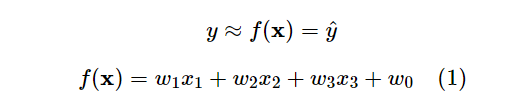
Quay lại [ví dụ đơn giản được nêu trong bài trước](https://machinelearningcoban.com/2016/12/27/categories/#regression-hoi-quy): một căn nhà rộng x1 m2x1 m2, có x2x2 phòng ngủ và cách trung tâm thành phố x3 kmx3 km có giá là bao nhiêu. Giả sử chúng ta đã có số liệu thống kê từ 1000 căn nhà trong thành phố đó, liệu rằng khi có một căn nhà mới với các thông số về diện tích, số phòng ngủ và khoảng cách tới trung tâm, chúng ta có thể dự đoán được giá của căn nhà đó không? Nếu có thì hàm dự đoán y=f(x)y=f(x) sẽ có dạng như thế nào. Ở đây x=[x1,x2,x3]x=[x1,x2,x3] là một vector hàng chứa thông tin input, yy là một số vô hướng (scalar) biểu diễn output (tức giá của căn nhà trong ví dụ này).

**Lưu ý về ký hiệu toán học:** trong các bài viết của tôi, các số vô hướng được biểu diễn bởi các chữ cái viết ở dạng không in đậm, có thể viết hoa, ví dụ x1,N,y,kx1,N,y,k. Các vector được biểu diễn bằng các chữ cái thường in đậm, ví dụ y,x1y,x1. Các ma trận được biểu diễn bởi các chữ viết hoa in đậm, ví dụ X,Y,WX,Y,W.

Một cách đơn giản nhất, chúng ta có thể thấy rằng: i) diện tích nhà càng lớn thì giá nhà càng cao; ii) số lượng phòng ngủ càng lớn thì giá nhà càng cao; iii) càng xa trung tâm thì giá nhà càng giảm. Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa giá nhà và 3 đại lượng đầu vào là:



trong đó, w1,w2,w3,w0w1,w2,w3,w0 là các hằng số, w0w0 còn được gọi là bias. Mối quan hệ y≈f(x)y≈f(x) bên trên là một mối quan hệ tuyến tính (linear). Bài toán chúng ta đang làm là một bài toán thuộc loại regression. Bài toán đi tìm các hệ số tối ưu {w1,w2,w3,w0}{w1,w2,w3,w0} chính vì vậy được gọi là bài toán Linear Regression.

**Chú ý 1:** yy là giá trị thực của outcome (dựa trên số liệu thống kê chúng ta có trong tập training data), trong khi ^yy^ là giá trị mà mô hình Linear Regression dự đoán được. Nhìn chung, yy và ^yy^ là hai giá trị khác nhau do có sai số mô hình, tuy nhiên, chúng ta mong muốn rằng sự khác nhau này rất nhỏ.

**Chú ý 2:** Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Trong không gian nhiều hơn 3 chiều, khái niệm mặt phẳng không còn phù hợp nữa, thay vào đó, một khái niệm khác ra đời được gọi là siêu mặt phẳng (hyperplane). Các hàm số tuyến tính là các hàm đơn giản nhất, vì chúng thuận tiện trong việc hình dung và tính toán. Chúng ta sẽ được thấy trong các bài viết sau, tuyến tính rất quan trọng và hữu ích trong các bài toán Machine Learning. Kinh nghiệm cá nhân tôi cho thấy, trước khi hiểu được các thuật toán phi tuyến (non-linear, không phẳng), chúng ta cần nắm vững các kỹ thuật cho các mô hình tuyến tính.

Text

Description automatically generated

Chú ý rằng x¯ là một vector hàng. ([Xem thêm về ký hiệu vector hàng và cột tại đây](https://machinelearningcoban.com/math/#luu-y-ve-ky-hieu))

2.2. Sai số dự đoán

Chúng ta mong muốn rằng sự sai khác ee giữa giá trị thực yy và giá trị dự đoán ^yy^ (đọc là *y hat* trong tiếng Anh) là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text, letter

Description automatically generated

3.1. Bài toán

Trong phần này, tôi sẽ chọn một ví dụ đơn giản về việc giải bài toán Linear Regression trong Python. Tôi cũng sẽ so sánh nghiệm của bài toán khi giải theo phương trình (5)(5) và nghiệm tìm được khi dùng thư viện [scikit-learn](http://scikit-learn.org/stable/) của Python. (*Đây là thư viện Machine Learning được sử dụng rộng rãi trong Python*). Trong ví dụ này, dữ liệu đầu vào chỉ có 1 giá trị (1 chiều) để thuận tiện cho việc minh hoạ trong mặt phẳng.

Chúng ta có 1 bảng dữ liệu về chiều cao và cân nặng của 15 người như dưới đây:

| **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.

### 4.2. Hạn chế của Linear Regression

Hạn chế đầu tiên của Linear Regression là nó rất **nhạy cảm với nhiễu** (sensitive to noise). Trong ví dụ về mối quan hệ giữa chiều cao và cân nặng bên trên, nếu có chỉ một cặp dữ liệu nhiễu (150 cm, 90kg) thì kết quả sẽ sai khác đi rất nhiều. Xem hình dưới đây:

Chart, scatter chart

Description automatically generated

Vì vậy, trước khi thực hiện Linear Regression, các nhiễu (outlier) cần phải được loại bỏ. Bước này được gọi là tiền xử lý (pre-processing).

Hạn chế thứ hai của Linear Regression là nó **không biễu diễn được các mô hình phức tạp**. Mặc dù trong phần trên, chúng ta thấy rằng phương pháp này có thể được áp dụng nếu quan hệ giữa outcome và input không nhất thiết phải là tuyến tính, nhưng mối quan hệ này vẫn đơn giản nhiều so với các mô hình thực tế. Hơn nữa, chúng ta sẽ tự hỏi: làm thế nào để xác định được các hàm x21,sin(x2),x1x2x12,sin⁡(x2),x1x2 như ở trên?!

### 4.3. Các phương pháp tối ưu

Linear Regression là một mô hình đơn giản, lời giải cho phương trình đạo hàm bằng 0 cũng khá đơn giản. Trong hầu hết các trường hợp, chúng ta không thể giải được phương trình đạo hàm bằng 0.

Nhưng có một điều chúng ta nên nhớ, **còn tính được đạo hàm là còn có hy vọng**.

## **5. Tài liệu tham khảo**

1. [Linear Regression - Wikipedia](https://en.wikipedia.org/wiki/Linear_regression)
2. [Simple Linear Regression Tutorial for Machine Learning](http://machinelearningmastery.com/simple-linear-regression-tutorial-for-machine-learning/)
3. [Least Squares, Pseudo-Inverses, PCA & SVD](http://www.sci.utah.edu/~gerig/CS6640-F2012/Materials/pseudoinverse-cis61009sl10.pdf)