ETH zürich

MAD exercise session 2

Least squares, Newton's method

Pascal Auf der Maur

Recap

- Least squares is an invaluable tool to analyze data
- We saw that we can use it to find the “best“ function that describes our data wrt. to our costfunction

Last Exercise

Problem 1

- First implemetation of least squares
- Plot the results with matplotlib
- Usually we solve the equation

$$A^T A w = A^T b$$

instead of inverting the matrix.

Last Exercise

P & P 1

- Performing LSQ by hand
- For a linear case use the simplified formula found in the script:

$$w_1 + w_2 x_i \approx y_i$$

$$w_1 = \frac{\left(\sum x_i^2\right)\left(\sum y_i\right) - \left(\sum x_i\right)\left(\sum x_i y_i\right)}{N\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}$$

$$w_2 = \frac{N\left(\sum x_i y_i\right) - \left(\sum x_i\right)\left(\sum y_i\right)}{N\left(\sum x_i^2\right) - \left(\sum x_i\right)^2}$$

Formulating matrices

- The only restriction in choosing functions which we want to fit to the data is that the parameters contained in *w* enter linearly.
- Examples:

$$w_0 x^3 + w_1 x^2 + w_2 x + w_3 = y$$

$$w_0 sin(x\pi) + w_1 cos(x\pi) = y$$

$$w_0 e^x + w_1 e^{2x} = y$$

Special case

Sometimes a function can be rewritten to fulfill this requirement:

$$w_0 e^{-w_1 x^2} = y \quad | ln()$$

$$ln(w_0) - w_1 x^2 = ln(y)$$

Condition number

- Last time we saw that the solution of the least squares problem is:

$$x = (A^T A)^{-1} A^T b$$

- But what if the matrix $A^T A$ is close to beeing singular?
- The condition number tells us how close the matrix is to beeing singular.
- The condition number is defined as:

$$\kappa(A) = \|A\| \|A^{-1}\| = \frac{\sigma_1}{\sigma_N}$$

- The used norm is the L_2 matrix norm where the result is the largest singular value of the matrix for positive definite matrices
- The condition number for our approach:

$$\kappa(A^T A) = \kappa(A)^2$$

Alternatives

- Instead of the solving the normal equation like we did until now, we can either use the QR or SVD decomposition

QR-Decomposition

$$A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1$$

Our problem becomes

$$w = R_1^{-1} Q_1^T y \qquad \kappa(A) = \kappa(R_1)$$

SVD-Decomposition

$$A = U \Sigma V^T$$

- We define the pseudo-inverse of A as

$$A^{\dagger} = V \Sigma^{\dagger} U^T \quad \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \dots, 0) \quad \Sigma^{\dagger} = \mathrm{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \dots, 0)$$

- Our problem becomes:

$$w = V \Sigma^{\dagger} U^T y$$