Linear Regression

# Giới Thiệu

# Phân tích toán học

## Dạng của Linear Regression

Trong phương trình (1) phía trên, nếu chúng ta đặt là vector (cột) hệ số cần phải tối ưu và  (đọc là x bar trong tiếng Anh) là vector (hàng) dữ liệu đầu vào mở rộng. Số 11 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:

## Sai số dự đoán

Chúng ta mong muốn rằng sừ sai khác giữa giá trị thực và giá trị dự đoán là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt:

Trong đó hệ số (lại) để thuận tiện cho việc tính toán(khi tính đạo hàm thì số sẽ bị triệt tiêu). Chúng ta cần vì có thể là một số âm, việc nói e nhỏ nhất sẽ không đúng vì khi là rất nhỏ nhưng sự sai lệch là rất lớn.

## Hàm mất mát

Điều tương tự xảy ra với tất cả các cặp (input, output) , với N là số lượng dữ liệu quan sát được. Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm để hàm số sau đạt giá trị nhỏ nhất:

Hàm số được gọi là hàm mất mát (loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai só) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point), ký hiệu:

Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát . Đặt là một vector cột chứa tất cả các output của training data; là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hang của nó là một điểm dữ liệu. Khi đó hàm số mất mát được viết dưới dạng ma trận đơn giản hơn:

Với là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác là tổng bình phương mỗi phần tử của vector . Tới đây, ta đã có một dạng đơn giản của hàm mất mát được viết như phương trình .

## Nghiệm cho bài toán Linear Regression

Cách phổ biến để tìm nghiệm cho một bài toán tối ưu (chúng ta đã biết từ khi học cấp 3) là giải phương trình đạo hàm (gradient) bằng 0! Tất nhiên đó là khi việc tính đạo hàm và việc giải phương trình đạo hàm bằng 0 không quá phức tạp. Thật may mắn, với mô hình tuyến tính, hai việc này là khả thi.

Đạo hàm theo của hàm mất mát là:

Phương trình đạo hàm bằng 0 tương đương với:

Với khái niệm giả nghịch đảo, điểm tối ưu của bài toán Linear Regression có dạng:

)

# Ví dụ trên python

Linear Regression đơn giản là cách tiếp cận để dự đoán response sử dụng single feature.

## Simple Linear Regression

### Bài toán

Giả sử rằng hai biến có quan hệ tuyến tính. Do đó, ta thử tìm một hàm tuyến tính để dự đoán giá trị (y) chính xác nhất có thể như một hàm của biến hoặc biến độc lập(x).

Chúng ta hãy xem xét một tập dữ liệu trong đó chúng ta có một giá trị x là chiều cao và y là cân nặng cần dự đoán như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Chiều cao(cm) | Cân nặng(kg) | Chiều cao(cm) | Cân nặng(kg) |
| 147 | 49 | 168 | 60 |
| 150 | 50 | 170 | 72 |
| 153 | 51 | 173 | 63 |
| 155 | 52 | 175 | 64 |
| 158 | 54 | 178 | 66 |
| 160 | 56 | 180 | 67 |
| 163 | 58 | 183 | 68 |
| 165 | 59 |  |  |

Bài toán đặt ra là: liệu có thể dự đoán cân nặng của một người dựa vào chiều cao của họ không? (Trên thực tế, tất nhiên là không, vì cân nặng còn phụ thuộc vào nhiều yếu tố khác nữa, thể tích chẳng hạn). Vì blog này nói về các thuật toán Machine Learning đơn giản nên tôi sẽ giả sử rằng chúng ta có thể dự đoán được.

Chúng ta có thể thấy là cân nặng sẽ tỉ lệ thuận với chiều cao (càng cao càng nặng), nên có thể sử dụng Linear Regression model cho việc dự đoán này. Để kiểm tra độ chính xác của model tìm được, chúng ta sẽ giữ lại cột 155 và 160 cm để kiểm thử, các cột còn lại được sử dụng để huấn luyện (train) model.

Ta cần định nghĩa:

X là feature vector ,

y là response vector ,

cho n quan sát(trong ví dụ trên, n=15)

### Hiển thị dữ liệu trên đồ thị

Chúng ta cần các thư viện cần thiết. numpy cho đại số tuyến tính và matplotlib cho việc vẽ hình.

*# To support both python 2 and python 3*

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

Tiếp theo, chúng ta khai báo và biểu diễn dữ liệu trên đồ thị

*# height (cm)*

X **=** np**.**array([[147, 150, 153, 158, 163, 165, 168, 170, 173, 175, 178, 180, 183]])**.**T

*# weight (kg)*

y **=** np**.**array([[ 49, 50, 51, 54, 58, 59, 60, 62, 63, 64, 66, 67, 68]])**.**T

*# Visualize data*

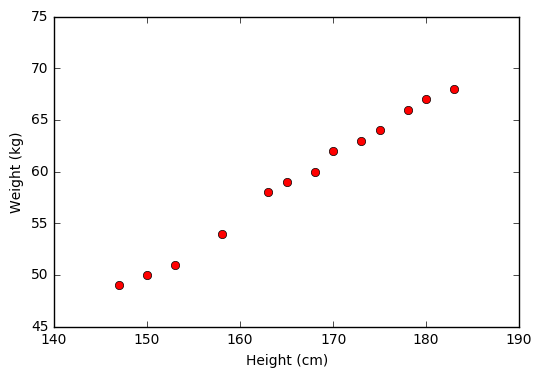
plt**.**plot(X, y, 'ro')

plt**.**axis([140, 190, 45, 75])

plt**.**xlabel('Height (cm)')

plt**.**ylabel('Weight (kg)')

plt**.**show()



Từ đồ thị này ta thấy rằng dữ liệu được sắp xếp gần như theo 1 đường thẳng, vậy mô hình Linear Regression nhiều khả năng sẽ cho kết quả tốt:

(cân nặng) = w\_1\*(chiều cao) + w\_0

### Nghiệm của bài toán

## Multiple linear regression