Principal Component Analysis

CSC 461: Machine Learning

Fall 2022

Prof. Marco Alvarez
University of Rhode Island

Dimensionality reduction

- Basically …
  - ✓ mapping high-dimensional data into low dimensional data
  - ✓ can be as simple as dropping some dimensions or using a combination of all dimensions
- Given $\mathscr{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ with $\mathbf{x_i} \in \mathbb{R}^d$, find a representation $\mathscr{Z} = \{\mathbf{z_1}, \ldots, \mathbf{z_n}\}$ with $\mathbf{z_i} \in \mathbb{R}^{d'}$ and $d' \ll d$
  - ✓ properties should be preserved (e.g., variance, distances, neighborhood)

$$\begin{bmatrix} --- & \mathbf{x_1}^T & --- \\ & \vdots & \\ --- & \mathbf{x_n}^T & --- \end{bmatrix}_{n \times d} \Rightarrow \begin{bmatrix} \mathbf{z_n}^T \\ \vdots \\ \mathbf{z_n}^T \end{bmatrix}_{n \times d'}$$

Why dimensionality reduction?

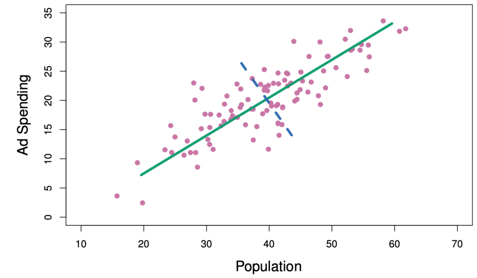
- Data visualization (2-d or 3-d)
- Preprocess data before machine learning
  - ✓ algorithms can focus on important features/patterns
  - ✓ training can be more efficient
- Removing noise and redundant information
- Data compression

Principal component analysis

(eigendecomposition formulation)

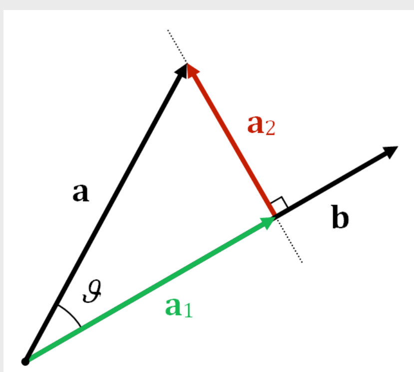
Principal component analysis

- Find projections of the data onto directions that maximize variance
  - ✓ directions are orthogonal to each other



https://www.dataschool.io/15-hours-of-expert-machine-learning-videos/

Preliminaries



The **vector projection** of vector **a** onto a nonzero vector **b** is the orthogonal projection of **a** onto a straight line parallel to **b**.

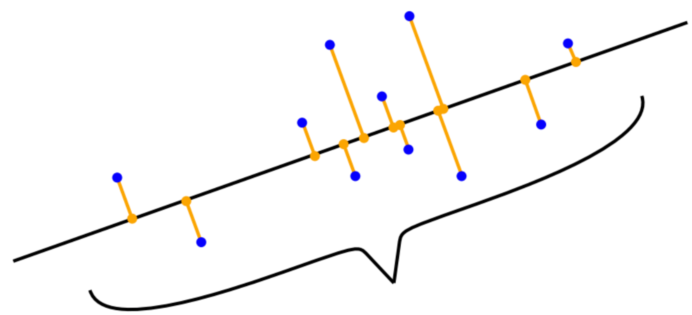
$$\mathbf{a}_1 = a_1 \hat{\mathbf{b}}$$

where a_1 is a scalar, called the **scalar projection** of **a** onto **b**, and $\hat{\mathbf{b}}$ is the unit vector in the direction of **b**. The scalar projection is defined as:

$$a_1 = \mathbf{a} \cdot \hat{\mathbf{b}}$$

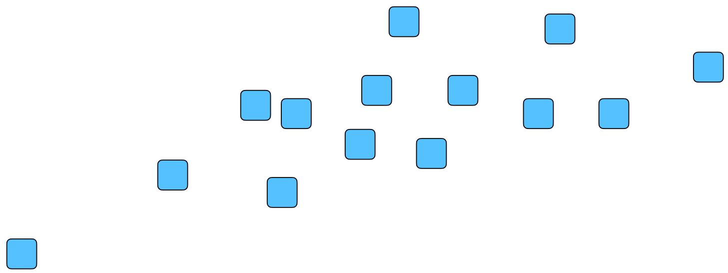
https://en.wikipedia.org/wiki/Vector_projection

Data projection on a line



Idea: finding a line (subspace) that maintains as much variance (spread) as possible when the data is projected onto this subspace.

What is the best line?

