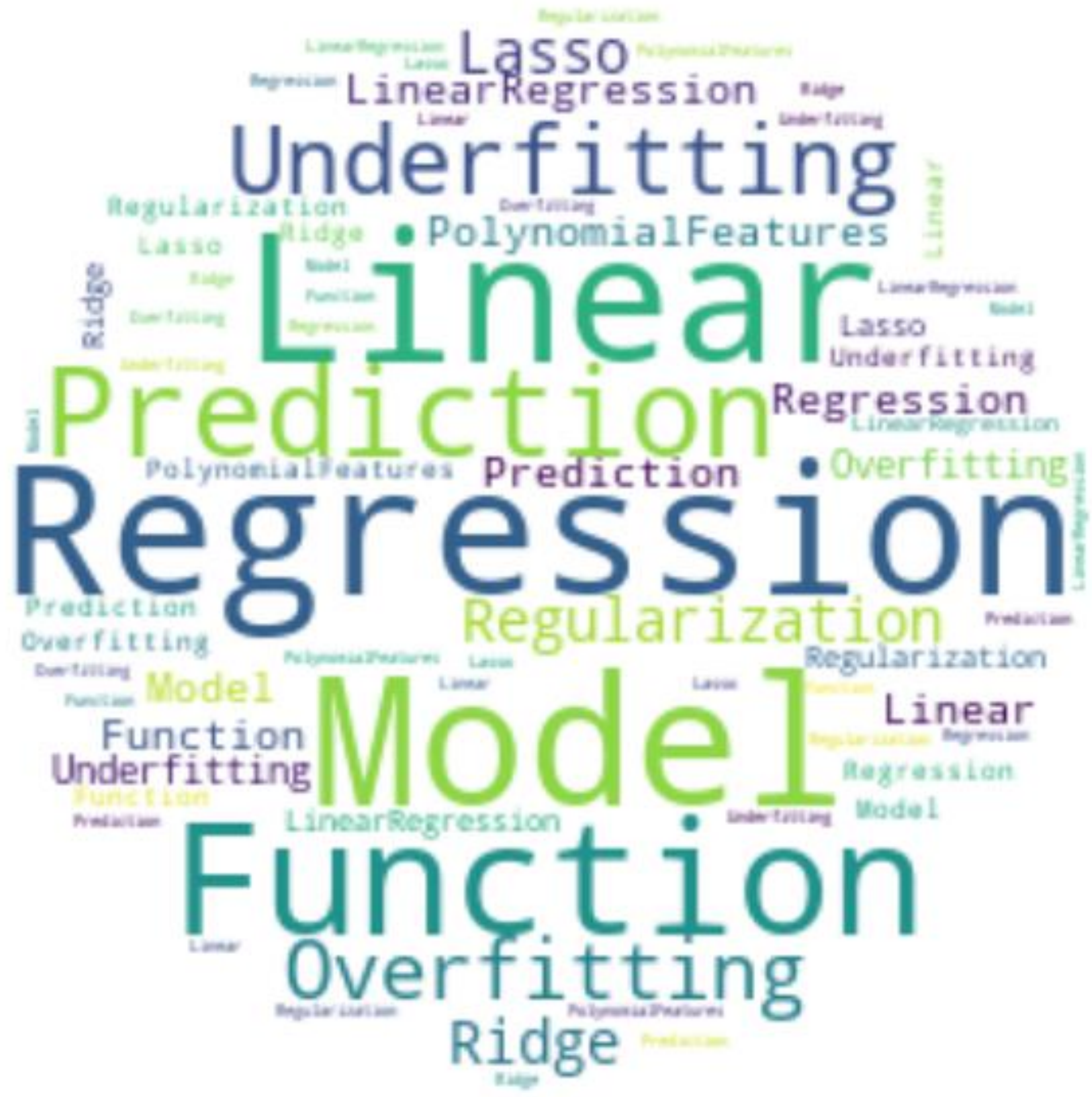

Lasso
LinearRegression
Underfitting
Regularization
Ridge
PolynomialFeatures
Linear
Ridge
Lasso
Underfitting
Prediction
Regression
LinearRegression
PolynomialFeatures
Prediction
Overfitting
Regression
Prediction
Regularization
Overfitting
Regularization
Model
Model
Linear
Function
Underfitting
Function
Regression
LinearRegression
Model
Function
Overfitting
Ridge

6222005 Học máy

Bài giảng: Xấp xỉ hàm

Chương 2: Xấp xỉ hàm và phân lớp

Ôn lại bài học trước

- Bạn có nhớ ? % ?

Nội dung chính

- 2.1 Khái niệm về xấp xỉ hàm và phân lớp
  - 2.1.1 Xấp xỉ hàm
  - 2.1.2 Phân lớp
- 2.2 Bài toán dùng xấp xỉ hàm
  - 2.2.1 Mô tả bài toán
  - 2.2.2 Hàm mục tiêu
  - 2.2.3 Các giải thuật hồi quy
  - 2.2.4 Ví dụ về bài toán xấp xỉ hàm

2.1 Khái niệm về xấp xỉ hàm và phân lớp

2.1.1 Xấp xỉ hàm

- Xét 1 ví dụ đơn giản

Tập dữ liệu nhà đất, nhiều căn nhà khác nhau
(Vị trí, diện tích, số phòng, …, giá)

$$f(.)$$

Xấp xỉ hàm
(Supervised L)

Dữ liệu mới
Một căn nhà **x**, (Vị trí, diện tích, số phòng, …, ~~giá~~)

Dự đoán
Giá căn nhà **x**?
Ví dụ là y (tỷ VND)

$$y \in \mathbb{R}$$

2.1.1 Xấp xỉ hàm

- Xấp xỉ hàm/hồi quy là quá trình tìm hàm số phù hợp nhất cho một tập dữ liệu, nhằm dự đoán mối quan hệ giữa các thuộc tính/biến đầu vào và các thuộc tính/biến đầu ra cần quan tâm
- Các thuộc tính đầu ra thường là số thực hoặc có giá trị liên tục

2.1.2 Phân lớp

- Xét 1 ví dụ đơn giản

Tập dữ liệu là các email (có đánh nhãn: bình thường, thư rác)

$$f(.)$$

Phân lớp (Supervised L)

Dữ liệu mới

Một bức thư mới nhận được **x** (dữ liệu mới)

Dự đoán (phân ra)

Bức thư mới thuộc lớp y nào? $y \in C\{C_0, C_1\}$

bình thường (0)

hay là thư rác (1)

2.1.2 Phân lớp

- Phân lớp là quá trình phân loại dữ liệu đầu vào thành các lớp hoặc danh mục khác nhau dựa trên các thuộc tính của nó.
- Mục tiêu là tìm cách ánh xạ từ các thuộc tính/biến đầu vào đến các thuộc biến đầu ra rời rạc
  – Thuộc tính/biến đầu ra đại diện cho lớp hoặc danh mục

2.2 Bài toán dùng xấp xỉ hàm

2.2.1 Mô tả bài toán

Có dữ liệu đầu vào, biểu diễn bởi vector đặc trưng $\mathbf{x}$

Tìm hàm số *f*() có thể dự đoán được giá trị đầu ra (số vô hướng, có giá trị liên tục)

$$f(\;)$$

$$\mathbf{x} \Rightarrow \boxed{\text{Mô hình ML}} \Rightarrow y \approx \hat{y} = f(\mathbf{x})$$

$$\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$$

M đặc trưng (feature)

x_i :Giá trị cụ thể của thuộc tính thứ *i*

2.2.2 Hàm mục tiêu

y : giá trị đúng (mong đợi)

$\hat{y}$: giá trị dự đoán

$e = y - \hat{y}$: sai số dự đoán, càng nhỏ càng tốt

Xét nhiều điểm/mẫu dữ liệu $\left(\mathbf{x}_i, y_i\right)\ i = 1, 2, ..., N$

N : số lượng điểm dữ liệu xem xét

$$MSE = \frac{1}{N}\sum_{i=1}^{N} e_i^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2$$

MSE: Mean squared error

Hàm mất mát => đạt giá trị nhỏ nhất

(loss function)

$$f^* = \arg\min_f L\left(f\right)$$

2.2.3 Các giải thuật hồi quy

- Hồi quy tuyến tính ?

$$f : \mathbb{R}^M \to \mathbb{R}$$

Số khả năng của hàm *f(.)* ?

