# EECE 5644: Principal Component Analysis (PCA)

**Mark Zolotas**

E-mail: m.zolotas@northeastern.edu
Webpage: https://coe.northeastern.edu/people/zolotas-mark/

# Tentative Course Outline (Wks. 1-2)

| Topics | Dates | Assignments | Additional Reading |
|---|---|---|---|
| ~~Course Overview~~<br>~~Machine Learning Basics~~ | ~~07/05~~ | **Optional Homework 0** released on Canvas on 07/08 but please do NOT submit on Canvas | ~~Chpt. 1~~<br>~~Murphy 2012~~ |
| ~~Foundations: Linear Algebra, Probability, Numerical Optimization (Gradient Descent), Regression~~ | ~~07/06-12~~ | | ~~Stanford LA Review~~<br>~~Stanford Prob. Review~~<br>~~Chpt. 8 Murphy 2022~~ |
| ~~*Quick Python Tutorial*~~ | ~~07/12~~ | **Homework 1** released on Canvas on 07/15<br>**Due 07/25** | ~~N/A~~ |
| ~~Linear Classifier Design, Linear Discriminant Analysis~~ and Principal Component Analysis (PCA) | 07/13-15 | | ~~Chpts. 9.2.6~~ & 20.1 Murphy 2022 |
| ~~Bayesian Decision Theory: Empirical Risk Min, Max Likelihood (ML), Max a Posteriori~~ | ~~07/14~~ | | ~~Chpt. 2~~<br>~~Duda & Hart 2001~~<br>~~Deniz Erdogmus Notes~~ |

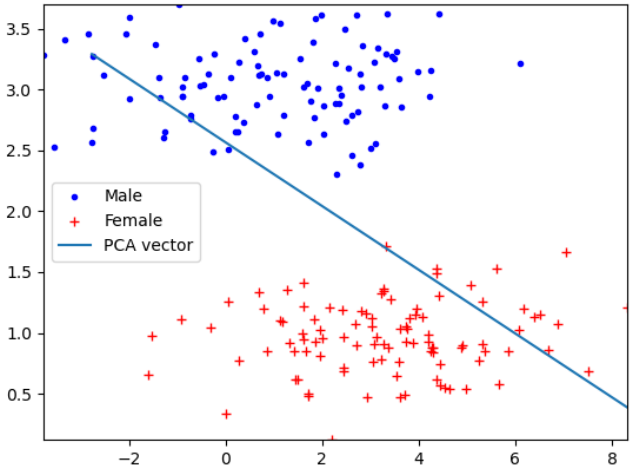
# Dimensionality Reduction

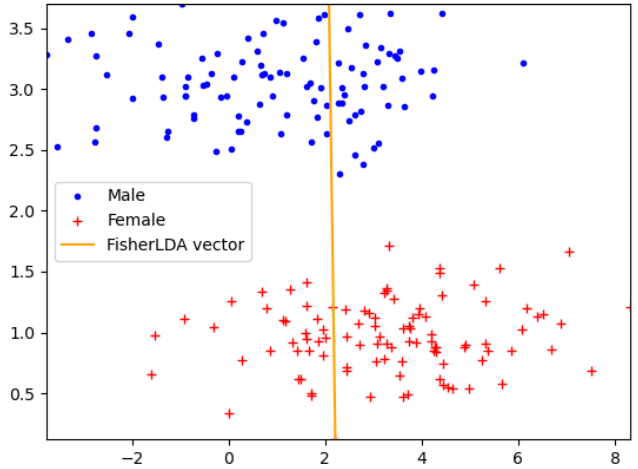
- First example of **unsupervised learning**!
- Learn a mapping from high-dimensional visible space **x** to a low-dimensional **latent** space **z**
- Notation:
  - Input data **x** with dimensionality $n$
  - Latent space **z** with dimensionality $k$

