



# Linear Dimensionality Reduction: PCA

Qinliang Su (苏勤亮)

Sun Yat-sen University

[suqliang@mail.sysu.edu.cn](mailto:suqliang@mail.sysu.edu.cn)

# Outline

---

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

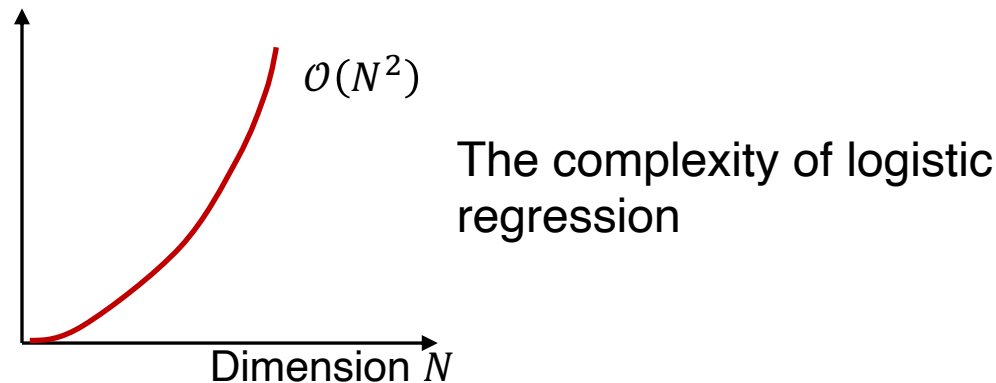
# Motivation

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

$$256 \times 256 = 65536$$

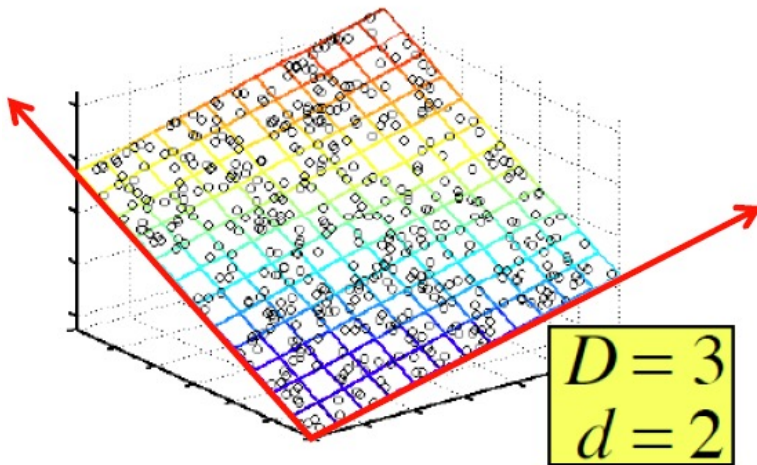


- If we work on the raw data directly, the complexity of subsequent tasks (*e.g.*, classification) could be extremely high

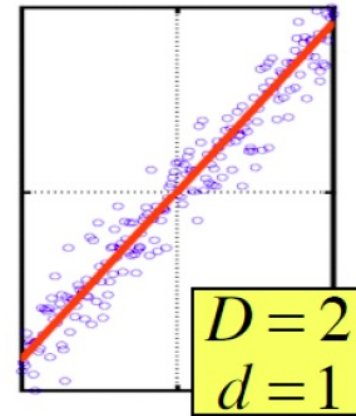


## Why the dimensionality could be reduced?

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



3-dimensional data lies on a 2-dimensional plane approximately

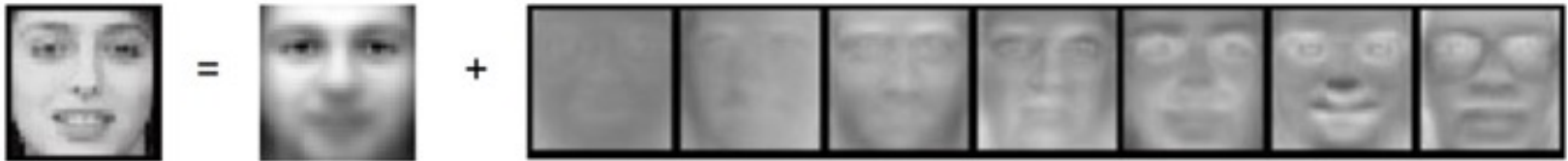


2-dimensional data lies on a 1-dimensional line approximately

The key is how to find *the principal directions* under which data samples could be represented with much fewer dimensions

- For the real-world data, it is also possible to find the low-dimensional space

*For instance, human faces can be well represented **with only several values** if appropriate directions can be found*



$$x \approx \mu_0 + a_1\mu_1 + \cdots + a_7\mu_7$$

The raw image  $x$  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 New Directions

- How to represent **orthogonal directions** in high dimensional space?

