

Assignment 2 - Numerical Linear Algebra

Prof.: Bernardo Freitas Paulo da Costa

TA: Beatriz Lúcia Teixeira de Souza

Student: Arthur Rabello Oliveira

28/04/2025

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et.

Contents

1. Linear Regression: The Least Squares Method	2
1.1. Linear Regression (a)	2
1.2. The Condition number of a matrix	5
1.2.1. The Conditioning number of a problem	5
1.2.2. The relative Conditioning Number	6
1.2.3. Conditioning Number of matrices	7
1.2.4. Application (b)	8
1.3. More Regression: A Polynomial Perspective (c)	9
1.4. How can one find the matrix A, given (m,n)? (d)	10
1.5. How Perturbations Affect The Condition Number of A (e)	10
1.6. A different dataset	11
1.6.1. A Numerical Approach	11
1.6.2. How Conditioning Changes (f)	11
2. Some Algorithms	11
2.1. The SVD and QR factorizations (a)	11
2.1.1. The factorizations	11
2.1.2. A Numerical Analysis (b)	11
2.2. The polynomial approach: An efficiency analysis (c)	11

1. Linear Regression: The Least Squares Method

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et.

1.1. Linear Regression (a)

We have a set of equally spaced points $S := \{t_i = \frac{i}{m}\}, i = 0, 1, \dots, m$, 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:

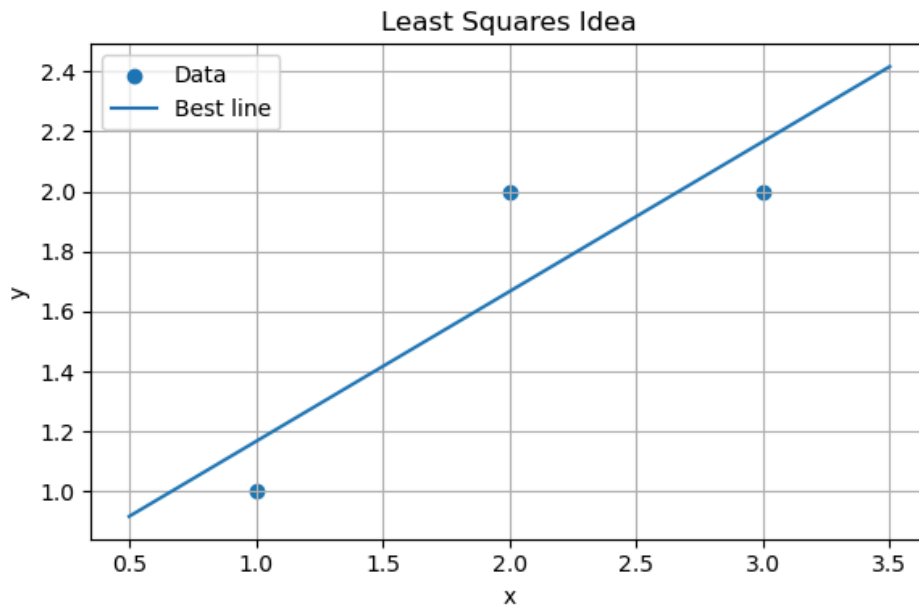


Figure 1: A glimpse into what we want to see

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:

$$f(1) = \alpha + \beta = 1$$

$$f(2) = \alpha + 2\beta = 2$$

$$f(3) = \alpha + 3\beta = 2$$

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:

$$\underbrace{\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} \alpha \\ \beta \end{pmatrix}}_x = \underbrace{\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}}_b \quad 2$$

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:

$$e_1^2 + e_2^2 + e_3^2 \quad 3$$

Is what we want to minimize, where e_i is the error (distance) from the i th 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:

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

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

$$\begin{aligned} 3\alpha + 5\beta &= 5 \\ 6\alpha + 14\beta &= 11 \end{aligned} \quad 5$$