Trong ML: thường chọn một dạng hàm cụ thể, dễ thao tác, hữu ích

$$f(\mathbf{x}) = \theta_0 + \theta_1 x_1$$

θ_i : Hệ số/trọng số (weight)

$$f(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

$$f(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

hyperplane

Hồi quy tuyến tính

- Quan hệ giữa đầu vào và đầu ra được mô tả bởi 1 hàm tuyến tính

$$\hat{y} = f(\mathbf{x}) = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_M x_M$$

$\hat{y}$: giá trị dự đoán

M : số lượng đặc trưng

x_i : giá trị của đặc trưng thứ *i*

θ_i : hệ số thứ *i* của mô hình

Độ lệch (bias)

$$\hat{y} = f(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_M x_M$$

Hồi quy tuyến tính

- Linear regression

$$\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_M]^T$$

$$\Longrightarrow \boxed{\hat{y} = f(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}}$$

Xét nhiều điểm/mẫu dữ liệu $(\mathbf{x}_i, y_i)\ i = 1, 2, \ldots, N$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - \boldsymbol{\theta}^T \mathbf{x}_i)^2$$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} MSE(\boldsymbol{\theta}) \Longrightarrow \boxed{\boldsymbol{\theta}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}}$$

$$\mathbf{X}_{N \times M} \qquad \mathbf{y} = [y_1, y_2, \ldots, y_N]^T$$

Hồi quy tuyến tính

- Chú ý:

normal equation

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} MSE(\boldsymbol{\theta}) \Longrightarrow \boldsymbol{\theta}^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

$\mathbf{X}^T\mathbf{X}$ khả nghịch

- giải phương trình đạo hàm theo hệ số $\dfrac{\nabla MSE(\boldsymbol{\theta})}{\boldsymbol{\theta}} = 0$

$$X_{N\times M} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \cdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}$$

- Tính toán ma trận nghịch đảo $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ khi *N* lớn phức tạp

Hồi quy tuyến tính

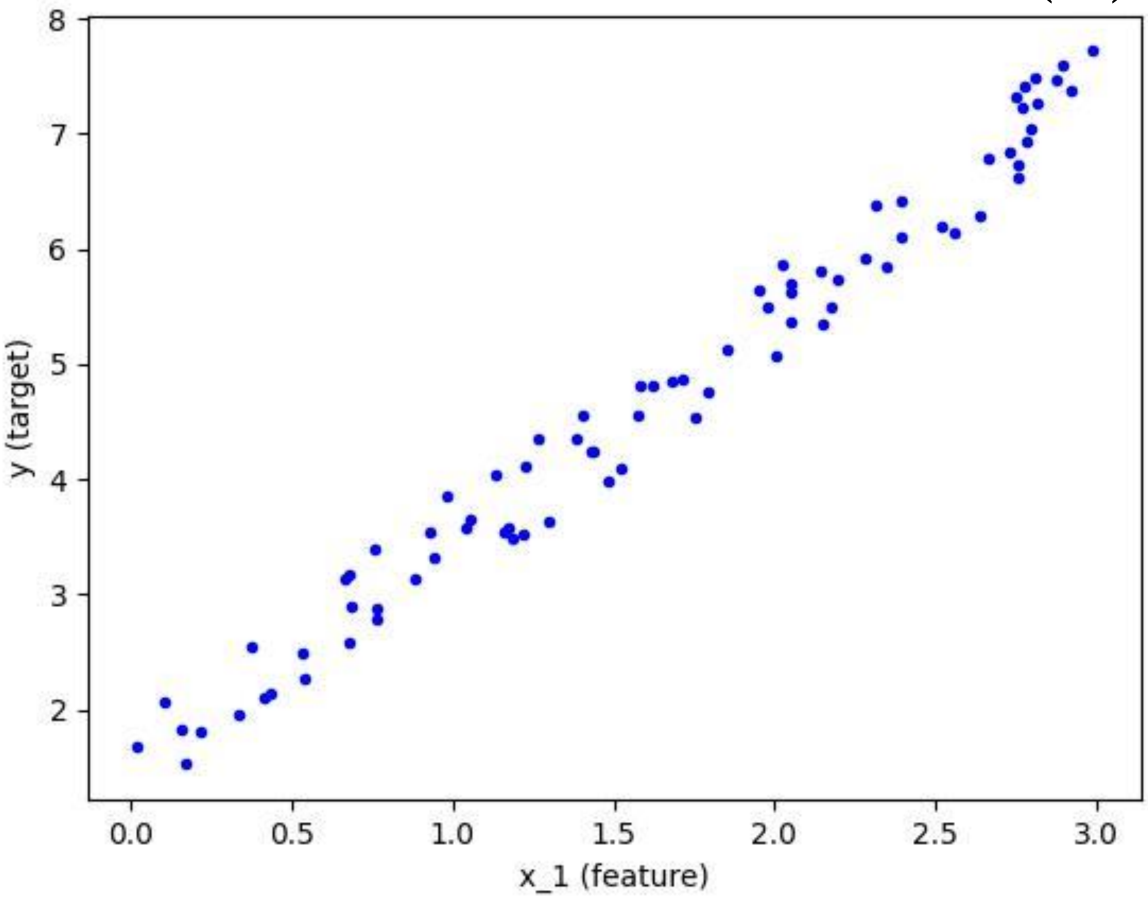
$$\boldsymbol{\theta}^{*} = \left(\mathbf{X}^{T}\mathbf{X}\right)^{-1}\mathbf{X}^{T}\mathbf{y}$$

- Tính nghiệm cần tìm ~ thu được các hệ số mô hình (trọng số đặc trưng)
- Có thể thực hiện bằng cách sử dụng hàm có sẵn trong Python, Octave