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

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

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

**Theorem:** Under the  $M$  given orthogonal directions  $\{\mathbf{u}_i\}_{i=1}^M$ , the **best approximation** to a data sample  $\mathbf{x}$  is

$$\tilde{\mathbf{x}} = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \cdots + \alpha_M \mathbf{u}_M$$

with the  $\alpha_i$  equal to

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

*Proof:*

$$\begin{aligned}\|\mathbf{x} - \tilde{\mathbf{x}}\|^2 &= \left\| \mathbf{x} - \sum_{i=1}^M \alpha_i \mathbf{u}_i \right\|^2 \\ &= \|\mathbf{x}\|^2 - 2 \sum_{i=1}^M \alpha_i \mathbf{u}_i^T \mathbf{x} + \sum_{i=1}^M \alpha_i^2\end{aligned}$$

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

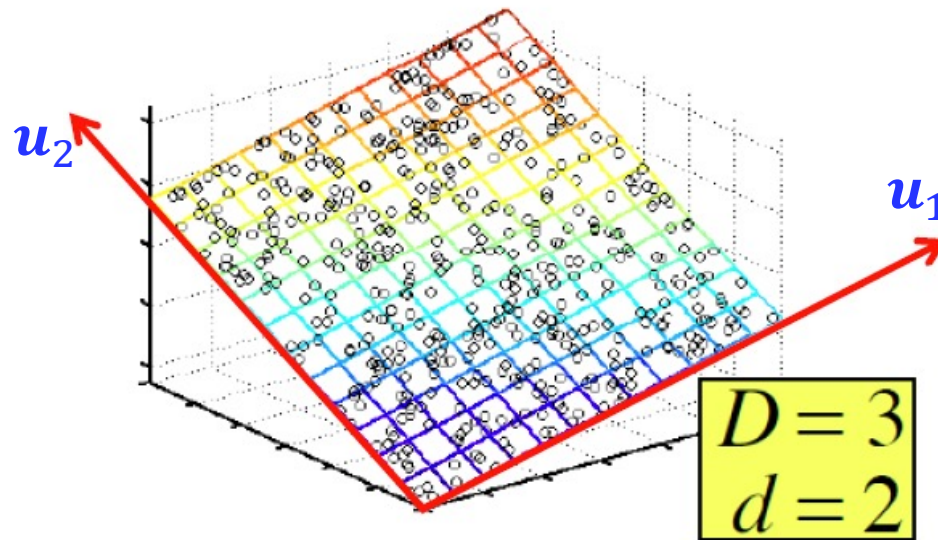
This is a quadratic function, and can be minimized when  $\alpha_i = \mathbf{u}_i^T \mathbf{x}$

Given the directions  $\{\mathbf{u}_i\}_{i=1}^M$ , the best coefficient is  $\alpha_i = \mathbf{u}_i^T \mathbf{x}$ .  
But *how to find the best directions is still unknown*



# Finding the Best Directions

- **Goal:** Given data samples  $\{\mathbf{x}^{(n)}\}_{n=1}^N$  from  $\mathbb{R}^D$ , finding  $M$  orthogonal directions  $\mathbf{u}_i$  such that the original data can be best represented under them



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

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

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

Instead of representing the data  $\mathbf{x}^{(n)}$  directly, we first center the data to the origin, *i.e.*, subtracting each data point  $\mathbf{x}^{(n)}$  by its mean

$$\mathbf{x}^{(n)} - \bar{\mathbf{x}}, \quad \text{where } \bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(n)}$$

