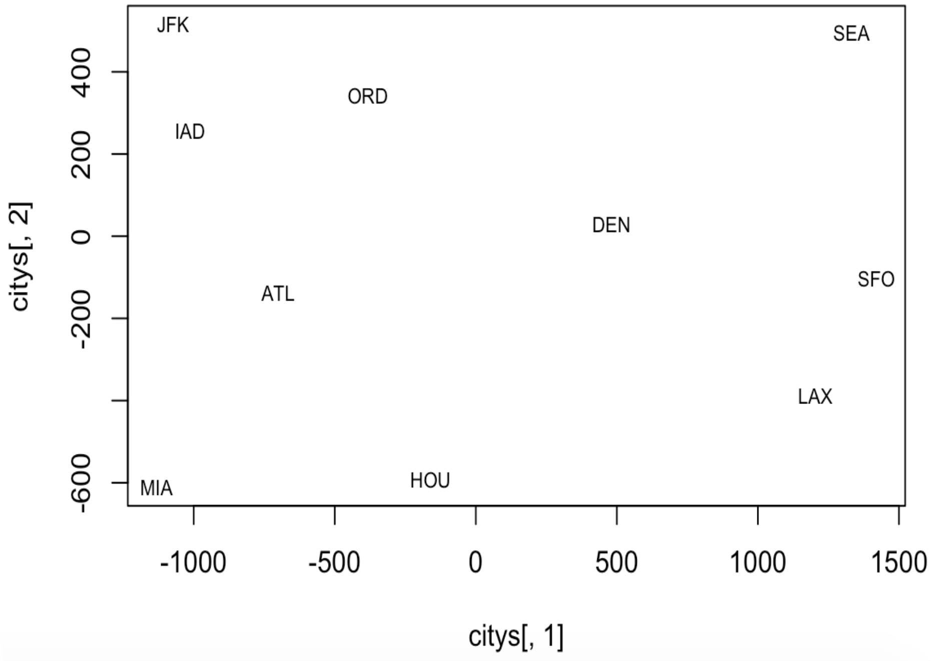


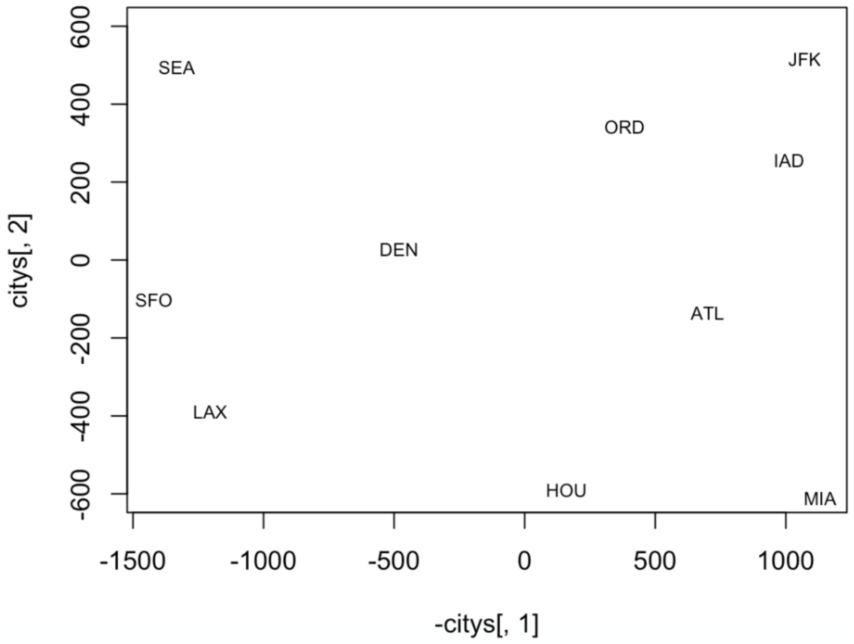
	A	B	C	D	E	F	G	H	I	J	K
1		ATL	ORD	DEN	HOU	LAX	MIA	JFK	SFO	SEA	IAD
2	ATL	0	587	1212	701	1936	604	748	2139	2182	543
3	ORD	587	0	920	940	1745	1188	713	1858	1737	597
4	DEN	1212	920	0	879	831	1726	1631	949	1021	1494
5	HOU	701	940	879	0	1374	968	1420	1645	1891	1220
6	LAX	1936	1745	831	1374	0	2339	2451	347	959	2300
7	MIA	604	1188	1726	968	2339	0	1092	2594	2734	923
8	JFK	748	713	1631	1420	2451	1092	0	2571	2408	205
9	SFO	2139	1858	949	1645	347	2594	2571	0	678	2442
10	SEA	2182	1737	1021	1891	959	2734	2408	678	0	2329
11	IAD	543	597	1494	1220	2300	923	205	2442	2329	0



An example of MDS

- The recovered 2D coordinates of those airports by MDS
- Horizontally mirror the map