Ví dụ 1

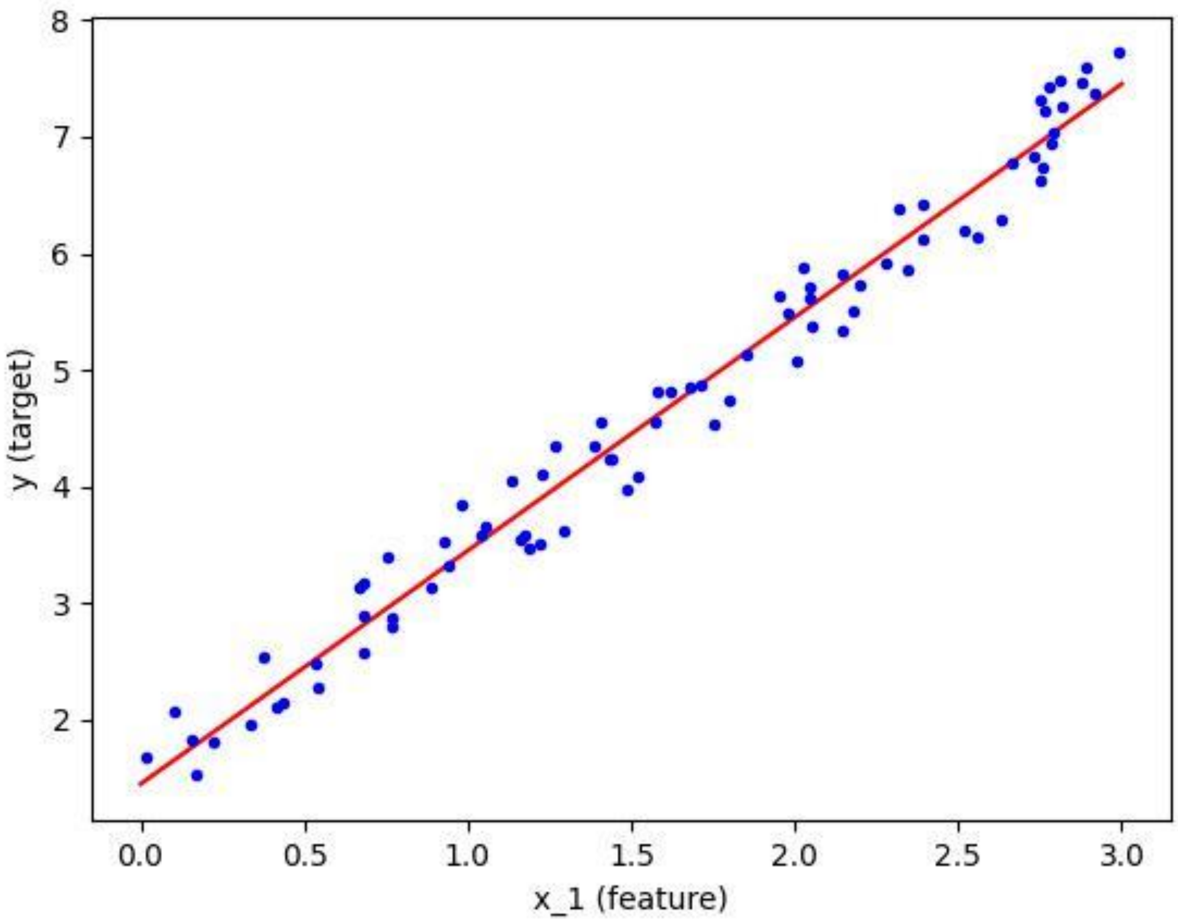
- Sử dụng Python, dùng normal equation

$$f(\mathbf{x}) = \theta_0 + \theta_1 x_1$$



Ví dụ 1

$$f(x_1) = 1.45046565 + 2.0005925x_1$$



Ví dụ 2

- Dùng Scikit-Learn

```
LinearRegression().fit(X_train, y_train)
```

Ví dụ 3

- Xét lại một ví dụ cũ

Mẫu	Khối lượng (kg)	Chiều dài (m)
1	30	70
2	40	90
3	40	100
4	50	120
5	50	130
6	50	150
7	60	160
8	70	190
9	70	200
10	80	200
11	80	220
12	80	230

$M = 1 \quad N = 12$

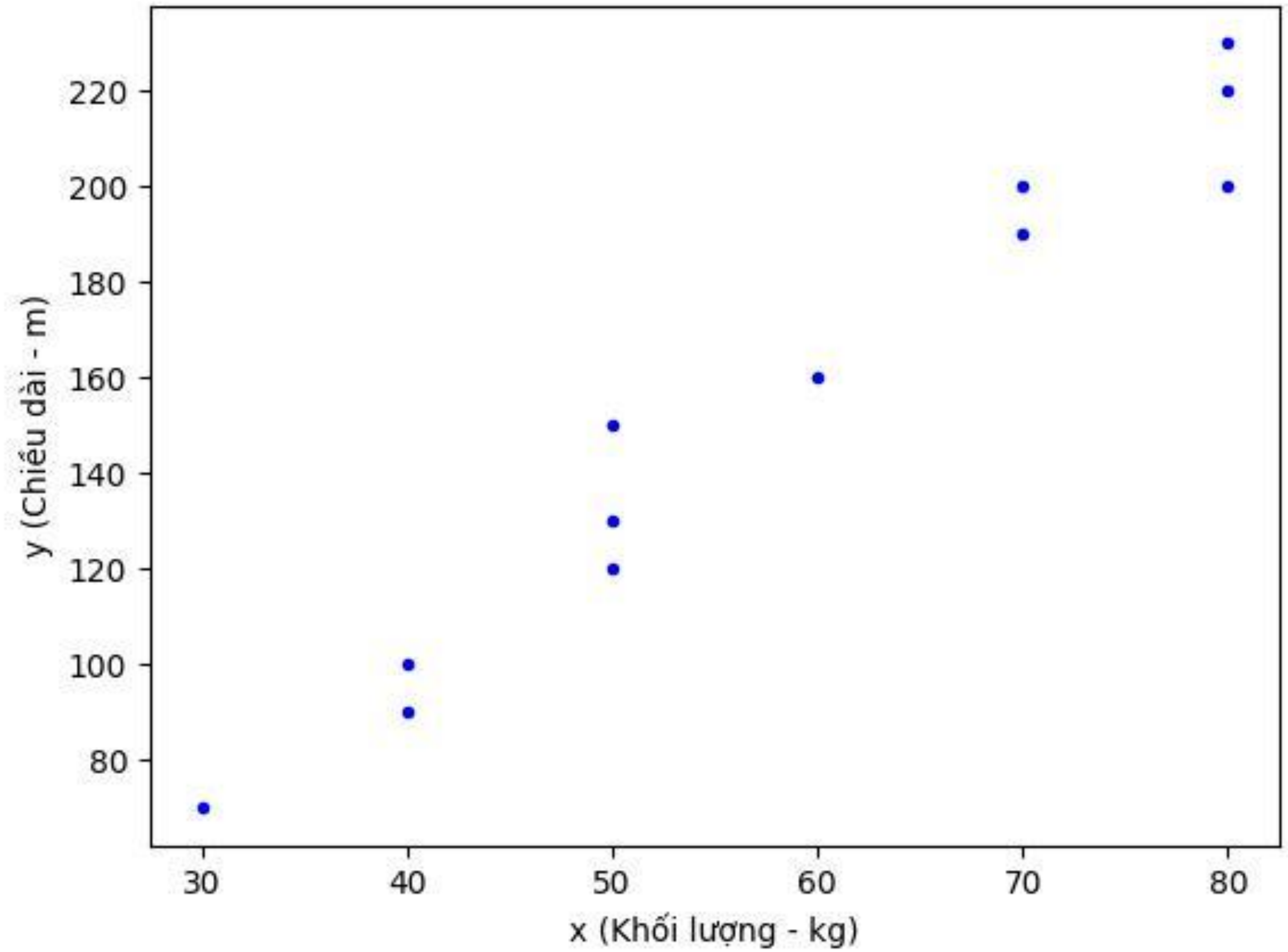
Dữ liệu mới

$$x = 35$$

Dự đoán ?

