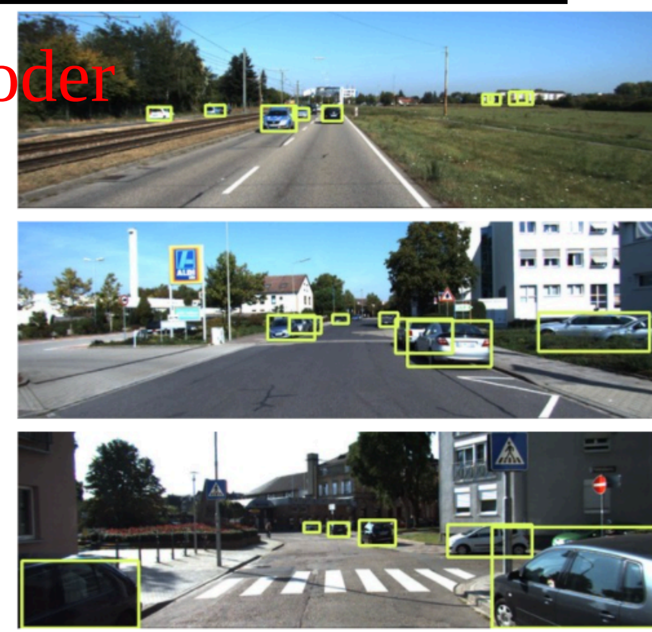
Dimensionality Reduction with Principal Component Analysis

Liang Zheng

Australian National University

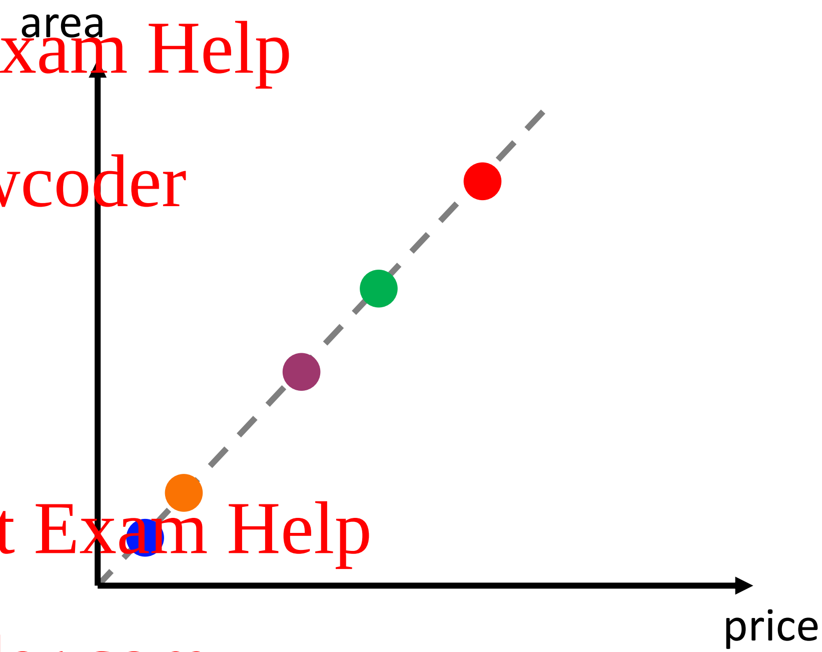
liang.zheng@anu.edu.au

Meta Sim: Learning to Generate Synthetic Datasets. Kar et al., ICCV 2019



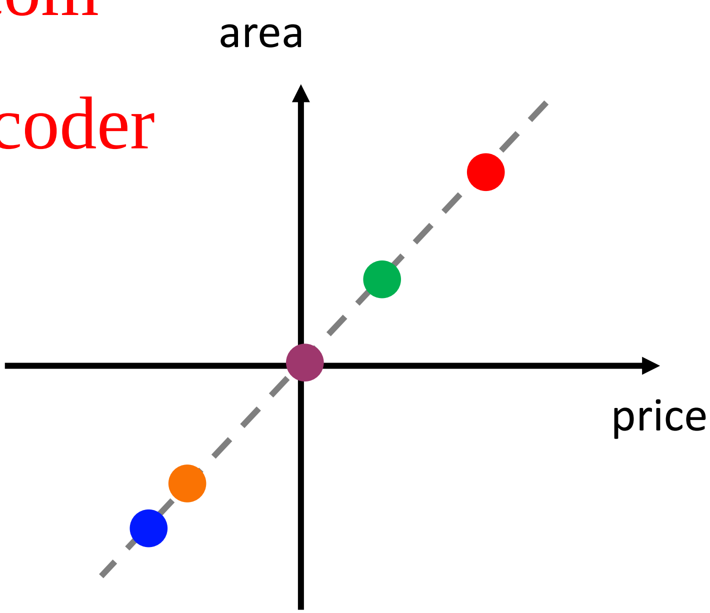
Idea of PCA

	House price (million)	House area ($100m^2$)
a	10	10
b	2	2
c	7	7
d	1	1
e	5	5

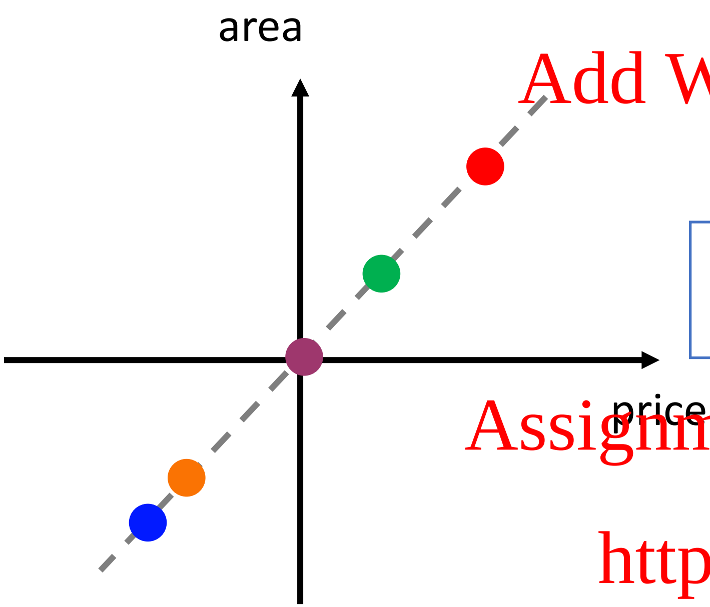


We subtract means from data points

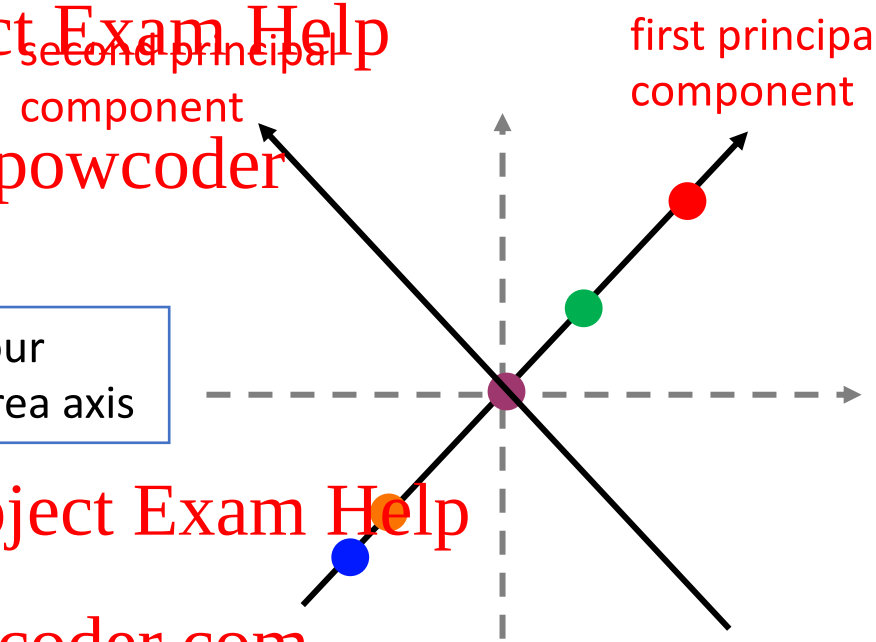
	House price (normalised)	House area (normalised)
a	5	5
b	-3	-3
c	2	2
d	-4	-4
e	0	0