- The objective can now be described as minimizing the error between  $\mathbf{x}^{(n)} - \bar{\mathbf{x}}$  and its best approximant  $\tilde{\mathbf{x}}^{(n)} = \sum_{i=1}^M \alpha_i^{(n)} \mathbf{u}_i$

$$\text{Min}_{\alpha_i^{(n)}, \mathbf{u}_i} E \quad \text{with} \quad E \triangleq \frac{1}{N} \sum_{n=1}^N \left\| (\mathbf{x}^{(n)} - \bar{\mathbf{x}}) - \sum_{i=1}^M \alpha_i^{(n)} \mathbf{u}_i \right\|^2$$

where the best coefficient  $\alpha_i^{(n)}$  is known to be

$$\alpha_i^{(n)} = \mathbf{u}_i^T (\mathbf{x}^{(n)} - \bar{\mathbf{x}})$$

- Reformulating the reconstruction error  $E$

a) Rewriting  $E = \frac{1}{N} \sum_{n=1}^N \left\| (\mathbf{x}^{(n)} - \bar{\mathbf{x}}) - \sum_{i=1}^M \alpha_i^{(n)} \mathbf{u}_i \right\|^2$  and noticing  $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$  gives

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



b) Substituting  $\alpha_i^{(n)} = \mathbf{u}_i^T (\mathbf{x}^{(n)} - \bar{\mathbf{x}})$  gives

$$E = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}^{(n)} - \bar{\mathbf{x}}\|^2 - \sum_{i=1}^M \mathbf{u}_i^T \underbrace{\frac{1}{N} \sum_{n=1}^N (\mathbf{x}^{(n)} - \bar{\mathbf{x}})(\mathbf{x}^{(n)} - \bar{\mathbf{x}})^T}_{\triangleq \mathbf{S}} \mathbf{u}_i$$

c) Writing it into a matrix form gives

$$E = \frac{1}{N} \|\mathbf{X} - \bar{\mathbf{X}}\|_F^2 - \frac{1}{N} \sum_{i=1}^M \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i$$

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

- Minimizing  $E = \frac{1}{N} \|\mathbf{X} - \bar{\mathbf{X}}\|_F^2 - \frac{1}{N} \sum_{i=1}^M \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i$  under the constraint  $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$  is equivalent to maximize

$$\max_{\mathbf{u}_1 \cdots \mathbf{u}_M} \sum_{i=1}^M \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i$$

$$s. t. : \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$$

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

$$\begin{aligned} \max_{\mathbf{u}_1} \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \\ \text{s.t.} : \mathbf{u}_1^T \mathbf{u}_1 = 1 \end{aligned}$$

- This is equivalent to maximize

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

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

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

from which we can see that  $\mathbf{u}_1$  should be **the eigenvector of  $\mathbf{S}$**

- It can be further checked that it is **the eigenvector *w.r.t. to the largest eigenvalue* that maximizes  $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1$**

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

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

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

- This is equivalent to maximize

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

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

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

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1, \quad \mathbf{S} \mathbf{u}_2 = \lambda_2 \mathbf{u}_2,$$

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

⇒ It can be seen that to maximize  $\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2$ ,  $\mathbf{u}_1$  and  $\mathbf{u}_2$  must be the eigenvectors **corresponding to the two largest eigenvalues**

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

$$\mathbf{A} = \mathbf{X} - \bar{\mathbf{X}}$$

*Question:* Will the eigenvectors  $\mathbf{u}_i$  of  $S$  satisfy  $\mathbf{u}_i^T \mathbf{u}_j = 0$  for  $i \neq j$ ?

- For any  $D \times D$  *real symmetric* matrix like  $S \triangleq \mathbf{A}\mathbf{A}^T$ , it has  $D$  eigenvectors that are orthogonal to each other
- For every  $S \triangleq \mathbf{A}\mathbf{A}^T$ , it can be decomposed as

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

where  $\mathbf{U}$  is comprised of the eigenvectors of  $S$  and  $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ ;  
 $\mathbf{\Lambda}$  is a diagonal matrix composed of eigenvalues of  $S$