Ví dụ 3

$$\hat{y} = f(x) = \theta_0 + \theta_1 x$$



Ví dụ 3

- Tìm 2 hệ số

$$L(\boldsymbol{\theta}) = \frac{1}{N}\sum\left(y-\hat{y}\right)^2 = \frac{1}{N}\sum\left(y-\theta_0-\theta_1 x\right)^2$$

$$\frac{\nabla L(\boldsymbol{\theta})}{\boldsymbol{\theta}} = 0 \Longrightarrow \begin{cases} \sum y - N\theta_0 - \theta_1 \sum x = 0 \\ \sum xy - \theta_0 \sum x - \theta_1 \sum x^2 = 0 \end{cases}$$

$$\boxed{\theta_1 = \frac{\dfrac{\sum xy}{N} - \dfrac{\sum x}{N}\dfrac{\sum y}{N}}{\dfrac{\sum x^2}{N} - \left(\dfrac{\sum x}{N}\right)^2}}$$

$$\boxed{\theta_0 = \frac{\sum y}{N} - \theta_1 \frac{\sum x}{N}}$$

Ví dụ 3

- Thay số và tính toán

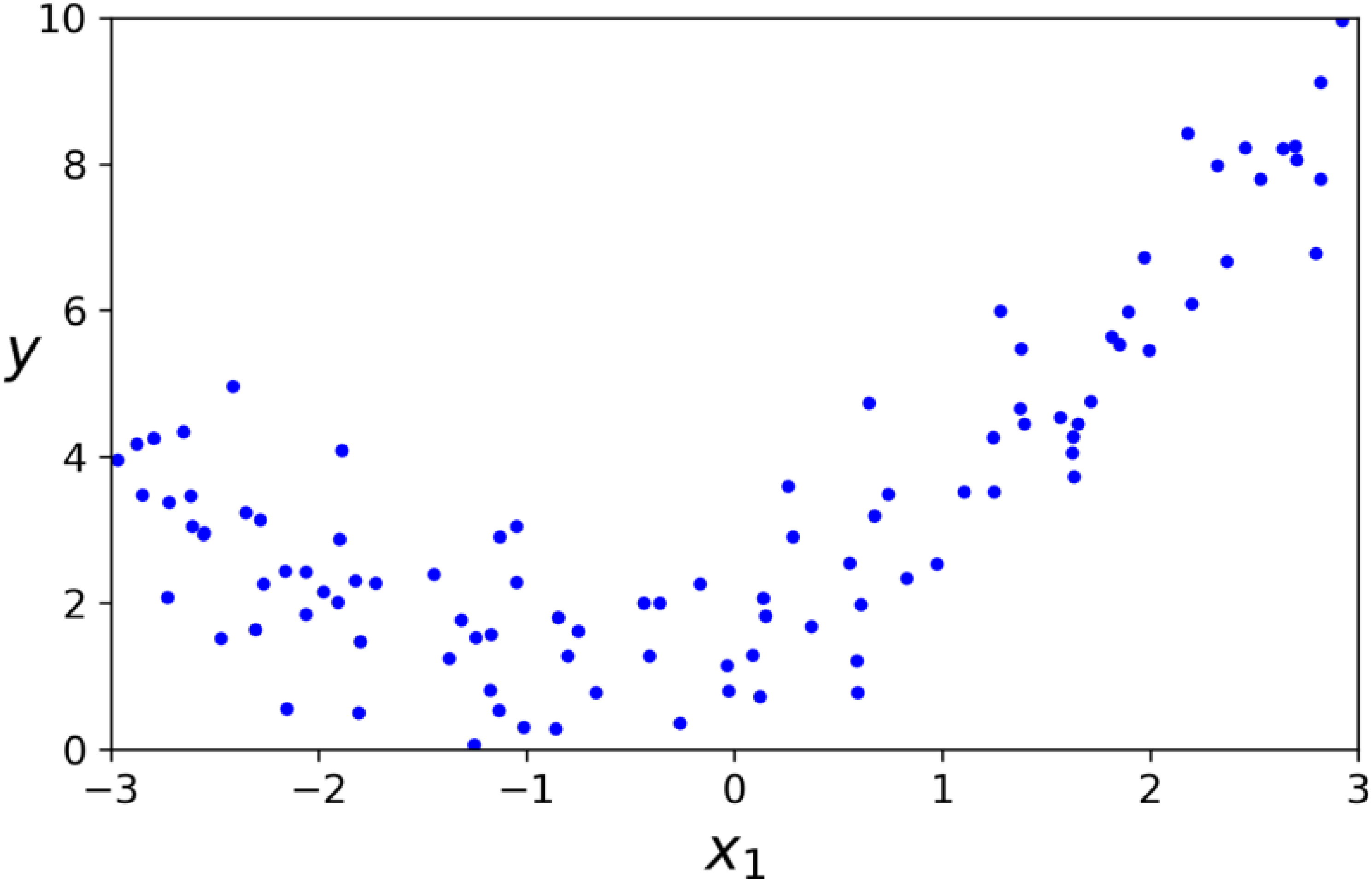
$$y = -20 + 3x$$

Hồi quy tuyến tính

- Biểu diễn cụ thể, đơn giản
  - => dễ dàng diễn giải, đánh giá

Polynomial Regression (PR)

- Khi dữ liệu phức tạp hơn?

