THUẬT TOÁN VÀ LÝ THUYẾT MÁY HỌC

CÁC BÀI TOÁN HỒI QUY TUYẾN TÍNH

TS. Lương Ngọc Hoàng

Nội dung

1. Hồi quy tuyến tính đơn giản
2. Hồi quy tuyến tính đa biến
3. Hồi quy đa thức
4. Quá khớp (overfitting) và các kỹ thuật điều chuẩn (regularization)

HỒI QUY TUYẾN TÍNH ĐƠN GIẢN

SIMPLE LINEAR REGRESSION

Hồi quy tuyến tính đơn giản – Mô hình (model)

Một mô hình (model) hồi quy tuyến tính đơn giản (simple linear regression) thực hiện dự đoán đầu ra (output) với một hàm tuyến tính (linear function) của một đặc trưng đầu vào (input feature) x:

$$f(x; w_0, w_1) = w_0 + w_1 x$$

w_0, w_1: các tham số (parameters) của mô hình.

Để tìm ra giá trị phù hợp của w_0 và w_1, ta sử dụng một tập dữ liệu (dataset) gồm các mẫu dữ liệu với cặp thông tin đầu vào (input) – đầu ra (output) tương ứng:

$$\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), ..., \left(x^{(N)}, y^{(N)}\right)\right\}$$

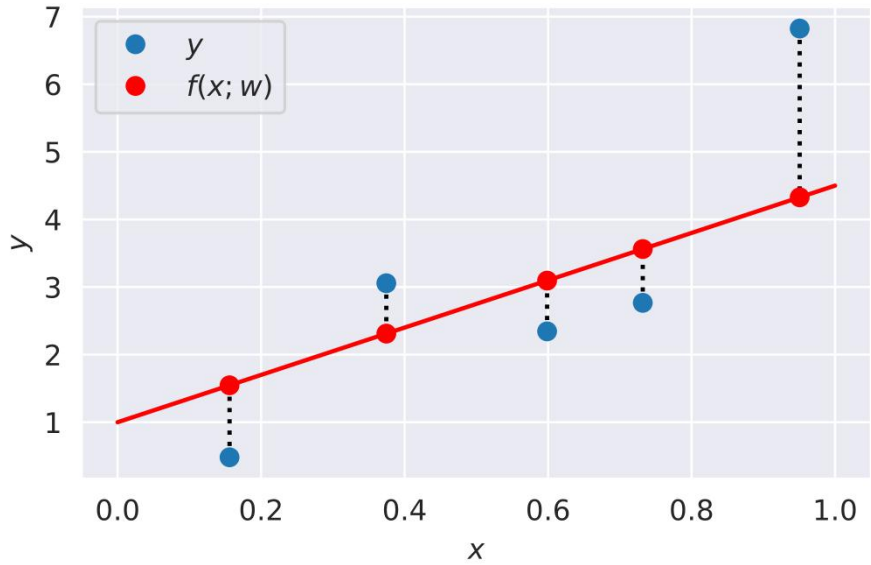
Tập dữ liệu cũng có thể được biểu diễn như sau: $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$.

Ta tìm giá trị phù hợp cho w_0 và w_1 dựa vào dữ liệu như thế nào? Ta cần một độ đo để đánh giá tính phù hợp của các giá trị tham số với dữ liệu.

Hồi quy tuyến tính đơn giản – Hàm mất mát

Hàm mất mát (loss function), hay cũng có thể gọi là hàm chi phí (cost function), phản ánh sự phù hợp của các giá trị tham số w_0, w_1 đối với tập dữ liệu:

$$L(w_0, w_1) = \sum_{i=1}^{N} \left(y^{(i)} - f(x^{(i)}; w_0, w_1)\right)^2$$



Hàm mất mát trên được gọi là hàm mất mát bình phương (squared loss) hoặc tổng bình phương phần dư (Residual Sum of Squares – RSS).