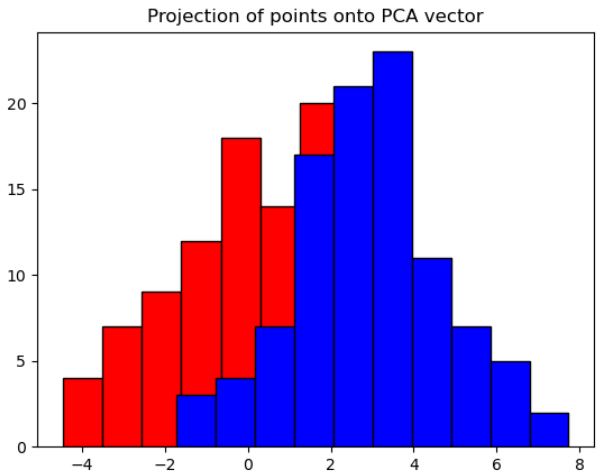
# Data Representation vs Data Classification

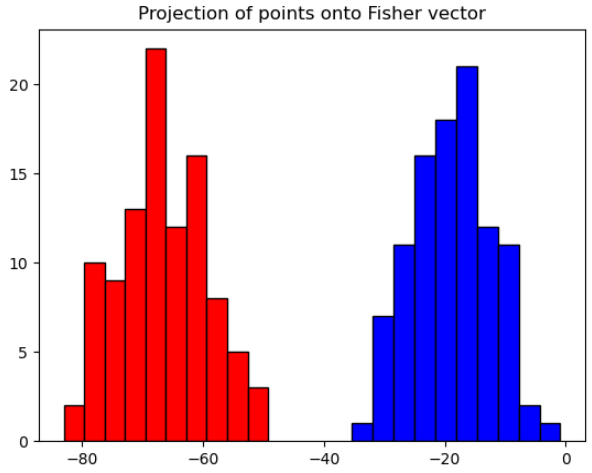
- **Data Representation:** Project data to lower dimensional space that *most accurately represents* the original data, e.g., PCA projects in directions of maximum variance

- **Data Classification:** Project data to a low dimensional space that preserves structure useful for classification, e.g., Fisher's LDA!




Kevin Murphy, *"Probabilistic Machine Learning: An Introduction"*, 2022

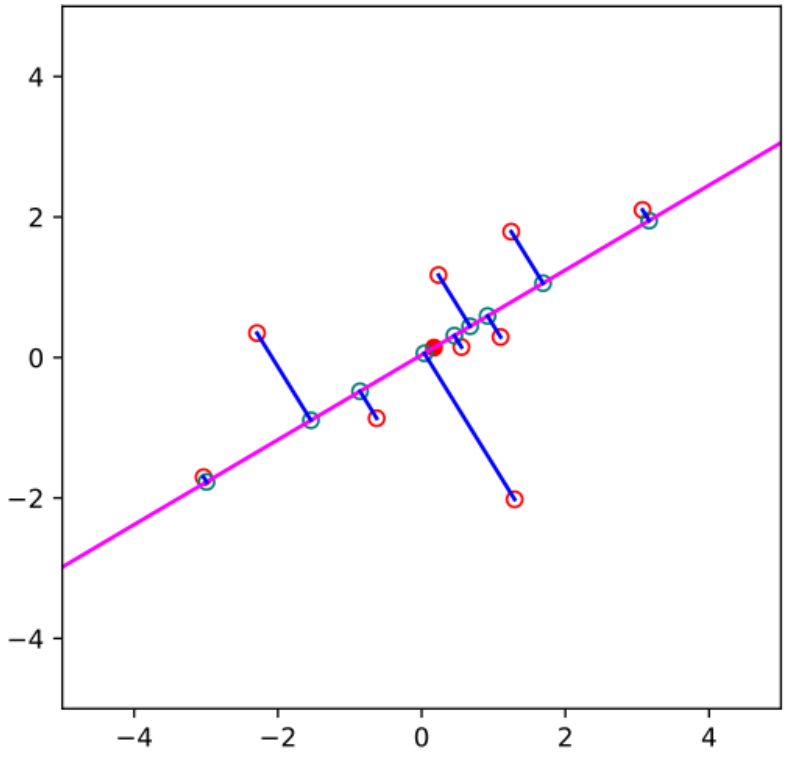
# Principal Component Analysis (PCA)