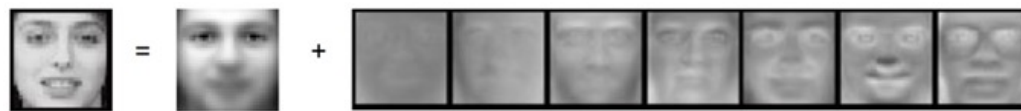


# Examples

Input data: each face image is a data point



Top 25 principal directions



$$x \approx \bar{x} + \alpha_1 u_1 + \cdots + \alpha_7 u_7$$

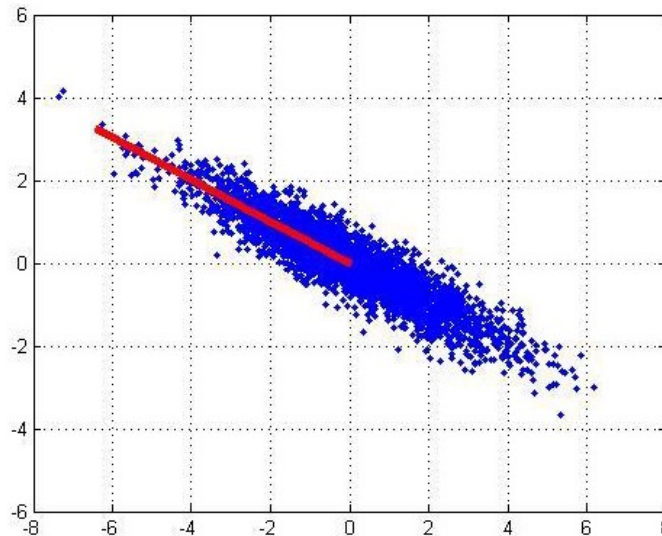
# Outline

---

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

# Problem Formulation

- **Goal:** Given a dataset  $\{\mathbf{x}^{(n)}\}_{n=1}^N$ , find orthogonal directions  $\{\mathbf{u}_k\}_{k=1}^M$  such that **the variance** of data projected onto these directions are maximized



Maximizing the variance amounts to *preserve the information of original data as much as possible*

- For the first direction  $\mathbf{u}_1$ , we hope the variance of projected data along the direction  $\mathbf{u}_1$ , i.e.,  $\{\mathbf{u}_1^T \mathbf{x}^{(n)}\}_{n=1}^N$ , is maximized

➤ The variance expression

$$\begin{aligned} var &= \frac{1}{N} \sum_{n=1}^N \left( \mathbf{u}_1^T (\mathbf{x}^{(n)} - \bar{\mathbf{x}}) \right)^2 \\ &= \mathbf{u}_1^T \frac{1}{N} \sum_{n=1}^N (\mathbf{x}^{(n)} - \bar{\mathbf{x}})(\mathbf{x}^{(n)} - \bar{\mathbf{x}})^T \mathbf{u}_1 \\ &= \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \end{aligned}$$

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

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

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

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

$$\mathbf{u}_2^T \mathbf{u}_2 = 1 \quad \mathbf{u}_1^T \mathbf{u}_2 = 0$$

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

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

# Outline

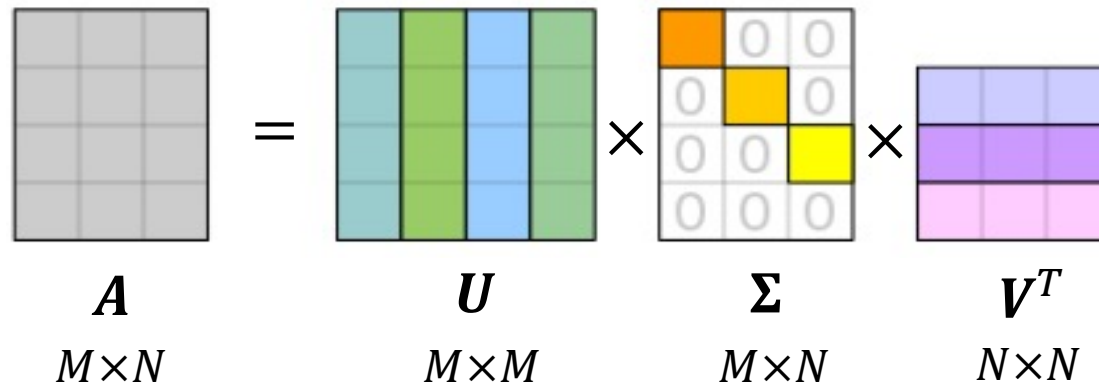
---

- 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  $A$ , it can always be decomposed as

