

# Regression

## (Hồi quy)

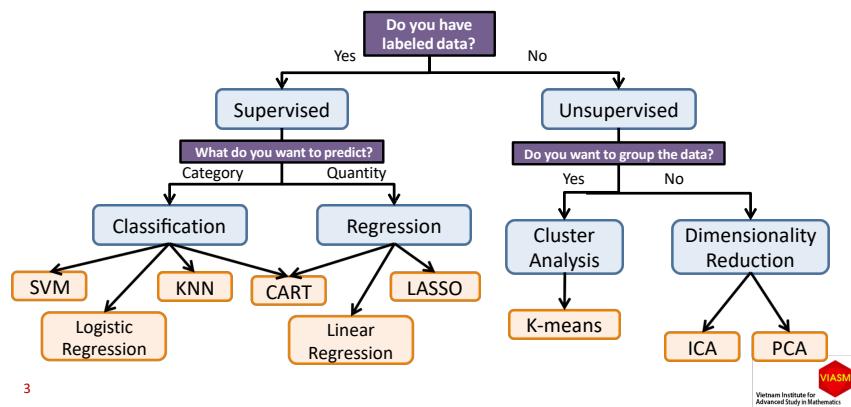
Nguyễn Thanh Tùng  
Bài giảng của DSLab  
Viện nghiên cứu cao cấp về Toán (VIASM)  
<https://www.facebook.com/tungntdhtl>

## Nội dung

1. Giới thiệu mô hình hồi quy
2. Hồi quy tuyến tính
3. Hồi quy phi tuyến

2

## Các dạng giải thuật học máy



3

## Mô hình Hồi quy

- 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
- $$(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)})$$
- $Y$  có kiểu dữ liệu liên tục

4



## Ví dụ về hồi quy

Cho bảng dữ liệu thông tin nhiên liệu như sau:

Bài toán đặt ra: liệu có thể dự đoán nhiên liệu do phi công lựa chọn (FUEL\_ORDER) của mỗi chuyến bay dựa vào nhiên liệu do máy tính cung cấp (BLOCK\_FUEL)?

LEG_NO	Ac_Type	VG_REGI	FLTNO	BLOCK_FUEL	FUEL_ORDER
40726405	321	VNA392	VN7239	8286	8800
42344242	321	VNA392	VN7239	8186	9200
40725994	321	VNA392	VN223	9140	10400
40749135	321	VNA392	VN233	8246	9300
40749680	321	VNA392	VN267	8054	9000
40726477	321	VNA392	VN249	17330	8000
40718529	321	VNA392	VN277	8159	9400
42344216	321	VNA392	VN7237	9120	9200
40749142	321	VNA392	VN233	8649	9000
40788017	321	VNA392	VN7225	7957	8700
40726029	321	VNA392	VN223	8041	9200
40718557	321	VNA392	VN277	8594	9900
40726459	321	VNA392	VN239	8590	9600
40734347	321	VNA392	VN7189	9428	11000
40713687	321	VNA392	VN237	8279	9500
40726053	321	VNA392	VN223	8021	8800
43229724	321	VNA392	VN223	8172	9600
43174863	321	VNA392	VN253	9352	10000
43177408	321	VNA392	VN253	9318	10000
43177213	321	VNA392	VN7255	8613	10000
43208049	321	VNA392	VN275	9492	10500
43159802	321	VNA392	VN233	8019	9600
43160123	321	VNA392	VN239	10768	11700
43160125	321	VNA392	VN239	10993	11000
43207863	321	VNA392	VN249	9986	11500

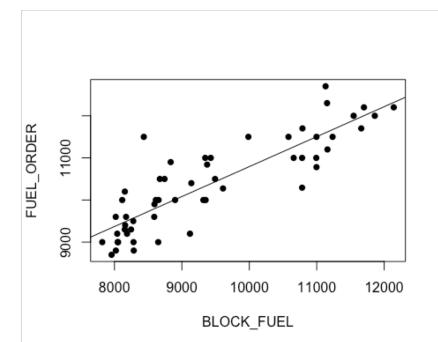
5

## Ví dụ về hồi quy

```
VNA392=read.csv("data/VNA392_
HANSGN_2016_1.csv")
```

```
attach(VNA392)
```

```
plot(BLOCK_FUEL, FUEL_ORDER,
pch=16)
```



6

## Mô hình Hồi quy

- Giải thuật học
  - 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

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

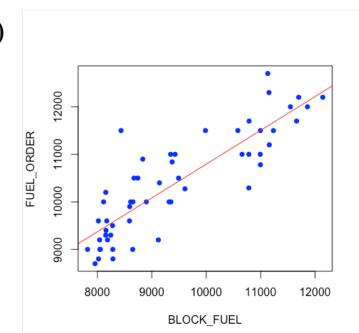
7

## Hàm tổn thất

$$L(\theta_i, \hat{\theta}_i)$$

Sai số bình phương (Squared error)  $\sum_i (\theta_i - \hat{\theta}_i)^2$

Sai số tuyệt đối (Absolute error)  $\sum_i |\theta_i - \hat{\theta}_i|$



8



## Bài toán Hồi quy

$$\hat{f} = \underset{\tilde{f}}{\operatorname{argmin}} E[L(Y, \tilde{f}(X))]$$

argument minimum: Cho giá trị nhỏ nhất của 1 hàm số trong miền xác định

9



## Nội dung

1. Giới thiệu mô hình hồi quy
2. **Hồi quy tuyến tính**
3. Hồi quy phi tuyến

11



## 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

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

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

10



## 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

12

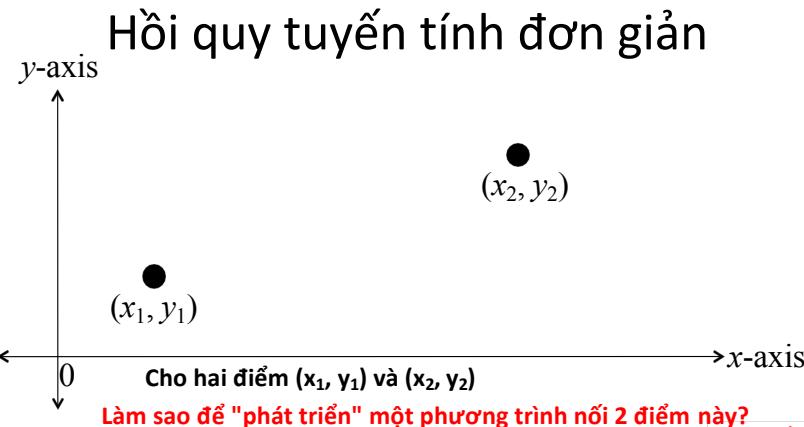


## Hồi quy tuyến tính

### 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
- Điển giải các mối quan hệ giữa biến đầu vào và đầu ra - sử dụng cho bài toán suy diễn

13



15

## 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:

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

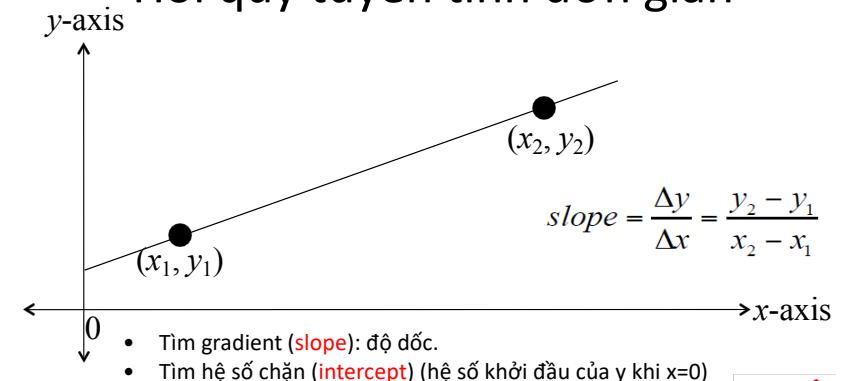
- Các tham số của mô hình:

$\beta_0$  intercept      hệ số chặn (khi các  $x_i=0$ )  
 $\beta_1$  slope              độ dốc

14

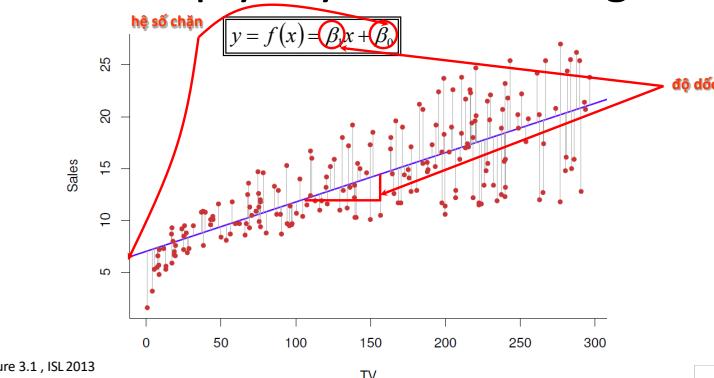


### Hồi quy tuyến tính đơn giản



16

## Hồi quy tuyến tính đơn giản



17



## 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 ( $\varepsilon$ ): phân bố chuẩn, trung bình 0, phương sai bất biến

$$\varepsilon \sim N(0, \sigma^2)$$

19



## Hồi quy tuyến tính đơn giản

- $\beta_0$  và  $\beta_1$  chưa biết  $\rightarrow$  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

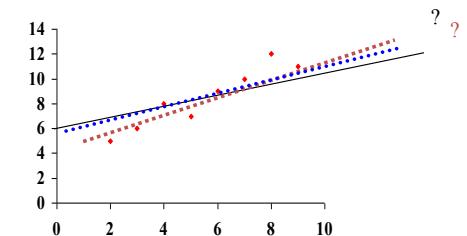
$$Y^{(i)} \approx \hat{\beta}_0 + \hat{\beta}_1 X^{(i)}, \quad i = 1, \dots, n$$

18



## 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



20



## 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):

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

21



## Phần dư (lỗi)

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

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

Đường thẳng phù hợp nhất tìm thấy khi tổng bình phương lỗi là nhỏ nhất

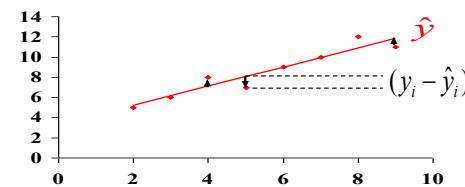
$$SSE = \sum_{i=1}^n (y_i - \hat{y})^2$$

23



## 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



22



## Ước lượng tham số

- Các ước số  $\hat{\beta}_0, \hat{\beta}_1$  tính được bằng cách cực tiểu hóa MSE

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

- Hệ số chặn của đường thẳng  $\hat{\beta}_1 = \frac{SS_{xy}}{SS_x}$
- trong đó:  $SS_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$  và  $SS_x = \sum_{i=1}^n (x_i - \bar{x})^2$

24



## Ước lượng tham số

Hệ số chẵn của đường thẳng

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

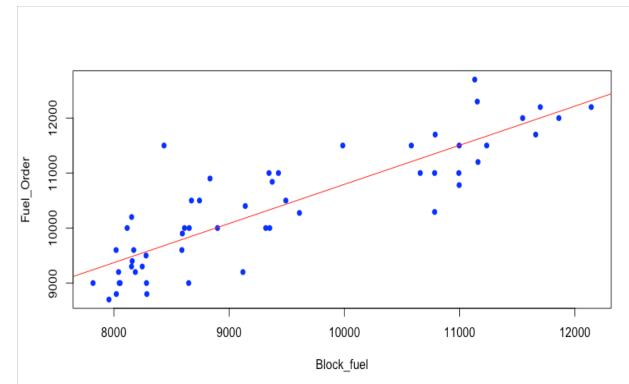
trong đó

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

25



## Hồi quy tuyến tính đơn giản



26

## Phương pháp đánh giá

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2}; MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i|$$

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

27



## Ví dụ

X kilograms	Y cost \$	
17	132	$\bar{x} = 37.83$
21	150	$\hat{\beta}_1 = \frac{SS_{xy}}{SS_x} = \frac{891.83}{1612.83} = 0.533$
35	160	$\bar{y} = 153.83$
39	162	$SS_{xy} = 891.83$
50	149	$SS_x = 1612.83$
65	170	$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 153.83 - 0.533 \times 37.83 = 132.91$

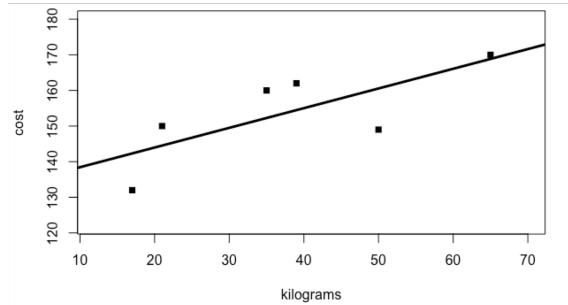
phương trình tìm được là

$$Y = 132.91 + 0.533 * X$$

28

## Điễn 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 \$

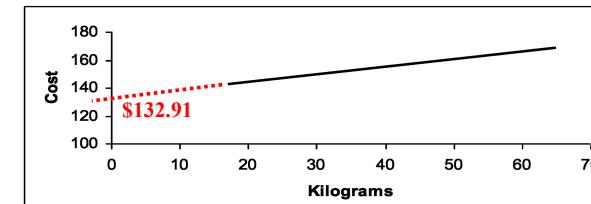


29



## Điễn 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



30



## Ước tính bằng R

- Chúng ta muốn ước tính mối liên quan giữa lượng nhiên liệu cung cấp bởi máy tính (Block\_Fuel) và nhiên liệu do phi công lựa chọn (Fuel\_order).
- Mô hình hồi qui tuyến tính:

$$\text{Fuel\_order} = \beta_0 + \beta_1 * \text{Block\_Fuel} + \varepsilon$$

- R

```
lm(FUEL_ORDER ~ BLOCK_FUEL, data = VNA392)
```

31



## Phân tích bằng R

```
# Phân tích hồi qui tuyến tính  
m1=lm(FUEL_ORDER ~ BLOCK_FUEL, data = VNA392)  
summary(m1)
```

```
# vẽ biểu đồ  
plot(VNA392$BLOCK_FUEL, VNA392$FUEL_ORDER,  
pch=16, col="blue", xlab = "Block_fuel", ylab =  
"Fuel_Order")  
abline(m1, col="red")
```

32



```
m1=lm(FUEL_ORDER ~ BLOCK_FUEL, data = VNA392)
summary(m1)
```

#### Residuals:

Min	1Q	Median	3Q	Max
-1057.4	-326.2	-100.2	274.8	1820.0

#### Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.680e+03	5.532e+02	6.653	1.77e-08 ***
BLOCK_FUEL	7.113e-01	5.813e-02	12.235	< 2e-16 ***

--Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 556.8 on 52 degrees of freedom  
Multiple R-squared: 0.7422, Adjusted R-squared: 0.7372  
F-statistic: 149.7 on 1 and 52 DF, p-value: < 2.2e-16

- $R^2$  (hệ số xác định): là chỉ số rất có ích trong mô hình hồi qui tuyến tính.
- $R^2 \times 100$  có nghĩa là phần trăm variation của biến y có thể giải thích bởi biến x
- $R^2=1$ : tất cả dữ liệu có mối liên hệ xác định
- $R^2=0$ : Không có mối quan hệ nào giữa X và Y.

