Linear Dimensionality Reduction: PCA

DCS310

Sun Yat-sen University

Outline

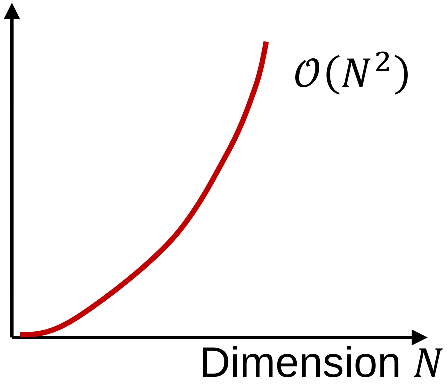
- Motivation
- Perspective 1: Minimizing Reconstruction Error
- Perspective 2: Maximizing Variance
- Perspective 3: SVD
- Other Applications of PCA

Motivation

- The dimensionality of many types of data is very high, *e.g.*, the dimension of each image below is as high as

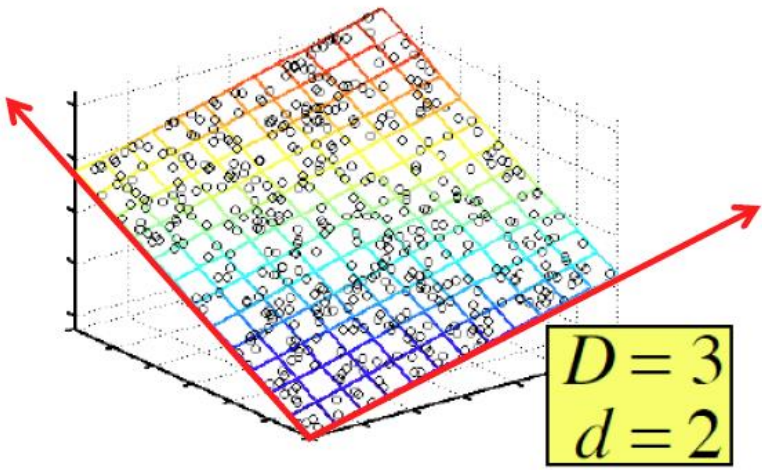
$$256 \times 256 = 65536$$

- If we work on the raw data directly, the complexity of subsequent tasks (e.g. classification) could be extremely high

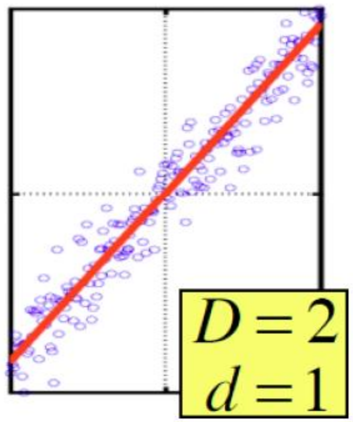


The complexity of logistic regression

- The high-dimensional data often resides on a low-dimensional intrinsic space approximately



3-dimensional data lies on a 2-dimensional plane



2-dimensional data lies on a 1-dimensional line

Finding *the principal directions* so that the dimensions of data represented under the new directions can be reduced significantly

- For the real-world data, this is also possible

e.g., an face image can be represented well *by only several values* if appropriate principal directions can be found

$$\boldsymbol{x} \approx \boldsymbol{\mu}_0 + a_1\boldsymbol{\mu}_1 + \cdots + a_7\boldsymbol{\mu}_7$$

The raw image $\boldsymbol{x}$ that has 65536 values can be represented by only 7 values of $a_1, \cdots a_7$

Outline

- Motivation
- Perspective 1: Minimizing the Reconstruction Error
- Perspective 2: Maximizing Variance
- Perspective 3: SVD
- Other Applications of PCA

