Dimensionality Reduction

Lecture 12

The Curse of Dimensionality

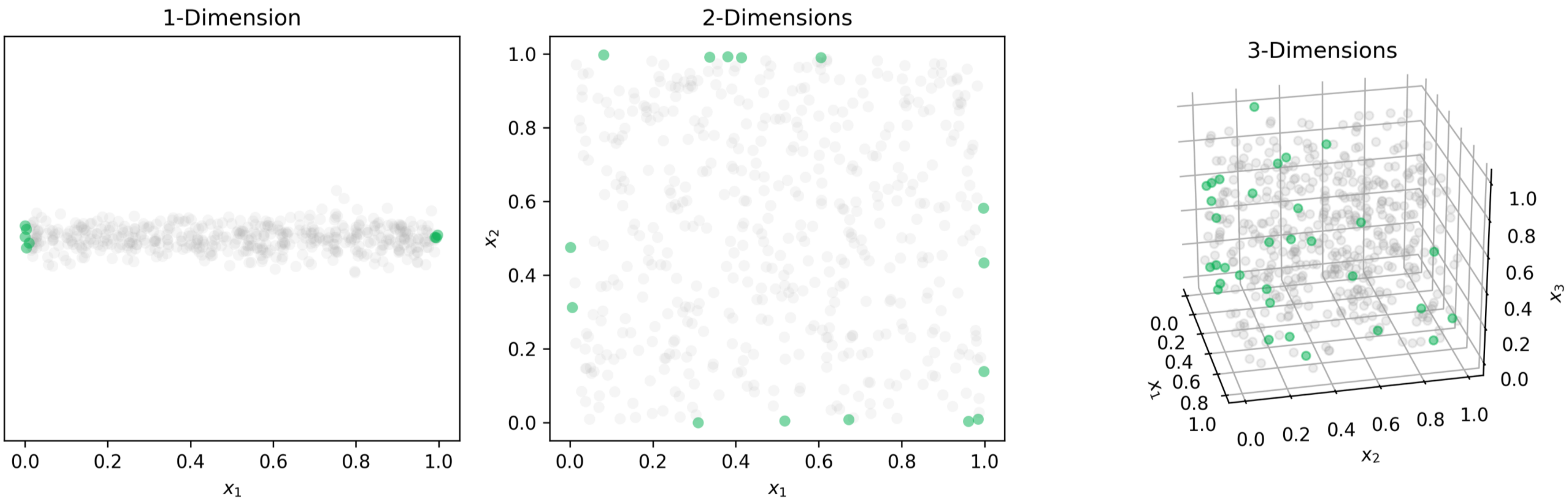
Challenge 1

In high dimensions, data become sparse

(increasing the risk of overfitting)

Random data points in a unit hypercube...

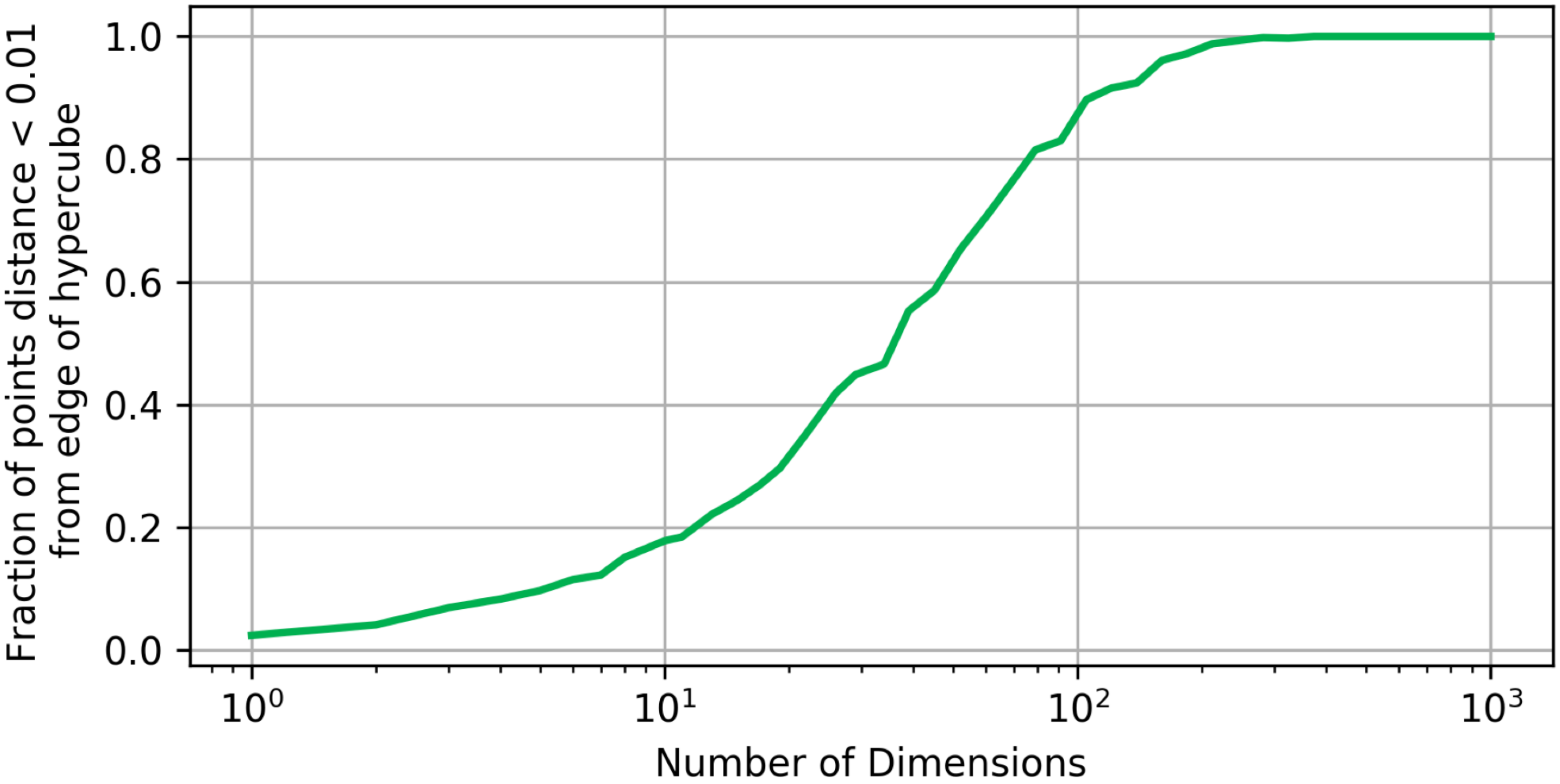
- Data point is a distance < 0.01 units from the edge of a unit hypercube
- All other data