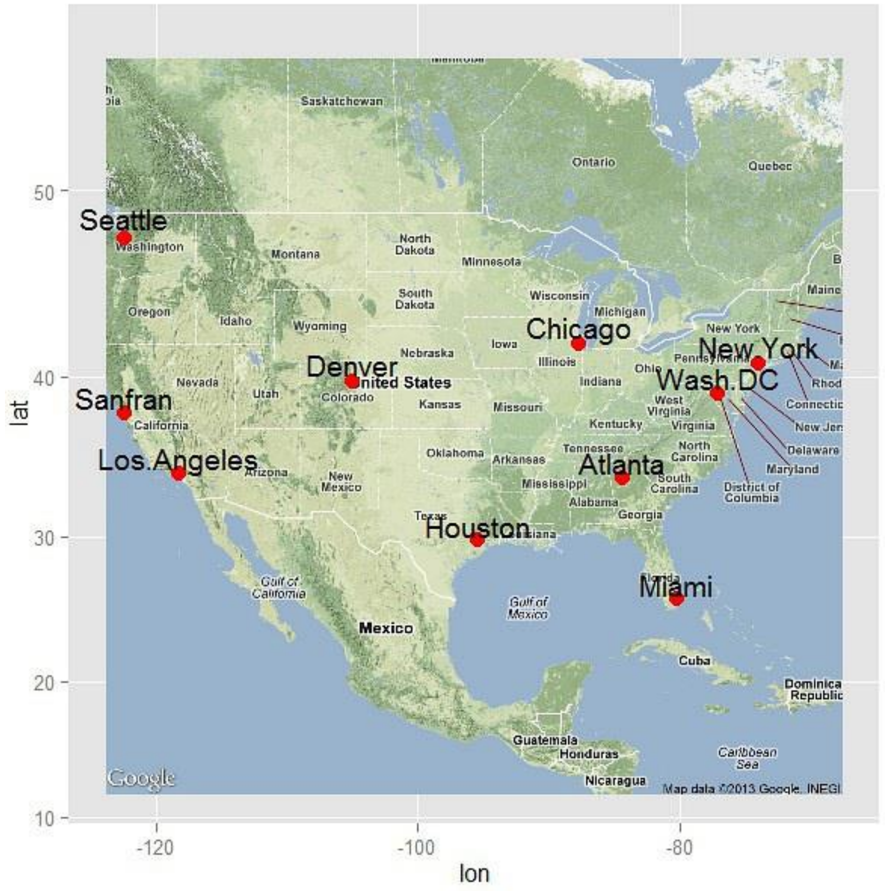


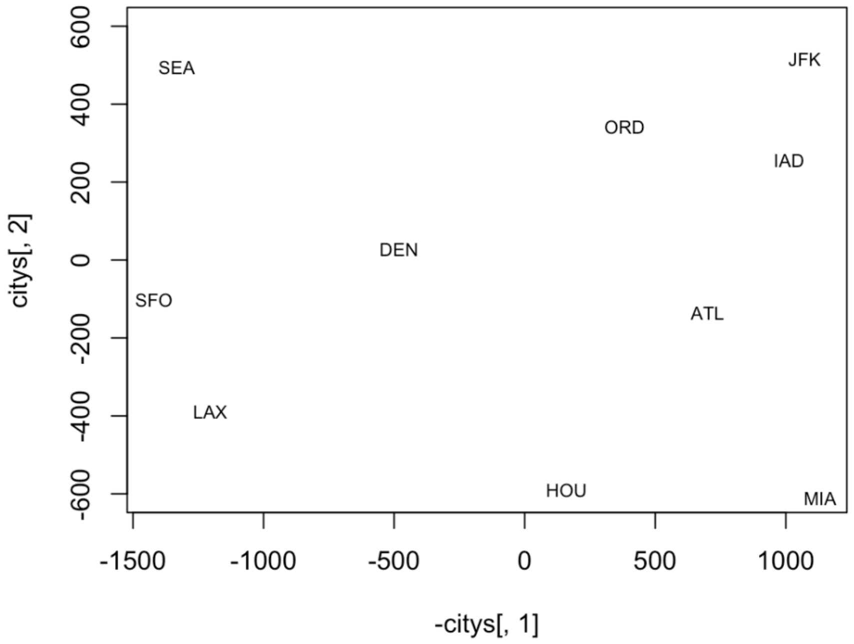


國立交通大學
National Chiao Tung University

An example of MDS

- MDS can find the locations of these airports





National Chiao Tung University

MDS: Two observations

- Observation 1: We can convert pairwise distances to pairwise inner products under some constraints
- Observation 2: The coordinate matrix can be derived from a Gram matrix, $G_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$
- If $\{\Delta_{ij}\}_{i,j=1}^{N}$ denote the pairwise Euclidean distances between zero mean vectors, then their inner products are

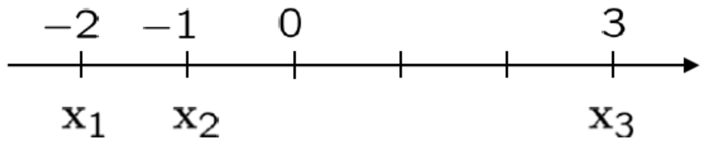
$$G_{ij} = \frac{1}{2}[\frac{1}{N}\sum_{k=1}^{N}\left(\Delta_{ik}^2 + \Delta_{kj}^2\right) - \Delta_{ij}^2 - \frac{1}{N^2}\sum_{k,l=1}^{N}\Delta_{kl}^2]$$

國立交通大學 National Chiao Tung University

Distance matrix to Gram matrix

$$G_{ij} = \frac{1}{2}[\frac{1}{N}\sum_{k=1}^{N}\left(\Delta_{ik}^2 + \Delta_{kj}^2\right) - \Delta_{ij}^2 - \frac{1}{N^2}\sum_{k,l=1}^{N}\Delta_{kl}^2]$$

- An example: three 1D points:



- Distance matrix:

$$[\Delta_{ij}] = \begin{bmatrix} 0 & 1 & 5 \\ 1 & 0 & 4 \\ 5 & 4 & 0 \end{bmatrix} \Rightarrow [\Delta_{ij}^2] = \begin{bmatrix} 0 & 1 & 25 \\ 1 & 0 & 16 \\ 25 & 16 & 0 \end{bmatrix}$$

- Inner product:

$$\begin{aligned} G_{12} =& \frac{1}{2}\left[\frac{1}{3}\sum_{k}\left(\Delta_{1k}^2 + \Delta_{k2}^2\right) - \Delta_{12}^2 - \frac{1}{9}\sum_{k,l}\Delta_{kl}^2\right] \\ =& \frac{1}{2}\left[\frac{1}{3}(0+1+25+1+0+16) - 1\right. \\ & \left. - \frac{1}{9}(0+1+25+1+0+16+25+16+0)\right] \\ =& 2 \end{aligned}$$

國立交通大學 National Chiao Tung University

MDS: Objective and optimization

- Objective of MDS

$$\text{err}(\mathbf{y}) = \sum_{i,j} \left(G_{ij} - \mathbf{y}_i \cdot \mathbf{y}_j\right)^2$$

- Perform eigen-decomposition for Gram matrix G

$$G = \sum_{\alpha=1}^{N} \lambda_\alpha \mathbf{v}_\alpha \mathbf{v}_\alpha^\mathrm{T} \quad \text{where} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$$

- Optimal (low-rank) approximation

$$\mathbf{y}_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i} \quad \text{for} \quad \alpha = 1, 2, \ldots, M$$

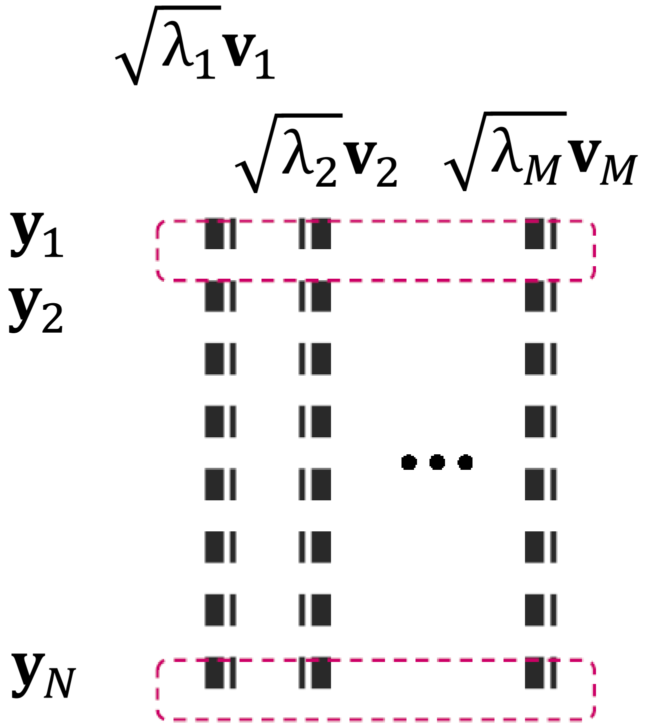
  - ➢ Each data vector is constructed by scaled, truncated eigenvectors

MDS: Embedding

- Eigen-decomposition

$$G = \sum_{\alpha=1}^{N} \lambda_\alpha \mathbf{v}_\alpha \mathbf{v}_\alpha^{\mathrm{T}} \quad \text{where} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$$

- Embedding of the N data points



$\sqrt{\lambda_1}\mathbf{v}_1$
$\sqrt{\lambda_2}\mathbf{v}_2$
$\sqrt{\lambda_M}\mathbf{v}_M$
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_N$

MDS: Interpretation

- Embedding

$$\mathbf{y}_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i} \quad \text{for} \quad \alpha = 1, 2, \ldots, M$$

- Eigenvectors
  - Ordered, scaled, and truncated to yield low-dimensional embedding
- Eigenvalues
  - Measure how each dimension contributes to dot products
- Estimated dimensionality
  - Number of significant (nonnegative) eigenvalues

PCA vs. MDS

- Perform eigen-decomposition
  - PCA: covariance matrix of size $D \times D$
  - MDS: Gram matrix of size $N \times N$
- Both PCA and MDS consider the M largest eigenvalues
- Eigenvectors in PCA serve as the projection matrix
- Scaled eigenvectors in MDS are used for embedding, the projected data vectors

Summary of MDS

- Convert the distance matrix to Gram matrix

$$G_{ij} = \frac{1}{2}[\frac{1}{N}\sum_{k=1}^{N}\left(\Delta_{ik}^2 + \Delta_{kj}^2\right) - \Delta_{ij}^2 - \frac{1}{N^2}\sum_{k,l=1}^{N}\Delta_{kl}^2]$$

- Perform eigen-decomposition of the Gram matrix

$$G = \sum_{\alpha=1}^{N} \lambda_\alpha \mathbf{v}_\alpha \mathbf{v}_\alpha^{\mathrm{T}} \quad \text{where} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 0$$