Idea of PCA



We rotate our price and area axis

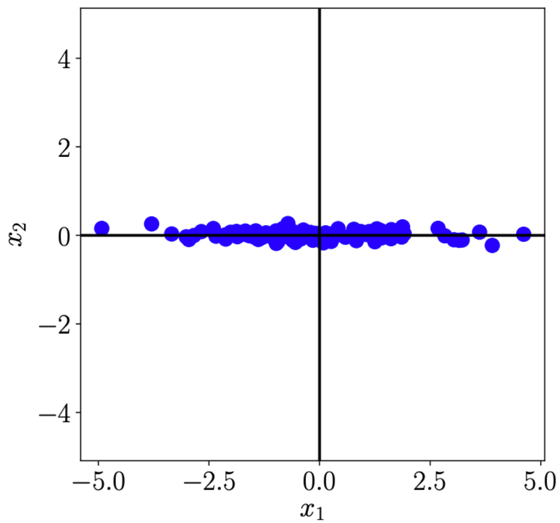


	House price (normalised)	House area (normalised)
a	5	5
b	-3	-3
c	2	2
d	-4	-4
e	0	0

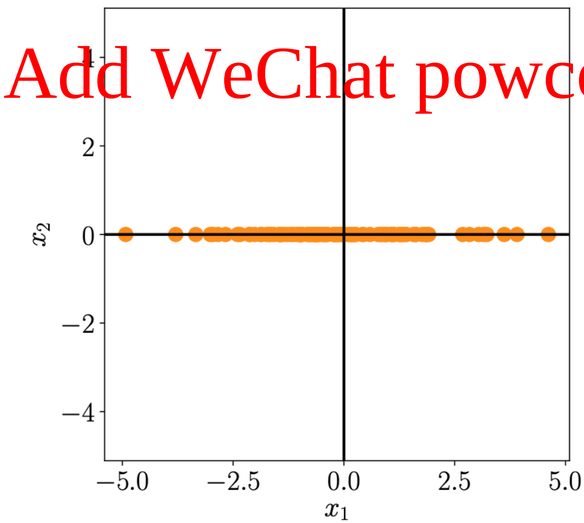
	First principal component	Second principal component
a	7.07	0
b	-4.24	0
c	2.82	0
d	-5.66	0
e	0	0

Motivation

- High-dimensional data, such as images, is hard to analyze, interpretate, and visualize, and expensive to store.
- Good news
- high-dimensional data is often overcomplete, i.e., many dimensions are redundant and can be explained by a combination of other dimensions
- Furthermore, dimensions in high-dimensional data are often correlated so that the data possesses an intrinsic lower-dimensional structure.



(a) Dataset with x_1 and x_2 coordinates.



(b) Compressed dataset where only the x_1 coordinate is relevant.

The data in (a) does not vary much in the x_2-direction, so that we can express it as if it were on a line – with nearly no loss; see (b).

To describe the data in (b), only the x_2 -coordinate is required, and the data lies in a one-dimensional subspace of $\mathbb{R}^2$

10.1 Problem Setting

- In PCA, we are interested in finding projections $\tilde{\boldsymbol{x}}_n$ of data points $\boldsymbol{x}_n$ that are as similar to the original data points as possible, but which have a significantly lower intrinsic dimensionality

- We consider an i.i.d. dataset $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}, \boldsymbol{x}_n \in \mathbb{R}^D$, with mean $\mathbf{0}$ that possesses the data covariance matrix

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{T}}$$

- We assume there exists a low-dimensional compressed representation (code)

$$\boldsymbol{z}_n = \boldsymbol{B}^{\mathrm{T}} \boldsymbol{x}_n \in \mathbb{R}^M$$

of $\boldsymbol{x}_n$, where we define the projection matrix

$$\boldsymbol{B} := [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$$

- **Example (Coordinate Representation/Code)**

- Consider $\mathbb{R}^2$ with the canonical basis $\boldsymbol{e}_1 = [1,0]^{\mathrm{T}}$, $\boldsymbol{e}_2 = [0,1]^{\mathrm{T}}$.
- $\boldsymbol{x} \in \mathbb{R}^2$ can be represented as a linear combination of these basis vectors, e.g.,

$$\begin{bmatrix} 5 \\ 3 \end{bmatrix} = 5\boldsymbol{e}_1 + 3\boldsymbol{e}_2$$

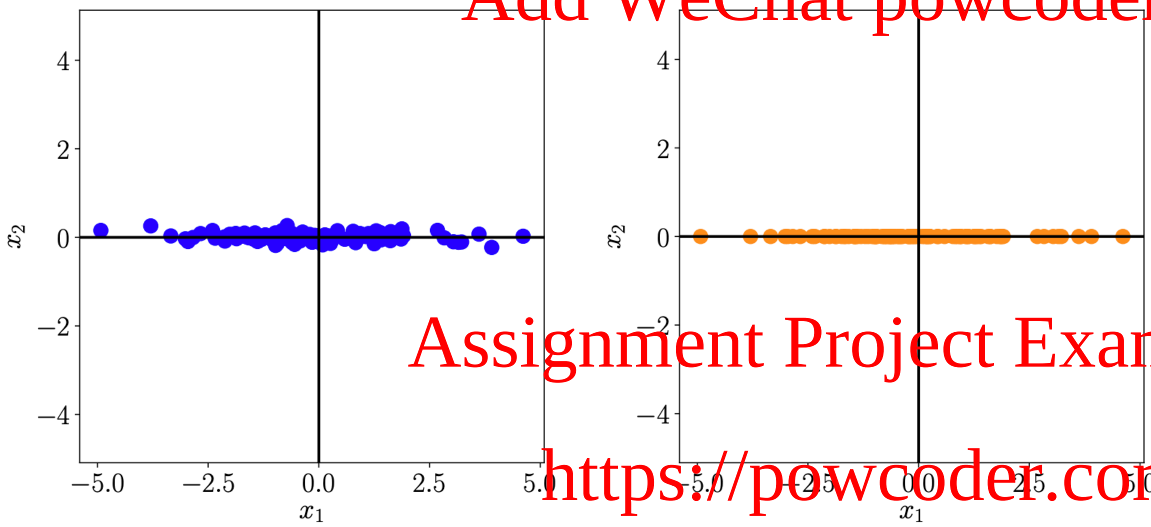
- However, when we consider vectors of the form

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 0 \\ z \end{bmatrix} \in \mathbb{R}^2, \qquad z \in \mathbb{R}$$

they can always be written as $0\boldsymbol{e}_1 + z\boldsymbol{e}_2$.

- To represent these vectors it is sufficient to store the coordinate/code z of $\tilde{\boldsymbol{x}}$ with respect to the $\boldsymbol{e}_2$ vector.

10.2 PCA from Maximum Variance Perspective



(a) Dataset with x_1 and x_2 coordinates. (b) Compressed dataset where only the x_1 coordinate is relevant.

- We ignore x_2 -coordinate of the data because it did not add too much information: the compressed data (b) is similar to the original data in (a)
- We derive PCA so as to maximize the variance in the low-dimensional representation of the data to retain as much information as possible
- Retaining most information after data compression is equivalent to capturing the largest amount of variance in the low-dimensional code (Hotelling, 1933)

10.2.1 Direction with Maximal Variance

- Data centering
- In the data covariance matrix, we assume centered data.