Re-representation under the New Directions

- Orthonormal directions in high dimensional space

A set of vectors $\boldsymbol{u}_i$ satisfying

$$\boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij}$$

where $\delta_{ij} = 1$ if $i = j$; 0 otherwise

Theorem: Under the given M orthonormal directions $\boldsymbol{u}_i$, the *best approximation* to a data sample $\boldsymbol{x}$ is

$$\widetilde{\boldsymbol{x}} = \alpha_1 \boldsymbol{u}_1 + \alpha_2 \boldsymbol{u}_2 + \cdots + \alpha_M \boldsymbol{u}_M$$

with α_i being equal to

$$\alpha_i = \boldsymbol{u}_i^T \boldsymbol{x}$$

Proof:

$$\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|^2 = \left\|\boldsymbol{x}-\sum_{i=1}^{M}\alpha_i\boldsymbol{u}_i\right\|^2$$

$$= \|\boldsymbol{x}\|^2 - 2\sum_{i=1}^{M}\alpha_i\boldsymbol{u}_i^T\boldsymbol{x} + \sum_{i=1}^{M}\alpha_i^2$$

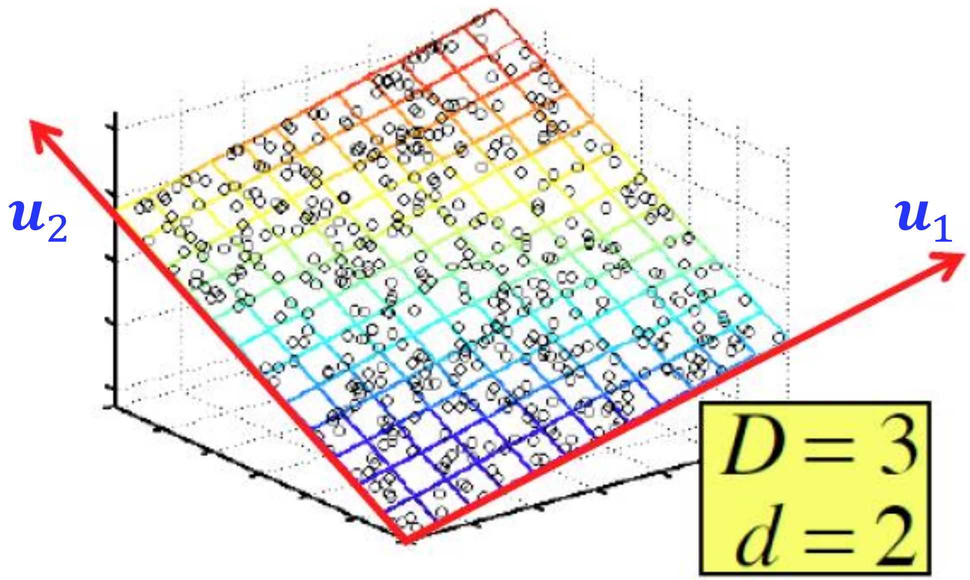
where we used $\boldsymbol{u}_i^T\boldsymbol{u}_j = 0$ for $i \neq j$ and 1 for $i = j$

This is a quadratic function, and can be minimized when $\alpha_i = \boldsymbol{u}_i^T\boldsymbol{x}$

Given the directions $\boldsymbol{u}_i$, the best coefficient is $\alpha_i = \boldsymbol{u}_i^T\boldsymbol{x}$. But *which directions are the best is still unknown*

Finding the Best Directions

- **Objective:** Given data $\{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ from $\mathbb{R}^D$, finding the orthonormal directions $\boldsymbol{u}_i$ under which the original data can be represented best



$$\boldsymbol{x}^{(n)} \approx \sum_{i=1}^{M} \alpha_i^{(n)} \boldsymbol{u}_i$$

- Suppose the best directions $\{\boldsymbol{u}_i\}_{i=1}^{M}$ are given, what is the coefficients $\alpha_i^{(n)}$?

