Unsupervised learning
18.11.2024

Outline

- Hour 1
  - Quiz discussion, review of last lecture
  - Introduction to unsupervised learning:
    - Principal component analysis (PCA)
- Hour 2:
  - PCA continued
  - K-means

Review of data statistics, ex: Quiz 1 grades

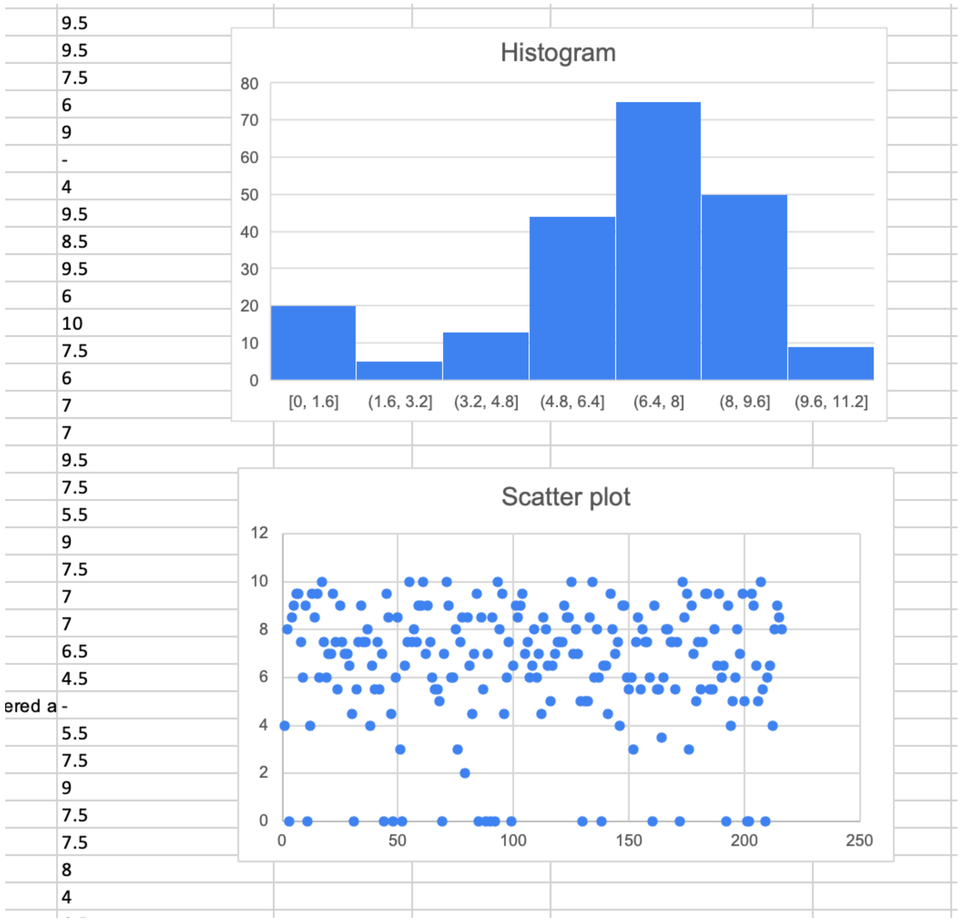
Mean: 7.16, Standard deviation: 1.75, Mode: 7.5

Quartiles:

1st: 6, 25% of grades is below this number

2nd (median): 7.5, 50% of grades is below this number

3rd: 8.5, 75% of grades is below this number



Review last time

Consider MNIST data

Each sample is an image of size 28×28, grascale valve

Features: $x \in [0,1]^{28 \times 28}$

$x_i \in [0,1]$

28 rows $\left\{ \begin{bmatrix} x_{11} & \cdots & \cdots \\ & & \end{bmatrix} \right.$ 28 column

Neural networks (NN)

Consider a single layer NN with 10 nodes in the first layer. How many parameters need to be determined to specify this NN? flatten the image $x \in [0,1]^{784}$

$784 \times 10 + 10 = 7850$ parameters.

(784 × 10: weight; 10: bias)

Convolutional neural networks (CNN)

Consider a single layer CNN with 5 filters, each of which is 3x3.

How many parameters need to be determined to specify this CNN?