33

## Phân tích bằng R



## Hồi quy tuyến tính đa biến

- Hồi quy tuyến tính đa biến:** mô hình có nhiều hơn 1 biến dùng để dự đoán biến đích

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

35



## Diễn giải kết quả

#### Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.680e+03	5.532e+02	6.653	1.77e-08 ***
BLOCK_FUEL	7.113e-01	5.813e-02	12.235	< 2e-16 ***

- Nhớ rằng mô hình là:

$$FUEL\_ORDER = \beta_0 + \beta_1 * BLOCK\_FUEL$$

- Phương trình:

$$FUEL\_ORDER = 3680 + 0.711 * BLOCK\_FUEL$$

- Ý nghĩa: phi công tăng 1000 kg mỗi khi chương trình máy tính tăng 711 kg nhiên liệu cho từng chuyến bay.

Mối tương quan này có ý nghĩa thống kê ( $P < 0.0001$ )

34



## Hồi quy tuyến tính đa biến

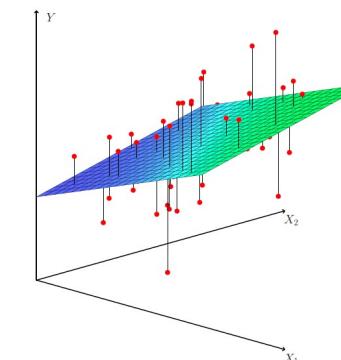


Figure 3.4 , ISL 2013

36



## Hồi quy tuyến tính đa biến

- Diễn giải hệ số  $\beta_j$ :  
khi tăng  $X_j$  lên một đơn vị → Y sẽ tăng trung bình một lượng là  $\beta_j$



37

## Hồi quy tuyến tính đa biến

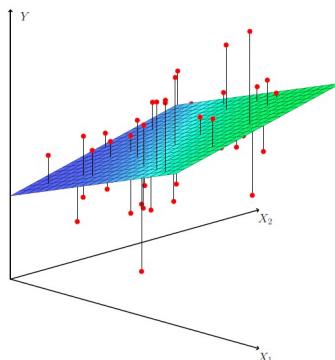


Figure 3.4 , ISL 2013

39

## Bình phương nhỏ nhất

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

$$\hat{\beta} = \arg \min_{\beta} \|Y - X^T \beta\|^2 \quad X = \begin{bmatrix} 1 & X^{(1)^T} \\ & \vdots \\ 1 & X^{(n)^T} \end{bmatrix} \quad \hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_d \end{bmatrix}$$
$$Y = \begin{bmatrix} Y^{(1)} \\ \vdots \\ Y^{(n)} \end{bmatrix}$$

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

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

38



## Ví dụ

Cho

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

40



## Ví dụ

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

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

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

41



## Ví dụ

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \begin{bmatrix} 2.59578 & -0.15375 & -0.01962 & -0.13737 \\ -0.15375 & 0.03965 & -0.00014 & -0.00144 \\ -0.01962 & -0.00014 & 0.01234 & -0.00431 \\ -0.13737 & -0.00144 & -0.00431 & 0.01406 \end{bmatrix} \begin{bmatrix} 47 \\ 203 \\ 277 \\ 598 \end{bmatrix} = \begin{bmatrix} 3.20975 \\ -0.07573 \\ -0.11162 \\ 0.46691 \end{bmatrix}$$

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

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

42



## 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
- 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 dung phương pháp bình phương nhỏ nhất

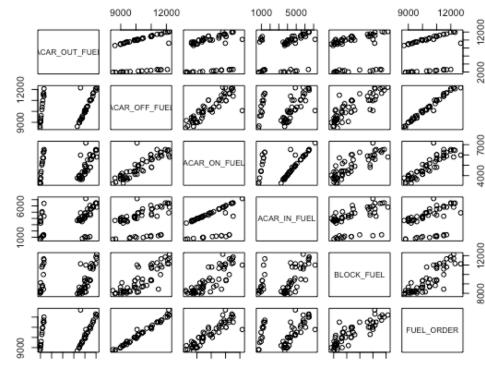
43



m2<-  
lm(FUEL\_ORDER ~ .,  
data = VNA392)  
pairs(VNA392)

Coefficients:  
(Intercept) ACAR\_OUT\_FUEL ACAR\_OFF\_FUEL ACAR\_ON\_FUEL ACAR\_IN\_FUEL BLOCK\_FUEL  
6.743e+02 8.724e-05 8.476e-01 1.225e-01 7.307e-03 3.569e-02

44



## Q?&A!

## Nội dung

1. Giới thiệu mô hình hồi quy
2. Hồi quy tuyến tính
3. Hồi quy phi tuyến

45



46



## Phương pháp kết hợp các mô hình (ensemble models)

Cây phân loại và hồi quy  
Classification and Regression Trees  
(CART)

47



48



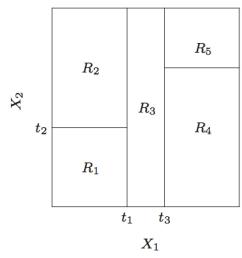
## Xây dựng cây CART thế nào?

Có 2 dạng:

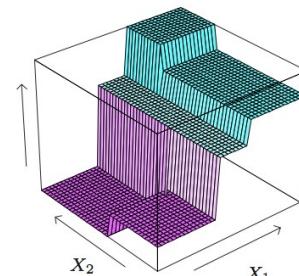
1. Hồi quy

2. Phân loại (lớp)

49



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



## Mô hình liên tục từng đoạn (piecewise)

- Dự đoán liên tục trong mỗi vùng

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m)$$

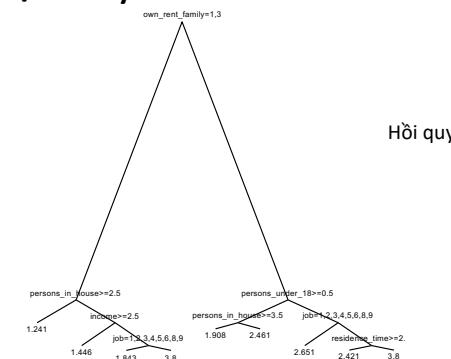
50

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



## Mô hình liên tục từng đoạn

## Minh họa cây CART



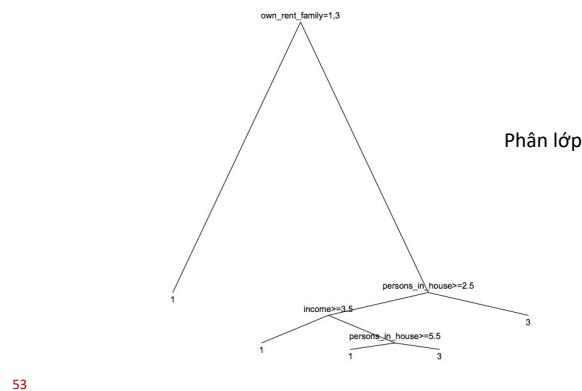
Hồi quy

51

52



## Minh họa cây CART



## Cây hồi quy

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m)$$

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$$

Giá trị dự đoán lưu tại lá của cây hồi quy. Nó được tính bằng giá trị trung bình của tất cả các mẫu (bản ghi) tại lá đó.

54



## Cây hồi quy

- Giả sử ta có 2 vùng  $R_1$  và  $R_2$  với  $\hat{Y}_1 = 10, \hat{Y}_2 = 20$
- Với các giá trị của  $X$  mà  $X \in R_1$  ta sẽ có giá trị dự đoán là 10, ngược lại  $X \in R_2$  ta có kết quả dự đoán là 20.

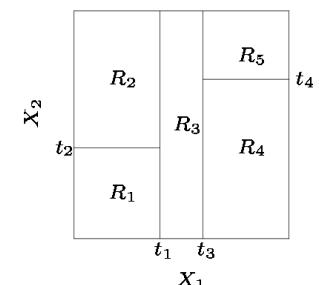
55



## Cây hồi quy

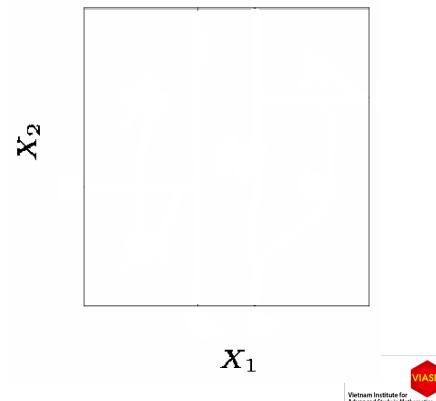
- Cho 2 biến đầu vào và 5 vùng
- Tùy theo từng vùng của giá trị mới  $X$  ta sẽ có dự đoán 1 trong 5 giá trị cho  $Y$ .

