OPTIMIZATION - OPTIMALISATIE

EXERCISE SESSION 2: FITTING PROBLEMS

The computational exercises are all in Python. First, we will set up the Python environment. To this end, we will import the following modules:

- **numpy:** package for multi-dimensional arrays
- **matplotlib:** package for plotting results
- **time:** package for various time-related functions
- **scipy:** library for optimization, integration, interpolation, eigenvalue problems, algebraic equations ...

You can do this by using the following commands:

```
import numpy as np
import matplotlib.pyplot as plt
import time
from scipy.optimize import linprog
```

Exercise 1 (**Least squares fitting).**

In linear least squares line fitting problems we have a set of measurements $(a_i, b_i) \in \mathbb{R}^2$, where $i \in \{1 \dots N\}$ onto which we would like to fit a line $b = x_1 a + x_2$. This can be expressed as an optimization problem by minimizing the residuals

$$\underset{x \in \mathbb{R}^2}{\textbf{minimize}} \sum_{i=1}^{N} (x_1 a_i + x_2 - b_i)^2 = \left\| \underbrace{\begin{pmatrix} a_1 & 1 \\ \vdots & \vdots \\ a_N & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{x} - \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix}}_{b} \right\|_2^2 . \tag{1.1}$$

The optimal solution can be calculated explicitly by solving the linear system

$$A^\top A x = A^\top b.$$

Tasks:

(1a) First generate a vector of "measurements": take $N = 50$ points. Define a to be a vector of evenly spaced points (tip: use the function `np.linspace(0,1,N)`). Then for b we set `b = 4a - 1`.

(1b) Use `b = b + np.random.randn(N, 1)` to add some Gaussian noise of standard deviation 1 to the vector b. Plot the obtained measurements using the command `plt.plot(a,b)`. (note: in Python you can always use `help()` to learn about the syntax of a command.)

(1c) Find the coefficients x_1, x_2 of the ℓ_2 fitting problem above by either using `np.linalg.pinv`(A) or `np.linalg.solve`$(A^\top A, A^\top b)$, which should give the same result. Plot the obtained line into the same graph as the measurements.

(1d) Introduce three outliers in your measurements b with standard deviation of 20, and check what happens with the fitted line in your plot.

(1e) Keep the measurements b (both with and without outliers) and the matrix A for the next exercise.

Exercise 2 **(ℓ_∞ fitting).**

In this case we also want to fit a line to a set of measurements, but we have a different cost function:

$$\underset{x\in\mathbb{R}^2}{\textbf{minimize}} \ \max_{i=1\ldots N} \{|x_1 a_i + x_2 - b_i|\} = \underset{x\in\mathbb{R}^2}{\textbf{minimize}} \ \|Ax - b\|_\infty .$$

This objective function is not differentiable, so we use a trick to form an equivalent problem. We introduce a slack variable s and solve the optimization problem

$$\begin{aligned} &\underset{x\in\mathbb{R}^2,\, s\in\mathbb{R}}{\textbf{minimize}} \ s \\ &\textbf{subject to} \ -s \le a_i x_1 + x_2 - b_i \le s, \quad i = 1, \ldots, N. \end{aligned} \tag{2.1}$$

Tasks:

(2a) Which type of optimization problem is this? The solver in Python to solve these problems is called `linprog`.

(2b) Use `? linprog` or `help(linprog)` to learn how to use the `linprog()` function to solve the problem.

(2c) Formulate the problem in `linprog`'s form of linear programming (use $\tilde{x} = \big(x_1,\, x_2,\, s\big)$)

$$\begin{aligned} \underset{\tilde{x}}{\textbf{minimize}} \quad & c^\top \tilde{x} \\ \textbf{subject to} \quad & A_{\text{ub}}\tilde{x} \le b_{\text{ub}} \\ & A_{\text{eq}}\tilde{x} = b_{\text{eq}} \\ & l \le \tilde{x} \le u. \end{aligned} \tag{LP}$$

(2d) Solve the problem using the measurements b from the previous exercise (both with and without outliers) and compare the results. (Disclaimer: the fit with outliers will be pretty bad.)

Exercise 3 **(ℓ_1 fitting).**

In this case we also want to fit a line to a set of measurements, but we have a different cost function:

$$\underset{x\in\mathbb{R}^2}{\textbf{minimize}} \sum_{i=1}^{N} |x_1 a_i + x_2 - b_i| = \underset{z}{\textbf{minimize}} \ \|Ax - b\|_1.$$

This objective function is not differentiable, so we use a trick to form an equivalent problem. Introduce the slack variables $s = (s_1, \ldots, s_N)$ and solve the optimization problem

$$\begin{aligned} &\underset{x\in\mathbb{R}^2,\ s\in\mathbb{R}^N}{\textbf{minimize}} && \sum_{i=1}^{N} s_i \\ &\textbf{subject to} && -s_i \leq a_i x_1 + x_2 - b_i \leq s_i, \quad i = 1, \ldots, N. \end{aligned} \tag{3.1}$$

Tasks:

(3a) Which type of optimization problem is this? The solver in Python to solve these problems is called `linprog`.

(3b) Formulate the problem in `linprog`'s form of linear programming (LP).

(3c) Solve the problem using the measurements b from the previous exercise (both with and without outliers) and compare the results.