$5 \times (3 \times 3 + 1) = 50$ parameters.

(3 × 3: weight; 1: bias)

Example applications of NN in IGM

Prof. Colin Jones: Predictive Control Lab

Goal: develop a dynamical model of the temperature evolution in a building for control (Control objective: minimize energy consumption while ensuring comfort of occupants)

Introduction

Issues of classical models

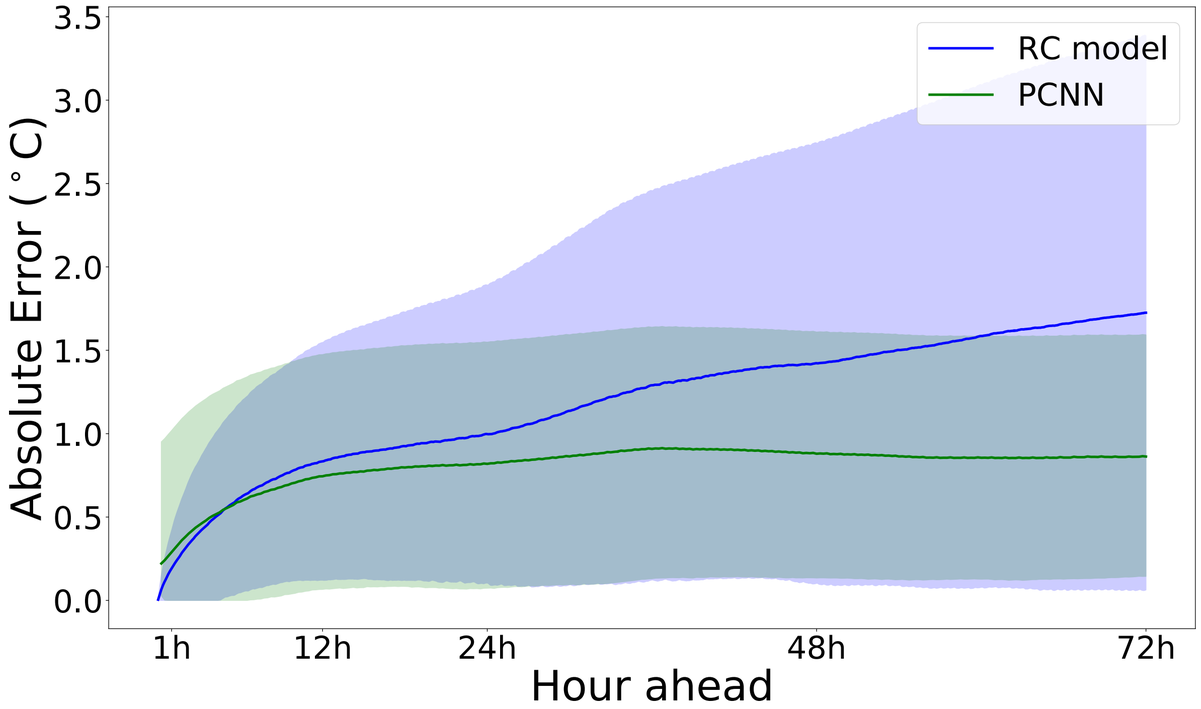
- Classically, people use **physics-based models**, built from first principles
  - Follow the **laws of physics**
  - Large **engineering overhead**

… but each building is different!

… can we leverage data in black-box methods?

© Steiner SA, Geneva

© Zooey Braun, Stuttgart



Introduction
Unsupervised learning

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a **data set with no pre-existing labels** and with a minimum of human supervision.

in the next techniques, we don’t use labels anymore
Note: the objective is vague but we will consider 2 concrete instances