56



## Tách các biến X

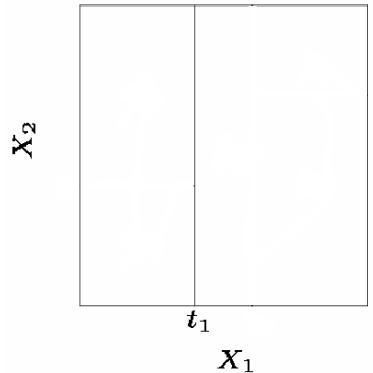
Ta tạo ra các phân vùng bằng cách tách lặp đi lặp lại một trong các biến X thành hai vùng



57

## Tách các biến X

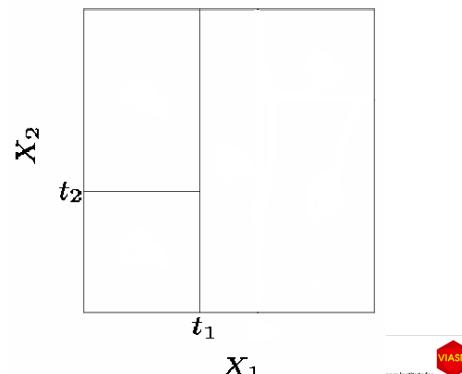
1. Đầu tiên tách trên  $X_1=t_1$



58

## Tách các biến X

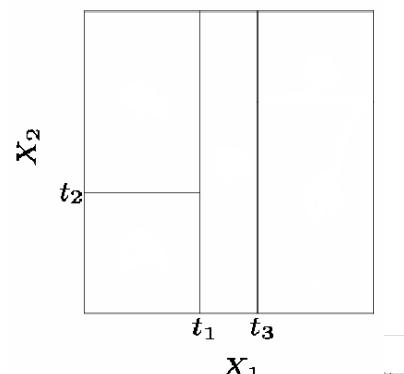
1. Đầu tiên tách trên  $X_1=t_1$
2. Nếu  $X_1 < t_1$ , tách trên  $X_2=t_2$



59

## Tách các biến X

1. Đầu tiên tách trên  $X_1=t_1$
2. Nếu  $X_1 < t_1$ , tách trên  $X_2=t_2$
3. Nếu  $X_1 > t_1$ , tách trên  $X_1=t_3$

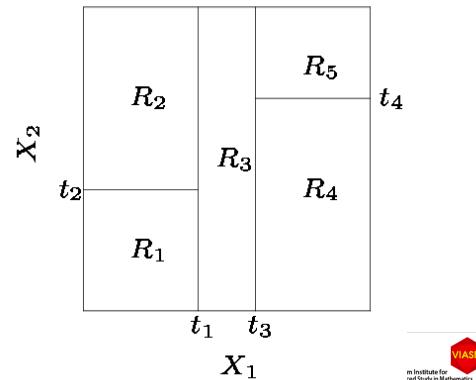


60



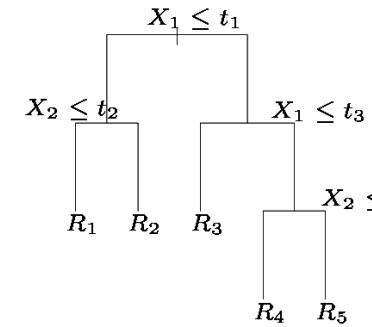
## Tách các biến X

1. Đầu tiên tách trên  $X_1=t_1$
2. Nếu  $X_1 < t_1$ , tách trên  $X_2=t_2$
3. Nếu  $X_1 > t_1$ , tách trên  $X_1=t_3$
4. Nếu  $X_1 > t_3$ , tách  $X_2=t_4$

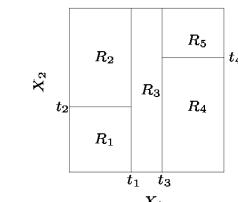


61

## Tách các biến X



62



- Khi ta tạo các vùng theo phương pháp này, ta có thể biểu diễn chúng dùng cấu trúc cây.
- Phương pháp này dễ diễn giải mô hình dự đoán, dễ diễn giải kết quả



## Ưu điểm của CART

- Dễ xử lý dữ liệu thiếu (surrogate splits)
- Mạnh trong xử lý dữ liệu chứa thông tin rác (non-informative data)
- Cho phép tự động lựa chọn thuộc tính (variable selection)
- Dễ giải thích, lý tưởng để giải thích “tại sao” đối với người ra quyết định
- Xử lý được tính tương tác cao giữa các thuộc tính

63

## Nhược điểm của CART

- Cây không ổn định (Instability of trees)
- Thiếu tính trơn (Lack of smoothness)
- Khó nắm bắt độ cộng tính (Hard to capture additivity)

64



# Ensemble Models

65



Some characteristics of different learning methods. Key: ▲ = good, ▼ = fair, and ▽ = poor.

Characteristic	Neural Nets	SVM	Trees	Random forest	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▲	▲
In sensitive to monotone transformations of inputs	▼	▼	▲	▲	▼
Computational scalability (large $N$ )	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▲	◊
Interpretability	▼	▼	◊	▼	▼
Predictive power	▲	▲	▼	▲	▲

Fernández-Delgado, Manuel, et al. "Do we need hundreds of classifiers to solve real world classification problems?" *The Journal of Machine Learning Research* 15.1 (2014): 3133-3181.

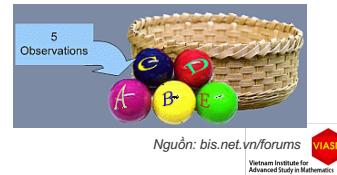
Kết luận của nghiên cứu trên của nhóm Manuel là phương pháp Random Forests hầu hết cho kết quả tốt nhất.

66

## Bootstrap là gì?

- Giả sử ta có 5 quả bóng gắn nhãn A,B,C,D, E và bỏ tất cả chúng vào trong 1 cái giỏ.
- Lấy ra ngẫu nhiên 1 quả từ giỏ và ghi lại nhãn, sau đó bỏ lại quả bóng vừa bốc được vào giỏ.
- Tiếp tục lấy ra ngẫu nhiên một quả bóng và lặp lại quá trình trên cho đến khi việc lấy mẫu kết thúc. Việc lấy mẫu này gọi là lấy mẫu có hoàn lại.
- Kết quả của việc lấy mẫu như trên có thể như sau (giả sử kích thước mẫu là 10):

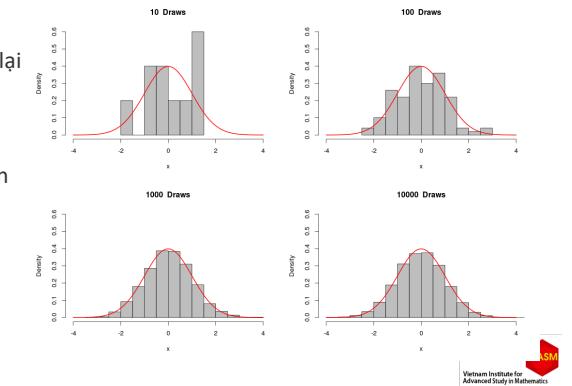
C, D, E, E, A, B, C, B, A, E



67

## Bootstrap là gì?

- Bootstrap là phương pháp lấy mẫu có hoàn lại (sampling with replacement)-> một mẫu có thể xuất hiện nhiều lần trong một lần lấy mẫu



