Principal Component Analysis

Introduction to ML

Eigen decomposition

Theorem A square matrix $A \in \mathbb{R}^{n \times n}$ can be factored into $A = PDP^{-1}$ where $P \in \mathbb{R}^{n \times n}$ and D is a diagonal matrix whose diagonal entries are the eigenvalues of A, *if and only if* the eigenvectors of A form a basis of $\mathbb{R}^n$ [A has a full set of n linearly independent eigenvectors].

$$A = PDP^{\intercal} = \begin{bmatrix} \vdots & \vdots & & \vdots \\ p_1 & p_2 & \cdots & p_n \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} \vdots & \vdots & & \vdots \\ p_1 & p_2 & \cdots & p_n \\ \vdots & \vdots & & \vdots \end{bmatrix}^{\intercal}$$

$P \in \mathbb{R}^{n \times n}$

$\Sigma \in \mathbb{R}^{n \times n}$

eigenvectors

eigenvalues

Singular Value Decomposition

Theorem (SVD) Let $A \in \mathbb{R}^{m \times n}$ be a *rectangular* matrix of rank $r \in [0, \min(m,n)]$. The SVD of A is a decomposition of the form:

$$A = U\Sigma V^\intercal = \begin{bmatrix} \vdots & \vdots & & \vdots \\ u_1 & u_2 & \cdots & u_m \\ \vdots & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & \sigma_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \vdots & \vdots & & \vdots \\ v_1 & v_2 & \cdots & v_n \\ \vdots & \vdots & & \vdots \end{bmatrix}^\intercal$$

$U \in \mathbb{R}^{m \times m}$ $\Sigma \in \mathbb{R}^{m \times n}$ $V \in \mathbb{R}^{n \times n}$

left singular vectors singular values right singular vectors

U and V are orthogonal matrices, $U^\intercal = U^{-1}, V^\intercal = V^{-1}$. Columns are orthonormal.

By convention, the singular values are ordered $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$

SVD construction: finding $V and \Sigma$

We can always eigen-decompose $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ and obtain

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{\mathrm{T}} = \boldsymbol{P}\begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}\boldsymbol{P}^{\mathrm{T}}$$

where $\boldsymbol{P}$ is an orthogonal matrix, which is composed of the orthonormal eigenbasis. $\lambda_i \geq 0$ are t

Let us assume the SVD of $\boldsymbol{A}$ exists and takes the form of $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\right) = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$$

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_n^2 \end{bmatrix}\boldsymbol{V}^{\mathrm{T}}$$

Leading to

$$\boldsymbol{V} = \boldsymbol{P}$$
$$\sigma_i^2 = \lambda_i$$

SVD construction: finding U

Note: $\boldsymbol{A} = \boldsymbol{U\Sigma V}^{\mathrm{T}} \Leftrightarrow \boldsymbol{AV} = \boldsymbol{U\Sigma V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{U\Sigma}$ which means

$$\boldsymbol{A}\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i, i = 1, \dots, r$$

where r is the rank of $\boldsymbol{A}$. So, we can calculate

$$\boldsymbol{u}_i = \frac{1}{\sigma_i}\boldsymbol{A}\boldsymbol{v}_i, i = 1, \dots, r \quad (1)$$

We look at matrices with full rank, i.e., $r = \min(m, n)$. Remember that $\boldsymbol{U}$ is an $m \times m$ matrix.

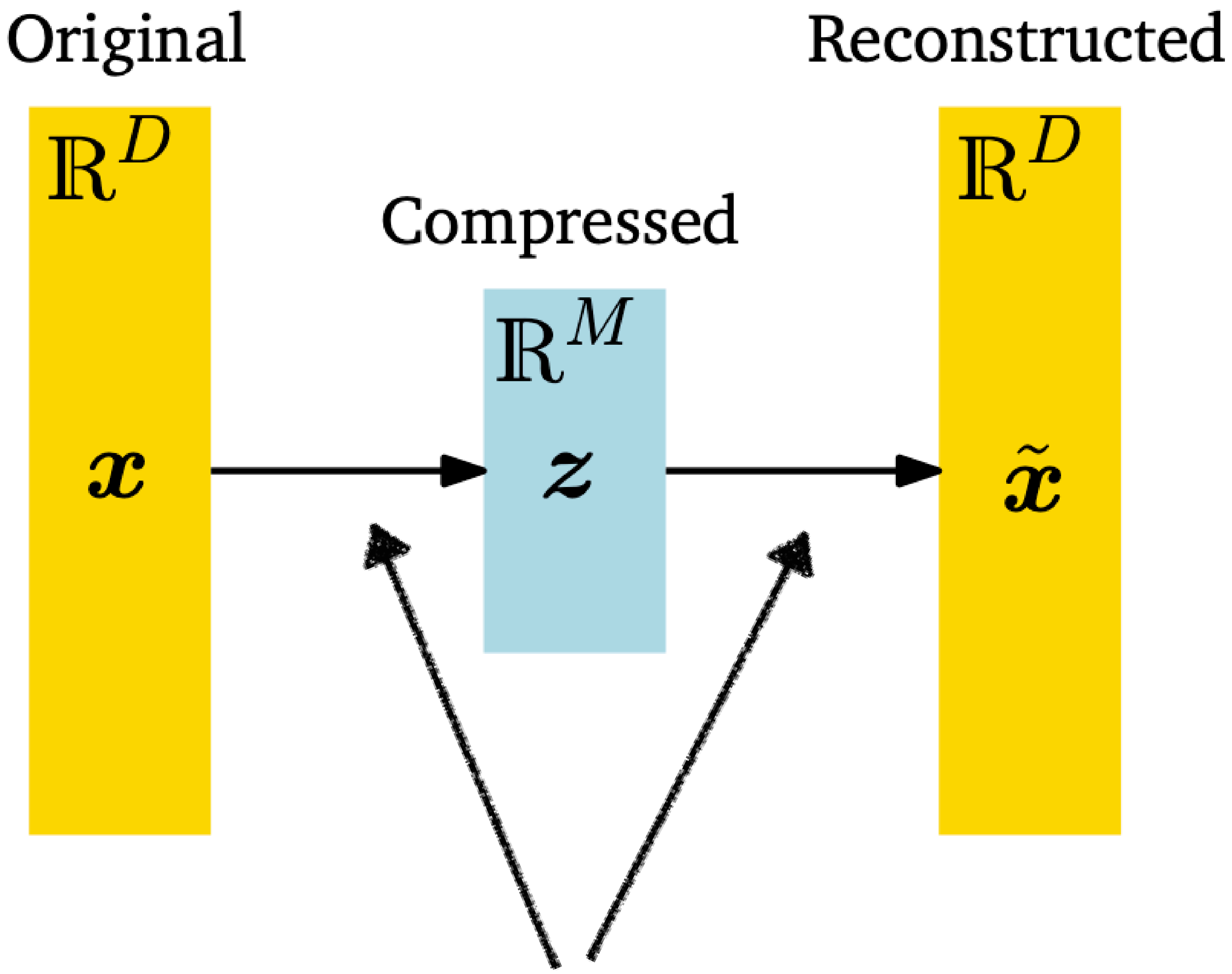
If $m \leq n$, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_m]$; All the $\boldsymbol{u}_i$ have been calculated through (1)

If $m > n$, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_n, \dots, \boldsymbol{u}_m]$;

$\boldsymbol{u}_1, \dots, \boldsymbol{u}_n$ have been calculate through (1)

In order to calculate $\boldsymbol{u}_{n+1}, \dots, \boldsymbol{u}_m$, you use the fact that $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_n, \dots, \boldsymbol{u}_m$ are orthonormal vectors.

Motivation



Dimensionality reduction as data compression

Find lower-dimensional data without losing much information

M < D

z captures desirable variations in **x**
Reconstructed data is similar to **x**

Key question: how to construct these mappings?

Why?

+ Data may have low *intrinsic* dimensionality [think about data living on a line in high dimensions]

+ visualisation / exploratory data analysis [e.g. compress 100-D data down to 2D to visualise patterns]

+ Using low dimensional data for learning [e.g. train a classifier using compressed data]

Motivation - example

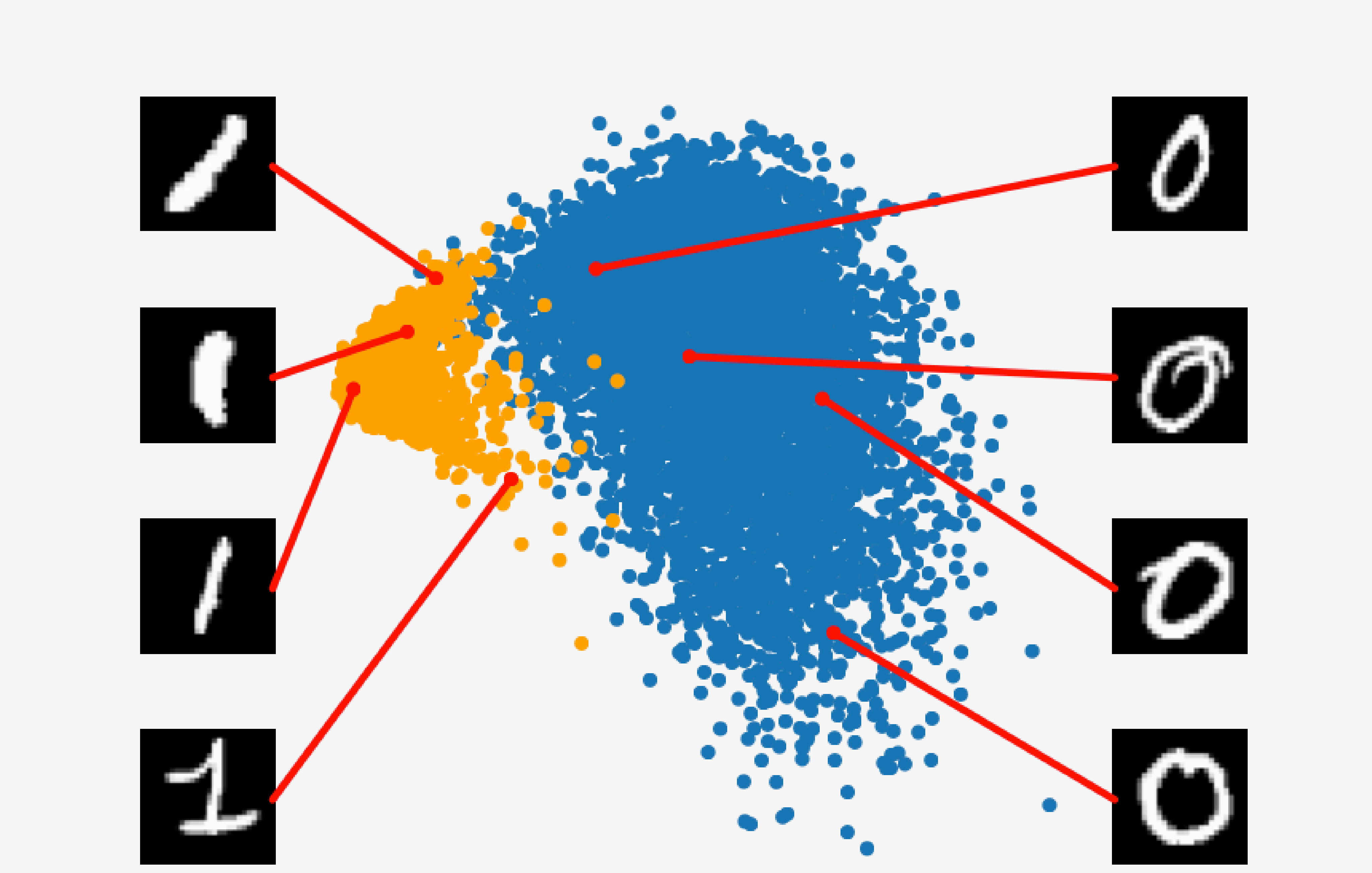
Dimensionality reduction as data compression

