Introduction to Data Mining
04 - Principal Component Analysis

Benjamin Paaßen

WS 2023/2024, Bielefeld University

Registration for Presentations

- Registration for homework presentations is open.

Link: Registration

Registration for Presentations

- Registration for homework presentations is open.
- First come-first serve principle.

Link: Registration

Registration for Presentations

- Registration for homework presentations is open.
- First come-first serve principle.
- Submit other questions for the tutorials: Link

Link: Registration

Moodle

- Moodle available now in Lernraum, including groups and submissions

Moodle

- Moodle available now in Lernraum, including groups and submissions
- Voluntary for now; your feedback is appreciated!

Outline for this lecture

UNIVERSITÄT BIELEFELD
Faculty of Technology

- (Almost) Full derivation of PCA, based on Ren and MacKay (2019)
- Factor analysis
- Factor rotations

Motivation

- Explain data as a **linear combination** of basis vectors (or factors) $v_1, \ldots, v_n$

Motivation

- Explain data as a linear combination of basis vectors (or factors) $v_1, \ldots, v_n$
- $x = z_1 \cdot v_1 + \ldots + z_n \cdot v_n$

Motivation

- Explain data as a **linear combination** of basis vectors (or factors) $v_1, \ldots, v_n$
- $x = z_1 \cdot v_1 + \ldots + z_n \cdot v_n$

Examples:

Motivation

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Explain data as a **linear combination** of basis vectors (or factors) $v_1, \ldots, v_n$
- $x = z_1 \cdot v_1 + \ldots + z_n \cdot v_n$

Examples:

- Underlying skills in educational data

Motivation

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Explain data as a linear combination of basis vectors (or factors) $v_1, \ldots, v_n$
- $x = z_1 \cdot v_1 + \ldots + z_n \cdot v_n$

Examples:

- Underlying skills in educational data
- Fourier transform

Motivation

- Explain data as a linear combination of basis vectors (or factors) $v_1, \ldots, v_n$
- $x = z_1 \cdot v_1 + \ldots + z_n \cdot v_n$

Examples:

- Underlying skills in educational data
- Fourier transform
- Eigenfaces

Spooky example

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Assume we are detectives and try to find the underlying patterns behind a serious of deaths

Spooky example

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Assume we are detectives and try to find the underlying patterns behind a serious of deaths

mysterious loss of blood	1	1	0	1	1	1	1	1	1	1	1	0
two punctures on the neck	1	1	0	1	1	1	1	1	1	1	1	0
slash and bite wounds	0	0	1	0	0	0	0	0	0	0	0	1
paw prints	0	0	1	0	0	0	0	0	0	0	0	1
animal hair	0	0	1	0	0	0	0	0	0	0	0	1
full moon	0	0	1	0	0	0	0	0	0	0	0	1
age	78	49	44	24	29	31	50	63	73	27	62	49

Spooky example

- Assume we are detectives and try to find the underlying patterns behind a serious of deaths

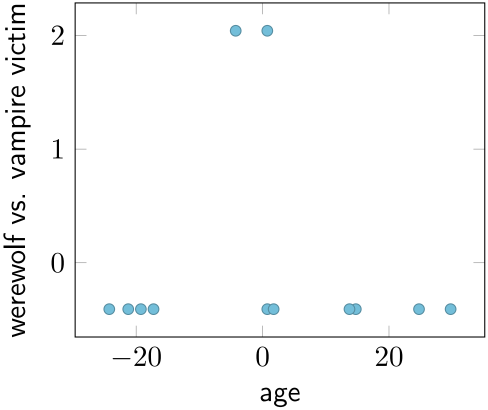
mysterious loss of blood	1	1	0	1	1	1	1	1	1	1	1	0
two punctures on the neck	1	1	0	1	1	1	1	1	1	1	1	0
slash and bite wounds	0	0	1	0	0	0	0	0	0	0	0	1
paw prints	0	0	1	0	0	0	0	0	0	0	0	1
animal hair	0	0	1	0	0	0	0	0	0	0	0	1
full moon	0	0	1	0	0	0	0	0	0	0	0	1
age	78	49	44	24	29	31	50	63	73	27	62	49

- Which patterns jump out at you?

Spooky example (continued)

UNIVERSITÄT BIELEFELD
Faculty of Technology

feature	v_1	v_2
mysterious loss of blood	0.00	-0.41
two punctures on the neck	0.00	-0.41
slash and bite wounds	0.00	0.41
paw prints	0.00	0.41
animal hair	0.00	0.41
full moon	0.00	0.41
age	1.00	0.00



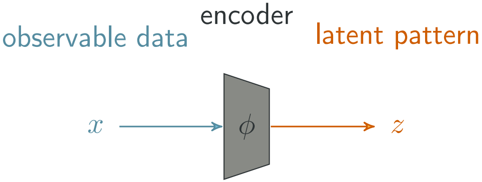
Principal Component Analysis

Setup

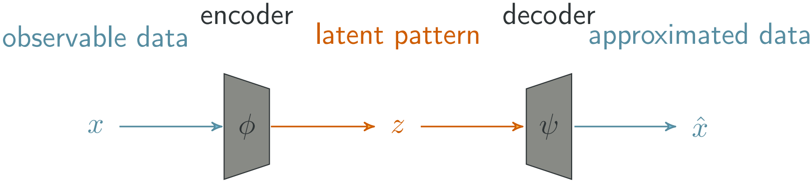
observable data

x

Setup



Setup


encoder
decoder
observable data
latent pattern
approximated data
x
ϕ
z
ψ
$\hat{x}$

Formalization

- Let $x_1, \ldots, x_N \in \mathbb{R}^m$ be our example data set

Formalization

- Let $x_1, \ldots, x_N \in \mathbb{R}^m$ be our example data set
- We try to find encoding function ϕ and decoding function ψ s.t. reconstruction error is minimized:

$$\min_{\phi,\psi} \quad \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2$$

Formalization

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Let $x_1, \ldots, x_N \in \mathbb{R}^m$ be our example data set
- We try to find encoding function ϕ and decoding function ψ s.t. reconstruction error is minimized:

$$\min_{\phi,\psi} \quad \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2$$

- **But** both ϕ and ψ are affine maps

$$z = \phi(x) = \boldsymbol{U} \cdot x + a \qquad \text{where } \boldsymbol{U} \in \mathbb{R}^{n \times m}, a \in \mathbb{R}^n$$
$$\hat{x} = \psi(z) = \boldsymbol{V} \cdot z + b \qquad \text{where } \boldsymbol{V} \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

Formalization

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Let $x_1, \ldots, x_N \in \mathbb{R}^m$ be our example data set
- We try to find encoding function ϕ and decoding function ψ s.t. reconstruction error is minimized:

$$\min_{\phi,\psi} \quad \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2$$

- **But** both ϕ and ψ are affine maps

$$z = \phi(x) = \boldsymbol{U} \cdot x + a \qquad \text{where } \boldsymbol{U} \in \mathbb{R}^{n \times m}, a \in \mathbb{R}^n$$
$$\hat{x} = \psi(z) = \boldsymbol{V} \cdot z + b \qquad \text{where } \boldsymbol{V} \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$

- Trick question: What is the solution for $n \geq m$?

Outline for the full derivation of PCA

UNIVERSITÄT BIELEFELD
Faculty of Technology

1. Do some geometry to understand decoding better – and why $\boldsymbol{V}$ can be (semi-)orthogonal without loss of generality

Outline for the full derivation of PCA

1. Do some geometry to understand decoding better – and why $\boldsymbol{V}$ can be (semi-)orthogonal without loss of generality
2. Understand why a can have a special form without loss of generality

Outline for the full derivation of PCA

1. Do some geometry to understand decoding better – and why $\boldsymbol{V}$ can be (semi-)orthogonal without loss of generality
2. Understand why a can have a special form without loss of generality
3. Find optimal b to minimize reconstruction error

Outline for the full derivation of PCA

1. Do some geometry to understand decoding better – and why $\boldsymbol{V}$ can be (semi-)orthogonal without loss of generality
2. Understand why a can have a special form without loss of generality
3. Find optimal b to minimize reconstruction error
4. Find optimal $\boldsymbol{U}$ to minimize reconstruction error

Outline for the full derivation of PCA

1. Do some geometry to understand decoding better – and why $\boldsymbol{V}$ can be (semi-)orthogonal without loss of generality
2. Understand why a can have a special form without loss of generality
3. Find optimal b to minimize reconstruction error
4. Find optimal $\boldsymbol{U}$ to minimize reconstruction error
5. Do some more geometry to find an easier optimization target for $\boldsymbol{V}$

Outline for the full derivation of PCA

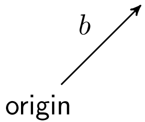
1. Do some geometry to understand decoding better – and why $\boldsymbol{V}$ can be (semi-)orthogonal without loss of generality
2. Understand why a can have a special form without loss of generality
3. Find optimal b to minimize reconstruction error
4. Find optimal $\boldsymbol{U}$ to minimize reconstruction error
5. Do some more geometry to find an easier optimization target for $\boldsymbol{V}$
6. Find optimal $\boldsymbol{V}$

Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.