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{T}}$$

- Let us assume that $\boldsymbol{\mu}$ is the mean of the data. Using the properties of the variance, we obtain

$$\mathbb{V}_{\boldsymbol{z}}[\boldsymbol{z}] = \mathbb{V}_{\boldsymbol{x}}[\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{\mu})] = \mathbb{V}_{\boldsymbol{x}}[\boldsymbol{B}^{\mathrm{T}}\boldsymbol{x} - \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\mu}] = \mathbb{V}_{\boldsymbol{x}}[\boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}]$$

- That is, the variance of the low-dimensional code does not depend on the mean of the data.
- With this assumption the mean of the low-dimensional code is also $\mathbf{0}$ since

$$\mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}] = \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}] = \boldsymbol{B}^{\mathrm{T}}\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}] = \mathbf{0}$$

- To maximize the variance of the low-dimensional code, we first seek a single vector $\boldsymbol{b}_1 \in \mathbb{R}^D$ that maximizes the variance of the projected data, i.e., we aim to maximize the variance of the first coordinate z_1 of $\boldsymbol{z} \in \mathbb{R}^M$ so that

$$V_1 := \mathbb{V}[z_1] = \frac{1}{N} \sum_{n=1}^{N} z_{1n}^2$$

is maximized, where we defined z_{1n} as the first coordinate of the low-dimensional representation $\boldsymbol{z}_n \in \mathbb{R}^M$ of $\boldsymbol{x}_n \in \mathbb{R}^D$. z_{1n} is given by,

$$z_{1n} = \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{x}_n$$

i.e., it is the coordinate of the orthogonal projection of $\boldsymbol{x}_n$ onto the one-dimensional subspace spanned by $\boldsymbol{b}_1$. We substitute z_{1n} into V_1 and obtain,

$$V_1 = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{x}_n)^2 = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{b}_1$$

$$= \boldsymbol{b}_1^{\mathrm{T}} \left(\frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{T}} \right) \boldsymbol{b}_1 = \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{S} \boldsymbol{b}_1$$

where $\boldsymbol{S}$ is the data covariance matrix.

- We further restrict all solutions to $\|\boldsymbol{b}_1\|^2 = 1$

- We have the following constrained optimization problem

$$\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{S} \boldsymbol{b}_1$$

$$\text{subject to } \|\boldsymbol{b}_1\|^2 = 1$$

- We obtain the Lagrangian (not required in this course),

$$\mathfrak{L}(\boldsymbol{b}_1, \lambda) = \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{S} \boldsymbol{b}_1 + \lambda_1 \left(1 - \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_1\right)$$

- The partial derivatives of $\mathfrak{L}$ with respect to $\boldsymbol{b}_1$ and λ_1 are

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{b}_1} = 2\boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{S} - 2\lambda_1 \boldsymbol{b}_1^{\mathrm{T}}, \qquad \frac{\partial \mathfrak{L}}{\partial \lambda_1} = 1 - \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_1$$

- Setting these partial derivatives to 0 gives us the relations

$$\boldsymbol{S} \boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1$$

$$\boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_1 = 1$$

- We see that $\boldsymbol{b}_1$ is an eigenvector of $\boldsymbol{S}$, and λ_1 is the corresponding eigenvalue. We rewrite our objective as,

$$V_1 = \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{S} \boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_1 = \lambda_1$$

- i.e., the variance of the data projected onto a one-dimensional subspace equals the eigenvalue that is associated with the basis vector $\boldsymbol{b}_1$ that spans this subspace.

- To maximize the variance of the low-dimensional code, we choose the basis vector associated with the largest eigenvalue of the data covariance matrix. This eigenvector is called the first principal component.

10.2.2 M-dimensional Subspace with Maximal Variance

- Assume we have found the $m-1$ eigenvectors of $\boldsymbol{S}$ that are associated with the largest $m-1$ eigenvalues.
- We want to find the mth principal component.
- We subtract the effect of the first $m-1$ principal components $\boldsymbol{b}_1 \cdots, \boldsymbol{b}_{m-1}$ from the data, and find principal components that compress the remaining information. We then arrive at the new data matrix,

$$\hat{\boldsymbol{X}} := \boldsymbol{X} - \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{B}_{m-1} \boldsymbol{X}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ contains the data points as column vectors and $\boldsymbol{B}_{m-1} := \sum_{i=1}^{m-1} \boldsymbol{b}_i \boldsymbol{b}_i^{\mathrm{T}}$ is a projection matrix that projects onto the subspace spanned by $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{m-1}$.

- To find the mth principal component, we maximize the variance

$$V_m = \mathbb{V}[z_m] = \frac{1}{N} \sum_{n=1}^{N} z_{mn}^2 = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{b}_m^{\mathrm{T}} \widehat{\boldsymbol{x}_n})^2 = \boldsymbol{b}_m^{\mathrm{T}} \hat{\boldsymbol{S}} \boldsymbol{b}_m$$

subject to $\|\boldsymbol{b}_m\|^2 = 1$, and we define $\hat{\boldsymbol{S}}$ as the data covariance matrix of the transformed dataset $\hat{\mathcal{X}} := \{\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_N\}$.

- The optimal solution $\boldsymbol{b}_m$ is the eigenvector of $\widehat{\boldsymbol{S}}$ that is associated with the largest eigenvalue of $\widehat{\boldsymbol{S}}$.

- In fact, we can derive that

$$\widehat{\boldsymbol{S}}\boldsymbol{b}_m = \boldsymbol{S}\boldsymbol{b}_m = \lambda_m \boldsymbol{b}_m \quad (1)$$

- $\boldsymbol{b}_m$ is not only an eigenvector of $\boldsymbol{S}$ but also of $\widehat{\boldsymbol{S}}$.

- Specifically, λ_m is the largest eigenvalue of $\widehat{\boldsymbol{S}}$ and λ_m is the mth largest eigenvalue of $\boldsymbol{S}$, and both have the associated eigenvector $\boldsymbol{b}_m$.

- Moreover, $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{m-1}$ are also eigenvectors of $\boldsymbol{S}$, but they are associated with eigenvalue 0.

- Considering (1) and, $\boldsymbol{b}_m^{\mathrm{T}}\boldsymbol{b}_m = 1$, the variance of the data projected onto the mth principal component is

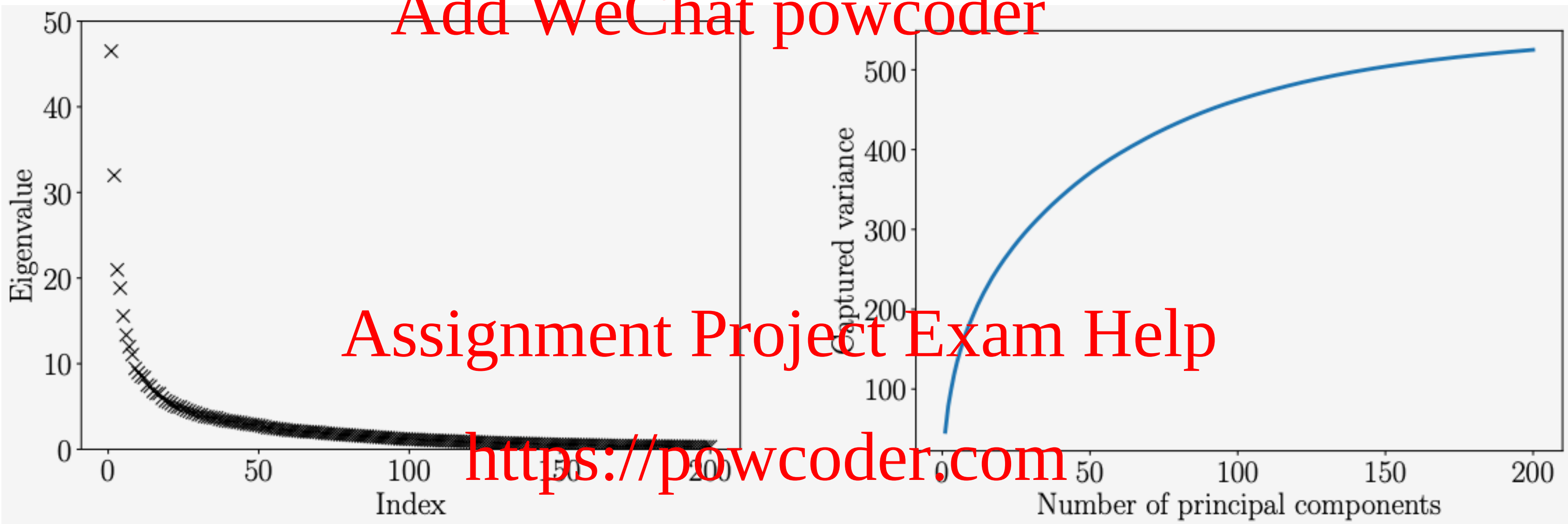
$$V_m = \boldsymbol{b}_m^{\mathrm{T}}\boldsymbol{S}\boldsymbol{b}_m = \lambda_m \boldsymbol{b}_m^{\mathrm{T}}\boldsymbol{b}_m = \lambda_m$$

- This means that the variance of the data, when projected onto an M-dimensional subspace, equals the sum of the eigenvalues that are associated with the corresponding eigenvectors of the data covariance matrix.