$$\alpha_i^{(n)} = \boldsymbol{u}_i^T \boldsymbol{x}^{(n)}$$

Instead of representing the data $\boldsymbol{x}^{(n)}$ directly, we first center the data to the origin, *i.e.*, representing data

$$\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}},$$

with

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}$$

- The objective can be formulated as minimizing the error between data $\boldsymbol{x}^{(n)}$ and its approximant $\widetilde{\boldsymbol{x}}^{(n)} = \sum_{i=1}^{M} \alpha_i^{(n)} \boldsymbol{u}_i$ in $span(\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M\})$

$$E = \frac{1}{N}\sum_{n=1}^{N} \left\| \left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right) - \widetilde{\boldsymbol{x}}^{(n)} \right\|^2$$

where the best coefficient α_i is known equal to

$$\alpha_i^{(n)} = \boldsymbol{u}_i^T \left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)$$

- Reformulating the reconstruction error E

a) Substituting $\tilde{\boldsymbol{x}}^{(n)} = \sum_{i=1}^{M} \alpha_i^{(n)} \boldsymbol{u}_i$ into
$E = \frac{1}{N}\sum_{n=1}^{N}\left\|\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right) - \tilde{\boldsymbol{x}}^{(n)}\right\|^2$ and using $\boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij}$ gives

$$E = \frac{1}{N}\left(\sum_{n=1}^{N}\left\|\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right\|^2 - 2\sum_{n=1}^{N}\sum_{i=1}^{M}\alpha_i^{(n)}\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)^T \boldsymbol{u}_i + \sum_{n=1}^{N}\sum_{i=1}^{M}\left(\alpha_i^{(n)}\right)^2\right)$$

b) Substituting $\alpha_i^{(n)} = \boldsymbol{u}_i^T\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)$ gives

$$E = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\left\|\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right\|^2}_{\text{Constant}} - \sum_{i=1}^{M}\boldsymbol{u}_i^T \underbrace{\frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)^T}_{\boldsymbol{S}} \boldsymbol{u}_i$$

Constant

c) Rewritting it in a matrix form gives

$$E = \|\boldsymbol{X} - \overline{\boldsymbol{X}}\|_F^2 - \sum_{i=1}^{M} \boldsymbol{u}_i^T \boldsymbol{S} \boldsymbol{u}_i$$

where $\boldsymbol{X} \triangleq \left[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(N)}\right]$ and $\|\cdot\|_F$ is the Frobenius norm

- Minimizing $E = \|\boldsymbol{X} - \overline{\boldsymbol{X}}\|_F^2 - \sum_{i=1}^{M} \boldsymbol{u}_i^T \boldsymbol{S} \boldsymbol{u}_i$ is equivalent to maximizing

$$\boxed{\begin{aligned} &\max_{\boldsymbol{u}_1 \cdots \boldsymbol{u}_M} \sum_{i=1}^{M} \boldsymbol{u}_i^T \boldsymbol{S} \boldsymbol{u}_i \\ &\qquad\qquad s.t.: \boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij} \end{aligned}}$$

- Consider the simple case with $M = 1$. The problem is reduced to:

$$\max_{\boldsymbol{u}_1} \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1$$
$$s.t.: \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$$

  - This is equivalent to maximizing (Lagrange method)

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 - \lambda(\boldsymbol{u}_1^T \boldsymbol{u}_1 - 1)$$

  - Taking the derivative *w.r.t.* $\boldsymbol{u}_1$ and setting it to 0 gives

$$\boldsymbol{S}\boldsymbol{u}_1 = \lambda \boldsymbol{u}_1,$$

from which we can see that $\boldsymbol{u}_1$ should be the eigenvector of $\boldsymbol{S}$ *w.r.t. to the largest eigenvalue*

- For the case with $M = 2$, the problem becomes

$$\max_{\boldsymbol{u}_1,\boldsymbol{u}_2} \boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 + \boldsymbol{u}_2^T \boldsymbol{S} \boldsymbol{u}_2$$

$$s.t.: \boldsymbol{u}_1^T \boldsymbol{u}_1 = 1, \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1, \boldsymbol{u}_1^T \boldsymbol{u}_2 = 0$$