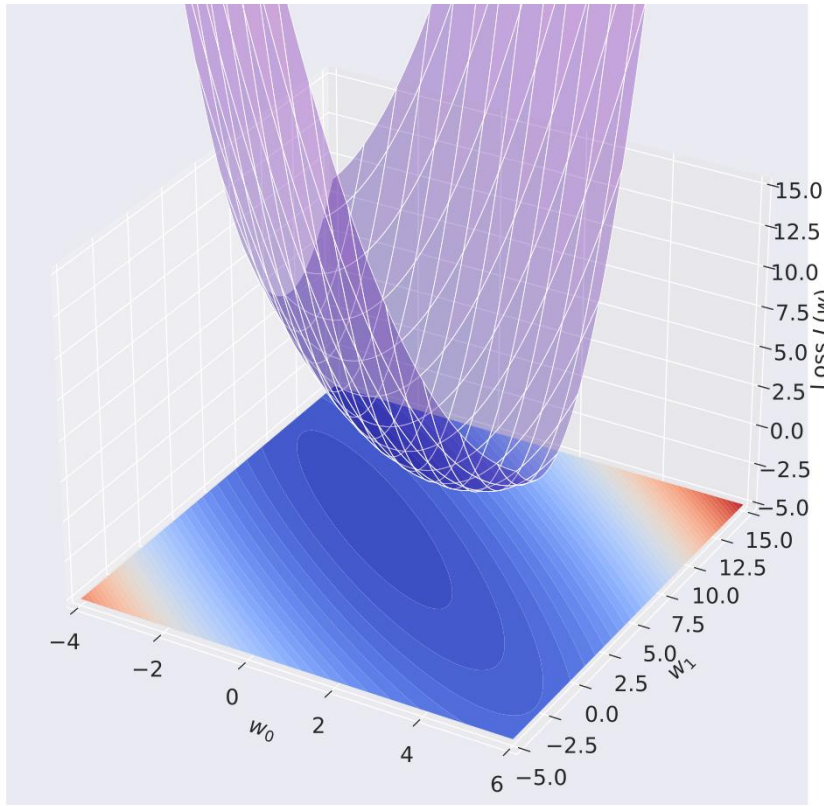
Ta có thể định nghĩa hàm mất mát sai số bình phương trung bình (Mean Squared Error – MSE) như sau:

$$L(w_0, w_1) = \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - f(x^{(i)}; w_0, w_1)\right)^2$$

Hồi quy tuyến tính đơn giản – Tối ưu hóa

Giá trị tham số w_0 và w_1 phù hợp nhất đối với tập dữ liệu giúp mô hình đạt giá trị hàm mất mát nhỏ nhất trên tập dữ liệu:

$$\hat{w}_0, \hat{w}_1 = \arg\min_{w_0, w_1} L(w_0, w_1)$$



Để giải quyết bài toán tối ưu hoá (optimization) trên, ta có thể tính đạo hàm của hàm mất mát theo từng tham số $\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}$, và tìm nghiệm của các phương trình $\frac{\partial L}{\partial w_0} = 0$ và $\frac{\partial L}{\partial w_1} = 0$.

$$L(w_0, w_1) = \sum_{i=1}^{N} \left(y^{(i)} - f(x^{(i)}; w_0, w_1)\right)^2$$

$$= \sum_{i=1}^{N} \left(y^{(i)} - \left(w_0 + w_1 x^{(i)}\right)\right)^2$$

Hồi quy tuyến tính đơn giản – Tối ưu hóa

$$L(w_0, w_1) = \sum_{i=1}^{N} \left(y^{(i)} - \left(w_0 + w_1 x^{(i)}\right)\right)^2$$

$$\frac{\partial L}{\partial w_0} = \frac{\partial}{\partial w_0} \sum_{i=1}^{N} \left(y^{(i)} - \left(w_0 + w_1 x^{(i)}\right)\right)^2$$

$$= \sum_{i=1}^{N} 2\left(y^{(i)} - \left(w_0 + w_1 x^{(i)}\right)\right)(-1)$$

Giải $\frac{\partial L}{\partial w_0} = 0 = \sum_{i=1}^{N} 2\left(y^{(i)} - \left(w_0 + w_1 x^{(i)}\right)\right)(-1)$

$$\sum_{i=1}^{N} w_0 = \sum_{i=1}^{N} y^{(i)} - w_1 \sum_{i=1}^{N} x^{(i)}$$

$$N w_0 = \sum_{i=1}^{N} y^{(i)} - w_1 \sum_{i=1}^{N} x^{(i)}$$

$$w_0 = \frac{1}{N} \sum_{i=1}^{N} y^{(i)} - w_1 \frac{1}{N} \sum_{i=1}^{N} x^{(i)} = \bar{y} - w_1 \bar{x}$$

$\bar{y}$: trung bình giá trị đích trong tập dữ liệu.
$\bar{x}$: trung bình giá trị đặc trưng trong tập dữ liệu.

