Assignment 2 - Numerical Linear Algebra

Prof.: Bernardo Freitas Paulo da Costa

TA: Beatriz Lúcia Teixeira de Souza

Student: Arthur Rabello Oliveira

28/04/2025

Contents

1. Problem 1: Traditional Least Squares (a - f)

1.1. Linear Regression (a)

We have a set of equally spaced points $S := \{t_i = \frac{i}{m}\}, i = 0, 1, ..., m$, we will find the best *line* $f(t) = \alpha + \beta t$ that approximates the points $(t_i, b_i) \in \mathbb{R}^2$

The system of equations to be solved is to be given as a function of t_i, b_i, m.

Solution:

Approximating 2 points in $\mathbb{R}^2$ by a line is trivial, now approximating more than 2 points is a task that requires linear algebra. To see this, we will analyze the following example to build intuition for the general case:

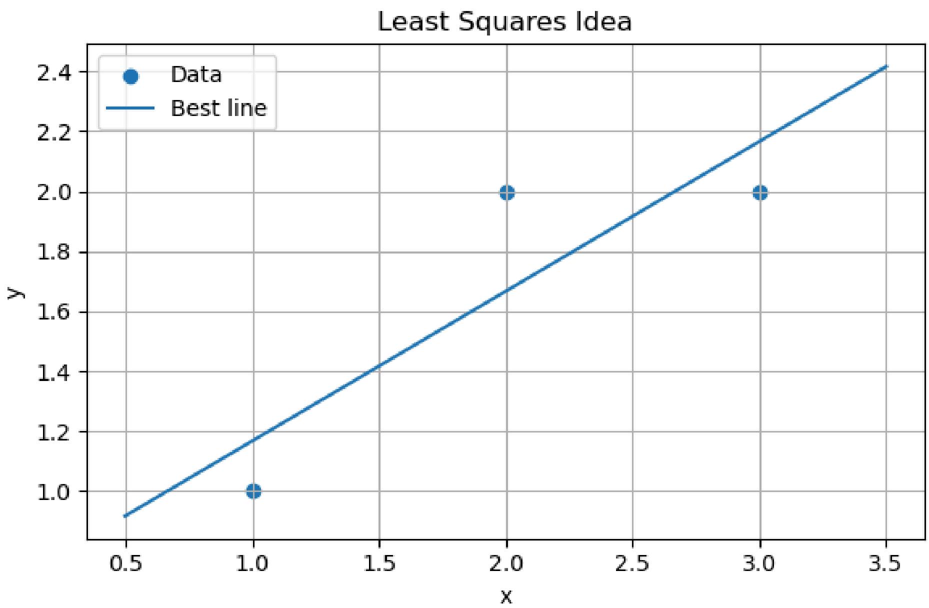

Figure 1: A glimpse into what we want to see

Given the points $(1, 1), (2, 2), (3, 2) \in \mathbb{R}^2$, we would like a *line* $f(t) = y(t) = \alpha + \beta t$ that best approximates these 3 points, in other words, since we know that the line does not pass through all of the 3 points, we would like to find the *closest* line to the line that would pass through the 3 points, so the system:

$$\begin{aligned} f(1) &= \alpha + \beta = 1 \\ f(2) &= \alpha + 2\beta = 2 \\ f(3) &= \alpha + 3\beta = 2 \end{aligned} \tag{1}$$

Clearly has no solution, (the line does not cross the 3 points), but it has a *closest solution*, which we can find through **projections**, the system is:

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} \alpha \\ \beta \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}}_{b} \tag{2}$$

Let $x^* \neq x$ be a solution to the system, we want to **minimize** the error produced by approximating the points through a line, so if the **error** is $e = Ax - b$, we want the smaller error *square* possible (that is why least squares). We square the error to avoid and detect outliers, so:

$$e_1^2 + e_2^2 + e_3^2 \tag{3}$$

Is what we want to minimize, where e_i is the error (distance) from the ith point to the line:

