Computer Vision:
Principal Component Analysis (PCA)

Siheng Chen 陈思衡

Homework 3 is online; due on June 13th

Please well choose your teammates

Principal Component Analysis (PCA)

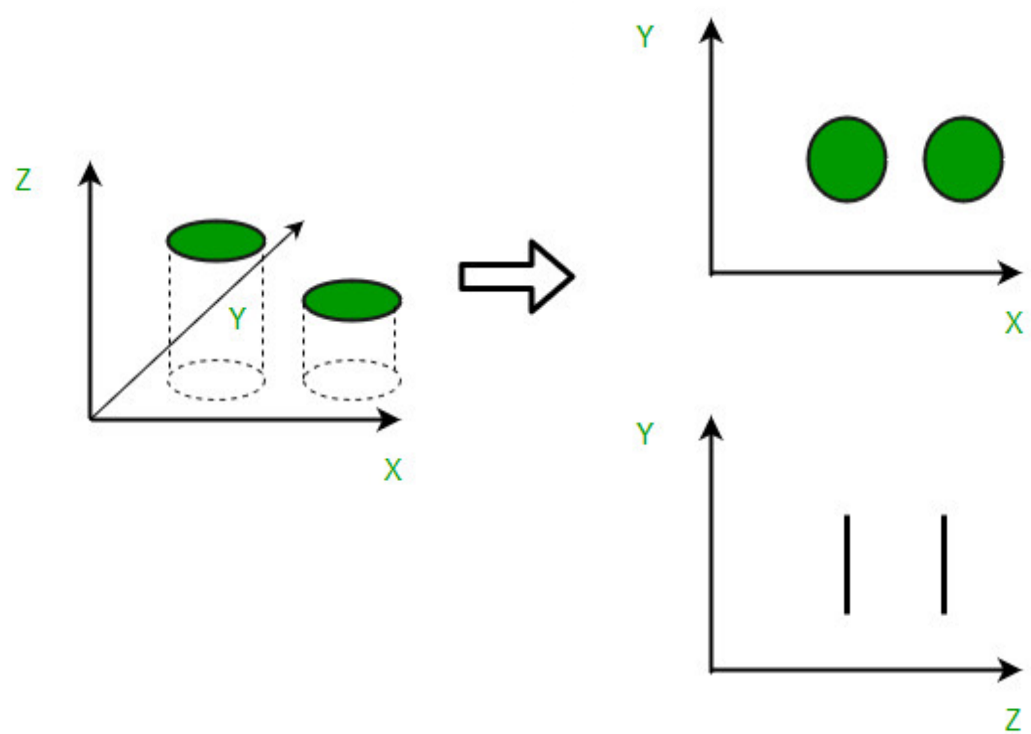
Unsupervised learning: Dimensionality reduction

Application: Image feature extraction

Method: PCA, Graph-based dimensionality reduction

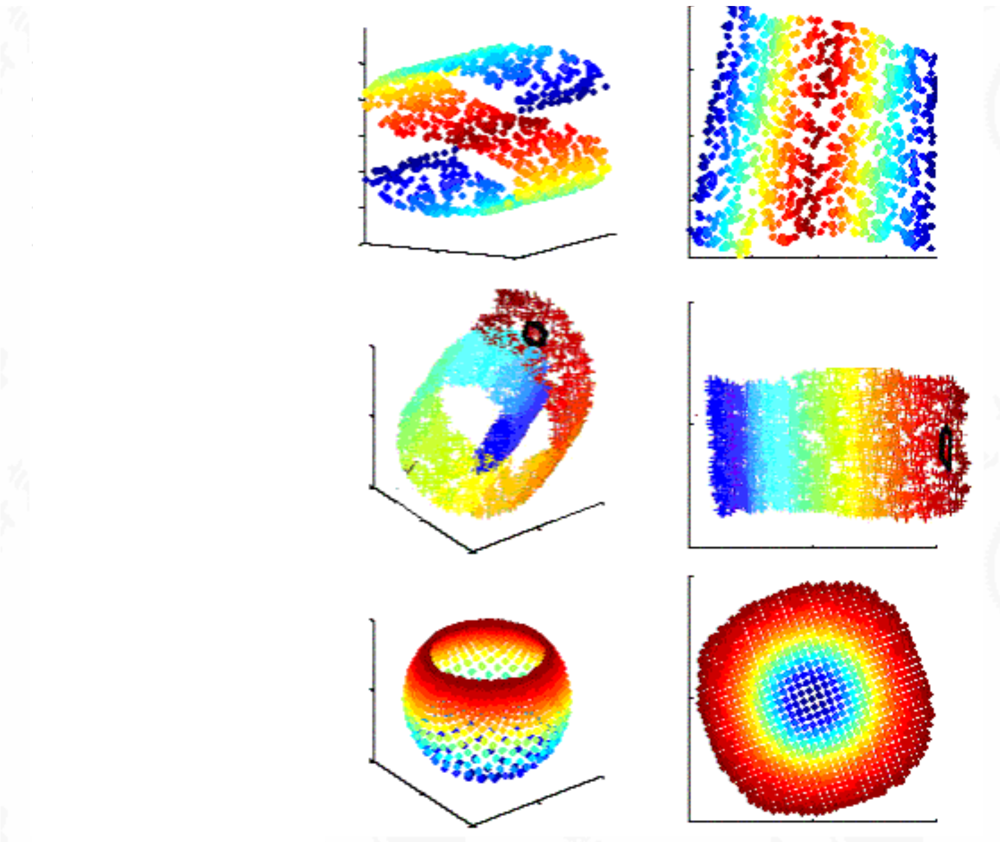
Dimensionality reduction

Transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its **intrinsic dimension**



Dimensionality reduction

Transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its **intrinsic dimension**

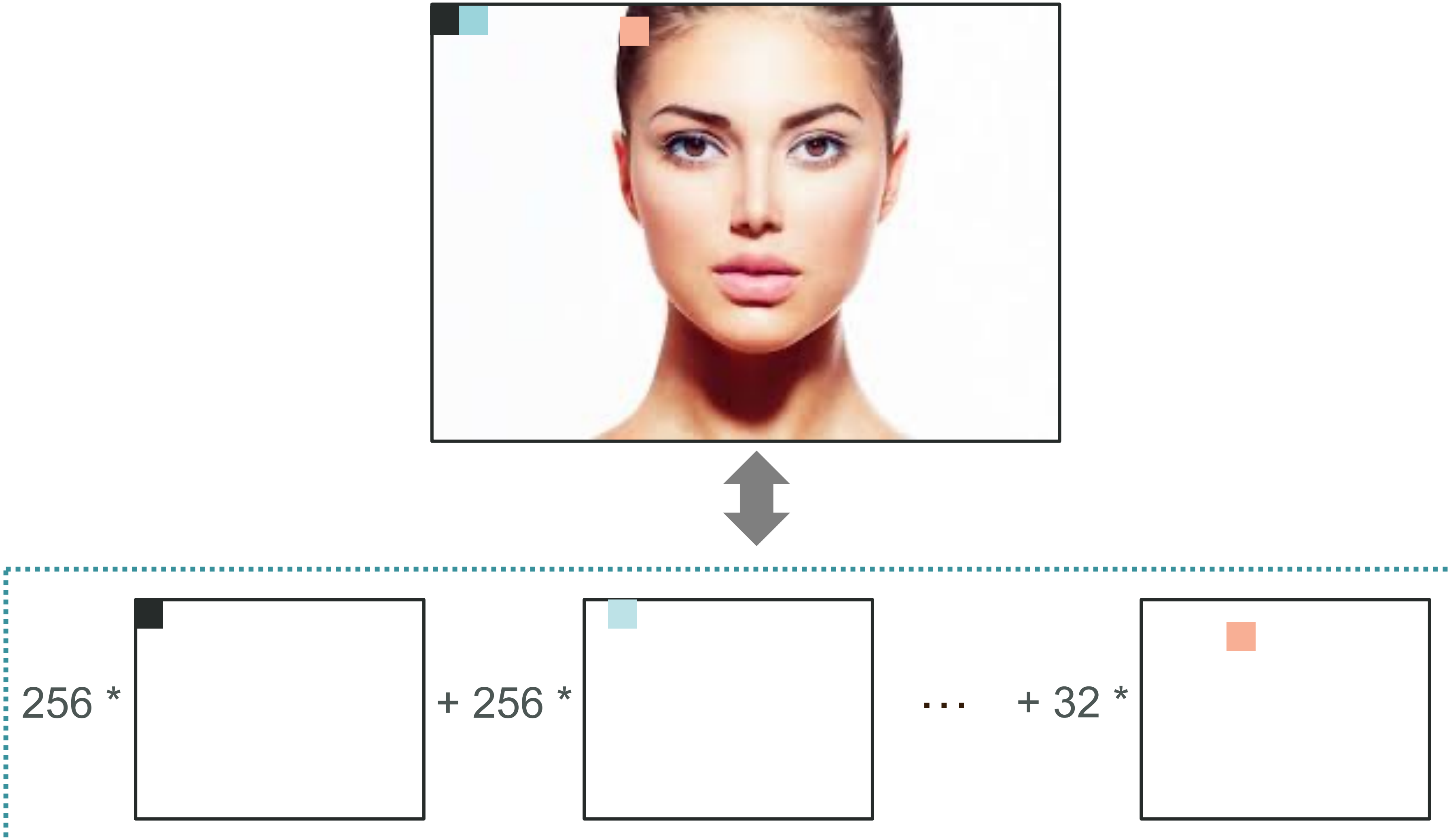


Dimensionality reduction

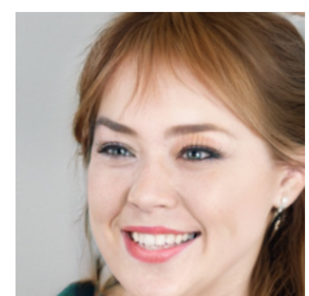
1 million pixels => feature space with 1 million dimensions

Shift by 1 pixel?