  - This is equivalent to maximizing

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 - \lambda_1(\boldsymbol{u}_1^T \boldsymbol{u}_1 - 1) + \boldsymbol{u}_2^T \boldsymbol{S} \boldsymbol{u}_2 - \lambda_2(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1)$$

    under the constraint $\boldsymbol{u}_1^T \boldsymbol{u}_2 = 0$

  - Taking the derivative *w.r.t.* $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ and setting it to 0 gives

$$\boldsymbol{S} \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1, \qquad \boldsymbol{S} \boldsymbol{u}_2 = \lambda_2 \boldsymbol{u}_2,$$

    ⇒ $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ must be the eigenvectors of $\boldsymbol{S}$

    ⇒ In fact, to have $\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 + \boldsymbol{u}_2^T \boldsymbol{S} \boldsymbol{u}_2$ maximized, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ must be the eigenvectors *corresponding to the largest two eigenvalues*

For the case $M > 1$, the directions $\boldsymbol{u}_i$ are *the eigenvectors of $\boldsymbol{S}$ corresponding to the largest M eigenvalues*

Question: Does the eigenvectors $\boldsymbol{u}_i$ of $\boldsymbol{S}$ satisfy $\boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij}$?

- For any $D \times D$ semi-positive definite matrix $\boldsymbol{S} \triangleq \boldsymbol{X}\boldsymbol{X}^T$, it has D eigenvectors, and they are orthogonal to each other
- For every $\boldsymbol{S} \triangleq \boldsymbol{X}\boldsymbol{X}^T$, it can be decomposed as

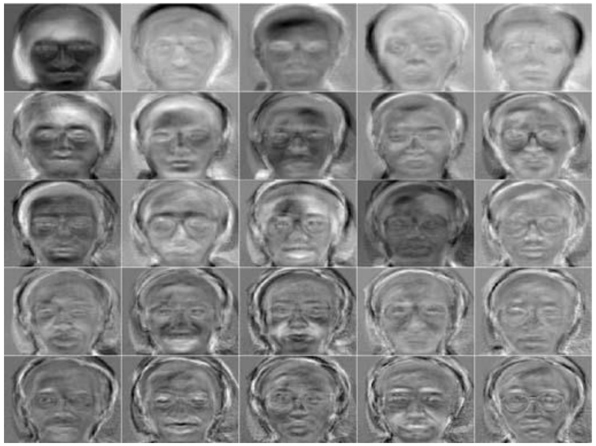
$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$$

where $\boldsymbol{U}$ consists of the eigenvectors and $\boldsymbol{U}\boldsymbol{U}^T = I$; $\boldsymbol{\Lambda}$ is a diagonal matrix consisting of eigenvalues of $\boldsymbol{S}$

Examples

Input data: each face image is a data point

Top 25 principal directions



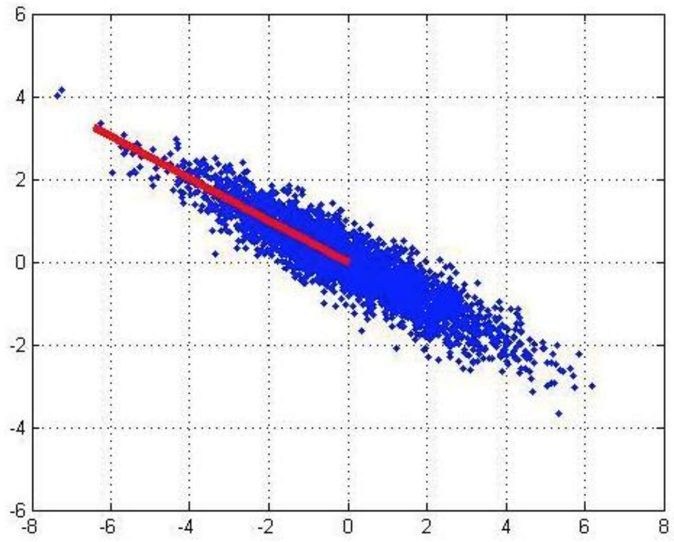
$$\boldsymbol{x} \approx \overline{\boldsymbol{x}} + \alpha_1 \boldsymbol{\mu}_1 + \cdots + \alpha_7 \boldsymbol{\mu}_7$$

