

MACHINE LEARNING

Cao Văn Chung
cvanchung@hus.edu.vn

Informatics Dept., MIM, HUS, VNU Hanoi

Linear Regression

Regression Analysis

- Linear Regression

- Normal Equation

- QR Decomposition for Normal Equation

- Single variable linear regression model

- Model Evaluation

Example

- Eg 1: Single variable Model

- Eg 1: Multi variables Model

QR decomposition & Pseudo-Inverses

- QR decomposition

- Gram - Schmidt Algorithm

Regression Analysis

- ▶ Phân tích hồi quy nghiên cứu sự phụ thuộc của một biến phản hồi (response variable) vào một hoặc nhiều biến dự báo (predictors).
 - ▶ y : biến trả lời - luôn là biến liên tục;
 - ▶ $\mathbf{x} = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}^d$: các biến dự báo - có thể liên tục hoặc rời rạc.
- ▶ Đây là kỹ thuật rất cơ bản của ngành thống kê toán học.
- ▶ Để hiểu được nhiều mô hình phân loại, dự báo hiện đại, ta cần hiểu rõ các phương pháp phân tích hồi quy cổ điển.
- ▶ Xét trên công thức tiếp cận, có 2 dạng phân tích hồi quy:
 - ▶ hồi quy tuyến tính
 - ▶ hồi quy phi tuyến
- ▶ Phân tích hồi quy đơn: $y \in \mathbb{R}$; phân tích hồi quy bội $y \in \mathbb{R}^c$, $c > 1$.
- ▶ Hồi quy đơn biến: $d = 1$, hồi quy nhiều biến $d > 1$ ($\mathbf{x} \in \mathbb{R}^d$).

Linear Regression

- ▶ Hàm dự báo $h_\theta(x)$ được xấp xỉ bởi một hàm tuyến tính của x :

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d.$$

- ▶ Nếu bổ sung thêm đặc trưng cố định $x_0 \equiv 1$ thì ta có thể biểu diễn h_θ dưới dạng

$$h_\theta(x) = \sum_{i=0}^d \theta_i x_i = \theta^T x.$$

Linear Regression

- ▶ Tập dữ liệu huấn luyện:

$$(x^1, y_1), (x^2, y_2), \dots, (x^N, y_N)$$

- ▶ Để ước lượng tham số $\theta \in \mathbb{R}^{(d+1)}$, ta cực tiểu hóa sai số của mô hình trên tập dữ liệu huấn luyện:

$$J(\theta) = \frac{1}{2} \sum_{n=1}^N (h_{\theta}(x^n) - y_n)^2. \quad (1)$$

- ▶ Phương pháp này được gọi là **bình phương tối thiểu** (OLS - Ordinary Least Squares)

$$J(\theta) \rightarrow \min_{\theta}.$$

Linear Regression

Xác định ước lượng hợp lý cực đại (MLE - tham khảo Phần 1 - Buổi 2)

- ▶ Để dự đoán giá trị y cho mỗi đối tượng x , ta dùng hàm $h_{\theta}(x)$ với sai khác là một nhiễu ngẫu nhiên ϵ :

$$y = h_{\theta}(x) + \epsilon.$$

- ▶ Chúng ta giả thiết ϵ có phân bố chuẩn một chiều

$$\epsilon \sim \mathcal{N}(0, \sigma^2).$$

Linear Regression

- ▶ Hàm mật độ của ϵ :

$$P(\epsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right).$$

- ▶ từ đó suy ra phân bố có điều kiện

$$P(y|x; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - h_\theta(x))^2}{2\sigma^2}\right).$$

Linear Regression

- ▶ Để tìm MLE, ta có Log-Likelihood function của dữ liệu trường hợp này là

$$\begin{aligned}\ell(\theta) &= \sum_{n=1}^N \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{(y_n - h_{\theta}(x^n))^2}{2\sigma^2} \right) \right] \\ &= N \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right) - \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - h_{\theta}(x^n))^2 =: T_1 - T_2.\end{aligned}$$

- ▶ Do T_1 là hằng số, cực đại hóa $\ell(\theta)$ tương đương với cực tiểu hóa

$$T_2 = \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - h_{\theta}(x^n))^2 = J(\theta) \quad - \text{chính là hàm mục tiêu trong (1).}$$