- Embedding via low rank approximaition

$$\mathbf{y}_{i\alpha} = \sqrt{\lambda_\alpha} v_{\alpha i} \quad \text{for} \quad \alpha = 1, 2, \ldots, M$$

Manifold (流形)

- In mathematics, a manifold is a topological space that locally resembles Euclidean space near each point
  - In the neighborhood of any point on a manifold, the space behaves just like it would in the neighborhood of any point in some n-dimensional Euclidean space

- A sphere in 3-dimensional Euclidean space
  - A very small patch from the sphere looks just like a piece of 2-dimensional Euclidean space
  - 2-manifold

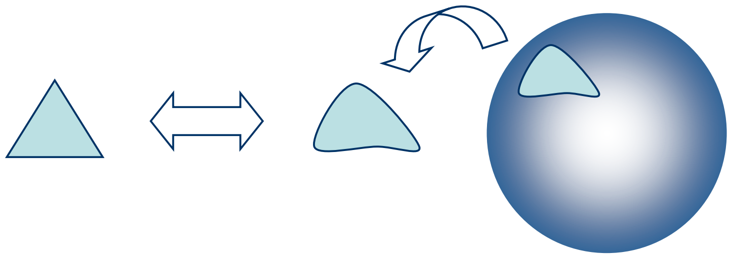
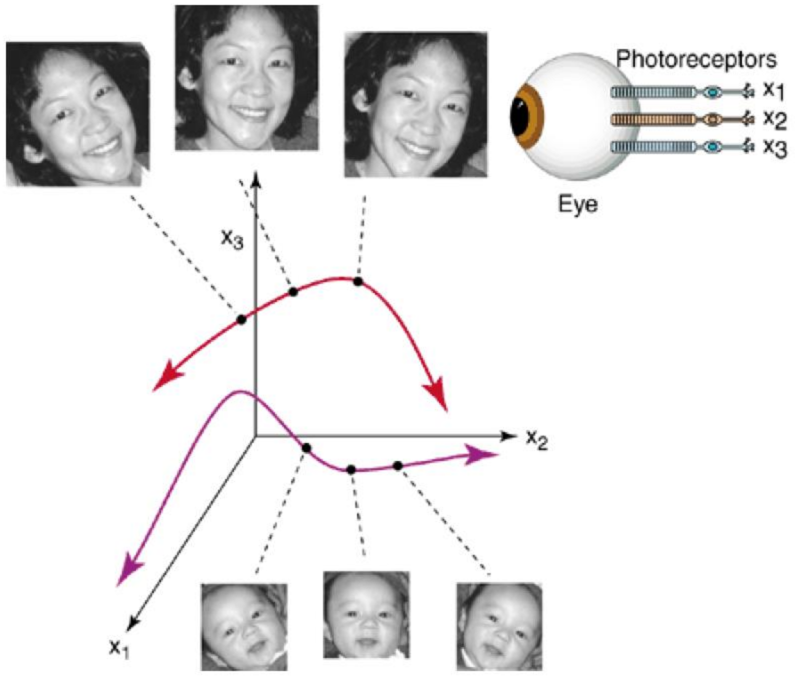
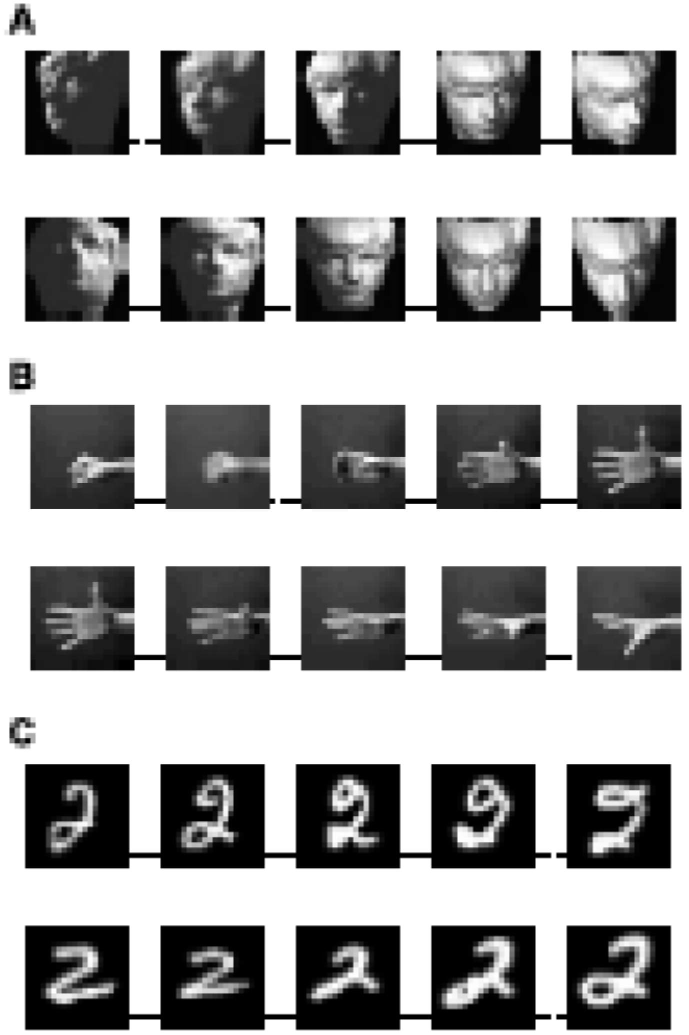


Image manifolds

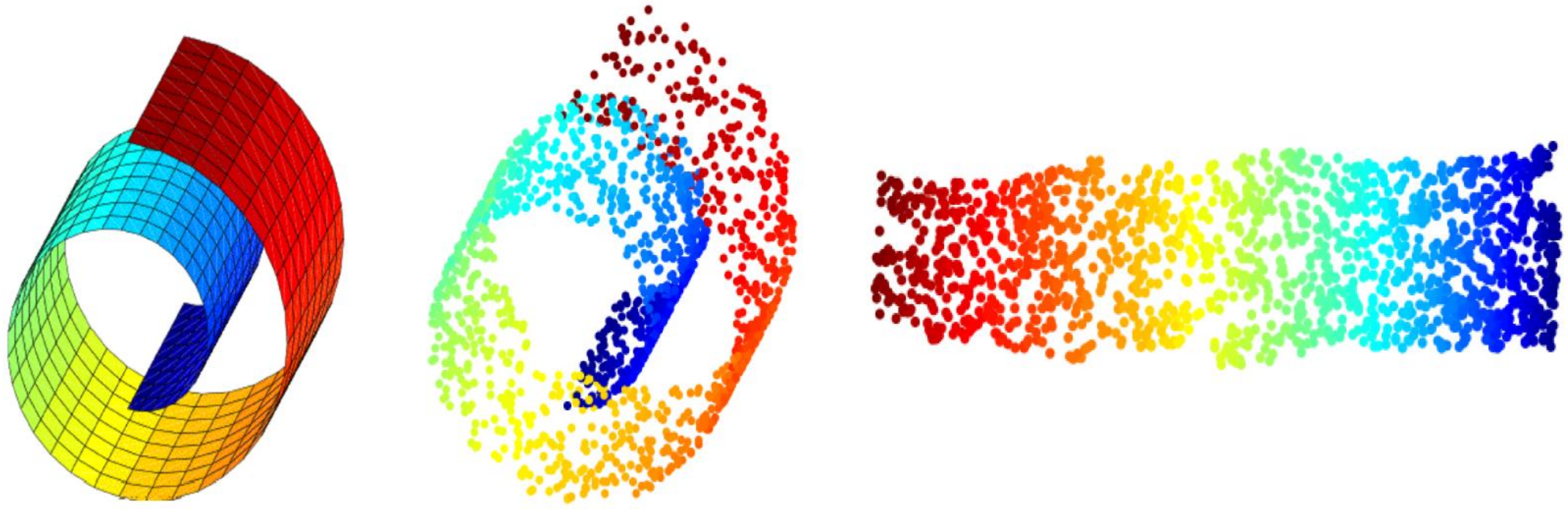




Seung & Lee, 2000
Tenebaum et al., 2000

Manifold learning

- Given high-dimensional data sampled from a low-dimensional manifold, how to compute a faithful embedding?