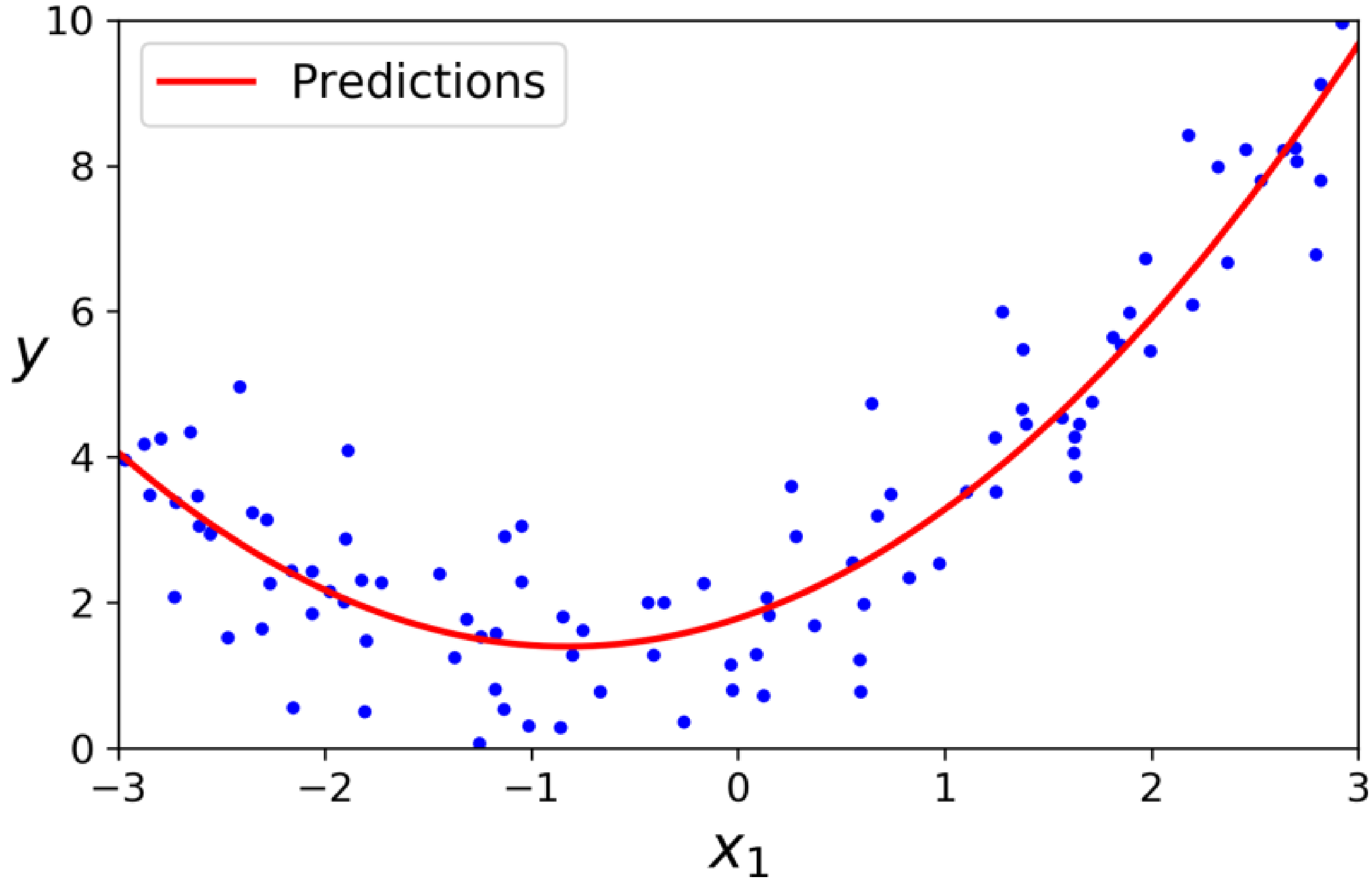


Polynomial Regression (PR)

- Có thể sử dụng hồi quy tuyến tính cho tập dữ liệu phi tuyến

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$$

PolynomialFeature (Scikit-Learn)



Polynomial Regression (PR)

- Hồi quy đa thức
  - Mô hình phức tạp hơn
  - Phù hợp với các tập dữ liệu phi tuyến
  - Mô hình PR có nhiều tham số hơn mô hình LR
    - Có xu hướng xảy ra overfitting (quá khớp/phù hợp) với dữ liệu huấn luyện
      - Sử dụng learning curve để kiểm tra xem có xảy ra overfitting không
      - Áp dụng một số kỹ thuật regularization để giảm nguy cơ xảy ra overfitting

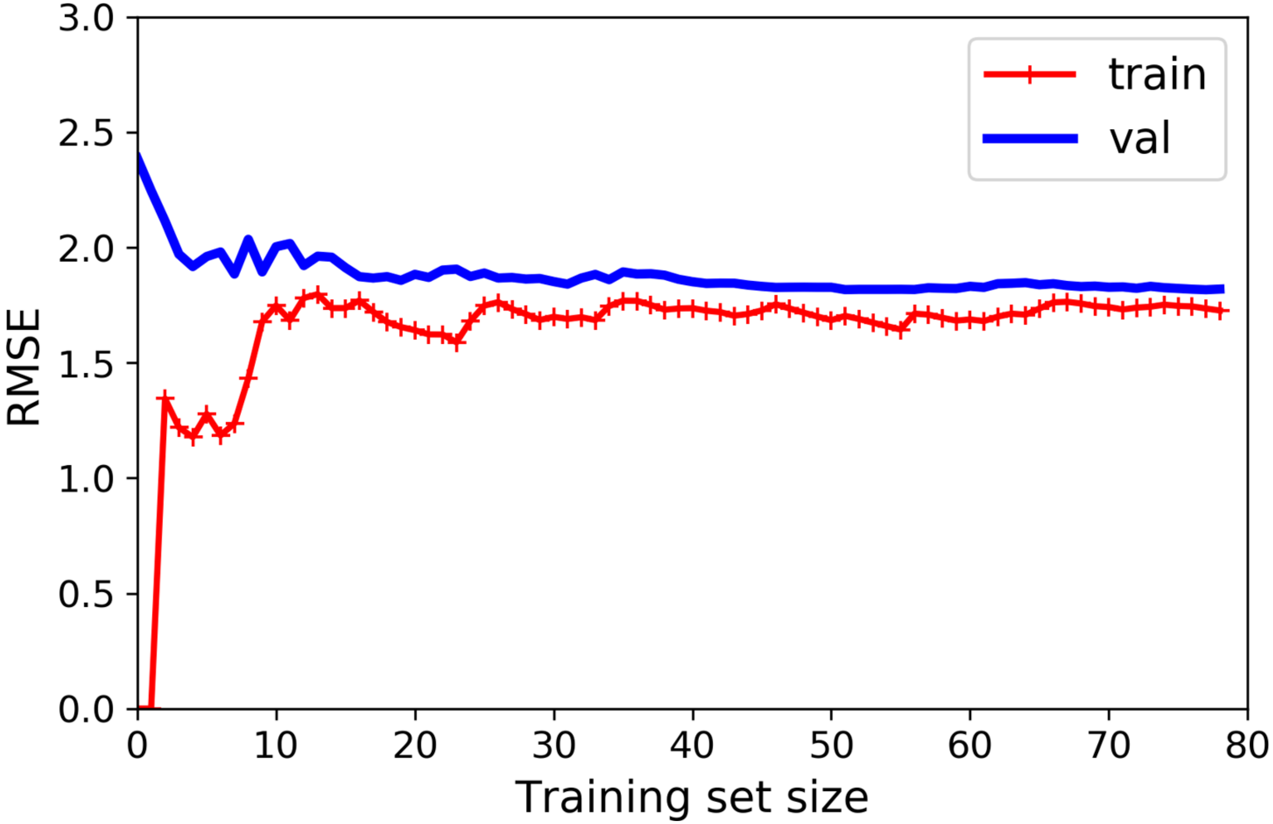
Learning curve

- Các đường biểu diễn quan hệ giữa
  - Hiệu quả của mô hình trên tập huấn luyện và tập xác thực
  - Với kích cỡ tập dữ liệu và tập xác thực (hoặc theo số bước lặp huấn luyện)

