HỌC CÓ GIÁM SÁT

Nguyễn Thanh Tùng, **Trần Thị Ngân**
Khoa Công nghệ thông tin – Đại học Thủy lợi
tungnt@tlu.edu.vn, ***ngantt@tlu.edu.vn***

GIỚI THIỆU VỀ HỌC CÓ GIÁM SÁT

Học giám sát

- Xét: $$Y = f(X) + \epsilon$$
- Các phương pháp học giám sát:
  - Học bởi các ví dụ (quan sát)-“Learn by example”
  - Xây dựng mô hình $\hat{f}$ sử dụng tập các quan sát đã được gắn nhãn

$$\left(X^{(1)}, Y^{(1)}\right), \ldots, \left(X^{(n)}, Y^{(n)}\right)$$

Nhận dạng đối tượng

Nhận dạng đối tượng

- Trong nhận dạng đối tượng, chúng ta muốn một bộ phân lớp nhận vào 1 ảnh và cho biết lớp của đối tượng có trong ảnh. (Xác thực I'm not robot trên google)
- Các bộ phân lớp thường học bởi các thuật toán học có giám sát.
  - Dữ liệu huấn luyện trong ImageNet gồm: 1000 lớp với hơn 1 triệu ảnh huấn luyện
  - Sai số của bộ phân lớp từ 2010 đến 2015: 28.2, 25.8, 16.4, 11.7, 6.7, 3.6
  - Bây giờ khả năng phần lớp của ImageNet xấp xỉ khả năng phân lớp của con người

Lọc thư rác

- Trong lọc thư rác, cần 1 bộ phân lớp nhận vào 1 email và xác định email đó là thư rác (spam) hay không (ham hoặc not spam)
- Thường được tạo ra bằng cách học.
- Được sử dụng rộng rãi trên hầu hết các tài khoản email của người dùng.

Tuy nhiên:

- Thay vì xác định 1 email là spam hay không, ta có thể đưa ra 1 số thực thể hiện xác suất email đó là spam.
- Thuật toán được xây dựng bằng cách sử dụng các ngưỡng khác nhau để cực tiểu hóa lỗi dương tính sai (false positive)
- Khi đó ta có bài toán hồi quy, cụ thể là hồi quy logistic

Dự đoán giá nhà, giá cổ phiếu

- Dự đoán giá nhà dựa vào vị trí, số phòng, …
- Dự đoán giá cổ phiếu dựa vào lượng giao dịch trong các ngày trước đó,…

Đây là các bài toán hồi quy.

Học có giám sát

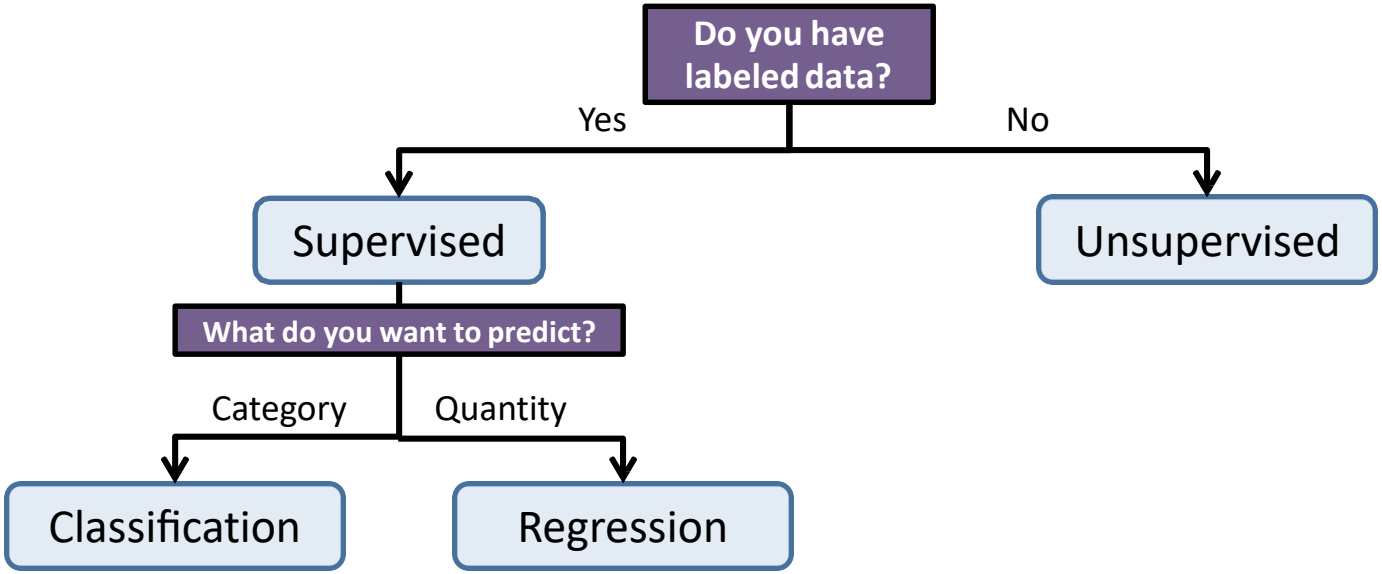
- Giải thuật học có giám sát
  - Lấy hàm ước lượng "tốt nhất" $\hat{f}$ trong tập các hàm
- Ví dụ: Hồi quy tuyến tính
  - Chọn 1 ước lượng tốt nhất từ *dữ liệu học* trong tập các hàm tuyến tính

$$f(X) = \beta_0 + \beta_1 X_1 + \ldots + \beta_d X_d$$

Phân lớp và Hồi quy

- Bài toán học có giám sát gồm 2 dạng:
  - Hồi quy: biến đầu ra Y là định lượng (quantitative)
  - Phân lớp: biến đầu ra Y là định tính/hạng mục/rời rạc

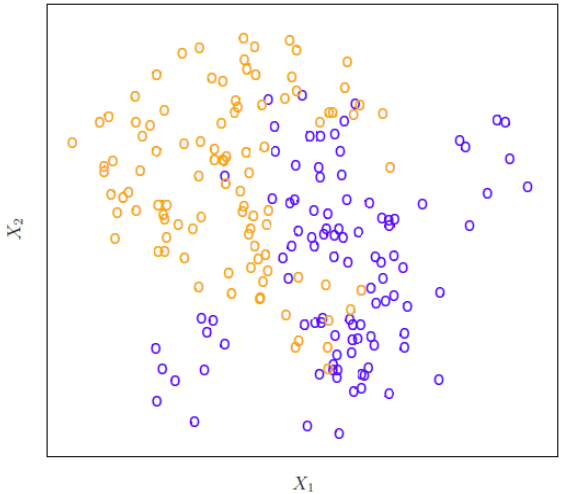
Các dạng giải thuật học máy



GIẢI THUẬT PHÂN LỚP ĐƠN GIẢN

Bộ phân lớp K-láng giềng gần nhất (K-Nearest Neighbor or KNN)

- Ý tưởng: phân lớp các mẫu dựa trên "hàng xóm" các mẫu đã biết nhãn



Bộ phân lớp K-NN

- Bộ phân lớp: Chia không gian thuộc tính thành nhiều vùng
  - Mỗi vùng được gắn với 1 nhãn lớp (class label)
  - *Ranh giới quyết định* chia tách các vùng quyết định
- Các phương pháp phân lớp xây dựng mô hình có dạng:

$$Pr\left(Y \mid X\right)$$

Bộ phân lớp K-NN

- Bộ phân lớp KNN
  - Việc dự đoán lớp cho mẫu X là *lớp phổ biến nhất giữa K láng giềng gần nhất* (trong tập học)
  - Mô hình phân lớp:

$$Pr(X \text{ belongs to class } Y) \approx \frac{\#\,(\text{neighbors of } X \text{ in class } Y)}{K}$$

Bộ phân lớp K-NN

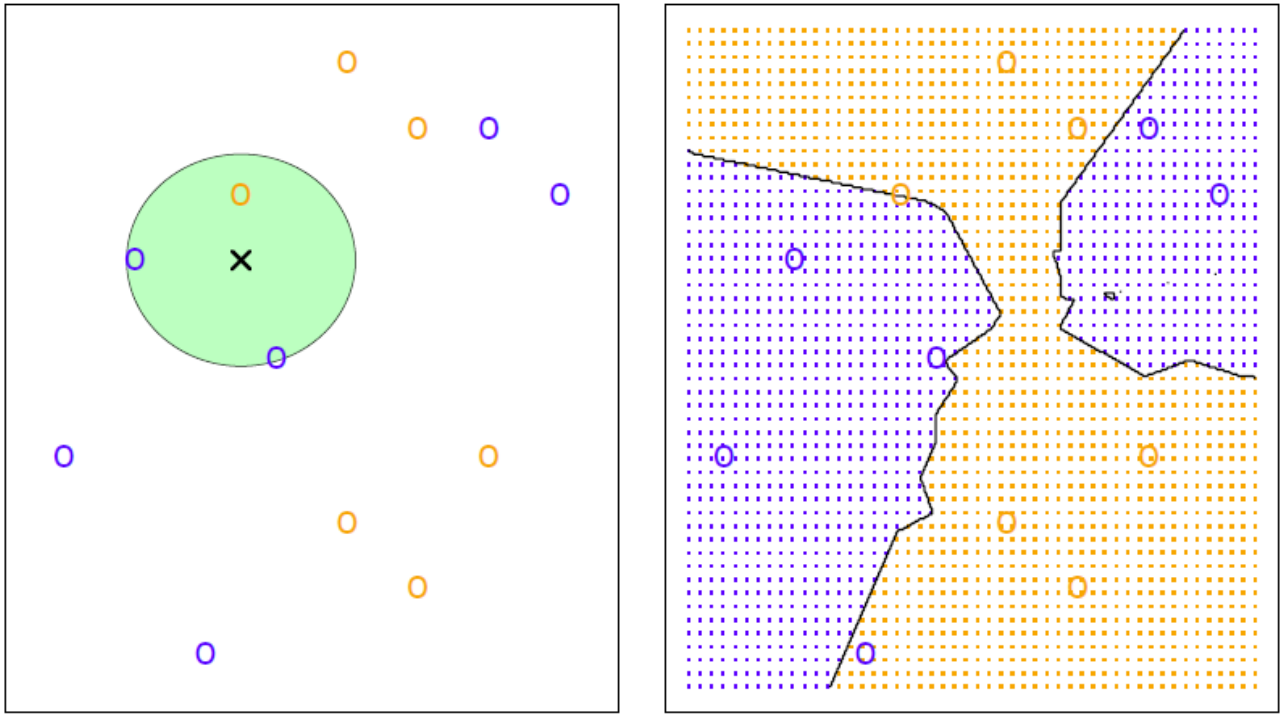


Figure 2.14, ISL 2013

Giải thuật phân lớp k-NN

- Giai đoạn huấn luyện (học): Lưu lại các mẫu trong tập huấn luyện
- Giai đoạn phân lớp: Để phân lớp cho một mẫu (mới) z

Các thành phần của K-NN

- Một tập các bản ghi lưu trữ (dữ liệu đầu vào)
- Một không gian khoảng cách để đo khoảng cách giữa các bản ghi (Euclidean, Minkowski, Manhatan)
- Giá trị của k, số láng giềng gần nhất

Các bước phân loại một mẫu chưa biết nhãn

- Tính khoảng cách đến các bản ghi trong training set
- Xác định k láng giềng gần nhất
- Sử dụng nhãn lớp của láng giềng gần nhất để

xác định nhãn lớp không xác định nhãn cho dữ liệu mới (ví dụ: bằng cách lấy đa số)

Bộ phân lớp K-NN

- Xét 1 láng giềng gần nhất

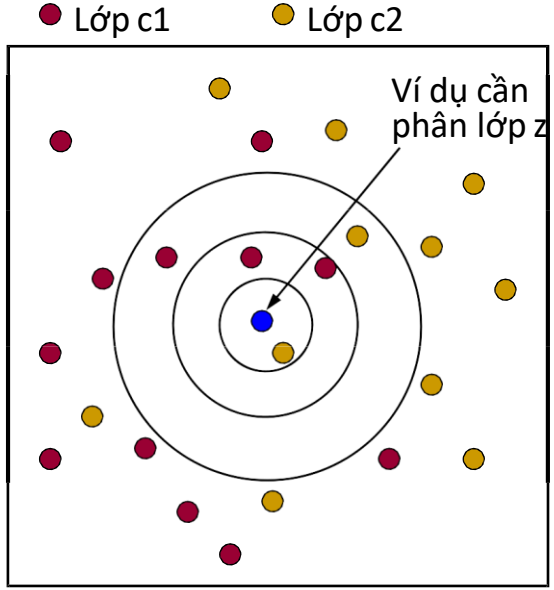
  → Gán z vào lớp c2

- Xét 3 láng giềng gần nhất

  → Gán z vào lớp c1

- Xét 5 láng giềng gần nhất

  → Gán z vào lớp c1



Nguồn hình vẽ: Học máy, Nguyễn Nhật Quang

Ví dụ bài toán phân lớp

Lựa chọn K (bộ phân lớp KNN)