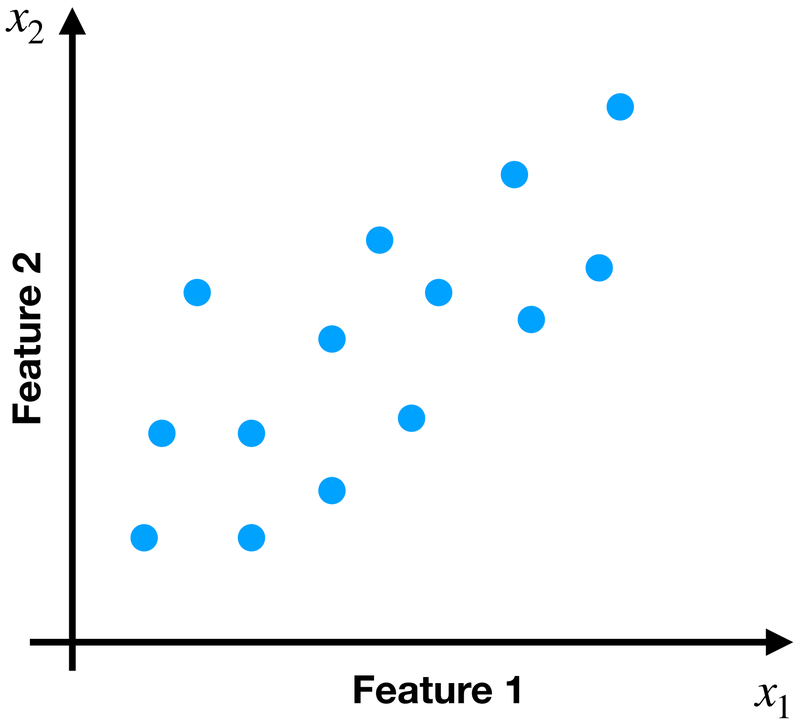
Dimensionality reduction

Through

principal component analysis

Motivation
Intuition

We have samples described by a series of features



We want to find a smaller set of new features that explain our sample because:

Less features is easier to visualize

Some of the current feature can be redundant

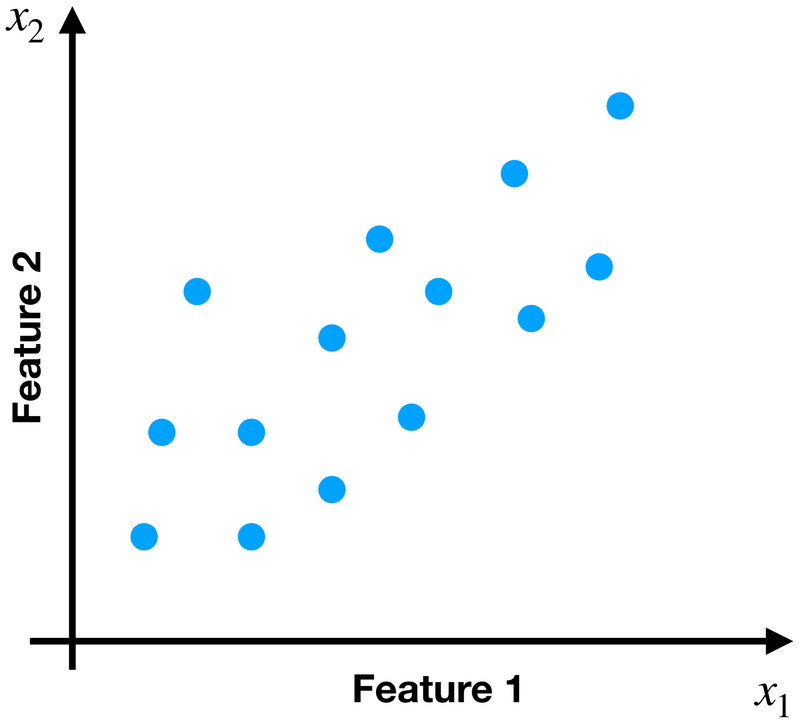
Some of the current features are not very useful to describe our samples

Principal component analysis (PCA)

Approach to dimensionality reduction

How to find this smaller set of new features ?