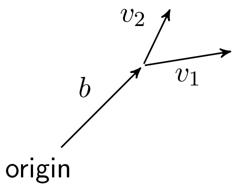
Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



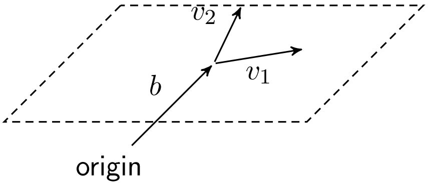
Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



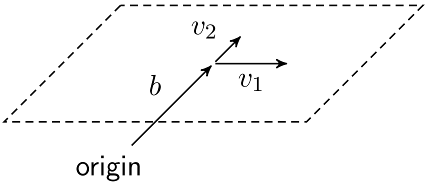
Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



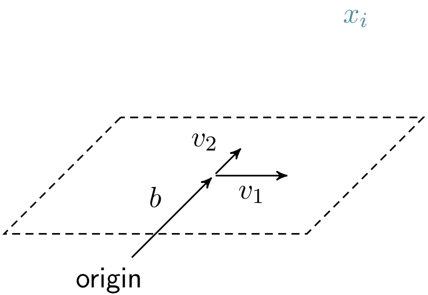
Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



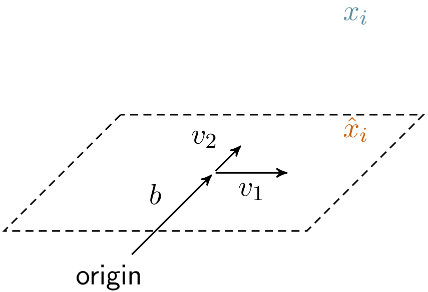
Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



Decoding: Geometric interpretation

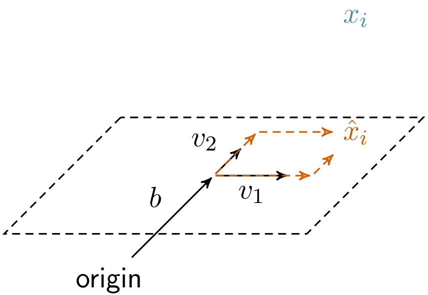
- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



Decoding: Geometric interpretation

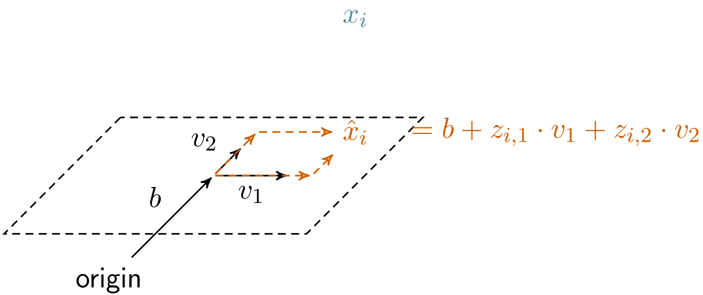
UNIVERSITÄT BIELEFELD
Faculty of Technology

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



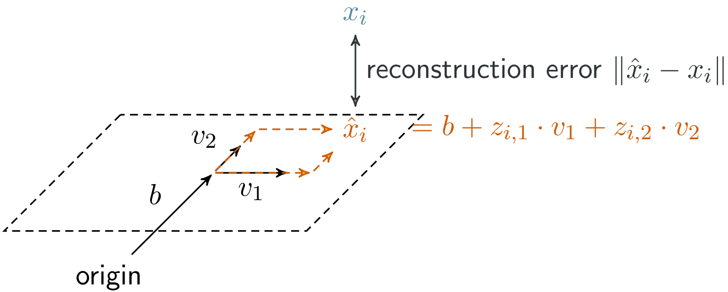
Decoding: Geometric interpretation

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



Decoding: Geometric interpretation

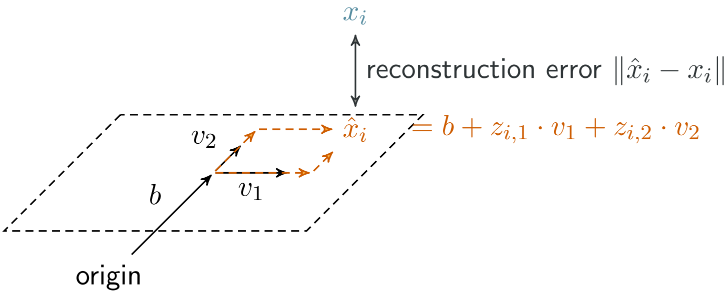
- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



Decoding: Geometric interpretation

UNIVERSITÄT
BIELEFELD
Faculty of Technology

- Assume $m = 3$ and $n = 2$, and let v_1, v_2 be the columns of $\boldsymbol{V}$.



In summary:

- Columns of $\boldsymbol{V}$ span a hyperplane that contains all possible decoded points
- Without loss of generality, we can assume $\boldsymbol{V}$ to be (semi-)orthogonal
- … which means $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ (but $\boldsymbol{V}\boldsymbol{V}^T \neq \boldsymbol{I}$!)

Deriving a

- Let's inspect the equation for $\hat{x}_i$:

$$\hat{x}_i = \boldsymbol{V} z_i + b$$

Deriving a

- Let's inspect the equation for $\hat{x}_i$:

$$\hat{x}_i = \boldsymbol{V} z_i + b = \boldsymbol{V}\boldsymbol{U} x_i + \boldsymbol{V} a + b$$

Deriving a

- Let's inspect the equation for $\hat{x}_i$:

$$\hat{x}_i = \boldsymbol{V} z_i + b = \boldsymbol{V}\boldsymbol{U} x_i + \boldsymbol{V} a + b$$

$\Rightarrow$ We can set a however we want, because b can correct for it

Deriving a

- Let's inspect the equation for $\hat{x}_i$:

$$\hat{x}_i = \boldsymbol{V} z_i + b = \boldsymbol{V}\boldsymbol{U} x_i + \boldsymbol{V} a + b$$

$\Rightarrow$ We can set a however we want, because b can correct for it

$\Rightarrow$ Without loss of generality, set $a = -\boldsymbol{U}\mu$, where μ is the mean: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$

Deriving the optimal b

- Let's optimize reconstruction error w.r.t. b

UNIVERSITÄT BIELEFELD
Faculty of Technology

Deriving the optimal b

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

Deriving the optimal b

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\ell(b) = \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2$$

Deriving the optimal b

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\ell(b) = \sum_{i=1}^{N} \|\psi\big(\phi(x_i)\big) - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + b - x_i\|^2$$

Deriving the optimal b

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\ell(b) = \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + b - x_i\|^2$$

$$= \sum_{i=1}^{N} \|\boldsymbol{V}(\boldsymbol{U} x_i - \boldsymbol{U}\mu) + b - x_i\|^2$$

Deriving the optimal b

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\ell(b) = \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + b - x_i\|^2$$

$$= \sum_{i=1}^{N} \|\boldsymbol{V}(\boldsymbol{U} x_i - \boldsymbol{U}\mu) + b - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i\|^2$$

Deriving the optimal b

Universität Bielefeld
Faculty of Technology

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\ell(b) = \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + b - x_i\|^2$$

$$= \sum_{i=1}^{N} \|\boldsymbol{V}(\boldsymbol{U} x_i - \boldsymbol{U}\mu) + b - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i\|^2$$

$$\Rightarrow \nabla_b \ell(b) = 2\sum_{i=1}^{N} \boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i$$

Deriving the optimal b

UNIVERSITÄT BIELEFELD

Faculty of Technology

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\ell(b) = \sum_{i=1}^{N} \|\psi\Big(\phi(x_i)\Big) - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + b - x_i\|^2$$

$$= \sum_{i=1}^{N} \|\boldsymbol{V}(\boldsymbol{U} x_i - \boldsymbol{U}\mu) + b - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i\|^2$$

$$\Rightarrow \nabla_b \ell(b) = 2\sum_{i=1}^{N} \boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i$$

$$= 2\boldsymbol{V}\boldsymbol{U}(\sum_{i=1}^{N} x_i - N\mu) + 2Nb - 2\sum_{i=1}^{N} x_i$$