- Simplest and most widely used dimensionality reduction techniques
- **Key Idea:** Find **linear** and **orthogonal** projection directions of the high dimensional **x** to a low-dimensional, "good representation" **z**
- What is a "good representation"?

$$L(\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{x}^{(i)} - \text{decode}(\text{encode}(\mathbf{x}^{(i)}; \mathbf{W}); \mathbf{W})||_2^2$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_x)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_x)^\intercal = \frac{1}{N}\mathbf{X}_\mathbf{c}^\intercal\mathbf{X}_\mathbf{c}$$

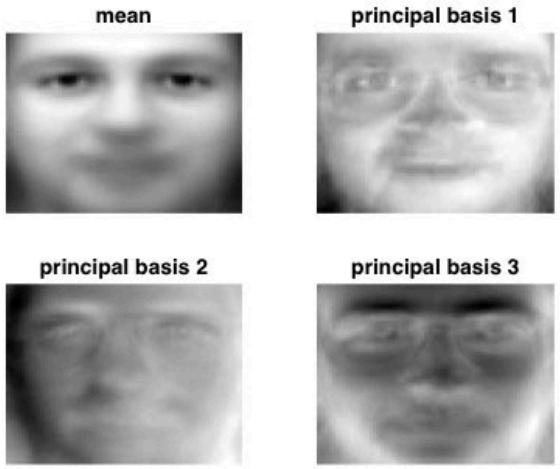
# PCA – Example (1)




Kevin Murphy, *"Probabilistic Machine Learning"*, 2022

# PCA – Example (2)

(a)




(b)

*Figure 20.3: a) Some randomly chosen 64 × 64 pixel images from the Olivetti face database. (b) The mean and the first three PCA components represented as images. Generated by code.probml.ai/book1/20.3.*

Kevin Murphy, *"Probabilistic Machine Learning"*, 2022

# PCA – Problem Setup

# PCA – Constrained Optimization

- Can solve **equality** constrained problems by forming a **Lagrangian**

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \quad \text{s.t.} \quad c_{i\in\mathcal{E}}(\boldsymbol{\theta}) = 0 \quad \Longrightarrow \quad \mathcal{L}(\boldsymbol{\theta}, \lambda) = \mathcal{L}(\boldsymbol{\theta}) + \sum_{i\in\mathcal{E}}^{m} \lambda_i c_i$$