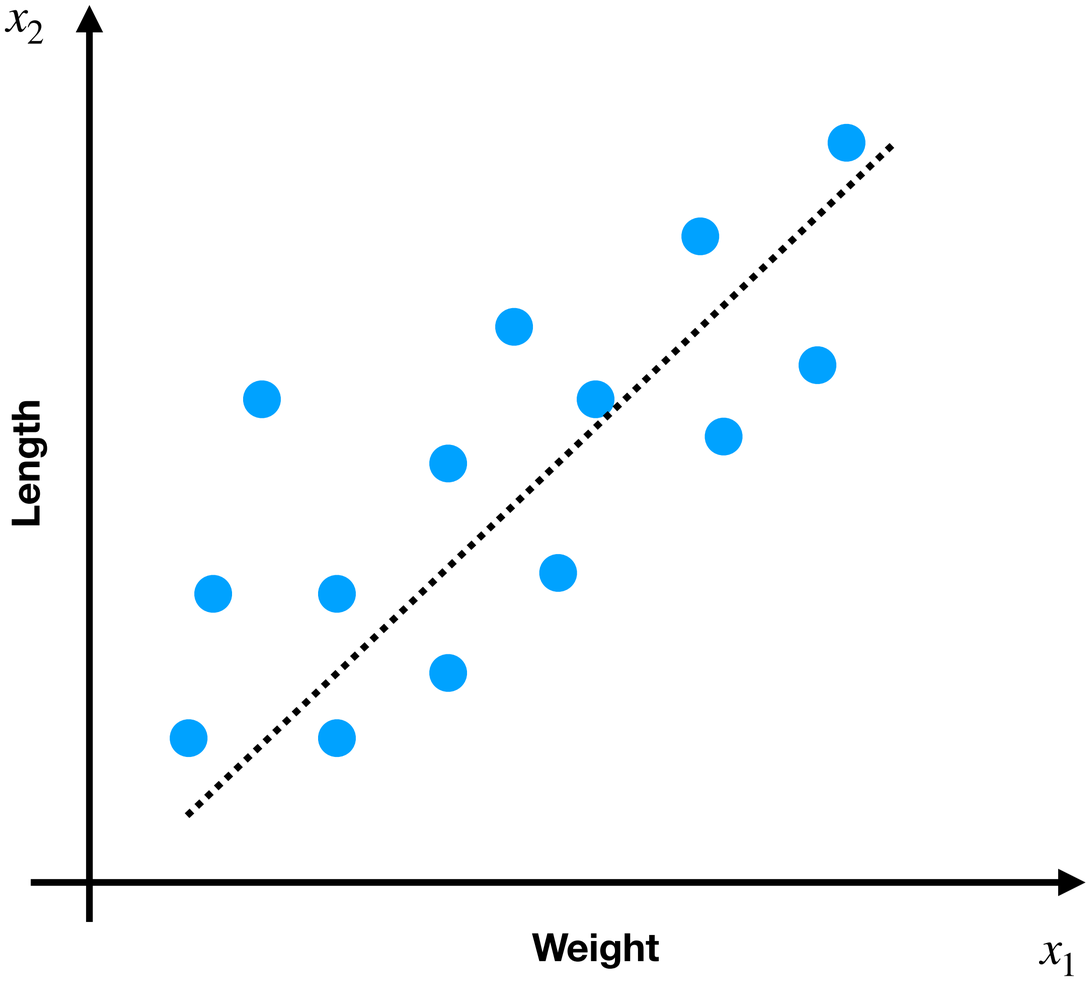
PCA : Find the best **linear combination** of features to create new features that explain our samples better



PCA

Projection of points onto a lower dimensional subspace

How to find this smaller set of new features ?



