ECE 595: Machine Learning I

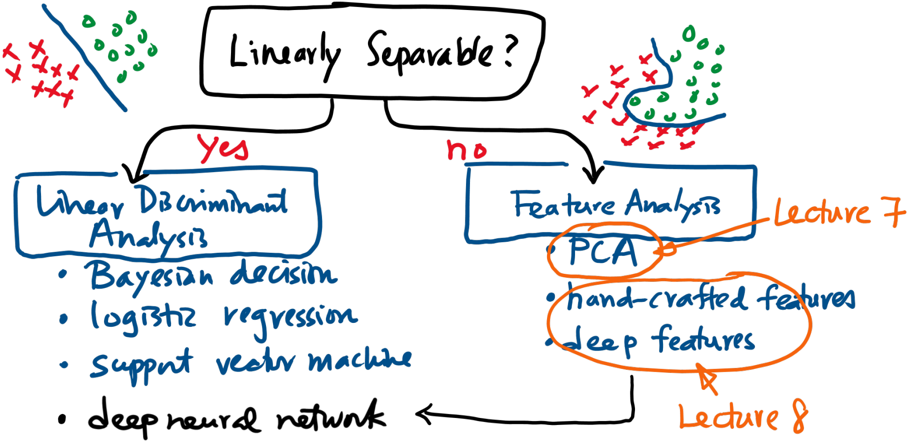
Lecture 7.1: Feature Analysis via PCA

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Supervised Learning for Classification



Outline

Feature Analysis

- Lecture 7 Principal Component Analysis (PCA)
- Lecture 8 Hand-Crafted and Deep Features

This Lecture

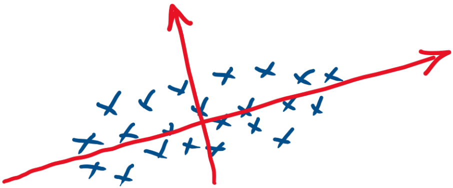
- PCA
  - Low-dimensional Representation
  - Geometric Interpretation
  - Eigen-Face Problem
- Kernel-PCA
  - Adding kernels to PCA
  - Algorithm
  - Examples

Low-Dimensional Representation

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

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

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



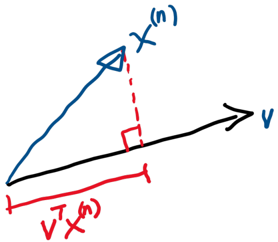
One Sample Analysis

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

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

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

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



One Sample Analysis

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

$$
\begin{aligned}
\underset{\|\boldsymbol{v}\|_2=1}{\text{argmin}} \quad \|\boldsymbol{x} - \alpha \boldsymbol{v}\|^2 &= \underset{\|\boldsymbol{v}\|_2=1}{\text{argmin}} \quad \left\{ \boldsymbol{x}^T \boldsymbol{x} - 2\alpha \boldsymbol{x}^T \boldsymbol{v} + \alpha^2 \cancel{\boldsymbol{v}^T \boldsymbol{v}} \right\} \\
&= \underset{\|\boldsymbol{v}\|_2=1}{\text{argmin}} \quad \left\{ -2\alpha \boldsymbol{x}^T \boldsymbol{v} + \alpha^2 \right\} \\
&= \underset{\|\boldsymbol{v}\|_2=1}{\text{argmin}} \quad \left\{ -2(\boldsymbol{x}^T \boldsymbol{v}) \boldsymbol{x}^T \boldsymbol{v} + (\boldsymbol{x}^T \boldsymbol{v})^2 \right\} \\
&= \underset{\|\boldsymbol{v}\|_2=1}{\text{argmax}} \quad \left\{ \boldsymbol{v}^T \boldsymbol{x} \boldsymbol{x}^T \boldsymbol{v} \right\}
\end{aligned}
$$

- Take expectation on both sides:

$$
\underset{\|\boldsymbol{v}\|_2=1}{\text{argmin}} \quad \mathbb{E}_{\boldsymbol{x}} \|\boldsymbol{x} - \alpha \boldsymbol{v}\|^2 = \underset{\|\boldsymbol{v}\|_2=1}{\text{argmax}} \; \boldsymbol{v}^T \mathbb{E}_{\boldsymbol{x}} \left\{ \boldsymbol{x} \boldsymbol{x}^T \right\} \boldsymbol{v}
$$

Eigenvalue Problem

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

$$\underset{\|\boldsymbol{v}\|_2=1}{\text{argmax}}\; \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}.$$

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

Theorem

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

$$\widehat{\boldsymbol{v}} = \underset{\|\boldsymbol{v}\|_2=1}{\text{argmax}}\; \boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v}.$$

has a solution $\widehat{\boldsymbol{v}} = \boldsymbol{u}_i$ for any $i = 1, \ldots, d$.