Learning curve

- Ví dụ:
  - Quan sát các đường học tập và đánh giá mô hình sử dụng

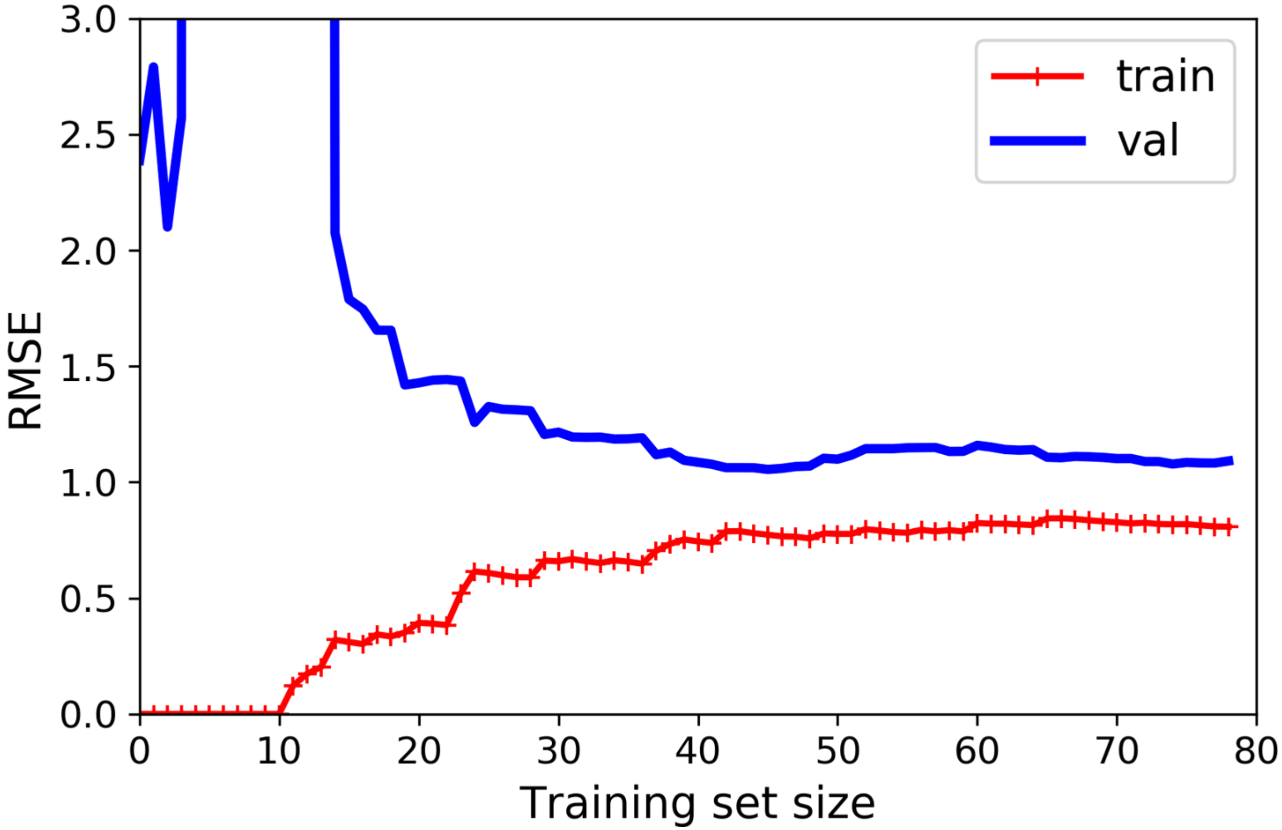
Sử dụng mô hình LR



Learning curve

- Ví dụ:
  - Quan sát các đường học tập và đánh giá mô hình sử dụng

Sử dụng mô hình PR với bậc 10



Bias/variance trade-off

generalization error = bias + variance + irreducible error

- bias: do những giả thuyết sai, ví dụ: mô hình với bias cao => gây ra underfitting

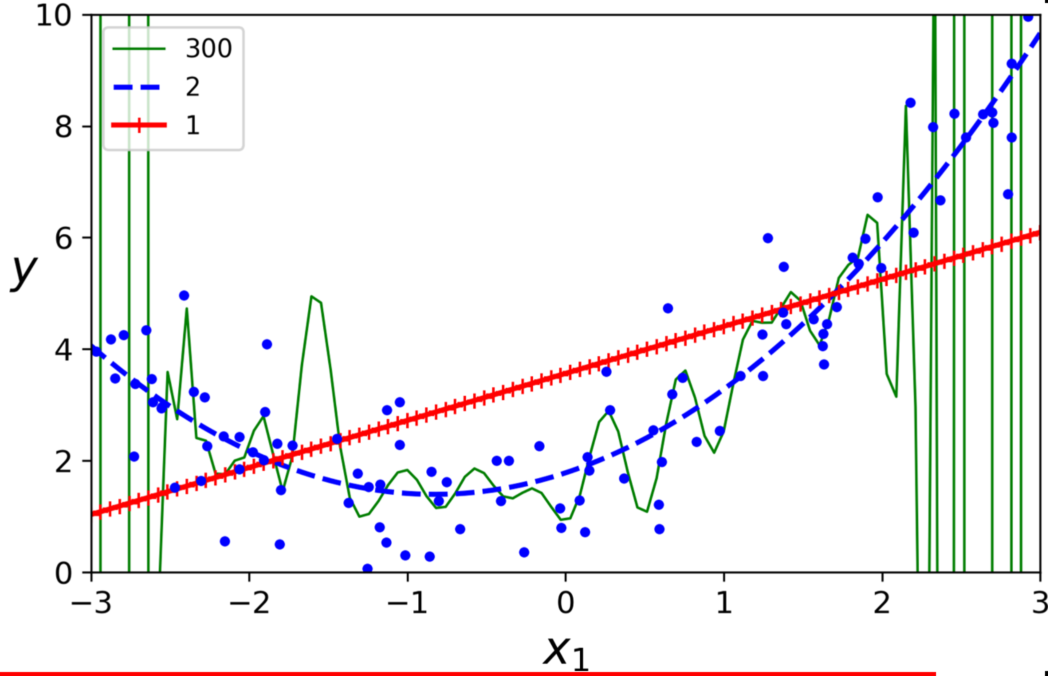
- variance: quá nhạy với những thay đổi nhỏ trong dữ liệu huấn luyện, ví dụ: mô hình với nhiều bậc tự do thường có variance cao => gây ra ovetfitting

Độ phức tạp của mô hình	Variance	Bias
Tăng	Tăng	Giảm
Giảm	Giảm	Tăng

trade-off

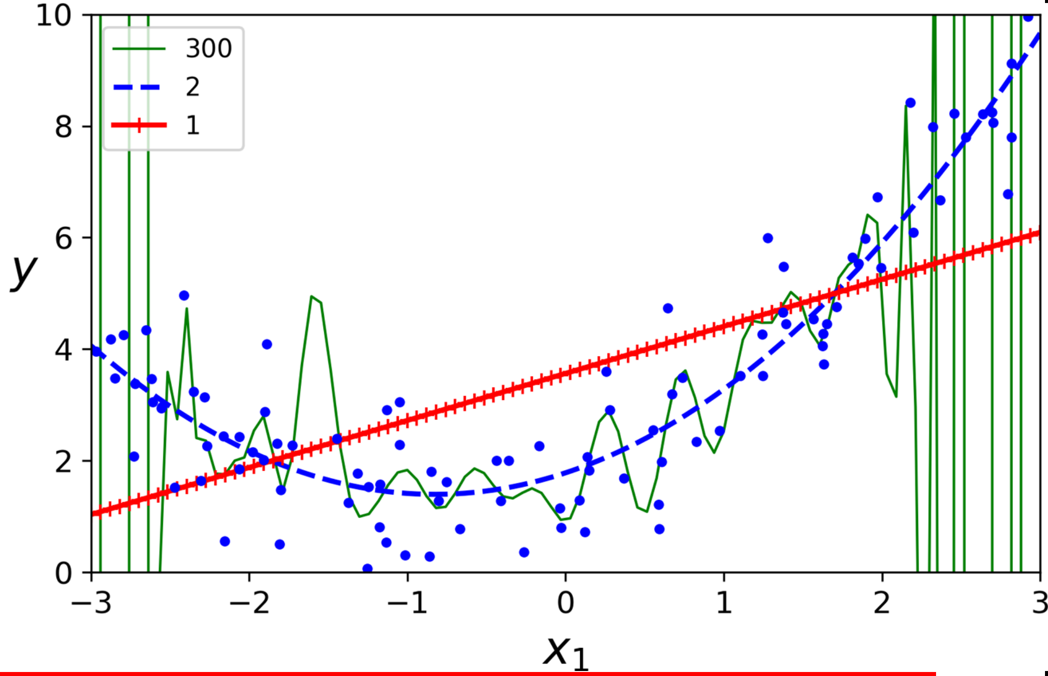
Bias/variance trade-off

- Ví dụ
  - Dùng mô hình PR có bậc cao cho bộ dữ liệu phi tuyến
    - Mức độ phức tạp của mô hình: tăng
    - Variance: tăng
    - Bias: giảm
    - Dẫn tới: overfitting