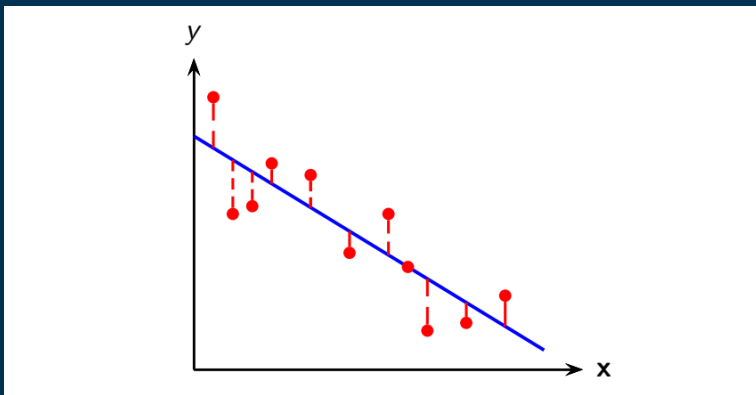
Linear Regression

$J(\theta)$ là sai số trong phương pháp bình phương tối thiểu. Vậy

- ▶ Nếu giả định nhiễu ngẫu nhiên tuân theo phân phối chuẩn thì phương pháp ước lượng hợp lý cực đại dẫn tới phương pháp hồi quy bình phương tối thiểu
- ▶ Ta thấy θ không phụ thuộc vào phương sai σ^2 , ngay cả nếu không biết σ^2 thì ta vẫn ước lượng được θ .

Linear Regression

Trong không gian nhiều chiều chứa $X \times Y \subset \mathbb{R}^d \times \mathbb{R}^n$, ta sử dụng siêu phẳng $h_\theta(x) = \theta^T x$ để dự báo các điểm (x, y) . Trong Machine Learning, $h_\theta(x)$ (hay θ) được xây dựng dựa vào trainingset $\{(x^n, y_n)\}$



Normal Equation

Công thức nghiệm đúng của θ có thể tìm được qua phương trình chuẩn như sau:

- ▶ Với $x^n \in \mathbb{R}^d$, ta có ma trận thiết kế X cỡ $N \times (d + 1)$

$$\mathbf{X} = \begin{pmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^N)^T \end{pmatrix} = \begin{pmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_1^N & x_2^N & \cdots & x_d^N \end{pmatrix}.$$

- ▶ Đặt y là vector cột chứa tất cả các giá trị của biến phản hồi:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}.$$

Normal Equation

Viết lại hàm mục tiêu dưới dạng biểu thức của các ma trận:

$$J(\theta) = \frac{1}{2}(\mathbf{X}\theta - \mathbf{y})^T(\mathbf{X}\theta - \mathbf{y}).$$

Đạo hàm theo θ của $J(\theta)$ là

$$\nabla J(\theta) = \mathbf{X}^T\mathbf{X}\theta - \mathbf{X}^T\mathbf{y}.$$

Để cực tiểu hóa $J(\theta)$, ta tìm θ từ phương trình $\nabla J(\theta) = 0$ và thu được công thức nghiệm đúng cho θ là

$$\hat{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \quad (2)$$

Phương trình (2) trên được gọi là **phương trình chuẩn - Normal Equation**.

Normal Equation

- ▶ Nếu tồn tại nghịch đảo của ma trận $(\mathbf{X}^T \mathbf{X})$ thì ta mới tìm được nghiệm duy nhất θ theo phương trình chuẩn.
- ▶ Nếu nghịch đảo không tồn tại, tức là $(\mathbf{X}^T \mathbf{X})$ không đủ hạng thì ước lượng hồi quy là không duy nhất.
 - ▶ Tồn tại một phụ thuộc tuyến tính giữa các cột của \mathbf{X} .
 - ▶ Ta cần tìm cách giảm số chiều của \mathbf{x} bằng việc loại bỏ các đặc trưng phụ thuộc sao cho ma trận thiết kế là đủ hạng.