Dimensionality reduction



Dimensionality reduction

0.2 *  + 0.2 *  ··· + 0.2 * 

https://globalnews.ca/news/4970640/artificial-intelligence-generate-human-face/

Dimensionality reduction

$\mathbf{x}$

$\mathbf{d_1}$ $\mathbf{d_2}$ $\mathbf{d_n}$

0.2 *  + 0.2 *  $\cdots$ + 0.2 * 

https://globalnews.ca/news/4970640/artificial-intelligence-generate-human-face/

Dimensionality reduction

$$\mathbf{x} = \mathbf{D}\mathbf{a} = \begin{bmatrix}\mathbf{d_1} & \mathbf{d_2} & \cdots & \mathbf{d_n}\end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

image

dictionary

basis vector (atom)

coefficient

How to obtain this dictionary?

DEFINITION 2.34 (BASIS) The set of vectors $\Phi = \{\varphi_k\}_{k\in\mathcal{K}} \subset V$, where $\mathcal{K}$ is finite or countably infinite, is called a *basis* for a normed vector space V when

(i) it is *complete* in V, meaning that, for any $x \in V$, there is a sequence $\alpha \in \mathbb{C}^{\mathcal{K}}$ such that

$$x = \sum_{k\in\mathcal{K}} \alpha_k \varphi_k; \tag{2.87}$$

and

(ii) for any $x \in V$, the sequence α that satisfies (2.87) is unique.

Dimensionality reduction

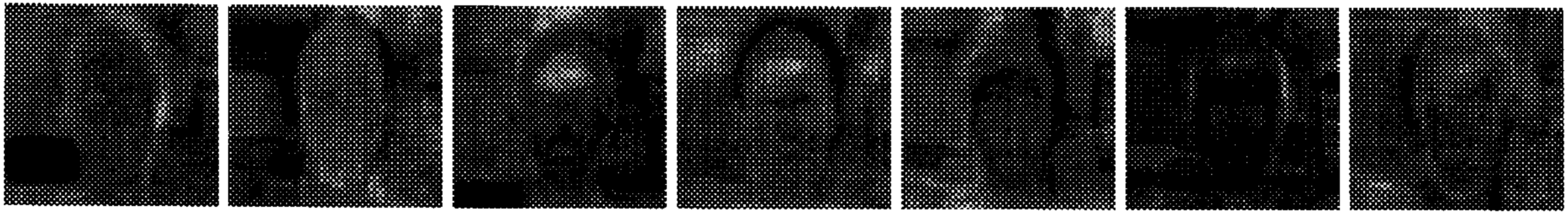


Figure 2: Seven of the eigenfaces calculated from the images of Figure 1, without the background removed.

Figure 2. Seven of the eigenfaces calculated from the input images of Figure 1.

Matthew A. Turk, Alex P. Pentland, Face recognition using eigenfaces, CVPR 1991

Principal Component Analysis (PCA)

Unsupervised learning: Dimensionality reduction

Application: Image feature extraction

Method: PCA, Graph-based method

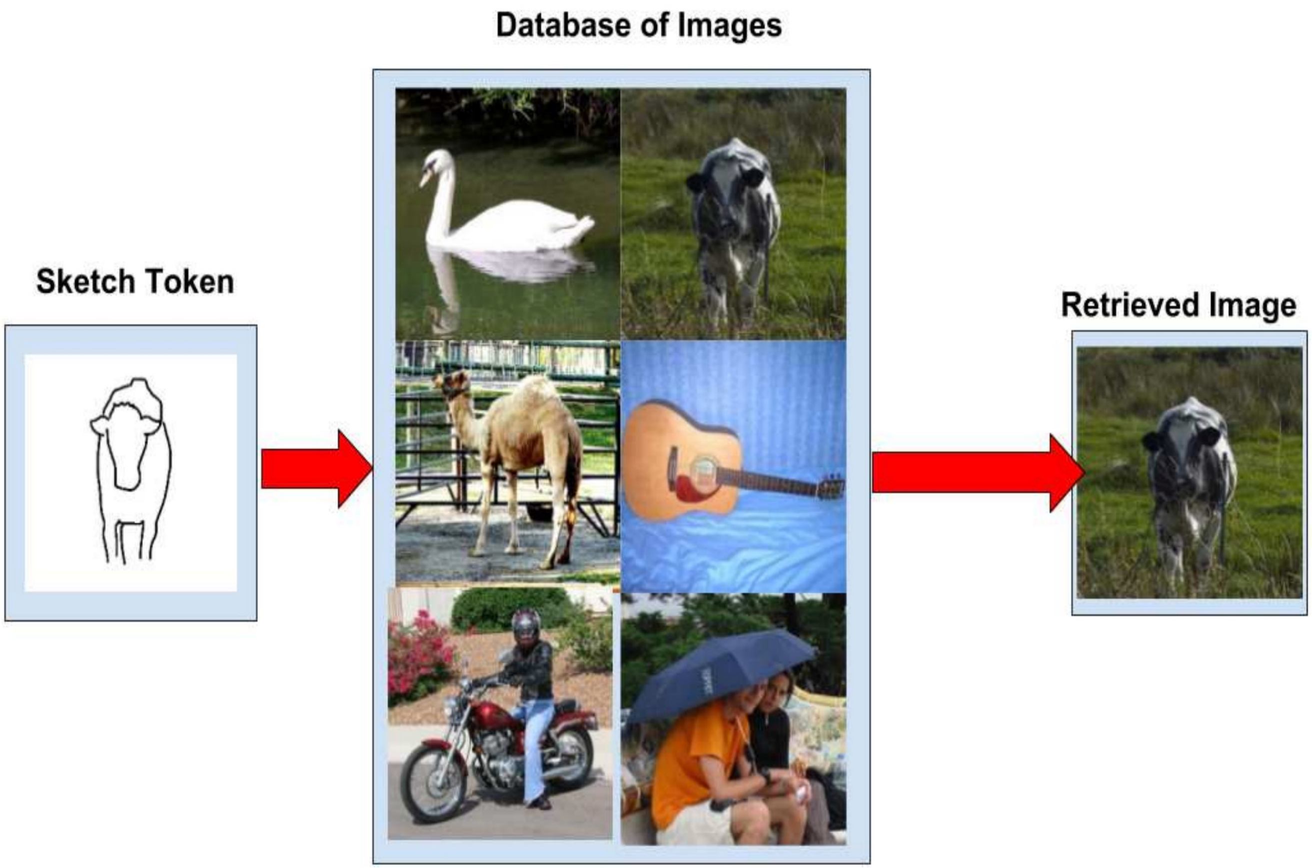
Application: Image feature extraction

Image classification

https://www.image-net.org/

Application: Image feature extraction

Image retrieval



Application: Image feature extraction

Image-based tasks

feature extraction

downstream task

Application: Image feature extraction

Image-based tasks

→ feature extraction →

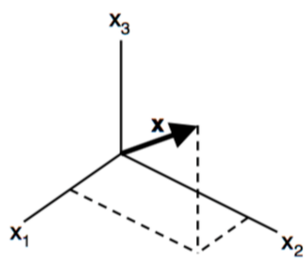
dimension reduction

Handcrafted feature
- Facial landmarks
- Color
- …

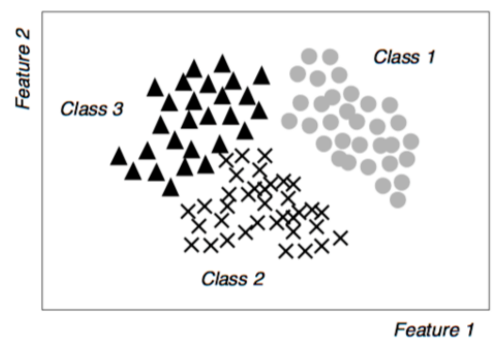
High-level feature
- **PCA**
- Dictionary learning
- …

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Feature vector



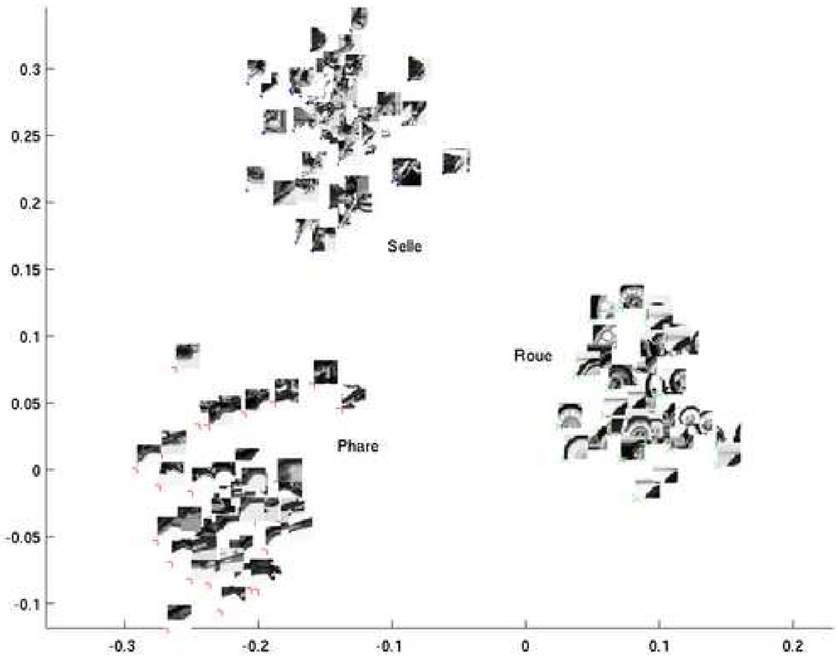
Feature space (3D)