68

## Bootstrap là gì?

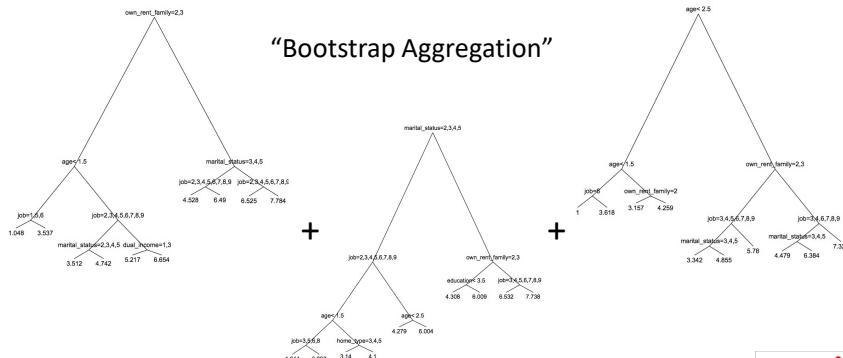
- Là kỹ thuật rất quan trọng trong thống kê
- Lấy mẫu có hoán lại từ tập dữ liệu ban đầu để tạo ra các tập dữ liệu mới

69



## Các phương pháp kết hợp: Bagging

### Bagging là gì?



71

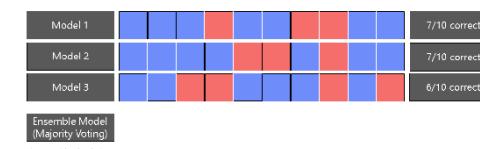


70



### Bagging là gì?

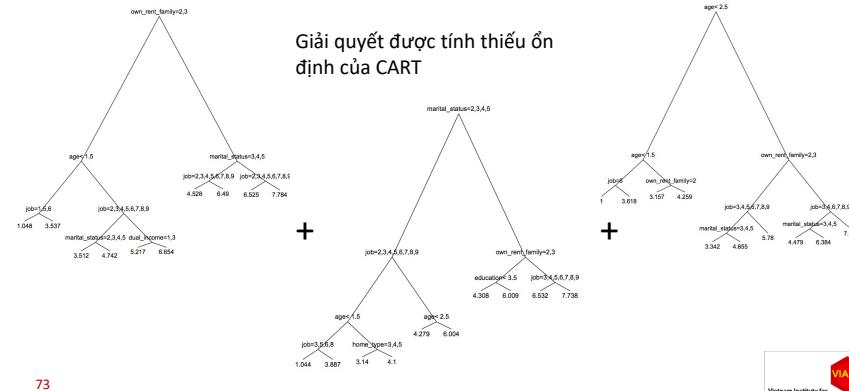
“Bootstrap Aggregation”  $\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$



72

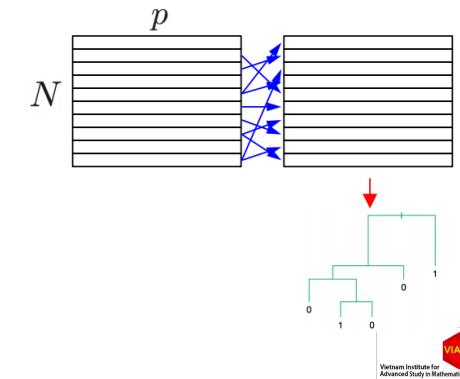


# Bagging



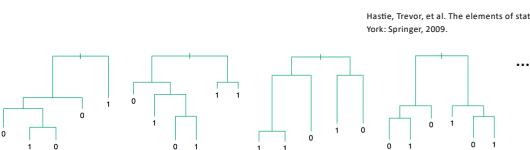
# Bagging

- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.



# Bagging

- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.



- Lấy trung bình (hoặc bình chọn theo số đông- majority vote) các bộ dự đoán độc lập.
- Bagging giảm phương sai (variance) và giữ bias.

75



# Bagging

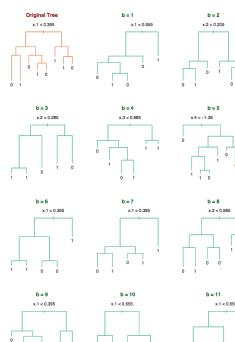
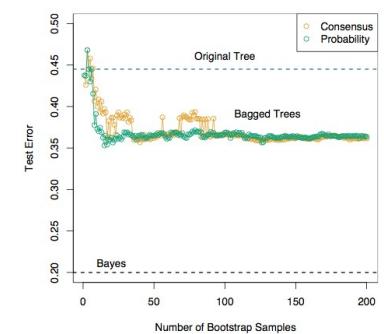


FIGURE 8.9 Bagging trees on simulated dataset. The top left panel shows the original tree. Eleven trees grown on bootstrap samples are shown. For each tree, the top split is annotated.

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

76



## Bagging

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

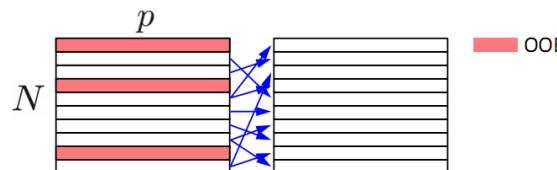
- Lấy mẫu có hoàn lại
- Xây dựng bộ phân lớp trên mỗi mẫu bootstrap
- Mỗi mẫu bootstrap chứa xấp xỉ 63.2% số lượng mẫu trong tập dữ liệu ban đầu
- Số lượng mẫu còn lại (36.8%) được dùng để kiểm thử

77



## Các mẫu Out-of-bag (OOB)

- Quá trình Bootstrapping:



- Mỗi cây chỉ sử dụng một tập con các mẫu huấn luyện (trung bình số mẫu  $\sim 2/3$ ).
- Số mẫu cho OOB khoảng  $\sim 1/3$  của cây quyết định.

79



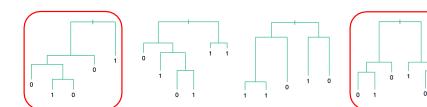
## Bonus! Out-of-bag cross-validation

78



## Dự đoán mẫu OOB

- Với mỗi mẫu, tìm các cây mà nó là OOB.



...

- Dự đoán giá trị của chúng từ các cây này.

- Ước lượng lỗi dự đoán của cây (bagged trees) dùng tất cả các dự đoán OOB.

- Tương tự như kỹ thuật kiểm tra chéo (cross-validation).

80



## Phương pháp Rừng ngẫu nhiên Random Forests (RF)

81

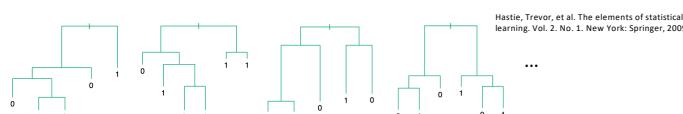


## Động lực để có Random forest

- Mô hình dựa trên cây phân loại và hồi quy (CART).
- Các mô hình cây có lỗi bias thấp, tuy nhiên phương sai lại cao (high variance).
- Phương pháp Bagging dùng để giảm phương sai.

## Nhắc lại: Bagging

- Lấy mẫu tập dữ liệu huấn luyện theo Bootstrap để tạo ra tập hợp các dự đoán.
- Lấy trung bình (hoặc bình chọn theo số đông-majority vote) các bộ dự đoán độc lập.
- Bagging giảm phương sai (variance) và giữ bias.



83



82



## Bagged trees vs. random forests

- Phương pháp Bagging biểu thị sự biến thiên (variability) giữa các cây bởi việc chọn mẫu ngẫu nhiên từ dữ liệu huấn luyện.
- Cây được sinh ra từ phương pháp Bagging vẫn có tương quan lẫn nhau, do đó hạn chế trong việc giảm phương sai.