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

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

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

$$\begin{aligned}
f(\alpha, \beta) &= (\alpha + \beta - 1)^2 + (\alpha + 2\beta - 2)^2 + (\alpha + 3\beta - 2)^2 \\
&= 3\alpha^2 + 14\beta^2 + 12\alpha\beta - 10\alpha - 22\beta + 9, \\
\frac{\partial f}{\partial \alpha} = \frac{\partial f}{\partial \beta} = 0 &\Leftrightarrow 6\alpha + 12\beta - 10 = 28\beta + 12\alpha - 22 = 0 \Leftrightarrow \begin{cases} 3\alpha + 6\beta = 5 \\ 6\alpha + 14\beta = 11 \end{cases}
\end{aligned}
\tag{7}$$

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:

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

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

With $S := \{t_i = \frac{i}{m}, i = 0, 1, \dots, m\}$, 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:

$$f(t) = \alpha + \beta t \tag{9}$$

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

$$\begin{aligned}
f(0) &= b_0 = \alpha, \\
f\left(\frac{1}{m}\right) &= b_1 = \alpha + \frac{\beta}{m}, \\
f\left(\frac{2}{m}\right) &= b_2 = \alpha + \frac{2}{m}\beta \\
&\dots \\
f(1) &= b_m = \alpha + \beta
\end{aligned}
\tag{10}$$

And the $Ax = b$ matrices alternative:

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 1 & \frac{1}{m} \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix}}_A \cdot \underbrace{\begin{pmatrix} \alpha \\ \beta \end{pmatrix}}_x = \underbrace{\begin{pmatrix} b_0 \\ \vdots \\ b_m \end{pmatrix}}_b
\tag{11}$$

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

$$\begin{aligned}
A^T A x &= A^T b \\
&= \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & \frac{1}{m} & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & \frac{1}{m} \\ \vdots & \vdots \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} m+1 & \frac{m+1}{2} \\ \frac{m+1}{2} & \frac{(m+1)(2m+2)}{6m} \end{pmatrix} \cdot \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 1 & \dots & 1 \\ 0 & \frac{1}{m} & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} b_0 + b_2 + \dots + b_m \\ \frac{1}{m}[b_1 + 2b_2 + \dots + (m-1)b_{m-1} + b_m] \end{pmatrix}
\end{aligned} \tag{12}$$

So the new system to be solved is:

$$\begin{pmatrix} m+1 & \frac{m+1}{2} \\ \frac{m+1}{2} & \frac{(m+1)(2m+2)}{6m} \end{pmatrix} \cdot \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} b_0 + b_1 + \dots + b_m \\ \frac{1}{m}[b_1 + 2b_2 + \dots + (m-1)b_{m-1} + b_m] \end{pmatrix} \tag{13}$$

Or:

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

And the least squares optimal solution is:

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} m+1 & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & \sum_{i=1}^m t_i^2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \sum_{i=1}^m b_i \\ \sum_{i=1}^m \frac{i}{m} \cdot b_i \end{pmatrix} \tag{15}$$

Now we will discuss if this method is efficient, we will use **Conditioning Numbers** as

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 $f : X \rightarrow Y$ from a **normed** vector space X of data (it has to be normed so we 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 $f : X \rightarrow Y$ 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 there is 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 perturbation of x , so $\delta f = f(x + \delta x) - f(x)$. The **absolute** conditioning number of f is:

$$\hat{\kappa} = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| \leq \delta} \frac{\|\delta f\|}{\|\delta x\|} \quad 16$$

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

$$\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|} \quad 17$$

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 $\|\delta x\| \rightarrow 0$. So the absolute conditioning number of f becomes:

$$\hat{\kappa} = \|J(x)\|, \quad 18$$

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 $f : X \rightarrow Y$ a problem, the *relative condition number* $\kappa(x)$ at $x \in X$ is:

$$\kappa(x) = \lim_{\delta \rightarrow 0} \sup_{\|\delta x\| \leq \delta} \left(\frac{\|\delta f\|}{\|f(x)\|} \right) \cdot \left(\frac{\|\delta x\|}{\|x\|} \right)^{-1} \quad 19$$

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

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

If f is differentiable:

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

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 differentiable, so by eq. (21):

$$\kappa(x) = (\|J\|) \cdot \left(\frac{\|f(x)\|}{\|x\|} \right)^{-1} = \left(\frac{1}{2} \right) \cdot \left(\frac{\frac{x}{2}}{x} \right)^{-1} = 1. \quad 22$$

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:

$$J = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \right) = (1 \quad -1) \quad 23$$

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

$$\kappa = (\|J\|_\infty) \cdot \left(\frac{\|f(x)\|}{\|x\|} \right)^{-1} = \frac{2}{|x_1 - x_2| \cdot \max\{|x_1|, |x_2|\}} \quad 24$$

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:

$$\kappa = \sup_{\delta x} \frac{\|A(x + \delta x) - Ax\|}{\|Ax\|} \cdot \left(\frac{\|\delta x\|}{\|x\|} \right)^{-1} = \sup_{\delta x} \frac{\|A\delta x\|}{\|\delta x\|} \cdot \left(\frac{\|Ax\|}{\|x\|} \right)^{-1} \quad 25$$

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

$$\kappa = \|A\| \cdot \frac{\|x\|}{\|Ax\|} \quad 26$$

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:

$$\frac{\|x\|}{\|Ax\|} \leq \|A^{-1}\| \quad 27$$

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

$$\|AA^{-1}x\| \leq \|Ax\| \|A^{-1}\| \Leftrightarrow \frac{\|x\|}{\|Ax\|} \leq \|A^{-1}\| \quad 28$$

□

So using this in eq. (26), we can write:

$$\kappa \leq \|A\| \cdot \|A^{-1}\| \quad 29$$

Or:

$$\kappa = \alpha \|A\| \cdot \|A^{-1}\| \quad 30$$

With

$$\alpha = \frac{\|x\|}{\|Ax\|} \cdot (\|A^{-1}\|)^{-1} \quad 31$$

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 , perturbing x has conditioning number:

$$\kappa = \|A\| \frac{\|x\|}{\|b\|} \leq \|A\| \cdot \|A^{-1}\| \quad 32$$

Proof: Read from eq. (25) to eq. (31). □

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 write $\kappa(A) = \infty$, notice that if $\|\cdot\| = \|\cdot\|_2$, then $\|A\| = \sigma_1$ and $\|A^{-1}\| = \frac{1}{\sigma_m}$, so:

$$\kappa(A) = \frac{\sigma_1}{\sigma_m} \quad 33$$

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:

$$\kappa(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_m} \quad 34$$

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):

$$\begin{aligned} \det(B - \lambda I) = 0 &\Leftrightarrow \det \left(\begin{pmatrix} m+1-\lambda & \frac{m+1}{2} \\ \frac{m+1}{2} & \frac{(m+1)(2m+1)}{6} - \lambda \end{pmatrix} \right) = 0 \\ &\Leftrightarrow (m+1-\lambda) \left[\frac{(m+1)(2m+1)}{6} - \lambda \right] - \left(\frac{m+1}{2} \right)^2 = 0 \\ &\Leftrightarrow \lambda^2 - \frac{(m+1)(8m+1)}{6m} \lambda + \frac{(m+1)^2(m+2)}{12m} = 0 \\ &\Leftrightarrow \lambda = \frac{m+1}{12m} \left[(8m+1) \pm \sqrt{52m^2 - 8m + 1} \right] \end{aligned} \quad 35$$

So the singular values are:

$$\begin{aligned}\sigma_1 &= \sqrt{\lambda_1} = \sqrt{\frac{m+1}{12m} \left[(8m+1) + \sqrt{52m^2 - 8m + 1} \right]}, \\ \sigma_2 &= \sqrt{\lambda_2} = \sqrt{\frac{m+1}{12m} \left[(8m+1) - \sqrt{52m^2 - 8m + 1} \right]}\end{aligned}\tag{36}$$