Deriving the optimal b

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$
\begin{aligned}
\ell(b) &= \sum_{i=1}^{N} \|\psi\big(\phi(x_i)\big) - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + b - x_i\|^2 \\
&= \sum_{i=1}^{N} \|\boldsymbol{V}(\boldsymbol{U} x_i - \boldsymbol{U}\mu) + b - x_i\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i\|^2 \\
\Rightarrow \nabla_b \ell(b) &= 2 \sum_{i=1}^{N} \boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i \\
&= 2\boldsymbol{V}\boldsymbol{U}(\sum_{i=1}^{N} x_i - N\mu) + 2Nb - 2\sum_{i=1}^{N} x_i = 2Nb - 2\sum_{i=1}^{N} x_i
\end{aligned}
$$

Deriving the optimal b

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Let's optimize reconstruction error w.r.t. b
- **note!** Reconstruction error is convex $\Rightarrow$ finding zero of gradient is sufficient

$$\begin{aligned}
\ell(b) &= \sum_{i=1}^N \|\psi\Big(\phi(x_i)\Big) - x_i\|^2 = \sum_{i=1}^N \|\boldsymbol{V} z_i + b - x_i\|^2 \\
&= \sum_{i=1}^N \|\boldsymbol{V}(\boldsymbol{U} x_i - \boldsymbol{U}\mu) + b - x_i\|^2 = \sum_{i=1}^N \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i\|^2 \\
\Rightarrow \nabla_b \ell(b) &= 2 \sum_{i=1}^N \boldsymbol{V}\boldsymbol{U}(x_i - \mu) + b - x_i \\
&= 2\boldsymbol{V}\boldsymbol{U}(\sum_{i=1}^N x_i - N\mu) + 2Nb - 2\sum_{i=1}^N x_i = 2Nb - 2\sum_{i=1}^N x_i
\end{aligned}$$

Setting the gradient to zero yields $b = \mu$.

Deriving the optimal encoder

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Now, let's optimize reconstruction error w.r.t. $\boldsymbol{U}$.

Deriving the optimal encoder

- Now, let's optimize reconstruction error w.r.t. $\boldsymbol{U}$.

$$\ell(\boldsymbol{U}) = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - x_i\|^2$$

Deriving the optimal encoder

- Now, let's optimize reconstruction error w.r.t. $\boldsymbol{U}$.

$$
\begin{aligned}
\ell(\boldsymbol{U}) &= \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - x_i\|^2 \\
&= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) - (x_i - \mu)\|^2
\end{aligned}
$$

Deriving the optimal encoder

- Now, let's optimize reconstruction error w.r.t. $\boldsymbol{U}$.

$$
\begin{aligned}
\ell(\boldsymbol{U}) &= \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - x_i\|^2 \\
&= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) - (x_i - \mu)\|^2 \\
&= \sum_{i=1}^{N} (x_i - \mu)^T \boldsymbol{U}^T \boldsymbol{V}^T \boldsymbol{V}\boldsymbol{U}(x_i - \mu) - 2(x_i - \mu)^T \boldsymbol{U}^T \boldsymbol{V}^T (x_i - \mu) + (x_i - \mu)^T (x_i - \mu)
\end{aligned}
$$

Deriving the optimal encoder

- Now, let's optimize reconstruction error w.r.t. $\boldsymbol{U}$.

$$\begin{aligned}
\ell(\boldsymbol{U}) &= \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - x_i\|^2 \\
&= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{U}(x_i - \mu) - (x_i - \mu)\|^2 \\
&= \sum_{i=1}^{N} (x_i - \mu)^T \boldsymbol{U}^T \boldsymbol{V}^T \boldsymbol{V}\boldsymbol{U}(x_i - \mu) - 2(x_i - \mu)^T \boldsymbol{U}^T \boldsymbol{V}^T (x_i - \mu) + (x_i - \mu)^T (x_i - \mu) \\
&= \sum_{i=1}^{N} (x_i - \mu)^T \boldsymbol{U}^T \boldsymbol{U}(x_i - \mu) - 2(x_i - \mu)^T \boldsymbol{U}^T \boldsymbol{V}^T (x_i - \mu) + (x_i - \mu)^T (x_i - \mu)
\end{aligned}$$

Deriving the optimal encoder (continued)

Let's compute the gradient:

$$\nabla_{\boldsymbol{U}} \ell(\boldsymbol{U}) = \sum_{i=1}^{N} 2\boldsymbol{U}(x_i - \mu)(x_i - \mu)^T - 2\boldsymbol{V}^T(x_i - \mu)(x_i - \mu)^T$$

Deriving the optimal encoder (continued)

Let's compute the gradient:

$$
\begin{aligned}
\nabla_{\boldsymbol{U}} \ell(\boldsymbol{U}) &= \sum_{i=1}^{N} 2\boldsymbol{U}(x_i - \mu)(x_i - \mu)^T - 2\boldsymbol{V}^T(x_i - \mu)(x_i - \mu)^T \\
&= 2\Big(\boldsymbol{U} - \boldsymbol{V}^T\Big) \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T
\end{aligned}
$$

Deriving the optimal encoder (continued)

UNIVERSITÄT BIELEFELD
Faculty of Technology

Let's compute the gradient:

$$
\begin{aligned}
\nabla_{\boldsymbol{U}} \ell(\boldsymbol{U}) &= \sum_{i=1}^{N} 2\boldsymbol{U}(x_i - \mu)(x_i - \mu)^T - 2\boldsymbol{V}^T(x_i - \mu)(x_i - \mu)^T \\
&= 2\left(\boldsymbol{U} - \boldsymbol{V}^T\right) \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \\
&= 2\left(\boldsymbol{U} - \boldsymbol{V}^T\right) \boldsymbol{C}
\end{aligned}
$$

Deriving the optimal encoder (continued)

Let's compute the gradient:

$$\begin{aligned}
\nabla_{\boldsymbol{U}} \ell(\boldsymbol{U}) &= \sum_{i=1}^{N} 2\boldsymbol{U}(x_i - \mu)(x_i - \mu)^T - 2\boldsymbol{V}^T(x_i - \mu)(x_i - \mu)^T \\
&= 2\left(\boldsymbol{U} - \boldsymbol{V}^T\right) \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \\
&= 2\left(\boldsymbol{U} - \boldsymbol{V}^T\right) \boldsymbol{C}
\end{aligned}$$

$\Rightarrow$ Setting gradient to zero yields: $\boldsymbol{U} = \boldsymbol{V}^T$

Deriving the optimal encoder (continued)

UNIVERSITÄT BIELEFELD
Faculty of Technology

Let's compute the gradient:

$$\begin{aligned}
\nabla_{\boldsymbol{U}} \ell(\boldsymbol{U}) &= \sum_{i=1}^{N} 2\boldsymbol{U}(x_i - \mu)(x_i - \mu)^T - 2\boldsymbol{V}^T(x_i - \mu)(x_i - \mu)^T \\
&= 2\left(\boldsymbol{U} - \boldsymbol{V}^T\right) \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \\
&= 2\left(\boldsymbol{U} - \boldsymbol{V}^T\right) \boldsymbol{C}
\end{aligned}$$

$\Rightarrow$ Setting gradient to zero yields: $\boldsymbol{U} = \boldsymbol{V}^T$

- **attention!** The last step is only valid because $\boldsymbol{C}$ is positive (semi-)definite and, hence, invertible

Summary thus far

- We set $a = -\boldsymbol{U}\mu$

Summary thus far

- We set $a = -\boldsymbol{U}\mu$

$\Rightarrow$ Therefore, $b = \mu$

Summary thus far

- We set $a = -\boldsymbol{U}\mu$

$\Rightarrow$ Therefore, $b = \mu$

$\Rightarrow$ Therefore (and because $\boldsymbol{V}$ is orthogonal), $\boldsymbol{U} = \boldsymbol{V}^T$

Summary thus far

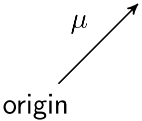
- We set $a = -\boldsymbol{U}\mu$

$\Rightarrow$ Therefore, $b = \mu$

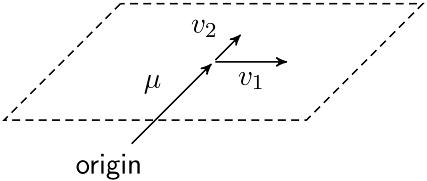
$\Rightarrow$ Therefore (and because $\boldsymbol{V}$ is orthogonal), $\boldsymbol{U} = \boldsymbol{V}^T$

$\Rightarrow$ Only $\boldsymbol{V}$ remains to be optimized – for which we need some geometry, again