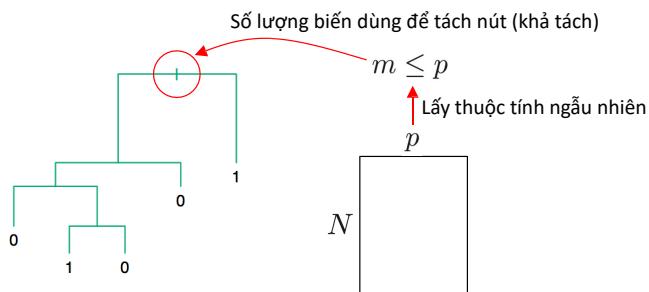
Random forests đưa ra thêm tính ngẫu nhiên (randomness):

- Làm giảm mối tương quan giữa các cây bằng cách lấy ngẫu nhiên các biến khi tách nút của cây.

84

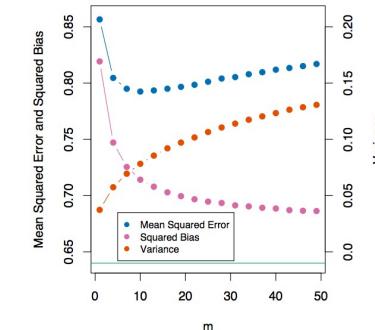


## Các biến dùng cho tách nút



85

## Các biến dùng cho tách nút

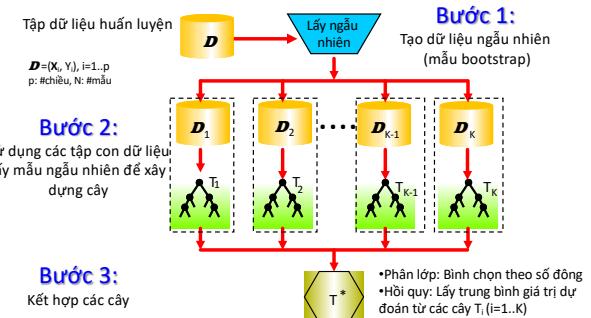


Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.



86

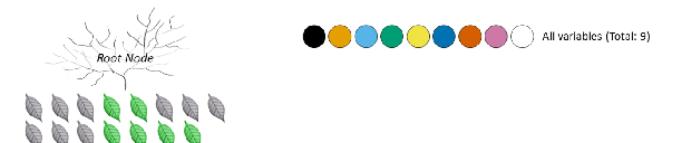
## Rừng ngẫu nhiên



Introduction to Data Mining – Tan, Steinbach, Kumar

87

## Rừng ngẫu nhiên



For more tutorials: digibearis.com

88



## Các tham số chính

Các tham số quan trọng của Rừng ngẫu nhiên:

- Số lượng biến khả tách tại mỗi nút ( $m$ )
- Độ sâu của từng cây trong rừng (số lượng mẫu tối thiểu tại mỗi nút của cây-minimum node size)
- Số lượng cây trong rừng

Giá trị mặc định

Bài toán phân lớp  $m = \lfloor \sqrt{p} \rfloor$

Bài toán hồi quy  $m = \lfloor p/3 \rfloor$

gói randomForest trong R dùng `mtry`

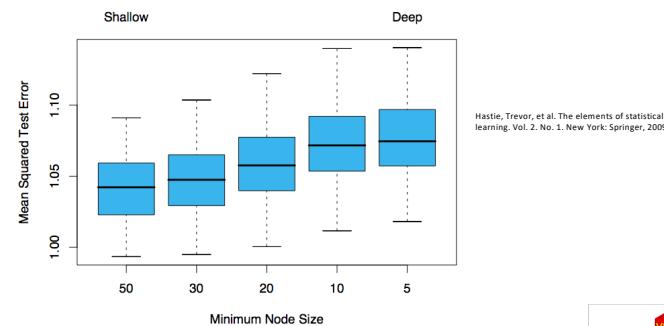
89



90



Độ sâu của từng cây  
(số lượng mẫu tối thiểu tại mỗi nút của cây)



91



Độ sâu của cây

92



92

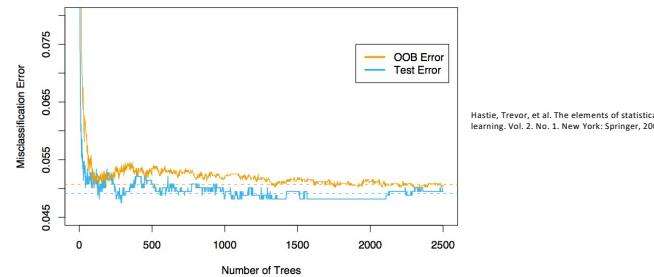
Giá trị mặc định

Bài toán phân lớp 1

Bài toán hồi quy 5



## Số lượng cây trong rừng



- Thêm nhiều cây không gây ra overfitting.

93

## Các tính năng khác của RF

- Các mẫu Out-of-bag (OOB)
- Độ quan trọng của biến (Variable importance measurements)

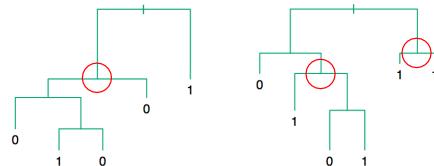
94



## Độ quan trọng của biến

### Dạng 1:

Độ giảm của lỗi dự đoán hoặc impurity từ các điểm tách nút liên quan đến các biến đó, cuối cùng lấy trung bình trên các cây trong rừng.

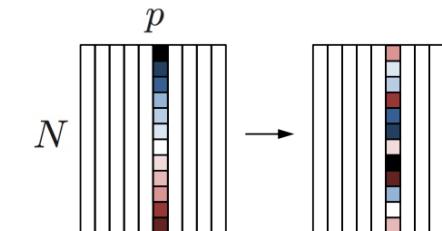


95

## Độ quan trọng của biến

### Dạng 2:

Độ tăng lỗi dự đoán tổng thể khi các giá trị của biến được hoán vị ngẫu nhiên giữa các mẫu.

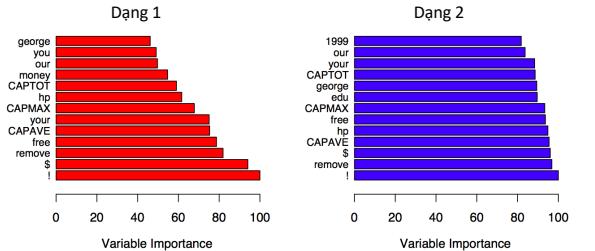


96



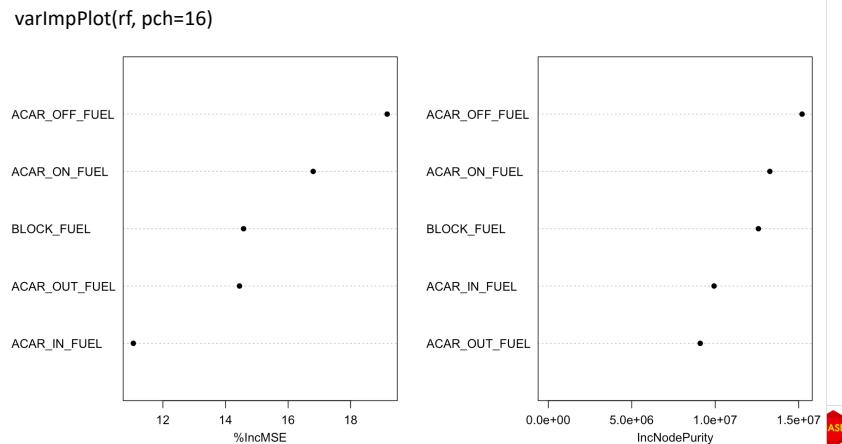
## Ví dụ về độ quan trọng của biến

- Cả 2 dạng biểu thị gần giống nhau, tuy nhiên có sự khác biệt về xếp hạng các biến:



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

97



98

```
library(randomForest)
rf=randomForest(FUEL_ORDER ~ ., data = VNA392, importance=T)
```

```
randomForest(formula = FUEL_ORDER ~ ., data = VNA392, importance = T)
```

