

Môn học: Thực hành toán ứng dụng thống kê

Báo cáo lab 4

Họ và tên: Bùi Thị Thanh Ngân

MSSV: 21120505

1. Các chức năng đã hoàn thành:

- Kiểm tra tính lồi lõm của hàm số (Bài 1)
- Tìm phương trình tuyến tính bằng phương pháp bình phương tối thiểu (Bài 2)
- Ước lượng các tham số của mô hình thông qua phương pháp bình phương cực tiểu. (Bài 3a)
- Vẽ đồ thị dữ liệu và dự đoán giá trị của y tại điểm x cụ thể (Bài 3b)
- Kiểm tra có nên dùng mô hình khác để xấp xỉ dữ liệu đề bài không (Bài 3c)

2. Ý tưởng thực hiện, mô tả các hàm chức năng, hình ảnh kết quả:

Bài 1:

Solution:

Để kiểm tra tính lồi lõm của hàm f , ta tìm các trị riêng của ma trận A và kiểm tra xem tất cả có không âm hay không như sau:

```
A = np.array([[1,-2,1],[-2,1,-2],[1,-2,1]])
A
```

```
[110] ✓ 0.1s Python
... array([[ 1, -2,  1],
        [-2,  1, -2],
        [ 1, -2,  1]])
```

```
eigenvalues = np.linalg.eig(A)[0]
print(eigenvalues)

rounded_eigenvalue = np.round(eigenvalues, decimals=2)
print(rounded_eigenvalue)
```

```
[111] ✓ 0.1s Python
... [ 4.37228132e+00 -1.11548217e-16 -1.37228132e+00]
     [ 4.37 -0. -1.37]
```

Do các giá trị riêng của A có cả giá trị âm, dương và 0 nên ma trận A là ma trận không xác định. Do đó không thể xác định tính lồi lõm của hàm f và không thể xác định cực trị toàn cục của hàm.

Bài 2:

Solution:

Giả sử phương trình tuyến tính là $y = ax + b$

Ta có bảng dữ liệu :

x	0	1	2	3	4
y	10	8	7	5	2

Ycvt \Leftrightarrow tìm nghiệm bình phương tối thiểu của $Ax = b$ với ma trận A và vector b như sau:

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}, b = \begin{bmatrix} 10 \\ 8 \\ 7 \\ 5 \\ 2 \end{bmatrix}$$

```
A = np.array([[0,1],[1,1],[2,1],[3,1],[4,1]])
b = np.array([[10],[8],[7],[5],[2]])
print(A)
print(b)
```

[166] ✓ 0.7s

Python

```
... [[0 1]
      [1 1]
      [2 1]
      [3 1]
      [4 1]]
      [[10]
      [ 8]
      [ 7]
      [ 5]
      [ 2]]
```

Tính tích $A^T A$ và gán cho ma trận X.

```
X = np.matmul(np.transpose(A), A)
```

[167] ✓ 0.1s

Python

```
... array([[30, 10],
          [10, 5]])
```

Tính tích $A^T b$ và gán cho ma trận Y.

```
Y = np.matmul(np.transpose(A), b)
```

[168] ✓ 0.1s

Python

```
... array([[45],
          [32]])
```

Tính nghiệm bình phương tối thiểu:

```
x_sol_square = np.matmul(np.linalg.inv(X), Y)
x_sol_square
```

[169] ✓ 0.2s

Python

```
... array([[ -1.9],
          [10.2]])
```

Vậy nghiệm bình phương tối thiểu của $Ax = b$ là $x_0 = \left(-\frac{19}{10}, \frac{51}{5}\right)$ \rightarrow Phương trình tuyến tính cần tìm là $y = -\frac{19x}{10} + \frac{51}{5}$

Bài 3:

Solution:

a)

Mô hình $y = a + bx + c \ln(x^2 + 1)$:Giải bài toán tìm nghiệm bình phương cực tiểu $Ax = b$ với:

$$A = \begin{bmatrix} 1 & -2 & \ln((-2)^2 + 1) \\ 1 & 0 & \ln(0^2 + 1) \\ 1 & 1 & \ln(1^2 + 1) \\ 1 & 2 & \ln(2^2 + 1) \\ 1 & 4 & \ln(4^2 + 1) \end{bmatrix}, b = \begin{bmatrix} -1 \\ 1.5 \\ 3.1 \\ 6.3 \\ 11.1 \end{bmatrix}$$

```
x = np.array([[-2],[0],[1],[2],[4]])
x
```

[170]

✓ 0.2s

Python

```
... array([[-2],
         [ 0],
         [ 1],
         [ 2],
         [ 4]])
```

Duyệt qua x và tạo ma trận A với số hàng bằng số hàng của x và 3 cột:

- Cột thứ nhất là hệ số của a : 1
- Cột thứ hai là hệ số của b : x
- Cột thứ ba là hệ số của c : $\ln(x^2 + 1)$

```
# số hàng của x
nr = len(x)
print("Số hàng: ", nr)

# khởi tạo ma trận nr hàng, 3 cột
A = np.zeros((nr, 3))
print("Sau khi khởi tạo ma trận A: ")
print(A)

for i in range(nr):
    A[i][0] = 1
    A[i][1] = x[i]
    A[i][2] = math.log(x[i] ** 2 + 1)
print("Ma trận A: ")
print(A)
```

[171]

✓ 0.1s

Python

```
... Số hàng: 5
Sau khi khởi tạo ma trận A:
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
Ma trận A:
[[ 1. -2.  1.60943791]
 [ 1.  0.  0.        ]
 [ 1.  1.  0.69314718]
 [ 1.  2.  1.60943791]
 [ 1.  4.  2.83321334]]
```

```
b = np.array([[-1],[1.5],[3.1],[6.3],[11.1]])
print(b)
```