Fraction of edge data =

0.016	0.030	0.064

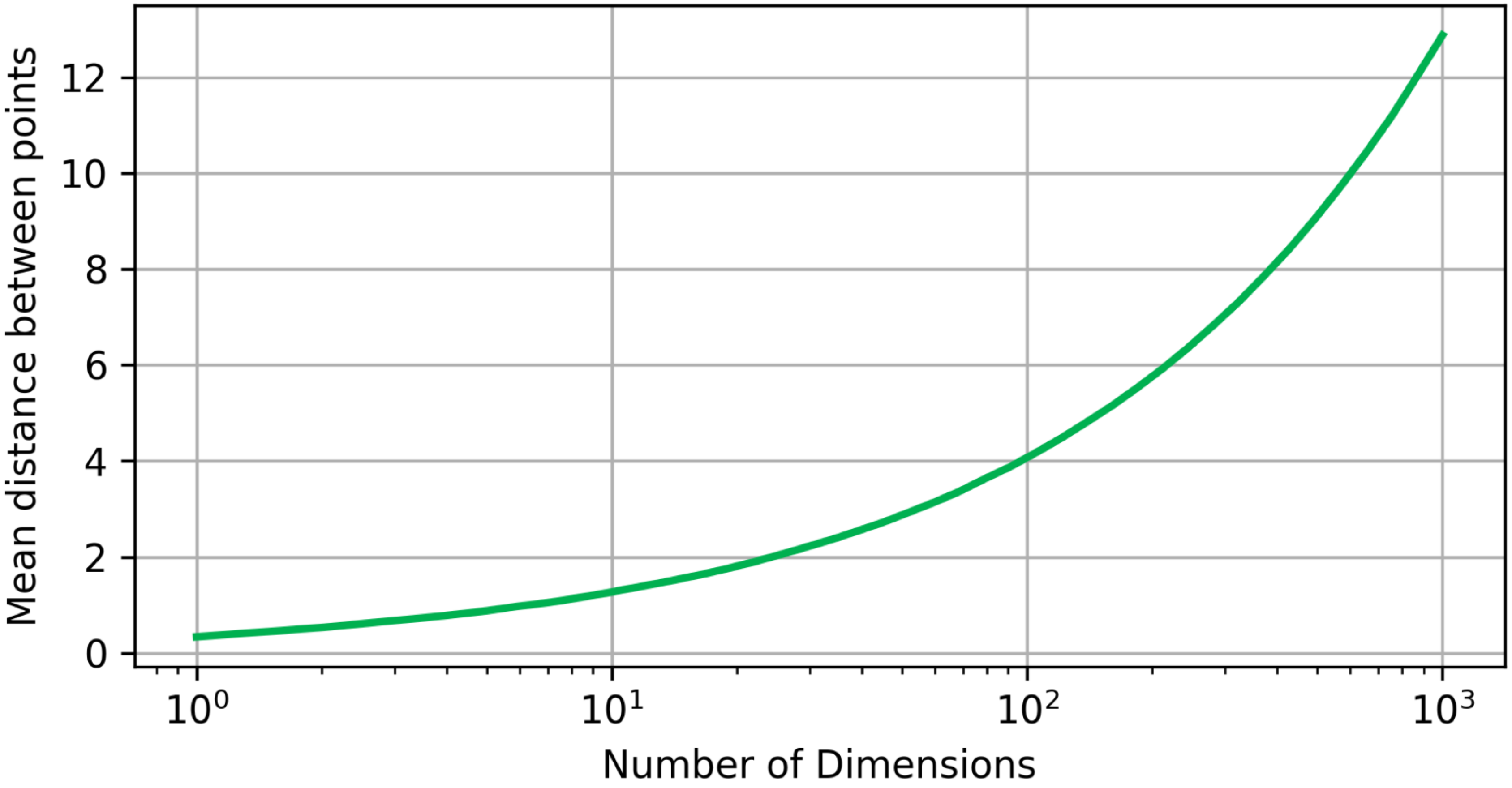
In high dimensions…



…nearly all of the high dimensional space is **far away from the center**

Note: figures constructed using 1,000 random points

In high dimensions…



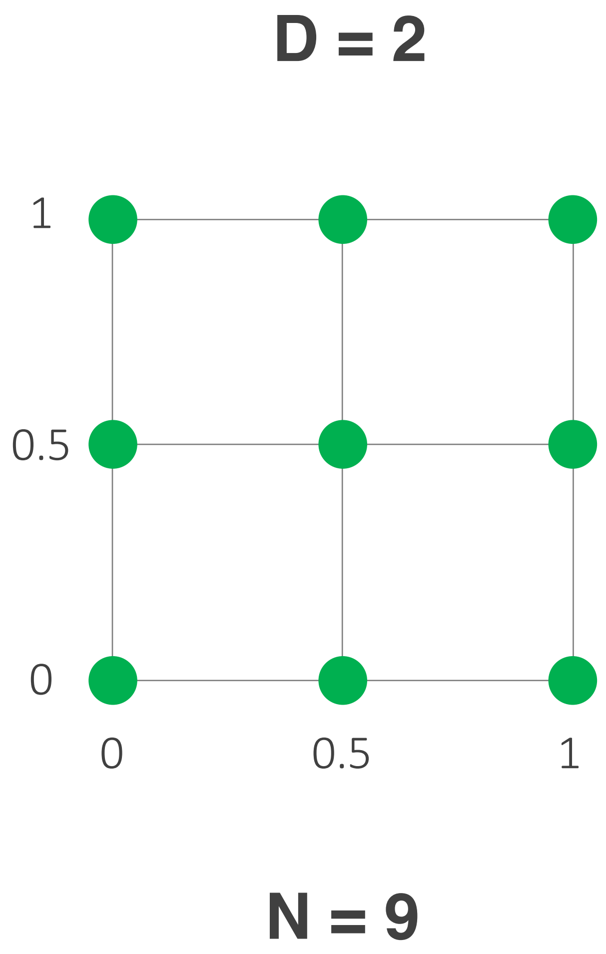
…data become sparse

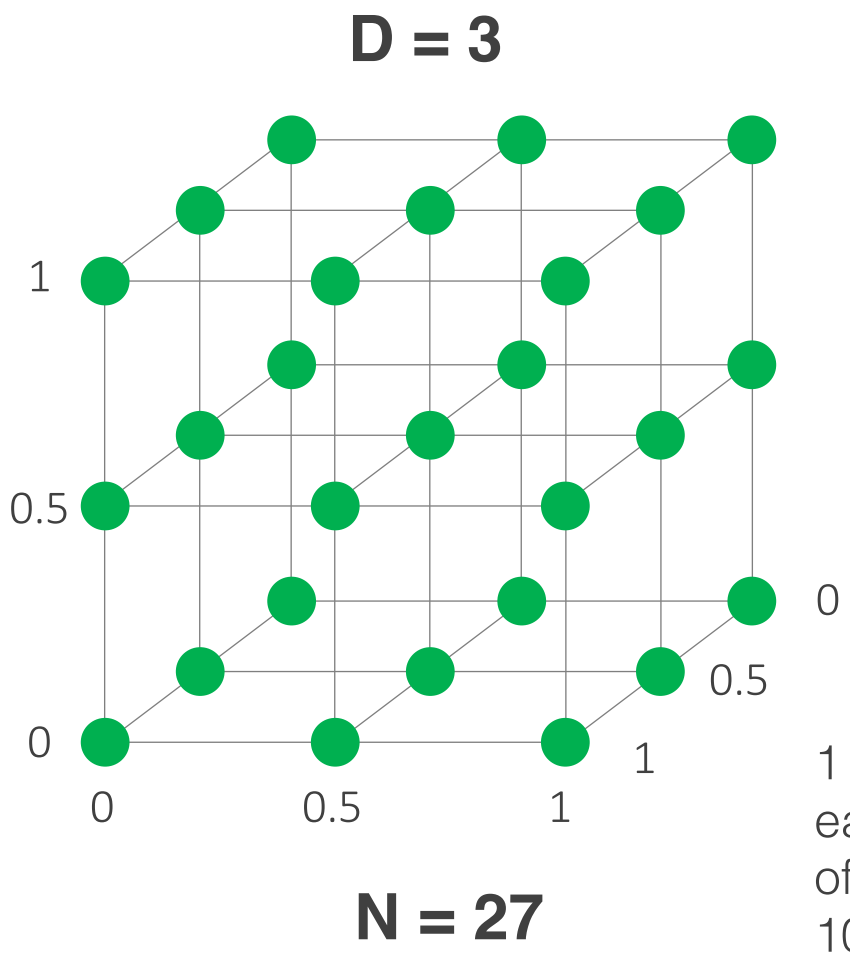
Note: figures constructed using 1,000 random points