***Lagrange multiplier $\lambda$*** *for $m$ constraints*

- At stationary point: $\mathcal{L}(\boldsymbol{\theta}, \lambda) = 0$

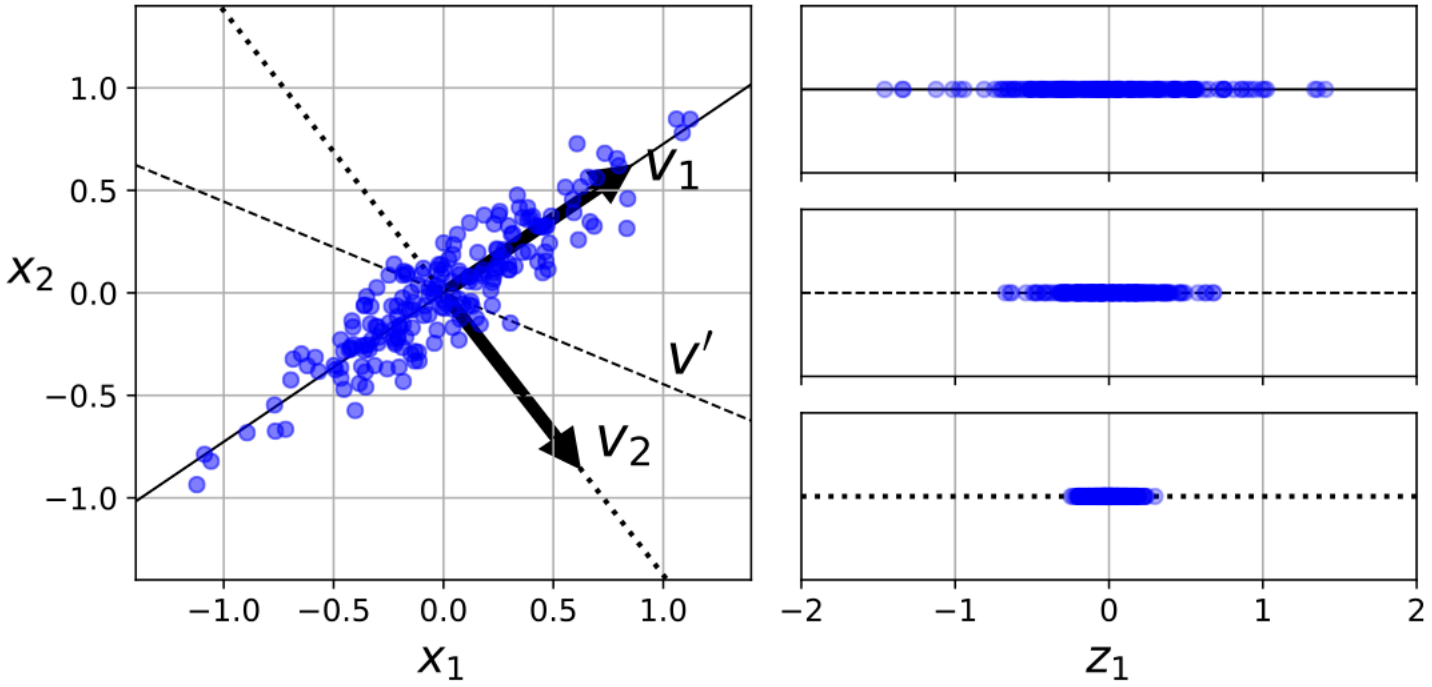
- Example on optimizing **quadratic forms**:

$$\max_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x}^\intercal \mathbf{A}\mathbf{x} \quad \text{s.t.} \quad ||\mathbf{x}||_2^2 = 1 \quad \Longrightarrow \quad \mathcal{L}(\mathbf{x}, \lambda) = \mathbf{x}^\intercal \mathbf{A}\mathbf{x} + \lambda(1 - \mathbf{x}^\intercal \mathbf{x})$$

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda) = 2\mathbf{A}^\intercal \mathbf{x} - 2\lambda\mathbf{x} = 0 \quad \Longrightarrow \quad \mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

*Optimal $x^*$ to min/max quadratic forms are* ***eigenvectors*** *of A*

# PCA – Why Max Variance?



# PCA – Finding 1st Component

# PCA – Finding 2nd Component

# PCA – Other Pointers

- PCA is a linear transformation that acts like a **coordinate rotation**
- Choose $k$ based on a metric known as **fraction of variance explained**:

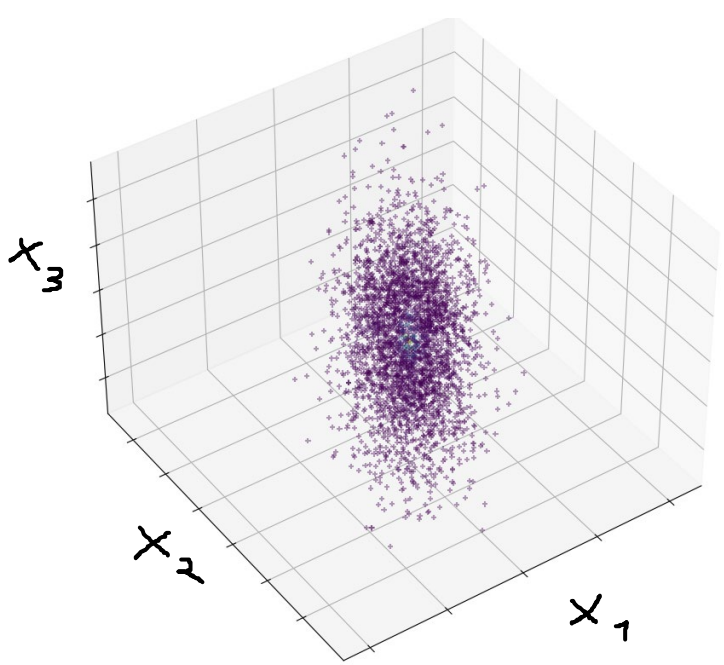
$$\mathrm{F}_k = \frac{\text{Variance Retained by PCs}}{\text{Total Variance in Data}} = \frac{\sum_{j=1}^{k} \lambda_j}{\mathrm{tr}(\boldsymbol{\Sigma})} \in [0, 1]$$