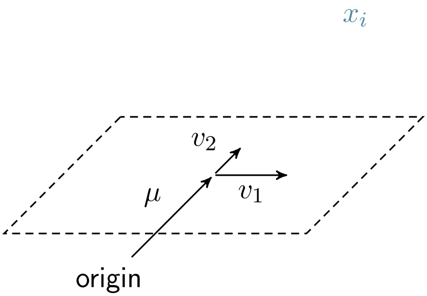
Geometric interpretation (continued)



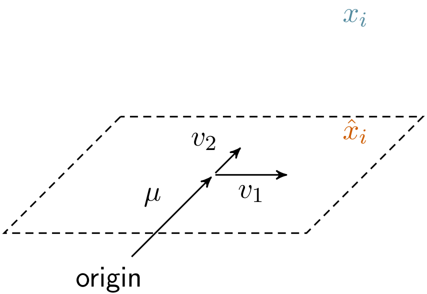
Geometric interpretation (continued)



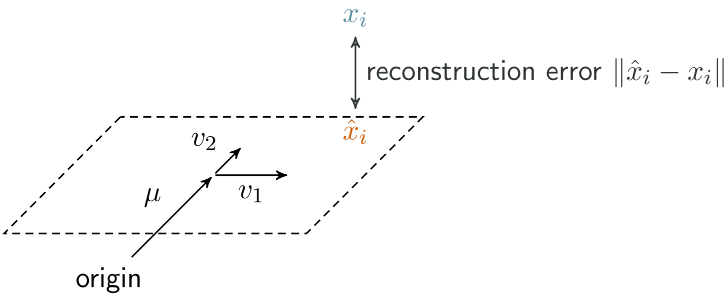
Geometric interpretation (continued)



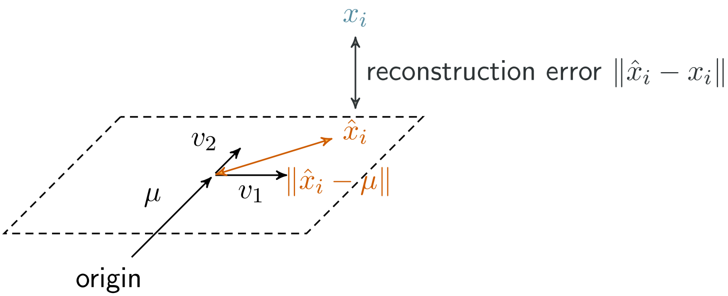
Geometric interpretation (continued)



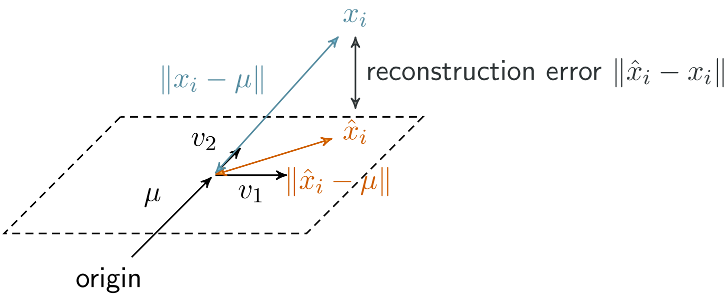
Geometric interpretation (continued)



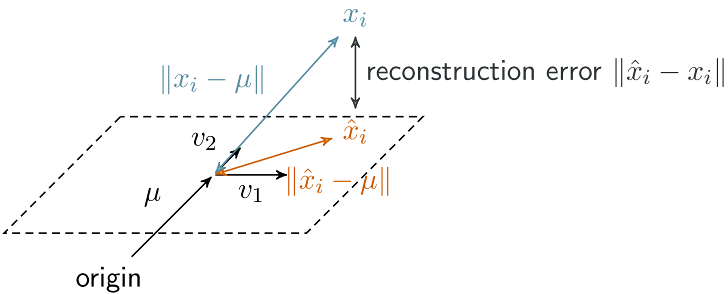
Geometric interpretation (continued)



Geometric interpretation (continued)

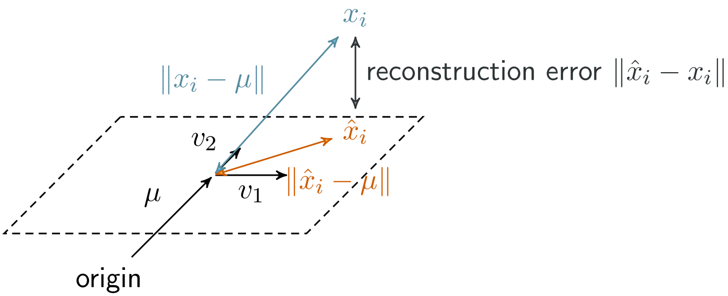


Geometric interpretation (continued)



- By construction: $\hat{x}_i$ is projection of x_i onto the hyperplane spanned by the columns of $\boldsymbol{V}$ (and anchored in μ)

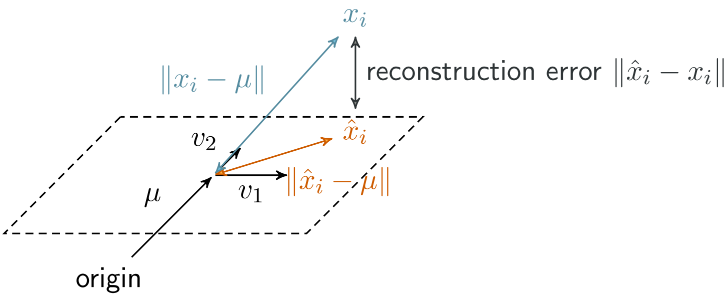
Geometric interpretation (continued)



- By construction: $\hat{x}_i$ is projection of x_i onto the hyperplane spanned by the columns of $\boldsymbol{V}$ (and anchored in μ)

$\Rightarrow$ The sides $\|\hat{x}_i - x_i\|$, $\|\hat{x}_i - \mu\|$, and $\|x_i - \mu\|$ form a right triangle

Geometric interpretation (continued)



- By construction: $\hat{x}_i$ is projection of x_i onto the hyperplane spanned by the columns of $\boldsymbol{V}$ (and anchored in μ)

$\Rightarrow$ The sides $\|\hat{x}_i - x_i\|$, $\|\hat{x}_i - \mu\|$, and $\|x_i - \mu\|$ form a right triangle

$\Rightarrow$ Pythagoras: $\|\hat{x}_i - x_i\|^2 + \|\hat{x}_i - \mu\|^2 = \|x_i - \mu\|^2$

New optimization target

$$\min_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2$$

New optimization target

$$\min_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2$$

$$\Leftrightarrow \min_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \quad \overbrace{\|x_i - \mu\|^2}^{\text{does not depend on } \boldsymbol{V}} \quad - \|\hat{x}_i - \mu\|^2$$

New optimization target

$$
\begin{aligned}
&\min_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2 \\
\Leftrightarrow &\min_{\boldsymbol{V}} \quad \sum_{i=1}^{N} \overbrace{\|x_i - \mu\|^2}^{\text{does not depend on } \boldsymbol{V}} - \|\hat{x}_i - \mu\|^2 \\
\Leftrightarrow &\min_{\boldsymbol{V}} \quad -\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2
\end{aligned}
$$

New optimization target

$$\min_{\boldsymbol{V}} \sum_{i=1}^{N} \|\hat{x}_i - x_i\|^2$$

$$\Leftrightarrow \min_{\boldsymbol{V}} \sum_{i=1}^{N} \overbrace{\|x_i - \mu\|^2}^{\text{does not depend on } \boldsymbol{V}} - \|\hat{x}_i - \mu\|^2$$

$$\Leftrightarrow \min_{\boldsymbol{V}} - \sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2$$

$$\Leftrightarrow \max_{\boldsymbol{V}} \sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2$$

Deriving the optimal $\boldsymbol{V}$

$$\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - \mu\|^2$$

Deriving the optimal $\boldsymbol{V}$

Universität Bielefeld
Faculty of Technology

$$
\begin{aligned}
\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 &= \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - \mu\|^2 \\
&= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{V}^T (x_i - \mu)\|^2
\end{aligned}
$$

Deriving the optimal $\boldsymbol{V}$

$$\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - \mu\|^2$$
$$= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{V}^T (x_i - \mu)\|^2$$

- Note: because $\boldsymbol{V}$ is orthogonal, it preserves the norm of vectors.

