Equation solving

marco milanesio
MScDSAI

UNIVERSITÉ CÔTE D'AZUR

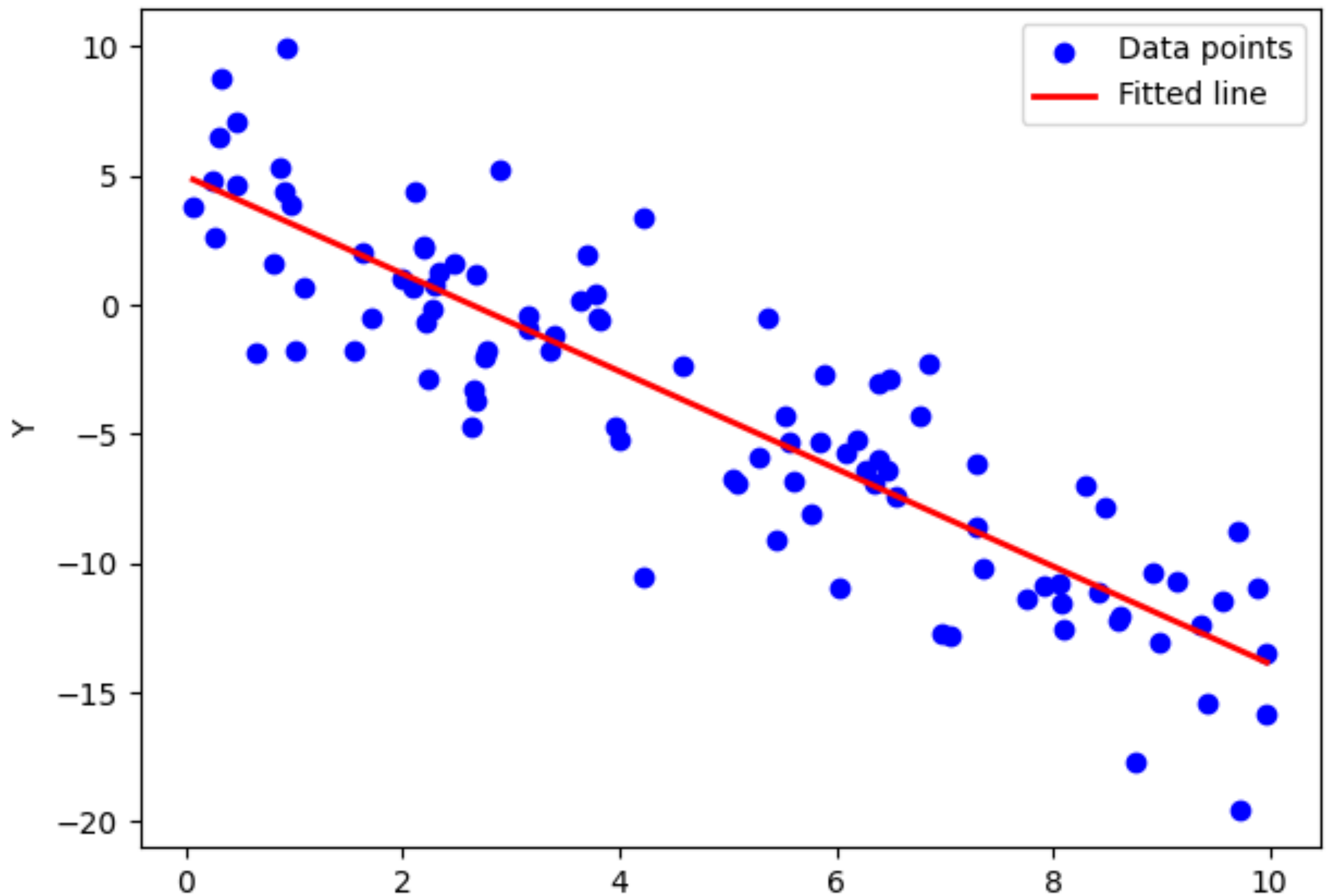
Linear regression

- Models the relation between one or more independent variables (X) and a dependent variable (y)
- i.e., find F s.t. F(X) = y
- Examples
  - X = [age]
  - y = blood pressure
  - $F(X) = a_0 + a_1 * X + \epsilon$
- Minimize the error