$$\frac{\partial L}{\partial w_1} = \frac{\partial}{\partial w_1} \sum_{i=1}^{N} \left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2$$

$$= \frac{\partial}{\partial w_1} \sum_{i=1}^{N} \left(y^{(i)} - \bar{y} + w_1 \bar{x} - w_1 x^{(i)}\right)^2$$

$$= \frac{\partial}{\partial w_1} \sum_{i=1}^{N} \left(y^{(i)} - \bar{y} - w_1\left(x^{(i)} - \bar{x}\right)\right)^2$$

$$= \sum_{i=1}^{N} 2\left(y^{(i)} - \bar{y} - w_1\left(x^{(i)} - \bar{x}\right)\right)(-1)\left(x^{(i)} - \bar{x}\right)$$

Đặt $\frac{\partial L}{\partial w_1} = 0 = \sum_{i=1}^{N} \left(y^{(i)} - \bar{y} - w_1\left(x^{(i)} - \bar{x}\right)\right)\left(x^{(i)} - \bar{x}\right)$

$$\sum_{i=1}^{N} w_1 \left(x^{(i)} - \bar{x}\right)^2 = \sum_{i=1}^{N} \left(y^{(i)} - \bar{y}\right)\left(x^{(i)} - \bar{x}\right)$$

$$w_1 = \frac{\sum_{i=1}^{N} \left(y^{(i)} - \bar{y}\right)\left(x^{(i)} - \bar{x}\right)}{\sum_{i=1}^{N} \left(x^{(i)} - \bar{x}\right)^2}$$

Giá trị w_0 và w_1 được gọi là ước lượng bình phương tối thiểu (least square estimates)

HỒI QUY TUYẾN TÍNH ĐA BIẾN

MULTIPLE LINEAR REGRESSION

Hồi quy tuyến tính đa biến – Mô hình (model)

Một mô hình (model) hồi quy tuyến tính đa biến (multiple linear regression) thực hiện dự đoán đầu ra (output) với một hàm tuyến tính (linear function) của các đặc trưng đầu vào (input features) $x_1, x_2, \cdots, x_D$:

$$f(x_1, x_2, ..., x_D; w_0, w_1, w_2, ..., w_D) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D$$

$w_0, w_1, w_2, ..., w_D$: các tham số (parameters) của mô hình.

w_0: tham số bias.

$w_1, w_2, ..., w_D$: trọng số (weights) của mỗi đặc trưng.

Ta có thể biểu diễn

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} \quad \text{và} \quad \boldsymbol{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix}$$

. Do đó, ta có mô hình $f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{x}$.

Hồi quy tuyến tính đa biến – Hàm mất mát

Hàm mất mát bình phương (squared loss) của mô hình hồi quy tuyến tính đa biến:

$$L(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y^{(i)} - f(\boldsymbol{x}^{(i)}; \boldsymbol{w})\right)^2 = \sum_{i=1}^{N} \left(y^{(i)} - (w_0 + w_1 x_1 + \ldots + w_D x_D)\right)^2 = \sum_{i=1}^{N} \left(y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(i)}\right)^2$$

Ta có thể định nghĩa:

$$\boldsymbol{X} = \begin{bmatrix} -- & \boldsymbol{x}^{(1)^T} & -- \\ -- & \boldsymbol{x}^{(2)^T} & -- \\ & \vdots & \\ -- & \boldsymbol{x}^{(N)^T} & -- \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}. \text{ Vector sai số là: } \boldsymbol{e} = \begin{bmatrix} y^{(1)} - \boldsymbol{x}^{(1)^T}\boldsymbol{w} \\ y^{(2)} - \boldsymbol{x}^{(2)^T}\boldsymbol{w} \\ \vdots \\ y^{(N)} - \boldsymbol{x}^{(N)^T}\boldsymbol{w} \end{bmatrix} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}.$$

Hàm mất mát có thể được biểu diễn như sau:

$$L(\boldsymbol{w}) = \boldsymbol{e}^T\boldsymbol{e} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$$= \boldsymbol{y}^T\boldsymbol{y} - \underbrace{\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w}}_{1\times 1} - \underbrace{\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y}}_{1\times 1} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}$$

Hồi quy tuyến tính đa biến – Tối ưu hóa

Hàm mất mát:

$$L(\boldsymbol{w}) = \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}$$

Ta tính gradient:

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{w}} &= \frac{\partial}{\partial \boldsymbol{w}}[\boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}] \\ &= -2\boldsymbol{X}^T\boldsymbol{y} + (\boldsymbol{X}^T\boldsymbol{X} + (\boldsymbol{X}^T\boldsymbol{X})^T)\boldsymbol{w} \\ &= -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}\end{aligned}$$

Ta sử dụng các công thức sau:

$$\frac{\partial \boldsymbol{x}^T\boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$$

$$\frac{\partial \boldsymbol{x}^T A\boldsymbol{x}}{\partial \boldsymbol{x}} = (A + A^T)\boldsymbol{x}$$

Ta đặt $\frac{\partial L}{\partial \boldsymbol{w}} = -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{0}$ và giải:

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Công thức trên có tên gọi là Normal Equations, trả về ước lượng bình phương tối thiểu.

Hồi quy tuyến tính – Hình học

Chiếu trực giao vector $\boldsymbol{b}$ lên đường thẳng đi qua vector $\boldsymbol{a}$.

Bởi vì $\boldsymbol{e}$ trực giao với $\boldsymbol{a}$, nên ta có:

$$\boldsymbol{a}^T\boldsymbol{e} = 0$$

$$\boldsymbol{a}^T(\boldsymbol{b} - \boldsymbol{p}) = 0$$

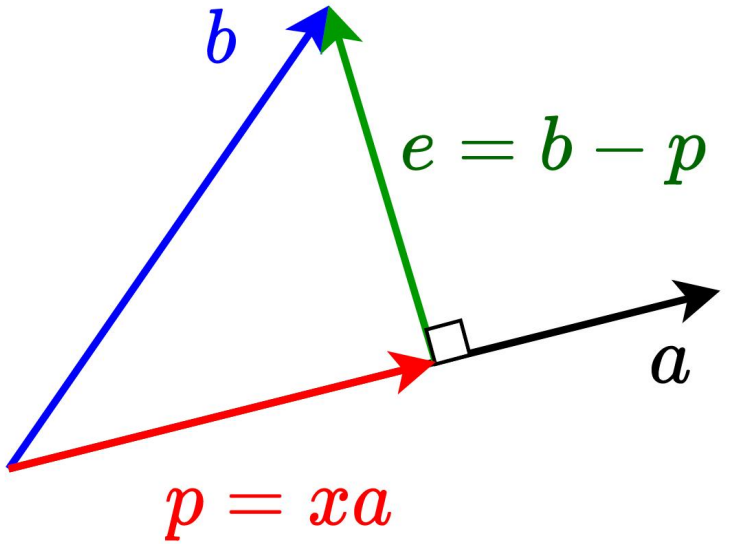
$$\boldsymbol{a}^T(\boldsymbol{b} - x\boldsymbol{a}) = 0$$

$$x\boldsymbol{a}^T\boldsymbol{a} = \boldsymbol{a}^T\boldsymbol{b}$$

$$x = \frac{\boldsymbol{a}^T\boldsymbol{b}}{\boldsymbol{a}^T\boldsymbol{a}}$$

Do đó, hình chiếu $\boldsymbol{p} = x\boldsymbol{a} = \boldsymbol{a}x = \boldsymbol{a}\frac{\boldsymbol{a}^T\boldsymbol{b}}{\boldsymbol{a}^T\boldsymbol{a}}$.

Ta có ma trận chiếu (projection matrix) P sao cho $p = P\boldsymbol{b}$. $\boxed{P = \frac{\boldsymbol{a}\boldsymbol{a}^T}{\boldsymbol{a}^T\boldsymbol{a}}}$



- Hình chiếu $\boldsymbol{p}$ của $\boldsymbol{b}$ nằm trên đường thẳng đi qua $\boldsymbol{a}$. Do đó: $\boldsymbol{p} = x\boldsymbol{a}$.
- Vector $\boldsymbol{e}$ thể hiện sự khác biệt giữa $\boldsymbol{b}$ và hình chiếu $\boldsymbol{p}$. Do đó: $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{p}$.

Hồi quy tuyến tính – Hình học

Chiếu trực giao vector $\boldsymbol{b}$ lên mặt phẳng chứa 2 vector $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$.