Deriving the optimal $\boldsymbol{V}$

$$\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - \mu\|^2$$
$$= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{V}^T (x_i - \mu)\|^2$$

- Note: because $\boldsymbol{V}$ is orthogonal, it preserves the norm of vectors.

$$= \sum_{i=1}^{N} \|\boldsymbol{V}^T (x_i - \mu)\|^2$$

Deriving the optimal $\boldsymbol{V}$

$$\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - \mu\|^2$$
$$= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{V}^T (x_i - \mu)\|^2$$

- Note: because $\boldsymbol{V}$ is orthogonal, it preserves the norm of vectors.

$$= \sum_{i=1}^{N} \|\boldsymbol{V}^T (x_i - \mu)\|^2$$

- Trick: To achieve orthogonal $\boldsymbol{V}$, optimize one column at a time

Deriving the optimal $\boldsymbol{V}$

$$\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \sum_{i=1}^{N} \|\boldsymbol{V} z_i + \mu - \mu\|^2$$
$$= \sum_{i=1}^{N} \|\boldsymbol{V}\boldsymbol{V}^T(x_i - \mu)\|^2$$

- Note: because $\boldsymbol{V}$ is orthogonal, it preserves the norm of vectors.

$$= \sum_{i=1}^{N} \|\boldsymbol{V}^T(x_i - \mu)\|^2$$

- Trick: To achieve orthogonal $\boldsymbol{V}$, optimize one column at a time; search for next column in orthogonal subspace, etc.

Deriving the first principal component

UNIVERSITÄT BIELEFELD
Faculty of Technology

Recall: We need to set v to maximize

$$\sum_{i=1}^{N} \|v^T(x_i - \mu)\|^2$$

Deriving the first principal component

Recall: We need to set v to maximize

$$\sum_{i=1}^{N} \|v^T(x_i - \mu)\|^2 = \sum_{i=1}^{N} \left(v^T(x_i - \mu)\right)^2$$

Deriving the first principal component

Recall: We need to set v to maximize

$$\sum_{i=1}^{N} \|v^T(x_i - \mu)\|^2 = \sum_{i=1}^{N} \left(v^T(x_i - \mu)\right)^2$$

$$= \sum_{i=1}^{N} v^T(x_i - \mu) \cdot (x_i - \mu)^T \cdot v$$

Deriving the first principal component

Recall: We need to set v to maximize

$$
\begin{aligned}
\sum_{i=1}^{N} \|v^T(x_i - \mu)\|^2 &= \sum_{i=1}^{N} \Big(v^T(x_i - \mu) \Big)^2 \\
&= \sum_{i=1}^{N} v^T(x_i - \mu) \cdot (x_i - \mu)^T \cdot v \\
&= v^T \Big(\sum_{i=1}^{N} (x_i - \mu) \cdot (x_i - \mu)^T \Big) \cdot v
\end{aligned}
$$

Deriving the first principal component

Recall: We need to set v to maximize

$$\begin{aligned}
\sum_{i=1}^{N} \|v^T(x_i - \mu)\|^2 &= \sum_{i=1}^{N} \Big(v^T(x_i - \mu)\Big)^2 \\
&= \sum_{i=1}^{N} v^T(x_i - \mu) \cdot (x_i - \mu)^T \cdot v \\
&= v^T \Big(\sum_{i=1}^{N} (x_i - \mu) \cdot (x_i - \mu)^T \Big) \cdot v \\
&= v^T \boldsymbol{C} v
\end{aligned}$$

Deriving the first principal component

Recall: We need to set v to maximize

$$\begin{aligned}
\sum_{i=1}^{N} \|v^T (x_i - \mu)\|^2 &= \sum_{i=1}^{N} \Big(v^T (x_i - \mu)\Big)^2 \\
&= \sum_{i=1}^{N} v^T (x_i - \mu) \cdot (x_i - \mu)^T \cdot v \\
&= v^T \Big(\sum_{i=1}^{N} (x_i - \mu) \cdot (x_i - \mu)^T \Big) \cdot v \\
&= v^T \boldsymbol{C} v
\end{aligned}$$

- Note: we also want $\|v\| = 1$ such that $\boldsymbol{V}$ is orthogonal

Deriving the first principal component

UNIVERSITÄT BIELEFELD
Faculty of Technology

Recall: We need to set v to maximize

$$
\begin{aligned}
\sum_{i=1}^{N} \|v^T(x_i - \mu)\|^2 &= \sum_{i=1}^{N} \left(v^T(x_i - \mu)\right)^2 \\
&= \sum_{i=1}^{N} v^T(x_i - \mu) \cdot (x_i - \mu)^T \cdot v \\
&= v^T \left(\sum_{i=1}^{N} (x_i - \mu) \cdot (x_i - \mu)^T \right) \cdot v \\
&= v^T \boldsymbol{C} v
\end{aligned}
$$

- Note: we also want $\|v\| = 1$ such that $\boldsymbol{V}$ is orthogonal

$\Rightarrow$ Lagrangian is given as $\ell(v, \lambda) = -v^T \boldsymbol{C} v - \lambda \cdot (1 - v^T v)$

Deriving the first principal component (continued)

UNIVERSITÄT BIELEFELD
Faculty of Technology

$$\nabla_v \ell(v, \lambda) = -2\boldsymbol{C}v + 2\lambda v$$

Deriving the first principal component (continued)

$$\nabla_v \ell(v, \lambda) = -2\boldsymbol{C}v + 2\lambda v$$

Setting the gradient to zero yields:

$$\boldsymbol{C}v = \lambda v$$

Deriving the first principal component (continued)

UNIVERSITÄT BIELEFELD
Faculty of Technology

$$\nabla_v \ell(v, \lambda) = -2\boldsymbol{C}v + 2\lambda v$$

Setting the gradient to zero yields:

$$\boldsymbol{C}v = \lambda v$$

$\Rightarrow$ v needs to be an eigenvector of $\boldsymbol{C}$! Lagrangian multiplier λ is the corresponding eigenvalue

Deriving the first principal component (continued)

$$\nabla_v \ell(v, \lambda) = -2\boldsymbol{C}v + 2\lambda v$$

Setting the gradient to zero yields:

$$\boldsymbol{C}v = \lambda v$$

$\Rightarrow$ v needs to be an eigenvector of $\boldsymbol{C}$! Lagrangian multiplier λ is the corresponding eigenvalue

$\Rightarrow$ objective: $v^T \boldsymbol{C} v = v^T \lambda v = \lambda$

Deriving the first principal component (continued)

UNIVERSITÄT BIELEFELD
Faculty of Technology

$$\nabla_v \ell(v, \lambda) = -2\boldsymbol{C}v + 2\lambda v$$

Setting the gradient to zero yields:

$$\boldsymbol{C}v = \lambda v$$

$\Rightarrow$ v needs to be an eigenvector of $\boldsymbol{C}$! Lagrangian multiplier λ is the corresponding eigenvalue

$\Rightarrow$ objective: $v^T \boldsymbol{C} v = v^T \lambda v = \lambda$

$\Rightarrow$ choose eigenvector v corresponding to largest eigenvalue

Summary: PCA procedure

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

end function

Summary: PCA procedure

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

end function

Summary: PCA procedure

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

end function

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.
Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.
Compute eigenvalue decomposition $\boldsymbol{C} = \boldsymbol{V} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{V}^T$.

end function

Summary: PCA procedure

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.
Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.
Compute eigenvalue decomposition $\boldsymbol{C} = \boldsymbol{V} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{V}^T$.
Keep only the columns of $\boldsymbol{V}$ corresponding to the n largest eigenvalues.

end function

Summary: PCA procedure

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

Compute eigenvalue decomposition $\boldsymbol{C} = \boldsymbol{V} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{V}^T$.

Keep only the columns of $\boldsymbol{V}$ corresponding to the n largest eigenvalues.

return $\phi(x) = \boldsymbol{V}^T \cdot (x - \mu)$ and $\psi(z) = \boldsymbol{V} \cdot z + \mu$.

end function

Summary: PCA procedure

function PCA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)
  Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.
  Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.
  Compute eigenvalue decomposition $\boldsymbol{C} = \boldsymbol{V} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{V}^T$.
  Keep only the columns of $\boldsymbol{V}$ corresponding to the n largest eigenvalues.
  return $\phi(x) = \boldsymbol{V}^T \cdot (x - \mu)$ and $\psi(z) = \boldsymbol{V} \cdot z + \mu$.
end function

Implementation: sklearn.decomposition.PCA

How to choose n

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Recall: Objective becomes variance of the data $\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \lambda$

How to choose n

- Recall: Objective becomes variance of the data $\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \lambda$

$\Rightarrow$ Sum of eigenvalues after PCA quantifies the remaining variance

How to choose n

- Recall: Objective becomes variance of the data $\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \lambda$

$\Rightarrow$ Sum of eigenvalues after PCA quantifies the remaining variance

$\Rightarrow$ Quantify fraction of retained variance as $\sum_{j=1}^{n} \lambda_i / \sum_{j=1}^{m} \lambda_i$

How to choose n

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Recall: Objective becomes variance of the data $\sum_{i=1}^{N} \|\hat{x}_i - \mu\|^2 = \lambda$