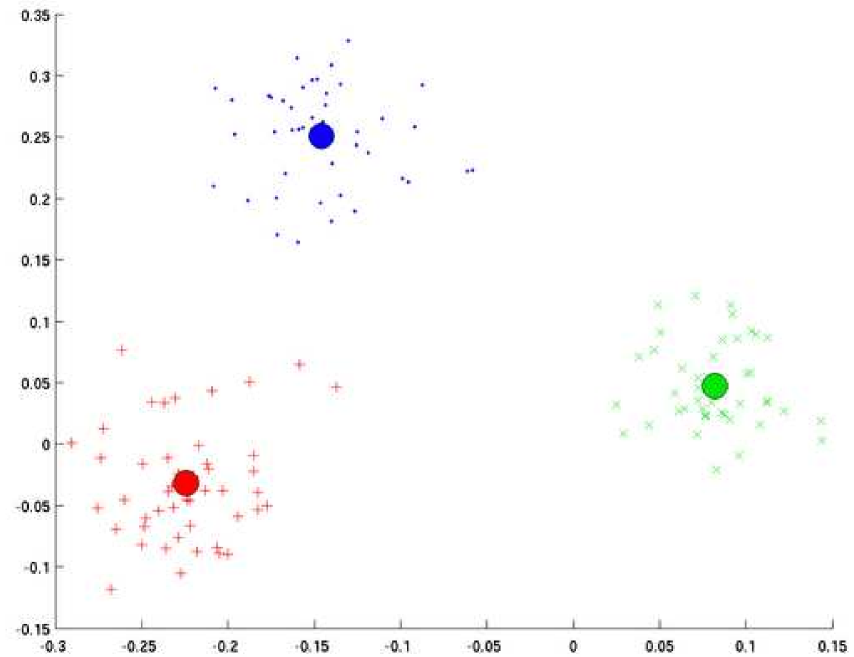


Scatter plot (2D)

Application: Image feature extraction

Image classification / retrieval





Principal Component Analysis (PCA)

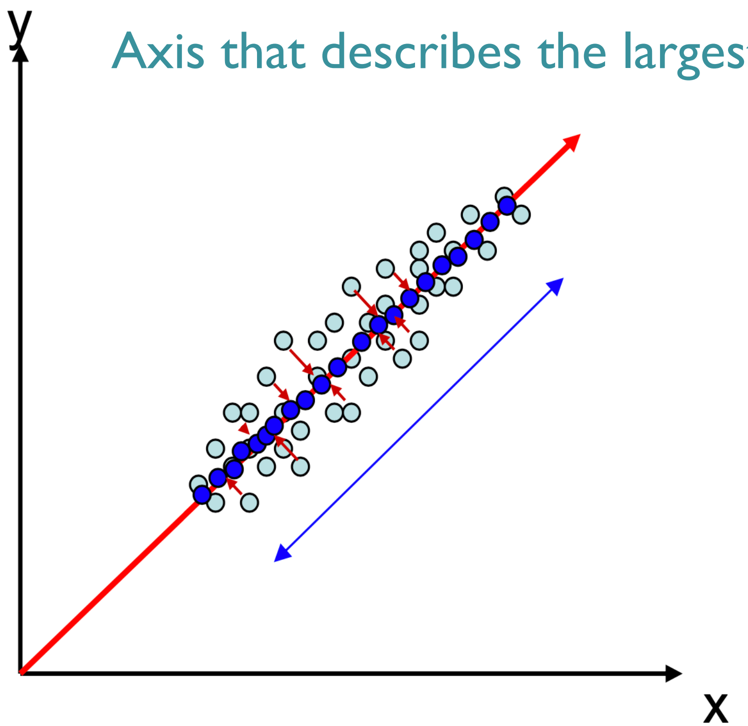
Unsupervised learning: Dimensionality reduction

Application: Image feature extraction

Method: PCA, Graph-based method

Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.



- d-dimensional vector
- N samples

$$\text{maximize} \quad \mathrm{Var}\,(\omega^T x)$$

$$\text{subject to:} \quad \|\,\omega\| = 1$$

Method: PCA

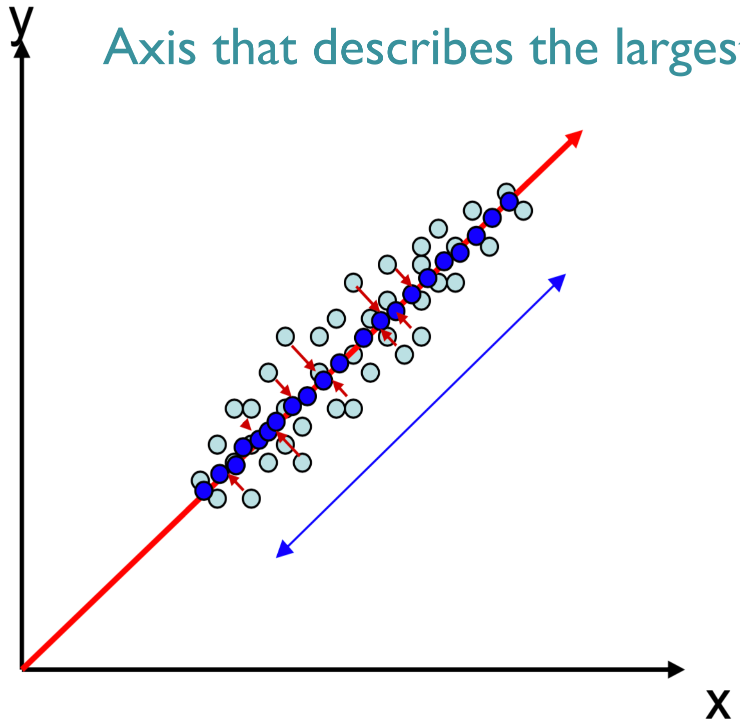
Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

$$
\begin{aligned}
J(\omega) &= Var(\omega^T x) \\
&= E(\omega^T x - \omega^T \mu)^2 \\
&= E(\omega^T x - \omega^T \mu)(x^T \omega - \mu^T \omega) \\
&= \omega^T E(x - \mu)(x - \mu)^T \omega \\
&= \omega^T \underset{d*d}{\Sigma}\ \underset{d*1}{\omega}
\end{aligned}
$$

Where covariance matrix $\Sigma = E(x - \mu)(x - \mu)^T$

Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.



$$\text{maximize} \quad \omega^T \Sigma \omega$$

$$\text{subject to:} \quad ||\omega|| = 1$$

Solution: Form Lagrangian optimization to take care of constraints, take derivative and set to zero to find ω vectors.

Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

Lagrange function $$L(\omega, \lambda) = \omega^{\mathbf{T}} \Sigma \omega - \lambda(\omega^{\mathbf{T}} \omega - 1)$$

Lagrange multiplier

$$\frac{\partial L(\omega, \lambda)}{\partial \omega} = 2\Sigma\omega - 2\lambda\omega = 0$$

Eigen-decomposition $$\Sigma\omega = \lambda\omega$$

- Should we to the full eigendecomposition?
- Which eigen-vector and eigenvalue pairs should we use?

Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

$$\boxed{\begin{aligned} &\text{maximize} \quad \omega^T \Sigma \omega \\ &\text{subject to:} \quad \|\omega\| = 1 \end{aligned}} = \lambda$$

Eigen-decomposition $\Sigma\omega = \lambda\omega$

We only need the eigenvector that corresponds to the largest eigenvalue!

Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

$$\text{maximize} \quad \omega^T \Sigma \omega$$

$$\text{subject to:} \quad \| \omega \| = 1$$

From a single dimension to multiple dimensions

Vector spaces

THEOREM 2.39 (ORTHONORMAL BASIS EXPANSIONS) Let $\Phi = \{\varphi_k\}_{k\in\mathcal{K}}$ be an orthonormal basis for a Hilbert space H. The unique expansion with respect to Φ of any x in H has expansion coefficients

analysis

$$\alpha_k = \langle x, \varphi_k\rangle \qquad \text{for } k \in \mathcal{K}, \quad \text{or}, \tag{2.93a}$$

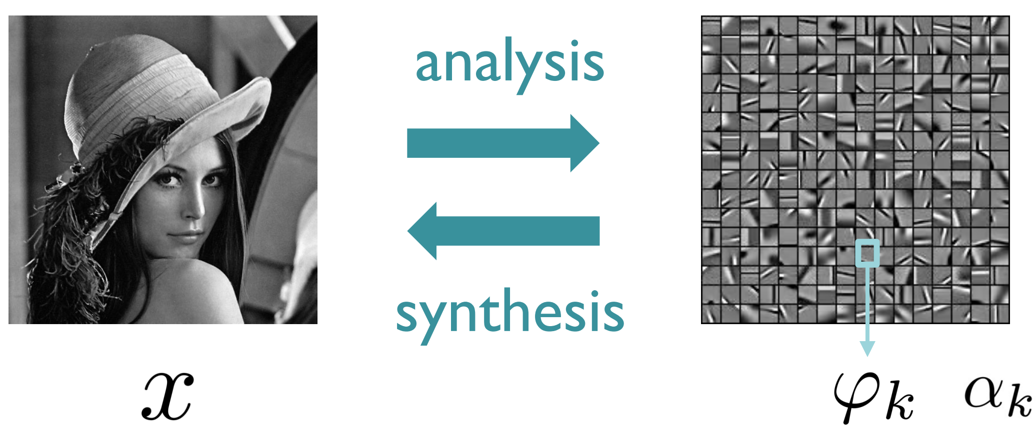
$$\alpha = \Phi^* x. \tag{2.93b}$$

Synthesis with these coefficients yields

synthesis

$$x = \sum_{k\in\mathcal{K}} \langle x, \varphi_k\rangle \varphi_k \tag{2.94a}$$

$$= \Phi\alpha = \Phi\Phi^* x. \tag{2.94b}$$



Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

Centering $\qquad \widetilde{\mathbf{x}} = \mathbf{x} - \mu$

Construct eigenbasis $\qquad \underbrace{\Sigma}_{d*d} = \mathbb{E}(\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^T) = \mathbf{V}\underbrace{\Lambda}_{d*d}\mathbf{V}^T$