Bias/variance trade-off

- Ví dụ
  - Dùng mô hình LR cho bộ dữ liệu phi tuyến
    - Mức độ phức tạp của mô hình: giảm
    - Variance: giảm
    - Bias: tăng
    - Dẫn tới: underfitting



Điều chuẩn mô hình

- Regularization: kỹ thuật làm giảm overfitting
  - Mô hình càng ít bậc tự do thì sẽ càng khó để làm phù hợp với dữ liệu huấn luyện
    - Ví dụ: giảm số bậc đa thức
  - Đối với mô hình tuyến tính: ràng buộc các trọng số của mô hình
    - Ví dụ: ridge regression, lasso regression, elastic net

Ridge regularization

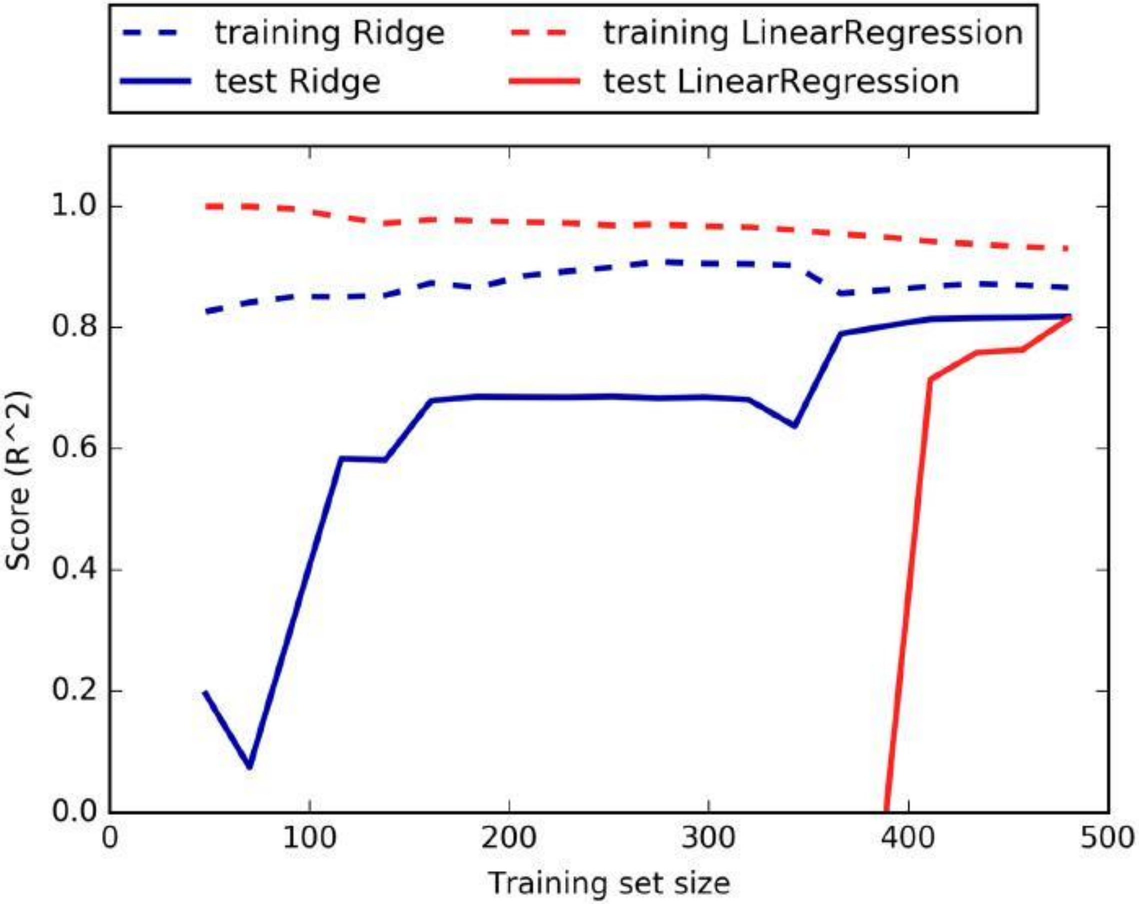
- Các hệ số mô hình được chọn không chỉ dự đoán tốt trên dữ liệu huấn luyện mà còn phù hợp với một ràng buộc bổ sung
- Độ lớn của các hệ số càng nhỏ càng tốt ~ tất cả các hệ số phải gần = 0
  - Mỗi đặc trưng sẽ có ảnh hưởng ít nhất đến kết quả đầu ra

$$J(\boldsymbol{\theta}) = MSE(\boldsymbol{\theta}) + \frac{\alpha}{2}\sum_{i=1}^{N}\theta_i^2$$

$$\boldsymbol{\theta}^* = \left(\mathbf{X}^T\mathbf{X} + \alpha\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

Ridge regularization

- Tạo ra sự đánh đổi giữa tính đơn giản của mô hình (các hệ số gần = 0) và hiệu quả của mô hình trên tập huấn luyện