MNIST dataset

- 60,000 examples of handwritten digits 0 through 9.
- Each digit is a grayscale image of size 28×28, i.e., it contains 784 pixels.
- We can interpret every image in this dataset as a vector $x \in \mathbb{R}^{784}$

Example - Eigenvalues of MNIST digit "8"



(a) Top 200 largest eigenvalues

(b) Variance captured by the principal components.

- A 784-dim vector is used to represent an image
- Taking all images of "8" in MNIST, we compute the eigenvalues of the data covariance matrix.
- We see that only a few of them have a value that differs significantly from 0.
- Most of the variance, when projecting data onto the subspace spanned by the corresponding eigenvectors, is captured by only a few principal components

Overall

- To find an M-dimensional subspace of $\mathbb{R}^D$ that retains as much information as possible,
- We choose the columns of $\boldsymbol{B} = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$ as the M eigenvectors of the data covariance matrix $\boldsymbol{S}$ that are associated with the M largest eigenvalues.

- The maximum amount of variance PCA can capture with the first M principal components is

$$V_M = \sum_{m=1}^{M} \lambda_m$$

where the λ_m are the M largest eigenvalues of the data covariance matrix $\boldsymbol{S}$.

- The variance lost by data compression via PCA is

$$J_M = \sum_{j=M+1}^{D} \lambda_j = V_D - V_M$$

- Instead of these absolute quantities, we can define the relative variance captured as $\frac{V_M}{V_D}$, and the relative variance lost by compression as $1 - \frac{V_M}{V_D}$.

10.3 PCA from Projection Perspective

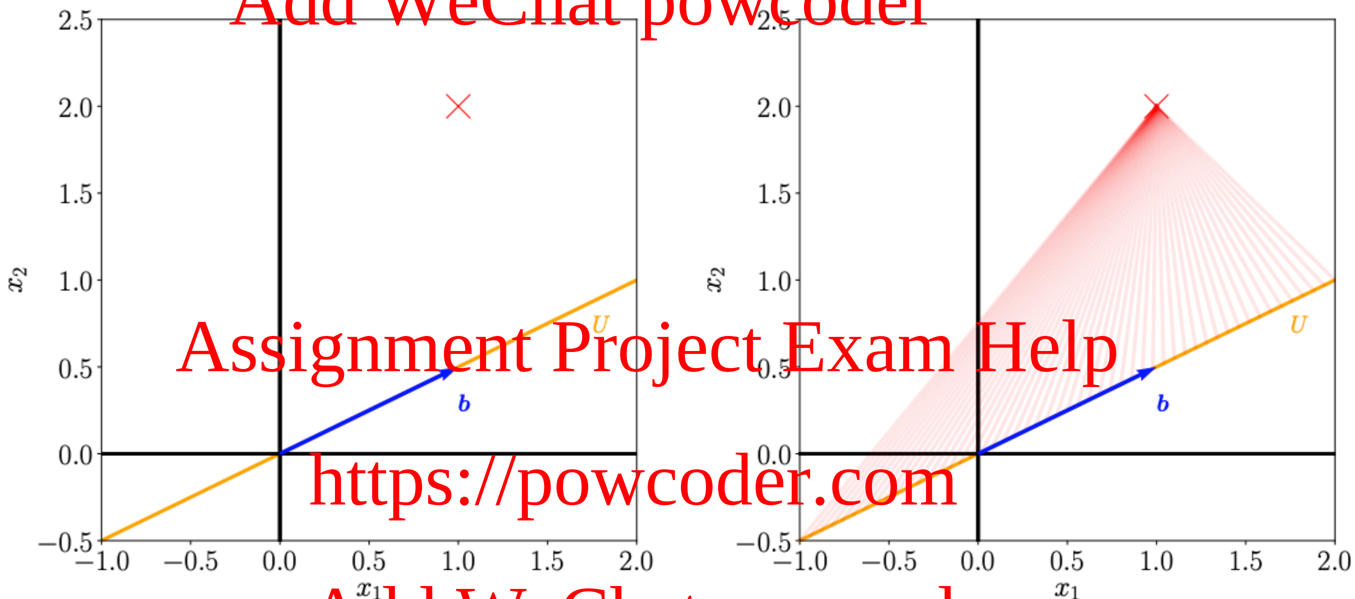
- Previously, we derived PCA by maximizing the variance in the projected space to retain as much information as possible

$$\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{S} \boldsymbol{b}_1$$

$$\text{subject to } \|\boldsymbol{b}_1\|^2 = 1$$

- Alternatively, we derive PCA as an algorithm that directly minimizes the average reconstruction error

10.3.1 Setting and Objective



(a) A vector $\boldsymbol{x} \in \mathbb{R}^2$ (red cross) shall be projected onto a one-dimensional subspace $U \subseteq \mathbb{R}^2$ spanned by $\boldsymbol{b}$

(b) Differences $\boldsymbol{x} - \tilde{\boldsymbol{x}}_i$ for 50 different $\tilde{\boldsymbol{x}}_i$ are shown by the red lines

- We wish to project $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ in a lower-dimensional space, such that $\tilde{\boldsymbol{x}}$ is similar to the original data point $\boldsymbol{x}$. That is,
- We aim to minimize the (Euclidean) distance $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$

- Given an orthonormal basis $(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_D)$ of $\mathbb{R}^D$, any $\boldsymbol{x} \in \mathbb{R}^D$ can be written as a linear combination of the basis vectors of $\mathbb{R}^D$:

$$\boldsymbol{x} = \sum_{d=1}^{D} \zeta_d \, \boldsymbol{b}_d = \sum_{m=1}^{M} \zeta_m \, \boldsymbol{b}_m + \sum_{j=M+1}^{D} \zeta_j \, \boldsymbol{b}_j$$

for suitable coordinates $\zeta_d \in \mathbb{R}$.

- We aim to find vectors $\tilde{\boldsymbol{x}} \in \mathbb{R}^D$, which live in an intrinsically lower-dimensional subspace $U \subseteq \mathbb{R}^D$, $\dim(U) = M$, so that

$$\tilde{\boldsymbol{x}} = \sum_{m=1}^{M} z_m \, \boldsymbol{b}_m \in U \subseteq \mathbb{R}^D$$

is as similar to $\boldsymbol{x}$ as possible.

- We have a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_N \in \mathbb{R}^D$ centered at $\mathbf{0}$, i.e., $\mathbb{E}[\mathcal{X}] = \mathbf{0}$.

- We want to find the best linear projection of $\mathcal{X}$ onto a lower dimensional subspace $U \subseteq \mathbb{R}^D$, $\dim(U) = M$. Also, U has orthonormal basis vectors $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M$.
- We call this subspace U the principal subspace.
- The projections of the data points are denoted by

$$\tilde{\boldsymbol{x}}_n := \sum_{m=1}^{M} z_{mn}\, \boldsymbol{b}_m = \boldsymbol{B}\boldsymbol{z}_n \in \mathbb{R}^D$$

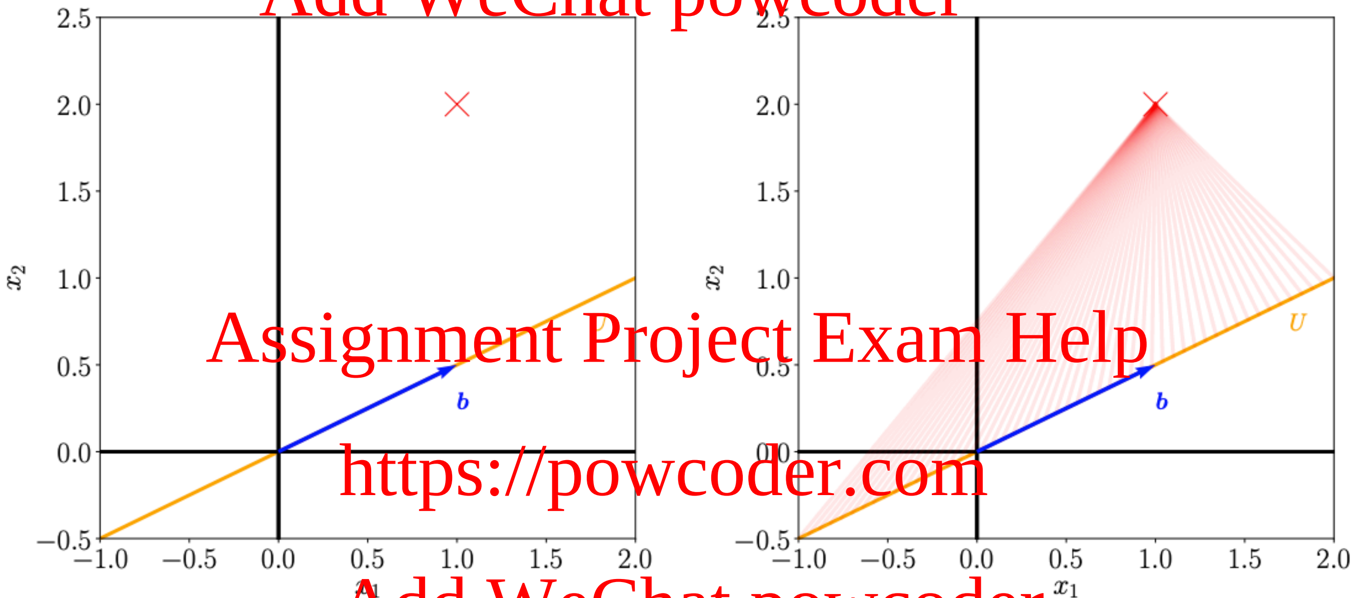
where $\boldsymbol{z}_n := [z_{1n}, \cdots, z_{Mn}]^{\mathrm{T}} \in \mathbb{R}^M$ is the coordinate vector of $\tilde{\boldsymbol{x}}_n$ with respect to the basis $(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M)$.