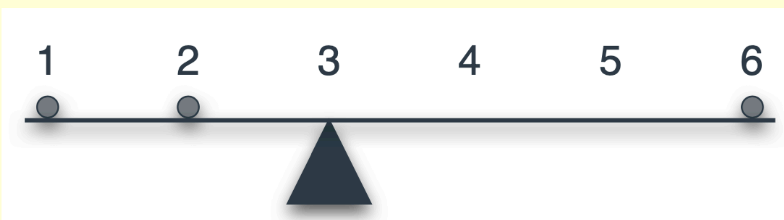


Idea: minimize sum of square distances to the line (or maximize the sum of squares of the scalar projections)

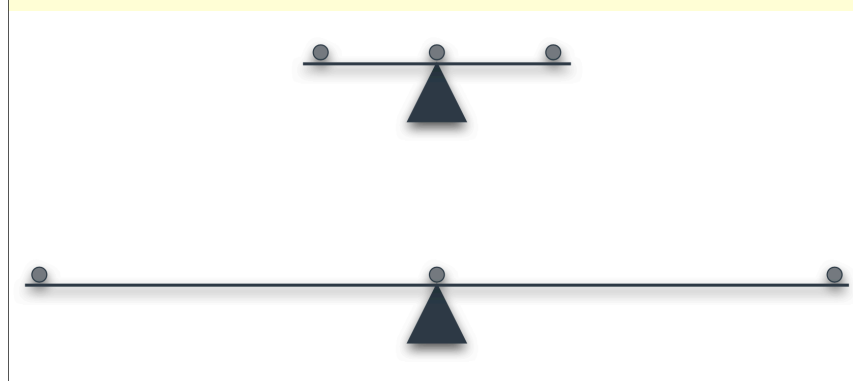
Disclaimer

- The following slides contain figures adapted from:
  - ✓ Unsupervised Learning SERRANO.ACADEMY the art of understanding
    - ✓ https://serrano.academy/unsupervised-learning/
- Video:
  - ✓ https://www.youtube.com/watch?v=g-Hb26agBFg

The mean



Doesn’t always help

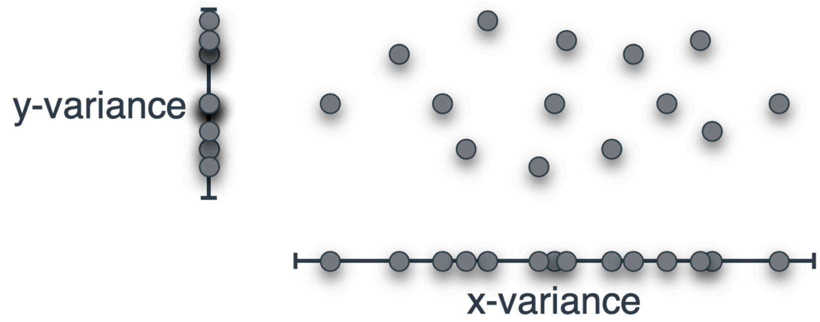


The variance

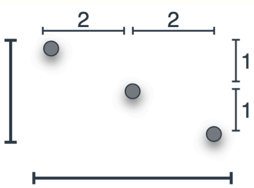
$$\text{Variance} = \frac{1^2+0^2+1^2}{3} = 2/3$$

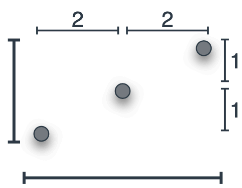
$$\text{Variance} = \frac{5^2+0^2+5^2}{3} = 10/3$$