Outline

- Motivation
- Perspective 1: Minimizing the Reconstruction Error
- Perspective 2: Maximizing Variance
- Perspective 3: SVD
- Other Applications of PCA

Problem Formulation

- **Objective:** Given data $\{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ from $\mathbb{R}^D$, finding the orthogonal directions $\boldsymbol{u}_i$ onto which the variance of data projected is maximized



Maximizing the variance is equivalent to *preserving the information of the original data as much as possible*

- For the first direction $\boldsymbol{u}_1$, we hope the variance in data projected onto the direction $\boldsymbol{u}_1$, *i.e.*, $\boldsymbol{u}_1^T\boldsymbol{x}^{(n)}$ is maximized
  - The variance expression

$$var = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{u}_1^T\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)\right)^2$$

$$= \boldsymbol{u}_1^T \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)^T \boldsymbol{u}_1$$

$$= \boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1$$

  - Subjecting to $\boldsymbol{u}_1^T\boldsymbol{u}_1 = 1$, as derived before, the variance is maximized when $\boldsymbol{u}_1$ is *the eigenvector of* $\boldsymbol{S}$ *corresponding to the largest eigenvalue*

- For the second direction $\boldsymbol{u}_2$, it also should maximize the variance

$$var = \boldsymbol{u}_2^T \boldsymbol{S} \boldsymbol{u}_2,$$

but should subject to the constraints $\boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij}$, that is,

$$\boldsymbol{u}_2^T \boldsymbol{u}_2 = 1 \qquad \boldsymbol{u}_1^T \boldsymbol{u}_2 = 0$$

- Due to $\boldsymbol{u}_1$ is the eigenvector *w.r.t.* the largest eigenvalue, it can be proved that $\boldsymbol{u}_2$ *is the eigenvector of* $\boldsymbol{S}$ *corresponding to the second largest eigenvalue*

$\boldsymbol{u}_i$ *is the eigenvector of* $\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)^T$ *corresponding to the i-th largest eigenvalue*

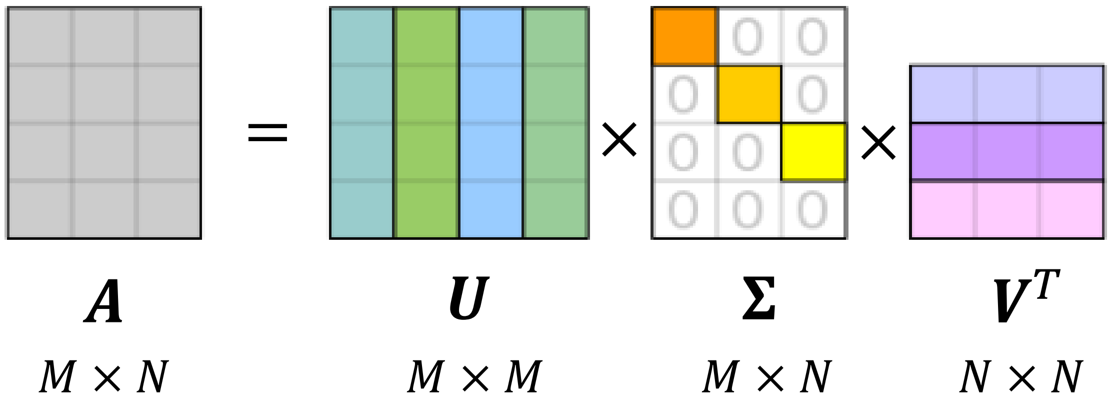
Outline

- Motivation
- Perspective 1: Minimizing Reconstruction Error
- Perspective 2: Maximizing Variance
- Perspective 3: SVD
- Other Applications of PCA