- We want to have $\tilde{\boldsymbol{x}}_n$ as similar to $\boldsymbol{x}_n$ as possible.
- We define our objective as minimizing the average squared Euclidean distance (reconstruction error)

$$J_M := \frac{1}{N} \sum_{n=1}^{N} \|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

- We need to find the orthonormal basis of the principal subspace and the coordinates $z_n \in \mathbb{R}^M$ of the projections with respect to this basis.

10.3.2 Finding Optimal Coordinates



(a) A vector $\boldsymbol{x} \in \mathbb{R}^2$ (red cross) shall be projected onto a one-dimensional subspace $U \subseteq \mathbb{R}^2$ spanned by $\boldsymbol{b}$

(b) Differences $\boldsymbol{x} - \tilde{\boldsymbol{x}}_i$ for 50 different $\tilde{\boldsymbol{x}}_i$ are shown by the red lines

- We want to find $\tilde{\boldsymbol{x}}$ in a subspace spanned by $\boldsymbol{b}$ that minimizes $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$.
- Apparently, this will be the orthogonal projection

$$\boxed{J_M := \frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2}$$

- Given an ONB $(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M)$ of $U \subseteq \mathbb{R}^D$, to find the optimal coordinates $\boldsymbol{z}_m$ with respect to this basis, we calculate the partial derivatives

$$\frac{\partial J_M}{\partial z_{in}} = \frac{\partial J_M}{\partial \tilde{\boldsymbol{x}}_n}\frac{\partial \tilde{\boldsymbol{x}}_n}{\partial z_{in}}$$

$$\frac{\partial J_M}{\partial \tilde{\boldsymbol{x}}_n} = -\frac{2}{N}(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^{\mathrm{T}} \in \mathbb{R}^{1\times D}$$

$$\boxed{\tilde{\boldsymbol{x}}_n := \sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m = \boldsymbol{B}\boldsymbol{z}_n \in \mathbb{R}^D}$$

$$\frac{\partial \tilde{\boldsymbol{x}}_n}{\partial z_{in}} = \frac{\partial}{\partial z_{in}}\left(\sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m\right) = \boldsymbol{b}_i$$

for $i = 1, \cdots, M$, such that we obtain

$$\frac{\partial J_M}{\partial z_{in}} = -\frac{2}{N}(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^{\mathrm{T}}\boldsymbol{b}_i = -\frac{2}{N}\left(\boldsymbol{x}_n - \sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m\right)^{\mathrm{T}}\boldsymbol{b}_i$$

$$\overset{\text{ONB}}{=} -\frac{2}{N}\left(\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{b}_i - z_{in}\boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{b}_i\right) = -\frac{2}{N}(\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{b}_i - z_{in})$$

$$\frac{\partial J_M}{\partial z_{in}} = \frac{2}{N}(\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{b}_i - z_{in})$$

- Setting this partial derivative to 0 yields immediately the optimal coordinates

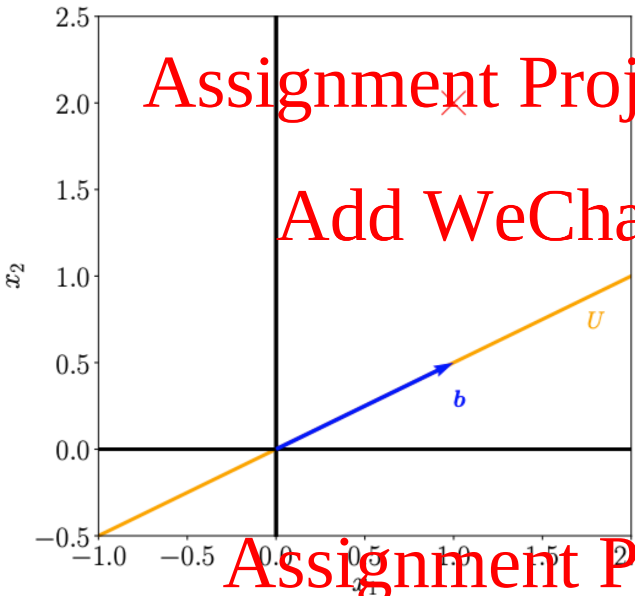
$$z_{in} = \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{b}_i = \boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}_n$$

for $i = 1, \cdots, M$, and $n = 1, \cdots, N$.

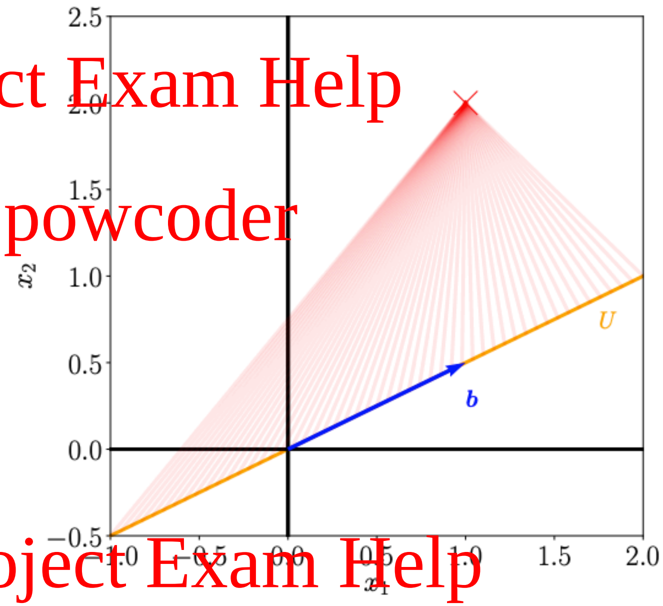
- The optimal coordinates z_{in} of the projection $\tilde{\boldsymbol{x}}_n$ are the coordinates of the orthogonal projection of the original data point $\boldsymbol{x}_n$ onto the one-dimensional subspace that is spanned by $\boldsymbol{b}_i$.

- The optimal linear projection $\tilde{\boldsymbol{x}}_n$ of $\boldsymbol{x}_n$ is an orthogonal projection.

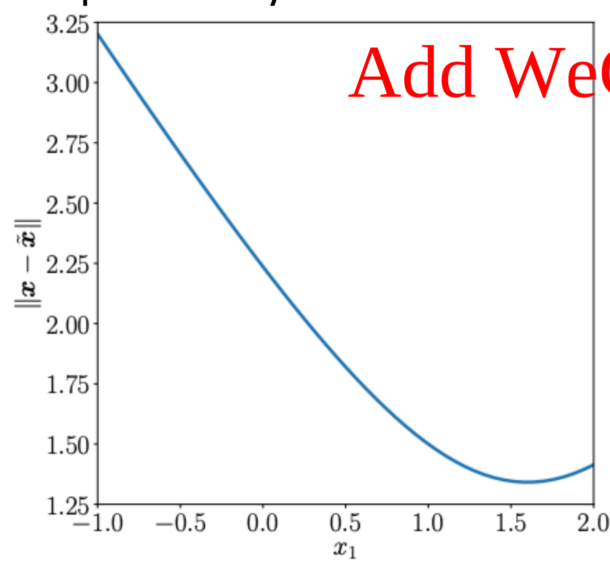
- The coordinates of $\tilde{\boldsymbol{x}}_n$ with respect to the basis $(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M)$ are the coordinates of the orthogonal projection of $\boldsymbol{x}_n$ onto the principal subspace.