Find lower-dimensional data without losing much information

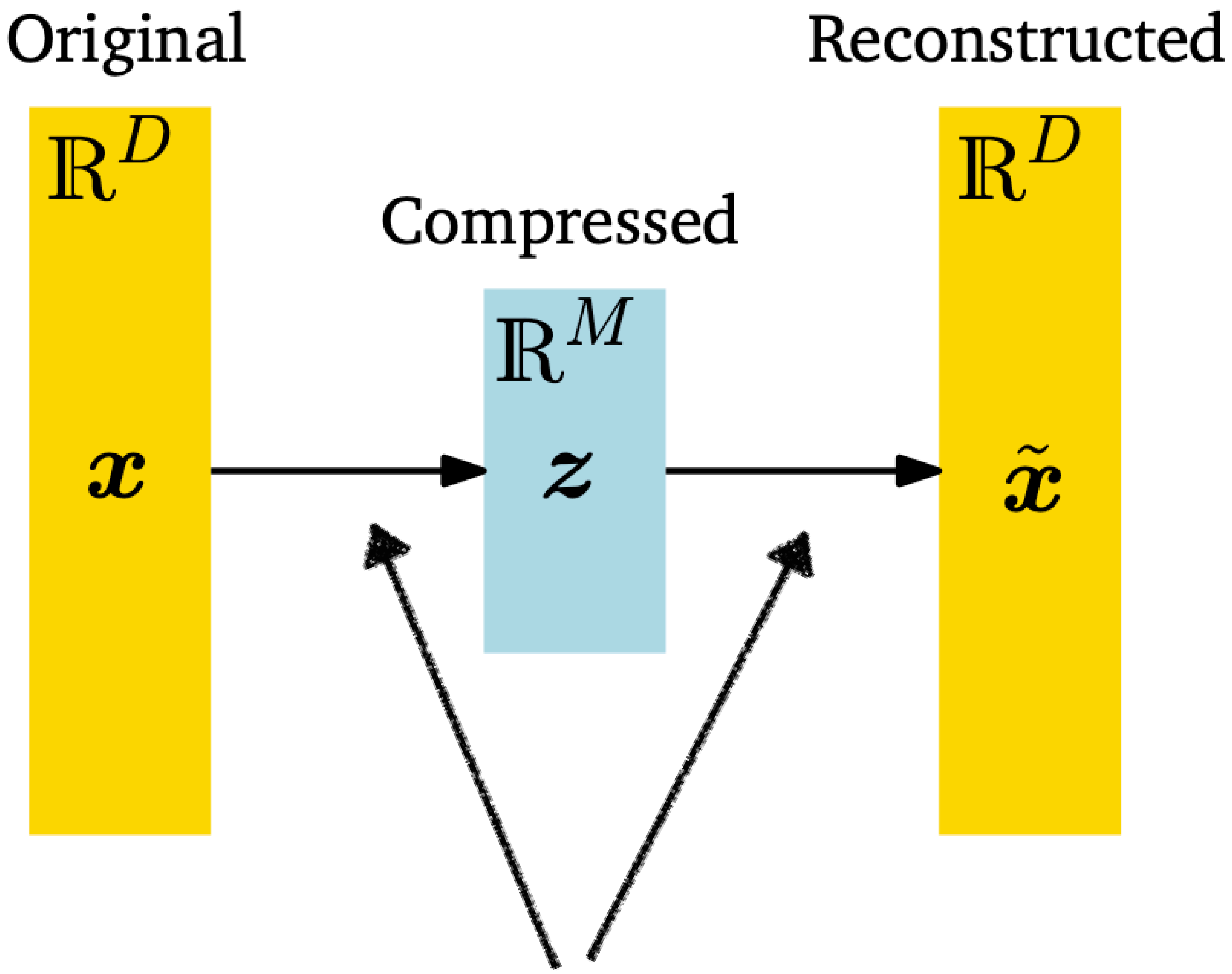
M < D

z captures desirable variations in **x**

Reconstructed data is similar to **x**



D = 784, M = 2



Key question: how to construct these mappings?

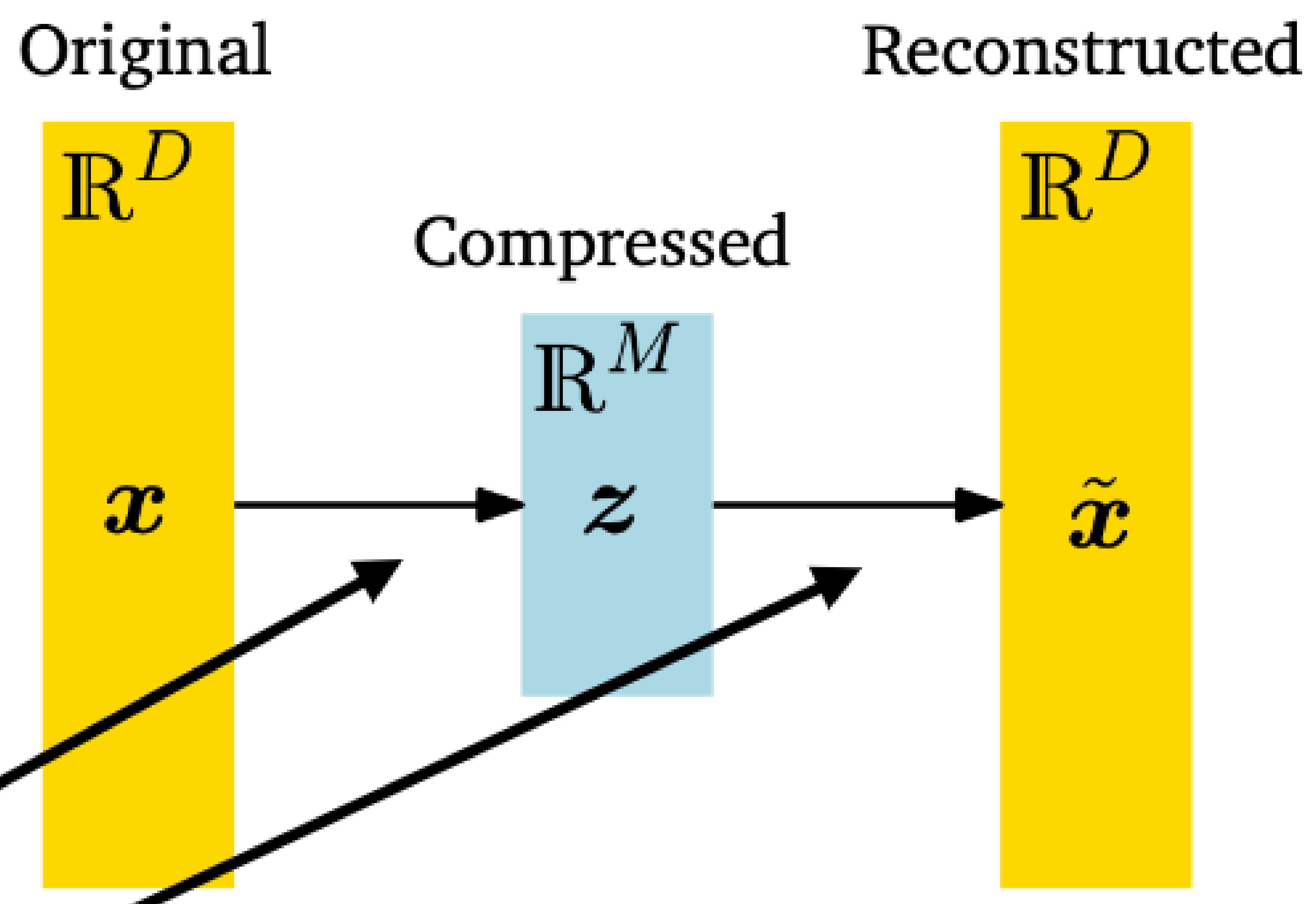
Problem setup

We consider an i.i.d. dataset $X = \{x_1, x_2, \ldots, x_N\}$, $x_n \in \mathbb{R}^D$, with mean $\mathbf{0}$ and covariance matrix $S = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^\intercal$

We assume there exists a *low-dimensional* compressed representation (code): $z_n = B^\intercal x_n,\ z_n \in \mathbb{R}^M,\ M < D$.

The projection matrix: $B = [b_1, b_2, \ldots, b_M] \in \mathbb{R}^{D\times M}$, columns are orthonormal.

Reconstruction using B: $\tilde{x}_n = B z_n$



PCA: linear mappings

Goal: find z_n and the *basis vectors* $b_1, b_2, \ldots, b_M$ so that the reconstructed data are *similar* to the original data, and the compressed data retain most of the *variation* in the original data

PCA - two perspectives

Goal: find z_n and the *basis vectors* $b_1, b_2, \dots, b_M$ so that the reconstructed data are *similar* to the original data, and the compressed data retain most of the *variation* in the original data.

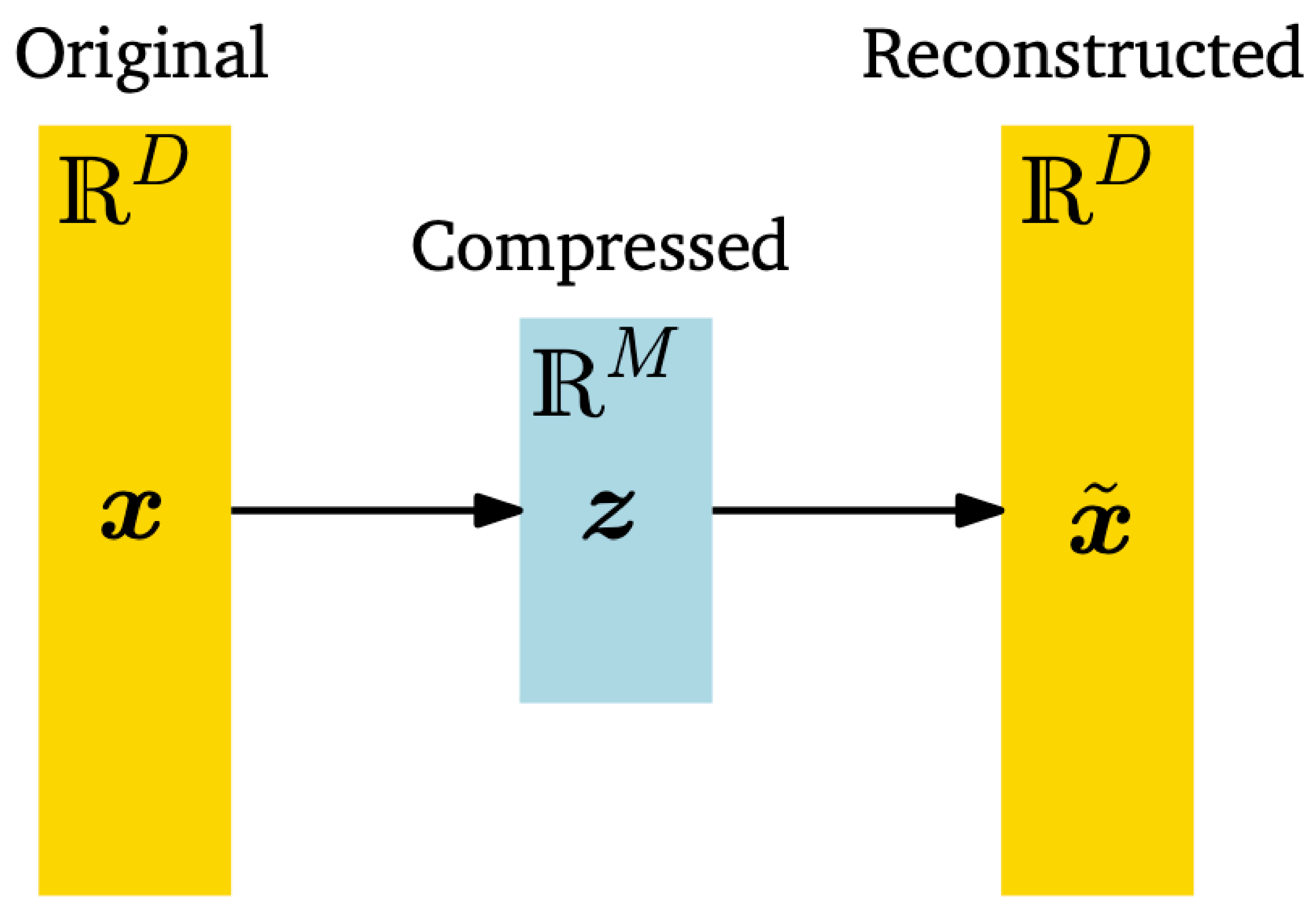
Question: Next steps? Ideas?