$\Rightarrow$ Sum of eigenvalues after PCA quantifies the remaining variance

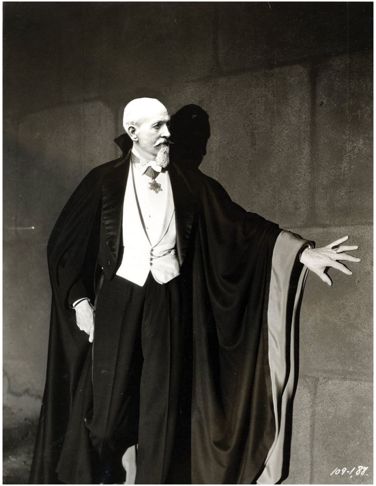
$\Rightarrow$ Quantify fraction of retained variance as $\sum_{j=1}^{n} \lambda_i / \sum_{j=1}^{m} \lambda_i$

$\Rightarrow$ Set n high enough to retain most of the variance (e.g. 95%)

Factor Analysis

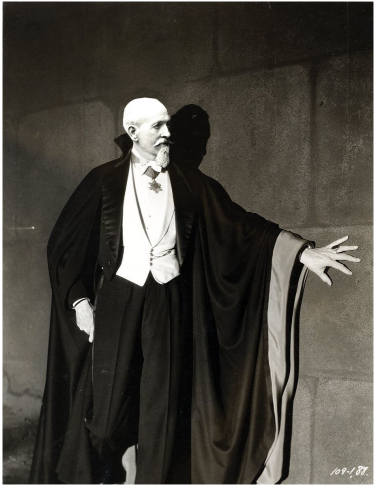
Intro

- IQ tests and Spearman $\rightarrow$ see first lecture



Intro

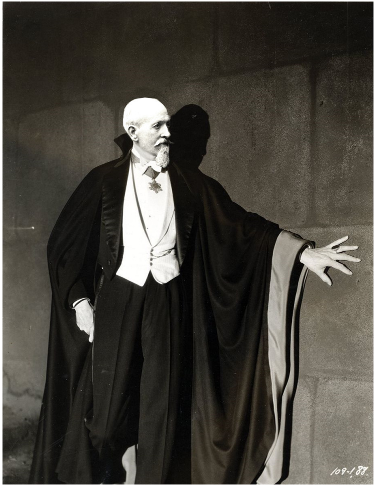
- IQ tests and Spearman → see first lecture
- "Modern" Factor Analysis: Probabilistic version of PCA



Intro

UNIVERSITÄT BIELEFELD
Faculty of Technology

- IQ tests and Spearman → see first lecture
- "Modern" Factor Analysis: Probabilistic version of PCA
- Full derivation bit too complicated for this lecture ⇒ Refer to Barber (2012)



Probabilistic Model

- Assume data is generated as $x = \boldsymbol{V} z + b + \epsilon$

Probabilistic Model

- Assume data is generated as $x = \boldsymbol{V} z + b + \epsilon$
- Assume $p_Z(z)$ is Gaussian with mean 0 and covariance $\boldsymbol{I}$

Probabilistic Model

- Assume data is generated as $x = \boldsymbol{V} z + b + \epsilon$
- Assume $p_Z(z)$ is Gaussian with mean 0 and covariance $\boldsymbol{I}$
- Assume $p_E(\epsilon)$ is Gaussian with mean 0 and covariance $\boldsymbol{\Psi}$

Probabilistic Model

- Assume data is generated as $x = \boldsymbol{V} z + b + \epsilon$
- Assume $p_Z(z)$ is Gaussian with mean 0 and covariance $\boldsymbol{I}$
- Assume $p_E(\epsilon)$ is Gaussian with mean 0 and covariance $\boldsymbol{\Psi}$

$\Rightarrow$ $p_{X|Z}(x|z)$ is Gaussian with mean $\boldsymbol{V} z + b$ and covariance $\boldsymbol{\Psi}$

Probabilistic Model

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Assume data is generated as $x = \boldsymbol{V}z + b + \epsilon$
- Assume $p_Z(z)$ is Gaussian with mean 0 and covariance $\boldsymbol{I}$
- Assume $p_E(\epsilon)$ is Gaussian with mean 0 and covariance $\boldsymbol{\Psi}$

$\Rightarrow$ $p_{X|Z}(x|z)$ is Gaussian with mean $\boldsymbol{V}z + b$ and covariance $\boldsymbol{\Psi}$

$\Rightarrow$ $p_X(x)$ is Gaussian with mean b and covariance $\boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$

Probabilistic Model

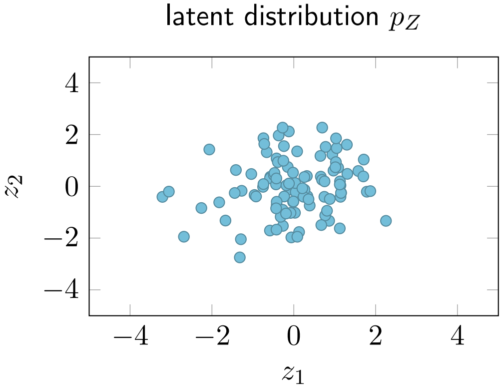
UNIVERSITÄT BIELEFELD
Faculty of Technology

- Assume data is generated as $x = \boldsymbol{V}z + b + \epsilon$
- Assume $p_Z(z)$ is Gaussian with mean 0 and covariance $\boldsymbol{I}$
- Assume $p_E(\epsilon)$ is Gaussian with mean 0 and covariance $\boldsymbol{\Psi}$

$\Rightarrow$ $p_{X|Z}(x|z)$ is Gaussian with mean $\boldsymbol{V}z + b$ and covariance $\boldsymbol{\Psi}$

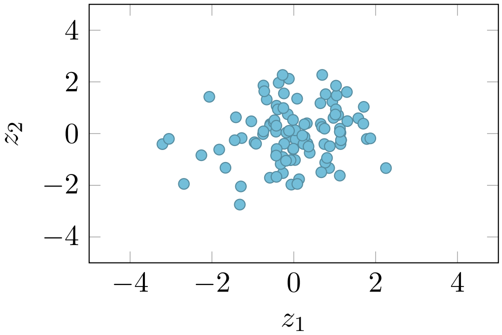
$\Rightarrow$ $p_X(x)$ is Gaussian with mean b and covariance $\boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$ (this is not a trivial result! Follows from theory of Gaussians)