Projection (analysis) $\qquad \underbrace{\mathbf{p}}_{d*1} = \mathbf{V}^T\widetilde{\mathbf{x}}$

Reconstruction (synthesis) $\qquad \widetilde{\mathbf{x}} = \mathbf{V}\mathbf{p}$

Method: PCA

Dataset

- CMU's AMP Lab facial Expression database
- 13 people.
- Images are 64x64 cropped and centered facial images.
- Variations are due to varying expressions in the video sequence.
- 75 images in each person's video sequence

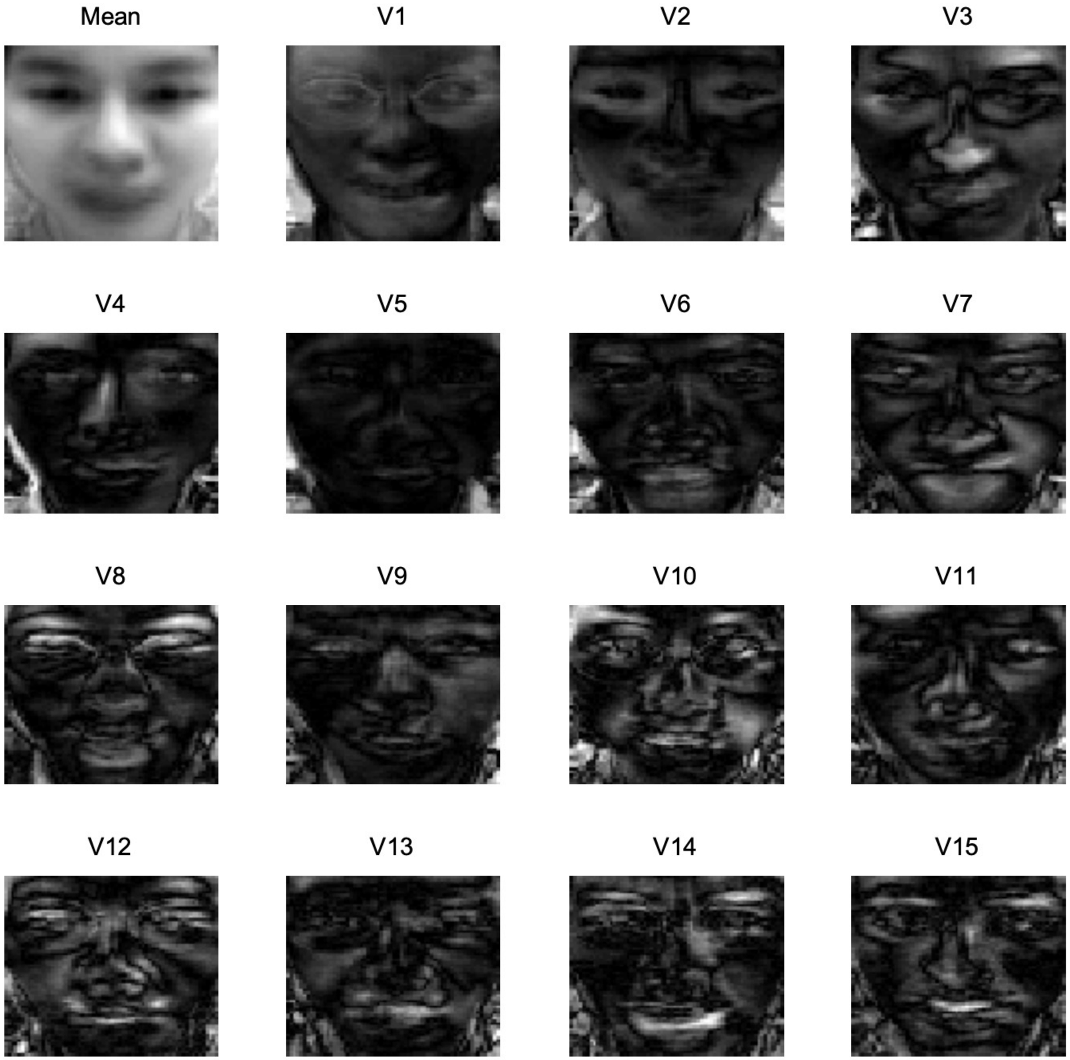
Method: PCA

Experimental setup

- Take the first 5 images per person and use them as training data
- Total of 5x13=65 training images
- Do PCA and compute basis eigenvectors.
- Measure Reconstruction ability
- Perform Dimensionality Reduction (to 3d, i.e. only use a few eigenvectors and look at how data clusters in this 3D linear subspace.