Answer: Two approaches

+ Search for B that **maximises** the **variance** of the low-dimensional representations [analysis/max var perspective]

+ Search for B and z that minimises the reconstruction loss [synthesis/projection perspective]

Both give *identical* solutions! **Why?**



PCA: linear mappings

$z_n = B^\intercal x_n, z_n \in \mathbb{R}^M, M < D$

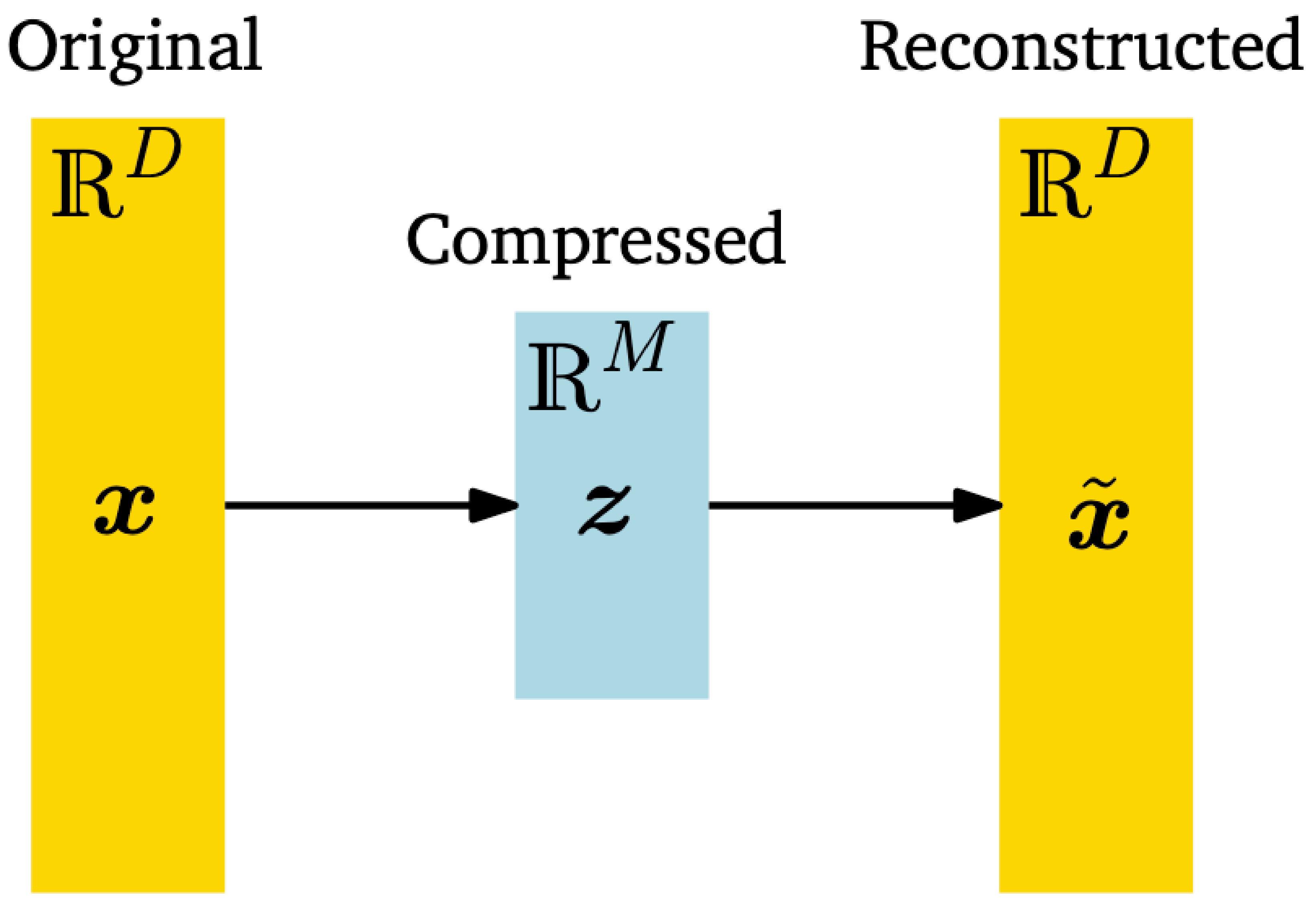
$\tilde{x}_n = B z_n$

First step: writing down the Variance

We have assumed that the mean of the data $\mu = 0$.

Data covariance matrix, $S = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^\intercal$

Variance of z: $\mathbb{V}_z[z] = \mathbb{V}_x[B^\intercal(x-\mu)] = \mathbb{V}_x[B^\intercal x]$



PCA: linear mappings

$z_n = B^\intercal x_n, z_n \in \mathbb{R}^M, M < D$

$\tilde{x}_n = B z_n$

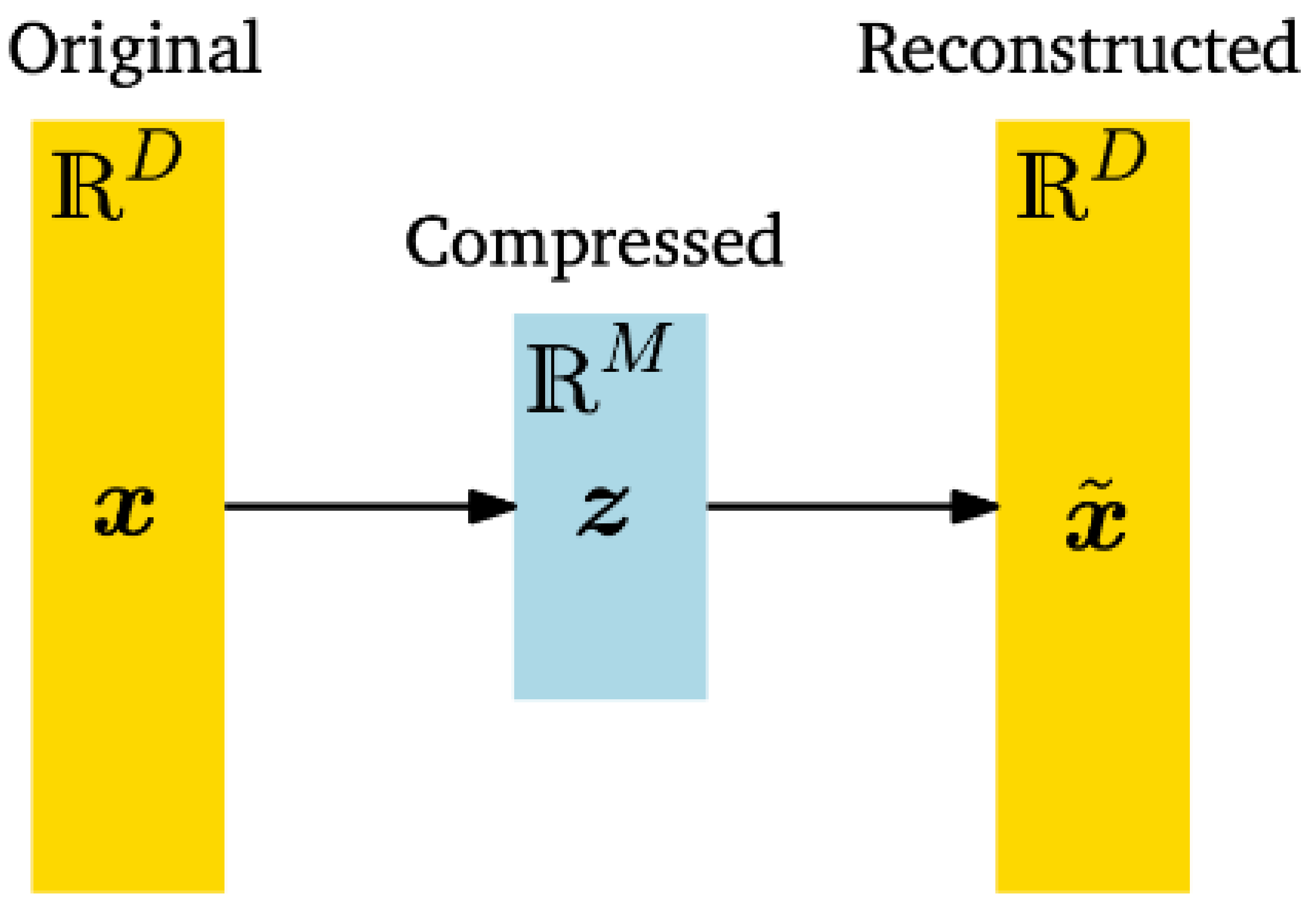
Strategy:

+ search for one single direction b_1 that gives the largest variance

+ Search for the next direction b_2 that gives the largest variance given b_1

+ … until we reach M directions

Direction with maximal variance



We first seek a single vector $b_1 \in \mathbb{R}^D$ that maximises the variance of the first coordinate z_1 of $z \in \mathbb{R}^M$: $\mathbb{V}[z_1] = \dfrac{1}{N}\sum_{n=1}^{N} z_{1n}^2$

We can show b_1 is an *eigenvector* of the data covariance matrix S

And the variance is the corresponding *eigenvalue*.

PCA: linear mappings

$z_n = B^\intercal x_n,\ z_n \in \mathbb{R}^M,\ M < D$

$\tilde{x}_n = Bz_n$

The variance of the data projected onto a one-dimensional subspace equals the eigenvalue that is associated with the basis vector b_1 that spans this subspace.

The first basis vector is the eigenvector associated with the **largest eigenvalue** of the data covariance matrix. This eigenvector is called the first **principal component**.

M-dimensional subspace with maximal variance - induction [1]

Assume we have found the $m-1$ eigenvectors of S that are associated with the largest $m-1$ eigenvalues. We want to find the m-th principal component.

We subtract the effect of the first $m-1$ principal components $b_1, \ldots, b_{m-1}$ from the data, and find principal components that compress the remaining information. We then arrive at the new data matrix, $\hat{X} = X - \sum_{i=1}^{m-1} b_i b_i^\top X = X - B_{m-1} X$, where $X, \hat{X} \in \mathbb{R}^{D\times N}$

To find the m-th principal component, we maximise the variance

$$\mathbb{V}[z_m] = \frac{1}{N}\sum_{n=1}^{N} z_{mn}^2 = b_m^\top \hat{S} b_m$$

subject to $\| b_m \|^2 = 1$, and we define $\hat{S}$ as the data covariance matrix of $\hat{X}$.

M-dimensional subspace with maximal variance - induction [2]

The optimal b_m is the eigenvector of $\hat{S}$ that is associated with the largest eigenvalue of $\hat{S}$

In fact, we can derive that

$$\hat{S}b_m = Sb_m = \lambda_m b_m$$

b_m is not only an eigenvector of S but also of $\hat{S}$.

Specifically, λ_m is the **largest** eigenvalue of $\hat{S}$ and the m**-th largest** eigenvalue of S, and both have the associated eigenvector b_m.

The variance of the data projected onto the m-th principal component is

$$V_m = b_m^\top \hat{S} b_m = b_m^\top \lambda_m b_m = \lambda_m$$

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.

Recap

Goal: To find an M-dimensional subspace of $\mathbb{R}^D$ that retains as much information as possible

Solution: We choose the columns of $B = \left[b_1, b_2, \ldots, b_M\right] \in \mathbb{R}^{D \times M}$ as the M eigenvectors of the data covariance matrix S that are associated with the M largest eigenvalues.

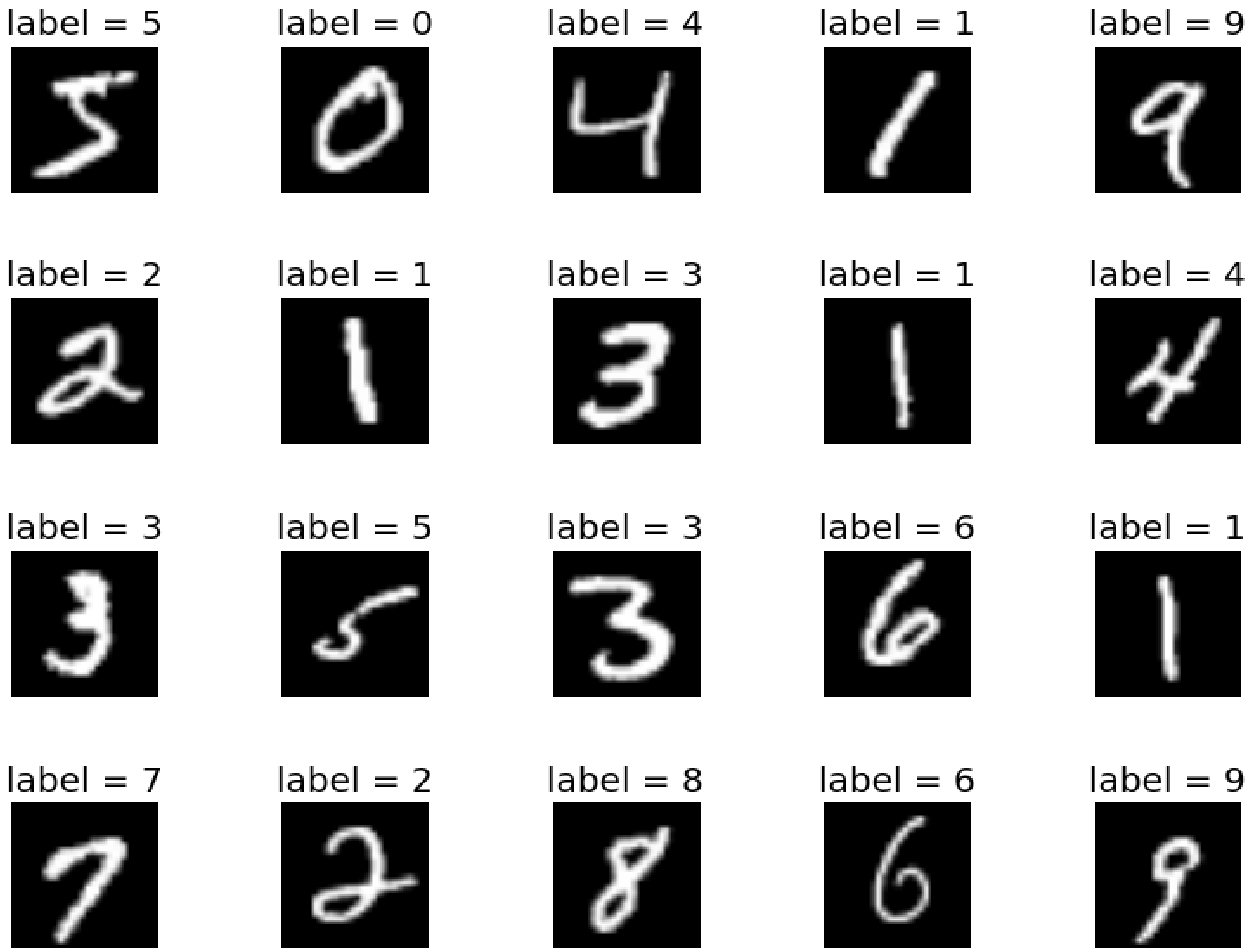
Captured variance: The maximum amount of variance PCA can capture with the first M principal components is $V_M = \sum_{m=1}^{M} \lambda_m$.

