# Principal Component Analysis

## A different perspective & Practical Considerations

Thomas Wedenig

April 24, 2024

Institute of Theoretical Computer Science
Graz University of Technology, Austria

- In Machine Learning, we often deal with **high-dimensional data** (features)
- e.g., an image $X \in \mathbb{R}^{1024 \times 1024}$ ($\approx$ 1 million dimensions) 🤯

# MOTIVATION

- In Machine Learning, we often deal with **high-dimensional data** (features)
- e.g., an image $X \in \mathbb{R}^{1024 \times 1024}$ ($\approx$ 1 million dimensions) 🤯

## Dimensionality Reduction

- Many of these features might be **redundant**
- We wish to find a **lower-dimensional** (compressed) representation of our data
- Useful for . . .
  - Feature extraction
  - Visualization
  - Reducing computational load
  - Compression *per se* (smaller file size)

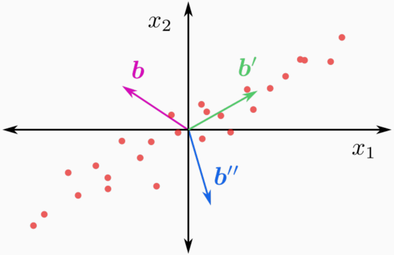
## Idea 🤔

- **Variance** in the data amounts to **information** ❗
  - e.g., a feature that is constant for all data points is **not informative**

## Idea 🤔

- **Variance** in the data amounts to **information** ❗
  - e.g., a feature that is constant for all data points is **not informative**

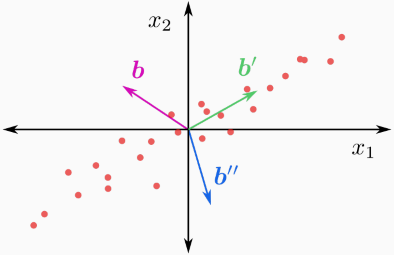
$\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with $\mathbf{x}^{(i)} \in \mathbb{R}^2$



## Idea 🤔

- **Variance** in the data amounts to **information** ❗
  - e.g., a feature that is constant for all data points is **not informative**

$\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with $\mathbf{x}^{(i)} \in \mathbb{R}^2$




- A given $\mathbf{b}$ projects $\mathbf{x}^{(i)}$ to $z^{(i)} \in \mathbb{R}$

$$z^{(i)} = \mathbf{b}^T \mathbf{x}^{(i)}$$

- In the original coordinate system, the projection is then $\hat{\mathbf{x}} = z\mathbf{b}$
- Find $\mathbf{b}$ (unit length) such that the **variance** in the projections $z$ is maximized, i.e.,

$$\mathbf{b}^* = \underset{\mathbf{b}^T\mathbf{b}=1}{\operatorname{argmax}} \ \operatorname{Var}\left(\mathbf{b}^T\mathbf{x}^{(1)}, \ldots, \mathbf{b}^T\mathbf{x}^{(N)}\right)$$

- Let's generalize this to $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ with $\mathbf{x}^{(i)} \in \mathbb{R}^D$

## Orthonormal Basis

For each subspace $U \subseteq \mathbb{R}^D$ there exists a set of **orthonormal basis vectors** that span $U$.

- Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_D\}$ be an orthonormal basis of $\mathbb{R}^D$, collected in $B = (\mathbf{b}_1 \ \mathbf{b}_2 \ \ldots \ \mathbf{b}_D)$
- $B$ is orthonormal, i.e., $B^T B = I$ and thus, $B^{-1} = B^T$
  - Columns have unit norm and are pairwise orthogonal

## Change of Basis

Recall that any $\mathbf{x} \in \mathbb{R}^D$ can be expressed as coordinates w.r.t. $B$ (called $\mathbf{z}$):

$$\mathbf{x} = B\mathbf{z} \Leftrightarrow B^{-1}\mathbf{x} = \mathbf{z} \Leftrightarrow B^T\mathbf{x} = \mathbf{z}$$