Visualization

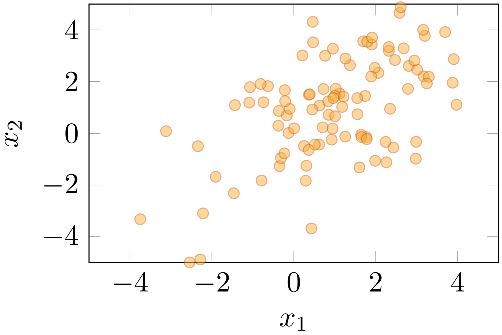


Visualization

latent distribution p_Z

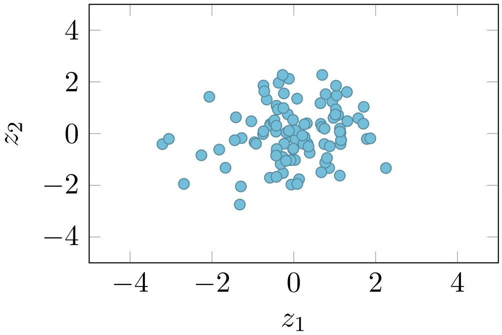


data distribution p_X

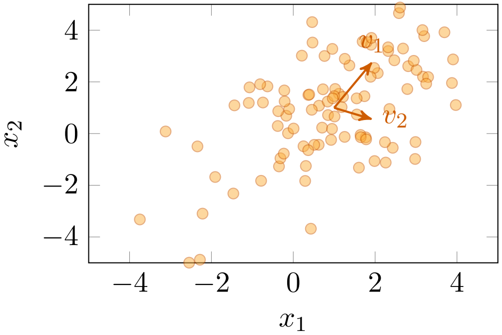


Visualization

latent distribution p_Z

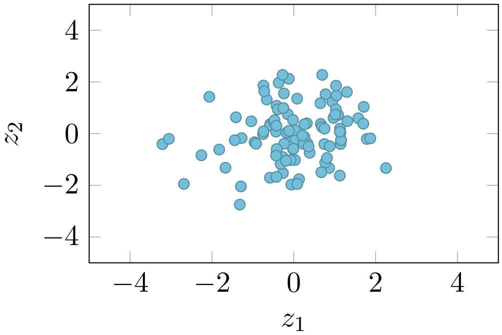


data distribution p_X

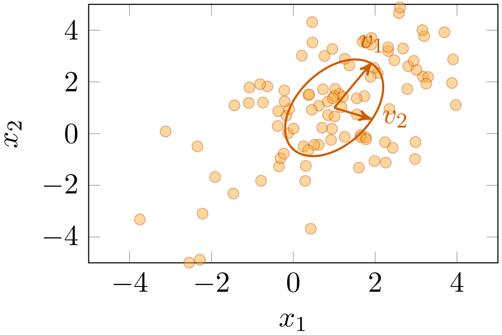


Visualization

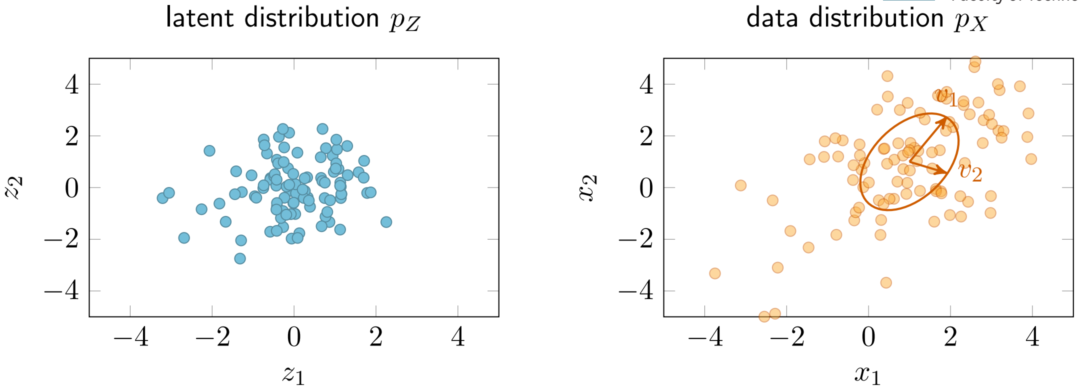
latent distribution p_Z



data distribution p_X

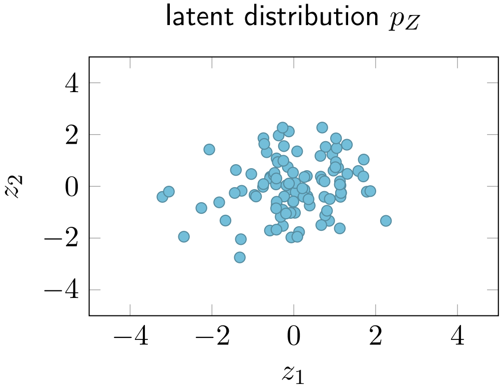


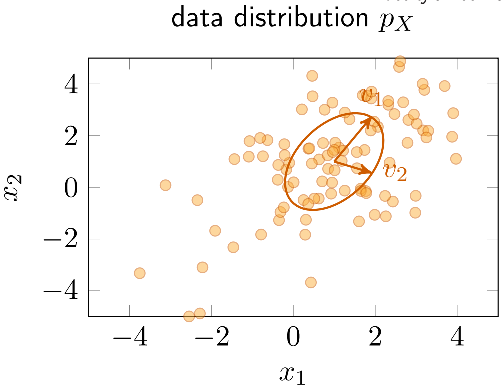
Visualization



- $\boldsymbol{V}$ rotates and stretches data distribution

Visualization





- $\boldsymbol{V}$ rotates and stretches data distribution
- columns of $\boldsymbol{V}$ can be interpreted as principal axes of the hyper-ellipse that forms the isoline of p_X (up to noise)

Maximum likelihood

- Let $\Sigma = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$

Maximum likelihood

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Let $\Sigma = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$

$$\Rightarrow p_X(x) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \cdot \exp\left(-\frac{1}{2}(x-b)^T \Sigma^{-1}(x-b)\right)$$

Maximum likelihood

- Let $\Sigma = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$

$\Rightarrow$ $p_X(x) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \cdot \exp\Big(-\frac{1}{2}(x-b)^T\Sigma^{-1}(x-b)\Big)$

$\Rightarrow$ negative log likelihood of the data:

$$\ell(\boldsymbol{V}, \boldsymbol{\Psi}, b) = \sum_{i=1}^{N} \frac{1}{2} \log\Big[\det(2\pi\boldsymbol{\Sigma})\Big] + \frac{1}{2}(x_i - b)^T\Sigma^{-1}(x_i - b)$$

Maximum likelihood

- Let $\Sigma = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$

$\Rightarrow$ $p_X(x) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \cdot \exp\Big(-\frac{1}{2}(x-b)^T \Sigma^{-1}(x-b)\Big)$

$\Rightarrow$ negative log likelihood of the data:

$$\ell(\boldsymbol{V}, \boldsymbol{\Psi}, b) = \sum_{i=1}^{N} \frac{1}{2} \log\Big[\det(2\pi\boldsymbol{\Sigma})\Big] + \frac{1}{2}(x_i - b)^T \Sigma^{-1}(x_i - b)$$

$\Rightarrow$ Optimal b is $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$.

Maximum likelihood

- Let $\Sigma = \boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{\Psi}$

$\Rightarrow$ $p_X(x) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \cdot \exp\Big(-\frac{1}{2}(x-b)^T\Sigma^{-1}(x-b)\Big)$

$\Rightarrow$ negative log likelihood of the data:

$$\ell(\boldsymbol{V}, \boldsymbol{\Psi}, b) = \sum_{i=1}^{N} \frac{1}{2} \log\Big[\det(2\pi\boldsymbol{\Sigma})\Big] + \frac{1}{2}(x_i - b)^T\Sigma^{-1}(x_i - b)$$

$\Rightarrow$ Optimal b is $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$.

- Optimal $\boldsymbol{V}$ is much harder to determine, requires a few tricks (Barber 2012)

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

end function

Summary: FA procedure

UNIVERSITÄT BIELEFELD
Faculty of Technology

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

end function

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

end function

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

end function

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

for desired number of iterations **do**

end for

end function

Summary: FA procedure

UNIVERSITÄT BIELEFELD
Faculty of Technology

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

  Compute mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$.

  Compute covariance matrix $\boldsymbol{C} = \frac{1}{N}\sum_{i=1}^{m}(x_i - \mu) \cdot (x_i - \mu)^T$.

  Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

  for desired number of iterations **do**

    Compute $\tilde{\boldsymbol{C}} \leftarrow \boldsymbol{\Psi}^{-\frac{1}{2}}\boldsymbol{C}\boldsymbol{\Psi}^{-\frac{1}{2}}$.

  end for

end function

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

  Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

  Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

  Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

  for desired number of iterations **do**

    Compute $\tilde{\boldsymbol{C}} \leftarrow \boldsymbol{\Psi}^{-\frac{1}{2}} \boldsymbol{C} \boldsymbol{\Psi}^{-\frac{1}{2}}$.

    Compute eigenvalue decomposition $\tilde{\boldsymbol{C}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$.

  end for

end function

Summary: FA procedure

UNIVERSITÄT
BIELEFELD
Faculty of Technology

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

for desired number of iterations **do**

Compute $\tilde{\boldsymbol{C}} \leftarrow \boldsymbol{\Psi}^{-\frac{1}{2}} \boldsymbol{C} \boldsymbol{\Psi}^{-\frac{1}{2}}$.

Compute eigenvalue decomposition $\tilde{\boldsymbol{C}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$.

Keep only the n largest eigenvalues in $\boldsymbol{\Lambda}$ and the corresponding columns of $\boldsymbol{U}$.

end for

end function

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

  Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.
  Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.
  Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.
  for desired number of iterations **do**
    Compute $\tilde{\boldsymbol{C}} \leftarrow \boldsymbol{\Psi}^{-\frac{1}{2}} \boldsymbol{C} \boldsymbol{\Psi}^{-\frac{1}{2}}$.
    Compute eigenvalue decomposition $\tilde{\boldsymbol{C}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$.
    Keep only the n largest eigenvalues in $\boldsymbol{\Lambda}$ and the corresponding columns of $\boldsymbol{U}$.
    $\boldsymbol{V} \leftarrow \boldsymbol{\Psi}^{\frac{1}{2}} \boldsymbol{U} \boldsymbol{\Lambda}^{\frac{1}{2}}$.

  end for

end function

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N} \sum_{i=1}^{m} (x_i - \mu) \cdot (x_i - \mu)^T$.

Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

for desired number of iterations **do**

Compute $\tilde{\boldsymbol{C}} \leftarrow \boldsymbol{\Psi}^{-\frac{1}{2}} \boldsymbol{C} \boldsymbol{\Psi}^{-\frac{1}{2}}$.

Compute eigenvalue decomposition $\tilde{\boldsymbol{C}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^T$.