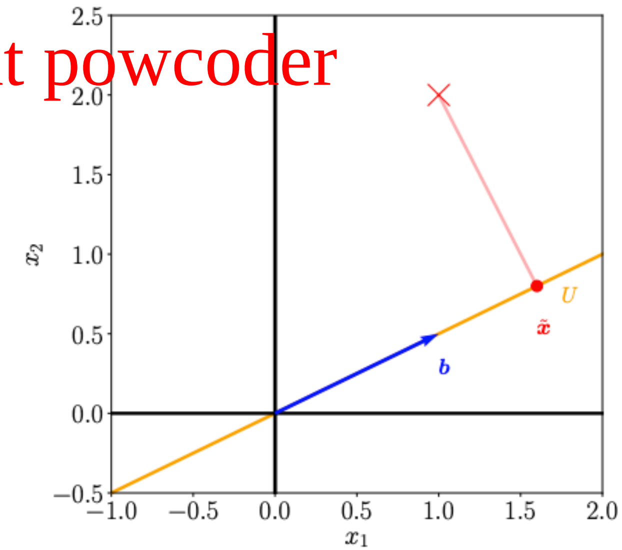
(a) A vector $\boldsymbol{x} \in \mathbb{R}^2$ (red cross) shall be projected onto a one-dimensional subspace $U \subseteq \mathbb{R}^2$ spanned by $\boldsymbol{b}$



(b) Differences $\boldsymbol{x} - \tilde{\boldsymbol{x}}_i$ for 50 different $\tilde{\boldsymbol{x}}_i$ are shown by the red lines



(c) Distances $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$ for some $\tilde{\boldsymbol{x}} = z_1 \boldsymbol{b} \in U = \text{span}[\boldsymbol{b}]$



(d) The vector $\tilde{\boldsymbol{x}}$ that minimizes $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$ is the orthogonal projection of $\boldsymbol{x}$ onto U.

- We briefly recap orthogonal projections from Section 3.8 (Analytic geometry).

- If $(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_D)$ is an orthonormal basis of $\mathbb{R}^D$ then

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{x}}{\|\boldsymbol{b}_j\|^2} \boldsymbol{b}_j = \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{x} \in \mathbb{R}^D$$

is the orthogonal projection of $\boldsymbol{x}$ onto the subspace spanned by the jth basis vector, and $z_j = \boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{x}$ is the coordinate of this projection with respect to the basis vector $\boldsymbol{b}_j$ that spans that subspace since $z_j \boldsymbol{b}_j = \tilde{\boldsymbol{x}}$.

- More generally, if we aim to project onto an M-dimensional subspace of $\mathbb{R}^D$, we obtain the orthogonal projection of $\boldsymbol{x}$ onto the M-dimensional subspace with orthonormal basis vectors $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M$ as

$$\tilde{\boldsymbol{x}} = \boldsymbol{B} \underbrace{\left(\boldsymbol{B}^{\mathrm{T}} \boldsymbol{B}\right)}_{= \boldsymbol{I}}{}^{-1} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{x} = \boldsymbol{B} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{x}$$

where we defined $\boldsymbol{B} := [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$. The coordinates of this projection with respect to the ordered basis $(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M)$ are $\boldsymbol{z} := \boldsymbol{B}^{\mathrm{T}} \boldsymbol{x}$

- Although $\tilde{\boldsymbol{x}} \in \mathbb{R}^{\boldsymbol{D}}$, we only need M coordinates to represent $\tilde{\boldsymbol{x}}$. The other $D - M$ coordinates with respect to the basis vectors $(\boldsymbol{b}_{M+1}, \cdots, \boldsymbol{b}_D)$ are always 0

10.3.3 Finding the Basis of the Principal Subspace

- So far we have shown that for a given ONB we can find the optimal coordinates of $\tilde{\boldsymbol{x}}$ by an orthogonal projection onto the principal subspace. In the following, we will determine what the best basis is.
- Recall the optimal coordinates of $\tilde{\boldsymbol{x}}$ given ONB is

$$z_{in} = \boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{b}_i = \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}_n$$

- We have

$$\tilde{\boldsymbol{x}}_n = \sum_{m=1}^{M} z_{mn} \, \boldsymbol{b}_m = \sum_{m=1}^{M} (\boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{b}_m) \, \boldsymbol{b}_m$$

- We now exploit the symmetry of the dot product, which yields

$$\tilde{\boldsymbol{x}}_n = \sum_{m=1}^{M} (\boldsymbol{b}_m^{\mathrm{T}} \boldsymbol{x}_n) \boldsymbol{b}_m = \sum_{m=1}^{M} \boldsymbol{b}_m (\boldsymbol{b}_m^{\mathrm{T}} \boldsymbol{x}_n) = \left(\sum_{m=1}^{M} \boldsymbol{b}_m \boldsymbol{b}_m^{\mathrm{T}} \right) \boldsymbol{x}_n$$

- Since we can generally write the original data point $\boldsymbol{x}_n$ as a linear combination of all basis vectors, it holds that

$$\boldsymbol{x}_n = \sum_{d=1}^{D} z_{dn} \, \boldsymbol{b}_d = \sum_{d=1}^{D} (\boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{b}_d) \, \boldsymbol{b}_d = \left(\sum_{d=1}^{D} \boldsymbol{b}_d \boldsymbol{b}_d^{\mathrm{T}} \right) \boldsymbol{x}_n$$

$$= \left(\sum_{m=1}^{M} \boldsymbol{b}_m \boldsymbol{b}_m^{\mathrm{T}} \right) \boldsymbol{x}_n + \left(\sum_{j=M+1}^{D} \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}} \right) \boldsymbol{x}_n$$

where we split the sum with D terms into a sum over M and a sum over $D - M$ terms.

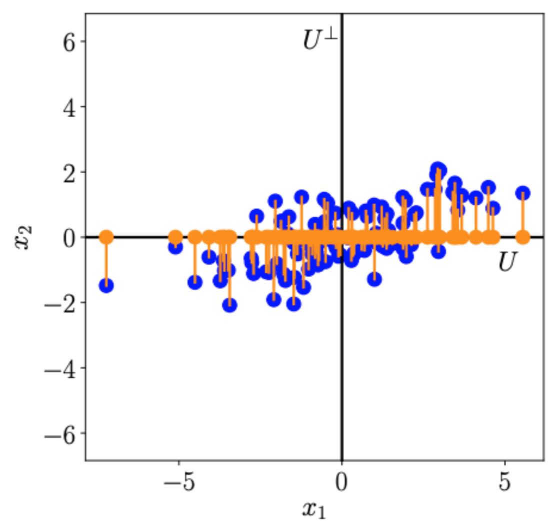
- With these results, the displacement vector $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$, i.e., the difference vector between the original data point and its projection, is

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \left(\sum_{j=M+1}^{D} \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}} \right) \boldsymbol{x}_n = \sum_{j=M+1}^{D} \left(\boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{b}_j \right) \boldsymbol{b}_j$$

- The displacement vector $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ is exactly the projection of the data point onto the orthogonal complement of the principal subspace.

- $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ lies in the subspace that is orthogonal to the principal subspace.

- We identify the matrix $\sum_{j=M+1}^{D} \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}}$ in the equation above as the projection matrix that performs this projection.



Orthogonal projection and displacement vectors. When projecting data points $\boldsymbol{x}_n$ (blue) onto subspace U_1, we obtain $\tilde{\boldsymbol{x}}_n$ (orange). The displacement vector $\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n$ lies completely in the orthogonal complement U_2 of U_1.

