

# Thực hành Nguyên Lý Máy Học

## Buổi 4: hồi quy tuyến tính

### Mục tiêu:

- Củng cố lý thuyết bài toán hồi quy tuyến tính
- Cài đặt giải thuật hồi quy tuyến tính bằng phương pháp giảm gradient.
- Kiểm thử và đánh giá

### A. HƯỚNG DẪN THỰC HÀNH

#### 1. Ví dụ dự đoán giá nhà (bài tập ví dụ trên lớp)

Cho tập dữ liệu gồm 3 phần tử như bảng bên dưới,

<b>X</b>	1	2	4
<b>Y</b>	2	3	6

Anh/chị hãy thực hiện các yêu cầu sau:

- Biểu diễn tập dữ liệu lên mặt phẳng tọa độ Oxy
- Tìm hàm hồi quy  $h(x)$  với giá trị khởi tạo  $\theta_0=0$ ,  $\theta_1=1$ , tốc độ học: 0.2, số bước lặp: 2
- Vẽ đường hồi quy lên mặt phẳng tọa độ
- Dự đoán giá trị  $y$  cho các phần tử có  $x$  có giá trị lần lượt là 0, 3, 5

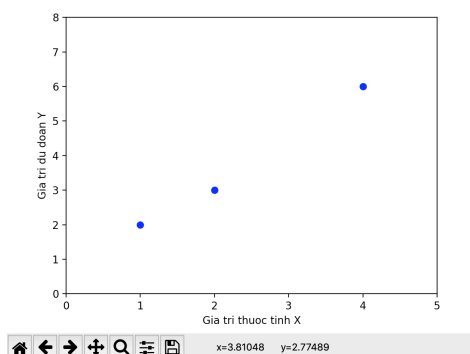
### Hướng dẫn

#### a. Biểu diễn dữ liệu lên mặt phẳng tọa độ

```
import numpy as np
import matplotlib.pyplot as plt

X = np.array([1,2,4])
Y = np.array([2,3,6])

plt.axis([0,5,0,8])
plt.plot(X,Y,"ro",color="blue")
plt.xlabel("Giá trị thuộc tính X")
plt.ylabel("Giá trị dự đoán Y")
plt.show()
```



- b. Tìm hàm hồi quy với  $\theta_0 = 0$ ,  $\theta_1 = 1$ , tốc độ học = 0.2, số lần lặp là 1
- $$\text{for } i=1 \text{ to } m, \{$$
- $$\theta_j := \theta_j + \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$
- $$\}$$

```
def LR1(X,Y,eta,lanlap, theta0,theta1):
    m = len(X) # so luong phan tu
    for i in range(0,lanlap):
        print("Lan lap: ", i)
        for j in range(0,m):
            #theta0
            h= theta0 + theta1*X[j]
            theta0 = theta0 + eta*(Y[j]-h)*1
            print ("Phan tu ", j, "y=", Y[j], "h=",h,"gia tri theta0 = ",theta0)
            #theta1
            theta1 = theta1 + eta*(Y[j]-h)*X[j]
            print ("Phan tu ", j, "gia tri theta1 = ",theta1)
        return [theta0,theta1]

theta = LR1(X,Y,0.2,1,0,1)
theta
```

Kết quả cho 1 lần lặp :  $\theta = \text{LR1}(X,Y,0.2,1,0,1)$

$\theta$   
[0.3360000000000001, 1.584]

Kết quả cho 2 lần lặp :  $\theta_2 = \text{LR1}(X,Y,0.2,2,0,1)$

$\theta$   
[0.28992000000000007, 1.57248]

- c. Vẽ đường hồi quy

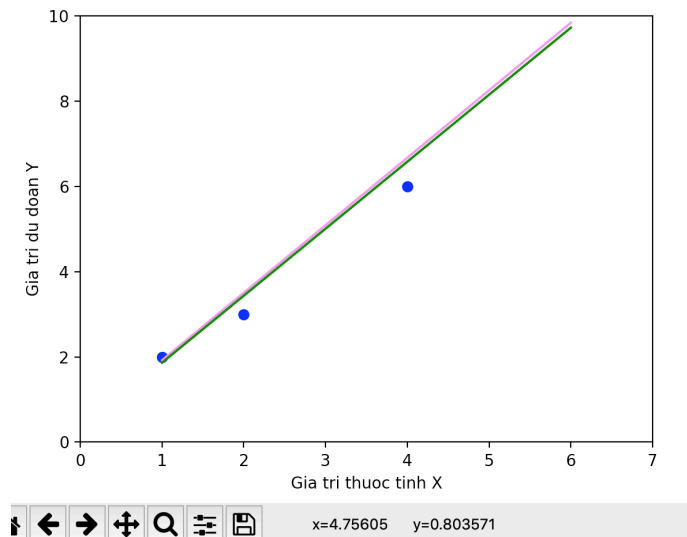
```
theta = LR1(X,Y,0.2,1,0,1) # theta 1 bước
X1= np.array([1,6])
Y1= theta[0] + theta[1]*X1

theta2 = LR1(X,Y,0.2,2,0,1) # theta 2 bước lap
X2= np.array([1,6])
Y2= theta2[0] + theta2[1]*X2

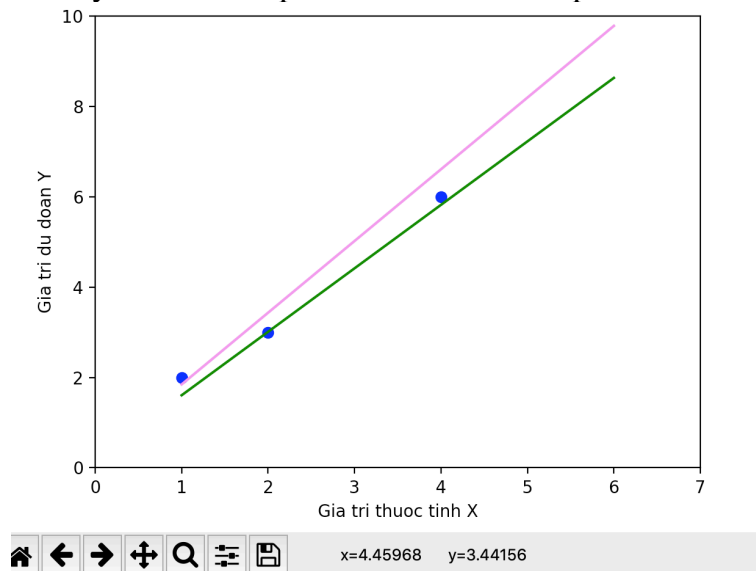
plt.axis([0,7,0,10])
plt.plot(X,Y,"ro",color="blue")

plt.plot(X1,Y1,color="violet") # duong hoi quy lan lap 1
plt.plot(X2,Y2,color="green") # duong hoi quy lan lap 2

plt.xlabel("Gia tri thuoc tinh X")
plt.ylabel("Gia tri du doan Y")
plt.show()
```