Variance and data



Doesn’t always help

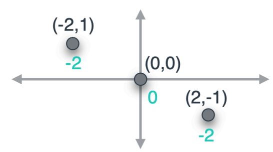


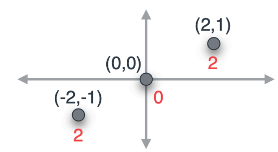


$$\text{x-variance} = \frac{2^2+0^2+2^2}{3} = 8/3$$

$$\text{y-variance} = \frac{1^2+0^2+1^2}{3} = 2/3$$

Covariance



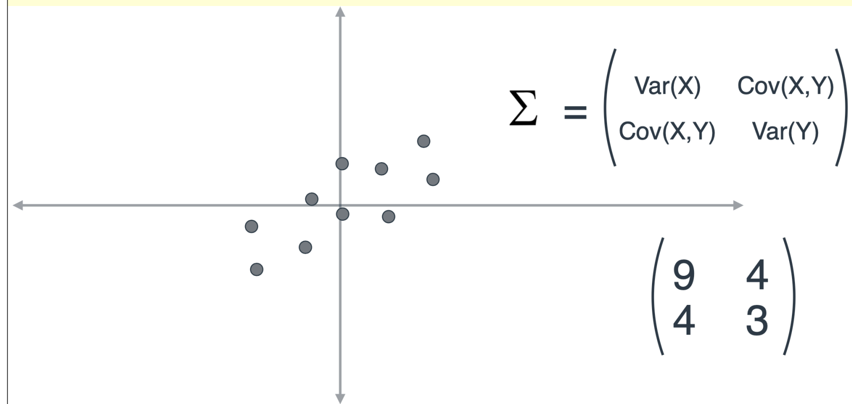


$$\text{covariance} = \frac{(-2)+0+(-2)}{3} = -4/3$$

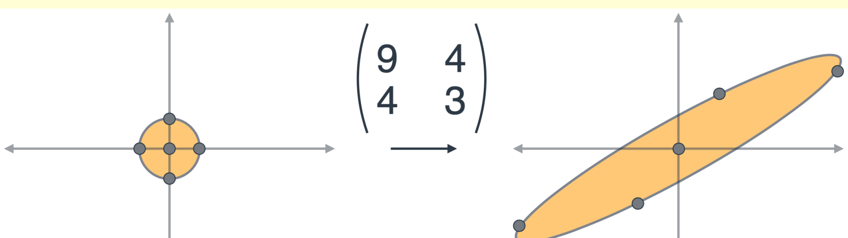
$$\text{covariance} = \frac{2+0+2}{3} = 4/3$$

Covariance

Covariance matrix

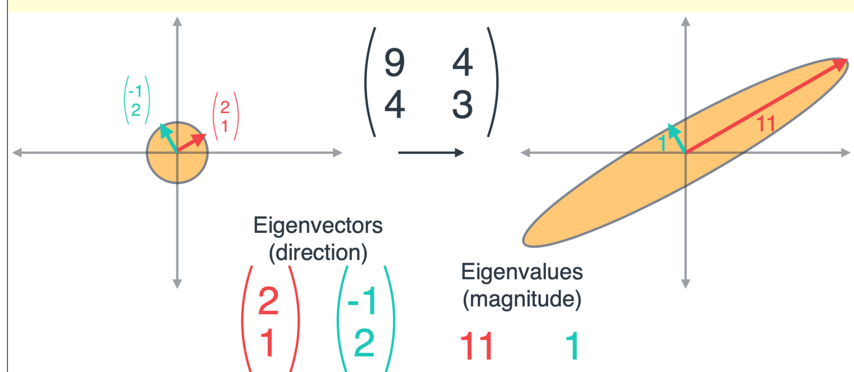


Linear transformations

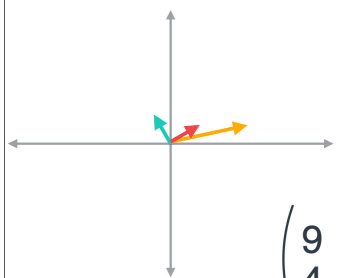


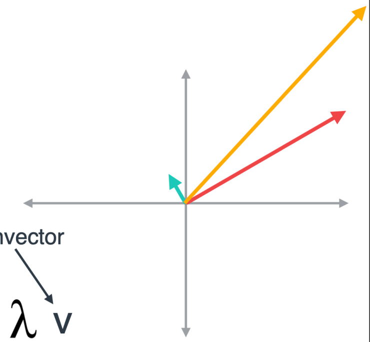
(x, y) ⟶	(9x+4y, 4x+3y)
(0,0)	(0,0)
(1,0)	(9,4)
(0,1)	(4,3)
(-1,0)	(-9,-4)
(0,-1)	(-4,-3)

Eigenvectors and eigenvalues



Eigenvectors and eigenvalues



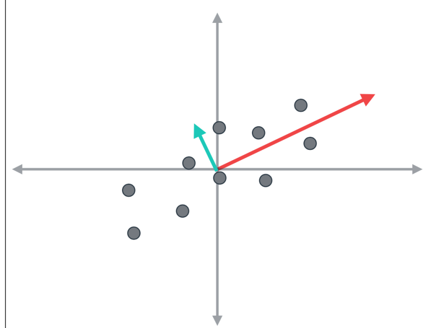


Eigenvector

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} v = \lambda\, v$$

Eigenvalue

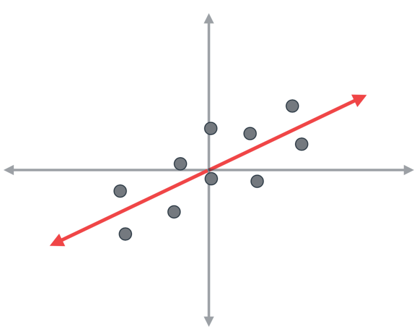
Principal component analysis (PCA)