Lost variance: $J_M = \sum_{m=M+1}^{D} \lambda_m = V_D - V_M$

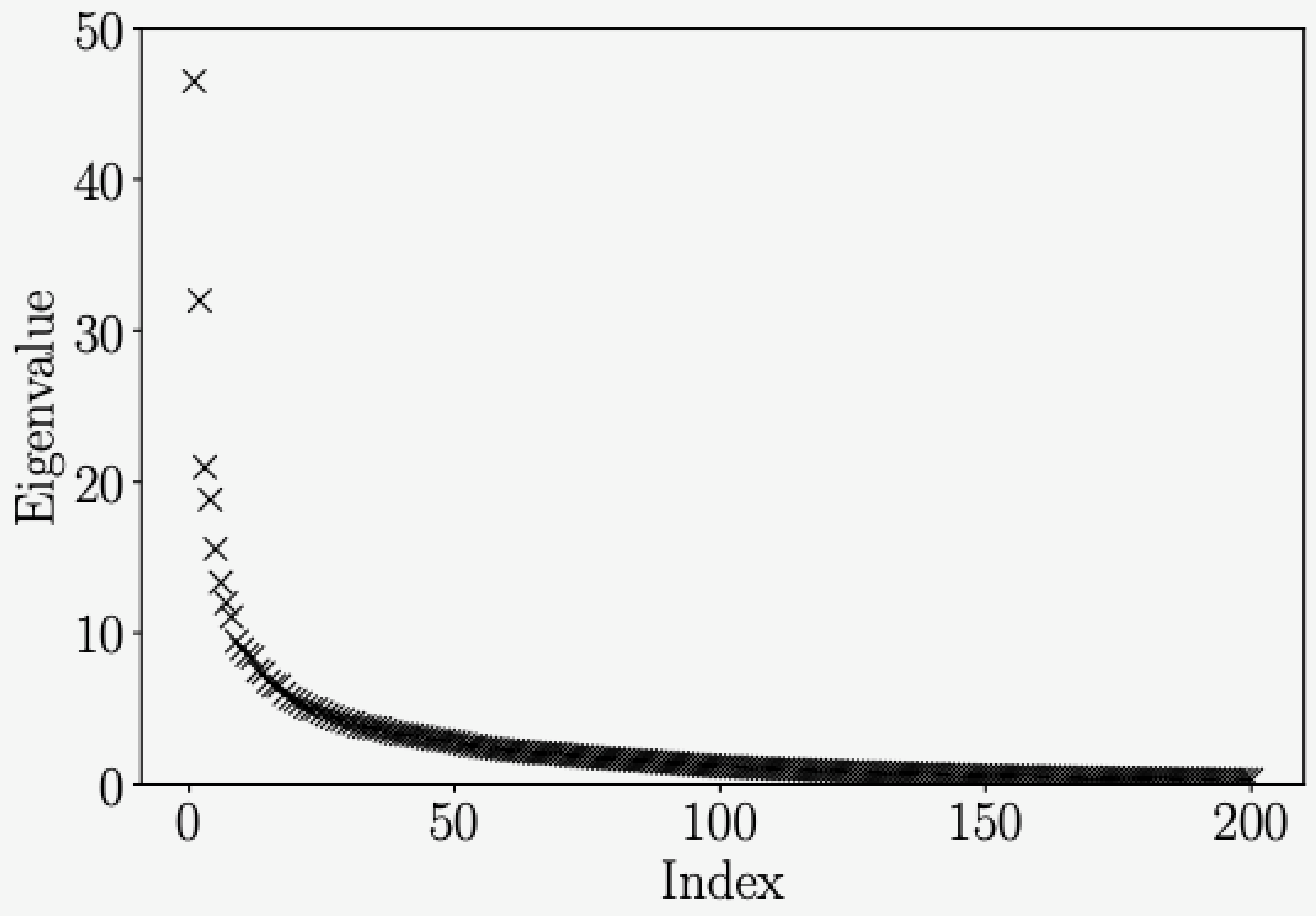
Instead of these absolute quantities, we can define the relative variance captured as V_M / V_D, and the relative variance lost by compression as $1 - V_M / V_D$.

Example - dataset

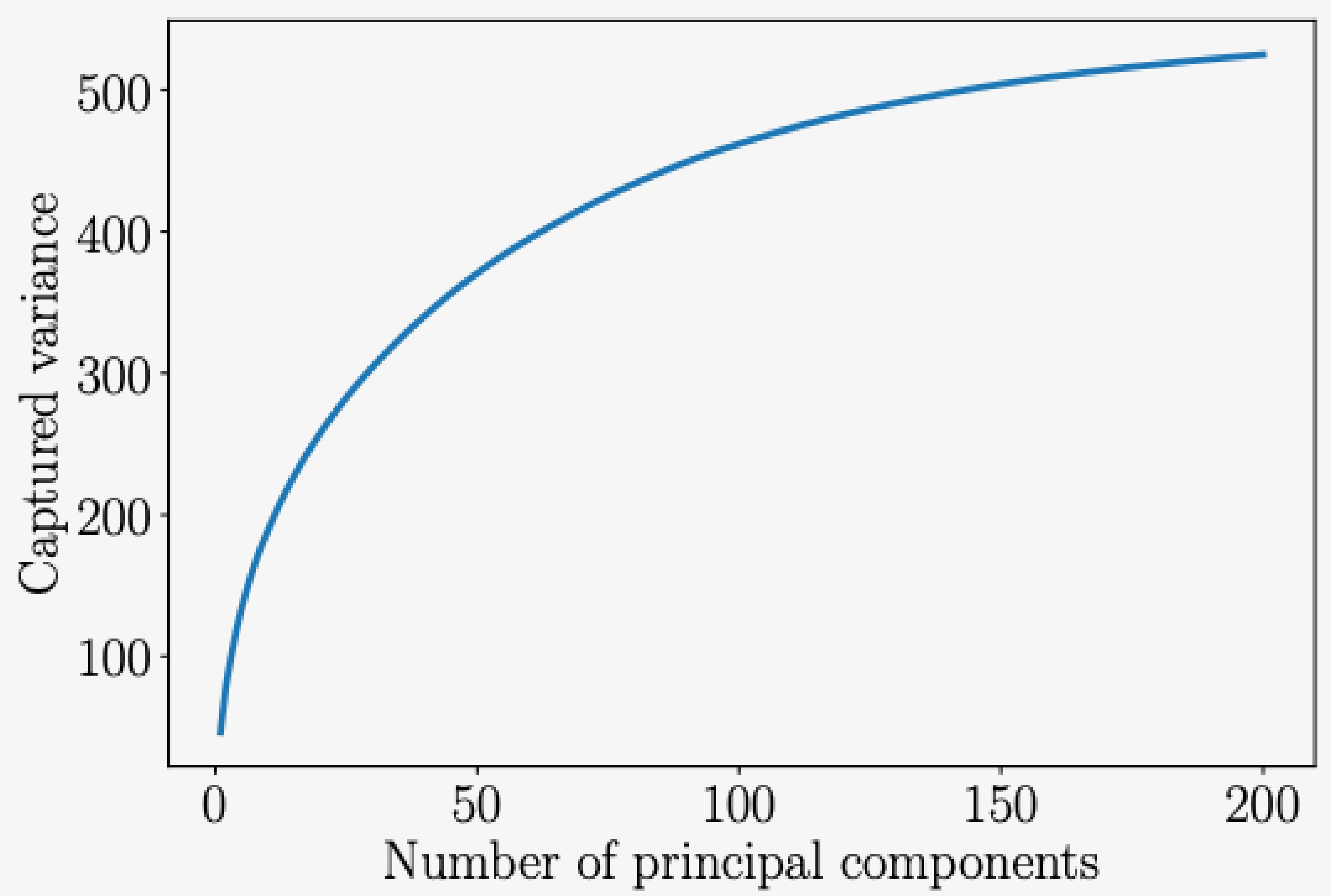
- 60,000 examples of handwritten digits 0 through 9.
- Each digit is a grayscale image of size 28×28, i.e., it contains 784 pixels.
- We can interpret every image in this dataset as a vector $x \in \mathbb{R}^{784}$



Example - captured variance



(a) Top 200 largest eigenvalues



(b) Variance captured by the principal components.

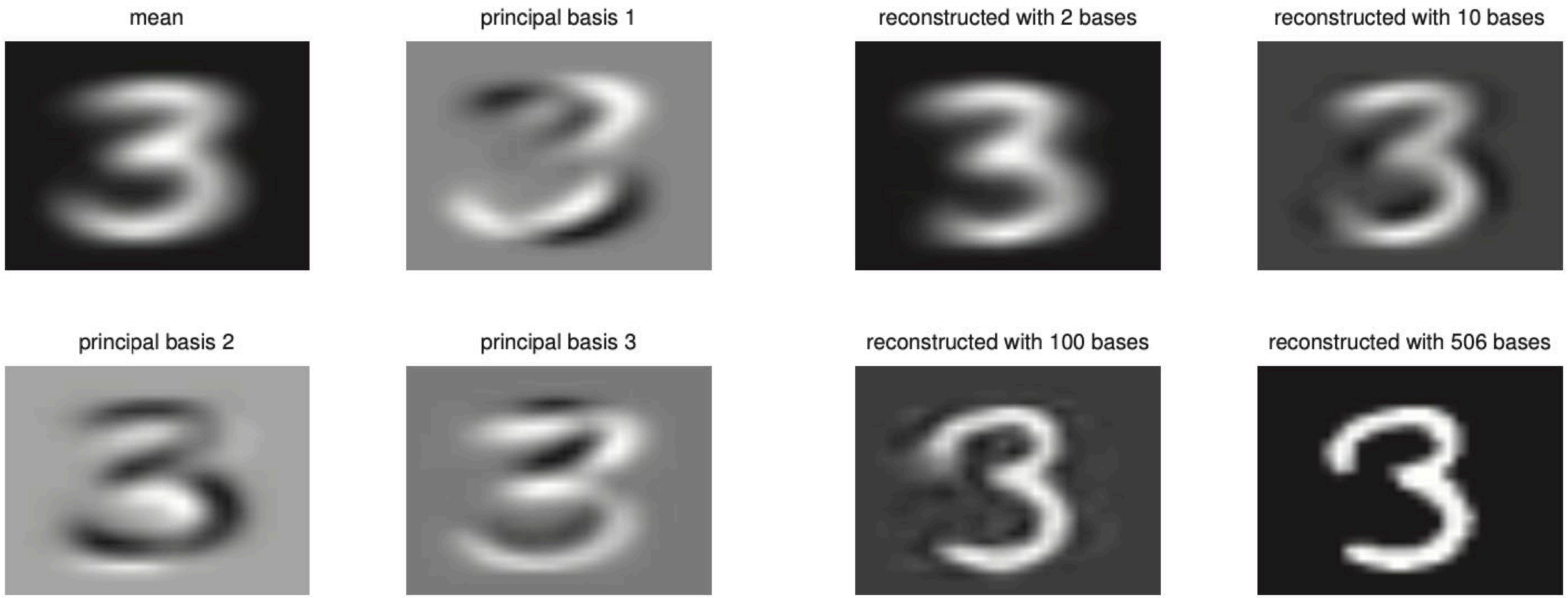
A 784-dim vector is used to represent an image