Isometric mapping (Isomap) [Tenenbaum et al. Science 2000]

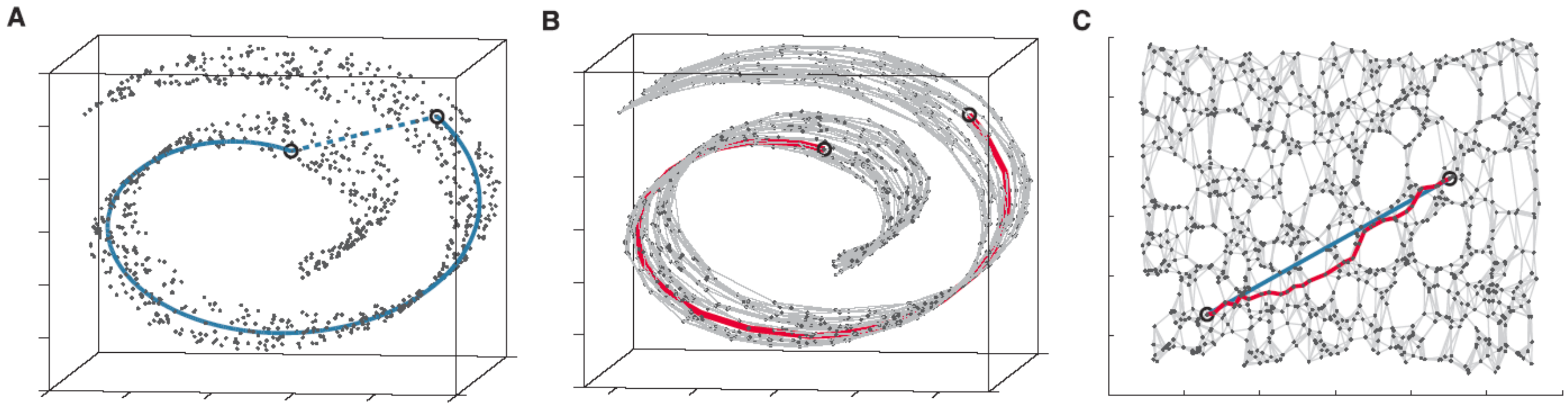
- Input: a data set of observations $\{\mathbf{x}_n\}$ where $n = 1, 2, \dots, N$ and $\mathbf{x}_n$ in a D-dimensional space

- Output: a low-dimensional (M-dimensional) data representation $\mathbf{y}_n$ for each input data point $\mathbf{x}_n$, where $M \ll D$

- Goal: The geodesic distance in the original space can be preserved in the low-dimensional space
  - Nonlinear dimensionality reduction

Geodesic distance

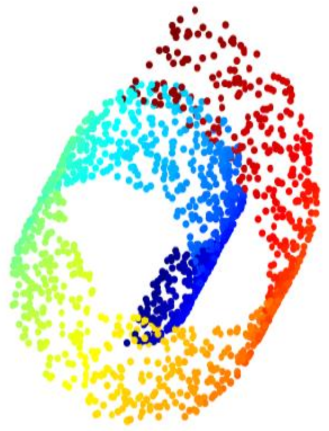
- The geodesic distance between two points on a manifold is the length of the shortest connecting path along the manifold
- Figure A: the geodesic distance between two points (circles)
- Figure B: An approximation of the geodesic distance based on sampled points
- Figure C: The outputs of Isomap, where geodesic distances can be approximately measured by Euclidean distances

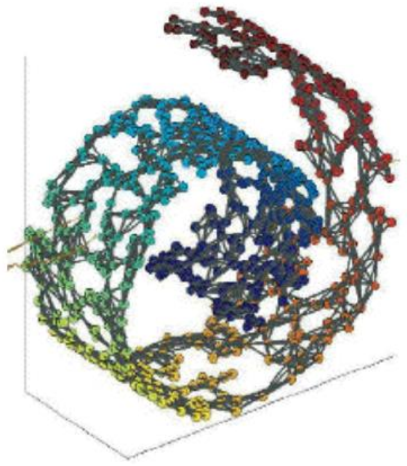


Step 1: Build an adjacency graph

- Adjacency graph
  - Vertices represent inputs
  - Undirected edges connect neighbors

- Neighborhood selection
  - Many options:
    - *k* nearest neighbors
    - ε-ball
    - Prior knowledge
  - Weight edges by local Euclidean distances





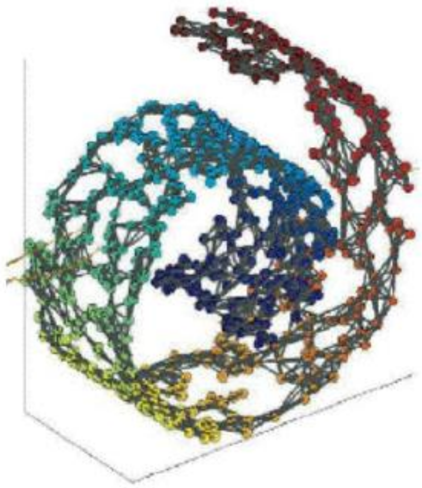
国立交通大學
National Chiao Tung University

Step 2: Estimate geodesics

- Dynamic programming
  - Weight edges by local Euclidean distances
  - Compute shortest paths through graph

- Dijkstra's algorithm for computing the shortest path between a pair of vertices

- Floyd's algorithm for computing the pair-wise shortest paths



Step 3: Apply MDS

- Embedding
  - Given the pair-wise distances between data points, we can apply MDS to recover the data vectors of these points in a low-dimensional space where the distances are preserved

Algorithm of Isomap

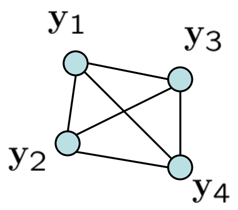
- Algorithm

  (1) k nearest neighbors

  (2) Shortest paths through graph (estimating geodesic distances)

  (3) MDS on geodesic distances

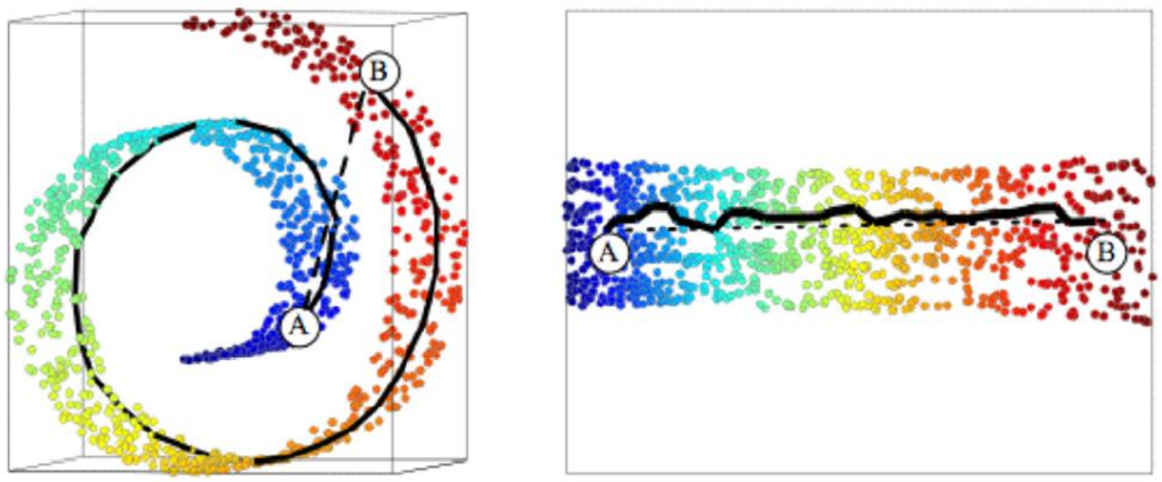
$$\begin{bmatrix} 0 & \Delta_{12} & \Delta_{13} & \Delta_{14} \\ \Delta_{12} & 0 & \Delta_{23} & \Delta_{24} \\ \Delta_{13} & \Delta_{23} & 0 & \Delta_{34} \\ \Delta_{14} & \Delta_{24} & \Delta_{34} & 0 \end{bmatrix}$$



- Impact
  - Much simpler than earlier algorithms for manifold learning
  - Does it work?