Linear regression

$$\sum_i \epsilon_i = 0$$

$$min(\sum_i \epsilon_i^2)$$



Matrix formulation

$$y = Xb \qquad b = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$$

- X and y are **KNOWN**
- Overdetermined problem with multiple inconsistent solutions
  - Multiple possible values for the coefficients
  - More equations than unknown values

Matrix formulation

- BUT:
  - unique solution by minimizing the square of the error

$$(\hat{y} - y)^2 = (Xb - y)^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{i,j}, b_j - y_i)^2$$

$$Xb = y \rightarrow X^T X b = X^T y \rightarrow b = (X^T X)^{-1} X^T y$$

- Why not: $b = X^{-1} y$??

Inverse of a matrix

$$b = (X^T X)^{-1} X^T y$$

- This is the solution
  - We need to transpose a matrix (easy)
  - We need to inverse a matrix (hard)
    - GAUSSIAN ELIMINATION PROCESS

Inverse of a matrix: example

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Inverse of A   adjoint of A

$$A^{-1} = \frac{1}{det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

determinant of A = ad - bc ***

$$A = \begin{bmatrix} -1 & 3/2 \\ 1 & -1 \end{bmatrix}$$

$$A^{-1} = \frac{1}{1 - 3/2} \begin{bmatrix} -1 & -3/2 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}$$

Gaussian elimination process

$$A = \begin{bmatrix} -1 & 3/2 \\ 1 & -1 \end{bmatrix}$$

$$\rightarrow \left[\begin{array}{cc|cc} -1 & 3/2 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{array}\right]$$

Augmented matrix

GOAL: convert the left side into the identity matrix via linear combinations

THEN: the right side will be the inverse

Gaussian elimination process

$$A = \begin{bmatrix} -1 & 3/2 \\ 1 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 & 3/2 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix} \begin{matrix} \text{R1} \\ \text{R2} \end{matrix}$$

- R1 = R1 / R1[0] * * $\rightarrow \begin{bmatrix} 1 & -3/2 & -1 & 0 \\ 1 & -1 & 0 & 1 \end{bmatrix}$

- R2 = R2 - ratio * R1 $\rightarrow \begin{bmatrix} 1 & -3/2 & -1 & 0 \\ 0 & 1/2 & 1 & 1 \end{bmatrix}$

Gaussian elimination process (cont)

- R2 = R2 / R2[1] * * $\rightarrow \begin{bmatrix} 1 & -3/2 & -1 & 0 \\ 0 & 1 & 2 & 2 \end{bmatrix}$

- R1 = R1 - ratio * R2 $\rightarrow \begin{bmatrix} 1 & 0 & 2 & 3 \\ 0 & 1 & 2 & 2 \end{bmatrix}$

$$A^{-1} = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}$$

Gaussian elimination process

$$\begin{bmatrix} 5 & 3 & 1 \\ 3 & 9 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 3 & 1 & 1 & 0 & 0 \\ 3 & 9 & 4 & 0 & 1 & 0 \\ 1 & 3 & 5 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0.250 & -0.091 & 0.023 \\ 0 & 1 & 0 & -0.083 & 0.182 & -0.129 \\ 0 & 0 & 1 & 0.000 & -0.091 & 0.273 \end{bmatrix}$$

LAB SESSION

- Implement your own gaussian elimination algorithm (NO NUMPY)
- A matrix is a list of lists
- def augmented_matrix(M): pass
- def inverse(M): pass
- Same process as last week!

LAB SESSION

- Return:
  - 1 python script "gaussian.py" containing the main algorithm
  - 1 python script "main.py" containing a function to hardcode the matrix and call the algorithm
  - 1 text file "yourname_gaussian.txt" containing a brief description of your implementation choices.
- Pack everything in a zip archive "yourname_gaussian.zip" and send email
- Deadline: 21-9-2025 08:00 AM