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

Example - reconstruction



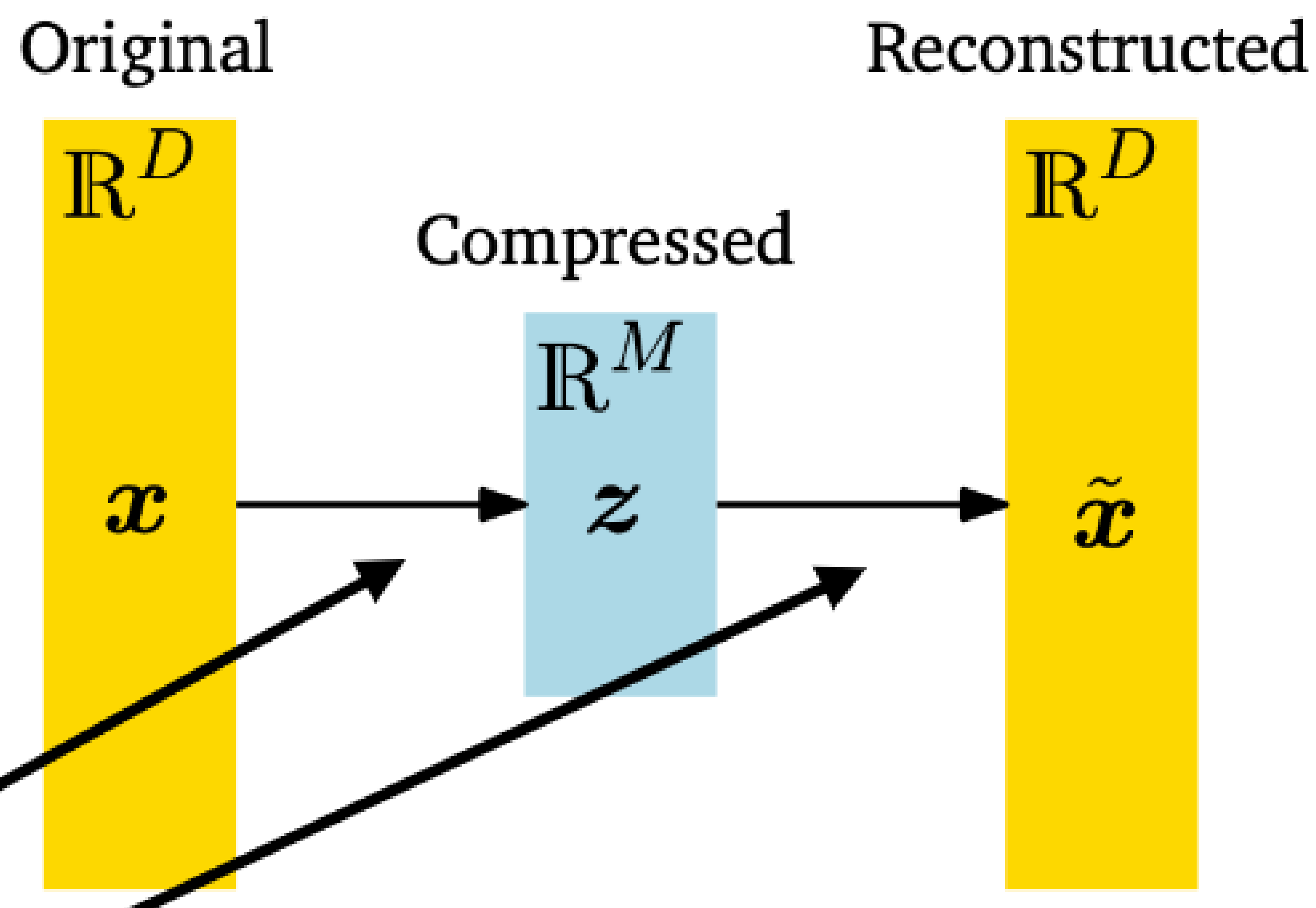
Recap: Problem setup

We consider an i.i.d. dataset $X = \{x_1, x_2, \ldots, x_N\}$, $x_n \in \mathbb{R}^D$, with mean $\mathbf{0}$ and covariance matrix $S = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^\intercal$

We assume there exists a *low-dimensional* compressed representation (code): $z_n = B^\intercal x_n$, $z_n \in \mathbb{R}^M$, $M < D$.

The projection matrix: $B = [b_1, b_2, \ldots, b_M] \in \mathbb{R}^{D \times M}$, columns are orthonormal.

Reconstruction using B: $\tilde{x}_n = Bz_n$

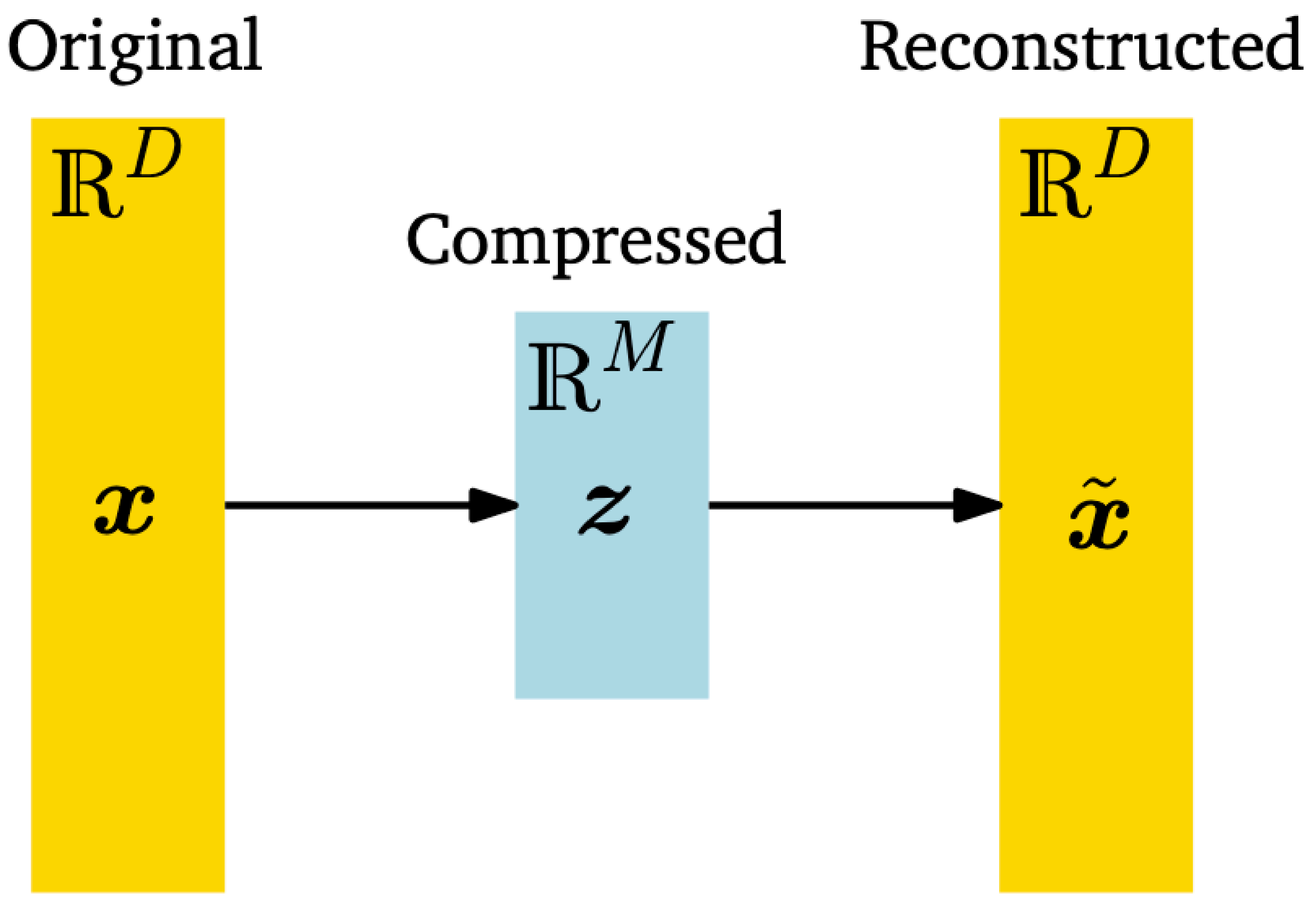


PCA: linear mappings

Goal: find z_n and the *basis vectors* $b_1, b_2, \ldots, b_M$ so that the reconstructed data are *similar* to the original data, and the compressed data retain most of the *variation* in the original data

Recap: PCA - two perspectives

Goal: find z_n and the *basis vectors* $b_1, b_2, \ldots, b_M$ so that the reconstructed data are *similar* to the original data, and the compressed data retain most of the *variation* in the original data.



Answer: Two approaches

- Search for B that **maximises** the **variance** of the low-dimensional representations [analysis/max var perspective]

  Variance of z: $\mathbb{V}_z[z] = \mathbb{V}_x[B^\intercal x]$

- Search for B and z that minimises the reconstruction loss [synthesis/projection perspective]

Both give *identical* solutions!

PCA: linear mappings

$z_n = B^\intercal x_n, z_n \in \mathbb{R}^M, M < D$

$\tilde{x}_n = B z_n$

Recap

Goal: To find an M-dimensional subspace of $\mathbb{R}^D$ that retains as much information as possible

Solution: We choose the columns of $B = \left[b_1, b_2, \ldots, b_M\right] \in \mathbb{R}^{D \times M}$ as the M eigenvectors of the data covariance matrix S that are associated with the M largest eigenvalues.

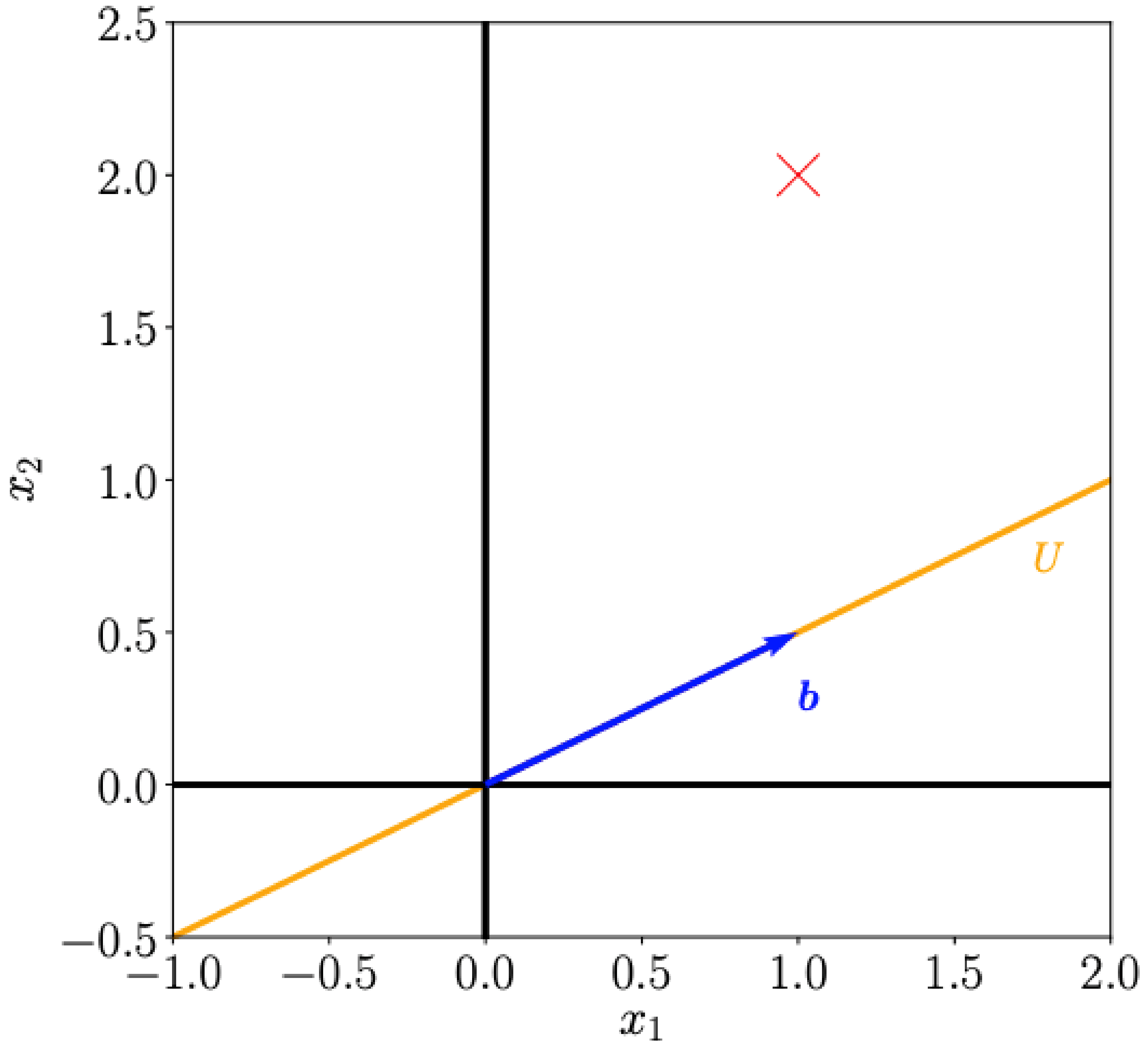
Captured variance: The maximum amount of variance PCA can capture with the first M principal components is $V_M = \sum_{m=1}^{M} \lambda_m$.