And the condition number of the matrix A is:

$$\begin{aligned}\kappa(A) &= \frac{\sigma_1}{\sigma_m} = \frac{\sqrt{\frac{m+1}{12m} \left[(8m+1) + \sqrt{52m^2 - 8m + 1} \right]}}{\sqrt{\frac{m+1}{12m} \left[(8m+1) - \sqrt{52m^2 - 8m + 1} \right]}} \\ &= \sqrt{\frac{(8m+1) + \sqrt{52m^2 - 8m + 1}}{(8m+1) - \sqrt{52m^2 - 8m + 1}}}\end{aligned}\tag{37}$$

When $m \rightarrow \infty$: COLOCAR O LIMITE

Here are some python examples:

(COLOCA AS PORRA DOS EXEMPLO)

1.3. More Regression: A Polynomial Perspective (c)

In this section we will discuss what changes when we decide to use **polynomials** instead of **lines** to approximate our dataset (still on least squares):

$$f(t) = \alpha + \beta t \rightarrow p(t) = \varphi_0 + \varphi_1 t + \dots + \varphi_n t^n\tag{38}$$

From a first perspective, it seems way more efficient to describe a dataset with many variables than with a simple line $\alpha + \beta t$, so let's use the same dataset $S := \{(t_i, b_i), t_i = \frac{i}{m}\}, i = 0, 1, \dots, m$.

Where b_i is arbitrary, as we did in Section 1.1, finding the new system to be solved:

$$\begin{aligned}p(t_0 = 0) &= b_0 = \varphi_0, \\ p\left(t_1 = \frac{1}{m}\right) &= b_1 = \varphi_0 + \varphi_1 \frac{1}{m} + \dots + \varphi_n \left(\frac{1}{m}\right)^n \\ p\left(t_2 = \frac{2}{m}\right) &= b_2 = \varphi_0 + \varphi_1 \frac{2}{m} + \varphi_2 \left(\frac{2}{m}\right)^2 + \dots + \varphi_n \left(\frac{2}{m}\right)^n \\ &\vdots \\ p(t_m = 1) &= b_m = \varphi_0 + \dots + \varphi_n\end{aligned}\tag{39}$$

And:

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \frac{1}{m} & \left(\frac{1}{m}\right)^2 & \dots & \left(\frac{1}{m}\right)^n \\ 1 & \frac{2}{m} & \left(\frac{2}{m}\right)^2 & \dots & \left(\frac{2}{m}\right)^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}}_{A_{n+1 \times n+1}} \cdot \underbrace{\begin{pmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_n \end{pmatrix}}_{\Phi_{n+1 \times 1}} = \underbrace{\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}}_{b_{n+1 \times 1}}\tag{40}$$

Projecting into $C(A)$:

$$\begin{aligned}
A^T A \hat{\Phi} &= \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & \frac{1}{m} & \frac{2}{m} & \dots & 1 \\ 0 & (\frac{1}{m})^2 & (\frac{2}{m})^2 & \dots & 1 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & (\frac{1}{m})^n & (\frac{2}{m})^n & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & \frac{1}{m} & (\frac{1}{m})^2 & \dots & (\frac{1}{m})^n \\ 1 & \frac{2}{m} & (\frac{2}{m})^2 & \dots & (\frac{2}{m})^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} \hat{\varphi}_0 \\ \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_n \end{pmatrix} \\
&= \begin{pmatrix} n+1 & \sum_{i=1}^m \frac{i}{m} & \sum_{i=1}^m (\frac{i}{m})^2 & \dots & \sum_{i=1}^m (\frac{i}{m})^n \\ \sum_{i=1}^m \frac{i}{m} & \sum_{i=1}^m (\frac{i}{m})^2 & \sum_{i=1}^m (\frac{i}{m})^3 & \dots & \sum_{i=1}^m (\frac{i}{m})^{n+1} \\ \sum_{i=1}^m (\frac{i}{m})^2 & \sum_{i=1}^m (\frac{i}{m})^3 & \sum_{i=1}^m (\frac{i}{m})^4 & \dots & \sum_{i=1}^m (\frac{i}{m})^n + 2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \sum_{i=1}^m (\frac{i}{m})^n & \sum_{i=1}^m (\frac{i}{m})^{n+1} & \sum_{i=1}^m (\frac{i}{m})^{n+2} & \dots & \sum_{i=1}^m (\frac{i}{m})^{2n} \end{pmatrix} \cdot \begin{pmatrix} \hat{\varphi}_0 \\ \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_n \end{pmatrix} \\
&= \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & \frac{1}{m} & \frac{2}{m} & \dots & 1 \\ 0 & (\frac{1}{m})^2 & (\frac{2}{m})^2 & \dots & 1 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & (\frac{1}{m})^n & (\frac{2}{m})^n & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^m b_i \\ \sum_{i=0}^m \frac{ib_i}{m} \\ \sum_{i=0}^m (\frac{i}{m})^2 m \\ \vdots \\ \sum_{i=0}^m (\frac{i}{m})^n b_i \end{pmatrix}
\end{aligned} \tag{41}$$

