

# ECE 302: Lecture 5.11 Principal Component Analysis

Prof Stanley Chan

School of Electrical and Computer Engineering  
Purdue University



## What is PCA?

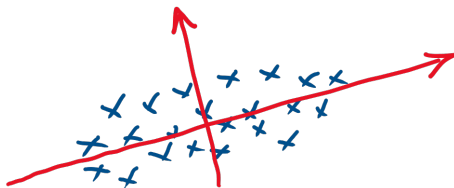
**The key idea of PCA is eigen-decomposition of the covariance matrix  $\Sigma$ .**

# Low-Dimensional Representation

- Consider a set of data point  $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$
- These data points are living in a **high dimensional space**  $\mathbf{x}^{(n)} \in \mathbb{R}^d$
- Find a **low dimensional representation** in  $\mathbb{R}^p$  where  $p < d$
- Equivalent to finding the **principal components**  $\mathbf{v}_1, \dots, \mathbf{v}_p$  such that

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

- Then every  $\mathbf{x}^{(n)} \in \mathbb{R}^d$  can be represented using  $\alpha^{(n)} \in \mathbb{R}^p$ .



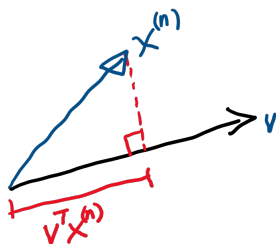
# One Sample Analysis

- Consider a simpler problem: One data point  $\mathbf{x}$  and one direction  $\mathbf{v}$ .
- We want to find a direction  $\hat{\mathbf{v}}$  and a scalar  $\hat{\alpha}$  such that

$$(\hat{\mathbf{v}}, \hat{\alpha}) = \underset{\|\mathbf{v}\|_2=1, \alpha}{\operatorname{argmin}} \left\| \begin{bmatrix} | \\ | \\ \mathbf{x} \\ | \\ | \end{bmatrix} - \alpha \begin{bmatrix} | \\ | \\ \mathbf{v} \\ | \\ | \end{bmatrix} \right\|^2$$

- First assume  $\mathbf{v}$  is available. Then take derivative w.r.t.  $\alpha$ :

$$2\mathbf{v}^T(\mathbf{x} - \alpha\mathbf{v}) = 0 \quad \Rightarrow \quad \alpha = \mathbf{v}^T\mathbf{x}.$$



# One Sample Analysis

- Substitute  $\alpha = \mathbf{x}^T \mathbf{v}$  into the optimization
- Then the optimization becomes

$$\begin{aligned}\operatorname{argmin}_{\|\mathbf{v}\|_2=1} \|\mathbf{x} - \alpha \mathbf{v}\|^2 &= \operatorname{argmin}_{\|\mathbf{v}\|_2=1} \left\{ \mathbf{x}^T \mathbf{x} - 2\alpha \mathbf{x}^T \mathbf{v} + \alpha^2 \mathbf{v}^T \mathbf{v} \right\} \\ &= \operatorname{argmin}_{\|\mathbf{v}\|_2=1} \left\{ -2\alpha \mathbf{x}^T \mathbf{v} + \alpha^2 \right\} \\ &= \operatorname{argmin}_{\|\mathbf{v}\|_2=1} \left\{ -2(\mathbf{x}^T \mathbf{v}) \mathbf{x}^T \mathbf{v} + (\mathbf{x}^T \mathbf{v})^2 \right\} \\ &= \operatorname{argmax}_{\|\mathbf{v}\|_2=1} \left\{ \mathbf{v}^T \mathbf{x} \mathbf{x}^T \mathbf{v} \right\}\end{aligned}$$

- Take expectation on both sides:

$$\operatorname{argmin}_{\|\mathbf{v}\|_2=1} \mathbb{E}_{\mathbf{x}} \|\mathbf{x} - \alpha \mathbf{v}\|^2 = \operatorname{argmax}_{\|\mathbf{v}\|_2=1} \mathbf{v}^T \mathbb{E}_{\mathbf{x}} \left\{ \mathbf{x} \mathbf{x}^T \right\} \mathbf{v}$$

# Eigenvalue Problem

- Let  $\Sigma \stackrel{\text{def}}{=} \mathbb{E}[\mathbf{x}\mathbf{x}^T]$ .
- Then the optimization problem is

$$\operatorname{argmax}_{\|\mathbf{v}\|_2=1} \mathbf{v}^T \Sigma \mathbf{v}.$$

- The solution to this problem is the eigenvalue and eigenvectors of  $\Sigma$ .

## Theorem

Let  $\Sigma$  be a  $d \times d$  matrix with eigen-decomposition  $\Sigma = \mathbf{U}\mathbf{S}\mathbf{U}^T$ . Then, the optimization

$$\hat{\mathbf{v}} = \operatorname{argmax}_{\|\mathbf{v}\|_2=1} \mathbf{v}^T \Sigma \mathbf{v}.$$

has a solution  $\hat{\mathbf{v}} = \mathbf{u}_i$  for any  $i = 1, \dots, d$ .