Figure 2: The errors (distances)

In this case, we will project this system into the column space of the matrix A, giving us the closest solution, and the least squares solutions is when $\hat{x}$ minimizes $\|Ax - b\|^2$, this occurs when the residual $e = Ax - b$ is orthogonal to $C(A)$, since $N(A^T) \perp C(A)$ and the dimensions sum up the left dimension of the matrix, so:

$$\begin{gathered} A^T A\hat{x} = A^T b \\ = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \\ = \begin{pmatrix} 3 & 6 \\ 6 & 14 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \end{pmatrix} \end{gathered} \tag{4}$$

So the system to find $\hat{x} = \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix}$ becomes:

$$\begin{gathered} 3\alpha + 5\beta = 5 \\ 6\alpha + 14\beta = 11 \end{gathered} \tag{5}$$

Notice that with the *errors* e_i^2 as:

$$\begin{gathered} e_1^2 = (\alpha + \beta - 1)^2 \\ e_2^2 = (\alpha + 2\beta - 2)^2 \\ e_3^2 = (\alpha + 3\beta - 2)^2 \end{gathered} \tag{6}$$

The system in eq. (5) is *precisely* what is obtained after using partial derivatives to minimize the erros sum as a function of (α, β):

$$\begin{gathered} f(\alpha, \beta) = (\alpha+\beta-1)^2 + (\alpha+2\beta-2)^2 + (\alpha+3\beta-2)^2 \\ = 3\alpha^2 + 14\beta^2 + 12\alpha\beta - 10\alpha - 22\beta + 9, \\ \frac{\partial f}{\partial \alpha} = \frac{\partial f}{\partial \beta} = 0 \Leftrightarrow 6\alpha + 12d - 10 = 28\beta + 12\alpha - 22 = 0 \Leftrightarrow \begin{cases} 3c + 6d = 11 \\ 6c + 14d = 11 \end{cases} \end{gathered} \quad 7$$

This new system has a solution in $\hat{\alpha} = \frac{2}{3}, \hat{\beta} = \frac{1}{2}$, so the equation of the optimal line, obtained through *linear regression* (or least squares) is:

$$y(t) = \frac{2}{3} + \frac{1}{2}t. \quad 8$$

If we have $n > 3$ points to approximate through a line, the reasoning is analogous:

With $S \coloneqq \{t_i = \frac{i}{m}\}, i = 0, 1, ..., m$, we will find the best *line* $f(t) = \alpha + \beta t$ that approximates the points $(t_i, b_i) \in \mathbb{R}^2$

The system of equations to be solved be given as a function of t_i, b_i, m.

We want to find the extended system as we did in eq. (7), so our line is:

$$f(t) = \alpha + \beta t \quad 9$$

That best approximates the points $(0, b_0), (\frac{1}{m}, b_1), ..., (1, b_m)$. The system is:

$$\begin{gathered} f(0) = b_0 = \alpha, \\ f\left(\frac{1}{m}\right) = b_1 = \alpha + \frac{\beta}{m}, \\ f\left(\frac{2}{m}\right) = b_2 = \alpha + \frac{2}{m}\beta \\ ... \\ f(1) = b_m = \alpha + \beta \end{gathered} \quad 10$$

And the $Ax = b$ matrices alternative:

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 1 & \frac{1}{m} \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} \alpha \\ \beta \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}}_{b} \quad 11$$

Projecting into $C(A)$, we have:

$$\begin{gathered} A^T A x = A^T b \\ = \begin{pmatrix} 1 & 1 & ... & 1 \\ 0 & \frac{1}{m} & ... & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & \frac{1}{m} \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} m+1 & \frac{m+1}{2} \\ \frac{m+1}{2} & \frac{(m+1)(2m+2)}{6m} \end{pmatrix} \cdot \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} \\ = \begin{pmatrix} 1 & 1 & ... & 1 \\ 0 & \frac{1}{m} & ... & 1 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} b_1 + b_2 + ... + b_m \\ \frac{1}{m}[b_2 + 2b_3 + ... + (m-1)b_m] \end{pmatrix} \end{gathered} \quad 12$$