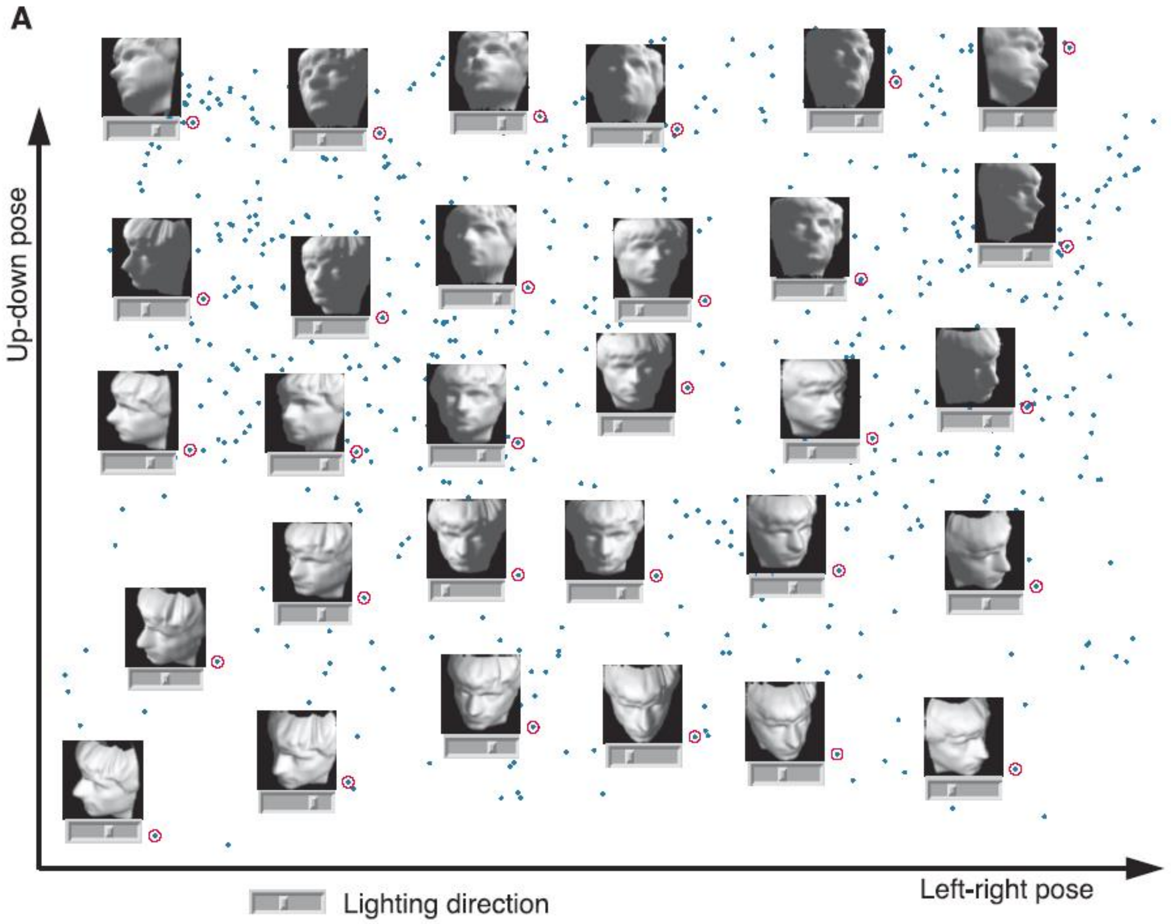
Visualization

- Swiss roll dataset
  - 1024 data points in a 3D space
  - *k*=12 in the adjacency graph construction
  - 2-dimensional space where the Euclidean distance between a pair of data points approximates their geodesic distance on the manifold



Visualization

- Face images
  - 64 x 64 (4096)-dim, 698 data points, *k* = 6



National Chiao Tung University

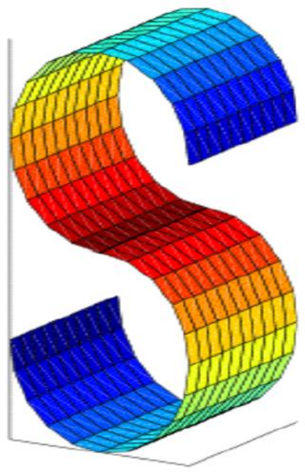
Locally linear embedding (LLE) [Roweis and Saul, Science 2000]

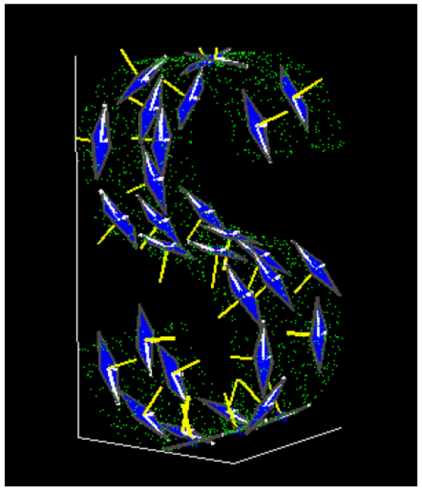
- MDS and Isomap
  - Preserve global pairwise distances
  - Construct large and dense matrices
  - Compute top eigenvectors

- Locally linear embedding (LLE)
  - Preserve local geometric relationships
  - Construct large and sparse matrices
  - Compute bottom eigenvectors

How to exploit local linearity?

- Manifolds are globally nonlinear, but locally linear
- Map the inputs into a single continuous global coordinate system of lower dimensionality
  - Think globally, fit locally





Locally linear embedding

- Input: a data set of observations $\{\mathbf{x}_i\}$ where $i = 1, 2, \ldots, N$ and $\mathbf{x}_i$ in a D-dimensional space

- Output: a low-dimensional (M-dimensional) data representation $\mathbf{y}_i$ for each input data point $\mathbf{x}_i$, where $M \ll D$

- Goal: The neighborhood of each point, i.e., local geometric structure, in the original space is preserved in the low-dimensional space
  - Nonlinear dimensionality reduction