Challenge 2

Much more data are needed for sampling higher dimensional spaces

Sample a unit hypercube on a grid spaced at intervals of 0.5





Dim (D)	Samples (N)
4	81
5	243
10	59,049
100	5.2×10^{47}
300	$\mathbf{1.4 \times 10^{143}}$

1 million Googles, each with 20 exabytes of data would only be 10^{25} bytes

…it takes more data to learn in high dimensional spaces

Dimensionality Reduction

Benefits:

Simplified data processing
Reduced redundancy of features
Improved numerical stability due to removed correlations

Approaches:

Feature selection (subset selection)
Feature extraction (including some regularization techniques)

Popular approach:

Principal Components Analysis (PCA)

PCA

Before you begin: Normalize the data!

For each feature, subtract the mean and divide by the standard deviation

Normalized feature

Raw, unscaled feature

$$x = \frac{x' - \overline{x'}}{\sigma_{x'}}$$

$$\begin{array}{cc} & \text{columns = features} \\ \boldsymbol{X} = & \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix} \text{rows = observations} \end{array}$$

We normalize each of the columns

Principal components analysis

a.k.a. the Karhunen–Loève Transform

Transform the data from a high dimensional space to a lower dimensional subspace, while minimizing the projection error

$$A \qquad \boldsymbol{x} \qquad \rightarrow \qquad \boldsymbol{z}$$

$$\mathbb{R}^{M\times D} \qquad \mathbb{R}^{D} \qquad \mathbb{R}^{M} \qquad M < D$$

$$\begin{bmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{M1} & \cdots & u_{MD} \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_D \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_M \end{bmatrix}$$

linear transformation matrix

sample of data in original D-dimensional space

(this is one of N observations)

Transformed data in M-dimensional (lower dimensional) subspace

Principal components analysis

$$A$$

$$\begin{bmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{M1} & \cdots & u_{MD} \end{bmatrix} = \begin{bmatrix} - \boldsymbol{u}_1^T - \\ \vdots \\ - \boldsymbol{u}_M^T - \end{bmatrix}$$

linear transformation matrix

Each $\boldsymbol{u}_i$ vector represents

The i^{th} principal component:

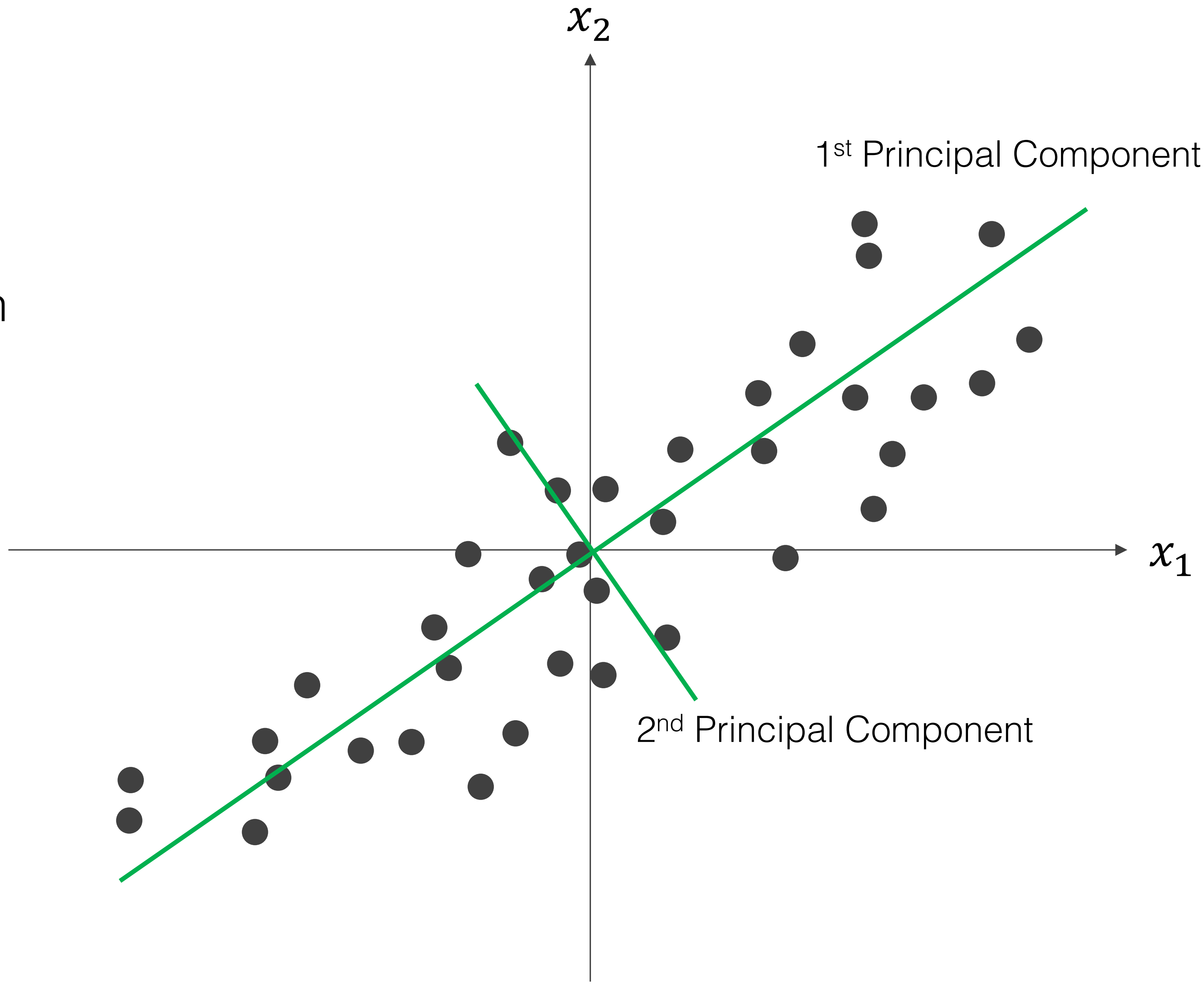
$$z_i = \boldsymbol{u}_i^T \boldsymbol{x}$$