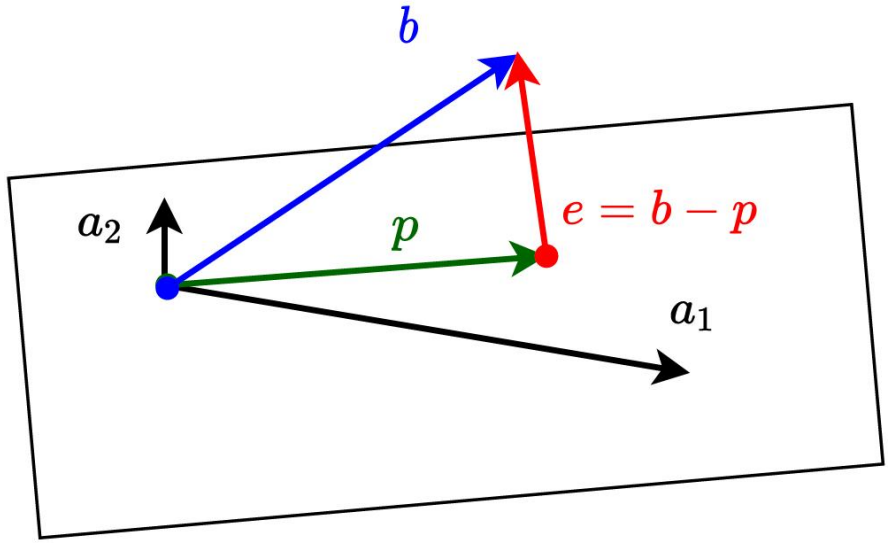
- Mặt phẳng chứa $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$ là không gian cột (column space) của A.
- Hình chiếu $\boldsymbol{p}$ của $\boldsymbol{b}$ do đó là một tổ hợp tuyến tính của $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$.

$$\boldsymbol{p} = x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2$$

$$\boldsymbol{p} = A\boldsymbol{x} = \begin{bmatrix} | & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 \\ | & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Ta cần tìm $\boldsymbol{x}$.
- Vector $\boldsymbol{e}$ thể hiện sự khác biệt giữa $\boldsymbol{b}$ và $\boldsymbol{p}$. Do đó:

$$\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{p}$$



➢ Chọn 2 vector $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$ độc lập với nhau trong mặt phẳng để tạo thành một cơ sở.

$$A = \begin{bmatrix} | & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 \\ | & | \end{bmatrix}$$

Hồi quy tuyến tính – Hình học

- $\boldsymbol{e} = \boldsymbol{b} - \boldsymbol{p}$ trực giao với mặt phẳng chứa $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$ nên ta có:

$$\begin{bmatrix} -- & \boldsymbol{a}_1^T & -- \\ -- & \boldsymbol{a}_2^T & -- \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{e} \\ | \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A^T \boldsymbol{e} = \boldsymbol{0}$$

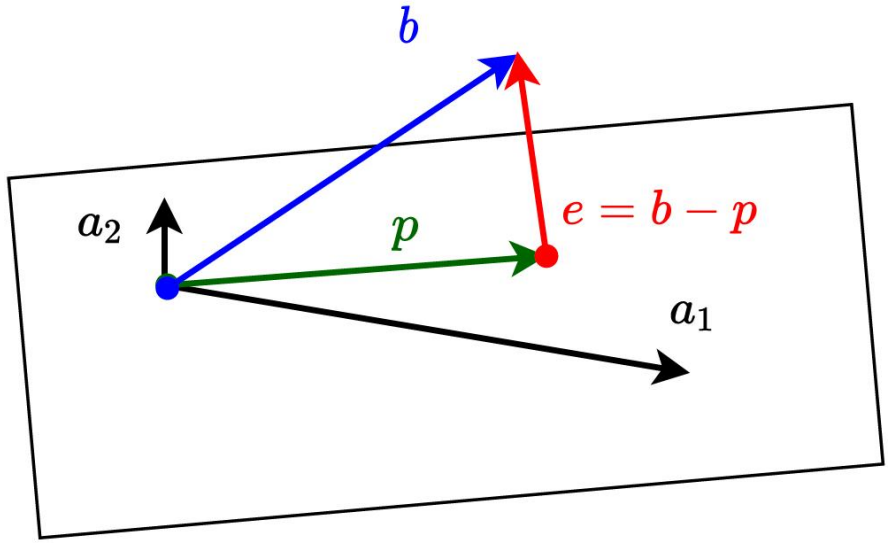
$$A^T (\boldsymbol{b} - A\boldsymbol{x}) = \boldsymbol{0}$$