Một số tính chất của ước lượng

Ta thấy

$$\begin{aligned}\mathbb{E}(\hat{\theta}|\mathbf{X}) &= \mathbb{E}\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}|\mathbf{X}\right) \\ &= \left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right)\mathbb{E}(\mathbf{y}|\mathbf{X}) \\ &= \left(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\right)\mathbf{X}\theta \\ &= \theta.\end{aligned}$$

Từ đây suy ra $\hat{\theta}$ là ước lượng không chệch của θ .

Một số tính chất của ước lượng

Sử dụng công thức $\text{var}(a + \mathbf{A}\mathbf{y}) = \mathbf{A}\text{var}(\mathbf{y})\mathbf{A}^T$ với a, \mathbf{y} là các vector và \mathbf{A} là ma trận hằng số, ta có phương sai của $\hat{\theta}$ là

$$\begin{aligned}\text{var}(\hat{\theta}|\mathbf{X}) &= \text{var}\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}|\mathbf{X}\right) \\ &= \left[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right] \text{var}(\mathbf{y}|\mathbf{X}) \left[\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\right] \\ &= \left[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\right] \sigma^2\mathbf{I} \left[\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\right] \\ &= \sigma^2 \left[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\right] \\ &= \sigma^2 (\mathbf{X}^T\mathbf{X})^{-1}.\end{aligned}$$

Ta thấy phương sai của $\hat{\theta}$ chỉ phụ thuộc vào \mathbf{X} mà không phụ thuộc vào \mathbf{y} .

Phương pháp QR

- ▶ Trong thực tế, ít khi sử dụng trực tiếp phương trình chuẩn để tính toán $\hat{\theta}$ vì phép lấy nghịch đảo ma trận $(\mathbf{X}^T \mathbf{X})^{-1}$ có thể dẫn tới các sai số làm tròn lớn trong quá trình tính toán.
- ▶ Thay vào đó, hầu hết các phần mềm thống kê sử dụng phương pháp **QR** để tính toán.
- ▶ *Tìm hiểu thêm QR decomposition trong đại số tuyến tính.*

Phương pháp QR

- ▶ Xét lại ma trận thiết kế $\mathbf{X} \in \mathbb{R}^{N \times (d+1)}$. Giả sử ta có thể tìm được một ma trận $\mathbf{Q} \in \mathbb{R}^{N \times (d+1)}$ và một ma trận $\mathbf{R} \in \mathbb{R}^{(d+1) \times (d+1)}$ sao cho.
 - ▶ $\mathbf{X} = \mathbf{QR}$.
 - ▶ \mathbf{Q} là ma trận cột trực chuẩn, tức là $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$.
 - ▶ \mathbf{R} là ma trận tam giác trên, tức là mọi phần tử nằm dưới đường chéo chính đều bằng 0.

Phương pháp QR

Từ tính chất của các ma trận \mathbf{Q} và \mathbf{R} , trong trường hợp \mathbf{R} hạng đủ, ta có

$$\mathbf{X} = \mathbf{QR}$$

$$\mathbf{X}^T \mathbf{X} = (\mathbf{QR})^T (\mathbf{QR}) = \mathbf{R}^T \mathbf{Q}^T \mathbf{QR} = \mathbf{R}^T \mathbf{R}$$

$$(\mathbf{X}^T \mathbf{X})^{-1} = (\mathbf{R}^T \mathbf{R})^{-1} = \mathbf{R}^{-1} (\mathbf{R}^T)^{-1}$$

$$\begin{aligned}\hat{\theta} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{R}^{-1} (\mathbf{R}^T)^{-1} (\mathbf{QR})^T \mathbf{y} \\ &= \mathbf{R}^{-1} (\mathbf{R}^T)^{-1} \mathbf{R}^T \mathbf{Q}^T \mathbf{y} \\ &= \mathbf{R}^{-1} \mathbf{Q}^T \mathbf{y}.\end{aligned}$$

Phương pháp QR

Từ đẳng thức cuối suy ra

$$\mathbf{R}\hat{\theta} = \mathbf{R}\mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y} = \mathbf{Q}^T\mathbf{y}.$$