Lost variance: $J_M = \sum_{m=M+1}^{D} \lambda_m = V_D - V_M$

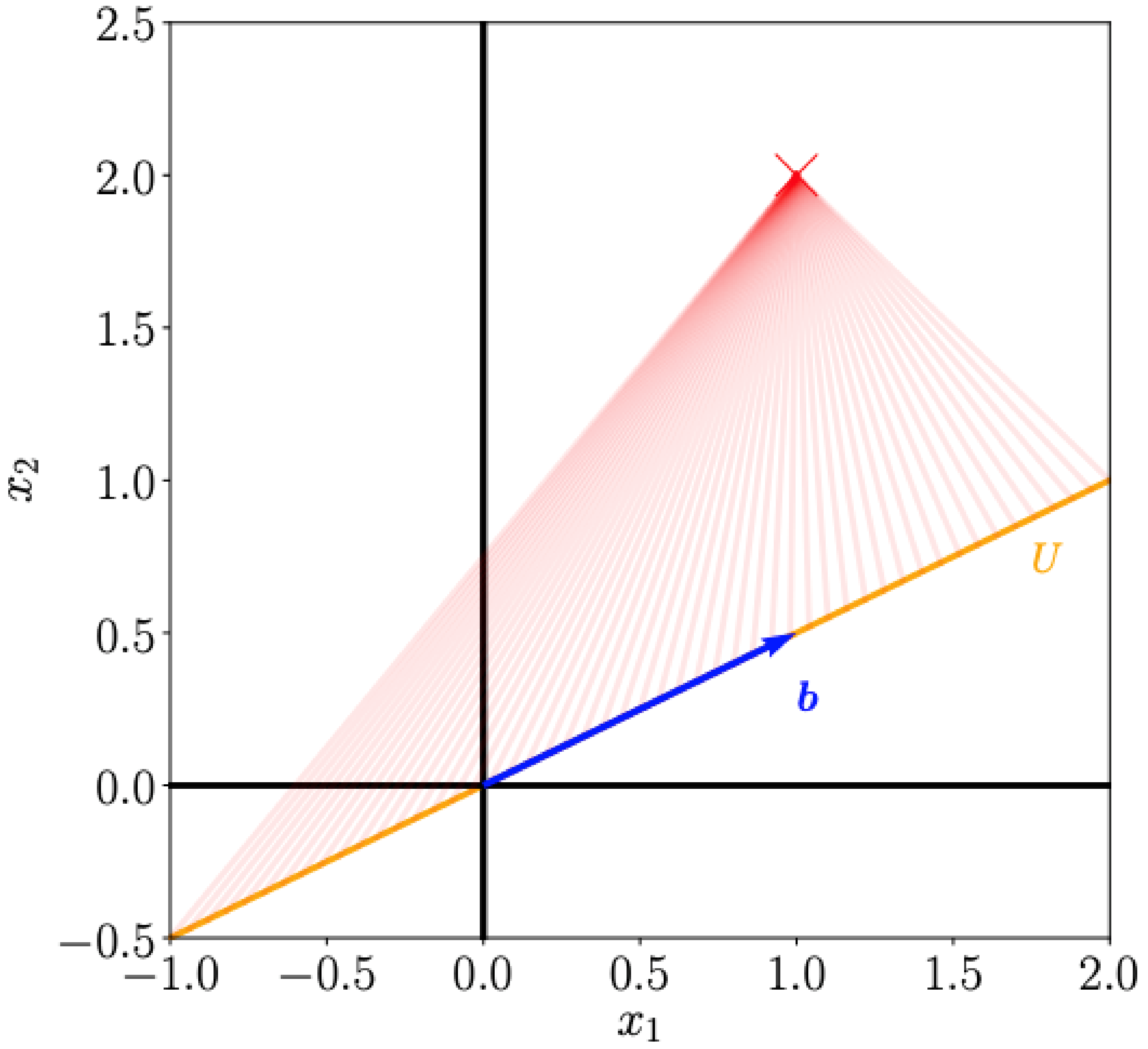
Instead of these absolute quantities, we can define the relative variance captured as V_M / V_D, and the relative variance lost by compression as $1 - V_M / V_D$.

PCA - projection perspective

Goal: Search for B and z that minimises the reconstruction loss



What low dimension subspace is best?



How to do projection onto that subspace?

We wish to project x to $\tilde{x}$ in a lower-dimensional subspace, such that $\tilde{x}$ is similar to the original data point . That is, we minimise the (Euclidean) distance between the projection and the original data point.

PCA - previous slide in maths

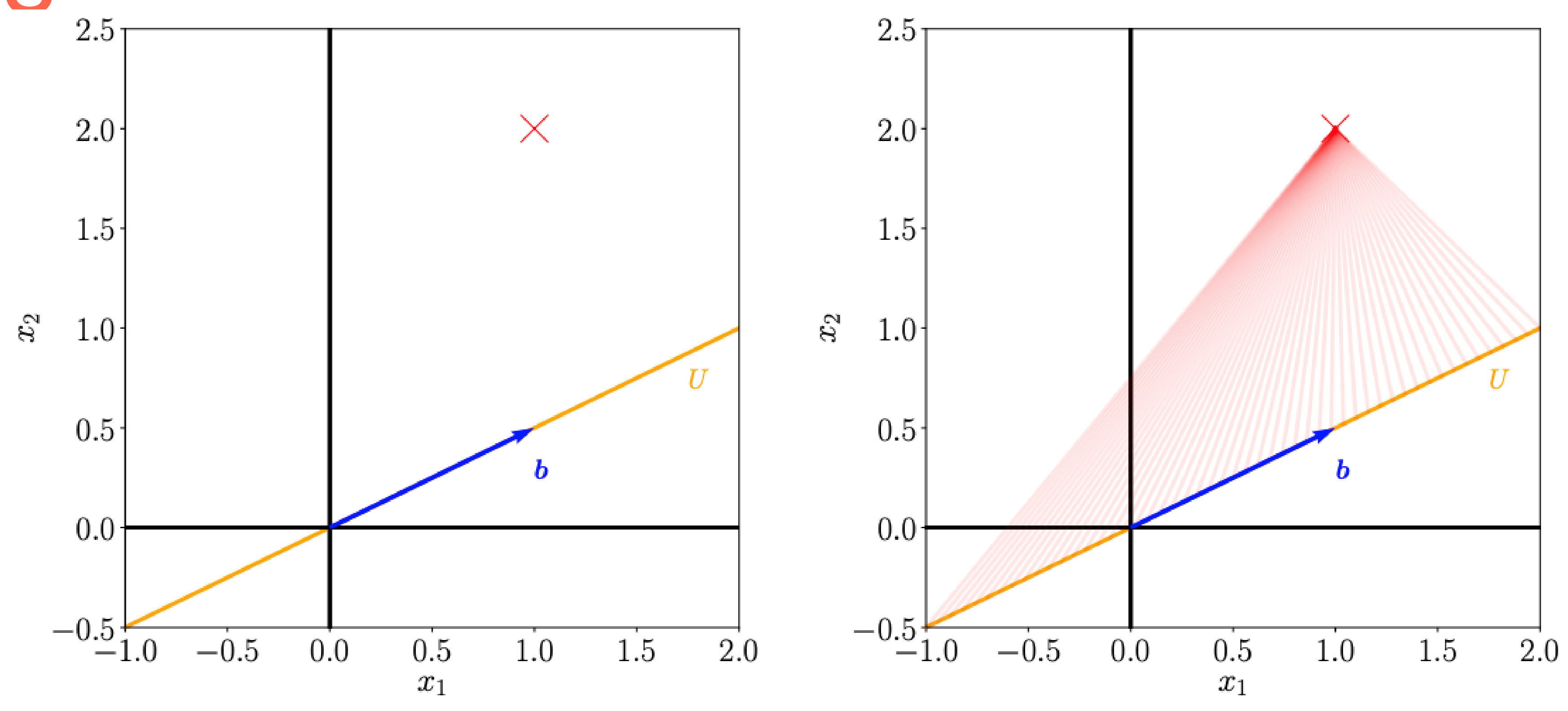
Goal: Search for B and z that minimises the reconstruction loss

Given an orthonormal basis $\{b_1, b_2, \dots, b_D\}$ of $\mathbb{R}^D$,

any $x_n \in \mathbb{R}^D$ can be written as a linear combination

of the basis vectors of $\mathbb{R}^D$: $x_n = \sum_{d=1}^{D} \epsilon_{nd} b_d = \sum_{m=1}^{M} \epsilon_{nm} b_m + \sum_{j=M+1}^{D} \epsilon_{nj} b_j$

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



U has orthonormal basis $b_1, \dots, b_M$

Called **principal subspace**

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

$z_n = [z_{1n}, \dots, z_{Mn}]^\intercal \in \mathbb{R}^M$

coordinate of $\tilde{x}$ wrt to the basis of U

Projection of x_n onto U $\quad \tilde{x}_n = \sum_{m=1}^{M} z_{mn} b_m = B z_n \in U$

Objective: minimising $J_M(B, \{z_n\}_{n=1}^{N}) = \frac{1}{N} \sum_{n=1}^{N} \| x_n - \tilde{x}_n \|_2^2$

find the orthonormal basis of the principal subspace B and the coordinates z

PCA - projection perspective

Objective: minimising $J_M(B, \{z_n\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^{N} \| x_n - \tilde{x}_n \|_2^2$ find the orthonormal basis of the principal subspace B and the coordinates z

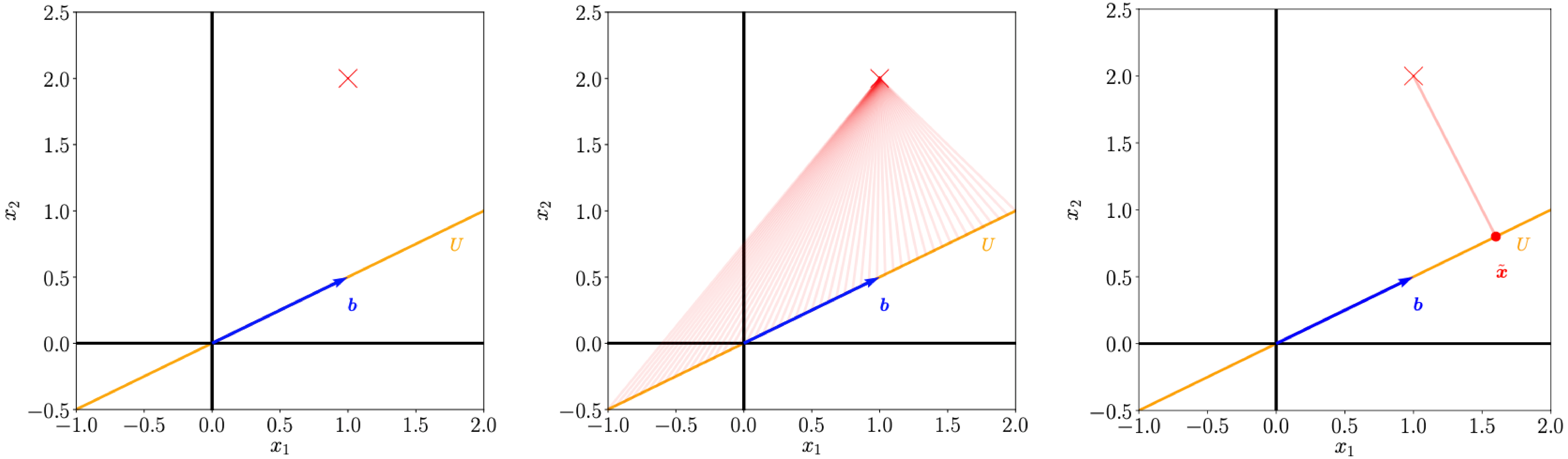
$$\tilde{x}_n = \sum_{m=1}^{M} z_{mn} b_m = B z_n \in U$$

Strategy: find the optimal coordinates given the basis, then find the optimal basis

PCA - finding optimal coordinates

Objective: minimising $J_M(B, \{z_n\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^{N} \| x_n - \tilde{x}_n \|_2^2$

$$\tilde{x}_n = \sum_{m=1}^{M} z_{mn} b_m = B z_n \in U$$



The optimal coordinates z_{in} are the coordinates of the orthogonal projection of the original data point x_n onto the one-dimensional subspace that is spanned by b_i. [see handwritten notes]

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

The coordinates of $\tilde{x}_n$ with respect to the basis $(b_1, \dots, b_M)$ are the coordinates of the orthogonal projection of x_n onto the principal subspace.

Recap - Analytic geometry - Orthogonal Projections - Week 3 L1 - Slides 12-26

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.