[172]

Python

```
... [[-1. ]
     [ 1.5]
     [ 3.1]
     [ 6.3]
     [11.1]]
```

Tính tích $A^T A$ và gán cho ma trận X:

```
X = np.matmul(np.transpose(A), A)
X
```

[173] Python

```
... array([[ 5.          ,  5.          ,  6.74523635],
          [ 5.          , 25.          , 12.02600056],
          [ 6.74523635, 12.02600056, 13.68813165]])
```

Tính tích $A^T b$ và gán cho ma trận Y

```
Y = np.matmul(np.transpose(A), b)
Y
```

[174] Python

```
... array([[21.          ],
          [62.1          ],
          [42.12744531]])
```

Tính nghiệm bình phương tối tiểu:

```
x_sol_square = np.matmul(np.linalg.inv(X), Y)
x_sol_square
```

[175] Python

```
... array([[1.14446483],
          [1.81151861],
          [0.92214453]])
```

Vậy nghiệm bình phương tối tiểu của $Ax = b$ là $x = (1.14446483, 1.81151861, 0.92214453)$

⇒ Các tham số:

$$a \simeq 1.14446483$$

$$b \simeq 1.81151861$$

$$c \simeq 0.92214453$$

b) Vẽ đồ thị cho dữ liệu:

```
# Dữ liệu đã được gán phía trên, x -> x, y -> b

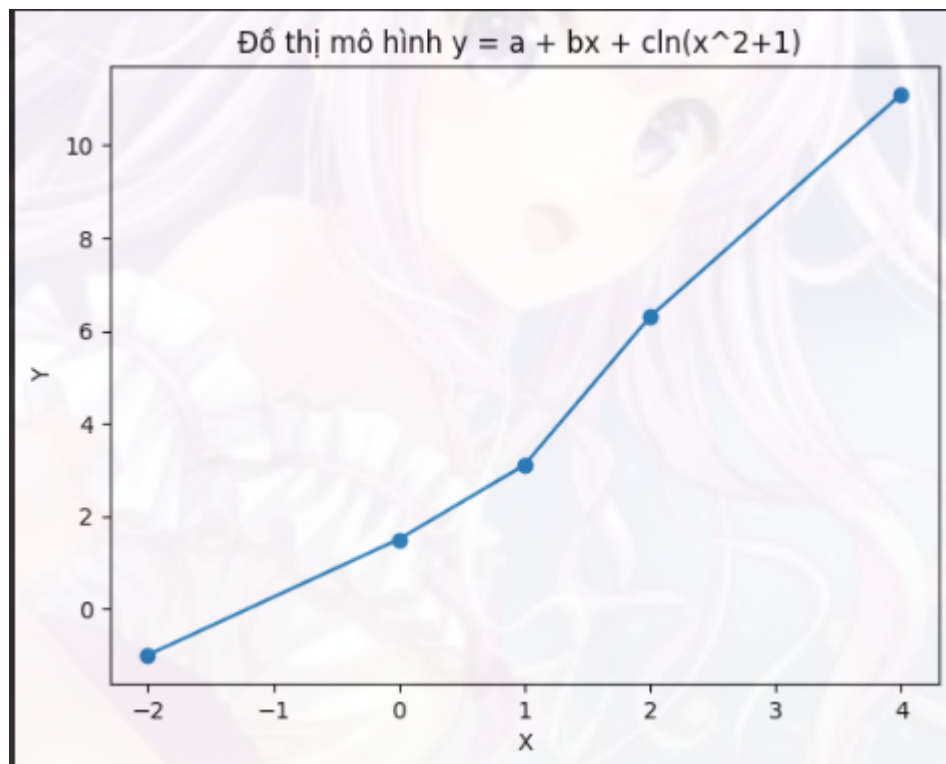
# Vẽ đồ thị
plt.plot(x, b, label='Đường thẳng')
plt.scatter(x, b, label='Dữ liệu', marker='o')

# Đặt tên cho trục x và trục y
plt.xlabel('X')
plt.ylabel('Y')

# Đặt tiêu đề cho đồ thị
title = 'Đồ thị mô hình y = a + bx + c ln(x^2+1)'
plt.title(title)

# Hiện thị đồ thị
plt.show()
```

[176] Python



- Dự đoán y khi x = 6.5:

```
x1 = 6.5
res = x_sol_square[0] + x_sol_square[1] * x1 + x_sol_square[2] * math.log(x1 ** 2 + 1)
print("Khi x = 6.5 : y = ", res)
```

[177]

Python

```
... Khi x = 6.5 : y = [16.39305168]
```

c) Có nên dùng mô hình $y = a + bx + c \ln(x)$ hoặc $y = a + bx + \frac{c}{x}$ để xấp xỉ dữ liệu trên không? Vì sao?

Ta thấy, số liệu thí nghiệm có các giá trị x (tính bằng năm) có thể âm, dương và bằng 0.

- Nếu dùng mô hình $y = a + bx + c \ln(x)$ để xấp xỉ dữ liệu trên thì tại các giá trị $x \leq 0$, giá trị của y sẽ không xác định. Vì thế ta không nên dùng mô hình này để xấp xỉ dữ liệu trên.
- Nếu dùng mô hình $y = a + bx + \frac{c}{x}$ để xấp xỉ dữ liệu trên thì tại giá trị $x = 0$, giá trị của y sẽ không xác định, vì thế ta không nên dùng mô hình này để xấp xỉ dữ liệu trên.