So the new system to be solved is:

$$\begin{pmatrix} m+1 & \frac{m+1}{2} \\ \frac{m+1}{2} & \frac{(m+1)(2m+2)}{6m} \end{pmatrix} \cdot \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} b_1 + b_2 + \ldots + b_m \\ \frac{1}{m}[b_2 + 2b_3 + \ldots + (m-1)b_m] \end{pmatrix} \quad (13)$$

Or:

$$\begin{pmatrix} m+1 & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & \sum_{i=1}^m t_i^2 \end{pmatrix} \cdot \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m b_i \\ \sum_{i=1}^m \frac{i}{m} \cdot b_i \end{pmatrix} \quad (14)$$

1.2. The Condition number of a matrix

Conditioning numbers are very important in numerical analysis and to the efficiency of numerical procedures, given the sad fact that machines with infinite memory have not been built (yet), conditioning and stability are of unquantifiable importance

1.2.1. The Conditioning number of a problem

A *problem* is usually described as a function $f : X \to Y$ from a **normed** vector space X of data (it has to be normed so qe can *quantify* data) and a *normed* vector space Y of solutions, f is not always a well-behaved continuous function, which is why we are interested in **well-conditioned** problems and **ill-conditioned** problems, which we define:

Definition 1.2.1.1: (Well-Conditioned Problem) A problem $f : X \to Y$ is *well-conditioned* at $x_0 \in X \Leftrightarrow \forall \varepsilon > 0, \exists \delta > 0 \mid \|x - x_0\| < \delta \Rightarrow \|f(x) - f(x_0)\| < \varepsilon$.

This means that small perturbations in x lead to small changes in $f(x)$, a problem is **ill-conditioned** if $f(x)$ can suffer huge changes with small changes in x.

We usually say f is well-conditioned if it is well-conditioned $\forall x \in X$, if thereis at least one x_i in which the problem is ill-conditioned, then the whole problem is ill-conditioned.

Conditioning numbers are a tool to quantify how well/ill conditioned a problem is:

Definition 1.2.1.2: (Absolute Conditioning Number) Let δx be a small pertubation of x, so $\delta f = f(x + \delta x) - f(x)$. The **absolute** conditioning number of f is:

$$\hat{\kappa} = \lim_{\delta \to 0} \sup_{\|\delta x\| \leq \delta} \frac{\|\delta f\|}{\|\delta x\|} \quad (15)$$

The limit of the supremum can be seen as the supremum of all *infinitesimal* perturbations, so this can be rewritten as:

$$\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} \quad (16)$$

If f is differentiable, we can evaluate the abs.conditioning number using its derivative, if J is the matrix whose $i \times j$ entry is the derivative $\frac{\partial f_i}{\partial x_j}$ (jacobian of f), then we know that $\delta f \approx J(x) \delta x$, with equality in the limit $\|\delta x\| \to 0$. So the absolute conditioning number of f becomes:

$$\hat{\kappa} = \|J(x)\|, \quad (17)$$

1.2.2. The relative Conditioning Number

When, instead of analyzing the whole set X of data, we are interested in *relative* changes, we use the **relative condition number:**

Definition 1.2.2.1: (Relative Condition Number) Given $f : X \to Y$ a problem, the *relative condition number* $\kappa(x)$ at $x \in X$ is:

$$\kappa(x) = \lim_{\delta \to 0} \sup_{\|\delta x\| \leq \delta} \left(\frac{\|\delta f\|}{\|f(x)\|}\right) \cdot \left(\frac{\|\delta x\|}{\|x\|}\right)^{-1} \quad (18)$$

Or, as we did in Definition 1.2.1.2, assuming that δf and δx are infinitesimal:

$$\kappa(x) = \sup_{\delta x}) \left(\frac{\|\delta f\|}{\|f(x)\|}\right) \cdot \left(\frac{\|\delta x\|}{\|x\|}\right)^{-1} \quad (19)$$

If f is differentiable:

$$\kappa(x) = (\|J(x)\|) \cdot \left(\frac{\|f(x)\|}{\|x\|}\right)^{-1} \quad (20)$$

Relative condition numbers are more useful than absolute conditioning numbers because the **floating point arithmetic** used in many computers produces *relative* errors, the latter is not a highlight of this discussion.