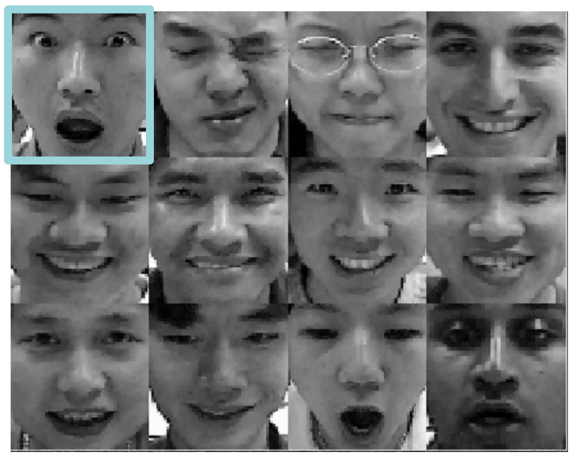
Method: PCA

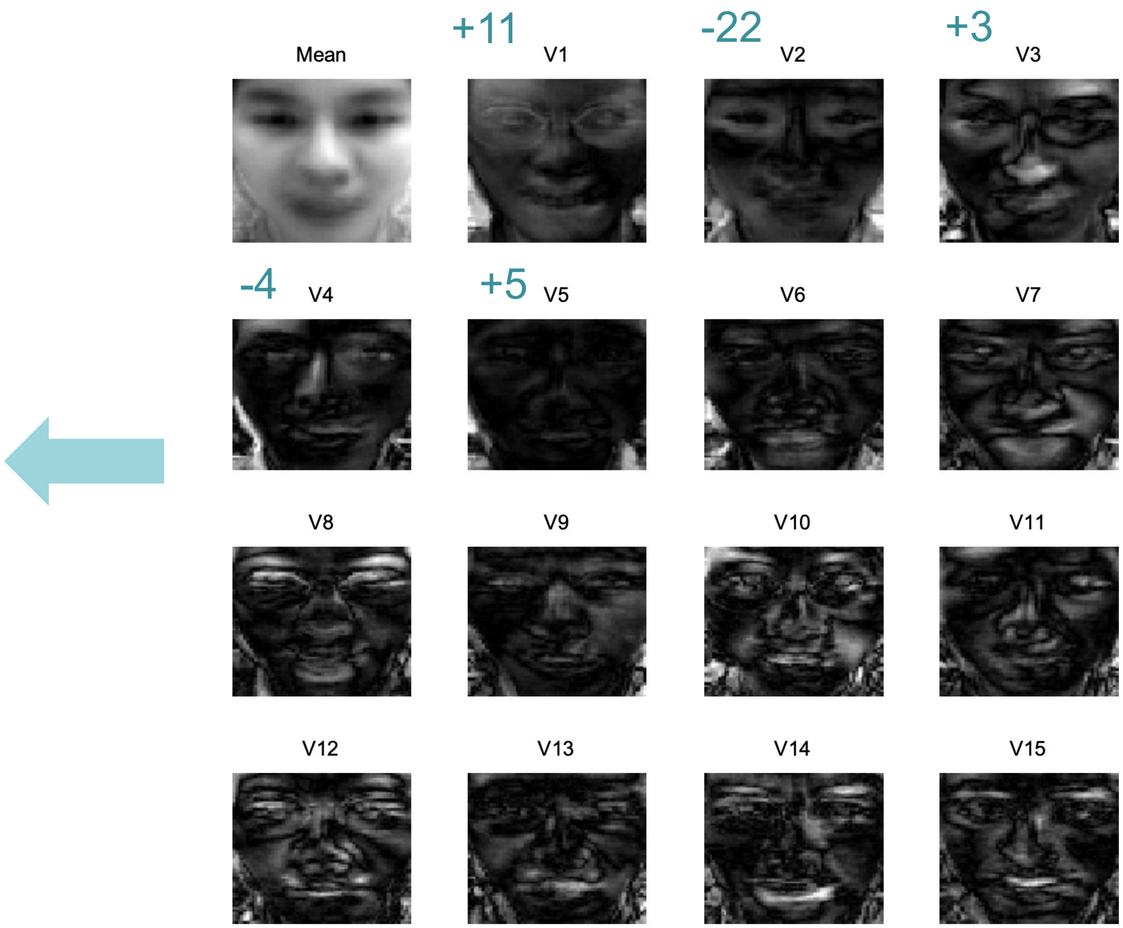
Eigenfaces



Method: PCA

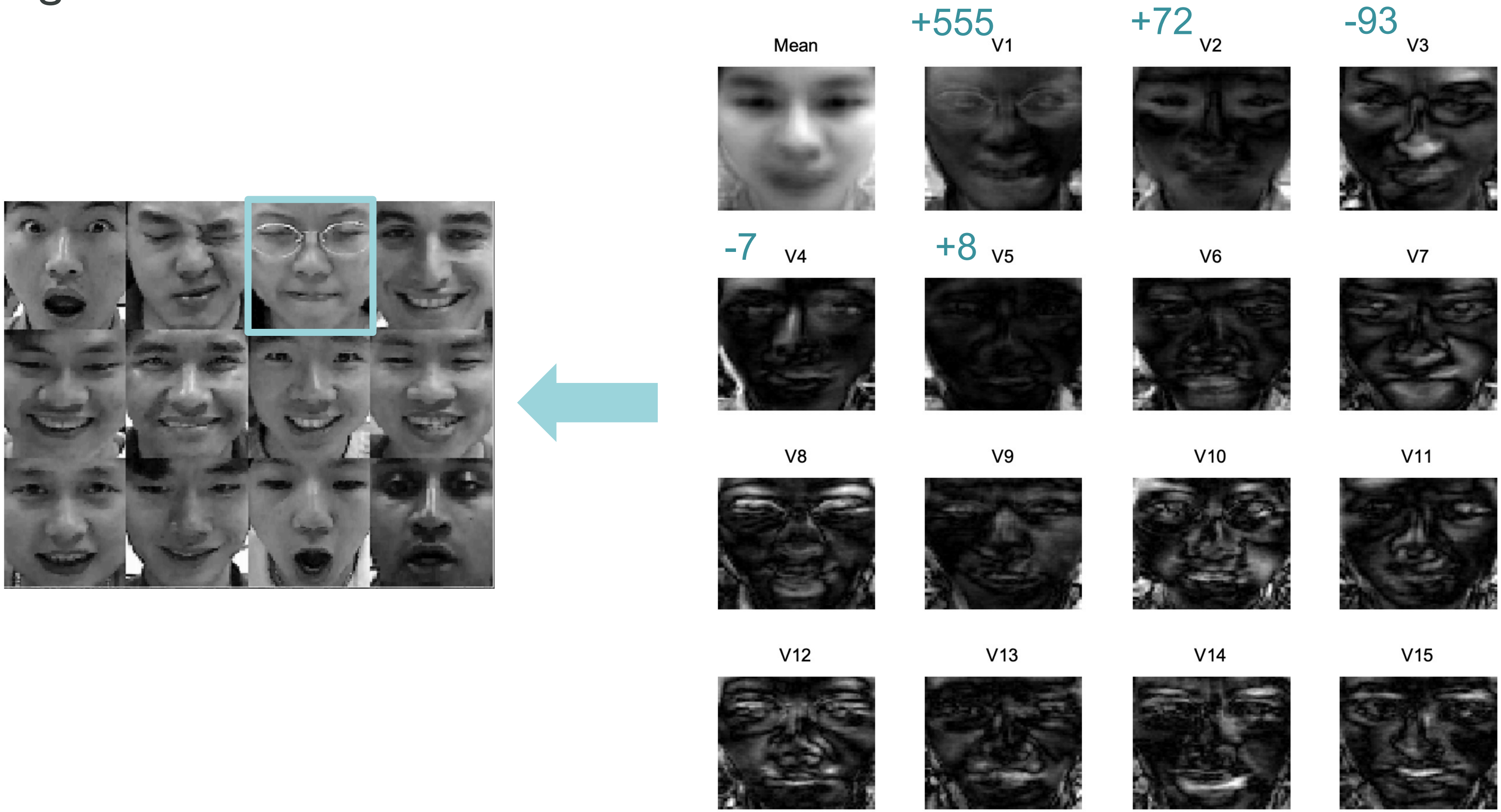
Eigenfaces





Method: PCA

Eigenfaces



Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

Centering $$\widetilde{\mathbf{x}} = \mathbf{x} - \mu$$

Construct eigenbasis $$\underbrace{\Sigma}_{d*d} = \mathbb{E}(\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^T) = \mathbf{V}\underbrace{\Lambda}_{d*d}\mathbf{V}^T$$

$$\Lambda \rightarrow [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d]$$

Projection (analysis) $$\underbrace{\mathbf{p}}_{m*l} = \mathbf{V}_m^T\widetilde{\mathbf{x}}$$

Reconstruction (synthesis) $$\underbrace{\widetilde{\mathbf{x}}'}_{d*l} = \underbrace{\mathbf{V}_m}_{d*m}\underbrace{\mathbf{p}}_{m*l}$$

Method: PCA

Representation

If we use only 1 eigenvector, MSE=1233

Method: PCA

Representation

If we use 2 eigenvectors, MSE=1027

Method: PCA

Representation

If we use 3 eigenvectors, MSE=758

Method: PCA

Representation

If we use 20 eigenvectors, MSE=87

Method: PCA

Representation

If we use all 64 eigenvectors, MSE=0

Method: PCA

Core idea: We want to find projections of data (i.e. direction vectors that we can project the data on to) that describe the maximum variation.

Centering $$\widetilde{\mathbf{x}} = \mathbf{x} - \mu$$

Construct eigenbasis $$\underbrace{\Sigma}_{d*d} = \mathbb{E}(\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^T) = \mathbf{V}\underbrace{\Lambda}_{d*d}\mathbf{V}^T$$

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d]$$

Projection (analysis) $$\boxed{\underbrace{\mathbf{p}}_{m*1} = \mathbf{V}_m^T\widetilde{\mathbf{x}}}$$

Useful features

Reconstruction (synthesis) $$\underbrace{\widetilde{\mathbf{x}}'}_{d*1} = \underbrace{\mathbf{V}_m}_{d*m}\underbrace{\mathbf{p}}_{m*1}$$

Method: PCA

Face recognition

PCA → K-NN → Lenna!

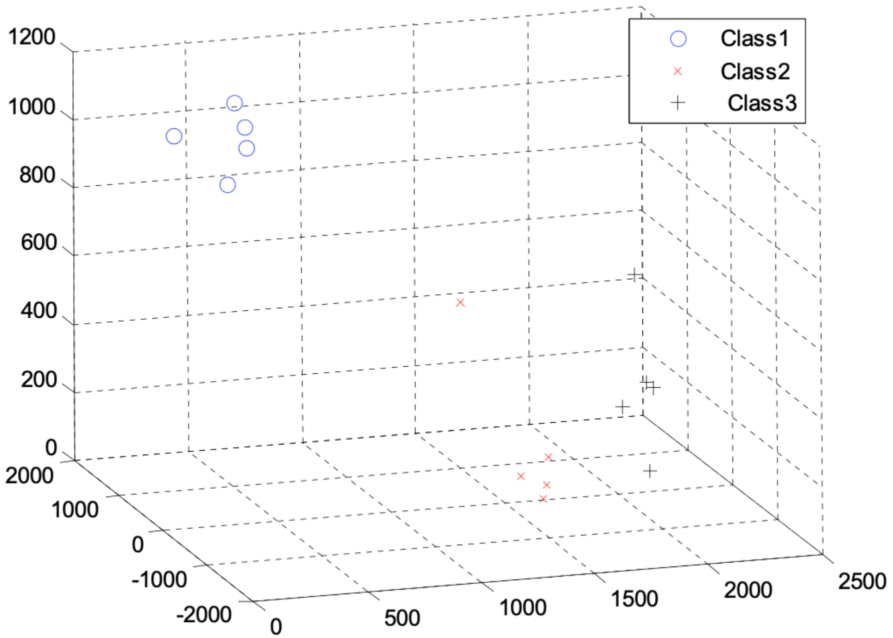
- Perform PCA and then only kept the 3 most dominant eigenvectors (i.e. with the 3 largest eigenvalues)

- Project each 64x64=4096 dimensional training image onto this basis to get a 3-dimensional vector. That 3d vector represents each training sample in a 3d subspace.

Method: PCA

Face recognition

The 3D linear subspace containing the projected training samples from class 1, 2 and 3

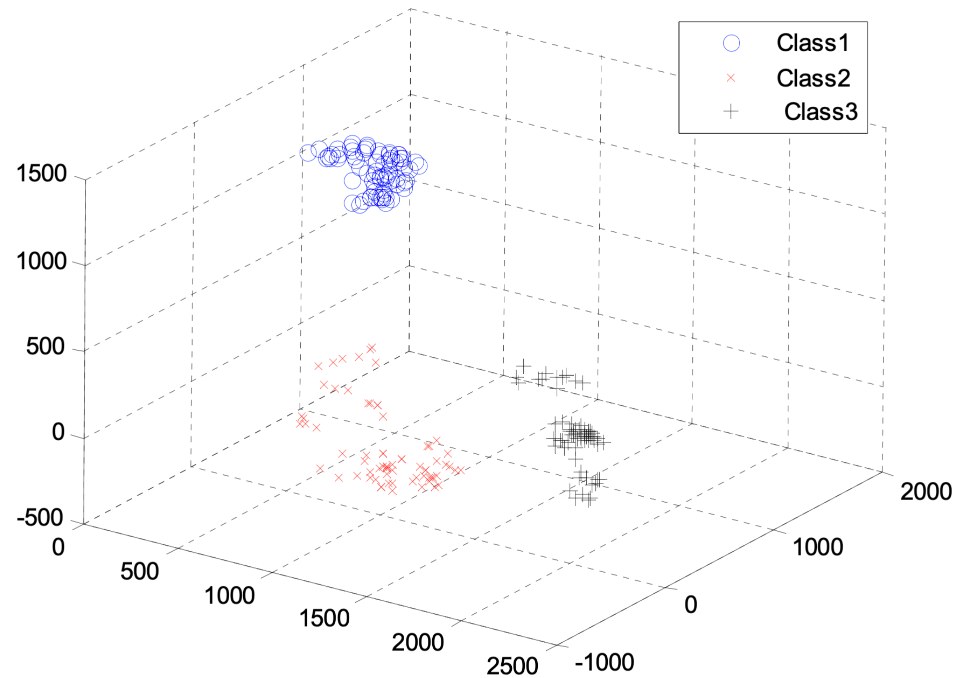


Method: PCA

Face recognition

The 3D linear subspace containing the projected training samples from class 1, 2 and 3

- Project each test face to get a 3d-vector.
- Use kNN classifier



How to choose the dimension?

Method: PCA

How to choose the dimension?

Centering $$\widetilde{\mathbf{x}} = \mathbf{x} - \mu$$

Construct eigenbasis $$\underset{d*d}{\Sigma} = \mathbb{E}(\widetilde{\mathbf{x}}\widetilde{\mathbf{x}}^T) = \mathbf{V}\underset{d*d}{\Lambda}\mathbf{V}^T$$

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_d]$$

Projection (analysis) $$\underset{m*1}{\mathbf{p}} = \mathbf{V}_m^T\widetilde{\mathbf{x}}$$

Reconstruction (synthesis) $$\underset{d*1}{\widetilde{\mathbf{x}}'} = \underset{d*m}{\mathbf{V}_m}\underset{m*1}{\mathbf{p}}$$

Method: PCA

How to choose the dimension?