Do \mathbf{R} là ma trận tam giác trên, ta có thể giải tìm $\hat{\theta}$ dễ dàng phương trình trên bằng phương pháp khử lùi. Ví dụ:

$$\mathbf{R}\hat{\theta} = \begin{pmatrix} 4 & 2 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \hat{\theta}_3 \end{pmatrix} = \mathbf{Q}^T\mathbf{y} = \begin{pmatrix} 4 \\ 11 \\ 8 \end{pmatrix}$$

- ▶ Giải từ dòng dưới lên, phương trình cuối dẫn đến: $\hat{\theta}_3 = 2$;
- ▶ Thay $\hat{\theta}_3$ vào phương trình thứ hai, thu được: $3\hat{\theta}_2 + 2 = 11$ hay $\hat{\theta}_2 = 3$;
- ▶ Thay $\hat{\theta}_2, \hat{\theta}_3$ vào phương trình đầu $4\hat{\theta}_1 + 6 + 2 = 4$ suy ra $\hat{\theta}_1 = -1$.

Single variable Linear Regression

Xét trường hợp $x_n, y_n \in \mathbb{R}$ ($d = n = 1$) - Hồi quy đơn biến, một chiều

$$\mathbb{E}(y|X = x) = h_{\theta}(x) = \theta_0 + \theta_1 x; \quad \text{var}(y|X = x) = \sigma^2.$$

θ_0 gọi là số hạng tự do (số chặn, *intercept*); θ_1 gọi là độ dốc (slope). Kí hiệu:

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x^n \quad \bar{y} = \frac{1}{N} \sum_{n=1}^N y_n$$

$$SXX = \sum_{n=1}^N (x^n - \bar{x})^2 \quad - \text{Phương sai mẫu}$$

$$SXY = \sum_{n=1}^N (x^n - \bar{x})(y_n - \bar{y}) \quad - \text{Hiệp phương sai mẫu.}$$

Single variable Linear Regression

Lúc đó

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

Do vậy

$$(\mathbf{X}^T \mathbf{X}) = \begin{pmatrix} N & \sum_{i=1}^N x_i \\ \sum_{n=1}^N x^n & \sum_{n=1}^N (x^n)^2 \end{pmatrix} \quad \mathbf{X}^T \mathbf{y} = \begin{pmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N y_i x_i \end{pmatrix}$$

Single variable Linear Regression

Kiểm tra trực tiếp sẽ thấy

$$(\mathbf{X}^T \mathbf{X})^{-1} = \frac{1}{SXX} \begin{pmatrix} \frac{1}{N} \sum_{n=1}^N (x^n)^2 & -\bar{x} \\ -\bar{x} & 1 \end{pmatrix}$$

Suy ra

$$\hat{\theta} = \begin{pmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \end{pmatrix} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \begin{pmatrix} \bar{y} - \bar{x} \hat{\theta}_1 \\ SXY / SXX \end{pmatrix}$$

Model Evaluation

- ▶ Các độ đo định lượng đánh giá tính phù hợp của mô hình hồi quy thường dựa vào độ sai lệch giữa đầu ra dự đoán $f_{\theta}(\mathbf{x})$ và đầu ra quan sát y : $|y - f_{\theta}(\mathbf{x})|$.
- ▶ Một số độ đo tuyệt đối (tính trên tập Training hoặc Validation):
 - ▶ Tổng bình phương sai số SSE (Sum Squared Error) hoặc Trung bình bình phương sai số MSE (Mean Squared Error).

$$SSE := \sum_{n=1}^N (y_n - f_{\theta}(\mathbf{x}_n))^2; \quad MSE := \frac{1}{N} \sum_{n=1}^N (y_n - f_{\theta}(\mathbf{x}_n))^2.$$

- ▶ Trung bình tuyệt đối sai số MAE (Mean Absolute Error).

$$MAE := \frac{1}{N} \sum_{n=1}^N |y_n - f_{\theta}(\mathbf{x}_n)|.$$

Model Evaluation

- ▶ Độ đo tuyệt đối không phản ánh được tương quan của sai số so với độ lớn dữ liệu thực.
- ▶ Độ đo tương đối (tỷ đối): R^2 hay R - Square:

$$R - square := 1 - \frac{\sum_{n=1}^N (y_n - f_{\theta}(\mathbf{x}_n))^2}{\sum_{n=1}^N (y_n - \bar{y})^2} = 1 - \frac{SSE}{TSS}.$$

trong đó $\bar{x} = \frac{1}{N} \sum_{n=1}^N x^n$; $\bar{y} = \frac{1}{N} \sum_{n=1}^N y_n$; ($TSS = N \times$ phương sai đầu ra y_n).