Since only direction matters, we assume the $\boldsymbol{u}_i$ are unit vectors

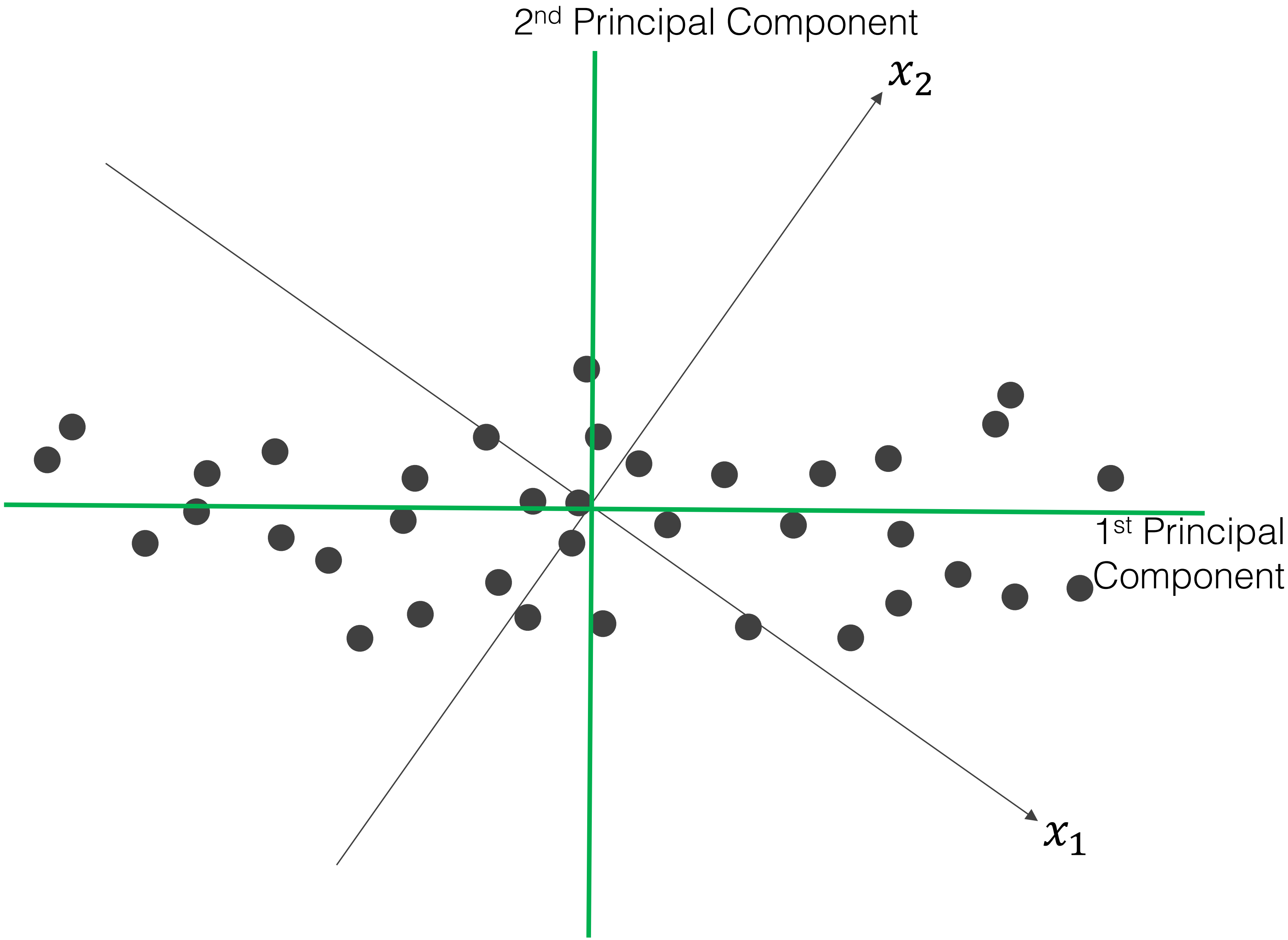
$$\boldsymbol{u}_i^T \boldsymbol{u}_i = 1$$

Principal Components

Maximum variance formulation



Reprojected Data onto Principal Components



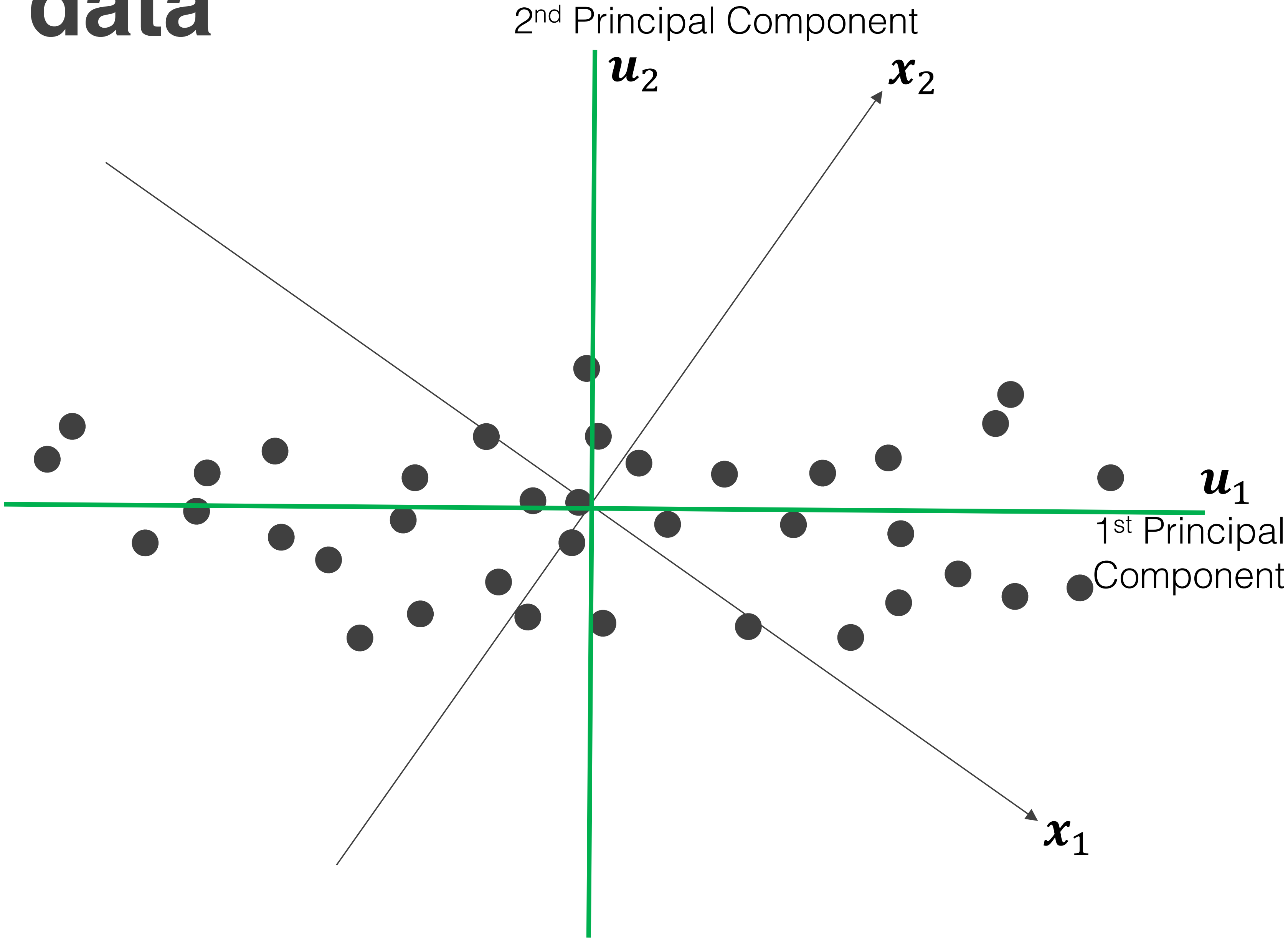
Any point $\boldsymbol{x}_j$ can be represented as a combination of the principle components

$$\boldsymbol{x}_j = \sum_{i=1}^{D} \beta_{ji} \boldsymbol{u}_i$$

The $\boldsymbol{u}_i$'s are an orthogonal basis for the space $\mathbb{R}^D$

Approximating data with principal components

$$\boldsymbol{x}_j = \sum_{i=1}^{D} \underbrace{(\boldsymbol{x}_j^T \boldsymbol{u}_i)}_{\text{scalar projection}} \underbrace{\boldsymbol{u}_i}_{\text{principal component direction}}$$