$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 2 \end{pmatrix}$	Eigenvectors (direction)
11	1	Eigenvalues (magnitude)

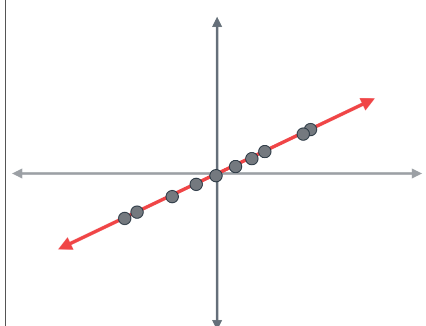
Principal component analysis (PCA)



$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$	Eigenvectors (direction)
11	Eigenvalues (magnitude)

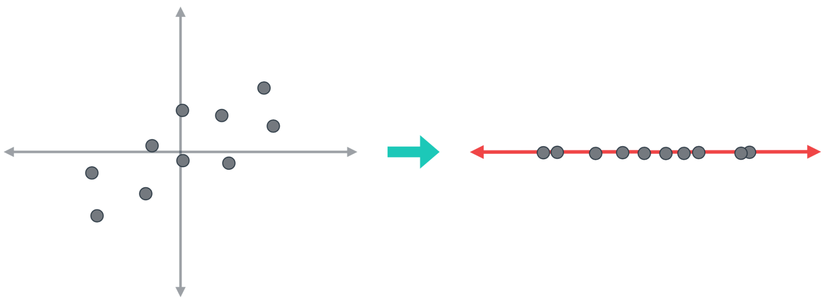
Principal component analysis (PCA)



$$\Sigma = \begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$$

$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$	Eigenvectors (direction)
11	Eigenvalues (magnitude)

Principal component analysis (PCA)



More formally

- Given $\mathscr{D} = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ with $\mathbf{x_i} \in \mathbb{R}^d$
- Find $\mathscr{Z} = \{\mathbf{z_1}, \ldots, \mathbf{z_n}\}$ with $\mathbf{z_i} \in \mathbb{R}^{d'}$ and $d' \ll d$
- Example from 3-d to 2-d

$$\mathbf{u_1} \in \mathbb{R}^3, \mathbf{u_2} \in \mathbb{R}^3$$

$$\mathbf{z_i} = [\mathbf{u_1}^T\mathbf{x_i}, \mathbf{u_2}^T\mathbf{x_i}], \mathbf{z_i} \in \mathbb{R}^2$$

PCA

- Input
  - ✓ data must be centered

$$X = \begin{bmatrix} --- & \mathbf{x_1}^T & --- \\ & \vdots & \\ --- & \mathbf{x_n}^T & --- \end{bmatrix}_{n \times d}$$

$$\frac{1}{n}\sum_{i=1}^{n} c_i = 0, \quad \text{for all columns } c \text{ in } X$$

- Output
  - ✓ orthonormal principal components $\mathbf{u_1}, \ldots \ \mathbf{u_k}$

Eigendecomposition of symmetric matrices

As a special case, for every n × n real symmetric matrix, the eigenvalues are real and the eigenvectors can be chosen real and orthonormal. Thus a real symmetric matrix A can be decomposed as:

$$A = Q\Lambda Q^T$$

where Q is a matrix whose columns are eigenvectors of A,
and Λ is a diagonal matrix whose entries are the eigenvalues of A.

https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix

First principal component

$\|\mathbf{u_1}\|_2 = 1$

$$\arg\max_{\mathbf{u_1}} \sum_{i=1}^{n} (\mathbf{x_i}^T \mathbf{u_1})^2$$

$$\arg\max_{\mathbf{u_1}} \mathbf{u_1}^T \underbrace{\left[\sum_{i=1}^{n} \mathbf{x_i}\mathbf{x_i}^T \right]}_{XX^T} \mathbf{u_1}$$

solution $\mathbf{u_1}$ is the first eigenvector of XX^T

Second principal component

$\|\mathbf{u_2}\|_2 = 1, \quad \mathbf{u_2}^T \mathbf{u_1} = 0$

$$\arg\max_{\mathbf{u_2}} \sum_{i=1}^{n} (\mathbf{x_i}^T \mathbf{u_2})^2$$

$$\arg\max_{\mathbf{u_2}} \mathbf{u_2}^T \left(XX^T\right) \mathbf{u_2}$$

solution $\mathbf{u_2}$ is the second eigenvector of XX^T

PCA algorithm

- Assuming X and K as inputs
  - ✓ Center the data in X
  - ✓ Compute eigendecomposition $\left(U \Lambda U^T\right)$ of XX^T
  - ✓ Return the top K eigenvectors from U
- Eigenvectors can be use for projecting the data in lower dimensions