Proof: See Appendix.

Finite Samples

- When there are N training samples, the optimization is

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

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

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

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

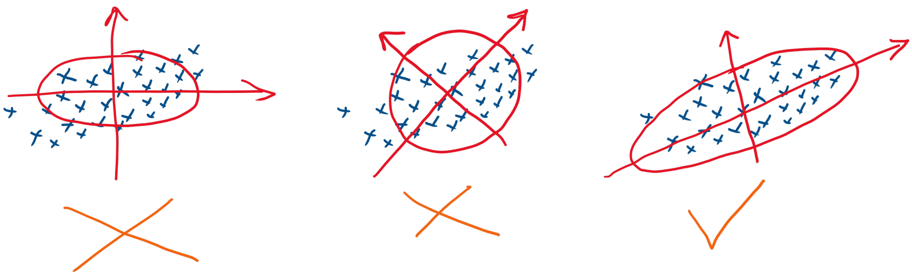
$$\boldsymbol{\Sigma} \approx \frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{x}^{(n)} - \boldsymbol{\mu})(\boldsymbol{x}^{(n)} - \boldsymbol{\mu})^T$$

Statistical Interpretation

- The optimization

$$\underset{\|\boldsymbol{v}\|_2=1}{\text{argmax}}\ \boldsymbol{v}^T\boldsymbol{\Sigma}\boldsymbol{v}.$$

  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: $\{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$.
- Each $\boldsymbol{x}^{(n)} \in \mathbb{R}^d$ 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

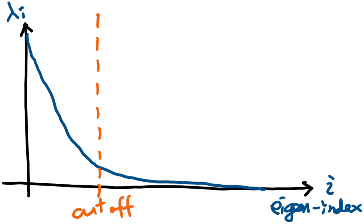
- Estimate the mean vector $\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}^{(n)}$.
- Estimate the covariance matrix

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

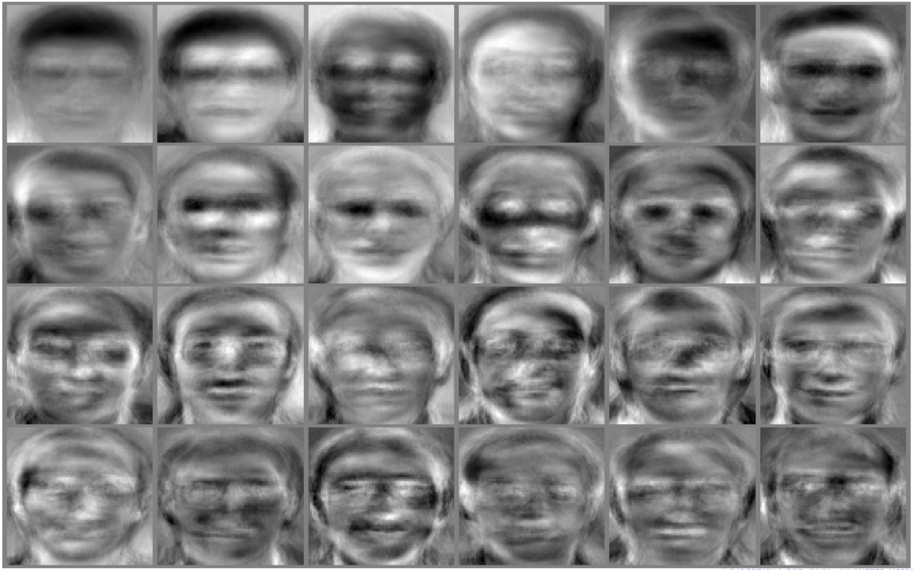
- Eigen-decomposition: $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$.
- When a new image $\boldsymbol{y}$ comes, estimate the coefficients:

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

- How many coefficients to use?



The Basis Vectors $\boldsymbol{u}_i$



Representing Faces

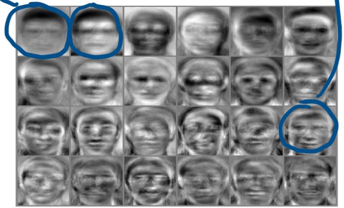
$$\begin{bmatrix} | \\ y \\ | \end{bmatrix} = \alpha_1 \begin{bmatrix} | \\ u_1 \\ | \end{bmatrix} + \alpha_2 \begin{bmatrix} | \\ u_2 \\ | \end{bmatrix} + \cdots \alpha_p \begin{bmatrix} | \\ u_p \\ | \end{bmatrix}$$

data

$$\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_p \end{bmatrix}$$

feature

Basis vectors



Discussion

What does PCA do?

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

When will PCA fail?

- When data intrinsically does not have orthogonal projections
- For example, the distributions below