Here are some examples of conditioning:

Example 1.2.2.1.: Consider the problem of obtaining the scalar $\frac{x}{2}$ from $x \in \mathbb{R}$, the function $f(x) = \frac{x}{2})$ is differentiablke, so by eq. (20):

$$\kappa(x) = (\|J\|) \cdot \left(\frac{\|f(x)\|}{\|x\|}\right)^{-1} = \left(\frac{1}{2}\right) \cdot \left(\frac{\frac{x}{2}}{x}\right)^{-1} = 1. \quad (21)$$

This problem is well-conditioned (κ is small).

Example 1.2.2.2.: Consider the problem of computing the scalar $x_1 - x_2$ from $(x_1, x_2) \in \mathbb{R}^2$ (Use the ∞-norm in $\mathbb{R}^2$ for simplicity). The function associated is differentiable and the jacobian is:

$$J = \begin{pmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 1 & -1 \end{pmatrix} \quad (22)$$

With $\|J\|_\infty$ = 2, so the condition number is:

$$\kappa = (\|J\|_\infty) \cdot \left(\frac{\|f(x)\|}{\|x\|}\right)^{-1} = \frac{2}{|x_1 - x_1| \cdot \max\{|x_1|, |x_2|\}} \quad (23)$$

This problem can be ill-conditioned if $|x_1 - x_2| \approx 0$ (κ gets huge), and well-conditioned otherwise

.

1.2.3. Conditioning Number of matrices

We will deduce the conditioning number of a matrix from the conditioning number of *matrix-vector* multiplication:

Consider the problem of obtaining Ax given $A \in \mathbb{C}^{m\times n}$, we will calculate the relative condition number with respect to perturbations in x, directly from Definition 1.2.2.1, we have:

$$\kappa = \sup_{\delta x} \frac{\|A(x+\delta x) - Ax\|}{\|Ax\|} \cdot \left(\frac{\|\delta x\|}{\|x\|}\right)^{-1} = \sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} \cdot \left(\frac{\|Ax\|}{\|x\|}\right)^{-1} \tag{24}$$

Since $\sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} = \|A\|$, we have:

$$\kappa = \|A\| \cdot \frac{\|x\|}{\|Ax\|} \tag{25}$$

This is a precise formula as a function of (A, x).

Suppose for a moment that A is square and non-singular:

Theorem 1.2.3.1: For $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $\det(A) \neq 0$, the following holds:

$$\frac{\|x\|}{\|Ax\|} \leq \|A^{-1}\| \tag{26}$$

Proof: Since $\forall A, B \in \mathbb{R}^{n\times n}, \|AB\| \leq \|A\| \ \|B\|$, we have:

$$\|AA^{-1}x\| \leq \|Ax\| \ \|A^{-1}\| \Leftrightarrow \frac{\|x\|}{\|Ax\|} \leq \|A^{-1}\| \tag{27}$$

□

So using this in eq. (25) , we can write:

$$\kappa \leq \|A\| \cdot \|A^{-1}\| \tag{28}$$

Or:

$$\kappa = \alpha \ \|A\| \cdot \|A^{-1}\| \tag{29}$$

With

$$\alpha = \frac{\|x\|}{\|Ax\|} \cdot \left(\|A^{-1}\|\right)^{-1} \tag{30}$$

We can choose x to make $\alpha = 1$, and therefore $\kappa = \|A\| \cdot \|A^{-1}\|$.