- Nếu k quá nhỏ, KNN nhạy cảm với các điểm nhiễu
- Nếu k quá lớn, vùng lân cận có thể bao gồm các điểm từ các lớp khác
- Giá trị tốt nhất của k phụ thuộc vào dữ liệu
- Có thể sử dụng các kỹ thuật kiểm tra chéo để so sánh k

Lựa chọn K (bộ phân lớp KNN)

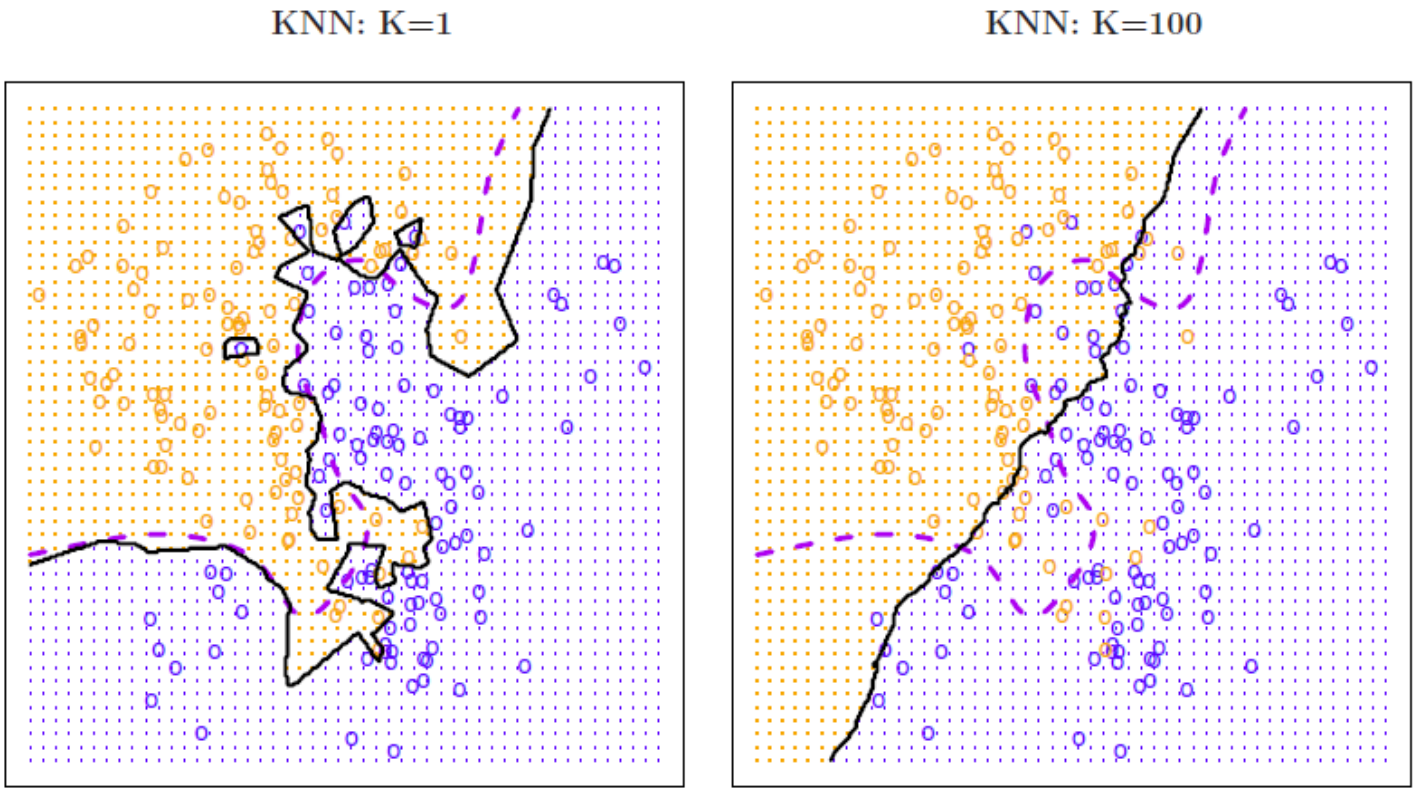


Figure 2.16, ISL 2013

Lựa chọn K (bộ phân lớp KNN)

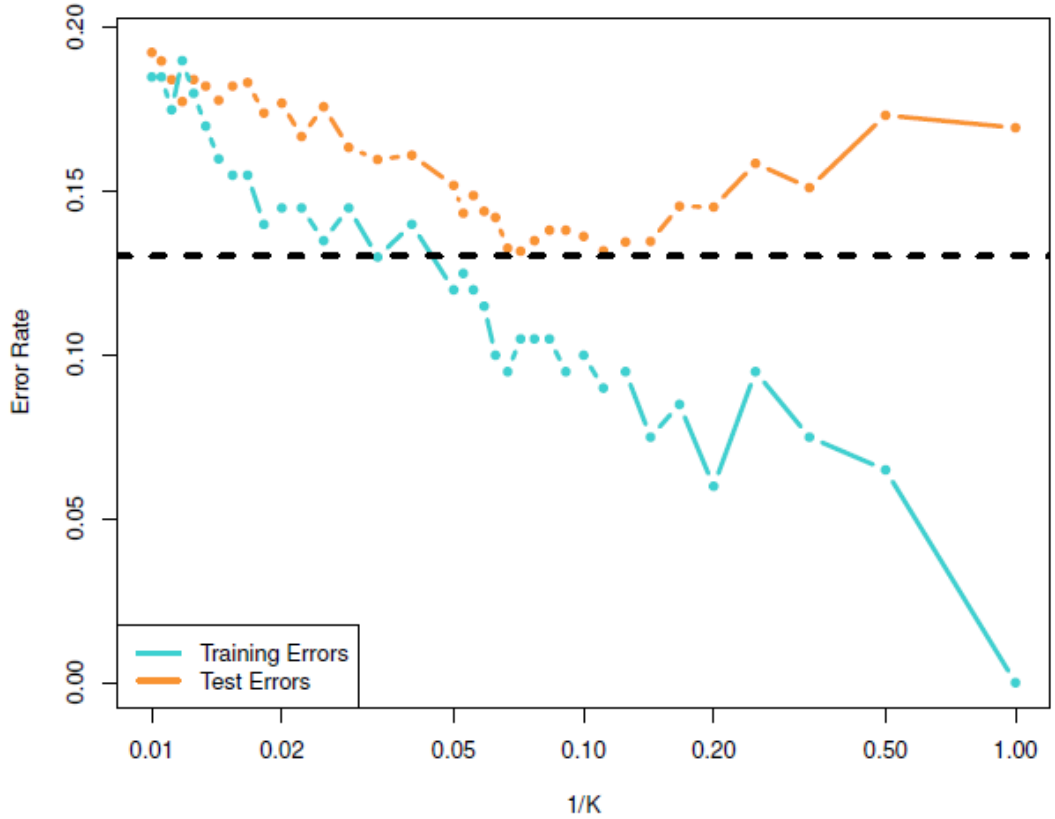


Figure 2.17, ISL 2013

Ưu – nhược điểm của K-NN

- ***Ưu điểm:***

- Dễ cài đặt
- Ít tham số mô hình (K, distance metric)
- Linh hoạt, các lớp không phải tách tuyến tính

Ưu – nhược điểm của K-NN

- ***Nhược điểm:***

- Thời gian tính toán lâu
- Khá nhạy với dữ liệu không cân bằng
- Nhạy với dữ liệu đầu vào không liên quan với nhau

Câu hỏi

- Bộ phân lớp KNN là tham số hay phi tham số?

Ví dụ: Áp dụng KNN trên bộ dữ liệu Iris

Mô tả bộ dữ liệu Iris của Fisher:

- Gồm 150 bản ghi đã được phân loại,
- Gồm 3 loại hoa: setosa, versicolor, virginica
  Có 4 thuộc tính: Độ dài cánh hoa, độ rộng cánh hoa (Sepal.Length Sepal.Width), độ dài đài hoa, độ rộng đài hoa (Petal.Length, Petal.Width)
- Mục đích: sử dụng 4 thuộc tính trên để phân loại các loài hoa

Ví dụ: Áp dụng KNN trên bộ dữ liệu Iris

Các bước thực hiện:

- Phân chia dữ liệu thành 2 phần: training và testing,
- Áp dụng KNN để phân lớp dữ liệu trên tập testing dựa vào nhãn của dữ liệu trong tập training (với giá trị k phù hợp)
- Lập bảng so sánh kết quả phân lớp với nhãn lớp đúng của dữ liệu trong testing

Ví dụ: Áp dụng KNN trên bộ dữ liệu Iris

```
library(gmodels)
library(class)
index = sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
iris.training = iris[index==1, 1:4]
iris.test = iris[index==2, 1:4]
iris.trainLabels = iris[index==1, 5]
iris.testLabels = iris[index==2, 5]
iris.pred = knn(train = iris.training, test= iris.test, cl = iris.trainLabels, k=3)
CrossTable(x = iris.testLabels, y = iris.pred, prop.chisq=FALSE)
```

Kết quả

Total Observations in Table: 45

iris.testLabels	iris.pred setosa	versicolor	virginica	Row Total
setosa	17	0	0	17
	1.000	0.000	0.000	0.378
	1.000	0.000	0.000	
	0.378	0.000	0.000	
versicolor	0	13	3	16
	0.000	0.812	0.188	0.356
	0.000	1.000	0.200	
	0.000	0.289	0.067	
virginica	0	0	12	12
	0.000	0.000	1.000	0.267
	0.000	0.000	0.800	
	0.000	0.000	0.267	
Column Total	17	13	15	45
	0.378	0.289	0.333	

HỒI QUY

Hồi quy tuyến tính

Hồi quy tuyến tính

- *Hồi quy tuyến tính*: là phương pháp học máy có giám sát đơn giản, được sử dụng để dự đoán giá trị biến đầu ra dạng số (định lượng)
  - Nhiều phương pháp học máy là dạng tổng quát hóa của hồi quy tuyến tính
  - Là ví dụ để minh họa các khái niệm quan trọng trong bài toán học máy có giám sát

Hồi quy tuyến tính

- Tại sao dùng hồi quy tuyến tính?

