Exercise Sheet 1:
Linear Fitting Problems with Different Norms

In this first exercise we choose a visualization and optimization environment where linear and quadratic programming solvers are available, such as MATLAB, and investigate estimation and fitting problems where we experiment with different norms.

1. Linear Least Squares ($L_2$) Fitting
In linear least squares fitting problems we have a set of measurements $(x_i, y_i) \in \mathbb{R}^2, i = 1, \ldots, N$ onto which we would like to fit a combination of basis functions. Here, we choose two basis functions, a constant and a first order monomial, i.e. we fit a line $ax + b$ to the data. The least squares fit can be expressed as the following optimization problem

$$\min_{a,b} \sum_{i=1}^{N} (ax_i + b - y_i)^2 = \min_{a,b} \left\| \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \right\|_2^2. \tag{1}$$

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

$$J^T J \begin{pmatrix} a \\ b \end{pmatrix} = J^T y, \tag{2}$$

where

$$J = \begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}. \tag{3}$$

Tasks:

(a) First, we have to generate “measurements”. For this aim, generate $N = 30$ points of a line, e.g. the line $y = 3x + 4$ with $x_1 = 1, \ldots, x_N = N$. 

1
(b) Use \( y = y + \text{randn}(30,1) \) to add some Gaussian-noise of standard deviation 1 to the \( y \) values. Plot the obtained measurements using \( \text{plot}(x,y) \). (note: in MATLAB you can always use \text{help} \ldots \) or \text{doc} \ldots to learn about the syntax of a command.)

(c) Find the coefficients \( a, b \) of the \( L_2 \) fitting problem by using \( \text{pinv} \) or \( (J^\prime J)(J^\prime y) \), which should give the same result. Plot the obtained line into the same graph as the measurements.

(d) Add 3 outliers of arbitrary size to your measurements \( y \) and check what happens with the fitted line in your plot.

(e) Keep the measurements \( x, y \) and the matrix \( J \) for the next exercise.

2. **Linear \( L_1 \) Fitting**

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

\[
\min_{a,b} \sum_{i=1}^{N} |(ax_i + b - y_i)| = \min_{a,b} \left\| \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \right\|_1 \quad (4)
\]

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

\[
\min_{a,b,s} \sum_{i=1}^{N} s_i \quad (5)
\]

subject to

\[-s_i \leq x_i a + b - y_i \leq s_i, \quad i = 1, \ldots, N \quad (6)\]

\[0 \leq s_i, \quad i = 1, \ldots, N \quad (7)\]

Tasks:

(a) Determine the type of optimization problem you got. The solver in MATLAB to solve these problems is called \text{linprog}

(b) Use \text{help} or doc \text{doc} to obtain the syntax of \text{linprog}

(c) Formulate the problem in \text{linprog}'s standard form

\[
\min_{z} f^T z \quad (8)
\]

subject to

\[Az \leq b \quad (9)\]

\[Cz = d \quad (10)\]

\[l_z \leq z \leq u_z, \quad (11)\]
Use the space below to define the matrix $A$ and the vectors $f, b$ if your variables are ordered as $z = (a, b, s_1, \ldots, s_N)$. Use the matrix $J$ from the previous exercise to define $A$.

(d) Repeat the tasks from the previous exercise and compare the results. In particular, increase considerably the size of the outliers.

(e) Now increase the number of outliers. How many outliers can be present before the line gets distorted by them?

3. **Linear Huber Penalty Fitting**

   In order to combine the best of $L_1$ and $L_2$ fitting, the Huber penalty function $\phi(z)$ was invented. It is a quadratic for arguments $|z| \leq 1$ and for larger arguments grows only linearly. Let us again fit a line to the set of measurements, but with the Huber penalty in the cost function:

   $$\min_{a, b} \sum_{i=1}^{N} \phi(ax_i + b - y_i)$$  \hspace{1cm} (12)

   As before, we can use a trick to form an equivalent problem that falls into a generic class. We introduce two sets of slack variables $u_1, \ldots, u_N$ and $v_1, \ldots, v_N$ and we solve the optimization problem

   $$\minimize_{a, b, u_i, v_i} \quad \sum_{i=1}^{N} \frac{1}{2} u_i^2 + v_i$$  \hspace{1cm} (13)

   subject to

   $$\begin{align*}
   -u_i - v_i &\leq x_i a + b - y_i \leq u_i + v_i, & i = 1, \ldots, N \\
   0 &\leq u_i \leq 1, & i = 1, \ldots, N \\
   0 &\leq v_i, & i = 1, \ldots, N
   \end{align*}$$  \hspace{1cm} (14, 15, 16)

   Tasks:

   (a) Determine the type of optimization problem you got. The solver in MATLAB to solve these problems is called quadprog
(b) Use help or doc doc to obtain the syntax of quadprog

(c) Formulate the problem in quadprog’s standard form.

\[
\begin{align*}
\text{minimize} & \quad f^T z + \frac{1}{2} z^T H z \\
\text{subject to} & \quad A z \leq b \\
& \quad C z = d \\
& \quad l_z \leq z \leq u_z,
\end{align*}
\]

Use the space below to define the matrices \(A, H\) and the vectors \(f, b\) if your variables are ordered as \(z = (a, b, u_1, \ldots, u_N, v_1, \ldots, v_N)\). Use the matrix \(J\) from the previous two exercises to define \(A\).

(d) Repeat the tasks from before and again compare the results. In particular, verify that for small noise levels and without outliers, the result is identical to the \(L_2\) fitting.