Singular Value Decomposition (SVD)

- For any $M \times N$ matrix $\boldsymbol{A}$, it can always be decomposed as

$$\boldsymbol{A} = \boldsymbol{U\Sigma V}^T$$



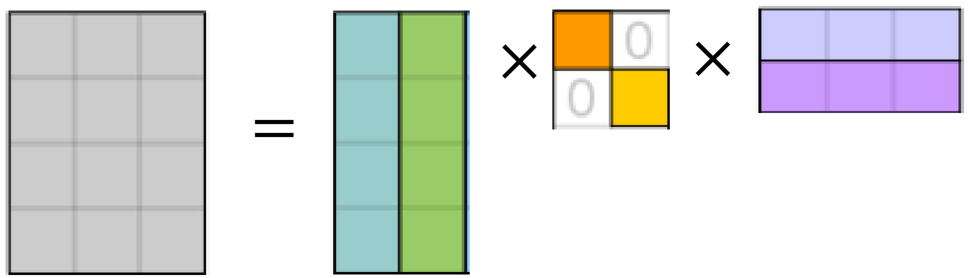
  - $\boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$, with $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ being the i-th eigenvector of $\boldsymbol{AA}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$, *and* $\boldsymbol{u}_i^T\boldsymbol{u}_j = \delta_{ij}$ and $\boldsymbol{v}_i^T\boldsymbol{v}_j = \delta_{ij}$
  - $\boldsymbol{\Sigma}$ has nonzero values on the diagonal, which are the squared roots of the eigenvalues of $\boldsymbol{AA}^T$ or $\boldsymbol{A}^T\boldsymbol{A}$ *(They are the same)*

    Σ_{ii} is called *singular values* and are stored in a decreasing order

- Because $\boldsymbol{\Sigma}$ only has nonzero values on the diagonal, $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A} = \boldsymbol{U}'\boldsymbol{\Sigma}'\boldsymbol{V}'^T = \sum_{i=1}^{r} \Sigma_{ii}\boldsymbol{u}_i\boldsymbol{v}_i^T$$

where $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the i-th column of $\boldsymbol{U}$ and $\boldsymbol{V}$; r is the number of nonzero diagonal elements in $\boldsymbol{\Sigma}$



$\boldsymbol{A}$ $\quad$ $\boldsymbol{U}'$ $\quad$ $\boldsymbol{\Sigma}'$ $\quad$ $\boldsymbol{V}'^T$

$M \times N$ $\quad$ $M \times r$ $\quad$ $r \times r$ $\quad$ $r \times N$

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{0.13} & 0.02 & -0.01 \\ \mathbf{0.41} & 0.07 & -0.03 \\ \mathbf{0.55} & 0.09 & -0.04 \\ \mathbf{0.68} & 0.11 & -0.05 \\ 0.15 & \mathbf{-0.59} & \mathbf{0.65} \\ 0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\ 0.07 & \mathbf{-0.29} & \mathbf{0.32} \end{bmatrix} \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix} \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\ 0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09 \end{bmatrix}$$

- The vector $\boldsymbol{u}_i$ in the SVD decomposition of $\boldsymbol{A}$ is the eigenvector of $\boldsymbol{A}\boldsymbol{A}^T$ *w.r.t.* its i-th largest eigenvalues