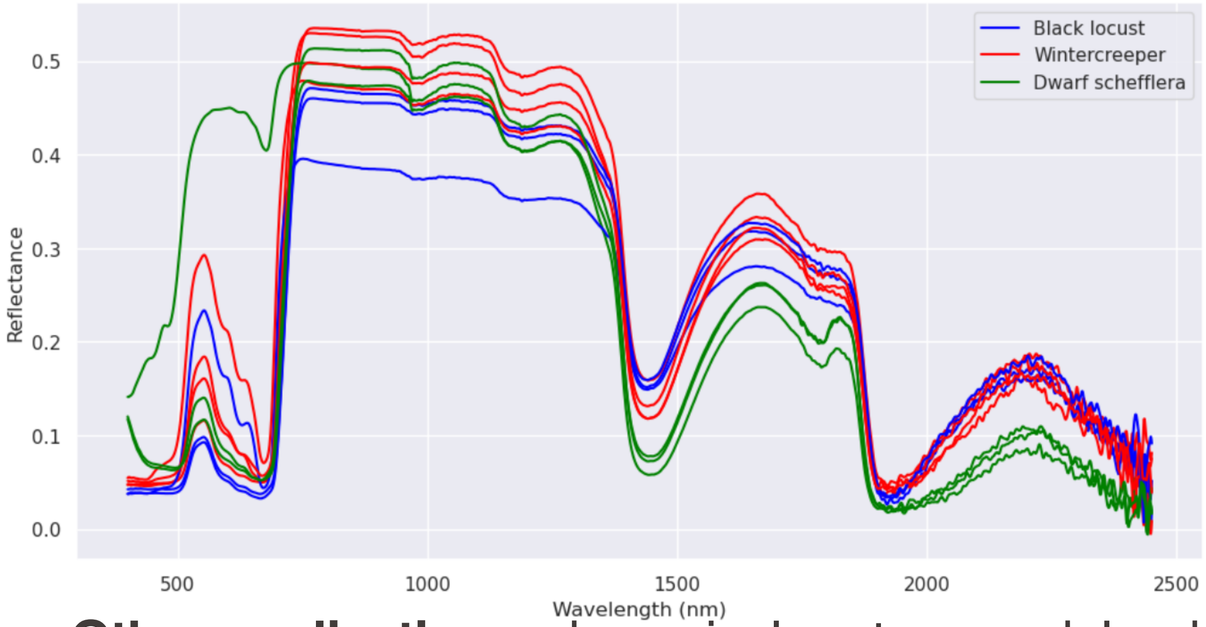
$$w_1 x_1 + w_2 x_2$$

The new feature

A mix of length and weight that describe our samples better

PCA - example application in python

- **Dataset:** Leaves' optical reflectance measured at each nm in the range of 450-2500 nm Wavelength, see more details about the experiment here
- **Question:** what is the size of feature vector for each leaf? $x \in \mathbb{R}$ $(2450 - 450 = 2000)$
- **Goal:** can we reduce the dimensionality of feature vector and still capture the distinguishing features of each leaf



- **Other applications:** dynamical system model order reduction, audio compression for recommendation algorithms, text processing for news recommendation

EPFL

Finding distance of a point to a subspace

Subspace $\theta_i \in \mathbb{R}^d$, $a_i \in \mathbb{R}$, $i = 1, 2, \dots, r$

a linear combination of $\{\theta_i\}_{i=1}^r$ is given by $\sum_{i=1}^r a_i \theta_i$

subspace spanned by $\{\theta_i\}_{i=1}^r$: $S := \{\sum_{i=1}^r a_i \theta_i \mid a_i \in \mathbb{R}, \theta_i \in \mathbb{R}^d, i = 1, \dots, r\}$

Distance of a point to a subspace $x \in \mathbb{R}^d$, let $\|x\|_2$ denote Euclidean norm

$$\text{dist}(x, S) = \min_{s \in S} \|x - s\|_2 = \min_a \|x - \theta a\|, \text{ where } a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_r \end{bmatrix},$$

$$\theta = [\theta_1 | \theta_2 | \cdots | \theta_r] \in \mathbb{R}^{d \times r}$$

Let us consider $\min_a \|x - \theta a\|^2$

$$\|x - \theta a\|^2 = \|x\|^2 - 2\langle x, \theta a \rangle + \|\theta a\|^2$$

$$= \|x\|^2 - 2 a^T \theta^T x + a^T \theta^T \theta a$$

Projection of a point onto a subspace

continued

Distance of a point onto a subspace $\text{(distance)}^2 : \|x\|^2 - 2a^T\Theta^T x + a^T\Theta^T\Theta a =: J(a)$

$$\frac{\partial}{\partial a} J(a) = 0 \Rightarrow a^* = (\Theta^T\Theta)^{-1}\Theta^T x$$