- d. Thay đổi tốc độ học bằng 0.1, anh/chị vẽ đường hồi quy màu hồng cho 1 lần lặp và màu xanh lá cây cho 2 lần lặp và so sánh với kết quả ở bước câu c.



- e. Dự báo cho phần tử mới tới  
Dự báo giá trị y cho 3 phần tử sau:  $x=0$ ,  $x=3$ ,  $x=5$

```
# Dự báo
y1 = theta[0] + theta[1]*0
y2 = theta[0] + theta[1]*3
y3 = theta[0] + theta[1]*5
```

Hoặc sử dụng vòng lặp for

```
# Dự báo
XX = [0,3,5]
for i in range(0,3):
    YY = theta[0] + theta[1]*XX[i]
    print (round(YY,3))
```

Kết quả dự 1 lần lặp:

0.336  
5.088  
8.256

Kết quả dự đoán 2 lần lặp:

0.29

5.007

8.152

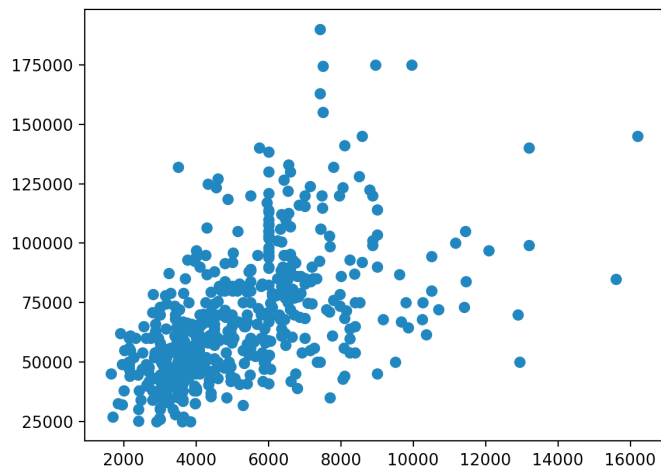
## 2. Sử dụng thư viện scikit-learn của Python để tìm các giá trị theta

- Đọc dữ liệu từ file csv

```
# đọc dữ liệu từ file Housing.csv
import pandas as pd
dt = pd.read_csv("Housing_2019.csv", index_col=0)
dt.iloc[2:4,]
X= dt.iloc[:,[1,2,3,4,10]]
X.iloc[1:5,]
Y = dt.price
```

- Hiển thị dữ liệu

```
plt.scatter(dt.lotsize,dt.price)
plt.show()
```



- Sử dụng sklearn để tìm các giá trị theta, sinh viên thực hành đoạn code bên dưới và trả lời các câu hỏi sau:
  - Có bao nhiêu thuộc tính, đó là những thuộc tính nào đã được sử dụng để dự đoán giá nhà?
  - Xác định số lượng theta và các giá trị của nó.
  - Dữ liệu được sử dụng để huấn luyện mô hình?
  - Dữ liệu được sử dụng để dự báo mô hình?
  - Độ chính xác được đánh giá bằng chỉ số gì và giá trị của nó?

```

# đọc dữ liệu từ file Housing.csv
import pandas as pd
dt = pd.read_csv("Housing_2019.csv", index_col=0)
dt.iloc[2:4,]
X= dt.iloc[:,[1,2,3,4,10]]
X.iloc[1:5,]
y = dt.price
plt.scatter(dt.lotsize,dt.price)
plt.show()

# huấn luyện mô hình
import sklearn
from sklearn import linear_model
lm = linear_model.LinearRegression()
lm.fit(X[0:520],y[0:520])

print lm.intercept_
print lm.coef_

# dự báo giá nhà cho 20 phần tử cuối cùng trong tập dữ liệu
y = dt.price
y_test = y[-20:]
X_test = X[-20:]
y_pred = lm.predict(X_test)

# so sánh giá trị thực tế và giá trị dự báo
y_pred
y_test

from sklearn.metrics import mean_squared_error
err = mean_squared_error(y_test, y_pred)
err
rmse_err = np.sqrt(err)
round(rmse_err,3)

```