- Mối quan hệ tuyến tính: là sự biến đổi tuân theo quy luật hàm bậc nhất
- Tìm một mô hình (phương trình) để mô tả một mối liên quan giữa X và Y
- Ta có thể biến đổi các biến đầu vào để tạo ra mối quan hệ tuyến tính
- Diễn giải các mối quan hệ giữa biến đầu vào và đầu ra $\rightarrow$ sử dụng cho bài toán suy diễn

Hồi quy tuyến tính đơn giản

- Biến đầu ra Y và biến đầu vào X có mối quan hệ tuyến tính giữa X và Y như sau:

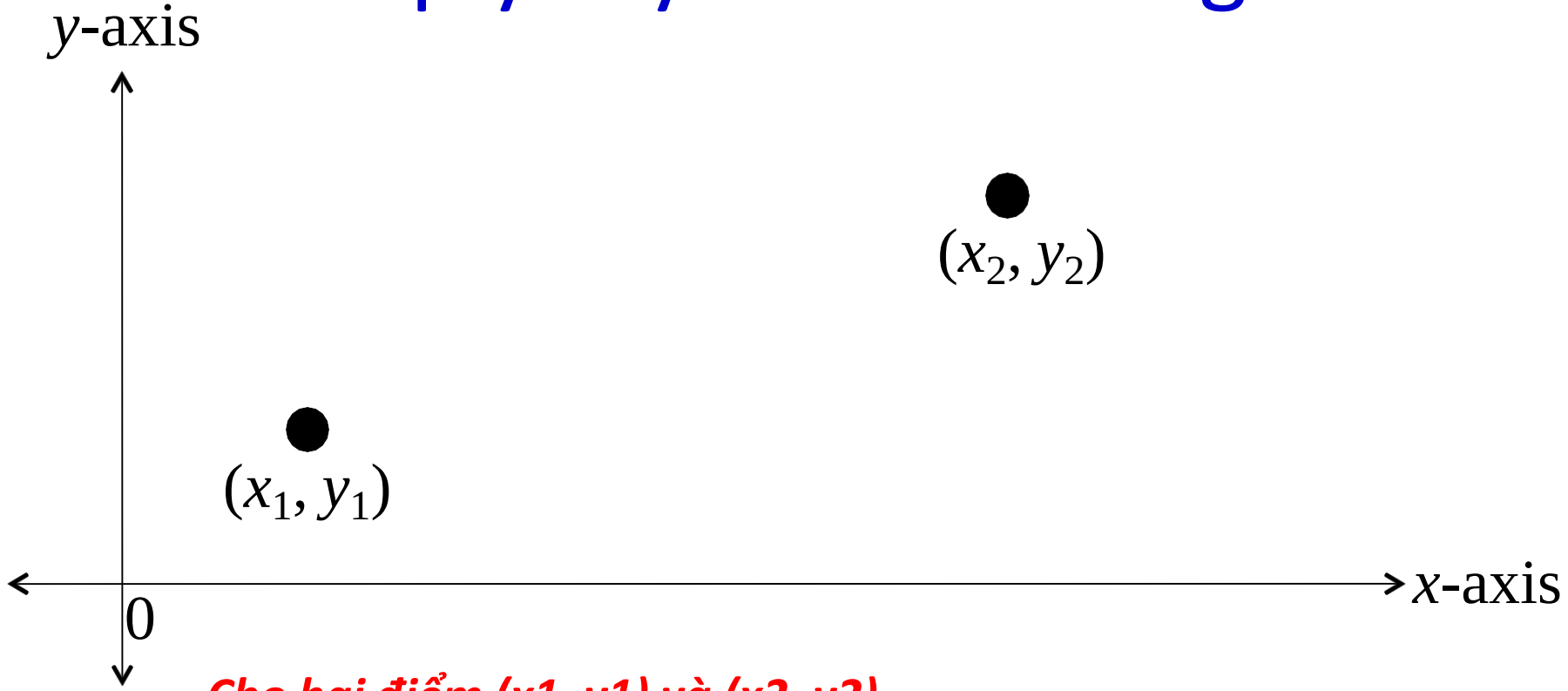
$$Y = \beta_0 + \beta_1 X + \epsilon$$

- Các tham số của mô hình:

β_0 intercept hệ số chặn (khi các x_i=0)

β_1 slope độ dốc

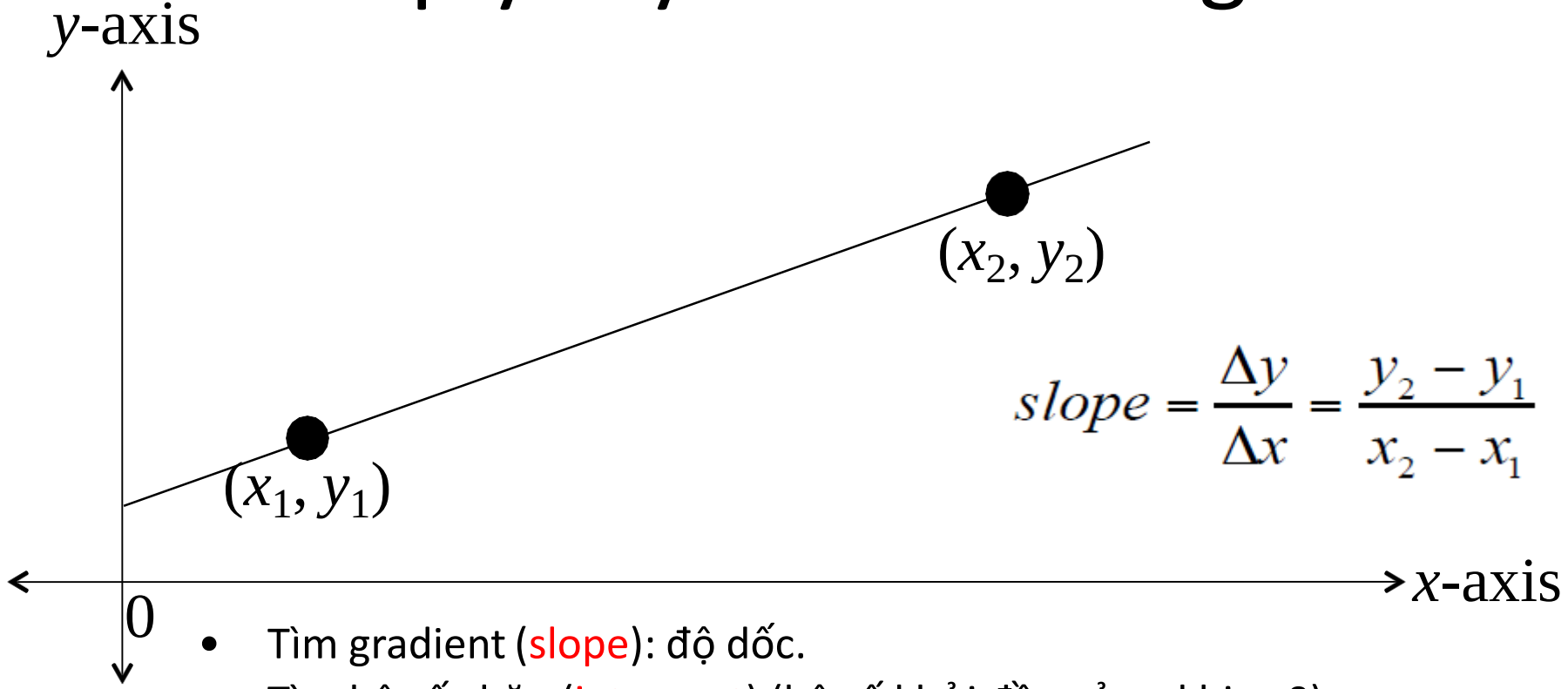
Hồi quy tuyến tính đơn giản



Cho hai điểm (x1, y1) và (x2, y2)

Làm sao để "phát triển" một phương trình nối 2 điểm này?

Hồi quy tuyến tính đơn giản



$$slope = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

- Tìm gradient (slope): độ dốc.
- Tìm hệ số chặn (intercept) (hệ số khởi đầu của y khi x=0)

Hồi quy tuyến tính đơn giản

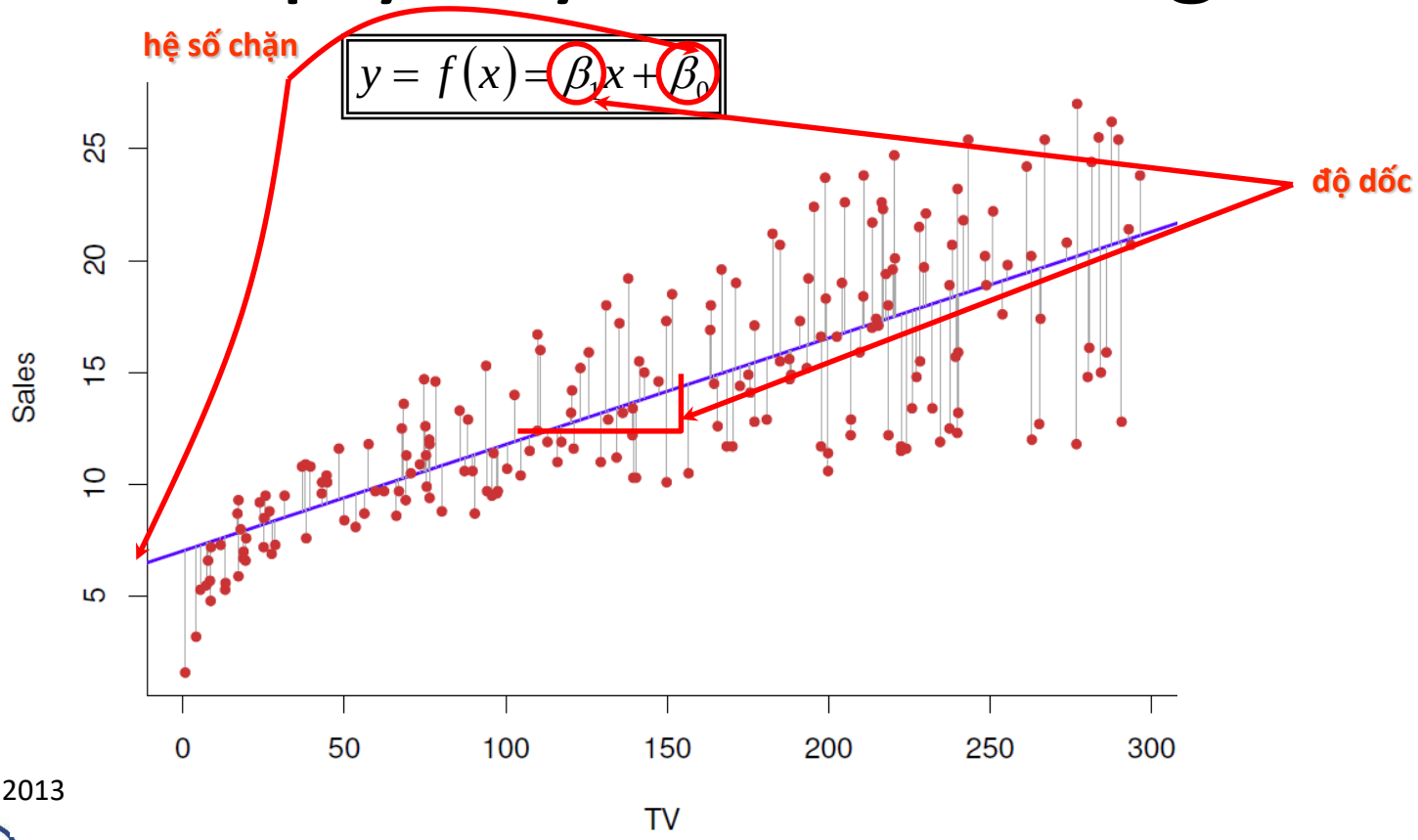


Figure 3.1 , ISL 2013

Hồi quy tuyến tính đơn giản

- β_0 và β_1 chưa biết → Ta ước tính giá trị của chúng từ dữ liệu đầu vào

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

- Lấy $\hat{\beta}_0, \hat{\beta}_1$ sao cho mô hình đạt "xấp xỉ tốt nhất" ("good fit") đối với tập huấn luyện

$$Y^{(i)} \approx \hat{\beta}_0 + \hat{\beta}_1 X^{(i)}, \quad i = 1, \ldots, n$$

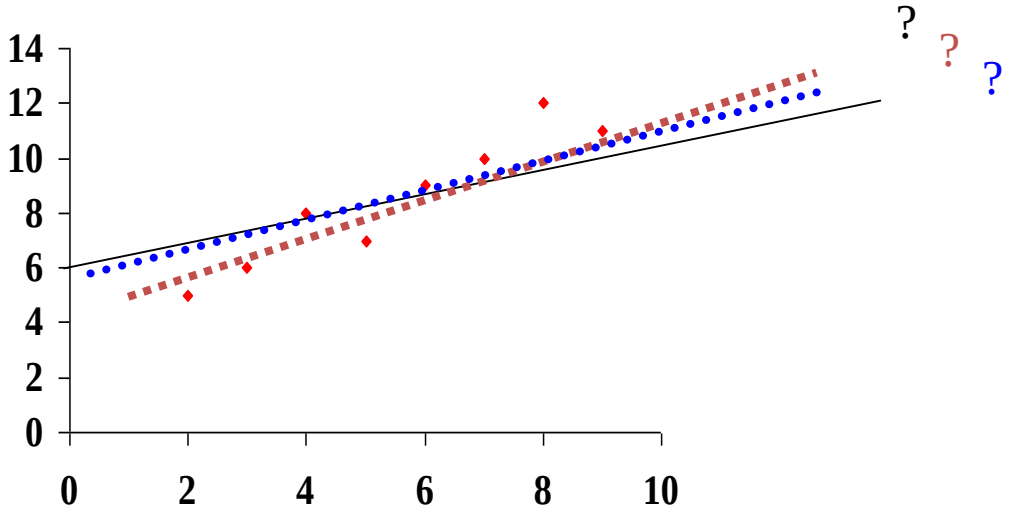
Các giả định

- Mối liên quan giữa X và Y là tuyến tính (linear) về *tham số*
- *X* không có sai số ngẫu nhiên
- Giá trị của *Y* độc lập với nhau (vd, Y_1 không liên quan với Y_2) ;
- Sai số ngẫu nhiên (ε): phân bố chuẩn, trung bình 0, phương sai bất biến $\varepsilon \sim N(0, \sigma^2)$

Hồi quy tuyến tính giả định các biến phải quan hệ tuyến tính → lỗi bias xuất hiện khi hệ thống là phi tuyến.