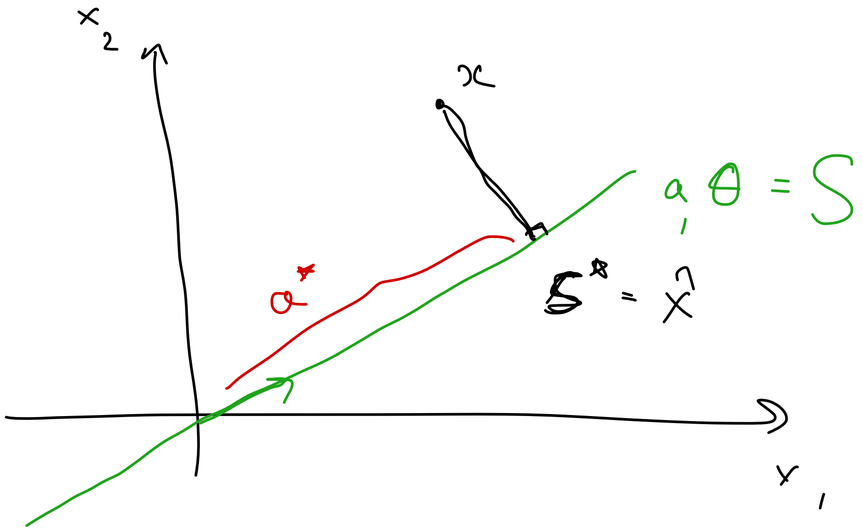
Choosing orthonormal vectors $\{\theta_i\}_{i=1}^r$ are orthonormal $\Rightarrow \Theta^T\Theta = I_{r\times r}$

$a^* = \Theta^T x$, s^*: closest point to x, $\Theta a^* = \Theta\Theta^T x \in \mathbb{R}^{d\times 1}$

Projection of a point onto a subspace

$$\hat{x} = s = \Theta\Theta^T x$$

$$\text{dist}(x, S) = \|x - \hat{x}\|_2 = \|x - \Theta\Theta^T x\|$$



Find a subspace to minimize average distance of data to it

Dataset $\{x^i\}_{i=1}^N \quad x^i \in \mathbb{R}^d$

Sum of distances of all data points to a subspace $S = \{\Theta a \mid a \in \mathbb{R}^r, \Theta \in \mathbb{R}^{d \times r}\}$ orthonormal

objective function in PCA is to find S such that

$J(S) = \frac{1}{N} \sum_{i=1}^{N} \text{dist}(x^i, S)$ is minimized

PCA chooses a subspace to minimize

$$J(S) = \frac{1}{N} \sum_{i=1}^{N} \|x^i - \hat{x}^i\| = \frac{1}{N} \sum_{i=1}^{N} \|x^i - \Theta\Theta^T x^i\|$$

for any given S spanned by $\{\theta_i\}_{i=1}^r \quad \Theta = [\theta_1 | \theta_2 | \cdots | \theta_r]$

Formulation of PCA objective using the data matrix

Standardize data: for each feature, subtract mean & divide by its std.

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & & & \\ x_1^N & x_1^N & \cdots & x_d^N \end{bmatrix} \in \mathbb{R}^{N \times d}$$

Frobenius norm of a matrix: $\|X\|_F = \sqrt{\text{trace}(X^T X)}$

Recall, we wanted to find $\{\theta_i\}_{i=1}^r$ to $\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \|x^i - \hat{x}^i\|$

PCA objective: $\min_{\theta} \frac{1}{N} \|X - X\theta\theta^T\|_F^2, \quad \theta \in \mathbb{R}^{d \times r}$

We are trying to find $\theta = [\theta_1 | \theta_2 | \cdots | \theta_r]$

Eigenvalue decomposition of a symmetric matrix

Let $C = X^T X \in \mathbb{R}^{d \times d}$ $\quad X \in \mathbb{R}^{N \times d}$

observe C is symmetric: $C = C^T$.

Fact: any symmetric matrix $C \in \mathbb{R}^{d \times d}$ has an eigenvalue decomposition as follows:

$$C = V D V^T, \quad D = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ 0 & & \ddots & \\ \vdots & & & \lambda_d \end{bmatrix} \in \mathbb{R}^{d \times d}, \quad \&$$

$$V = [v_1 \mid v_2 \mid \cdots \mid v_d] \in \mathbb{R}^{d \times d},$$ where $\{\lambda_i\}_{i=1}^d \subset \mathbb{R}$ are the eigenvalues of C and $\{v_i\}_{i=1}^d \subset \mathbb{R}^d$ are the corresponding eigenvectors.