- ▶ $R - square \approx 1$: Mô hình phù hợp; $R - square$ nhỏ: Mô hình không phù hợp.

Model Evaluation

Ta có thể đánh giá tiên nghiệm mức độ phụ thuộc tuyến tính của đầu ra y vào đầu vào $x \in \mathbb{R}$ (biến đơn):

- ▶ Hệ số tương quan Pearson (Pearson's Correlation)

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

- ▶ Đánh giá độ phụ thuộc:
 - ▶ $r \approx 1$: (x, y) có sự phụ thuộc tuyến tính đồng biến;
 - ▶ $r \approx -1$: (x, y) có sự phụ thuộc tuyến tính nghịch biến;
 - ▶ $r \approx 0$: (x, y) không có sự phụ thuộc tuyến tính.

Ví dụ 1: Nhiệt độ & áp suất

- ▶ Năm 1857, James D. Forbes (Scotland) thực hiện thí nghiệm tìm hiểu liên hệ giữa áp suất và nhiệt độ sôi của nước.
- ▶ Ông biết rằng có thể xác định độ cao từ áp suất không khí đo bằng áp kế: càng lên cao áp suất càng thấp.
- ▶ Do vào thời đó, áp kế thường không chính xác. Forbes muốn sử dụng nhiệt độ sôi của nước để suy ra áp suất.
- ▶ Forbes thu thập dữ liệu ở các dãy Alps và ở Scotland. Tại mỗi điểm, ông đo áp suất (theo inch thủy ngân) bằng áp kế và đo độ sôi của nước (theo độ Fahrenheit) bằng nhiệt kế.

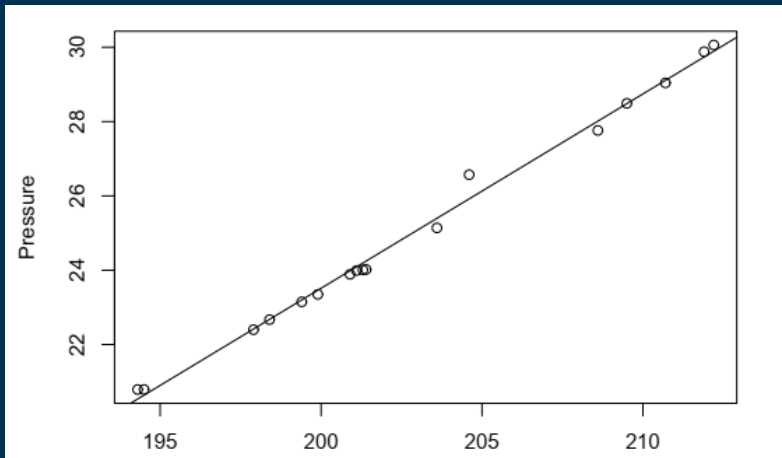
Ví dụ 1: Nhiệt độ & áp suất

Dữ liệu đo ở 17 điểm đo được cho trong bảng sau:

i	Nhiệt độ	Áp suất
1	194.5	20.79
2	194.3	20.79
3	197.9	22.40
4	198.4	22.67
5	199.4	23.15
6	199.9	23.35
7	200.9	23.89
8	201.1	23.99
9	201.4	24.02
10	201.3	24.01
11	203.6	25.14
12	204.6	26.57
13	209.5	28.49
14	208.6	27.76
15	210.7	29.04
16	211.9	29.88
17	212.2	30.06

Ví dụ 1: Nhiệt độ & áp suất

Áp dụng ước lượng hồi quy đơn ta được mô hình $h_{\hat{\theta}}(x) = -81.06373 + 0.52289x$.