- By defining $\widetilde{\boldsymbol{X}} = \left[\boldsymbol{x}^{(1)} - \overline{\boldsymbol{x}}, \boldsymbol{x}^{(2)} - \overline{\boldsymbol{x}}, \cdots, \boldsymbol{x}^{(N)} - \overline{\boldsymbol{x}}\right]$, we can see that

$$\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{X}}^T = \sum_{n=1}^{N}\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}^{(n)} - \overline{\boldsymbol{x}}\right)^T$$

$$= N \cdot \boldsymbol{S},$$

which has the same eigenvectors as the matrix $\boldsymbol{S}$

If we do SVD on $\widetilde{\boldsymbol{X}}$, we can obtain the principal directions of the data $\left\{\boldsymbol{x}^{(n)}\right\}_{n=1}^{N}$

Outline

- Motivation
- Perspective 1: Minimizing Reconstruction Error
- Perspective 2: Maximizing Variance
- Perspective 3: SVD
- Other Applications of PCA

Image Compression

Divide the 372×492 image below into many 12×12 patches

- Each patch is viewed as an data instance
- Performing PCA on the patches $12 \times 12 \rightarrow 5 \times 5$

Reconstruction Error vs # PCA components

降低维数越多
相对误差越大

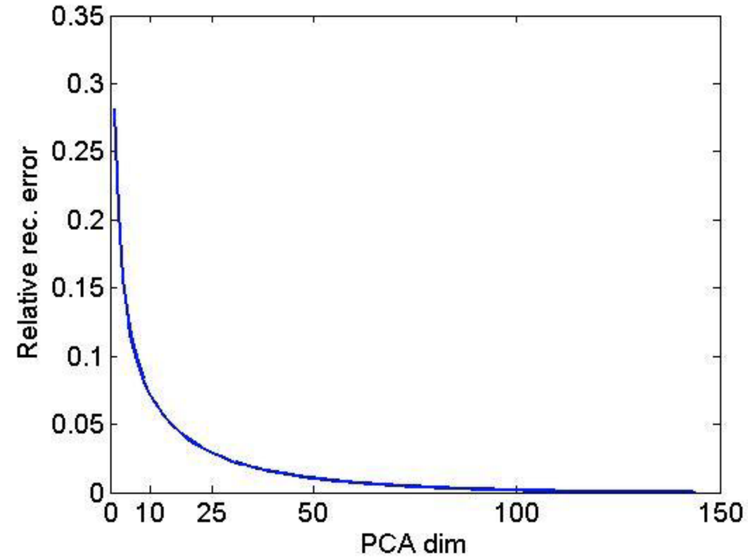
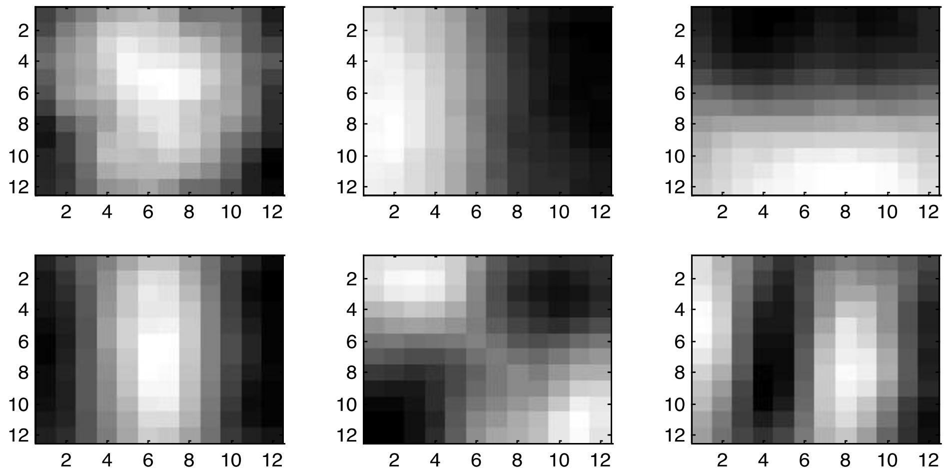


Illustration of the top 6 PCA components



Reconstruction with the top 60 components

Reconstruction with the top 16 components

Denoising

Noisy Image

Denoised Image

Reconstructed from the top 15 principal components