SoSe 2020 Kiel University
Dr. Joscha Reimer Department of Computer Science

Optimization and Data Science
4. Homework exercises

Theoretical exercise 1:
Calculate a solution of the following linear regression problem (by hand):

$$\min_{x \in \mathbb{R}^2} \|Ax - z\|_2^2 \quad where \quad A := \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \end{pmatrix} \quad and \quad z := \begin{pmatrix} 6 \\ 12 \\ 6 \end{pmatrix}$$

Is this solution unique?

Programming exercise 1:
We want to compare linear regression approximations resulting from a different number of data values. For this purpose we use the daily COVID-19 data provided by the WHO which is stored in the "WHO-COVID-19-global-data.cvs" file.

a) *Extract the daily cumulative deaths caused by COVID-19 in Germany since the first of March and visualize this data. (You are not allowed to alter the file content itself.)*

b) *Calculate a linear regression approximation using only the extracted data which belongs to the first week, the first two weeks, the first three weeks and so on and visualize all these approximations together with the data for the entire period of the extracted data.*

Programming exercise 2:
We want to compare approximations resulting from FFT as well as polynomial regression. For this purpose we use the daily DAX date from the first programming exercise of the second homework exercises.

a) *Calculate approximations of the data using polynomial regression with one to ten parameters (which corresponds to degree zero to nine).*

b) *Plot the data, the approximations resulting from FFT with one, five and ten Fourier coefficients as well as resulting from polynomial regression with one, five and ten parameters.*
Which approximation method would you prefer based on this visualization?

c) *Plot the mean squared error resulting from the FFT approximations from one to ten Fourier coefficients as well as from the polynomial regression approximations with one to ten parameters.*
Which approximation method would you prefer based on this visualization?

The mean squared error between some vectors $x, y \in \mathbb{R}^n$ is defined as $\frac{1}{n}\sum_{i=0}^{n}(x_i - y_i)^2$.
You do not have to implement the polynomial regression method. Most software packages (e.g. NumPy) should have a method that allow to determine the parameters of the polynomial regression (coefficients of the polynomials).

Programming exercise 3:
We want to compare approximations of images resulting from FFT as well as SVD. For this purpose we use the image from the second programming exercise of the second homework exercises.

a) Use SVD to approximate this image using 20%, 10% and 2% of the (biggest) singular values and plot the three resulting approximated images.

b) Determine how many (of the biggest) singular values you would use to get a good compromise between a low number of coefficients and a still recognizable picture.

c) How many real valued numbers do you have to store for the approximation determined in b)? What percentage is this compared to the number of real values in the original image?

d) Based on your results from this programming exercise and the second one of the second homework exercises, would you recommend FFT or SVD for this image?

Hint: Since SVD can only be applied to matrices, each color should be approximated separately by SVD.

The solutions of the theoretical exercises will be discussed on 04. Mai 2020.