Type of random forest: regression

Number of trees: 500

No. of variables tried at each split: 1

Mean of squared residuals: 89095.72

% Var explained: 92.31

98

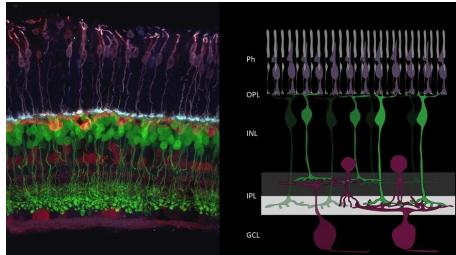


## Mạng Nơ-ron nhân tạo Neural Networks

100

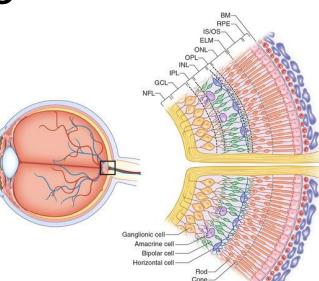


## Mạng nơ-ron nhân tạo

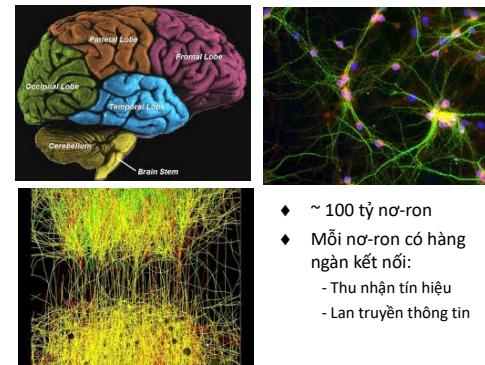


Bleckert A, Schwartz GW, Turner MH, Rieke F, Wong RO. Visual space is represented by nonmatching topographies of distinct mouse retinal ganglion cell types. *Curr Biol*. 2014 Feb 3;24(3):310-5.

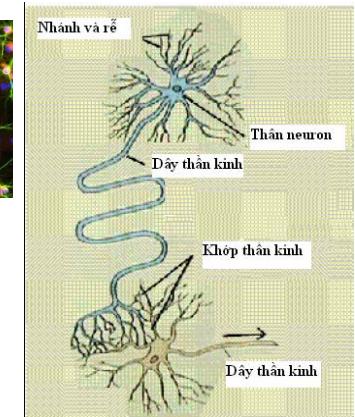
101



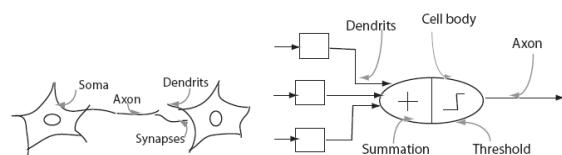
## Mạng nơ-ron sinh học



102



## Mô hình mạng nơ-ron nhân tạo



Biological vs. Artificial

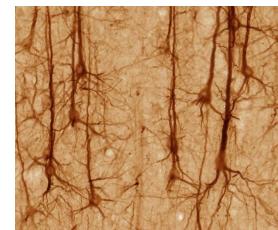
Biological vs. Artificial

103



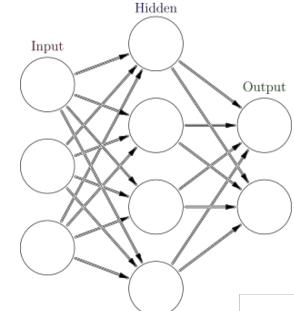
## Mạng nơ-ron nhân tạo

Mạng Nơ-ron

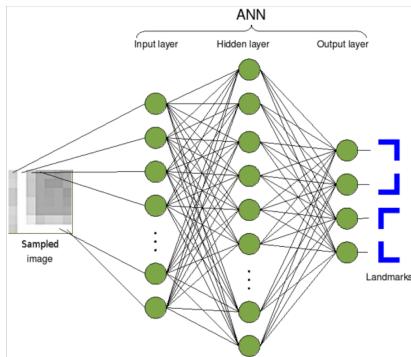


104

Mô hình mạng Nơ-ron nhân tạo



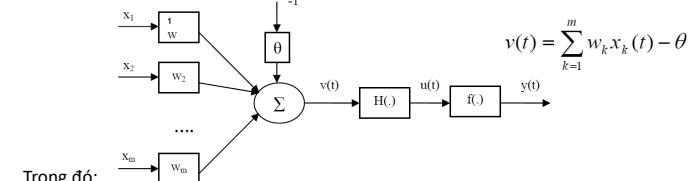
## Mạng nơ-ron nhân tạo



105



## Cấu trúc nơ-ron nhân tạo



Trong đó:

$v(t)$ : Tổng tất cả các đầu vào mô tả toàn bộ thế năng tác động ở thân nơ-ron.

$x_k(t)$ : Các biến đầu vào (các đặc trưng),  $k=1..M$ .

$w_k$ : Trọng số liên kết ngoài giữa các đầu vào k với nơ-ron hiện tại.

$H(\cdot)$ : Hàm kích hoạt.

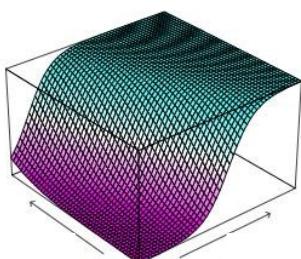
$y(t)$ : Tín hiệu đầu ra nơ-ron.

$\theta$ : Ngưỡng (là hằng số), xác định ngưỡng kích hoạt.

106

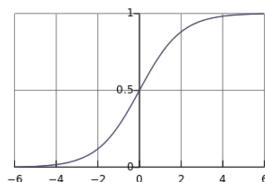


## Hàm Ridge



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

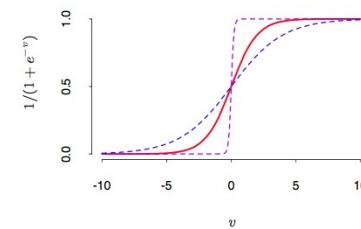
Hàm logistic



107



## Hàm kích hoạt Sigmoidal

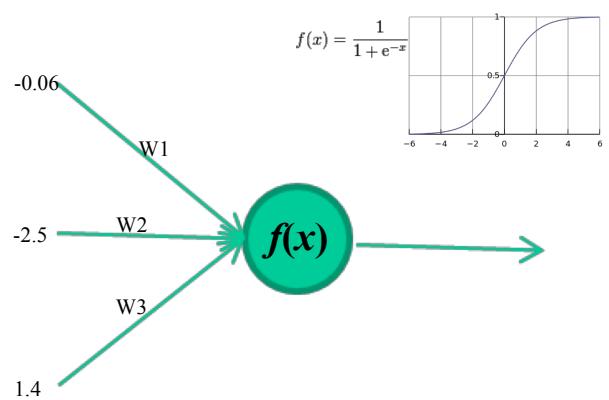


Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

**FIGURE 11.3.** Plot of the sigmoid function  $\sigma(v) = 1/(1+e^{-v})$  (red curve), commonly used in the hidden layer of a neural network. Included are  $\sigma(sv)$  for  $s = \frac{1}{2}$  (blue curve) and  $s = 10$  (purple curve). The scale parameter  $s$  controls the activation rate, and we can see that large  $s$  amounts to a hard activation at  $v = 0$ . Note that  $\sigma(s(v - v_0))$  shifts the activation threshold from 0 to  $v_0$ .

108

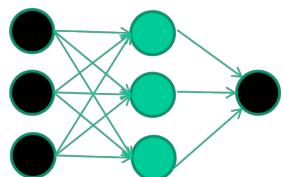




109

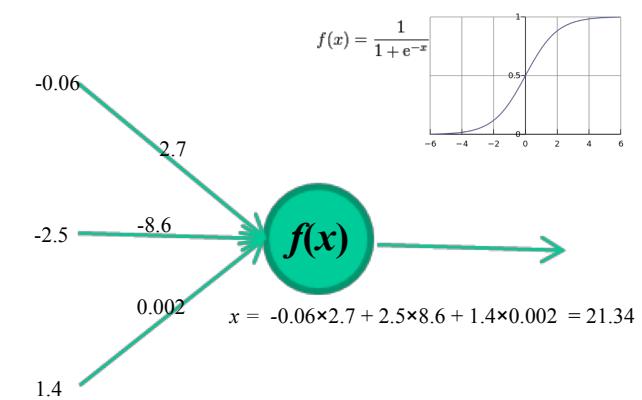
David Corne, Heriot-Watt University  


*Dữ liệu*  
**Các trường**    **Lớp**  
 1.4 2.7 1.9 0  
 3.8 3.4 3.2 0  
 6.4 2.8 1.7 1  
 4.1 0.1 0.2 0  
 V.V...