Reconstruction error. Use the eigenvectors with largest m eigenvalues satisfying the following ratio:

sum of the eigenvalues of the preserved eigenvectors

$$\text{Reconstruction ratio} = \frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

sum of all eigenvalues

Method: PCA

How to handle a few high-resolution images?

$$\underbrace{\Sigma}_{d*d}=E(x - \mu)(x - \mu)^T$$

depend on the data dimension

Given 10 images with 1-million-pixel, let's compute the covariance matrix?

Method: PCA

How to handle a few high-resolution images?

$$\underset{d*d}{\Sigma} = E(x - \mu)(x - \mu)^T = \underset{d*n}{X}\,\underset{n*d}{X^T}$$

Gram Matrix Trick! Compute the eigenvector much easier

$$\Sigma v = \lambda v$$

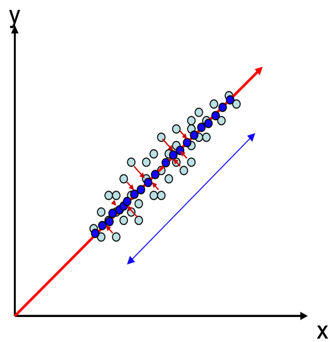
$$XX^T v = \lambda v \longrightarrow Xv' = \lambda v$$

Eigen-decomposition $\underbrace{\boxed{X^T X}}_{n*n} v' = \lambda v'$ $\qquad v' = X^T v$

depend on the number of samples

Method: PCA

motivation



optimization problem

maximize $\omega^T \Sigma \omega$

subject to: $||\omega|| = 1$

solver: eigen-decomposition

$$\Sigma = \mathbb{E}(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T) = \mathbf{V}\Lambda\mathbf{V}^T$$

