Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Models, Algorithms and Data (MAD): Introduction to Computing

Spring semester 2019

Prof. Dr. Jens Honore Walther
Dr. Julija Zavadlav
ETH Zentrum, CLT
CH-8092 Zürich

Set 1

Issued: 22.02.2019; Due: 03.03.2019

In this exercise, you will learn about least squares fitting of data points and the sensitivity of the fit to the data quality and the amount of noise. In order to complete the full task, you should proceed incrementally and always remember to test and debug your program after every modification.

Question 1: Linear least squares on 2D data

A scientist wants to determine the resistance R of a conductor. He applies different currents I through the conductor and measures the voltage V across. He uses the Ohm's law, i.e., $V = RI$. Furthermore, he assumes that the voltmeter is not properly calibrated and outputs values with a constant shift. Thus, $V = V_0 + RI$. He performs a set of $N = 10$ experimental measurements $\{I_j, V_j\}_{j=1}^{N=10}$.

I [A]	V [V]
0.01	3.92
0.02	6.23
0.03	5.52
0.04	8.45
0.05	10.11
0.06	15.11
0.07	15.11
0.08	18.22
0.09	19.97
0.10	12.77

a) Start your program by initializing two arrays containing the experimental measurements. Visualize the data by plotting it. We suggest using `pyplot`.

b) The scientist decides to determine the two unknown parameters V_0 and R using linear least squares. Rewrite the problem in the matrix form and use the least squares solution (given in the lectures) to evaluate the unknowns. To achieve this, you will need to perform matrix inversion, the transpose of the matrix, matrix multiplication etc. You can either use the Python numpy package or derive the solution yourself. We suggest you do both and confirm that you get the same result. Visualize the line you obtained as the best least squares fit.

c) Now, we will examine the least squares sensitivity to noise. Instead of using the experimental data, we will generate the synthetic data using $V_0 = -0.5$ [V] and $R = 150$ [Ω]. Evaluate V for the same 10 currents. Perform the least squares fit and visualize the result. Now add some random noise to the initial data. In Python, uniform noise over an interval $[a, b]$

can be added using the function random.uniform(a,b) in the random package. Use uniform distribution over the interval $[-1.0, 1.0]$ and add corresponding random numbers to the elements of `V[j]`: $V_j = -0.5 + 150 I_j + G U_{[-1.0,1.0]}$, where G corresponds to different noise levels. Get the least squares estimates for the parameters using different $G = 1, 1.5, 2$. What would you expect to see? Does your plot meet your expectations?

d) Now add an outlier to the data by changing one of the data points as `V[7] = 0.5*V[7]` (i.e. halve one of the values). Estimate the least squares solution and visualize the resulting estimated line as in the previous subquestion. What would you expect to see now? Does your plot meet your expectation?

Question 2: Linear least squares on 3D data

We will now perform a least squares fit using 3-D data $\{x_i, y_i, z_i\}_{i=1}^N$ and consider a function $z(x, y) = \alpha + \beta x + \gamma y$ with unknown coefficients α, β and γ.

a) Write the problem in the matrix formulation. Using the same procedure as in the 2D case given in the lecture notes in section (1.4.3), derive the least squares solution for the 3x3 matrix.

b) Use the code from question 1 and make the appropriate modifications to compute the least squares fit of 3-D data. Note that you are no longer fitting a line to the data, but a plane. Calculate the least square approximation based on the function $z(x, y)$. Generate a total of $N = 100$ data points using a 10 by 10 grid on the x and y 2D domain, i.e., $x_i = 0.1 \cdot i, y_j = 0.1 \cdot j, 0 \leqslant i, j \leqslant 9$. Similar to exercise 1, add noises to the 3D data, i.e., $z = 1.0\, x + 1.0\, y + G U_{[-0.1,0.1]}$ $(\alpha = 0,\ \beta, \gamma = 1.0)$. Test different $G = 1, 10, ..$ Are the estimated parameters consistent with what you expected?

c) Note that we used 100 points as our data point for the previous question, whereas we used N=10 points in the 2-D example. Repeat the estimation in question 2b) but only with 10 points (randomly generate 10 pairs of x and y values within the range 0 to 1). How does the reduced amount of data (qualitatively) affect the parameter estimation?

Question 3: [Advanced] Dependency of the LSQ fit on the noise

In this question you will understand how the LSQ fit behaves as a function of the number of data points (N) in the presence of noisy data. Let x_i, $i = 1, \dots, N$ be points from a bounded domain (i.e., from an interval). Consider the data points $y_i = \alpha_0 + \beta_0 x_i$, $i = 1, \dots, N$, and the noisy data $y_i^* = y_i + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Proceed as follows:

1. Consider the model $f(x)$ and $f^*(x)$ with parameters α, β and α^*, β^*, that correspond to the data (x_i, y_i) and (x_i, y_i^*), respectively.
2. Express the difference between the two models in terms of differences in the paramaters, $\alpha - \alpha^*$ and $\beta - \beta^*$.
3. Express $\alpha - \alpha^*$ in terms of elements of the matrix $H = A^T A$. Do the same for $\beta - \beta^*$.
4. Find out how the elements of matrix H^{-1} behave as a function of N

Hint: Assume x_i is bounded. Use the Central Limit Theorem.