So the system to be solved is:

$$\begin{pmatrix} n+1 & \sum_{i=1}^m \frac{i}{m} & \dots & \sum_{i=1}^m (\frac{i}{m})^n \\ \sum_{i=1}^m \frac{i}{m} & \sum_{i=1}^m (\frac{i}{m})^2 & \dots & \sum_{i=1}^m (\frac{i}{m})^{n+1} \\ \sum_{i=1}^m (\frac{i}{m})^2 & \sum_{i=1}^m (\frac{i}{m})^3 & \dots & \sum_{i=1}^m (\frac{i}{m})^n + 2 \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^m (\frac{i}{m})^n & \sum_{i=1}^m (\frac{i}{m})^{n+1} & \dots & \sum_{i=1}^m (\frac{i}{m})^{2n} \end{pmatrix} \cdot \begin{pmatrix} \hat{\varphi}_0 \\ \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_n \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^m b_i \\ \sum_{i=0}^m \frac{ib_i}{m} \\ \sum_{i=0}^m (\frac{i}{m})^2 m \\ \vdots \\ \sum_{i=0}^m (\frac{i}{m})^n b_i \end{pmatrix} \tag{42}$$

Therefore the least squares *polynomial regression* optimal solution is:

$$\begin{pmatrix} \hat{\varphi}_0 \\ \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_n \end{pmatrix} = \begin{pmatrix} n+1 & \sum_{i=1}^m \frac{i}{m} & \dots & \sum_{i=1}^m (\frac{i}{m})^n \\ \sum_{i=1}^m \frac{i}{m} & \sum_{i=1}^m (\frac{i}{m})^2 & \dots & \sum_{i=1}^m (\frac{i}{m})^{n+1} \\ \sum_{i=1}^m (\frac{i}{m})^2 & \sum_{i=1}^m (\frac{i}{m})^3 & \dots & \sum_{i=1}^m (\frac{i}{m})^n + 2 \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^m (\frac{i}{m})^n & \sum_{i=1}^m (\frac{i}{m})^{n+1} & \dots & \sum_{i=1}^m (\frac{i}{m})^{2n} \end{pmatrix}^{-1} \cdot \begin{pmatrix} \sum_{i=0}^m b_i \\ \sum_{i=0}^m \frac{ib_i}{m} \\ \sum_{i=0}^m (\frac{i}{m})^2 m \\ \vdots \\ \sum_{i=0}^m (\frac{i}{m})^n b_i \end{pmatrix} \tag{43}$$

1.4. How can one find the matrix A, given (m,n)? (d)

Here we show a python-implemented function that calculates the matrix A as a function of the dimensions (m, n) :

BOTAR A FUNÇÃO PYTHON

1.5. How Perturbations Affect The Condition Number of A (e)

In this section we analyze what happens to $\kappa(A)$, when A is perturbed with $m = 100$ and $n = 1, \dots, 20$

1.6. A different dataset

1.6.1. A Numerical Approach

1.6.2. How Conditioning Changes (f)

2. Some Algorithms

2.1. The SVD and QR factorizations (a)

2.1.1. The factorizations

2.1.2. A Numerical Analysis (b)

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