PCA

We want to **maximize the variance** of the projected data

Let's start by finding the **unit vector in the direction of greatest variation** in the dataset

Here the magnitude is unimportant, but the direction matters

We seek to project each point $\boldsymbol{x}_i$ onto a unit PC vector. $z_i = \boldsymbol{u}_i^T \boldsymbol{x}_i$

PCA Example: find the first principal component

Mean of the data:

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$$

$\boldsymbol{x}_i$ $[D \times 1]$

The projected mean of the data:

$$\bar{z} = \boldsymbol{u}_i^T \, \bar{\boldsymbol{x}}$$

We can compute the sample variance as:

$$\sigma_z^2 = \frac{1}{N}\sum_{i=1}^{N} (z_i - \bar{z})^2$$

[scalar]

$$\sigma_z^2 = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{u}_1^T \boldsymbol{x}_i - \boldsymbol{u}_1^T \bar{\boldsymbol{x}})^2$$

PCA Example: find the first principal component

We can compute the sample variance as:

$$\sigma_z^2 = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{u}_1^T\boldsymbol{x}_i - \boldsymbol{u}_1^T\overline{\boldsymbol{x}})^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{u}_1^T(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T\boldsymbol{u}_1$$

$$= \boldsymbol{u}_1^T\boldsymbol{\Sigma}\boldsymbol{u}_1$$

Variance of the projected data

Define:

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$$

Covariance matrix

$$\boldsymbol{u}_1^T \Sigma \boldsymbol{u}_1 = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{u}_1^T (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T \boldsymbol{u}_1$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \Sigma_{DD} \end{bmatrix}$$

$$\begin{aligned} \Sigma_{ij} &= E[(X_i - \mu_i)(X_j - \mu_j)] \\ &= cov(X_i, X_j) \end{aligned}$$

If $\mu_i = 0 \; \forall \, i$

$$\Sigma_{ij} = E[X_i X_j] = \frac{1}{N} \boldsymbol{x}_i^T \boldsymbol{x}_j$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \sigma_2^2 & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \Sigma_D^2 \end{bmatrix}$$

$$\sigma_i^2 = E[(X_i - \mu_i)^2]$$

Covariance matrix properties

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \Sigma_{12} & \cdots & \Sigma_{1D} \\ \Sigma_{21} & \sigma_2^2 & \cdots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \cdots & \Sigma_D^2 \end{bmatrix}$$

Positive semidefinite and symmetric ($\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{\mathrm{T}}$)

All eigenvalues are positive

Eigenvectors are orthogonal

If the features, $x_1, x_2, \dots, x_D$ are independent, $\boldsymbol{\Sigma}$ is diagonal because $cov(x_i, x_j) = 0$ if $i \neq j$

PCA

We want to **maximize variance** $\sigma_z^2 = \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1$

subject to $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$

We can use **Lagrange multipliers**:

Maximize $f(x)$

subject to the constraint $g(x)$

We maximize this: $L(x, \lambda) = f(x) - \lambda g(x)$

$$f(\boldsymbol{x}, \boldsymbol{u}_i) = \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1$$

$$g(\boldsymbol{x}, \boldsymbol{u}_i) = \boldsymbol{u}_1^T \boldsymbol{u}_1 - 1$$

For our case: $L(\boldsymbol{x}, \boldsymbol{u}_\mathbf{1}, \lambda) = \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1 - \lambda(\boldsymbol{u}_1^T \boldsymbol{u}_1 - 1)$

We take the derivative and set it equal to zero

PCA

$$L(\boldsymbol{x}, \boldsymbol{u}_1, \lambda) = \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1 - \lambda(\boldsymbol{u}_1^T \boldsymbol{u}_1 - 1)$$

We take the derivative with respect to $\boldsymbol{u}_1$ and set it equal to zero

$$\frac{\partial L}{\partial \boldsymbol{u}_1} = \frac{\partial}{\partial \boldsymbol{u}_1} \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1 - \frac{\partial}{\partial \boldsymbol{u}_1} \lambda(\boldsymbol{u}_1^T \boldsymbol{u}_1 - 1)$$

$$= 2\boldsymbol{\Sigma}\boldsymbol{u}_1 - 2\lambda\boldsymbol{u}_1 = 0$$ (since Σ is symmetric)

$$\boldsymbol{\Sigma}\boldsymbol{u}_1 = \lambda\boldsymbol{u}_1 \quad \rightarrow$$

$\boldsymbol{u}_1$ is an eigenvector of the covariance matrix $\boldsymbol{\Sigma}$, and λ is an eigenvalue

PCA

Since we want to maximize the variance in the projected features:

We want to maximize: $\sigma_z^2 = \boldsymbol{u}_1^T \boldsymbol{\Sigma} \boldsymbol{u}_1$

And we know that: $\boldsymbol{\Sigma} \boldsymbol{u}_1 = \lambda \boldsymbol{u}_1$

So we can write: $\sigma_z^2 = \boldsymbol{u}_1^T \lambda \boldsymbol{u}_1 = \lambda \boldsymbol{u}_1^T \boldsymbol{u}_1 = \lambda$

Therefore we choose the eigenvector that corresponds to the **largest eigenvalue**