$$A = U \Sigma V^T$$



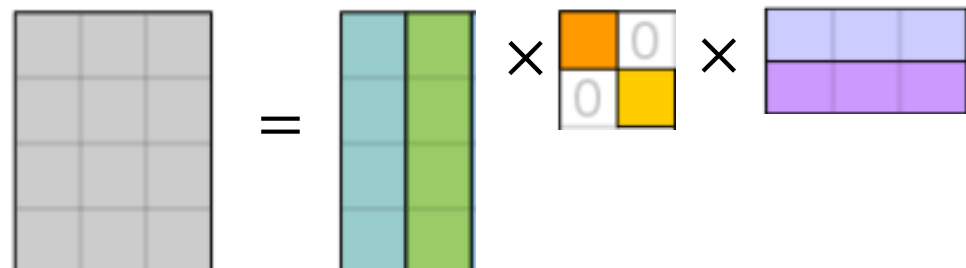
- $U = [\mathbf{u}_1, \dots, \mathbf{u}_M]$  and  $V = [\mathbf{v}_1, \dots, \mathbf{v}_N]$ , with  $\mathbf{u}_i$  and  $\mathbf{v}_i$  being the  $i$ -th eigenvector of  $AA^T$  and  $A^T A$ , and  $\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$  and  $\mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$
- $\Sigma$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

$\Sigma_{ii}$  is called *singular values* and are stored in a descending order

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

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

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



$$\begin{matrix}
 \mathbf{A} & = & \mathbf{U}' & \times & \mathbf{\Sigma}' & \times & \mathbf{V}'^T \\
 M \times N & & M \times r & & r \times r & & r \times N
 \end{matrix}$$

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



- The vector  $\mathbf{u}_i$  in the SVD decomposition of  $A$  is the eigenvector of  $AA^T$  w.r.t. its  $i$ -th largest eigenvalues
- By setting  $A = \tilde{X}$  with  $\tilde{X} \triangleq [\mathbf{x}^{(1)} - \bar{\mathbf{x}}, \mathbf{x}^{(2)} - \bar{\mathbf{x}}, \dots, \mathbf{x}^{(N)} - \bar{\mathbf{x}}]$ , we can see that

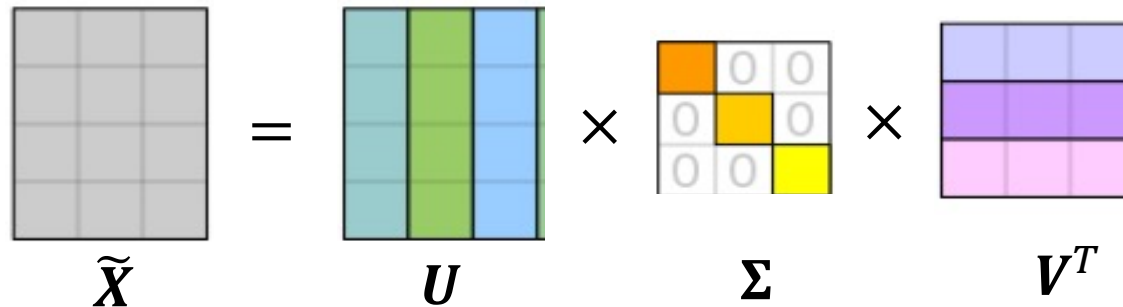
$$\begin{aligned}\tilde{X}\tilde{X}^T &= \sum_{n=1}^N (\mathbf{x}^{(n)} - \bar{\mathbf{x}})(\mathbf{x}^{(n)} - \bar{\mathbf{x}})^T \\ &= N \cdot \mathbf{S}\end{aligned}$$

⇒ The eigenvectors of  $\tilde{X}\tilde{X}^T$  are the same as the matrix  $\mathbf{S}$

By performing SVD on  $\tilde{X}$ , principal directions of data  $\{\mathbf{x}^{(n)}\}_{n=1}^N$  can be directly obtained, which are the columns of left SVD matrix  $\mathbf{U}$

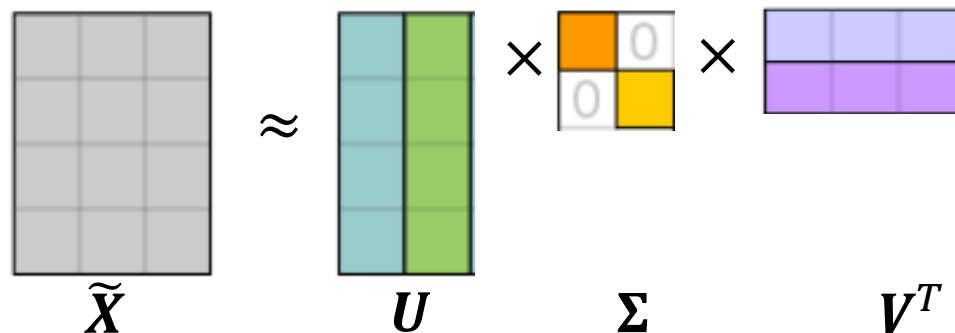
- Question:

Given the SVD decomposition of  $\tilde{X}$  as shown below, what are the principle directions and the coefficients  $\alpha_i$ 's for  $\tilde{x}^{(n)}$ ?