Đường thẳng phù hợp nhất

Cho tập dữ liệu đầu vào, ta cần tìm cách tính toán các tham số của phương trình đường thẳng



Bình phương nhỏ nhất

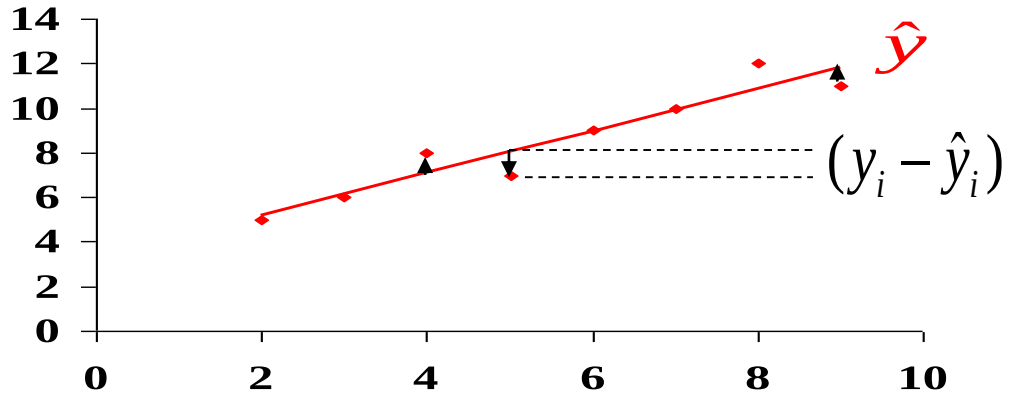
- Thông thường, để đánh giá độ phù hợp của mô hình từ dữ liệu quan sát ta sử dụng phương pháp *bình phương nhỏ nhất* (*least squares)*
- Lỗi bình phương trung bình (Mean Squared Error):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y^{(i)} - \hat{Y}^{(i)}\right)^2$$

Đường thẳng phù hợp nhất

Rất hiếm để có 1 đường thẳng khớp chính xác với dữ liệu, do vậy luôn tồn tại lỗi gắn liền với đường thẳng

Đường thẳng phù hợp nhất là đường giảm thiểu độ dao động của các lỗi này



Phần dư (lỗi)

Biểu thức (y_i - $\hat{y}$) được gọi là lỗi (error) hoặc *phần dư (residual)*

$$\varepsilon_i = (y_i - \hat{y})$$

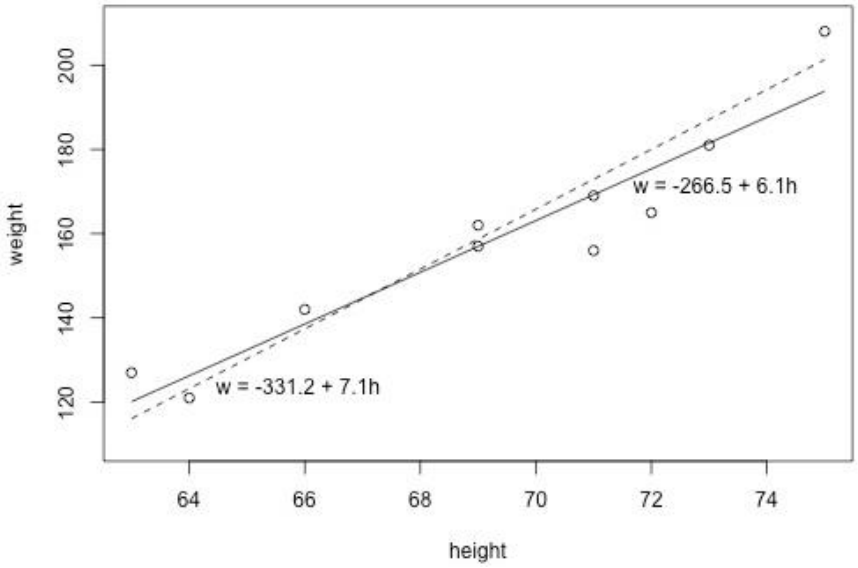
Đường thẳng phù hợp nhất tìm thấy khi tổng bình phương lỗi là nhỏ nhất (SSE = Sum of Squares of Error)

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

VÍ DỤ

Tìm mô hình đạt "xấp xỉ tốt nhất" về mối quan hệ giữa chiều cao (inch), cân nặng (pound) dựa trên dữ liệu đo được từ 10 sinh viên?

i	x_i	y_i
1	63	127
2	64	121
3	66	142
4	69	157
5	69	162
6	71	156
7	71	169
8	72	165
9	73	181
10	75	208

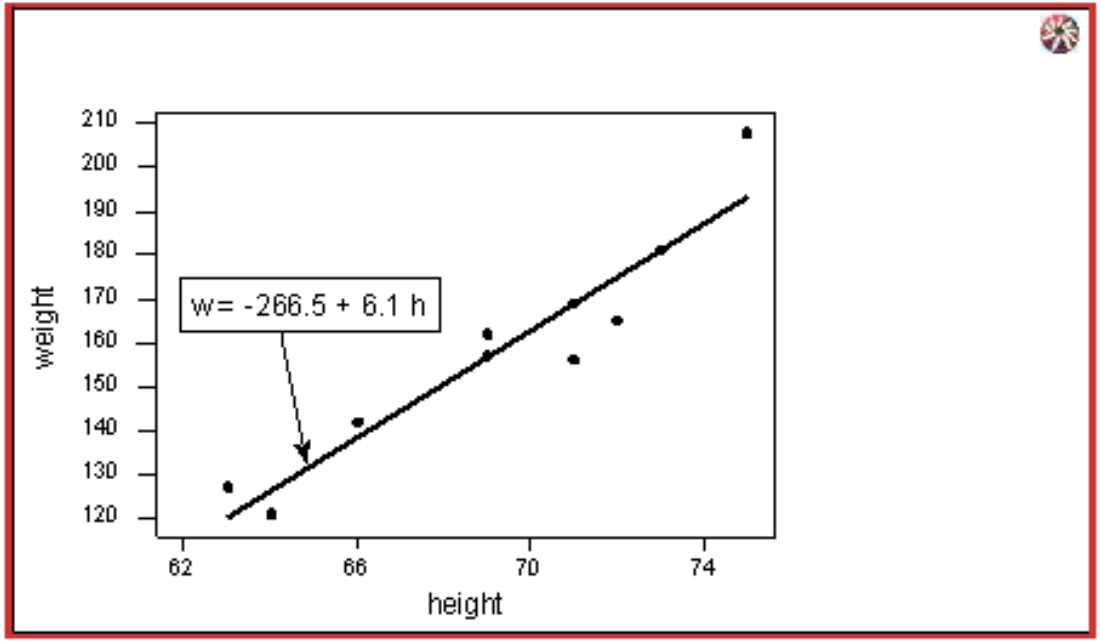


VÍ DỤ

w = -331.2 + 7.1 h (the dashed line)					
i	x_i	y_i	$\hat{y}_i$	$(y_i - \hat{y}_i)$	$(y_i - \hat{y}_i)^2$
1	63	127	116.1	10.9	118.81
2	64	121	123.2	-2.2	4.84
3	66	142	137.4	4.6	21.16
4	69	157	158.7	-1.7	2.89
5	69	162	158.7	3.3	10.89
6	71	156	172.9	-16.9	285.61
7	71	169	172.9	-3.9	15.21
8	72	165	180.0	-15.0	225.00
9	73	181	187.1	-6.1	37.21
10	75	208	201.3	6.7	44.89
					766.5

w = -266.53 + 6.1376 h (the solid line)					
i	x_i	y_i	$\hat{y}_i$	$(y_i - \hat{y}_i)$	$(y_i - \hat{y}_i)^2$
1	63	127	120.139	6.8612	47.076
2	64	121	126.276	-5.2764	27.840
3	66	142	138.552	3.4484	11.891
4	69	157	156.964	0.0356	0.001
5	69	162	156.964	5.0356	25.357
6	71	156	169.240	-13.2396	175.287
7	71	169	169.240	-0.2396	0.057
8	72	165	175.377	-10.3772	107.686
9	73	181	181.515	-0.5148	0.265
10	75	208	193.790	14.2100	201.924
					597.4

VÍ DỤ



Giá trị
$\beta_0 = -266.5$
và
$\beta_1 = 6.1$
nói lên điều gì?

Ước lượng tham số

- Các hệ số $\hat{\beta}_0, \hat{\beta}_1$ được xác định bằng cách cực tiểu hóa MSE

$$\min_{(\hat{\beta}_0, \hat{\beta}_1)} \left[\frac{1}{n} \sum_{i=1}^{n} \left(Y^{(i)} - \left(\hat{\beta}_0 + \hat{\beta}_1 X^{(i)} \right) \right)^2 \right]$$

- Độ dốc của đường thẳng là: $\hat{\beta}_1 = \frac{SS_{xy}}{SS_x}$

trong đó: $SS_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$ và $SS_x = \sum_{i=1}^{n} (x_i - \bar{x})^2$

Ước lượng tham số

Hệ số chặn của đường thẳng

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

trong đó

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \qquad \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Hồi quy tuyến tính đơn giản