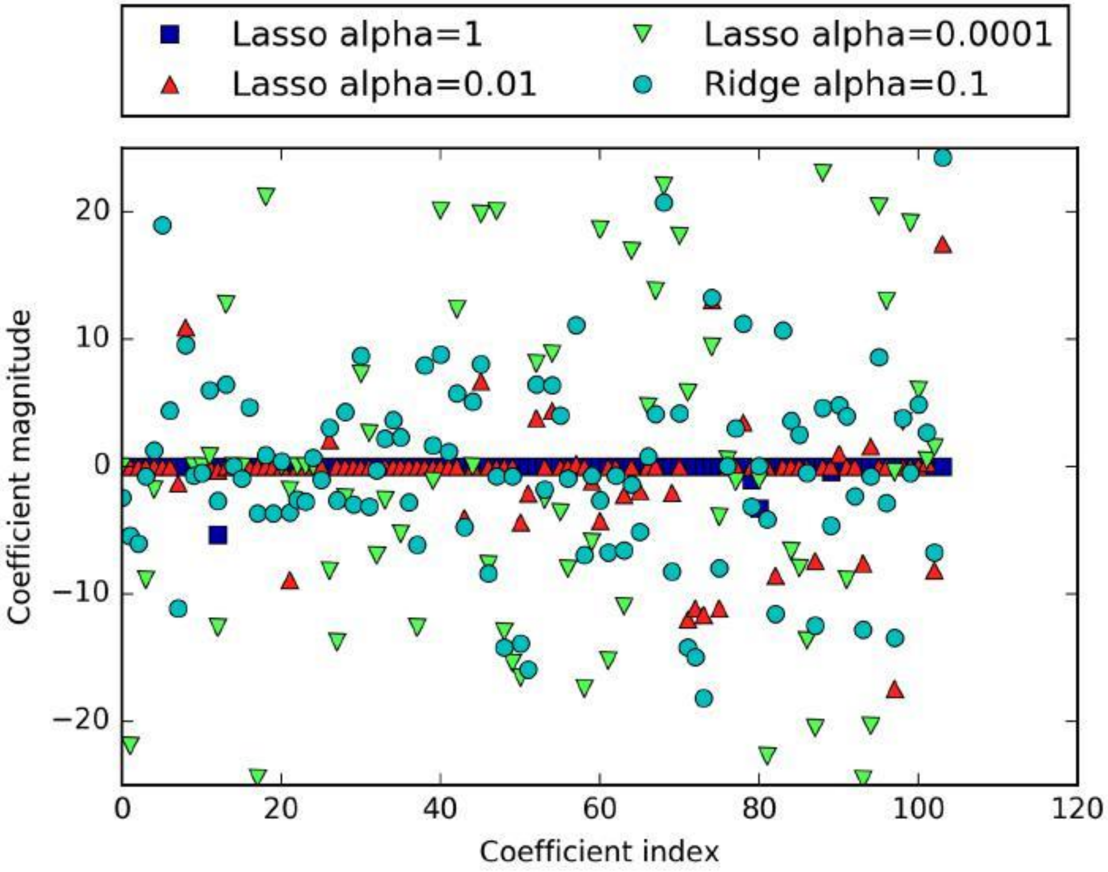
Lasso regression

- Thêm một ràng buộc vào hàm mất mát
- Loại bỏ các trọng số của các đặc trưng ít quan trọng nhất bằng các gán chúng = 0
- Tự động thực hiện việc lựa chọn đặc trưng và đưa đến 1 mô hình thưa, sparse model (chỉ chứa 1 số ít các trọng số đặc trưng ≠ 0)

$$J(\boldsymbol{\theta}) = MSE(\boldsymbol{\theta}) + \alpha \sum_{i=1}^{N} |\theta_i|$$

Lasso regression

- Có một số hệ số bằng 0 thường làm cho mô hình dễ diễn giải hơn
- Cho biết các đặc trưng quan trọng nhất của mô hình



Elastic net

- Kết hợp giữa RR và LR

$$J(\boldsymbol{\theta}) = MSE(\boldsymbol{\theta}) + \frac{1-r}{2}\alpha\sum_{i=1}^{N}\theta_i^2 + r\alpha\sum_{i=1}^{N}|\theta_i|$$

Ridge Regression

Lasso Regression

$r = 0$

$r = 1$

Elastic Net

Nhận xét

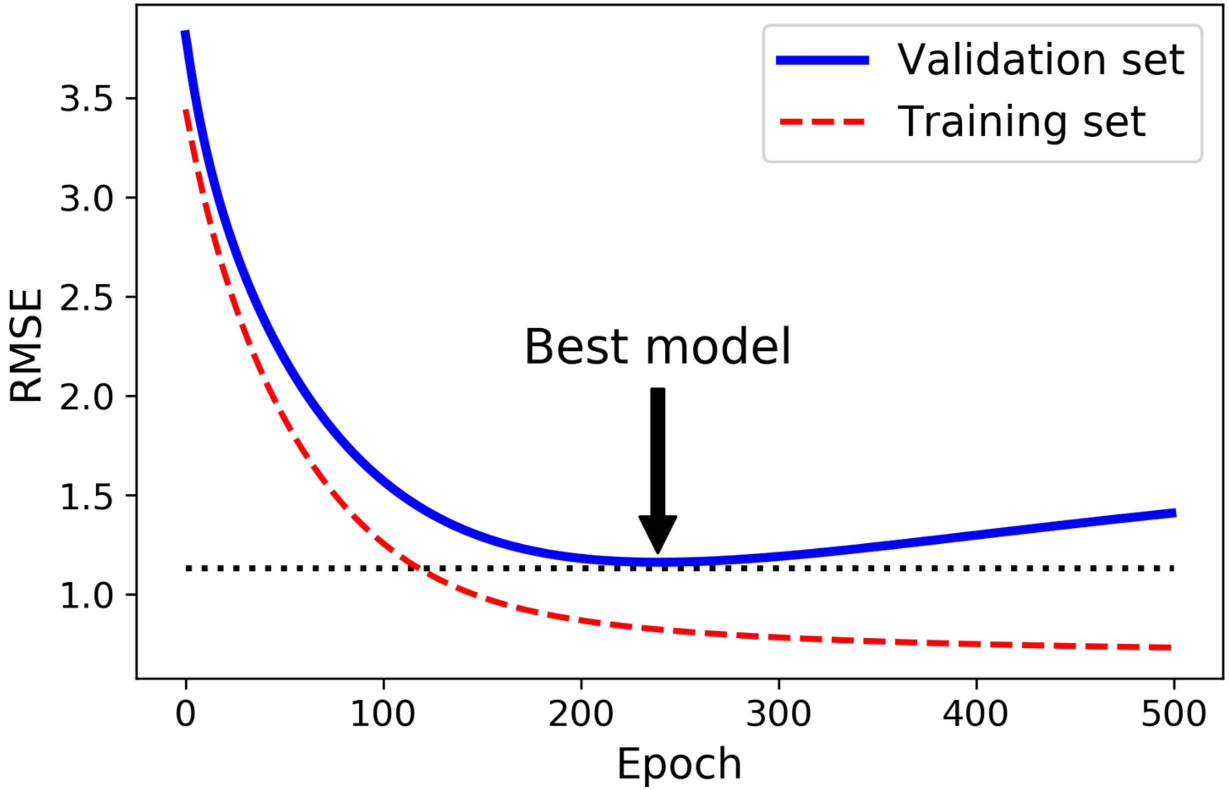
Ridge Regression
- Lựa chọn mặc định

Lasso Regression
- Có nhiều đặc trưng nhưng mong đợi chỉ một vài đặc trưng là quan trọng
- Muốn có một mô hình dễ diễn giải

Elastic Net
- Được ưa chuộng hơn

Early stopping

- Dừng việc huấn luyện ngay khi lỗi xác thực đạt tới một mức ngưỡng nhỏ nhất
  - Khi epoch tăng lên đến lúc lỗi xác thực dừng giảm và bắt đầu tăng lên => chỉ ra overfitting
- Một cách khác để điều chuẩn các thuật toán như GD



Tổng kết

- Hiểu và phân loại được: xấp xỉ hàm, phân lớp
- Ý nghĩa của hàm mất mát trong việc tối ưu
- Vận dụng được hồi quy tuyến tính, chú ý tới regularization

Hoạt động sau buổi học

- Làm BTVN

Chuẩn bị cho buổi học tiếp theo

- Tìm hiểu về Gradient Descent
- Tìm hiểu về Logistic Regression

Tài liệu tham khảo

- https://phamdinhkhanh.github.io/deepai-book/ch_ml/prediction.html
- https://blog.econocom.com/en/blog/iiot-how-cable-drums-are-becoming-smart/