The diagram shows the SVD decomposition of a 4x4 matrix  $\tilde{X}$  (gray grid) into three matrices:  $U$  (4x3, with columns colored teal, green, and blue),  $\Sigma$  (3x3, with diagonal elements colored orange, yellow, and light yellow, and off-diagonal elements white), and  $V^T$  (3x4, with rows colored light blue, purple, and pink). The equation is  $\tilde{X} = U \Sigma V^T$ .

If only top two directions are kept, what are the coefficients  $\alpha_i$ 's?



The diagram shows the approximate SVD decomposition of a 4x4 matrix  $\tilde{X}$  (gray grid) into three matrices:  $U$  (4x2, with columns colored teal and green),  $\Sigma$  (2x2, with diagonal elements colored orange and yellow, and off-diagonal elements white), and  $V^T$  (2x4, with rows colored light blue and purple). The equation is  $\tilde{X} \approx U \Sigma V^T$ .

# Outline

---

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

# Image Compression

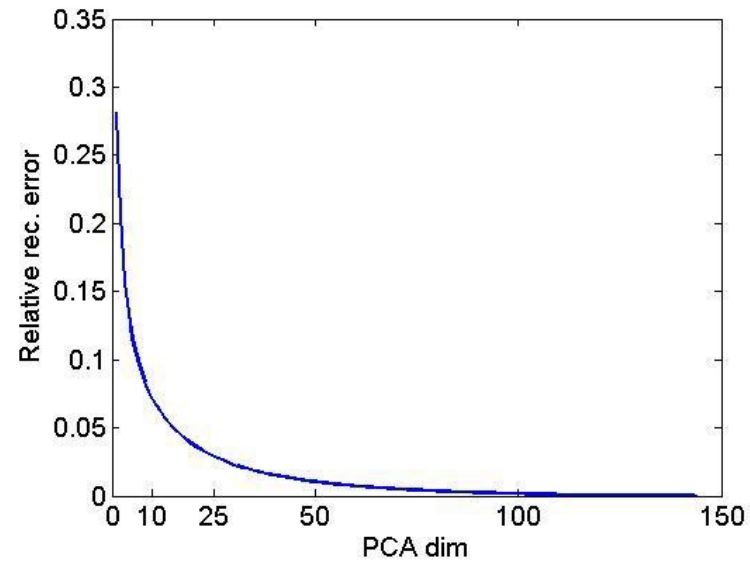
---

Partition a  $372 \times 492$  image below into many  $12 \times 12$  patches

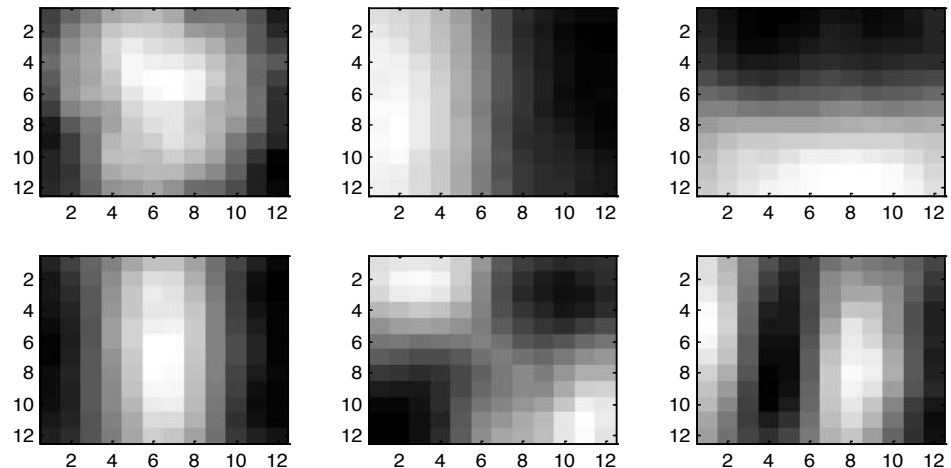
- Each patch is viewed as a data instance
- Performing PCA on the patches



## Reconstruction Error vs # PCA components



## 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