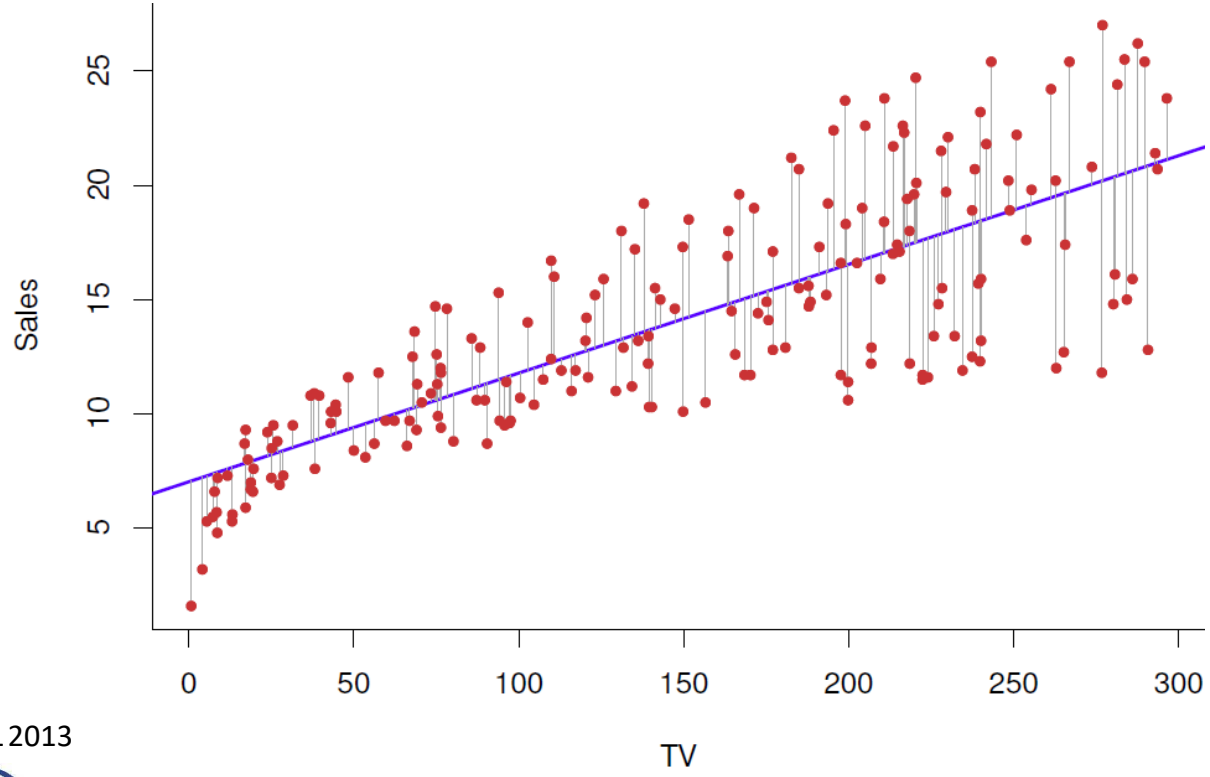
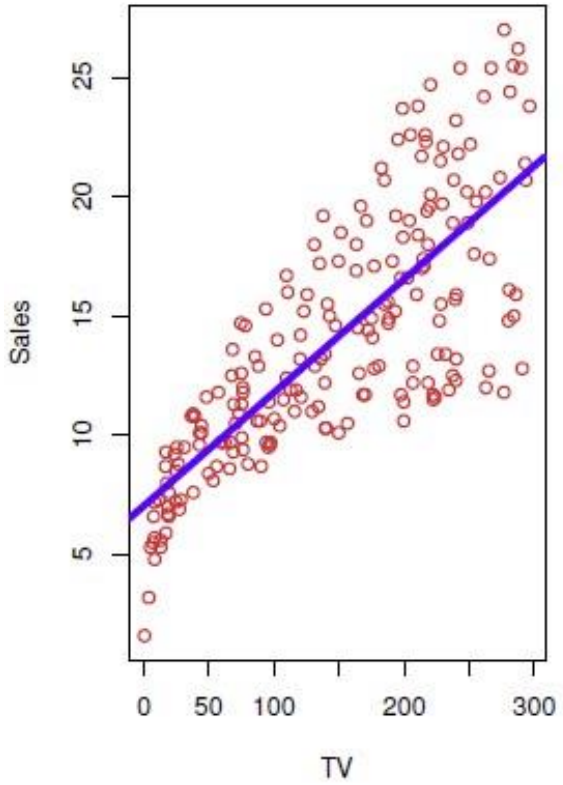
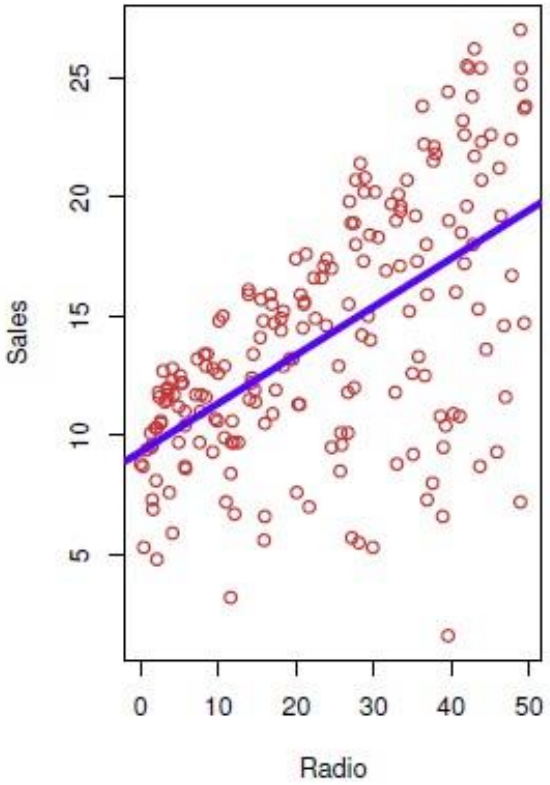
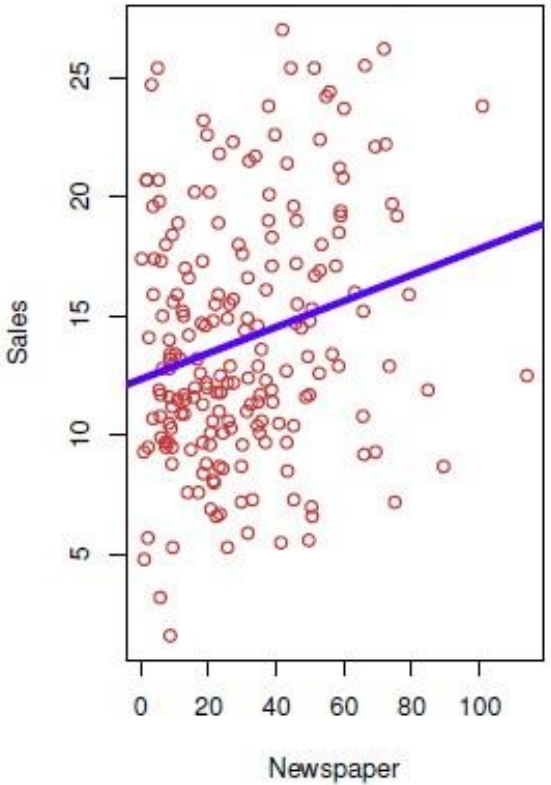


Figure 3.1 , ISL 2013

Hồi quy tuyến tính đơn giản







Phương pháp đánh giá

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2}; \textit{(the square Root of the variance of the residuals)}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i| \textit{ (The Mean Absolute Error)}$$

và $R^2 = 1 - \sum_{i=1}^{N}(Y_i - \hat{Y}_i)/\sum_{i=1}^{N}(Y_i - \bar{Y}_i)$.

VÍ DỤ

X kilos	Y giá $
17	132
21	150
35	160
39	162
50	149
65	170

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X \qquad \hat{\beta}_1 = \frac{SS_{xy}}{SS_x}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$SS_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \qquad SS_x = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n} \qquad \bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

VÍ DỤ

X kilos	Y giá $
17	132
21	150
35	160
39	162
50	149
65	170

$$\bar{x} = 37.83$$

$$\bar{y} = 153.83$$

$$SS_{xy} = 891.83$$

$$SS_x = 1612.83$$

$$\hat{\beta}_1 = \frac{SS_{xy}}{SS_x} = \frac{891.83}{1612.83} = 0.553$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 153.83 - 0.553 \times 37.83 = 132.91$$

Phương trình tìm được là: $Y = 132.91 + 0.553*X$

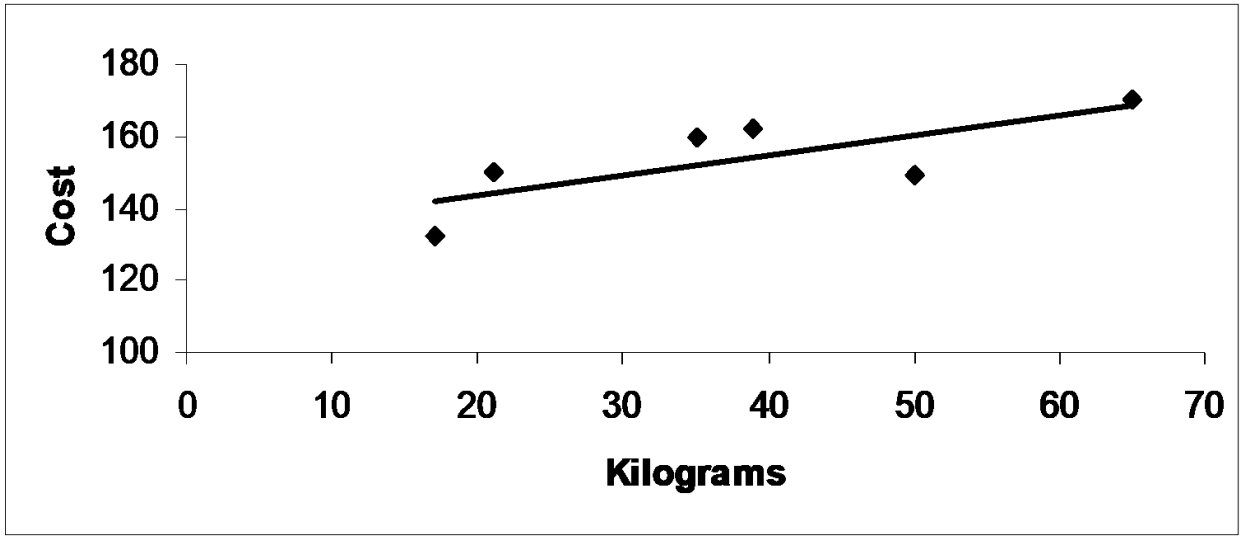
VÍ DỤ

X kilos	Y giá $
17	132
21	150
35	160
39	162
50	149
65	170

```
Linear_Model = lm(y ~ x)
summary (Linear_Model)
plot(chol~age)
abline(Linear_Model,col="blue")
```

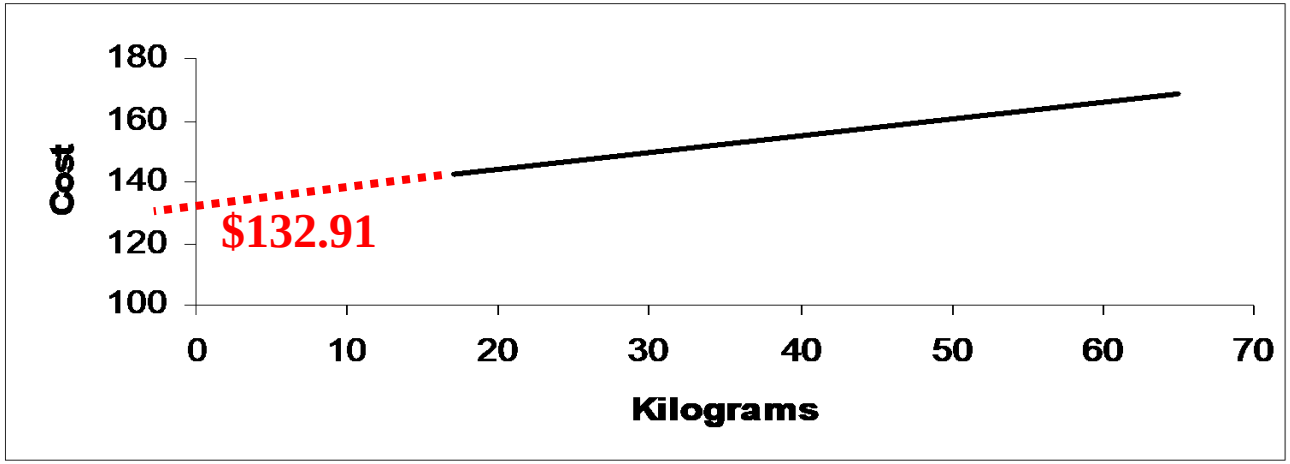
Diễn giải tham số

Trong ví dụ trước, tham số ước lượng $\hat{\beta}_1$ của độ dốc là 0.553. Điều này có nghĩa là khi thay đổi 1 kg của X, giá của Y thay đổi 0.553 $



Diễn giải tham số

$\hat{\beta}_0$ là hệ số chặn của Y. Nghĩa là, điểm mà đường thẳng cắt trục tung Y. Trong ví dụ này là $132.91



Đây là giá trị của Y khi X = 0

Dữ liệu phân tích: Boston

- **Boston data:** liên quan đến giá nhà đất
- Các biến số
  - **crim**: tỉ lệ tội phạm của thị trấn
  - **zn**: tỉ lệ khu đất có diện tich trên 25,000 feet vuông
  - **indus**: tỉ lệ doanh nghiệp tương đối lớn
  - **chas**: gần sông Charles (1=yes, 0=no)
  - **nos**: nồng độ nitric oxides (parts/10 triệu)
  - **rm**: số phòng trung bình mỗi nhà
  - **age**: tỉ lệ căn hộ (unit) xây trước 1940
  - **dis**: khoảng cách đến các trung tâm kĩ nghệ (tìm việc làm)

Dữ liệu phân tích: Boston

- **Boston data:** liên quan đến giá nhà đất
- Các biến số
  - **rad**: chỉ số gần xa lộ radial
  - **tax**: tỉ suất thuế tinh trên $10,000
  - **ptratio:** tỉ số học trò trên giáo viên của thị trấn
  - **black**: chỉ số về số người da đen trong thị trấn (Bk – 0.63)^2
  - **lstat**: tỉ lệ dân số thành phần kinh tế thấp
  - **medv:** trị giá nhà ($1000)

Ước tính bằng R

- Chúng ta muốn ước tính mối liên quan giữa số phòng (rm) và giá căn nhà (medv)
- Mô hình hồi qui tuyến tính:

$$medv = \alpha + \beta * rm + \varepsilon$$

- Lệnh trên R:

```
lm(medv ~ rm, data=Boston)
```

Phân tích bằng R

```
library (MASS)
attach(Boston)
# Phân tích hồi qui tuyến tính
m1 = lm(medv ~ rm, data= Boston)
summary(m1)

# vẽ biểu đồ
plot(medv ~ rm, pch=16)
abline(m1, col="red")
```