- $\mathbf{x} = B\mathbf{z}$ and $B^T\mathbf{x} = \mathbf{z}$
- We can transform $\mathbf{z}$ back into the original coordinate system: $\mathbf{x} = \underbrace{BB^T}_{I}\mathbf{x}$
- So far we have just played with coordinate systems, no compression yet
  - Since we have $D$ basis vectors
- Let's compress $\mathbf{x} \in \mathbb{R}^D$ into a representation $\mathbf{z} \in \mathbb{R}^M$ with $M < D$

- Let $B_M = (\mathbf{b}_1 \ \dots \ \mathbf{b}_M)$ and $B_R = (\mathbf{b}_{M+1} \ \dots \ \mathbf{b}_D)$
- Assume we are given $B_M$. How do we project $\mathbf{x}$ into the subspace spanned by $B_M$?

**Optimal Projection**

Given $\mathbf{x} \in \mathbb{R}^D$ and $B_M$, find $\mathbf{z}^* \in \mathbb{R}^M$ such that

$$\mathbf{z}^* = \operatorname*{argmin}_{\mathbf{z}} ||B_M\mathbf{z} - \mathbf{x}||_2^2.$$

- Let $B_M = (\mathbf{b}_1 \ \dots \ \mathbf{b}_M)$ and $B_R = (\mathbf{b}_{M+1} \ \dots \ \mathbf{b}_D)$
- Assume we are given $B_M$. How do we project $\mathbf{x}$ into the subspace spanned by $B_M$?

## Optimal Projection

Given $\mathbf{x} \in \mathbb{R}^D$ and $B_M$, find $\mathbf{z}^* \in \mathbb{R}^M$ such that

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \, ||B_M \mathbf{z} - \mathbf{x}||_2^2.$$

## Solution

- Looks just like finding parameters for linear regression !
- Closed form solution recovers **orthogonal projection**:

$$\mathbf{z}^* = ({B_M}^T B_M)^{-1} {B_M}^T \mathbf{x} = {B_M}^T \mathbf{x}.$$

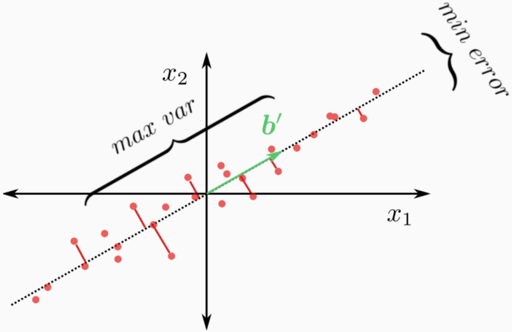
- Transform back into original coordinates: $\hat{\mathbf{x}} = B_M {B_M}^T \mathbf{x}$

- $\hat{\mathbf{x}} = B_M {B_M}^T \mathbf{x}$
- Note that $B_M {B_M}^T$ **is not identity anymore** ❗
  - The inverse of $B_M$ does not exist
- How to find a *good* $B_M$?

**Idea** 🤔

We want to find $B_M$ s.t. projections have **minimal average squared projection error**!

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \underbrace{B_M {B_M}^T \mathbf{x}^{(i)}}_{\hat{\mathbf{x}}^{(i)}}\|_2^2$$



- $B = (\mathbf{b}_1 \; \mathbf{b}_2 \; \ldots \; \mathbf{b}_D)$, $B_M = (\mathbf{b}_1 \; \ldots \; \mathbf{b}_M)$ and $B_R = (\mathbf{b}_{M+1} \; \ldots \; \mathbf{b}_D)$
- Note that

$$BB^T = \sum_{j=1}^{N} \mathbf{b}_j \mathbf{b}_j^T = \sum_{j=1}^{M} \mathbf{b}_j \mathbf{b}_j^T + \sum_{j=M+1}^{D} \mathbf{b}_j \mathbf{b}_j^T = B_M B_M^T + B_R B_R^T$$