111

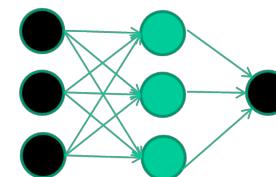
Vietnam Institute for  
Advanced Study in Mathematics  

110

David Corne, Heriot-Watt University  


*Huấn luyện mạng Nơ-ron*  
**Các trường**    **Lớp**  
 1.4 2.7 1.9 0  
 3.8 3.4 3.2 0  
 6.4 2.8 1.7 1  
 4.1 0.1 0.2 0  
 V.V...

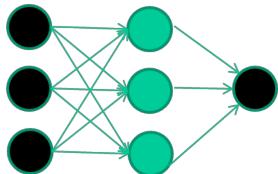


112

Vietnam Institute for  
Advanced Study in Mathematics  


Dữ liệu huấn luyện		
Các trường	Lớp	
1.4 2.7 1.9	0	
3.8 3.4 3.2	0	
6.4 2.8 1.7	1	
4.1 0.1 0.2	0	
v.v...		

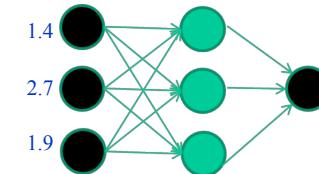
Khởi tạo các trọng số ngẫu nhiên



113

Dữ liệu huấn luyện		
Các trường	Lớp	
1.4 2.7 1.9	0	
3.8 3.4 3.2	0	
6.4 2.8 1.7	1	
4.1 0.1 0.2	0	
v.v...		

Huấn luyện mẫu

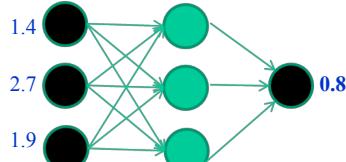


114



Dữ liệu huấn luyện		
Các trường	Lớp	
1.4 2.7 1.9	0	
3.8 3.4 3.2	0	
6.4 2.8 1.7	1	
4.1 0.1 0.2	0	
v.v...		

cung cấp đầu ra

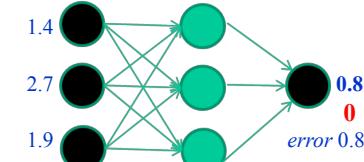


115



Dữ liệu huấn luyện		
Các trường	Lớp	
1.4 2.7 1.9	0	
3.8 3.4 3.2	0	
6.4 2.8 1.7	1	
4.1 0.1 0.2	0	
v.v..		

So sánh giá trị đầu ra

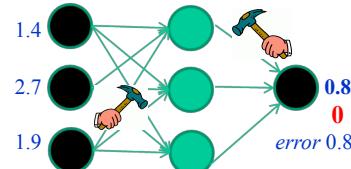


116



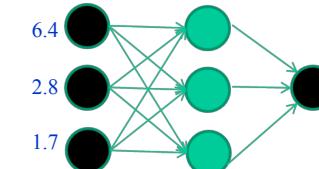
Dữ liệu huấn luyện			
Các trường	Lớp		
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v...			

Điều chỉnh các trọng số dựa vào đầu ra



Dữ liệu huấn luyện			
Các trường	Lớp		
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v...			

Huấn luyện mẫu



117

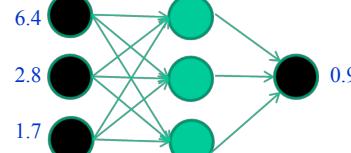


118



Dữ liệu huấn luyện			
Các trường	Lớp		
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v...			

cung cấp đầu ra



119

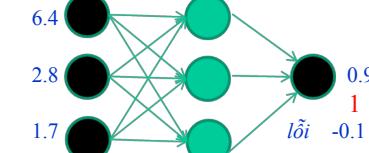


120



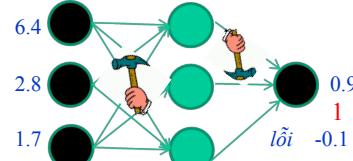
Dữ liệu huấn luyện			
Các trường	Lớp		
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
v.v...			

So sánh giá trị đầu ra



Dữ liệu huấn luyện			
Các trường	Lớp		
1.4	2.7	1.9	0
3.8	3.4	3.2	0
<b>6.4</b>	<b>2.8</b>	<b>1.7</b>	<b>1</b>
4.1	0.1	0.2	0
V.V...			

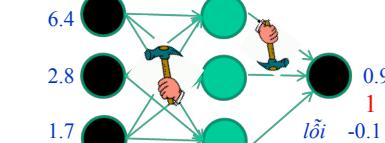
Điều chỉnh các trọng số dựa vào đầu ra



121

Dữ liệu huấn luyện			
Các trường	Lớp		
1.4	2.7	1.9	0
3.8	3.4	3.2	0
<b>6.4</b>	<b>2.8</b>	<b>1.7</b>	<b>1</b>
4.1	0.1	0.2	0
V.V...			

tiếp tục ....

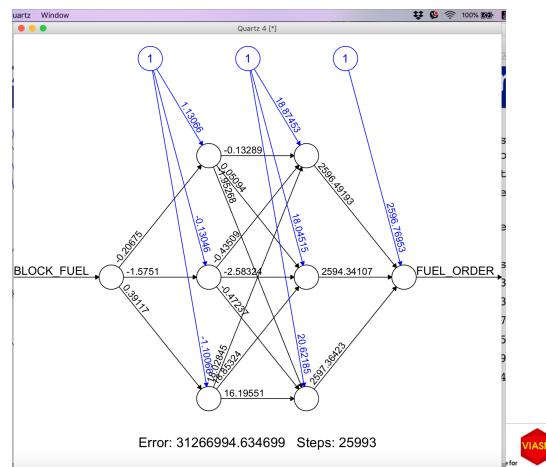


Lặp lại hàng ngàn, hàng triệu lần – mỗi lần sẽ lấy tập mẫu ngẫu nhiên, và tạo các điều chỉnh về trọng số  
Các giải thuật điều chỉnh trọng số được thiết kế để tạo ra các thay đổi mà chúng sẽ giúp giảm lỗi của mô hình

122



```
>library(neuralnet)
>nn = neuralnet
(FUEL_ORDER ~
BLOCK_FUEL,
data=VNA392,
hidden=c(3,3),
threshold=0.01)
>plot(nn)
```



123

## Dự đoán dữ liệu kiểm thử

```
trainData=VNA392[1:40, ]
testData=VNA392[41:54, ]
model.lm <- lm(FUEL_ORDER ~ BLOCK_FUEL, data= trainData)
pred=predict(model.lm, testData)
MSE=mean((testData$FUEL_ORDER - pred)^2)
MSE
# 299666.4399
model.rf <- randomForest(FUEL_ORDER ~ BLOCK_FUEL, data= trainData, trees=2000)
pred=predict(model.rf, testData)
MSE=mean((testData$FUEL_ORDER - pred)^2)
MSE
# 293049.5744
model.nn = neuralnet (FUEL_ORDER ~ BLOCK_FUEL, data=trainData, hidden=c(3,3),
threshold=0.01)
pred=compute(model.nn, as.matrix(testData$FUEL_ORDER))
pred$net.result
MSE=mean((testData$FUEL_ORDER - pred$net.result)^2)
MSE
#1200793.374
```

124



## Q?&A!