PCA: Variance explained

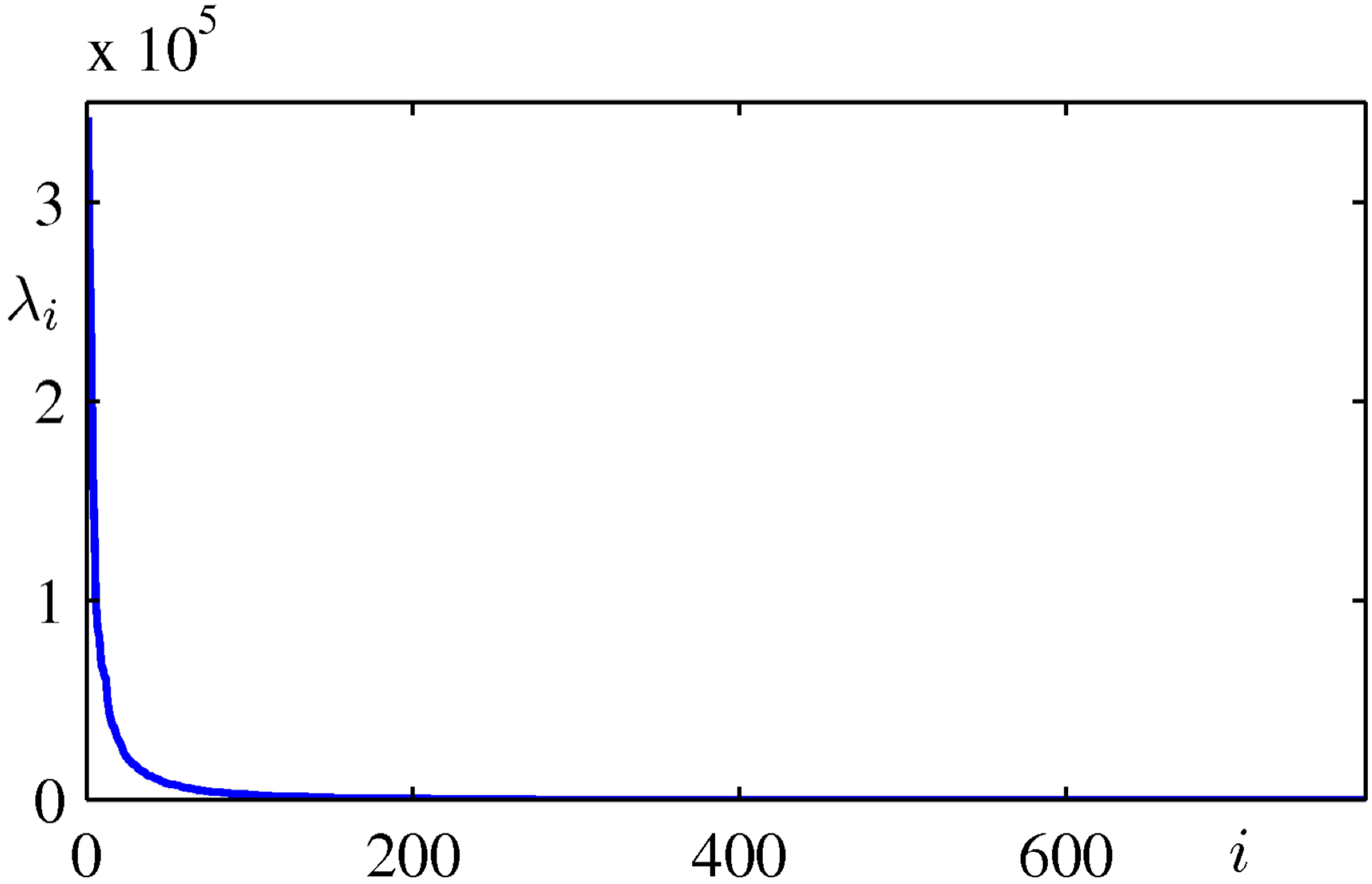
The **fraction of variance explained** $= \dfrac{\sum_{i=1}^{M} \lambda_i}{\sum_{i=1}^{D} \lambda_i}$

M = dimensionality of the subspace

D = dimensionality of the original data space

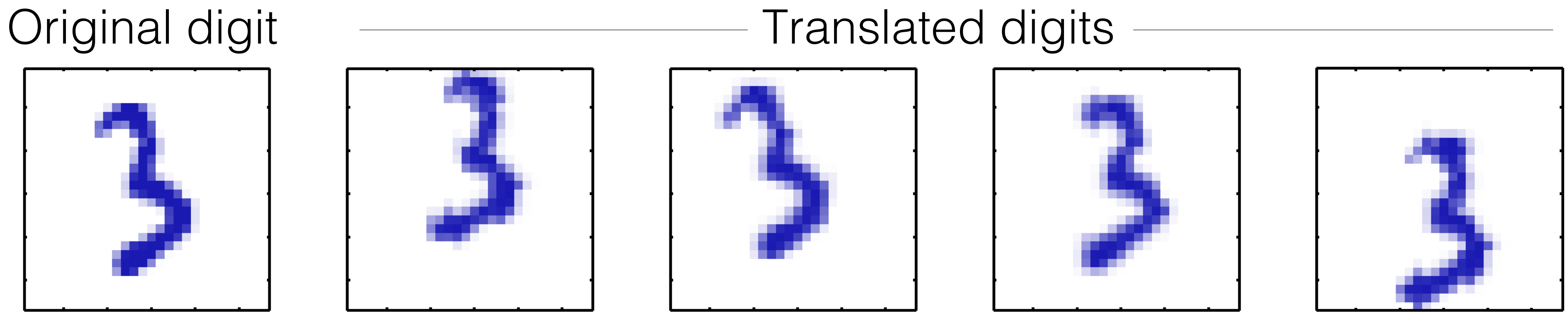
The more principle components included, the more of the variance will be represented in the projected data

Eigenvalues by principal component i



Bishop, Pattern Recognition, 2006

Example: translated digits



Types of translation:
1. Horizontal translation
2. Vertical translation
3. Rotation

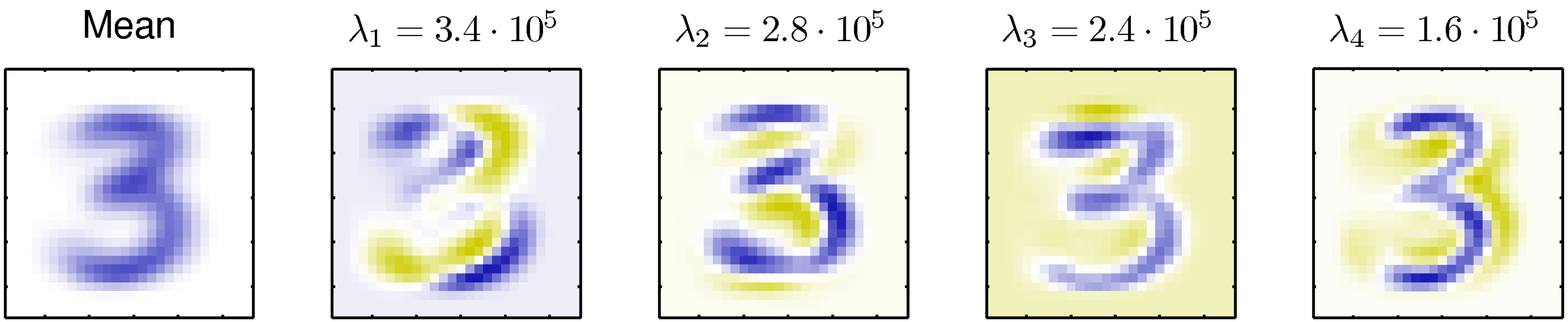
Original digits: 64 x 64 pixels

New size: 100 x 100 pixels

Bishop, Pattern Recognition, 2006

Example: translated digits

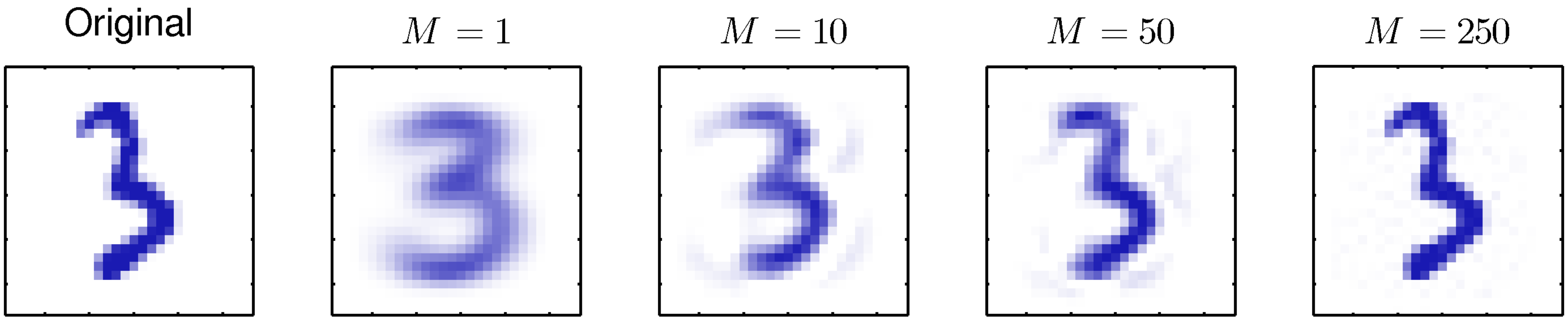
Examples of first four principle component eigenvectors and eigenvalues:



Bishop, Pattern Recognition, 2006

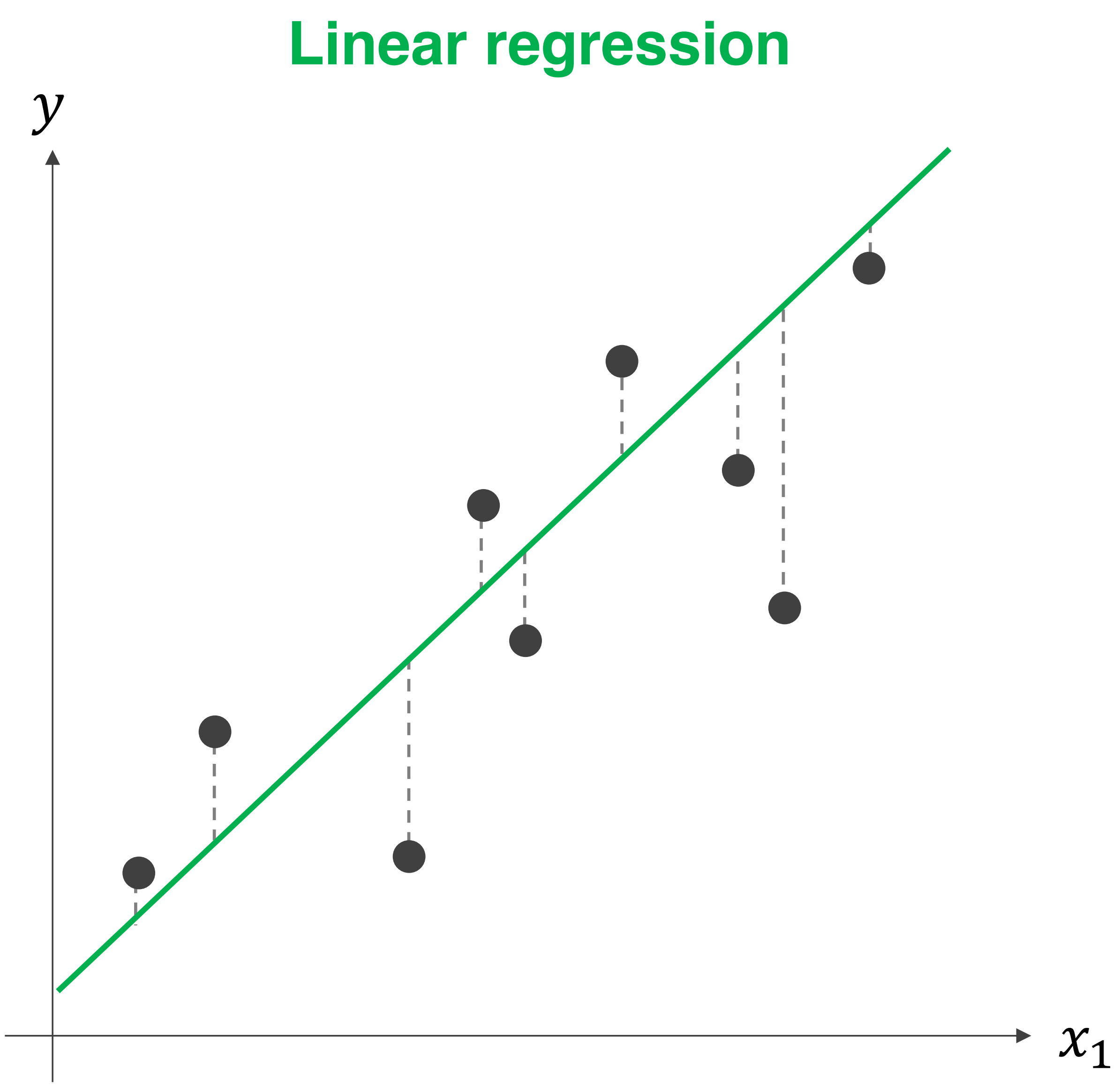
Example: translated digits

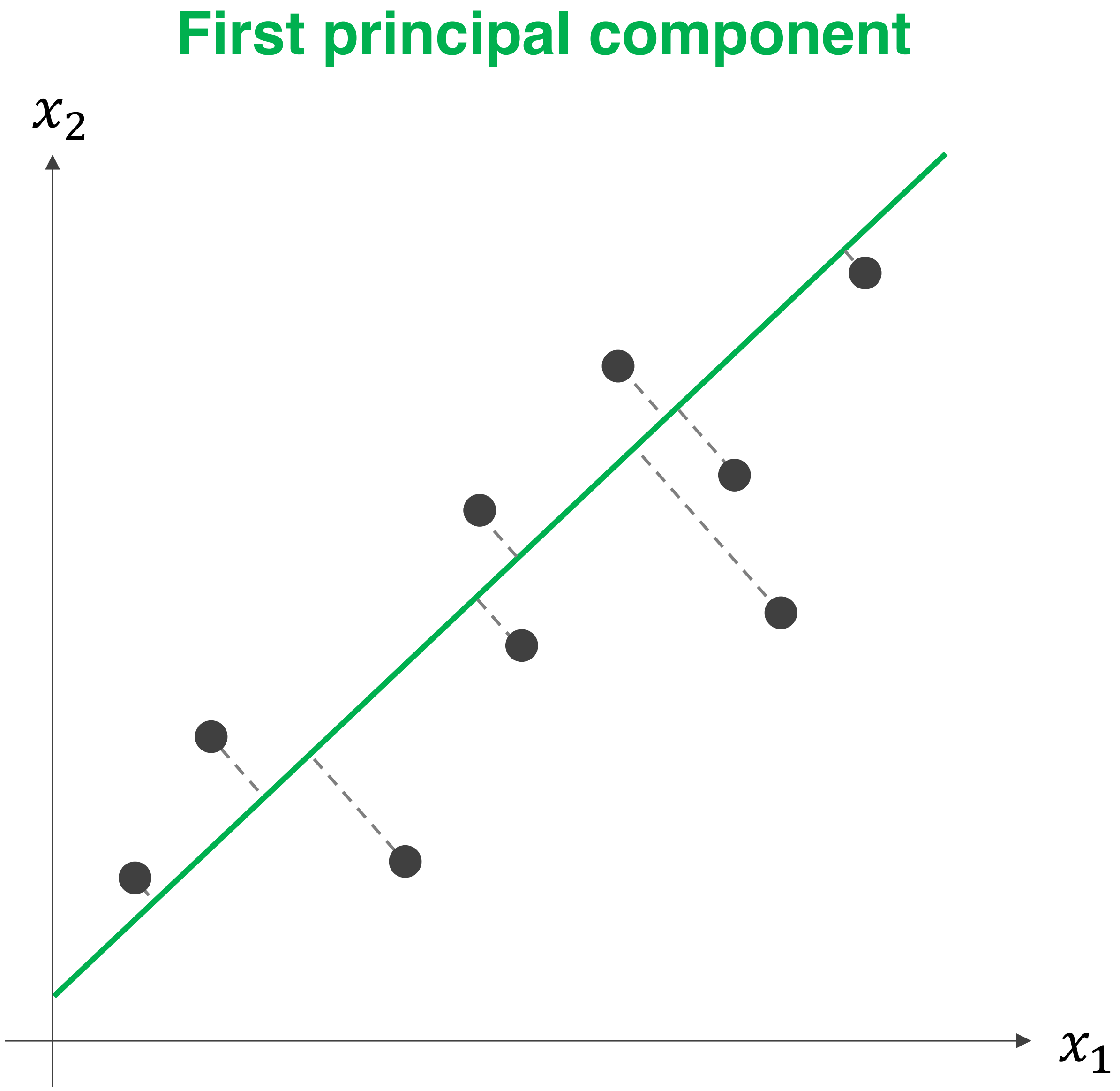
Reconstructed examples using different numbers of principal components:



Bishop, Pattern Recognition, 2006

Relationship between the objective for least squares and PCA





Extracting principal components

size | example

0 **Goal**: reduce the dimensionality of our data from D to M, where $\mathrm{M} < D$

columns = features (D)

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix}$$

rows = observations (N)

$[N \times D]$

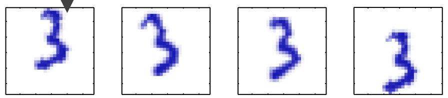
1 Normalize each feature to mean zero and a standard deviation of 1

Each observation as a vector:

$\boldsymbol{x}_i \quad i = 1, \dots, D$

$[D \times 1]$

Each pixel represents a feature



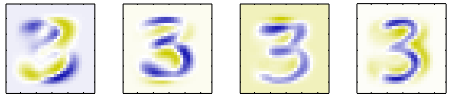
$\boldsymbol{x}_1$ $\boldsymbol{x}_2$ $\boldsymbol{x}_3$ $\boldsymbol{x}_4$

2 Determine the principal components

Calculate the eigenvectors and eigenvalues of the data covariance matrix, $\boldsymbol{\Sigma}$

Eigenvectors in descending order of their eigenvalues are the principal components

$\boldsymbol{u}_i$ eigenvectors / principal components $[D \times 1]$



$\boldsymbol{u}_1$ $\boldsymbol{u}_2$ $\boldsymbol{u}_3$ $\boldsymbol{u}_4$

λ_i eigenvalues (how much of the variance is explained) [scalar]

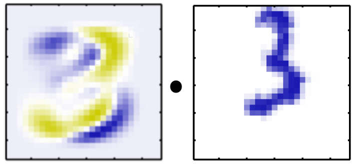
$i = 1, \dots, D$

3 Project the data features on the principal components

$$z_{ij} = \boldsymbol{u}_j^T \boldsymbol{x}_i \qquad j = 1, \dots, D \quad i = 1, \dots, N$$

[scalar]

$\boldsymbol{u}_1 \cdot \boldsymbol{x}_1 = z_{11}$



$= 2.43$

$[D \times 1]$ $[D \times 1]$ [scalar]

4 Keep the top M principal components to reduce into a lower dimension

$$\boldsymbol{A} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_M]$$

$[D \times M]$

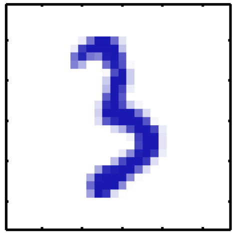
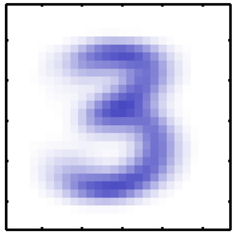
$$\boldsymbol{z}_i = \boldsymbol{A}^T \boldsymbol{x}_i \qquad i = 1, \dots, N$$

Images from Bishop, Pattern Recognition, 2006

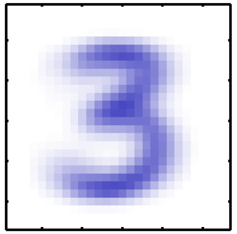
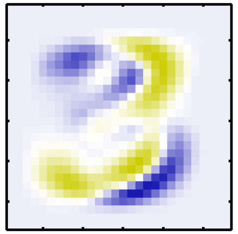
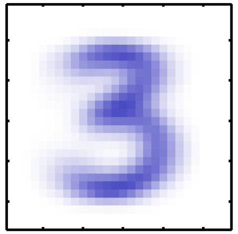
Reconstructing our data from principal components

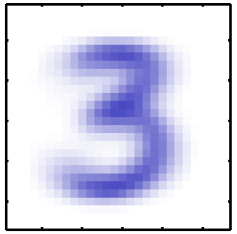
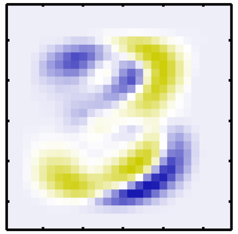
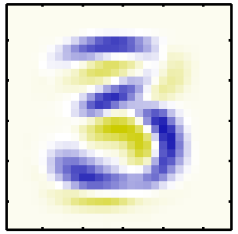
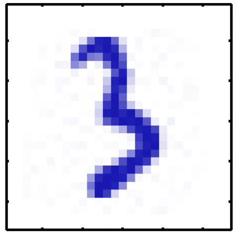
Sum the product of our projected data, $\boldsymbol{z}_i$, and our principle components

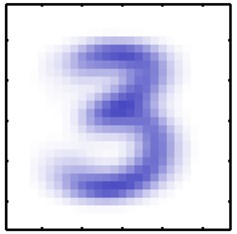
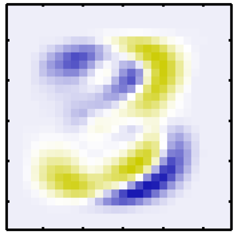
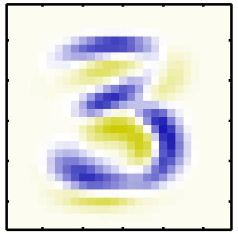
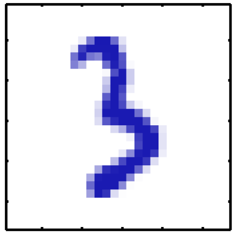
$$\hat{\boldsymbol{x}}_i = \sum_{j=1}^{M} z_{ij}\boldsymbol{u}_j$$

Example: the i[th] observation: $\boldsymbol{x}_i =$  $\bar{\boldsymbol{x}} =$ 

PCA-projected data: $\boldsymbol{z}_i = [z_{i1}, z_{i2}, \dots, z_{iM}]$

$M = 1$ $\quad \hat{\boldsymbol{x}}_i =$ $\bar{\boldsymbol{x}}$  $+ z_{i1} \cdot$ $\boldsymbol{u}_1$  $=$ 

$M = 250$ $\quad \hat{\boldsymbol{x}}_i =$ $\bar{\boldsymbol{x}}$  $+ z_{i1} \cdot$ $\boldsymbol{u}_1$  $+ z_{i2} \cdot$ $\boldsymbol{u}_2$  $+ \sum_{j=3}^{250} z_{ij}\boldsymbol{u}_j =$ 

$M = 10{,}000$ $\quad \hat{\boldsymbol{x}}_i =$ $\bar{\boldsymbol{x}}$  $+ z_{i1} \cdot$ $\boldsymbol{u}_1$  $+ z_{i2} \cdot$ $\boldsymbol{u}_2$  $+ \sum_{j=3}^{10{,}000} z_{ij}\boldsymbol{u}_j =$ 

(perfect reconstruction)

Images from Bishop, Pattern Recognition, 2006

Why PCA?

- Dimensionality reduction
- Feature extraction
- Data visualization
- Lossy data compression

Other dimensionality reduction techniques

- Kernel PCA
- Random projections
- Multidimensional scaling
- Locality sensitive hashing
- Autoencoders
- Isomap

e.g. Manifold Learning