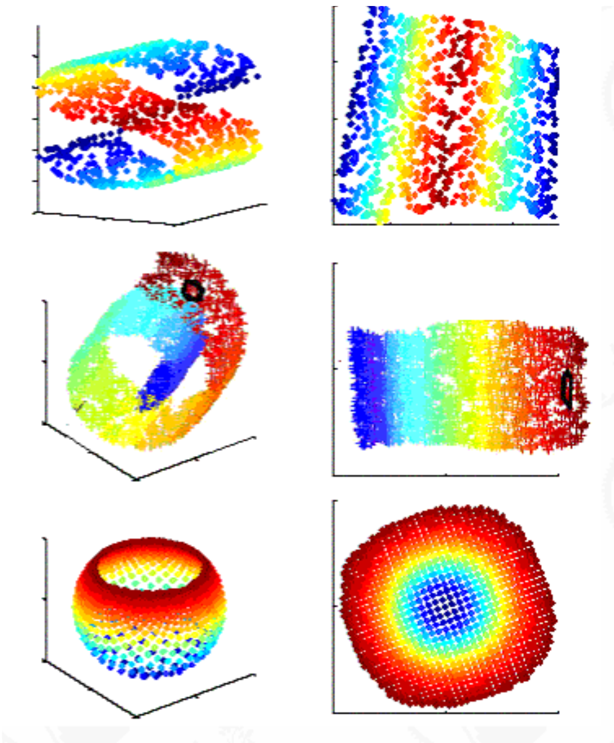
How to choose m?
reconstruction error

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

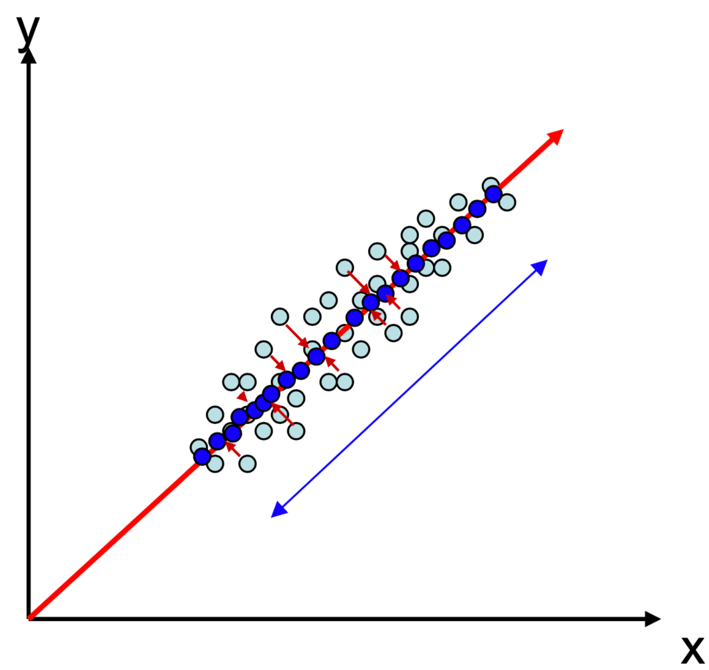
How to handle huge images?
Gram trick

$$XX^T v = \lambda v$$

Method: PCA

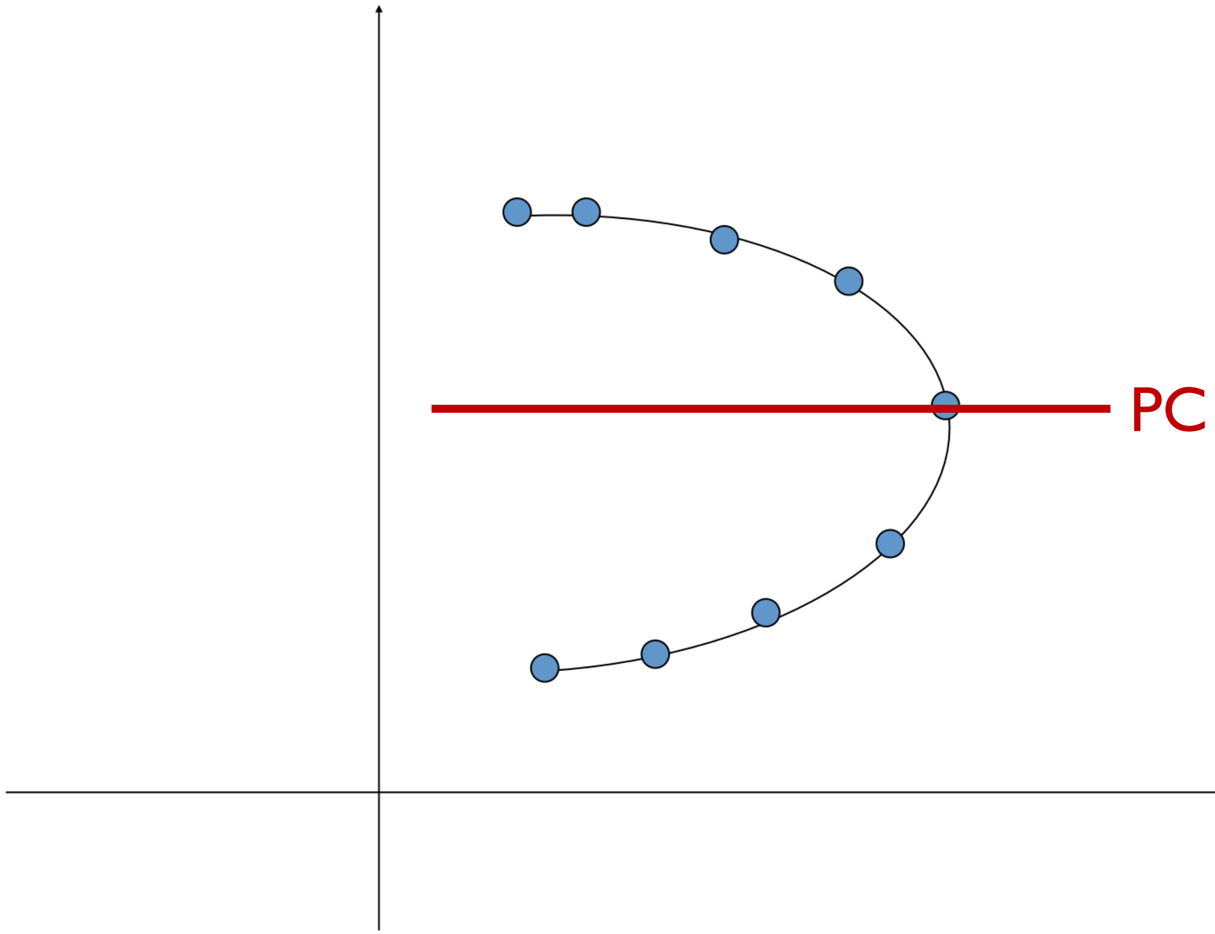


nonlinear

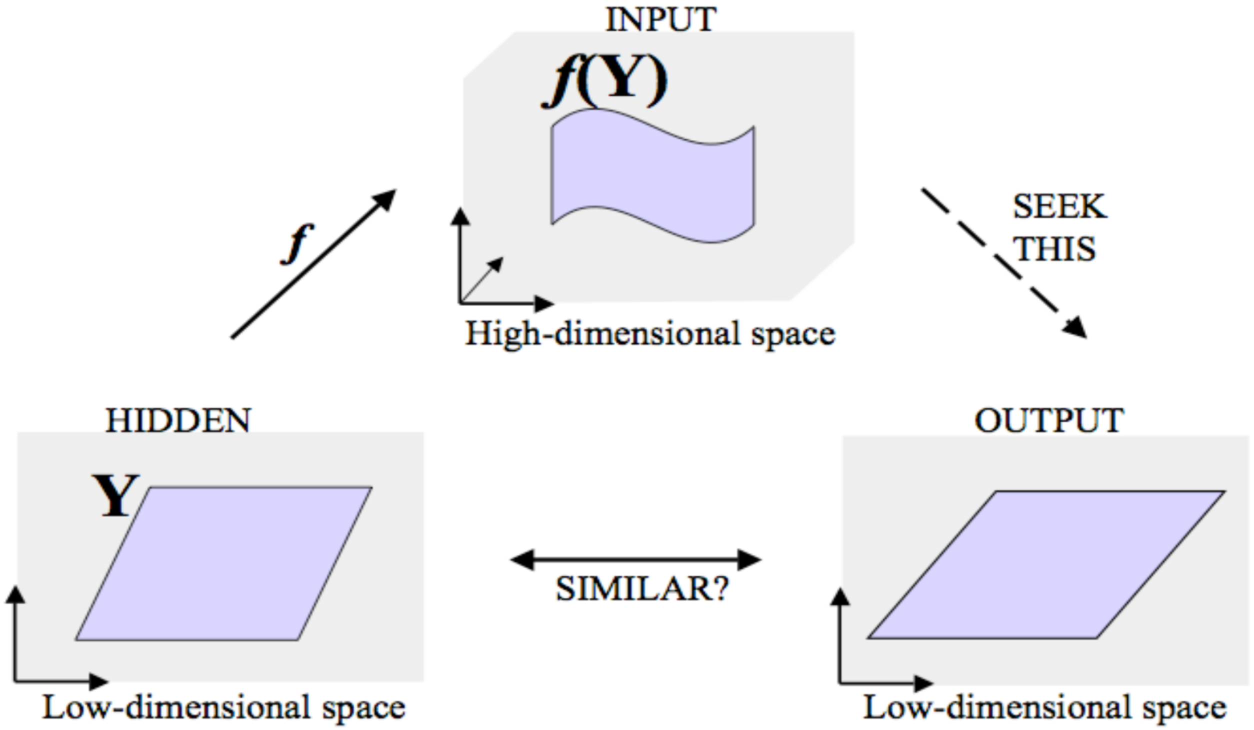


linear

Method: Nonlinear dimensionality reduction



Method: Nonlinear dimensionality reduction

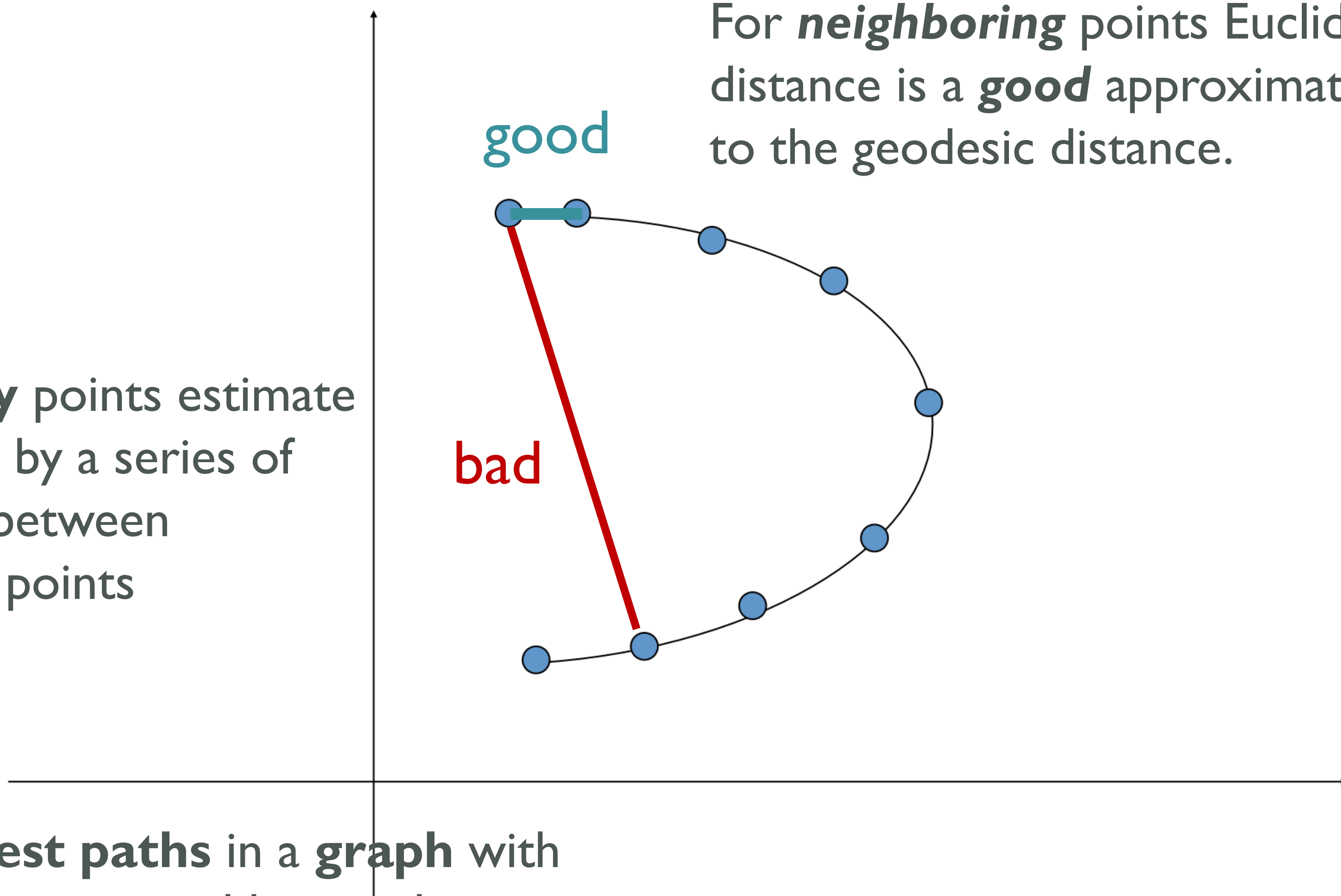


Method: Nonlinear dimensionality reduction

- Kernel PCA (Schoelkopf, et al, 98)
- ISOMAP (Tenenbaum, et al, 00)
- LLE (Roweis, Saul, 00)
- Laplacian Eigenmaps (Belkin, Niyogi, 01)
- Local Tangent Space Alignment (Zhang, Zha, 02)
- Hessian Eigenmaps (Donoho, Grimes, 02)
- Diffusion Maps (Coifman, Lafon, et al, 04)

Method: ISOMAP

For ***neighboring*** points Euclidean distance is a ***good*** approximation to the geodesic distance.

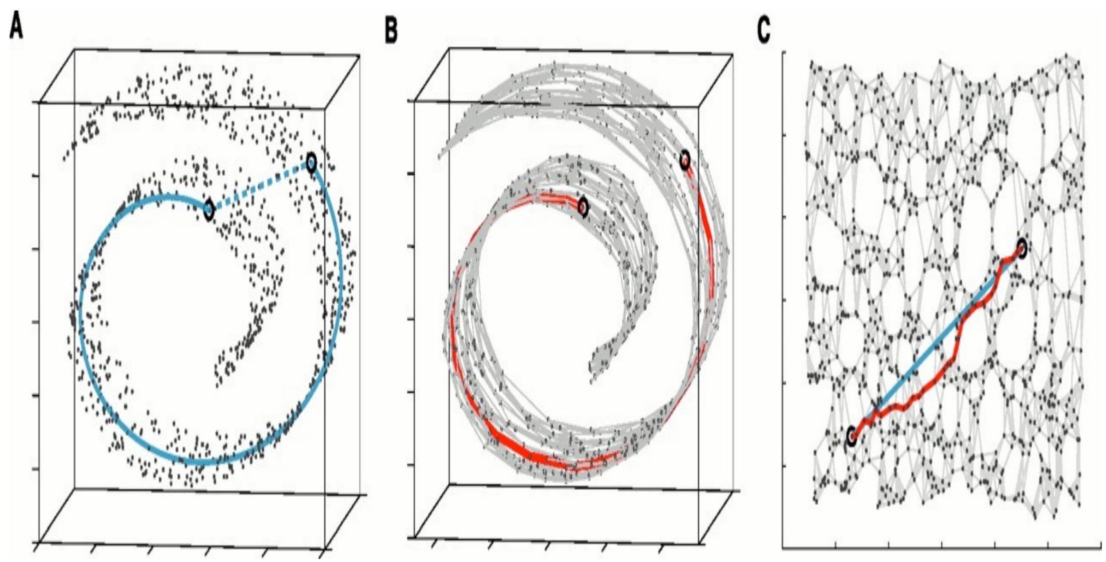


For **faraway** points estimate the distance by a series of short hops between neighboring points