- Now we reformulate the loss function.

$$J_M = \frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{j=M+1}^{D}(\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{x}_n)\boldsymbol{b}_j\right\|^2$$

- We explicitly compute the squared norm and exploit the fact that the $\boldsymbol{b}_j$ form an ONB:

$$J_M = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D}(\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D}\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{x}_n\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{x}_n$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D}\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{b}_j$$

where we exploited the symmetry of the dot product in the last step to write $\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{x}_n = \boldsymbol{x}_n^{\mathrm{T}}\boldsymbol{b}_j$. We now swap the sums and obtain

$$J_M = \sum_{j=M+1}^{D}\boldsymbol{b}_j^{\mathrm{T}}\underbrace{\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{T}}\right)}_{=:\boldsymbol{S}}\boldsymbol{b}_j = \sum_{j=M+1}^{D}\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{S}\boldsymbol{b}_j$$

$$= \sum_{j=M+1}^{D}\mathrm{tr}(\boldsymbol{b}_j^{\mathrm{T}}\boldsymbol{S}\boldsymbol{b}_j) = \sum_{j=M+1}^{D}\mathrm{tr}(\boldsymbol{S}\boldsymbol{b}_j\boldsymbol{b}_j^{\mathrm{T}}) = \mathrm{tr}\left(\underbrace{\left(\sum_{j=M+1}^{D}\boldsymbol{b}_j\boldsymbol{b}_j^{\mathrm{T}}\right)}_{\text{projection matrix}}\boldsymbol{S}\right)$$

where we exploited the property that the trace operator $\mathrm{tr}(\cdot)$ is linear and invariant to cyclic permutations of its arguments

$$J_M = \sum_{j=M+1}^{D} \boldsymbol{b}_j^{\mathrm{T}} \boldsymbol{S} \boldsymbol{b}_j = \mathrm{tr}\left(\underbrace{\left(\sum_{j=M+1}^{D} \boldsymbol{b}_j \boldsymbol{b}_j^{\mathrm{T}} \right)}_{\text{projection matrix}} \boldsymbol{S} \right)$$

- The loss is formulated as the covariance matrix of the data, projected onto the orthogonal complement of the principal subspace.
- Minimizing the average squared reconstruction error is therefore equivalent to minimizing the variance of the data when projected onto the subspace we ignore, i.e., the orthogonal complement of the principal subspace.
- Equivalently, we maximize the variance of the projection that we retain in the principal subspace, which links the projection loss immediately to the maximum-variance formulation of PCA in Section 10.2.
- In Section 10.2, the average squared reconstruction error, when projecting onto the M-dimensional principal subspace, is

$$J_M = \sum_{j=M+1}^{D} \lambda_j$$

- where λ_j are the eigenvalues of the data covariance matrix.

$$J_M = \sum_{j=M+1}^{D} \lambda_j$$

- To minimize it, we need to select the smallest $D - M$ eigenvalues. Their corresponding eigenvectors are the basis of the orthogonal complement of the principal subspace.

- Consequently, this means that the basis of the principal subspace comprises the eigenvectors $\boldsymbol{b}_1, \dots, \boldsymbol{b}_M$ that are associated with the largest M eigenvalues of the data covariance matrix.

10.5 PCA in High Dimensions

- In order to do PCA, we need to compute the data covariance matrix $\boldsymbol{S}$
- In D dimensions, $\boldsymbol{S}$ is a $D \times D$ matrix.
- Computing the eigenvalues and eigenvectors of this matrix is computationally expensive as it scales cubically in D.

- Therefore, PCA will be infeasible in very high dimensions

- For example, if $\boldsymbol{x}_n$ are images with $10{,}000$ pixels, we would need to compute the eigendecomposition of a $10{,}000 \times 10{,}000$ matrix.

- We provide a solution to this problem for the case that we have substantially fewer data points than dimensions, i.e., $N \ll D$
- Assume we have a centered dataset $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N,\ \boldsymbol{x}_n \in \mathbb{R}^D$. Then the data covariance matrix is given as

$$\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \in \mathbb{R}^{D \times D}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$ is a $D \times N$ matrix whose columns are the data points.

- We now assume that $N \ll D$, i.e., the number of data points is smaller than the dimensionality of the data.

- With $N \ll D$ data points, the rank of the covariance matrix S is at most N, so it has at least $D - N$ eigenvalues that are 0.
- Intuitively, this means that there are some redundancies. In the following, we will exploit this and turn the $D \times D$ covariance matrix into an $N \times N$ covariance matrix whose eigenvalues are all positive.

- In PCA, we ended up with the eigenvector equation

$$\boldsymbol{S}\boldsymbol{b}_m = \lambda_m \boldsymbol{b}_m, \qquad m = 1, \ldots, M$$

where $\boldsymbol{b}_m$ is a basis vector of the principal subspace. Let us rewrite this equation a bit: With $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \in \mathbb{R}^{D\times D}$, we obtain

$$\boldsymbol{S}\boldsymbol{b}_m = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{b}_m = \lambda_m \boldsymbol{b}_m$$

- We now multiply $\boldsymbol{X}^{\mathrm{T}} \in \mathbb{R}^{N\times D}$ from the left-hand side, which yields

$$\frac{1}{N}\underbrace{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}}_{N\times N}\underbrace{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{b}_m}_{=:\boldsymbol{c}_m} = \lambda_m \boldsymbol{X}^{\mathrm{T}}\boldsymbol{b}_m \Leftrightarrow \frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{c}_m = \lambda_m \boldsymbol{c}_m$$

$$\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{c}_m = \lambda_m \boldsymbol{c}_m$$

- We get a new eigenvector/eigenvalue equation: λ_m remains eigenvalue, which confirms our results from exercise 4.11 that the nonzero eigenvalues of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ equal the nonzero eigenvalues of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$.
- We obtain the eigenvector of the matrix $\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} \in \mathbb{R}^{N\times N}$ associated with λ_m as $\boldsymbol{c}_m := \boldsymbol{X}^{\mathrm{T}}\boldsymbol{b}_m$.

- This also implies that $\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ has the same (nonzero) eigenvalues as the data covariance matrix $\boldsymbol{S}$.

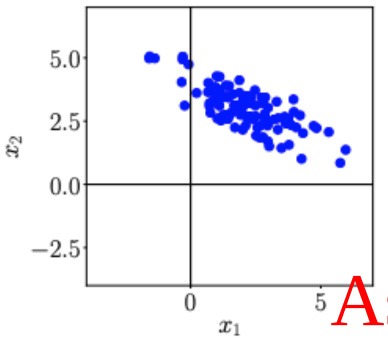
- But $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ now an $N\times N$ matrix, so that we can compute the eigenvalues and eigenvectors much more efficiently than for the original $D\times D$ data covariance matrix.

- Now that we have the eigenvectors of $\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$, we are going to recover the original eigenvectors, which we still need for PCA. Currently, we know the eigenvectors of $\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. If we left-multiply our eigenvalue/ eigenvector equation with $\boldsymbol{X}$, we get

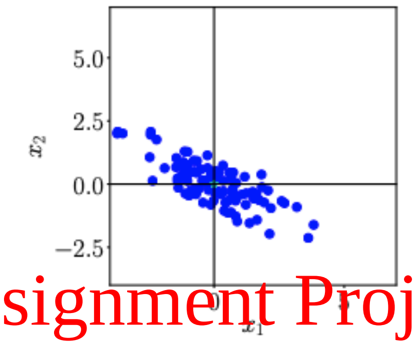
$$\underbrace{\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}}_{\boldsymbol{S}}\boldsymbol{X}\boldsymbol{c}_m = \lambda_m \boldsymbol{X}\boldsymbol{c}_m$$

and we recover the data covariance matrix again. This now also means that we recover $\boldsymbol{X}\boldsymbol{c}_m$ as an eigenvector of $\boldsymbol{S}$.

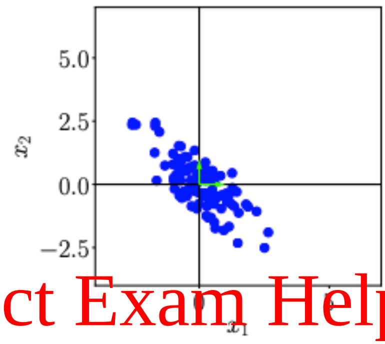
10.6 Key Steps of PCA in Practice



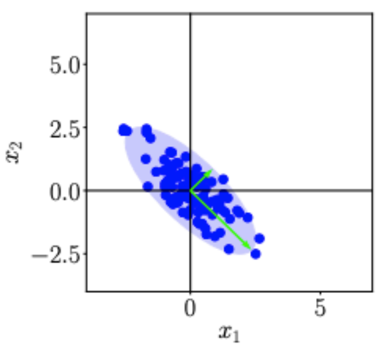
(a) Original dataset.