Keep only the n largest eigenvalues in $\boldsymbol{\Lambda}$ and the corresponding columns of $\boldsymbol{U}$.

$\boldsymbol{V} \leftarrow \boldsymbol{\Psi}^{\frac{1}{2}} \boldsymbol{U} \boldsymbol{\Lambda}^{\frac{1}{2}}$.

$\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C}) - \text{diag}(\boldsymbol{V}\boldsymbol{V}^T)$.

end for

end function

Summary: FA procedure

function FA(data matrix $\boldsymbol{X}$ with N rows and m columns, desired latent dimensionality $n \leq m$)

Compute mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$.

Compute covariance matrix $\boldsymbol{C} = \frac{1}{N}\sum_{i=1}^{m}(x_i - \mu)\cdot(x_i - \mu)^T$.

Set initial noise to $\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C})$.

for desired number of iterations **do**

Compute $\tilde{\boldsymbol{C}} \leftarrow \boldsymbol{\Psi}^{-\frac{1}{2}}\boldsymbol{C}\boldsymbol{\Psi}^{-\frac{1}{2}}$.

Compute eigenvalue decomposition $\tilde{\boldsymbol{C}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$.

Keep only the n largest eigenvalues in $\boldsymbol{\Lambda}$ and the corresponding columns of $\boldsymbol{U}$.

$\boldsymbol{V} \leftarrow \boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}$.

$\boldsymbol{\Psi} \leftarrow \text{diag}(\boldsymbol{C}) - \text{diag}(\boldsymbol{V}\boldsymbol{V}^T)$.

end for

return $\boldsymbol{V}$, μ, $\boldsymbol{\Psi}$.

end function

Notes on factor analysis

UNIVERSITÄT BIELEFELD
Faculty of Technology

- Implementation: sklearn.decomposition.FactorAnalysis

Notes on factor analysis

- Implementation: sklearn.decomposition.FactorAnalysis
- For $\mathbf{\Psi} = \mathbf{0}$ (i.e.: no noise), model becomes equivalent to PCA

Notes on factor analysis

UNIVERSITÄT BIELEFELD

Faculty of Technology

- Implementation: sklearn.decomposition.FactorAnalysis
- For $\boldsymbol{\Psi} = \mathbf{0}$ (i.e.: no noise), model becomes equivalent to PCA
- But: $\boldsymbol{V}$ is not normalized

Notes on factor analysis

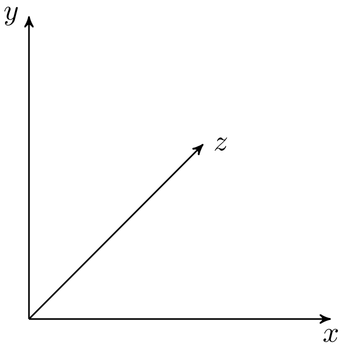
- Implementation: sklearn.decomposition.FactorAnalysis
- For $\mathbf{\Psi} = \mathbf{0}$ (i.e.: no noise), model becomes equivalent to PCA
- But: $\boldsymbol{V}$ is not normalized
- Note: Encoding is **not** the focus of FA (it still works, though)

UNIVERSITÄT BIELEFELD

Faculty of Technology

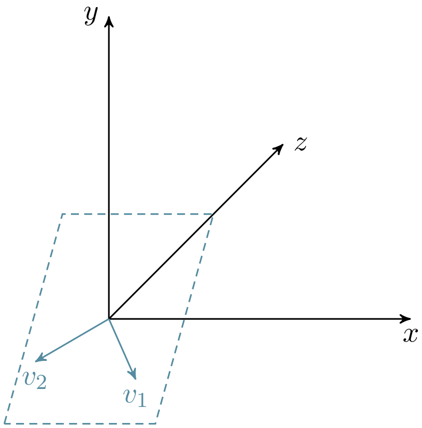
Factor Rotations

Factor Rotation Example

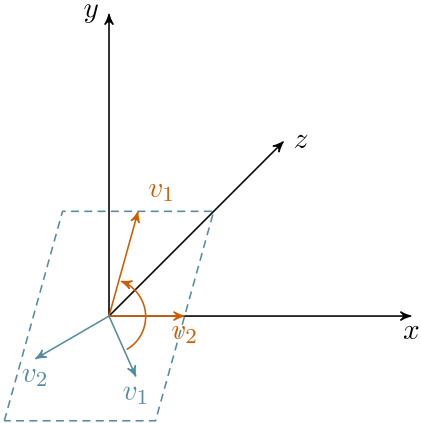


Factor Rotation Example

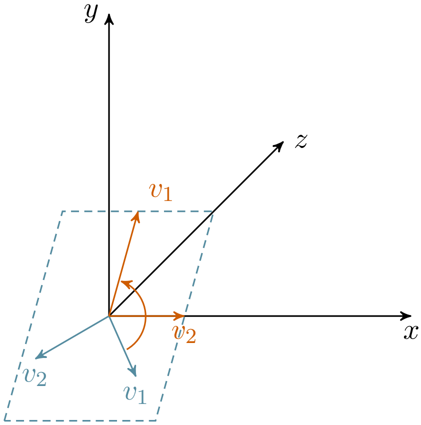
UNIVERSITÄT BIELEFELD
Faculty of Technology



Factor Rotation Example

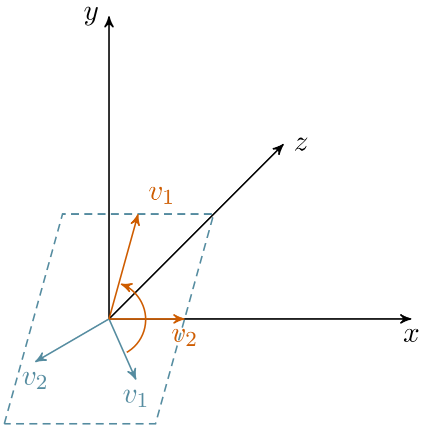


Factor Rotation Example



If $\boldsymbol{R}$ is a rotation matrix,
using $\boldsymbol{V}\boldsymbol{R}$ instead of $\boldsymbol{V}$ has no effect

Factor Rotation Example



If $\boldsymbol{R}$ is a rotation matrix,
using $\boldsymbol{V}\boldsymbol{R}$ instead of $\boldsymbol{V}$ has no effect

because $\boldsymbol{V}\boldsymbol{R}(\boldsymbol{V}\boldsymbol{R})^T = \boldsymbol{V}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{V}^T$

Varimax rotation

- Choose the rotation $\boldsymbol{R}$ that makes the factors "easiest to interpret"

Varimax rotation

- Choose the rotation $\boldsymbol{R}$ that makes the factors "easiest to interpret"
- Maximize variance in latent coordinates $\max_{\boldsymbol{R}} \sum_{i=1}^{N} \sum_{j=1}^{n} z_{i,j}^2$

Varimax rotation

- Choose the rotation $\boldsymbol{R}$ that makes the factors "easiest to interpret"
- Maximize variance in latent coordinates $\max_{\boldsymbol{R}} \sum_{i=1}^{N} \sum_{j=1}^{n} z_{i,j}^2$
- Nonlinear optmization, not discussed here, but implemented in Implementation: sklearn.decomposition.FactorAnalysis

Summary

UNIVERSITÄT BIELEFELD
Faculty of Technology

- PCA is essentially eigenvalue decomposition of covariance matrix

Summary

- PCA is essentially eigenvalue decomposition of covariance matrix
- highly useful for efficiently discovering underlying factors and dimensionality reduction

Summary

UNIVERSITÄT BIELEFELD
Faculty of Technology

- PCA is essentially eigenvalue decomposition of covariance matrix
- highly useful for efficiently discovering underlying factors and dimensionality reduction
- Selection of n: percentage of variance covered

Summary

- PCA is essentially eigenvalue decomposition of covariance matrix
- highly useful for efficiently discovering underlying factors and dimensionality reduction
- Selection of n: percentage of variance covered
- factor analysis is more robust to noise but needs more iterations

Summary

UNIVERSITÄT BIELEFELD
Faculty of Technology

- PCA is essentially eigenvalue decomposition of covariance matrix
- highly useful for efficiently discovering underlying factors and dimensionality reduction
- Selection of n: percentage of variance covered
- factor analysis is more robust to noise but needs more iterations
- Interpretability can be enhanced with factor rotations

Literature I

Barber, David (2012). **Bayesian Reasoning and Machine Learning**. Cambridge, UK: Cambridge University Press. url: http://www.cs.ucl.ac.uk/staff/d.barber/brml/.

Ren, Mengye and Matthew MacKay (2019). **CSC 411 Lecture 12: Principal Component Analysis**. url: https://www.cs.toronto.edu/~mren/teach/csc411_19s/lec/lec12_matt.pdf.