Solution to PCA using eigenvalue decomposition

order $\{\lambda_i\}_{i=1}^{d}$: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$

Then, let $\theta_1^* = v_1$, $\theta_2^* = v_2$, $\cdots$, $\theta_r^* = v_r$

be the corresponding eigenvectors.

$$\arg\min_{\theta} \frac{1}{N} \| X - X\theta\theta^T \|_F^2 = [\theta_1^* | \theta_2^* | \cdots | \theta_r^*]$$

$\hat{X}$

$\{\theta_i^*\}_{i=1}^{r}$ are called the r principal components of our matrix $C = X^T X$.

EPFL

Principal components

Example: projection of data onto 2 principal components

Suppose λ_1, λ_2 of $C = X^T X$ are much larger than $\lambda_3, \lambda_4, \cdots, \lambda_d$

Then, we can let $r = 2$, and define the subspace S as span of v_1, v_2, where v_1, v_2 are eigenvectors corresponding to λ_1, λ_2, respectively.

$$S = \left\{ [v_1 \mid v_2] \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \;\middle|\; a_1, a_2 \in \mathbb{R} \right\}$$

* compressed features: $A = X\Theta \in \mathbb{R}^{N \times 2}$, $\Theta = [v_1 \mid v_2]$

each data point x^i is compressed to $a^i = \Theta^T x^i$, $A = \begin{bmatrix} a^1 \\ a^2 \\ \vdots \\ a^N \end{bmatrix} \in \mathbb{R}^{N \times 2}$

Reconstruction of data $\hat{X} = X\Theta\Theta^T$ based on this new subspace

PCA
Pseudo-code

1. **Standardize data** (subtract mean, divide by std) $X \in \mathbb{R}^{N \times d}$

   $\{x^i\}_{i=1}^{N}$

2. **Eigenvalue decomposition of the data covariance matrix** $C = X^T X \in \mathbb{R}^{d \times d}$

   $X^T X = V D V^T$

3. **To reduce the dimension to** $r << d$, order eigenvalues in decreasing value, and choose the eigenvectors corresponding to r largest eigenvalues.

4. **Compressed feature matrix** $A = X\Theta \in \mathbb{R}^{N \times r}$

PCA example - distinguishing texts

Defining features

Each data sample is a document $u^i \quad i = 1, \cdots, N$

There are d unique words in all the documents $x \in \mathbb{R}^d$

Feature j is positive for document i if word j is in document u^i

$x^i_j > 0 \iff$ word j is in document i

EPFL

TFIDF feature definition for documents

term frequency — inverse document frequency

Term frequency of word j in document u^i

$$f_{term}(u^i, j) = \frac{\#\text{ times word } j \text{ appears in } u^i}{\#\text{ all words in document } u^i}$$

Document frequency of word j

$$f_{doc}(j) = \frac{\#\text{ times word } j \text{ appears in any document}}{\#\text{ of all document}}$$

TFIDF for each word j in document u^i

$$x_j^i = f_{term}(u^i, j) \times \log \frac{1}{f_{doc}(j)} \quad , \quad i = 1, \dots, N$$

$$j = 1, \dots, d$$

PCA example

Based on “Principal Component Analysis” lecture of Stanford EE104: https://ee104.stanford.edu/lectures/pca.pdf

Distinguishing text: *The Critique of Pure Reason* by Immanuel Kant and *The Problems of Philosophy* by Bertrand Russell

d can be very high dimensional

Dimensionality reduction

Other techniques

There are several other techniques for dimensionality reduction :

Linear discriminant analysis (LDA)

Generalized discriminant analysis (GDA)

T-distributed Stochastic Neighbor Embedding (t-SNE)

Autoencoders (neural networks)

Autoencoder
Introduction

- An autoencoder is a type of neural network often used for dimensionality reduction.
- Autoencoders are trained in an **unsupervised** manner, by minimizing the reconstruction error / loss $\frac{1}{N}\sum_{i=1}^{N} L\left(x^i, \hat{x}^i\right)$