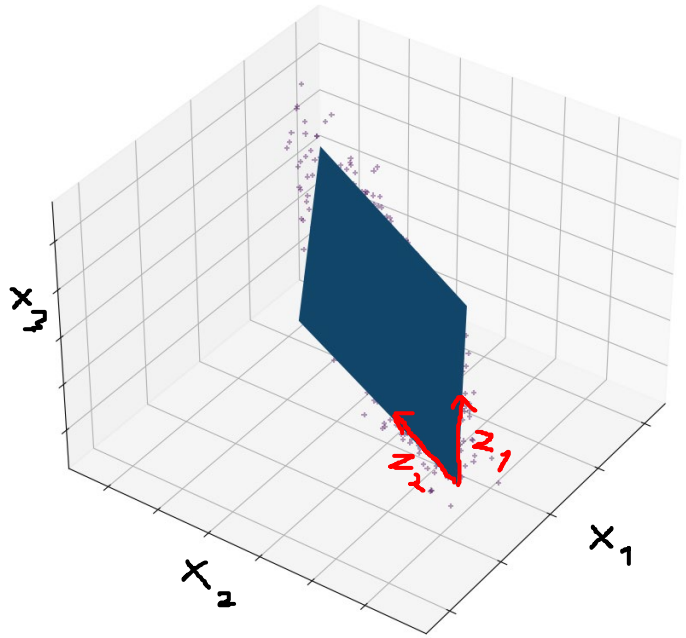
- Usually go for about 95% captured by first $k$ components

# PCA – Minimum Reconstruction Error

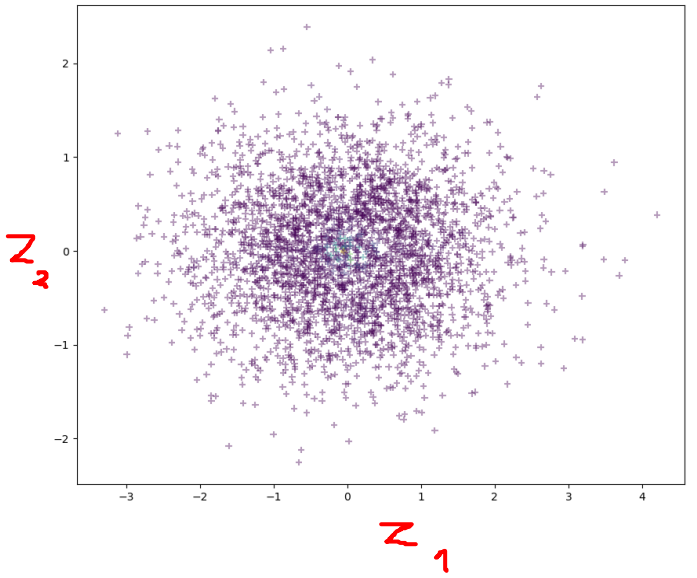
# PCA – Use-Cases

$\mathbf{x} \in \mathbb{R}^d \rightarrow \mathbf{z} \in \mathbb{R}^m$ where $m < d$




$x \in \mathbb{R}^3$ $z \in \mathbb{R}^2$




Useful for:

- Compression (remove redundancies)
- Visualization

Beware:

- Lossy transformation
- Prevent overfitting?

# Coding Break

# Concluding Remarks

- Introduced our first **unsupervised learning** algorithm: **PCA**
- Look at *"pca_example.py"* Python or corresponding Matlab script
- And Jupyter notebook comparing against MSE reconstruction:

https://github.com/mazrk7/EECE5644_IntroMLPR_LectureCode/blob/main/notebooks/unsupervised_learning/pca_dim_reduction.ipynb

- Check out additional notes uploaded on Canvas