KẾT QUẢ

```
Residuals:
    Min      1Q  Median      3Q     Max
-23.346  -2.547   0.090   2.986  39.433

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -34.671      2.650  -13.08   <2e-16 ***
rm            9.102      0.419   21.72   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
' 1

Residual standard error: 6.616 on 504 degrees of freedom
Multiple R-squared:  0.4835,    Adjusted R-squared:  0.4825
F-statistic: 471.8 on 1 and 504 DF,  p-value: < 2.2e-16
```

Diễn giải kết quả

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -34.671      2.650  -13.08   <2e-16 ***
rm             9.102      0.419   21.72   <2e-16 ***
```

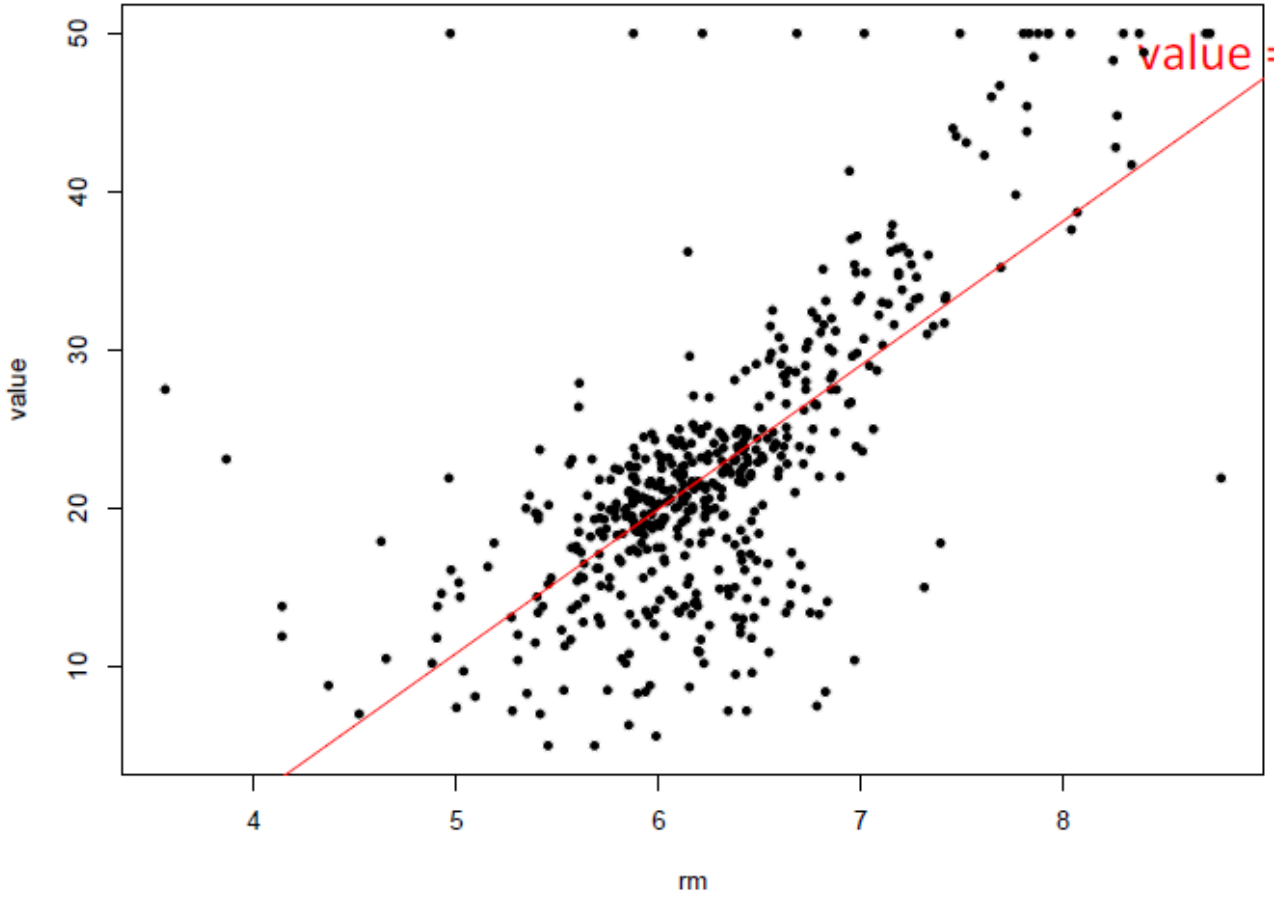
- Nhớ rằng mô hình là:

$$medv = a + b*rm$$

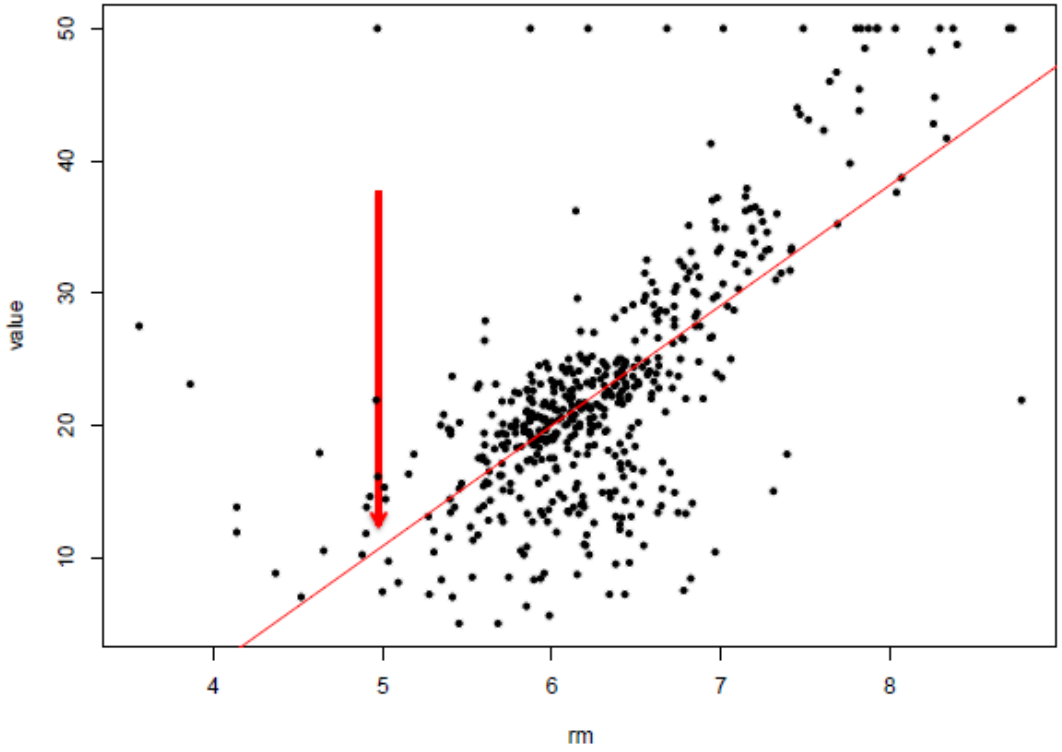
- Phương trình:

$$medv = -34.67 + 9.10*rooms$$

- Ý nghĩa: Nếu căn nhà có thêm 1 phòng thì giá trị căn nhà sẽ tăng 9100 USD. Mối tương quan này có **ý nghĩa thống kê (P < 0.0001)**


value = -34.67 + 9.10*rooms
value
rm
10
20
30
40
50
4
5
6
7
8

Ý nghĩa của đường biểu diễn



Giá trị trung bình (kì vọng)

medv= -34.67 + 9.10*rooms

Khi room = 5,

medv= -34.67 + 9.10*5 = **10.83**

Khi room **= 6**

medv= -34.67 + 9.10*6 = **19.93**

Khi room **= 8**

medv= -34.67 + 9.10*8 = **38.13**

Hồi quy tuyến tính đa biến

- Hồi quy tuyến tính đa biến: mô hình có nhiều hơn một biến dùng để dự đoán biến đích

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_d X_d + \epsilon$$

Hồi quy tuyến tính đa biến

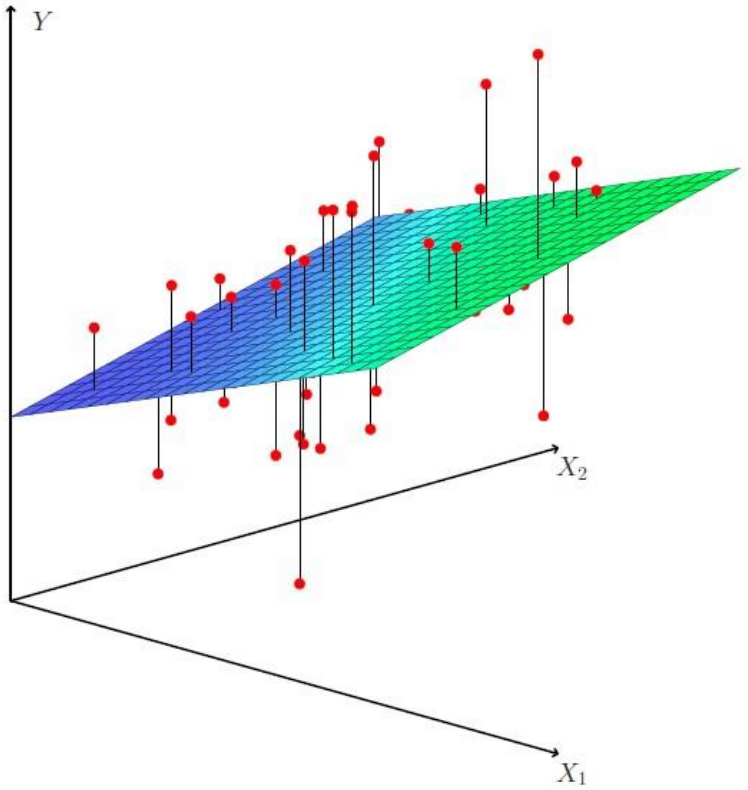


Figure 3.4 , ISL 2013

Hồi quy tuyến tính đa biến

- Diễn giải hệ số β_j: khi tăng Xj lên một đơn vị ➔ Y sẽ tăng trung bình một lượng là βj

	Coefficient
Intercept	2.939
TV	0.046
radio	0.189
newspaper	−0.001

Bình phương nhỏ nhất

- Tìm các ước số bằng phương pháp bình phương nhỏ nhất

$$\hat{\beta} = \arg\min_{\beta} \|Y - X^T\beta\|^2$$

$$X = \begin{bmatrix} 1 & X^{(1)^T} \\ & \cdots \\ 1 & X^{(n)^T} \end{bmatrix} \qquad Y = \begin{bmatrix} Y^{(1)} \\ \cdots \\ Y^{(n)} \end{bmatrix} \qquad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \cdots \\ \hat{\beta}_d \end{bmatrix}$$

Bình phương nhỏ nhất

- Giải phương trình sau để tìm: $\hat{\beta}$

$$X^T X\hat{\beta} = X^T Y \quad \rightarrow \quad \hat{\beta} = (X^T X)^{-1} X^T Y$$

Dữ liệu phân tích: Boston

- **Boston data:** liên quan đến giá nhà đất
- Các biến số
  - **crim**: tỉ lệ tội phạm của thị trấn
  - **zn**: tỉ lệ khu đất có diện tich trên 25,000 feet vuông
  - **indus**: tỉ lệ doanh nghiệp tương đối lớn
  - **chas**: gần sông Charles (1=yes, 0=no)
  - **nos**: nồng độ nitric oxides (parts/10 triệu)
  - **rm**: số phòng trung bình mỗi nhà
  - **age**: tỉ lệ căn hộ (unit) xây trước 1940
  - **dis**: khoảng cách đến các trung tâm kĩ nghệ (tìm việc làm)

Dữ liệu định tính

- Xử lý dữ liệu dạng định tính (định danh, hạng mục) trong mô hình hồi quy tuyến tính

  VD: biến "giới tính" gồm "male" hoặc "female"

- Nếu chỉ có 2 khả năng trên, ta tạo *biến giả* (*dummy variable)*

$$X_j = \begin{cases} 1 & \text{if female} \\ 0 & \text{if male} \end{cases}$$

Dữ liệu định tính

- Nếu có nhiều hơn 2 giá trị, ta biểu diễn biến chúng dùng nhiều biến giả

  VD: biến "màu mắt" gồm "blue", "green" or "brown"

$$X_j = \begin{cases} 1 & \text{if blue} \\ 0 & \text{if not blue} \end{cases}$$

$$X_{j+1} = \begin{cases} 1 & \text{if brown} \\ 0 & \text{if not brown} \end{cases}$$

Hồi quy tuyến tính - Ưu điểm

- Mô hình đơn giản, dễ hiểu
- Dễ diễn giải hệ số hồi quy
- Nhận được kết quả tốt khi dữ liệu quan sát nhỏ
- Nhiều cải tiến/mở rộng

Hồi quy tuyến tính- Nhược điểm

- Mô hình hơi đơn giản nên khó dự đoán chính xác với dữ liệu có miền giá trị rộng
- Khả năng ngoại suy (extrapolation) kém
- Nhạy cảm với dữ liệu ngoại lai (outliers) – do dùng phương pháp bình phương nhỏ nhất

VÍ DỤ

Cho ma trận X và vector y như sau:

$$\mathbf{y} = \begin{bmatrix} 6 \\ 9 \\ 12 \\ 5 \\ 13 \\ 2 \end{bmatrix} \qquad \boldsymbol{X} = \begin{bmatrix} 1 & 3 & 9 & 16 \\ 1 & 6 & 13 & 13 \\ 1 & 4 & 3 & 17 \\ 1 & 8 & 2 & 10 \\ 1 & 3 & 4 & 9 \\ 1 & 2 & 4 & 7 \end{bmatrix} \qquad \hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{bmatrix} = ?$$

$$X^T X \hat{\beta} = X^T Y \quad \rightarrow \quad \hat{\beta} = (X^T X)^{-1} X^T Y$$