$\hat{x}^i = g(x^i) \longrightarrow$

$f \circ g(x^i) = \hat{x}^i$

f, g defined by neural net

- Example: Squared error

$$\min_{P_1, P_2} \frac{1}{N} \| x^i - f \circ g(x^i) \|_2^2$$

P_1, P_2 ~ parameters of the network

Autoencoder

Autoencoder vs. PCA

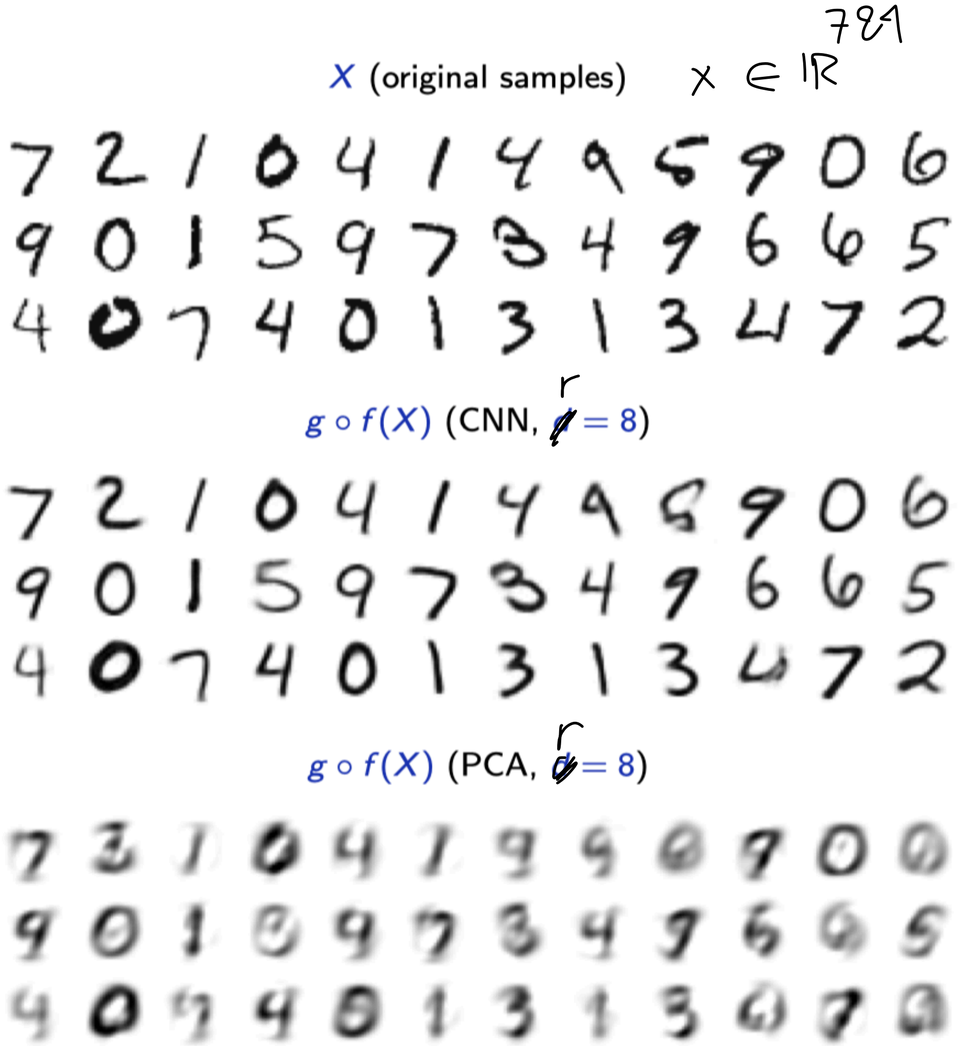


Image credit: F. Fleuret, Deep Learning (EPFL)

Top: Some examples of the original MNIST test samples

Middle: Reconstructed output from an auto-encoder with a latent space of 8 dimensions

This auto-encoder uses convolutional layers, and was trained on the MNIST training set

Bottom: Reconstructed output from PCA with 8 reduced dimensions

Summary - dimensionality reduction

Used for

- Exploratory data analysis
- Visualizing data
- Help reduce overfitting by reducing feature dimension

PCA: an approach to dimensionality reduction

- Projects data onto a linear subspace
- Useful in case there is approximately linear dependence between different features
- Easy to compute
- Connection to singular value decomposition (see Problem set 2)