Ví dụ 2: Dự đoán tiêu thụ nhiên liệu

Trong ví dụ này, ta sử dụng mô hình hồi quy tuyến tính để

- ▶ Dự báo mức độ tiêu thụ nhiên liệu trong 50 bang của Hoa Kỳ và quận Columbia.
- ▶ Tìm hiểu hiệu ứng của tiêu thụ nhiên liệu đối với thuế xăng của các bang.
- ▶ Các biến dự báo được sử dụng trong ví dụ¹. Dữ liệu được thu thập bởi Cục Đường bộ Hoa Kỳ vào năm 2001.

Drivers	Số bằng lái được cấp phép trong bang
FuelC	Lượng xăng sử dụng cho giao thông đường bộ, theo ngàn gallons
Income	Thu nhập bình quân đầu người năm 2000, theo ngàn đôla
Miles	Số dặm đường cao tốc của bang được hỗ trợ từ liên bang
Pop	Dân số lớn hơn hoặc bằng 16 tuổi
Tax	Thuế xăng của bang, theo cents trên một gallon
State	Tên bang
Fuel	$1000 \times \text{FuelC} / \text{Pop}$
Dlic	$1000 \times \text{Drivers} / \text{Pop}$
log(Miles)	Loga cơ số 2 của Miles

¹ *Applied Linear Regression*, 3rd edition, Sanford Weisberg, Wiley-Interscience, 2005.

Ví dụ 2: Dự đoán tiêu thụ nhiên liệu

Một phần bảng thống kê tiêu thụ nhiên liệu

Biến	N	Trung bình	Độ lệch chuẩn	Nhỏ nhất	Trung vị	Lớn nhất
Tax	51	20.15	4.5447	7.5	20.0	29.0
Dlic	51	903.7	72.858	700.2	909.1	1075.3
Income	51	28404	4451.637	20993	27871	40640
logMiles	51	15.75	1.4867	10.58	16.27	18.20
Fuel	51	613.1	88.96	317.5	626.0	842.8

Ta cần ước lượng mô hình

$$\mathbb{E}(\text{Fuel}|X) = \theta_0 + \theta_1 \text{Tax} + \theta_2 \text{Dlic} + \theta_3 \text{Income} + \theta_4 \text{logMiles}.$$

Ví dụ 2: Dự đoán tiêu thụ nhiên liệu

Ta có ma trận thiết kế \mathbf{X} và biến phản hồi \mathbf{y}

$$\mathbf{X} = \begin{pmatrix} 1 & 18.00 & 1031.38 & 23471 & 16.5271 \\ 1 & 8.00 & 1031.641 & 30064 & 13.7343 \\ 1 & 18.00 & 908.597 & 25578 & 15.7536 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 25.65 & 904.8936 & 21915 & 15.1751 \\ 1 & 27.30 & 882.329 & 28232 & 16.7817 \\ 1 & 14.00 & 970.7526 & 27230 & 14.7362 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} 690.264 \\ 514.279 \\ 621.475 \\ \vdots \\ 562.411 \\ 571.794 \\ 842.792 \end{pmatrix}$$

Các cột của ma trận thiết kế tương ứng với hệ số chặn, Tax, Dlic, Income và logMiles. Ma trận \mathbf{X} có cỡ 51×5 , còn \mathbf{y} có cỡ 51×1 .

Ví dụ 2: Dự đoán tiêu thụ nhiên liệu

Áp dụng phương pháp hồi quy thu được các hệ số

Biến	Hệ số	Sai số chuẩn
(Intercept)	154.192845	194.906161
Tax	-4.227983	2.030121
Dlic	0.471871	0.128513
Income	-0.006135	0.002194
logMiles	18.545275	6.472174

QR decomposition & Pseudo-Inverses (Bài đọc thêm)

Các thuật toán phân rã QR

- ▶ Trong phương pháp hồi quy tuyến tính, chúng ta nhắc đến việc sử dụng phân rã QR để tìm tham số $\hat{\theta}$ thay cho việc giải trực tiếp phương trình chuẩn.
- ▶ Có nhiều phương pháp phân rã QR khác nhau. Ví dụ:
 - ▶ Gram - Schmidt
 - ▶ Kỹ thuật quay Givens
 - ▶ Householder
 - ▶ ...
- ▶ Trong phần đọc thêm này ta chỉ tham khảo thuật toán phân rã QR Gram - Schmidt.