LLE algorithm

- Steps

  (1) Nearest neighbor search for each input data point

  (2) Least squares fitting for neighborhood representation

  (3) Eigen-decomposition for embedding

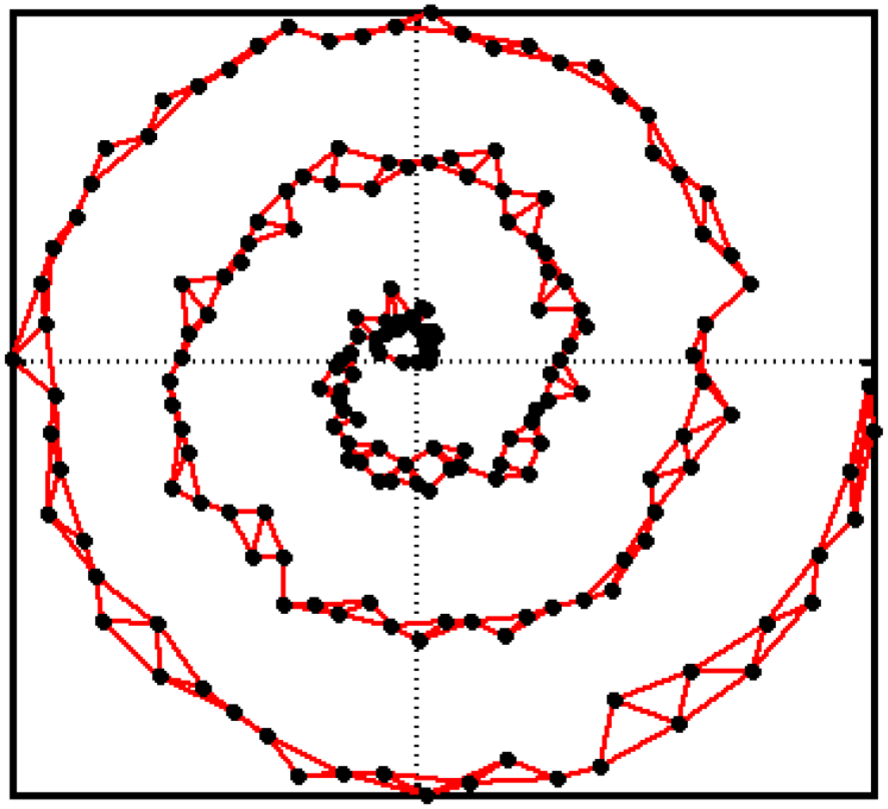
Step 1: Identify neighbors

- Identify the neighbors of each data point $\mathbf{x}_i$
  - *k* nearest neighbors
  - ε-ball
  - Prior knowledge

- Assumptions
  - Data are densely sampled from a manifold
  - Sufficient neighboring data can be retrieved for each data point

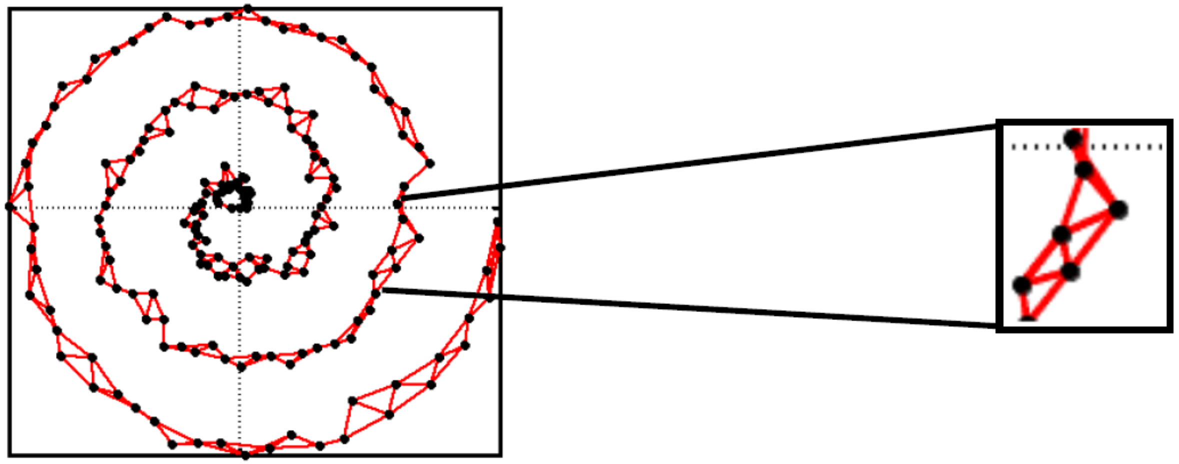
Neighborhood graph

- Assumption
  - Neighborhoods on the graph correspond to neighborhoods on the manifold



Step 2: Compute reconstruction weights

- Characterize local geometry of each neighborhood by weights $\{W_{ij}\}$ for each data $\mathbf{x}_i$



- Compute the weights by linearly reconstructing each point $\mathbf{x}_i$ from its neighbors

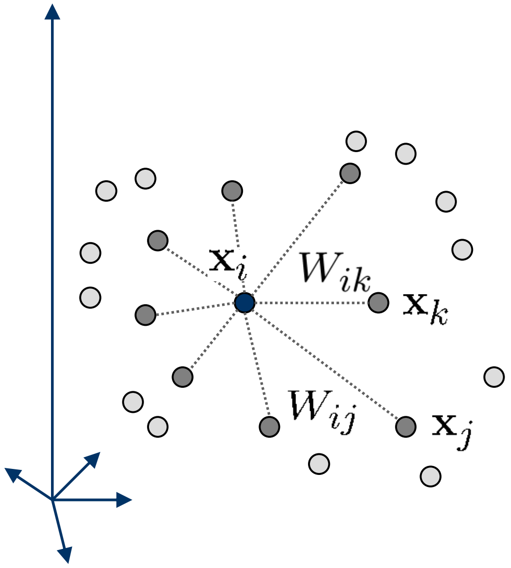
國立交通大學
National Chiao Tung University

Linear reconstructions

- Local linearity
  - Neighbors lie on a locally linear patch of a manifold

- Reconstruction errors
  - Least squared errors should be small

$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right|^2$$



Least squares fitting

- Objective

$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij}\mathbf{x}_j \right|^2$$

- Constraints
  - Nonzero W_{ij} only if $\mathbf{x}_j$ is a neighbor of $\mathbf{x}_i$
  - Weights must sum to one for each $\mathbf{x}_i$: $\sum_j W_{ij} = 1$
- Solver: least squares fitting
- Local invariance:
  - Optimal weights $\{W_{ij}\}$ are invariant to rotation, translation, and scaling

Step 3: Complete embedding

- Low dimensional representation
  - $\mathbf{x}_i$ in a D-dimensional space -> $\mathbf{y}_i$ in an M-dimensional space

- Objective: Minimize the reconstruction errors

$$\Psi(\mathbf{y}) = \sum_i \left| \mathbf{y}_i - \sum_j W_{ij} \mathbf{y}_j \right|^2$$

- Constraints
  - The mean of the projected data is on the origin: $\sum_i \mathbf{y}_i = \mathbf{0}$
  - Impose unit covariance matrix: $\frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i \ \mathbf{y}_i^{\mathrm{T}} = I$

Solving an eigenvalue problem

- Quadratic form

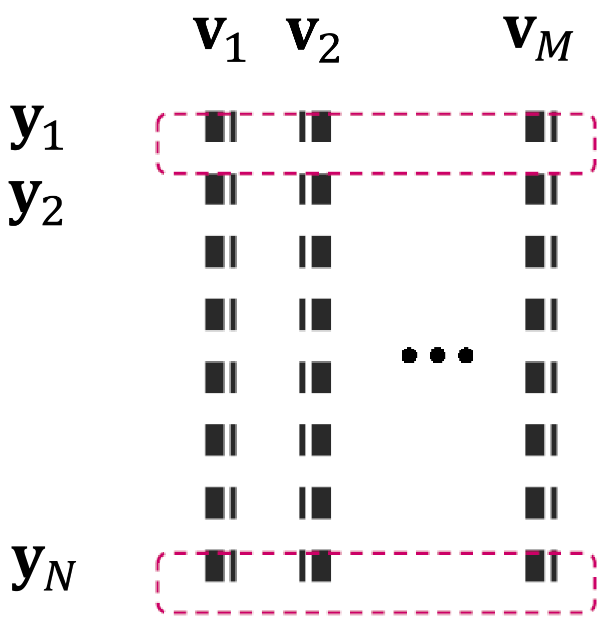
$$\Psi(\mathbf{y}) = \sum_{i,j} \left(\mathbf{y}_i \cdot \mathbf{y}_j\right) \Psi_{ij} \quad \text{with } \Psi = (I - W)^T (I - W)$$

- The optimal embedding is given by the bottom $M + 1$ eigenvectors

- Solution
  - Discard the bottom eigenvector $[1\ 1\ \cdots\ 1]^T$
  - The other eigenvectors are used to yield the low-dimensional data $\{\mathbf{y}_i\}$

Embedding

- After discarding the eigenvector $[1\ 1\ \cdots\ 1]^T$, the low-dimensional data $\{\mathbf{y}_i\}$ are given below