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

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

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 $\tilde{\boldsymbol{x}} \in \mathbb{R}^D$, 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

PCA - finding basis of principal subspace

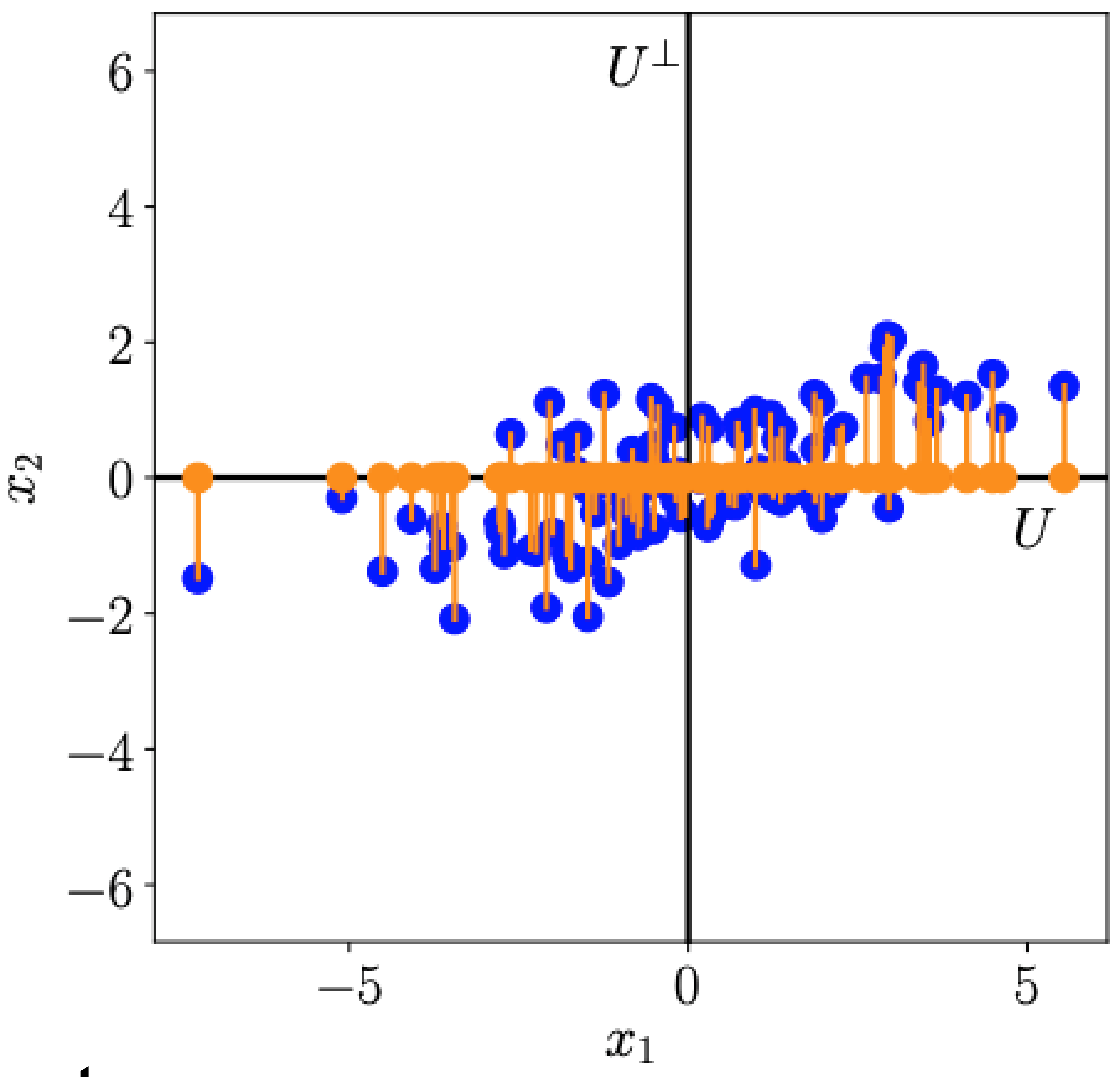
$$\tilde{x}_n = \sum_{m=1}^{M} z_{mn} b_m = B z_n \in U$$

Objective: minimising $J_M(B, \{z_n\}_{n=1}^{N}) = \frac{1}{N} \sum_{n=1}^{N} \| x_n - \tilde{x}_n \|_2^2$

Remember: The coordinates of $\tilde{x}_n$ with respect to the basis $(b_1, \dots, b_M)$ are the coordinates of the orthogonal projection of x_n onto the principal subspace.

Strategy:

+ Write down the displacement vector $x_n - \tilde{x}_n$
+ Minimising loss = minimising the variance of the data when projected onto the subspace we ignore, i.e. the orthogonal complement of the principal subspace
+ Select the smallest $D - M$ eigenvalues and corresponding eigenvectors as the basis of the orthogonal complement of the principal subspace. Equivalent to selecting largest M to construct the principal subspace (aka max variance perspective)

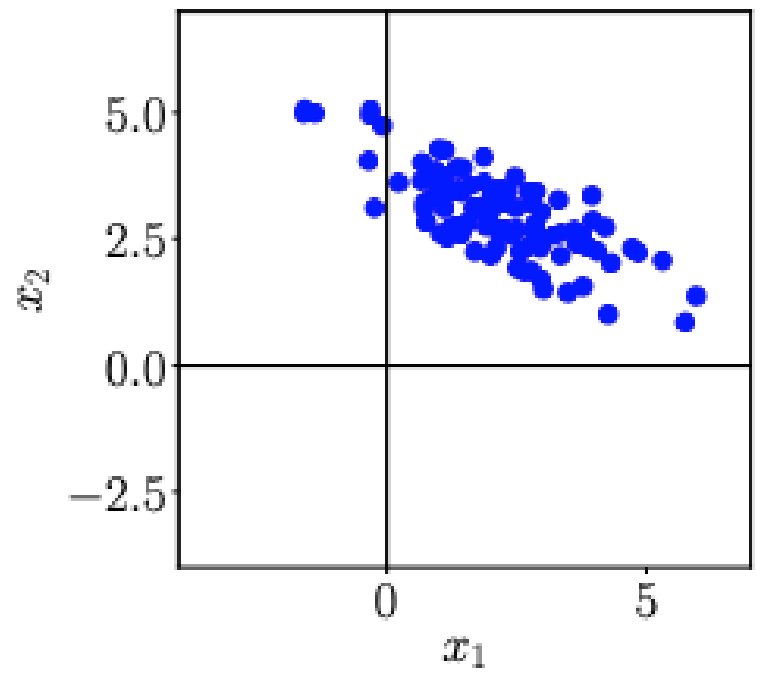


PCA in high dimensions

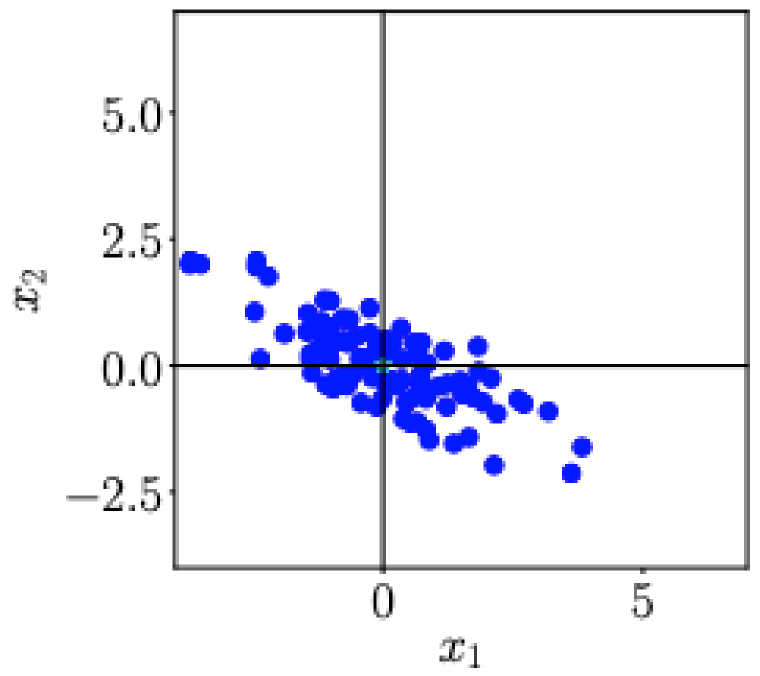
Covariance matrix: $S = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^{\intercal}, S \in \mathbb{R}^{D\times D}$

Eigendecomposition has cubic complexity $\mathcal{O}(D^3)$, expensive for large D

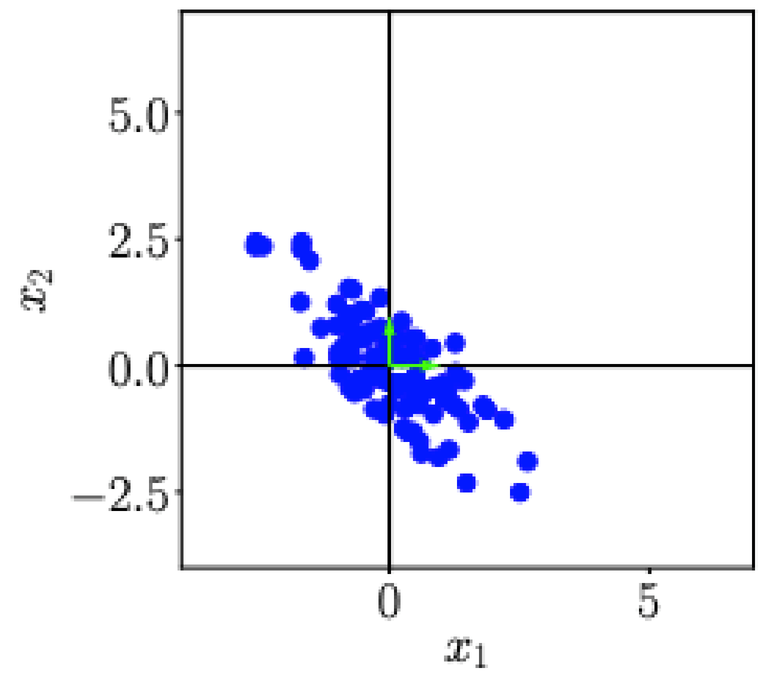
PCA in practice



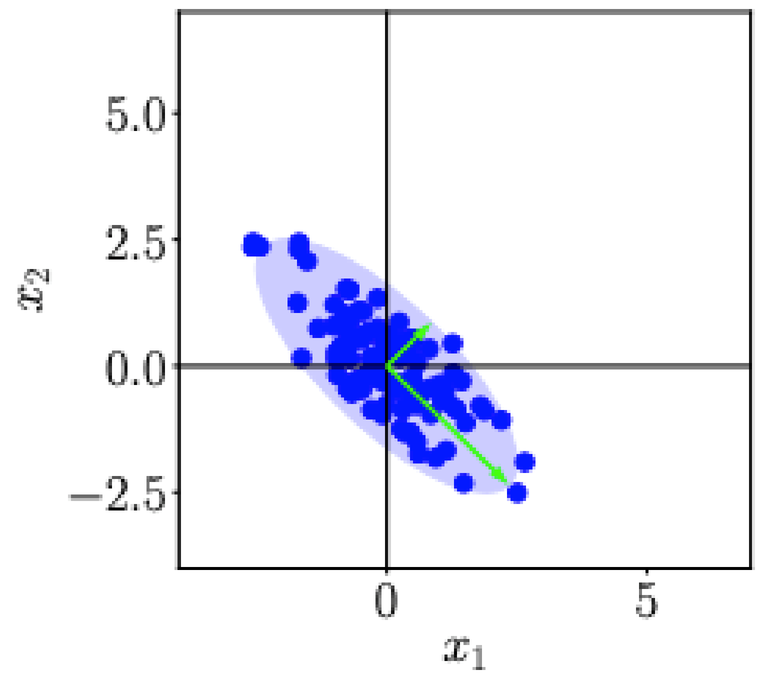
(a) Original dataset.



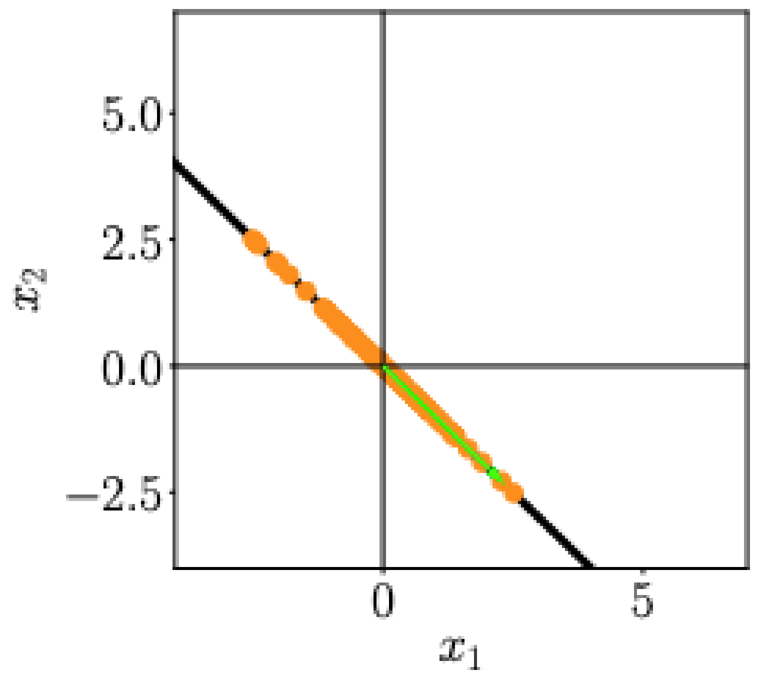
(b) Step 1: Centering by subtracting the mean from each data point.



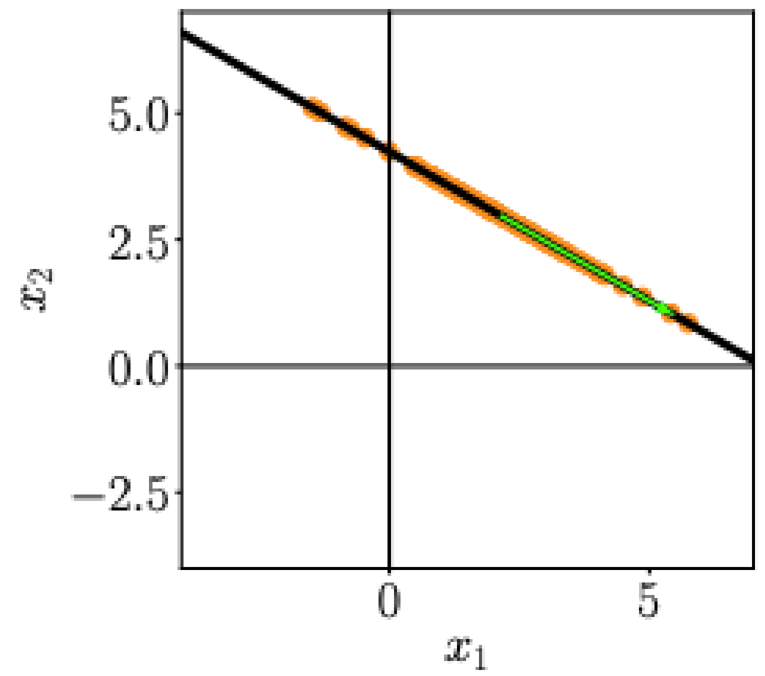
(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.