- Since $\mathbf{x} = BB^T\mathbf{x}$, we have $\mathbf{x} = \left(B_M B_M^T + B_R B_R^T\right)\mathbf{x}$
- We rewrite the residual

$$\mathbf{x} - \hat{\mathbf{x}} = \underbrace{\left(B_M B_M^T + B_R B_R^T\right)\mathbf{x}}_{\mathbf{x}} - \underbrace{B_M {B_M}^T \mathbf{x}}_{\hat{\mathbf{x}}} = B_R B_R^T \mathbf{x}$$

- The error is the **projection on the orthogonal complement of the principal subspace**
- We thus minimize

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \|B_R B_R^T \mathbf{x}^{(i)}\|_2^2$$

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \|B_R B_R^T \mathbf{x}^{(i)}\|_2^2 = \frac{1}{N}\sum_{i=1}^{N} \left(B_R B_R^T \mathbf{x}^{(i)}\right)^T \left(B_R B_R^T \mathbf{x}^{(i)}\right)$$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)T} B_R \underbrace{B_R^T B_R}_{I} B_R^T \mathbf{x}^{(i)}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)T} B_R B_R^T \mathbf{x}^{(i)}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)T} \sum_{j=M+1}^{D} \mathbf{b}_j \mathbf{b}_j^T \mathbf{x}^{(i)}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \sum_{j=M+1}^{D} \mathbf{x}^{(i)T} \mathbf{b}_j \mathbf{b}_j^T \mathbf{x}^{(i)} = \frac{1}{N}\sum_{i=1}^{N} \sum_{j=M+1}^{D} \mathbf{b}_j^T \mathbf{x}^{(i)} \mathbf{x}^{(i)T} \mathbf{b}_j$$

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=M+1}^{D}\mathbf{b}_j^T\mathbf{x}^{(i)}\mathbf{x}^{(i)T}\mathbf{b}_j$$

$$= \sum_{j=M+1}^{D}\mathbf{b}_j^T\underbrace{\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}\mathbf{x}^{(i)T}\right)}_{C}\mathbf{b}_j$$

**Data Covariance Matrix**

$C$ is the **empirical covariance matrix** of $X$:

$$C = \frac{1}{N}X^TX \qquad X = \begin{pmatrix} \mathbf{x}^{(1)^T} \\ \mathbf{x}^{(2)^T} \\ \vdots \\ \mathbf{x}^{(N)^T} \end{pmatrix}$$

How to pick $B_R = (\mathbf{b}_{M+1} \; \dots \; \mathbf{b}_D)$ such that $\sum_{j=M+1}^{D}\mathbf{b}_j^T C\mathbf{b}_j$ is minimized?

# Minimizing the Loss (cont.)

$$\min_{\mathbf{b}_{M+1},\ldots,\mathbf{b}_D} \mathbf{b}_{M+1}^T C \mathbf{b}_{M+1} + \cdots + \mathbf{b}_D^T C \mathbf{b}_D$$

- Consider $\min_{\mathbf{b}} \mathbf{b}^T C \mathbf{b}$ subject to $\mathbf{b}^T \mathbf{b} = 1$

$$\min_{\mathbf{b}_{M+1},\ldots,\mathbf{b}_D} \mathbf{b}_{M+1}^T C \mathbf{b}_{M+1} + \cdots + \mathbf{b}_D^T C \mathbf{b}_D$$