Thuật toán Gram - Schmidt

- ▶ Cho ma trận $A \in \mathbb{R}^{m \times n}$, ta tìm các ma trận trực chuẩn $Q \in \mathbb{R}^{m \times n}$ và ma trận tam giác trên (right-triangle) $R \in \mathbb{R}^{n \times n}$ để $A = QR$ bằng cách trực giao hoá các cột của ma trận A . Giả sử $A = (a_1, \dots, a_n)$, với $a_i \in \mathbb{R}^m$ là cột thứ i của A .
- ▶ **Tìm Q :** Ta xây dựng hệ trực giao $\{v_1, \dots, v_n\}$ từ các $\{a_i\}$, sau đó chuẩn hóa chúng.
 - ▶ Dưới đây ký hiệu tích vô hướng của u, v là uv . Chọn $v_1 = a_1$;
 - ▶ Với $i = 2$, bỏ thành phần cùng phương với a_1 khỏi a_2 để tìm v_2 , tức là: $v_2 = a_2 - tv_1$, trong đó $t \in \mathbb{R}$ được chọn để v_1 trực giao với v_2 , tức là

$$v_1 v_2 = v_1(a_2 - tv_1) = v_1 a_2 - tv_1 v_1 = 0 \quad \Rightarrow \quad t = \frac{v_1 a_2}{v_1 v_1} = \frac{v_1 a_2}{\|v_1\|^2}.$$

Thuật toán Gram - Schmidt

- ▶ Tiếp tục quá trình với các cột a_i , $i = 1, 2, \dots$. Tại bước thứ k (ứng với cột thứ k), ta có:

$$v_k = a_k - \sum_{i=1}^{k-1} \frac{v_i a_k}{\|v_i\|^2} v_i.$$

- ▶ Chuẩn hóa các cột $\{v_1, \dots, v_n\}$: $q_k := v_k / \|v_k\|$, ta thu được hệ $\{q_1, q_2, \dots, q_n\}$ là các cột trực chuẩn Q cần tìm.
- ▶ **Tìm R :** Từ quá trình tính v_i và $A = QR$, ta tìm $R = (r_{ij})_{n \times n}$ như sau

$$r_{11} = \|a_1\|;$$

$$r_{ik} = q_i^T a_k \quad \text{với} \quad 1 \leq i < k;$$

$$r_{kk} = \left\| a_k - \sum_{i=1}^{k-1} \frac{v_i a_k}{\|v_i\|^2} v_i \right\|.$$

Thuật toán Gram - Schmidt

Pseudo-code của thuật toán Gram - Schmidt được trình bày đầy đủ như sau:

```
function [Q, R] = GramSchmidtQR(A)
```

```
% Phân tích QR ma trận bằng thuật toán Gram - Schmidt
```

```
[m,n] = size(A);
```

```
R(1,1) = norm(A(:, 1));
```

```
Q(:,1) = A(:, 1)/R(1, 1);
```

```
for k = 2 to n
```

```
    R(1:k - 1, k) = Q(1:m, 1:k-1)'*A(1:m, k);
```

```
    z = A(1:m, k) - Q(1:m, 1:k-1)*R(1:k-1, k);
```

```
    R(k,k) = norm(z);
```

```
    Q(1:m,k) = z/R(k, k);
```

```
endfor;
```

```
return Q, R;
```

```
end
```

Thuật toán Gram - Schmidt

- ▶ Một số thư viện các ngôn ngữ lập trình có thể có sẵn các công cụ tính khai triển QR của một ma trận. Ví dụ: Thông qua đối tượng NumPy trong Python:

```
import numpy as np
```

```
...
```

```
Q, R = np.linalg.qr(X)
```

- ▶ Nếu A hạng thiếu, cần loại bỏ các cột phụ thuộc tuyến tính.
- ▶ Nếu không thể thu gọn dữ liệu, có thể sử dụng nghịch đảo suy rộng (*pseudo-inverse*) $(A^T A)^\dagger$ thay cho $(A^T A)^{-1}$.
- ▶ Ví dụ NumPy.linalg.pinv() trong Python:

```
import numpy as np
```

```
...
```

```
Q, R = np.linalg.pinv(X)
```