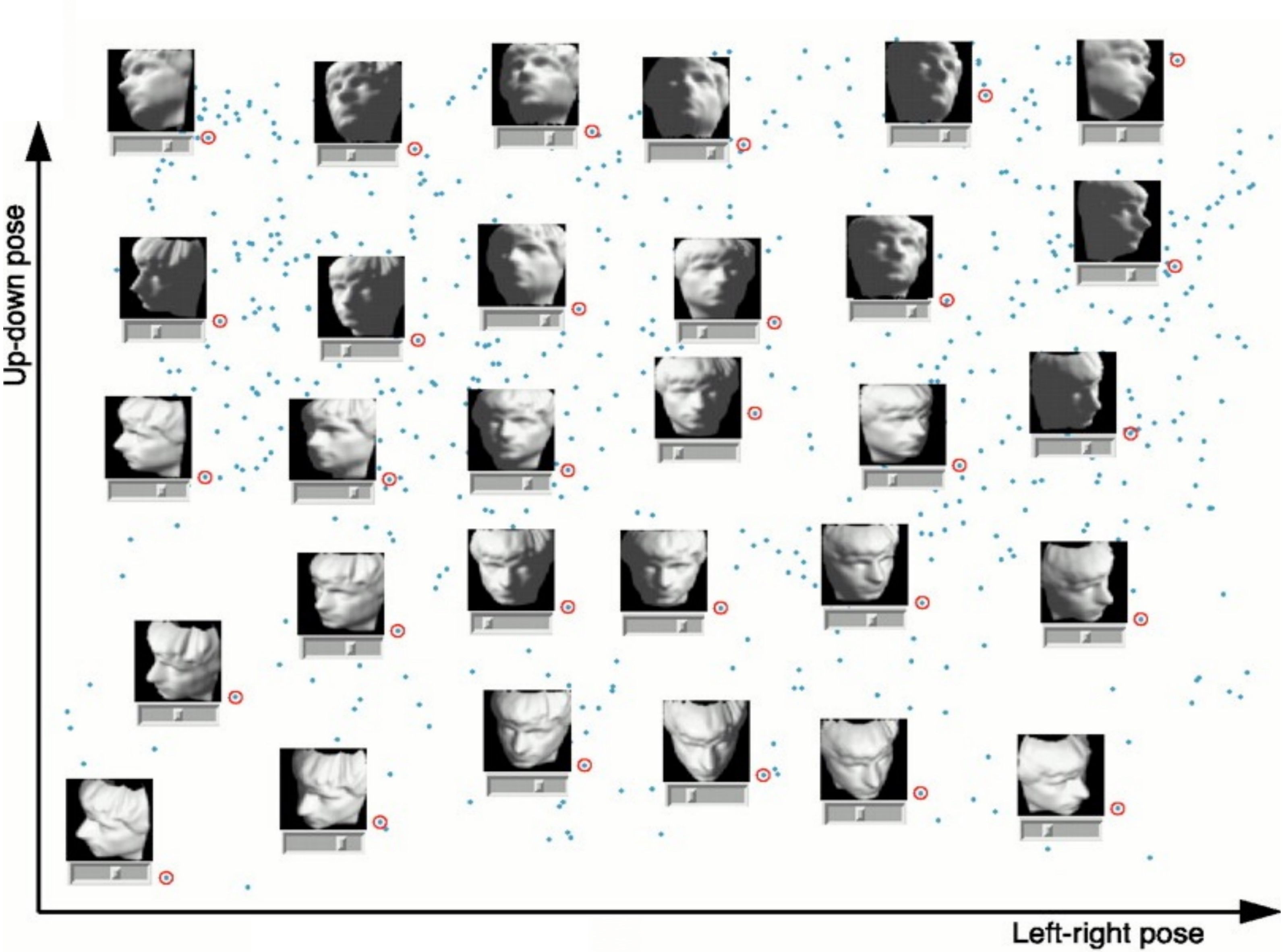
Find **shortest paths** in a **graph** with edges connecting neighboring data points

Method: ISOMAP

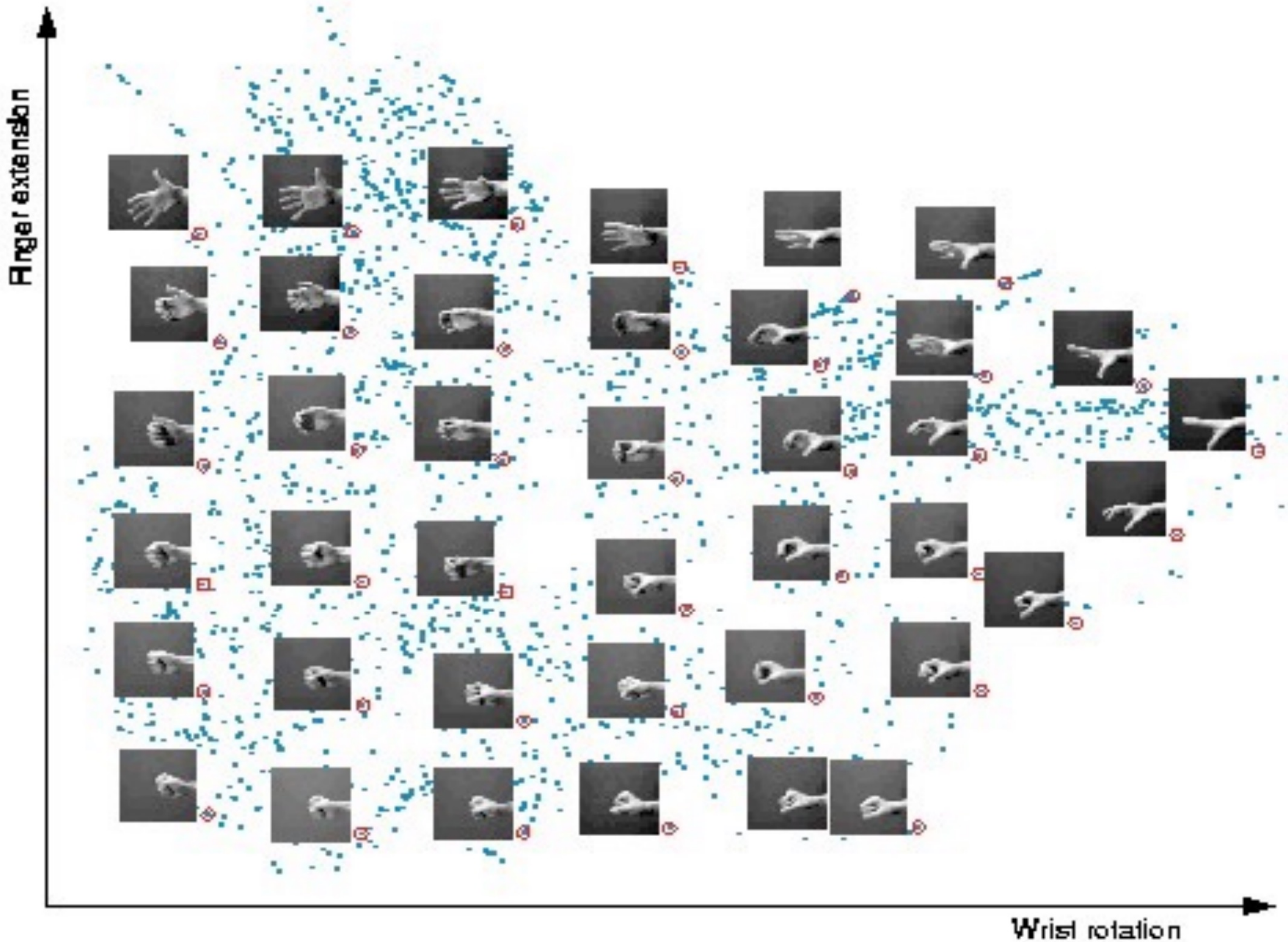
- Construct an n-by-n neighborhood graph
  - connecting points whose distances are within a fixed radius.
  - K nearest neighbor graph
- Compute the shortest path (geodesic) distances between nodes
- Construct a lower dimensional embedding



Method: ISOMAP



Method: ISOMAP



Application: Image feature extraction

Image-based tasks

feature extraction → downstream task

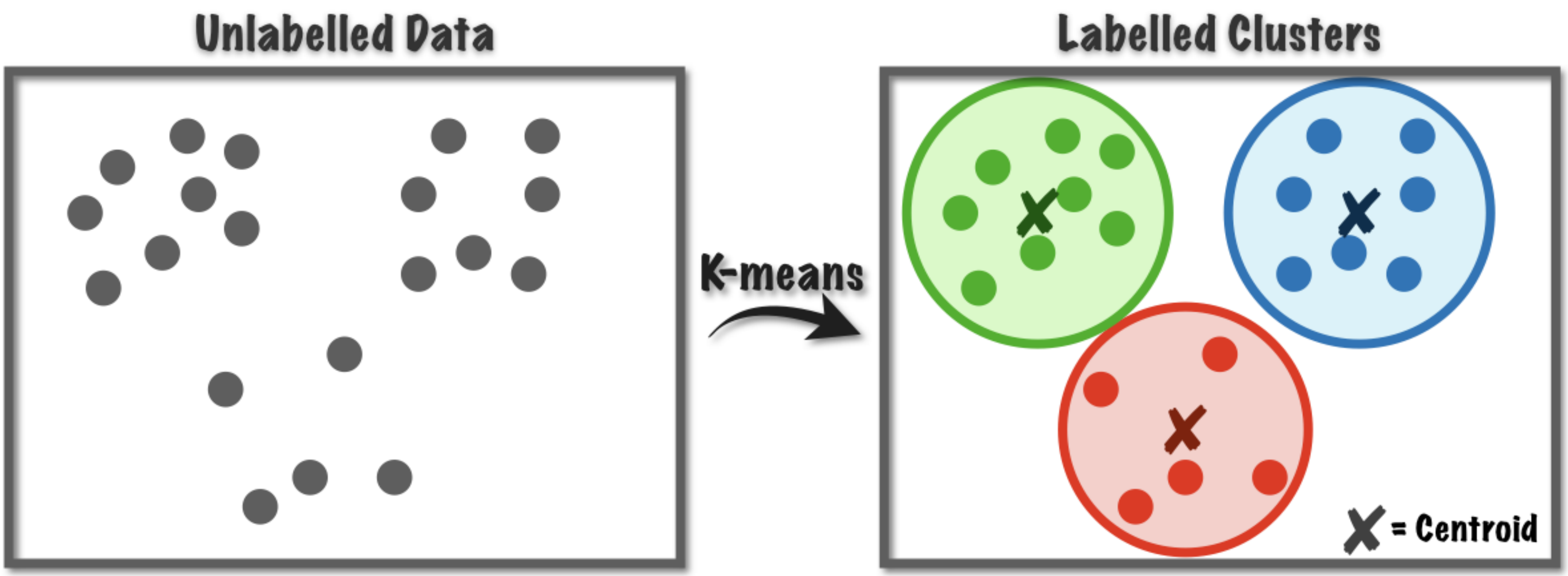
dimension reduction (PCA, ISOMAP)

not related to downstream task

backbone (convolutional neural network)

end-to-end learning, related to downstream task

Next lecture: K-means



Homework 3 is online; due on June 13th

Thank you very much!

sihengc@sjtu.edu.cn