Consider now the problem of calculating $A^{-1}b$ given A, mathematically it is identical to the problem we just considered, so the following has already been proven:

Theorem 1.2.3.2: Let $A \in C^{n\times n}$, $\det(A) \neq 0$, and consider $Ax = b$, the problem of computing b, perturbating x has conditioning number:

$$\kappa = \|A\| \frac{\|x\|}{\|b\|} \leq \|A\| \cdot \|A^{-1}\| \tag{31}$$

Proof: Read from eq. (24) to eq. (30). □

Finally, $\|A\| \cdot \|A^{-1}\|$ is so useful it has a name: **the condition number of A** (relative to the norm $\| \cdot \|$)

If A is singular, usually we wreite $\kappa(A) = \infty$, notice that if $\| \cdot \| = \| \cdot \|_2$, then $\|A\| = \sigma_1$ and $\|A^{-1}\| = \frac{1}{\sigma_m}$, so:

$$\kappa(A) = \frac{\sigma_1}{\sigma_m} \tag{32}$$

This is very useful, as we show an interesting application:

1.2.4. Application (b)

Still on least squares, using $\| \cdot \|_2$ the conditioning number of eq. (14) is:

$$\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_m} \tag{33}$$

And the singular values σ_i are the square roots of the eigenvalues λ_i of $B = A^T A$ (non-crescent order), so from eq. (13):

$$\begin{aligned} \det(B - \lambda I) = 0 \Leftrightarrow \det\left(\begin{pmatrix} m+1-\lambda & \frac{m+1}{2} \\ \frac{m+1}{2} & \frac{(m+1)(2m+1)}{6} - \lambda \end{pmatrix}\right) = 0 \\ \Leftrightarrow (m+1-\lambda)\left[\frac{(m+1)(2m+1)}{6} - \lambda\right] - \left(\frac{m+1}{2}\right)^2 = 0 \\ \Leftrightarrow \lambda^2 - \frac{(m+1)(8m+1)}{6m}\lambda + \frac{(m+1)^2(m+2)}{12m} = 0 \\ \Leftrightarrow \lambda = \frac{m+1}{12m}\left[(8m+1) \pm \sqrt{52m^2 - 8m + 1}\right] \end{aligned} \tag{34}$$

So the singular values are:

$$\begin{aligned} \sigma_1 = \sqrt{\lambda_1} = \sqrt{\frac{m+1}{12m}\left[(8m+1) + \sqrt{52m^2 - 8m + 1}\right]}, \\ \sigma_2 = \sqrt{\lambda_2} = \sqrt{\frac{m+1}{12m}\left[(8m+1) - \sqrt{52m^2 - 8m + 1}\right]} \end{aligned} \tag{35}$$

And the condition number of the matrix A is:

$$\begin{aligned} \kappa(A) = \frac{\sigma_1}{\sigma_m} = \frac{\sqrt{\frac{m+1}{12m}\left[(8m+1) + \sqrt{52m^2 - 8m + 1}\right]}}{\sqrt{\frac{m+1}{12m}\left[(8m+1) - \sqrt{52m^2 - 8m + 1}\right]}} \\ = \sqrt{\frac{(8m+1) + \sqrt{52m^2 - 8m + 1}}{(8m+1) - \sqrt{52m^2 - 8m + 1}}} \end{aligned} \tag{36}$$

Here are some python examples:

(COLOCA AS PORRA DOS EXEMPLO)

1.3. More Regression: A Polynomial Perspective (c)

1.4. Finding the matrix A through Python (d)

1.5. How Perturbations Affect The Conditioning Number (e)

1.6. Another Set of Points (f)

2. Least Squares Algorithms

2.1. The SVD, QR factorizations and the normal approach (a)

2.2. Testing the Algorithms and analyzing plots (b)

2.3. The polynomial approach: An efficiency analysis (c)