Proof: See eBook

# Finite Samples

- When there are  $N$  training samples, the optimization is

$$\operatorname{argmin}_{\|\mathbf{v}\|_2=1} \underbrace{\frac{1}{N} \sum_{n=1}^N \|\mathbf{x}^{(n)} - \alpha^{(n)} \mathbf{v}\|^2}_{=\mathbb{E}[\|\mathbf{x} - \alpha \mathbf{v}\|^2], N \rightarrow \infty} = \operatorname{argmax}_{\|\mathbf{v}\|_2=1} \mathbf{v}^T \underbrace{\left\{ \frac{1}{N} \sum_{n=1}^N \mathbf{x}^{(n)} (\mathbf{x}^{(n)})^T \right\}}_{=\mathbb{E}[\mathbf{x} \mathbf{x}^T], N \rightarrow \infty} \mathbf{v}$$

- In practice, given  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$ , we approximate  $\Sigma$  by its empirical estimate

$$\Sigma \approx \frac{1}{N} \sum_{n=1}^N \mathbf{x}^{(n)} (\mathbf{x}^{(n)})^T$$

- You can also remove the mean vectors:  $\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^{(n)}$ :

$$\Sigma \approx \frac{1}{N} \sum_{n=1}^N (\mathbf{x}^{(n)} - \boldsymbol{\mu})(\mathbf{x}^{(n)} - \boldsymbol{\mu})^T$$

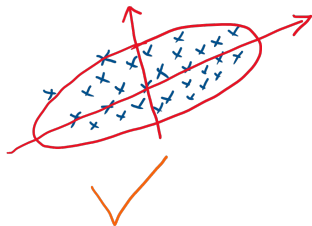
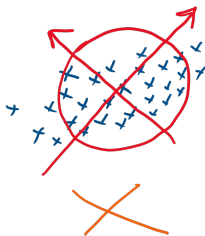
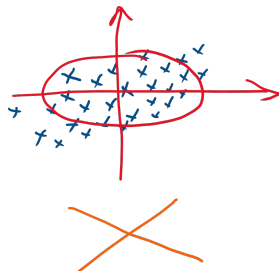
# Statistical Interpretation

- The optimization

$$\operatorname{argmax}_{\|\mathbf{v}\|_2=1} \mathbf{v}^T \boldsymbol{\Sigma} \mathbf{v}.$$

asks us to find a principal direction that maximizes the **variance**.

- Belief: Large variance = “signal”, small variance = “noise”





# The Eigenface Problem



Figure: The extended Yale Face Database B.

- Dataset:  $\{\mathbf{x}^{(n)}\}_{n=1}^N$ .
- Each  $\mathbf{x}^{(n)} \in \mathbb{R}^d$  is a vector representation of a  $\sqrt{d} \times \sqrt{d}$  image.
- Task 1: Find a low-dimensional representation (This lecture)
- Task 2: Classify faces for a new image (Later)

## Low Dimensional Representation

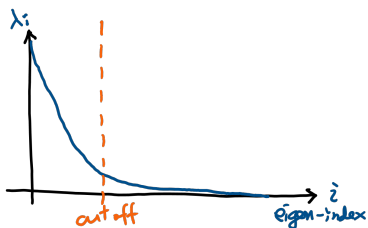
- Estimate the mean vector  $\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^{(n)}$ .
- Estimate the covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}^{(n)} - \boldsymbol{\mu})(\mathbf{x}^{(n)} - \boldsymbol{\mu})^T. \quad (1)$$

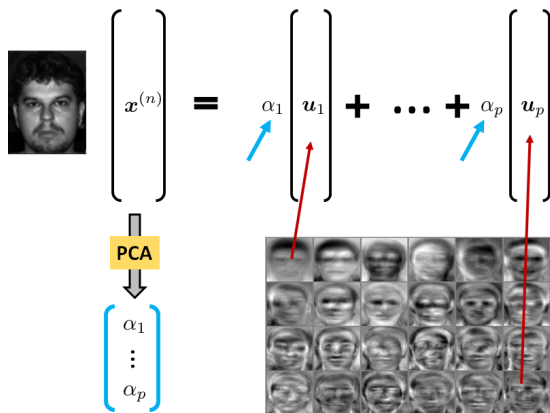
- Eigen-decomposition:  $\boldsymbol{\Sigma} = \mathbf{U}\mathbf{S}\mathbf{U}^T$ .
- When a new image  $\mathbf{y}$  comes, estimate the coefficients:

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

- How many coefficients to use?



## The Basis Vectors $u_i$



**Figure:** Given a face image, the learned basis vectors (from the eigen-decomposition of the covariance matrix) can be used to compress the image  $x$  into a feature vector  $\alpha$  where the dimension of  $\alpha$  is significantly lower than that of  $x$ .

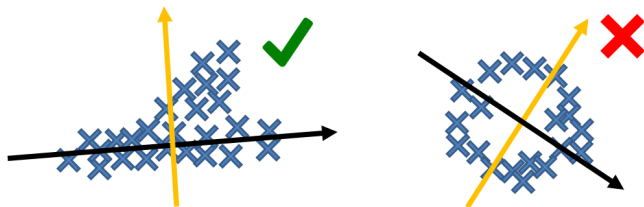
# Discussions

## What does PCA do?

- PCA is a tool for **dimension reduction**.
- It compresses a raw data vector  $\mathbf{y} \in \mathbb{R}^d$  into a smaller feature vector  $\alpha \in \mathbb{R}^p$ .
- You can now do classification in  $\mathbb{R}^p$  instead of  $\mathbb{R}^d$ .

## Limitations of PCA

- Fails when the raw data are not orthogonal.
- Basis returned by PCA are not interpretable.
- PCA does not return you the most influential “component”.



**Questions?**