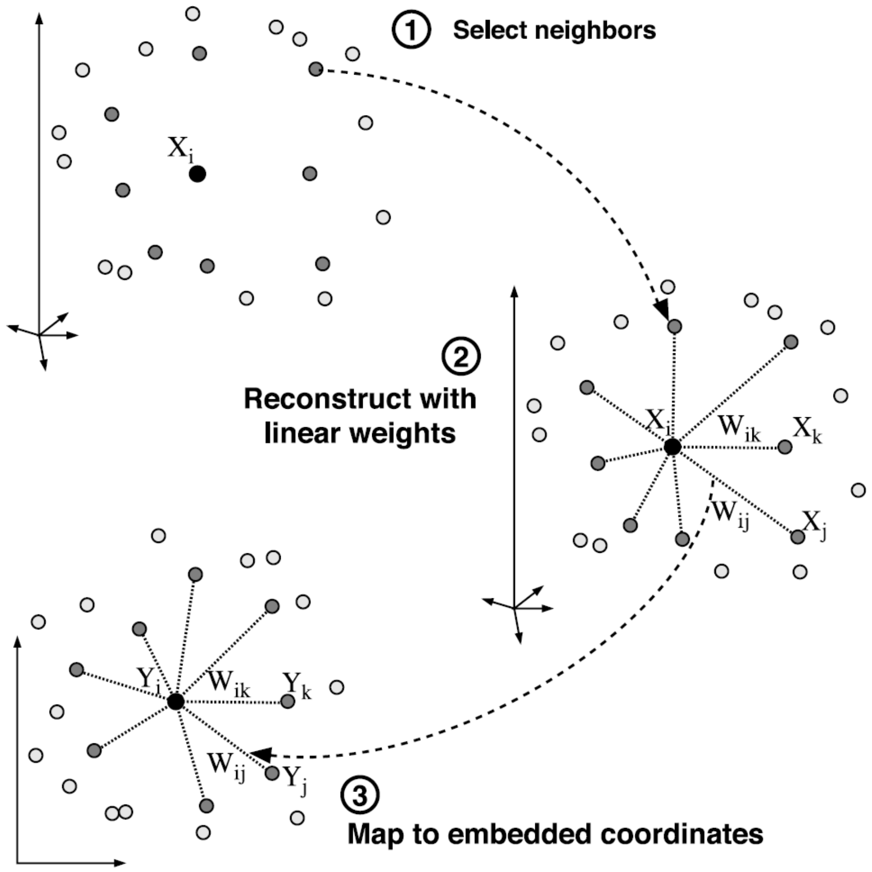
Summary of LLE

- Three steps
  - 1. Get *k*-nearest neighbors
  - 2. Compute weights
  - 3. Complete embedding

- Optimizations

$$\Phi(W) = \sum_i \left| \mathbf{x}_i - \sum_j W_{ij} \mathbf{x}_j \right|^2$$

$$\Psi(\mathbf{y}) = \sum_i \left| \mathbf{y}_i - \sum_j W_{ij} \mathbf{y}_j \right|^2$$