- Consider $\min_{\mathbf{b}} \mathbf{b}^T C \mathbf{b}$ subject to $\mathbf{b}^T \mathbf{b} = 1$
- **Constrained minimization** $\rightarrow$ setup **Lagrangian**
- $L(\mathbf{b}, \lambda) = \mathbf{b}^T C \mathbf{b} - \lambda \underbrace{(\mathbf{b}^T \mathbf{b} - 1)}_{\text{Constraint}}$
- $\nabla_\lambda L(\mathbf{b}) = 1 - \mathbf{b}^T \mathbf{b} = 0 \Leftrightarrow \mathbf{b}^T \mathbf{b} = 1$ (just the constraint)
- $\nabla_{\mathbf{b}} L(\mathbf{b}, \lambda) = 2C\mathbf{b} - 2\lambda \mathbf{b} = 0 \Leftrightarrow C\mathbf{b} = \lambda \mathbf{b}$
- The optimal $\mathbf{b}$ is an **eigenvector** of $C$
- Since $\mathbf{b}^T C \mathbf{b} = \mathbf{b}^T (\lambda \mathbf{b}) = \lambda \mathbf{b}^T \mathbf{b} = \lambda$, pick $\mathbf{b}$ to be the eigenvector associated with the **smallest** eigenvalue

- Since $C$ is symmetric, its eigenvectors (with different eigenvalues) are **orthogonal** to each other (Spectral Theorem)
- Pick $B_R = (\mathbf{b}_{M+1} \ \ldots \ \mathbf{b}_D)$ to be the orthonormal eigenvectors of $C$ with the **smallest eigenvalues**
- $B_R$ is the orthogonal complement of $B_M = (\mathbf{b}_1 \ \ldots \ \mathbf{b}_M)$
- Thus, we construct $B_M$ with the **remaining** eigenvectors (with the **largest eigenvalues**)

# Summary

## TL;DR

To **minimize projection error**, pick the subspace spanned by the **orthonormal eigenvectors** of $C$ that have the **largest eigenvalues**.

## Duality

Minimizing Projection Error $\Leftrightarrow$ Maximizing Variance in the Projections

- We need to find the $M$ eigenvectors **with the largest eigenvalues** of $X^T X$
- How do we *actually* do this? 🤔
- `np.linalg.eigh(X.T @ X)` works in theory, but is **wasteful**
    - $X^T X \in \mathbb{R}^{D \times D}$
    - We compute $D$ eigenvalues, although we only need the top-$M$ ($M \ll D$) 😩
    - Use **algorithms that only compute** the $D$ largest eigenvalues (and their eigenvectors) ❗

What if $D$ is large (e.g. images) and we have $N \ll D$ data points? 🤔

- We want to find an eigenvector $\mathbf{b}$ of $C = \frac{1}{N}X^T X \in \mathbb{R}^{D \times D}$ (with eigenvalue $\lambda$)

$$\frac{1}{N}X^T X\mathbf{b} = \lambda\mathbf{b}$$

$$\frac{1}{N}XX^T \underbrace{X\mathbf{b}}_{\mathbf{a}} = \lambda \underbrace{X\mathbf{b}}_{\mathbf{a}}$$

$$\frac{1}{N}XX^T\mathbf{a} = \lambda\mathbf{a}$$

What if $D$ is large (e.g. images) and we have $N \ll D$ data points? 🤔

- We want to find an eigenvector $\mathbf{b}$ of $C = \frac{1}{N}X^TX \in \mathbb{R}^{D\times D}$ (with eigenvalue $\lambda$)

$$\frac{1}{N}X^TX\mathbf{b} = \lambda\mathbf{b}$$

$$\frac{1}{N}XX^T\underbrace{X\mathbf{b}}_{\mathbf{a}} = \lambda\underbrace{X\mathbf{b}}_{\mathbf{a}}$$

$$\frac{1}{N}XX^T\mathbf{a} = \lambda\mathbf{a}$$

- $\mathbf{a}$ is an eigenvector of $\frac{1}{N}XX^T \in \mathbb{R}^{N\times N}$
- If we find $\mathbf{a}$, we can convert it into an eigenvector of $C$:

$$\frac{1}{N}XX^T\mathbf{a} = \lambda\mathbf{a}$$

$$\frac{1}{N}X^TX(X^T\mathbf{a}) = \lambda(X^T\mathbf{a})$$

$$C(X^T\mathbf{a}) = \lambda(X^T\mathbf{a})$$

# PCA Demo with `scikit-learn`