VÍ DỤ

$$X^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 6 & 4 & 8 & 3 & 2 \\ 9 & 13 & 3 & 2 & 4 & 4 \\ 16 & 13 & 17 & 10 & 9 & 7 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 6 & 26 & 35 & 72 \\ 26 & 138 & 153 & 315 \\ 35 & 153 & 295 & 448 \\ 72 & 315 & 448 & 944 \end{bmatrix}$$

$$X^T y = \begin{bmatrix} 47 \\ 203 \\ 277 \\ 598 \end{bmatrix}$$

Ví dụ

$$\hat{\beta} = (X^T X)^{-1} X^T y = \begin{bmatrix} 2.59578 & -0.15375 & -0.01962 & -0.13737 \\ -0.15375 & 0.03965 & -0.00014 & -0.00144 \\ -0.01962 & -0.00014 & 0.01234 & -0.00431 \\ -0.13737 & -0.00144 & -0.00431 & 0.01406 \end{bmatrix} \begin{bmatrix} 47 \\ 203 \\ 277 \\ 598 \end{bmatrix}$$

$$= \begin{bmatrix} 3.20975 \\ -0.07573 \\ -0.11162 \\ 0.46691 \end{bmatrix}$$

$$\hat{\beta}_0 = 3.20975 \quad \hat{\beta}_1 = -0.07573 \quad \hat{\beta}_2 = -0.11162 \quad \hat{\beta}_3 = 0.46691$$

$$\hat{y} = 3.20975 - 0.07573x_1 - 0.11162x_2 + 0.46691x_3$$

ĐỘ CHÍNH XÁC CỦA MÔ HÌNH

Đo hiệu năng bài toán hồi quy

- Hàm tổn thất (Loss function): loại hàm dùng để đo lường sai số của mô hình
- Vd: Sai số bình phương trung bình (Mean squared error - MSE)
  - Độ đo thông dụng dùng để tính độ chính xác bài toán hồi quy

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

  - Tập trung đo các sai số lớn hơn là các sai số nhỏ

Đo hiệu năng bài toán hồi quy

- Mục tiêu: xây dựng mô hình khái quát hóa (*generalizes)*

  Ta muốn cực tiểu hóa lỗi trên dữ liệu chưa biết, không phải trên dữ liệu học.

VD: Dự đoán giá cổ phiếu *trong tương lai* dựa trên giá cổ phiếu trong quá khứ

- Chúng ta muốn cực tiểu tổn thất kỳ vọng (*expected* loss):

Vấn đề: Không thể cực tiểu lỗi trên dữ liệu huấn luyện.

Overfitting

- *Quá khớp (Overfitting)*: Học sự biến thiên ngẫu nhiên trong dữ liệu hơn là xu hướng cơ bản
- Đặc điểm của overfitting: Mô hình có hiệu năng cao trên tập dữ liệu học nhưng kém trên tập dữ liệu thử nghiệm.

Underfitting và Overfitting

- Có 50 điểm dữ liệu được tạo bằng một đa thức bậc ba cộng thêm nhiễu.
- Đồ thị của đa thức (true model) có màu xanh lá cây
- Bài toán: Giả sử ta không biết mô hình ban đầu mà chỉ biết các điểm dữ liệu, hãy tìm một mô hình "tốt" để mô tả dữ liệu đã cho?

Underfitting và Overfitting

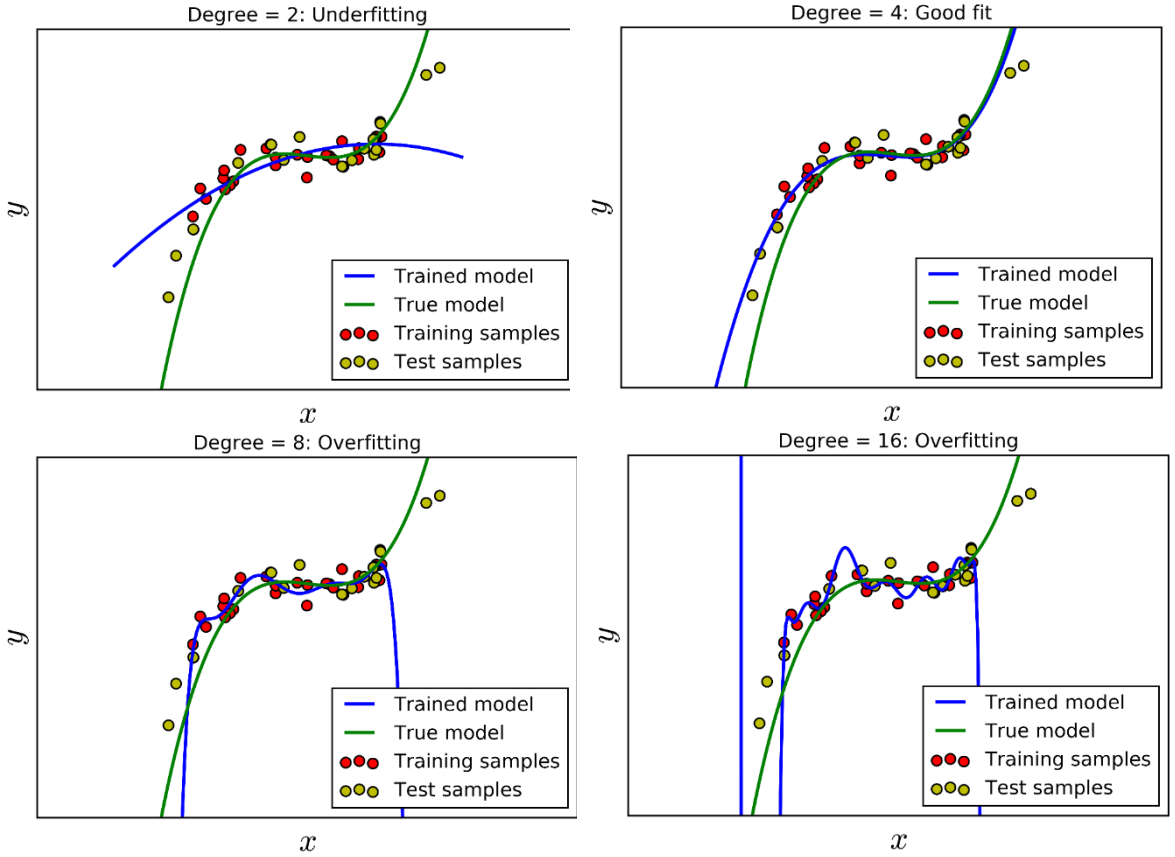
- Với d=2, mô hình không thực sự tốt vì dự đoán quá khác so với mô hình thực

→ *underfitting*

- Với d=8 và d=16, với các điểm dữ liệu trong khoảng của training data, mô hình dự đoán và mô hình thực là khá giống nhau. Tuy nhiên, về phía phải, đa thức bậc 8 và 16 cho kết quả hoàn toàn ngược với xu hướng của dữ liệu

→ *Overfitting.*

- d=4, mô hình tốt nhất.



Đánh giá hiệu năng

- Lỗi huấn luyện và lỗi kiểm thử thể hiện khác nhau.

Khi tính linh hoạt của mô hình tăng lên:

- *Lỗi huấn luyện* giảm
- *Lỗi kiểm thử ban đầu* giảm, nhưng sau đó tăng lên do có hiện tượng overfitting → lỗi kiểm thử dạng chữ U - “U-shaped”

Đánh giá hiệu năng

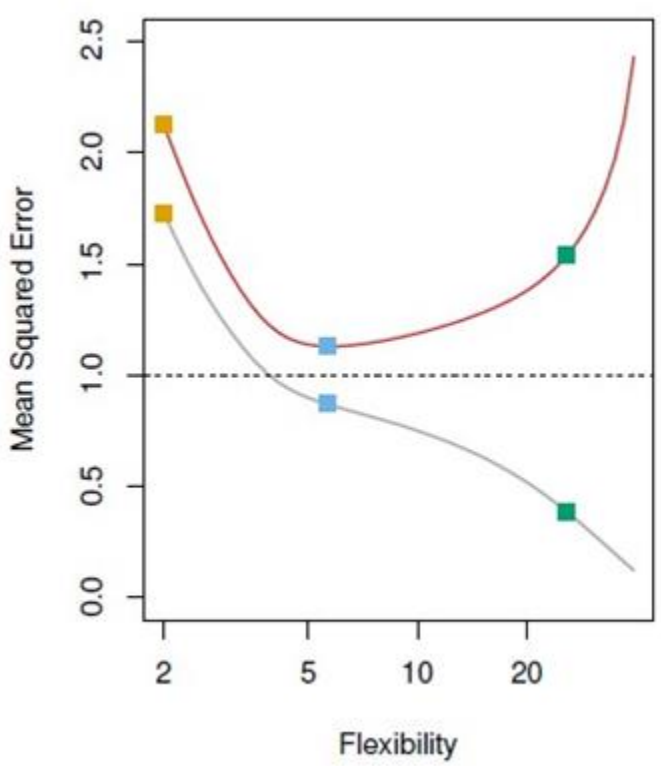


Figure 2.9 , ISL 2013

Đánh giá hiệu năng

- Làm sao để ước lượng lỗi kiểm thử để tìm một mô hình tốt?
- *Kỹ thuật kiểm tra chéo (Cross-validation hay CV)*

Một tập các kỹ thuật nhằm sử dụng dữ liệu huấn luyện để ước lượng lỗi tổng quát (generalization error)

Dữ liệu trong CV

- *Dữ liệu huấn luyện (Training data):* Tập các quan sát (bản ghi) được sử dụng để xây dựng (học) mô hình.
- *Dữ liệu kiểm chứng (Validation data):* Tập các quan sát dùng để ước lượng lỗi nhằm tìm tham số hoặc lựa chọn mô hình.
- *Dữ liệu kiểm thử (Test data):*
  - Tập các quan sát dùng để đánh giá hiệu năng trên dữ liệu chưa biết (unseen) trong tương lai.
  - Dữ liệu này không sử dụng cho giải thuật học máy trong quá trình xây dựng mô hình.

Trade-off: Độ lệch vs. Phương sai

- Lỗi kiểm thử đường cong hình chữ U (U-shaped) xảy ra dựa trên 2 đặc điểm của mô hình học máy:

$$\mathbb{E}\left[\text{test error}\right] = var(\hat{f}) + bias(\hat{f})^2 + var(\epsilon)$$

$var(\hat{f})$: *Phương sai (variance)* của hàm ước lượng

$bias(\hat{f})$: *Độ chệch/sai lệch (bias)* của hàm ước lượng

Trade-off: Độ lệch vs. Phương sai

- **Phương pháp đơn giản:** bias cao, phương sai thấp
- **Phương pháp linh hoạt (phức tạp):** bias thấp, phương sai cao

Trade-off: Độ lệch vs. Phương sai