(b) Step 1: Centering by subtracting the mean from each data point.

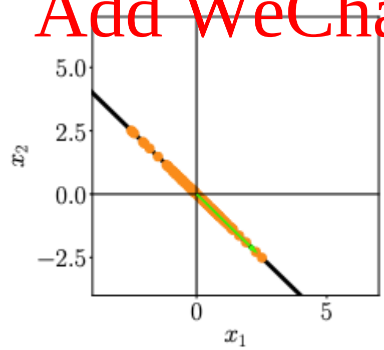


(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

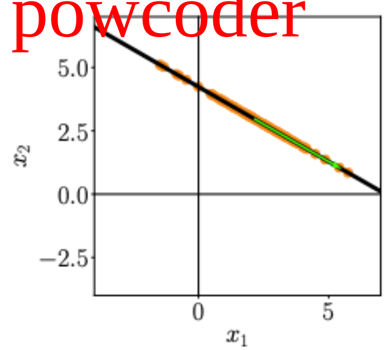


(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

eigendecomposition



(e) Step 4: Project data onto the principal subspace.

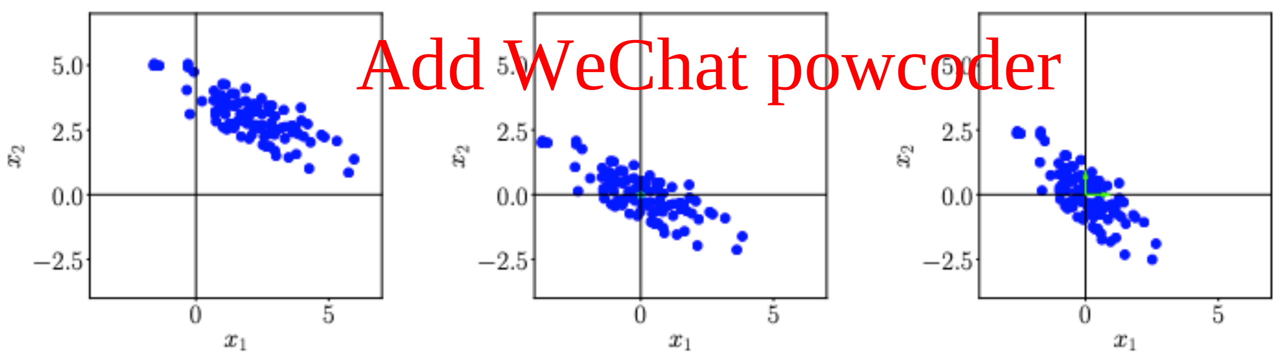


(f) Undo the standardization and move projected data back into the original data space from (a).

- **Step 1. Mean subtraction**

- We center the data by computing the mean μ of the dataset and subtracting it from every single data point. This ensures that the dataset has mean 0.

- **Step 2. Standardization** Divide the data points by the standard deviation σd of the dataset for every dimension $d = 1, \ldots, D$. Now the data has variance 1 along each axis.



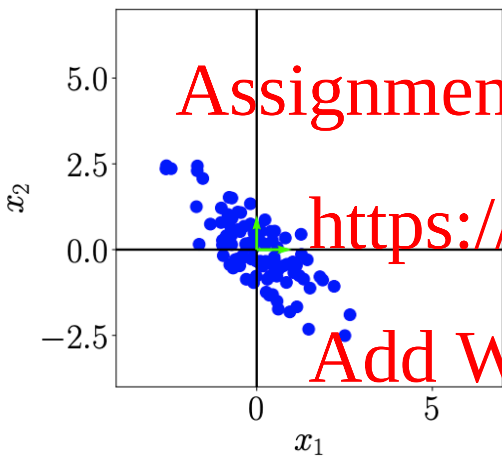
(a) Original dataset.

(b) Step 1: Centering by subtracting the mean from each data point.

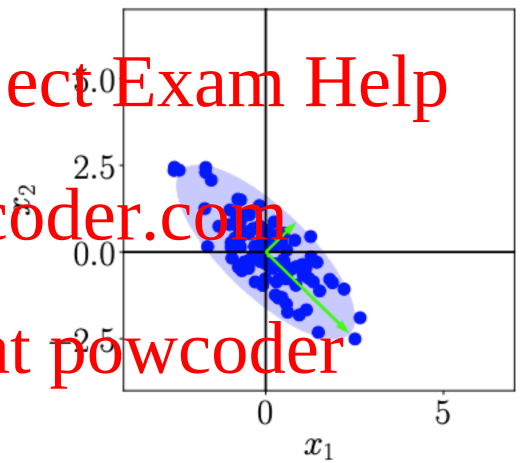
(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

- **Step 3. Eigendecomposition of the covariance matrix**

- Compute the data covariance matrix and its eigenvalues and corresponding eigenvectors. The longer vector (larger eigenvalue) spans the principal subspace U



(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.



(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

- 4. **Projection** We can project any data point $x_* \in \mathbb{R}^D$ onto the principal subspace: To get this right, we need to standardize $\boldsymbol{x}_*$ using the mean μ_d and standard deviation σ_d of the training data in the dth dimension, respectively, so that

$$x_*^{(d)} \leftarrow \frac{x_*^{(d)} - \mu_d}{\sigma_d}, \qquad d = 1, \cdots, D$$

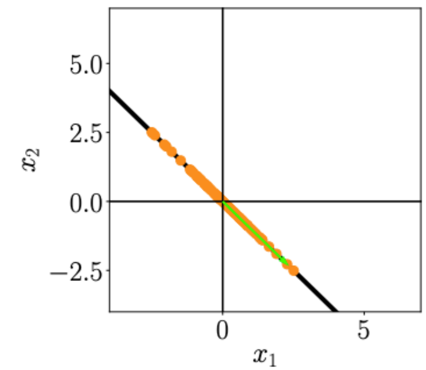
where $x_*^{(d)}$ is the dth component of $\boldsymbol{x}_*$.

- We obtain the projection as

$$\tilde{\boldsymbol{x}}_* = \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}_*$$

with coordinates

$$\boldsymbol{z}_* = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}_*$$



(e) Step 4: Project data onto the principal subspace.

with respect to the basis of the principal subspace. Here, $\boldsymbol{B}$ is the matrix that contains the eigenvectors that are associated with the largest eigenvalues of the data covariance matrix as columns.

- Note that PCA returns the coordinates $\boldsymbol{z}_*$, not the projections of $\boldsymbol{x}_*$.

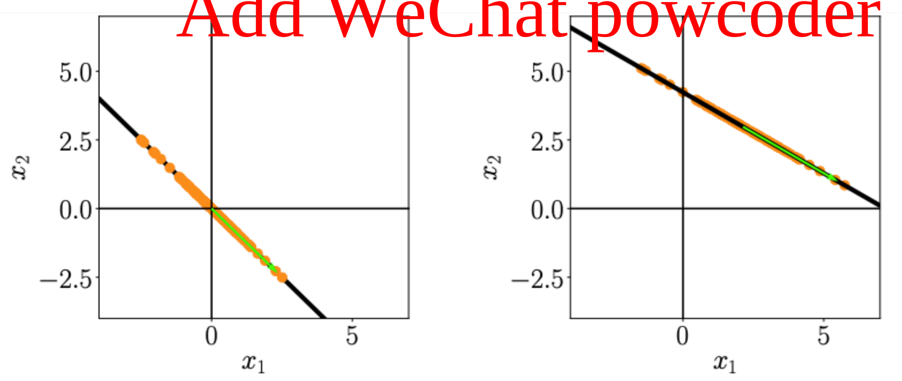
- Having standardized our dataset, $\tilde{\boldsymbol{x}}_* = \boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}_*$ only yields the projections in the context of the standardized dataset.

- To obtain our projection in the original data space (i.e., before standardization), we need to undo the standardization: multiply by the standard deviation before adding the mean.

- We obtain

$$\tilde{x}_*^{(d)} \leftarrow \tilde{x}_*^{(d)} \sigma_d + \mu_d \,, \qquad d = 1, \ldots, D$$

- Figure 10.10(f) illustrates the projection in the original data space.



(e) Step 4: Project data onto the principal subspace.

(f) Undo the standardization and move projected data back into the original data space from (a).