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



(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).

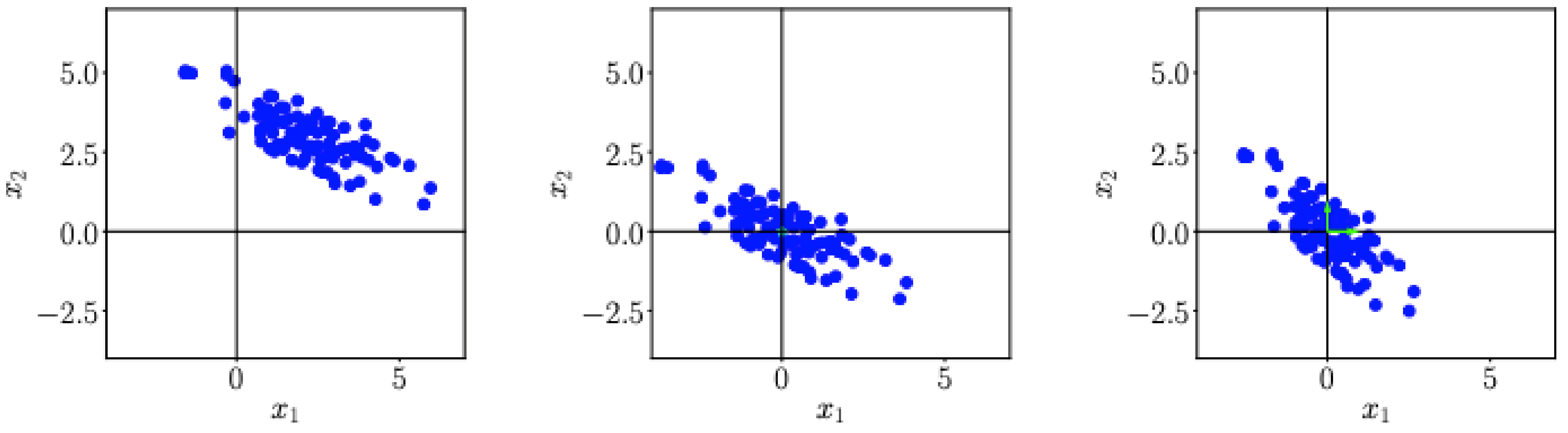
eigendecomposition

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. Standardisation

Divide the data points by the standard deviation σ_d of the dataset for every dimension. Now the data has variance 1 along each axis.



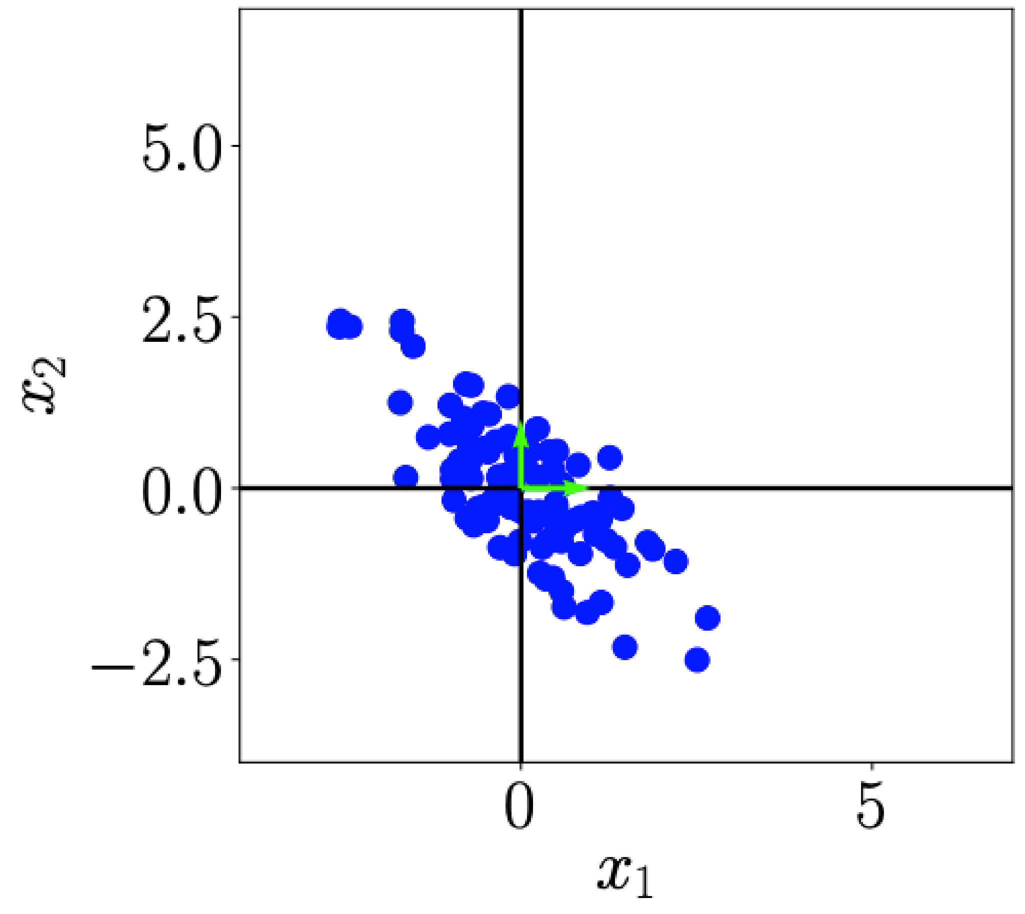
(a) Original dataset.

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

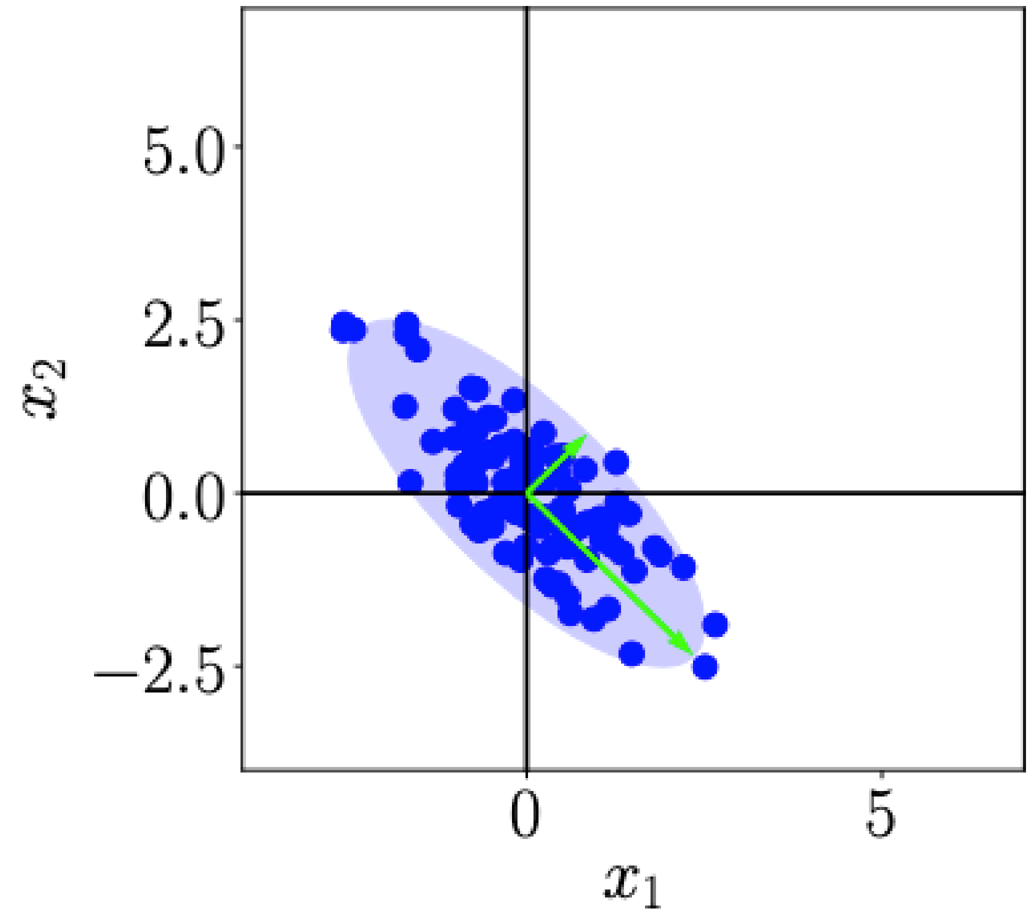
(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

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**

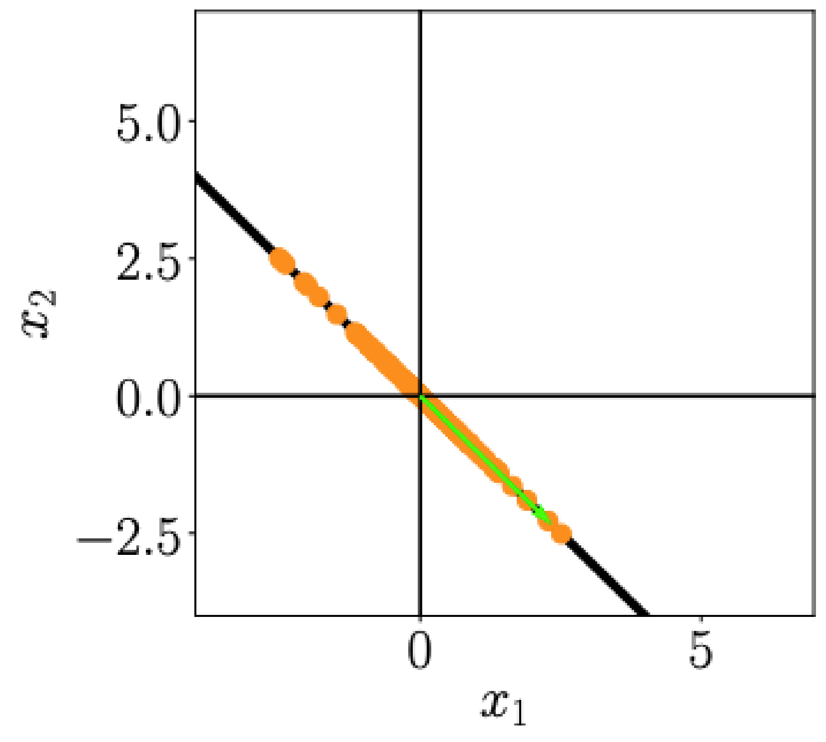
We can project any data point $\boldsymbol{x}_* \in \mathbb{R}^D$ onto the principal subspace.

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

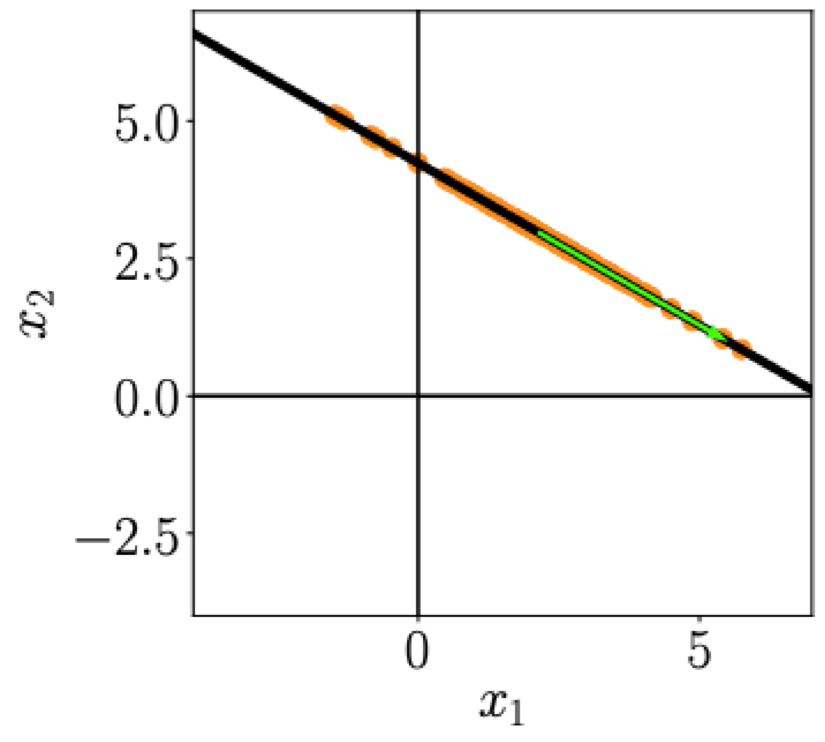
coordinates $\boldsymbol{z}_* = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}_*$ with respect to the basis of the principal subspace. Here, $\boldsymbol{B}$ is the matrix that contains the eigenvectors that are associated with the largest eigenvalues of the data covariance matrix as columns.

5. Rescaling data

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.



(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).