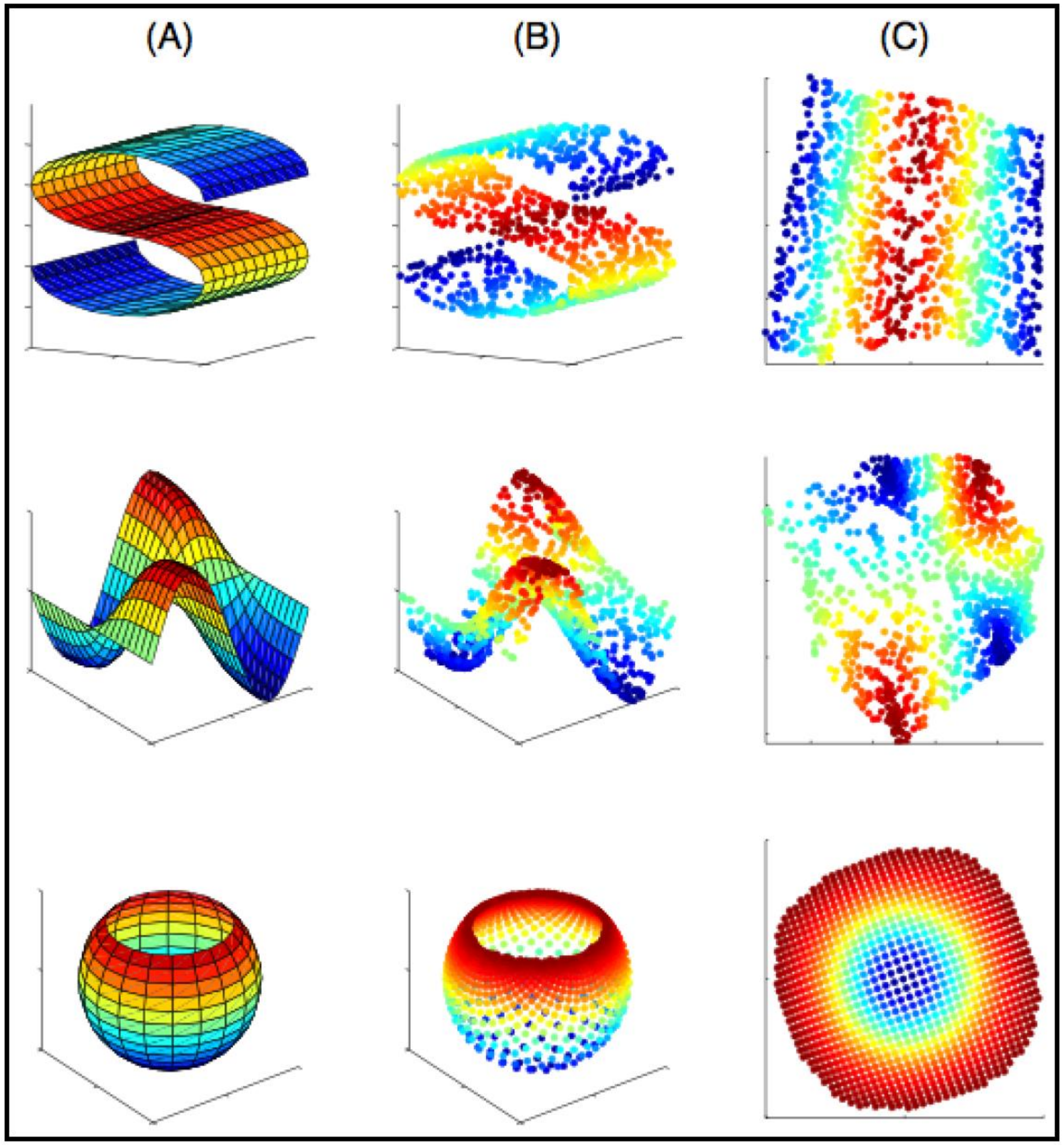
國立交通大學
National Chiao Tung University

Surfaces

$m = 1000$
inputs

$k = 8$
nearest
neighbors

$D = 3$
$d = 2$
dimensions



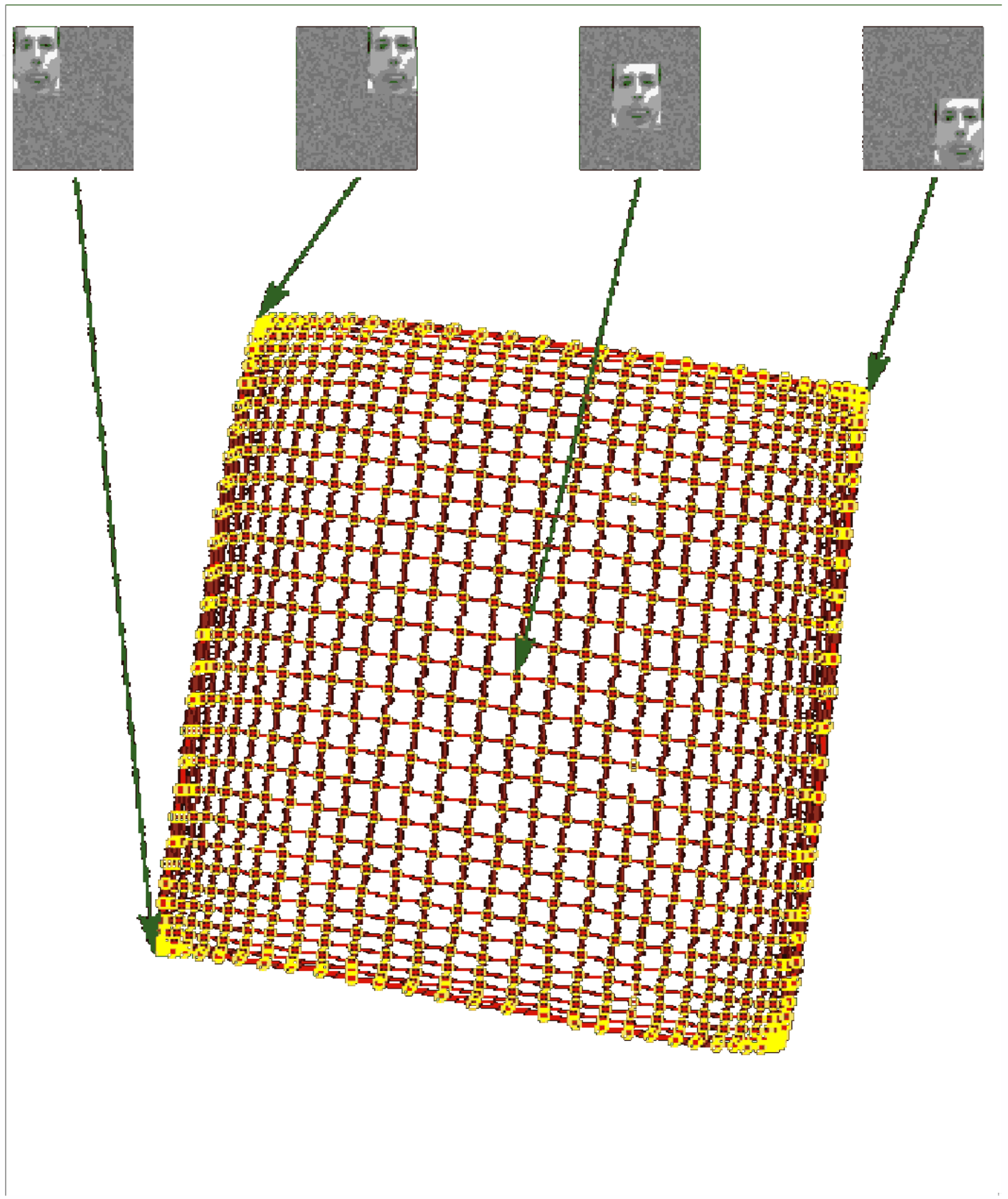
Translated faces

$m = 961$
images

$k = 4$
nearest
neighbors

$D = 3009$
pixels

$d = 2$
manifold



國立交通大學
National Chiao Tung University

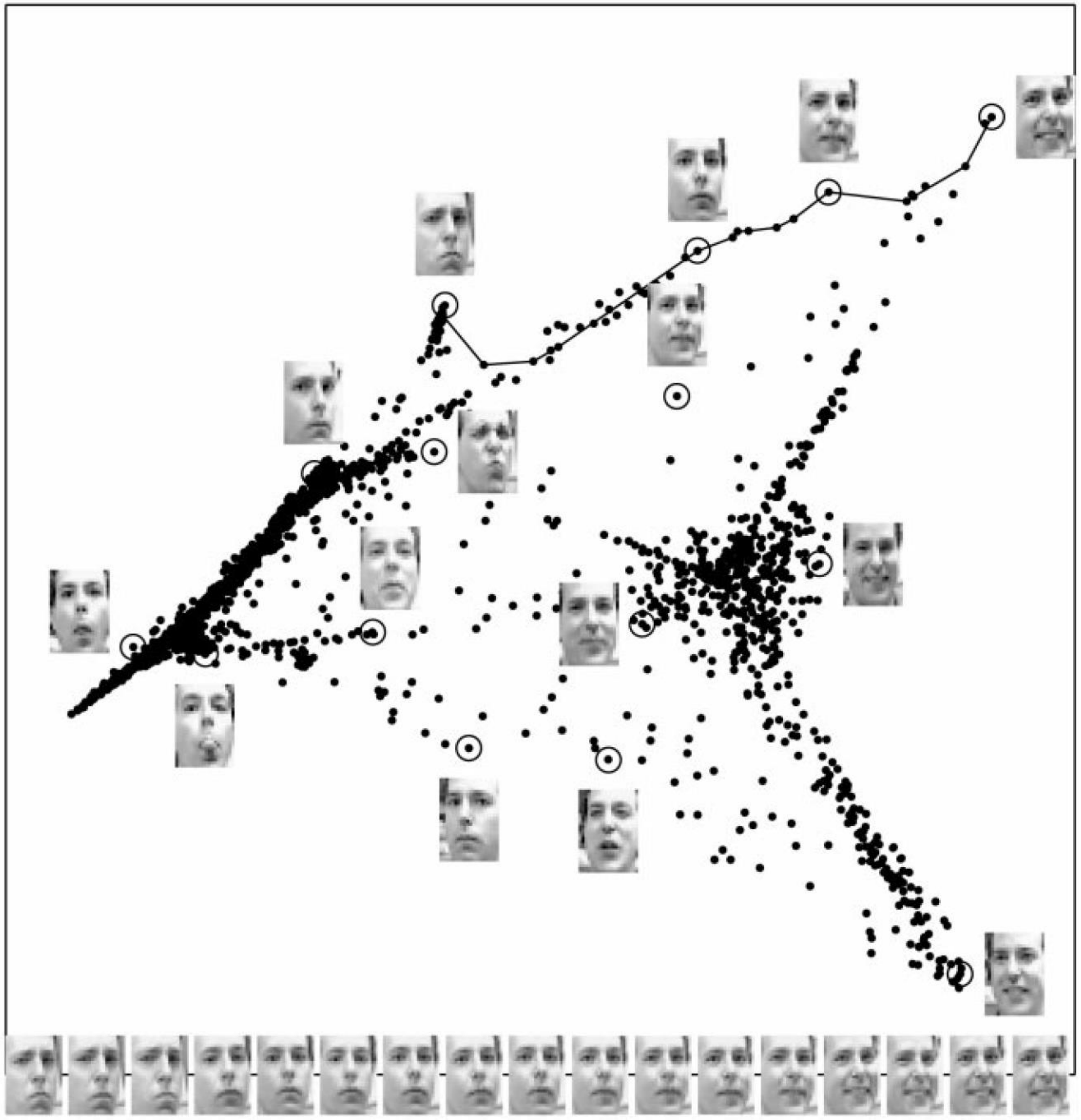
Pose and expression

$m = 1965$
images

$k = 12$
nearest
neighbors

$D = 560$
pixels

$d = 2$
(shown)



National Chiao Tung University

Isomap vs. LLE

- Similarities
  - Nonlinear dimensionality reduction for manifold learning
  - Graph-based, spectral method
  - No local minima
  - Does not estimate dimensionality

- Differences
  - Constructs dense vs. sparse matrices
  - Preserves distances vs. local geometric structure

References

- PCA
  - Chapter 12.1 in the PRML textbook
- MDS
  - https://en.wikipedia.org/wiki/Multidimensional_scaling
- Isomap
  - J. B. Tenenbaum, V. de Silva and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science 290 (5500): 2319-2323, 2000.
- LLE
  - S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science 290 (5500): 2323-2326 2000.

Thank You for Your Attention!

Yen-Yu Lin (林彥宇)

Email: lin@cs.nctu.edu.tw

URL: https://www.cs.nctu.edu.tw/members/detail/lin

國立交通大學
National Chiao Tung University