- Dễ đạt được phương sai thấp/bias cao *hoặc* phương sai cao/bias thấp,
- Tuy nhiên rất khó để đạt được cả phương sai *và* bias cùng thấp

Kỹ thuật đánh giá chéo

"Dùng lỗi trên tập dữ liệu kiểm thử để ước lượng lỗi dự đoán"

$$err = E[L(Y, \hat{f}(X))]$$

Khi xảy ra hiện tượng Underfitting và Overfitting???

Tập đánh giá (Validation)

- Phương pháp: trích từ training data ra một tập con nhỏ và thực hiện việc đánh giá mô hình trên tập con này.
- Tập con nhỏ được trích ra từ training set này được gọi là validation set.
- Training set là phần còn lại của training set ban đầu (sau khi bỏ đi tập đánh giá).

Tập đánh giá (Validation)

- Train error: được tính trên training set mới này.
- Validation error: Lỗi được tính trên tập validation.
- Tìm mô hình sao cho cả *train eror* và *validation error* đều nhỏ, qua đó có thể dự đoán được rằng *test error* cũng nhỏ.

Tập đánh giá (Validation)

- Phương pháp thường được sử dụng là sử dụng nhiều mô hình khác nhau. Mô hình nào cho validation error nhỏ nhất sẽ là mô hình tốt.
- Tuy nhiên, khi ta có số lượng dữ liệu để xây dựng mô hình rất hạn chế, nếu lấy quá nhiều dữ liệu trong tập training ra làm dữ liệu validation, phần dữ liệu còn lại của tập training là không đủ để xây dựng mô hình.

Tập đánh giá (Validation)

- Nếu ta giữ tập validation phải thật nhỏ để có được lượng dữ liệu cho training đủ lớn. Một vấn đề khác nảy sinh, hiện tượng overfitting lại có thể xảy ra với tập training còn lại.
- Giải pháp: Cross-validation (Kỹ thuật đánh giá chéo).

Kỹ thuật đánh giá chéo

- *Cross validation* là một cải tiến của *validation* với lượng dữ liệu trong tập validation là nhỏ nhưng chất lượng mô hình được đánh giá trên nhiều tập *validation* khác nhau.
- Chia tập training ra ***k*** tập con không có phần tử chung, có kích thước gần bằng nhau.
- Tại mỗi lần kiểm thử, một trong số ***k*** tập con được lấy ra làm *validation set*. Mô hình sẽ được xây dựng dựa vào hợp của ***k−1*** tập con còn lại.

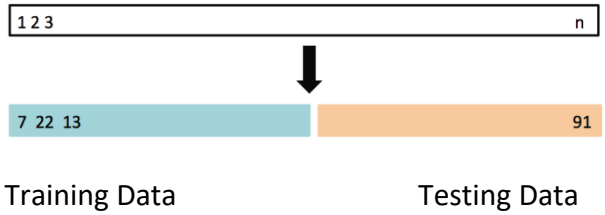
Kỹ thuật đánh giá chéo

- Mô hình cuối được xác định dựa trên trung bình của các train error và validation error. Cách làm này còn có tên gọi là k-fold cross validation.

Tập huấn luyện - Training Set

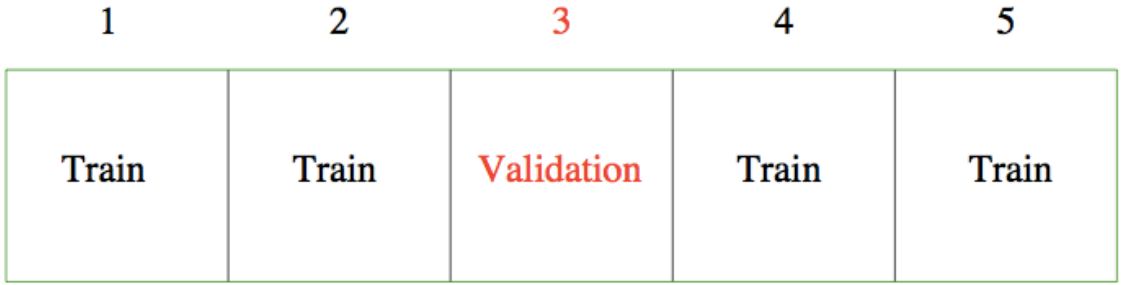
Tập kiểm thử - Test Set

Tập đánh giá - Validation Set



Kỹ thuật đánh giá chéo K–fold

Ví dụ 5–fold

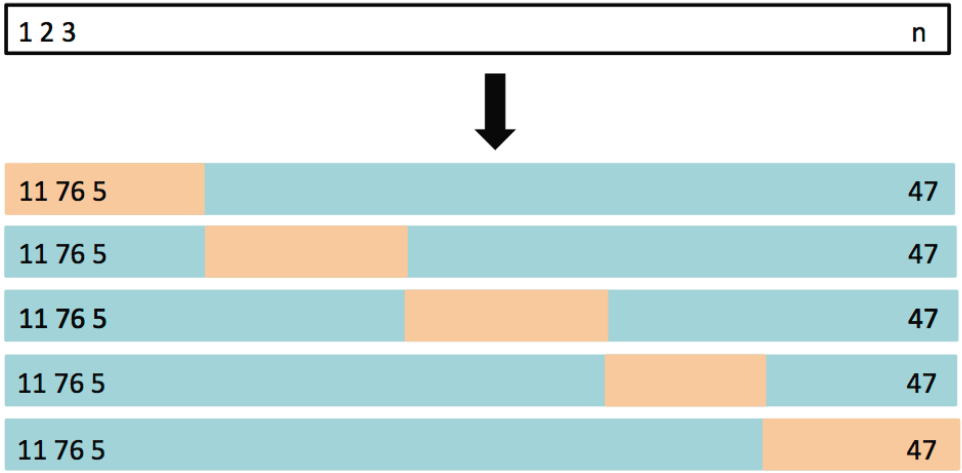


$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i))$$

Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

Kỹ thuật đánh giá chéo

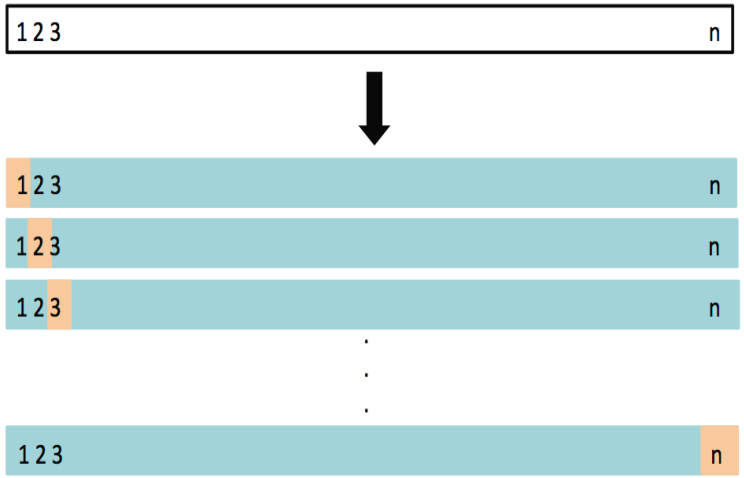
5-fold và 10-fold thường được ưa dùng (lỗi bias cao, phương sai thấp)



Kỹ thuật đánh giá chéo

- Khi ***k*** bằng với số lượng phần tử N trong tập *training* ban đầu, tức mỗi tập con có đúng 1 phần tử, ta gọi kỹ thuật này là leave-one-out cross validation (LOOCV).

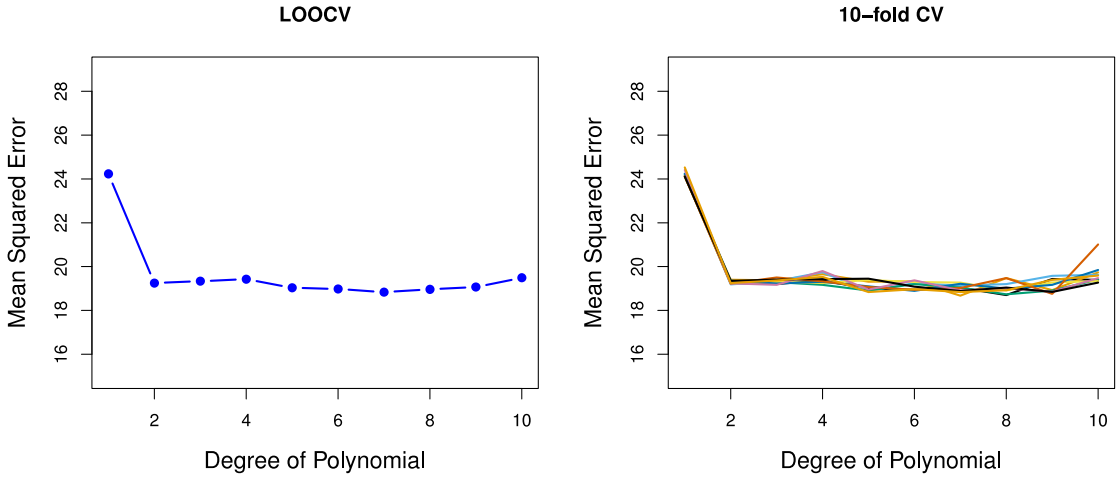
(lỗi bias thấp, phương sai cao)



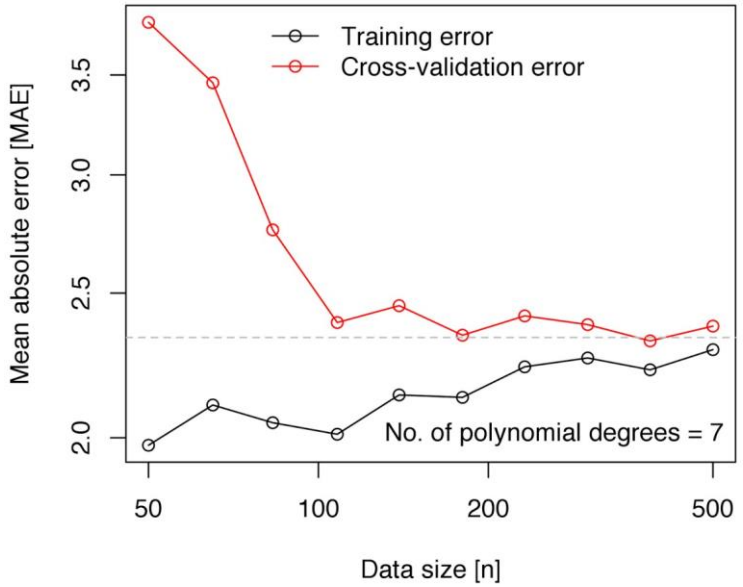
LOOCV vs. K-fold CV trên bộ dữ liệu Auto

Hình trái: Sai số LOOCV *là trường hợp đặc biệt của k-fold, khi k = N (392)*
Hình phải: 10-fold CV được chạy nhiều lần, đồ thị biểu diễn sai khác nhỏ về lỗi CV

Cả hai đều ổn định, tuy nhiên LOOCV mất nhiều thời gian tính toán hơn!



Kỹ thuật đánh giá chéo



Ta cần thêm biến (mô hình mới) hoặc thêm dữ liệu?

Nhược điểm của CV

- Nhược điểm lớn của *cross-validation* là số lượng *training runs* tỉ lệ thuận với k. Trong các bài toán Machine Learning, lượng tham số cần xác định thường lớn và khoảng giá trị của mỗi tham số cũng rộng. Việc xây dựng một mô hình cũng rất phức tạp.

$\Rightarrow$ Giải pháp giúp số mô hình cần huấn luyện giảm đi nhiều, thậm chí chỉ một mô hình. Cách này có tên gọi chung là *điều chỉnh mô hình* (*regularization*).

Điều chỉnh mô hình

- *Regularization*, một cách cơ bản, là điều chỉnh mô hình một chút để tránh overfitting trong khi vẫn giữ được tính tổng quát của nó (tính tổng quát là tính mô tả được nhiều dữ liệu, trong cả tập training và test).
- Một cách cụ thể hơn, ta sẽ tìm cách *di chuyển* nghiệm của bài toán tối ưu hàm tổn thất tới một điểm gần nó. Hướng di chuyển sẽ là hướng làm cho mô hình *ít phức tạp hơn* mặc dù giá trị của hàm tổn thất có tăng lên một chút.