$$A^T A\boldsymbol{x} = A^T \boldsymbol{b}$$

$$\boldsymbol{x} = (A^T A)^{-1} A^T \boldsymbol{b}$$

- Vector hình chiếu $\boldsymbol{p} = A\boldsymbol{x} = A(A^T A)^{-1} A^T \boldsymbol{b}$.
- Ta có ma trận chiếu (projection matrix) P là:

$$\boxed{P = A(A^T A)^{-1} A^T}$$



➢ Chọn 2 vector $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$ độc lập với nhau trong mặt phẳng để tạo thành một cơ sở.

$$A = \begin{bmatrix} | & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 \\ | & | \end{bmatrix}$$

➢ $\boldsymbol{p}$ là một tổ hợp tuyến tính của $\boldsymbol{a}_1$ và $\boldsymbol{a}_2$.

Hồi quy tuyến tính – Hình học

$$\boldsymbol{X} = \begin{bmatrix} -- & \boldsymbol{x}^{(1)^T} & -- \\ -- & \boldsymbol{x}^{(2)^T} & -- \\ & \vdots & \\ -- & \boldsymbol{x}^{(N)^T} & -- \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$$

- Hàm mất mát bình phương (squared loss) của mô hình hồi quy tuyến tính:

$$L(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y^{(i)} - f(\boldsymbol{x}^{(i)}; \boldsymbol{w})\right)^2 = \sum_{i=1}^{N} \left(y^{(i)} - \boldsymbol{w}^T \boldsymbol{x}^{(i)}\right)^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$$

- Bộ giá trị trọng số tối ưu $\boldsymbol{w}^*$ cực tiểu hóa (minimize) hàm mất mát trên:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} L(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2$$

- Dự đoán của mô hình $\boldsymbol{X}\boldsymbol{w}$ tương ứng với một tổ hợp tuyến tính các cột của ma trận $\boldsymbol{X}$, là một vector trong không gian cột của $\boldsymbol{X}$.
- Ta muốn tìm bộ giá trị trọng số tối ưu của $\boldsymbol{w}$ sao cho sự sai lệch giữa dự đoán của mô hình $\boldsymbol{X}\boldsymbol{w}$ và vector giá trị đích $\boldsymbol{y}$ là nhỏ nhất có thể.
- $\boldsymbol{X}\boldsymbol{w}^*$ tối ưu do đó sẽ là hình chiếu trực giao của $\boldsymbol{y}$ vào không gian cột của $\boldsymbol{X}$.

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

HỒI QUY ĐA THỨC

POLYNOMIAL REGRESSION

Hồi quy đa thức

Hồi quy tuyến tính đa biến:

$$f(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x} = w_0 + w_1x_1 + ... + w_Dx_D$$

Huấn luyện trên tập dữ liệu $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^N$ sử dụng hàm mất mát:

$$L(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y^{(i)} - f(\boldsymbol{x}^{(i)};\boldsymbol{w})\right)^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

với

$$\boldsymbol{X} = \begin{bmatrix} -- & \boldsymbol{x}^{(1)^T} & -- \\ -- & \boldsymbol{x}^{(2)^T} & -- \\ & \vdots & \\ -- & \boldsymbol{x}^{(N)^T} & -- \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

Lời giải: $\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$

Hồi quy đa thức:

Nếu ta muốn sử dụng mô hình

$$f(x;\boldsymbol{w}) = w_0 + w_1x + w_2x^2$$

ta có thể định nghĩa:

$$\phi(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}$$

Ta có thể biểu diễn $f(x;\boldsymbol{w}) = \boldsymbol{w}^T\phi(x)$ và giải tương tự như hồi quy tuyến tính đa biến với $\phi(x)$ đóng vai trò như vector đặc trưng $\boldsymbol{x}$.

Ma trận dữ liệu trở thành:

$$\boldsymbol{\Phi} = \begin{bmatrix} -- & \phi(x^{(1)})^T & -- \\ -- & \phi(x^{(2)})^T & -- \\ & \vdots & \\ -- & \phi(x^{(N)})^T & -- \end{bmatrix} = \begin{bmatrix} 1 & x^{(1)} & x^{(1)^2} \\ 1 & x^{(2)} & x^{(2)^2} \\ & \vdots & \\ 1 & x^{(N)} & x^{(N)^2} \end{bmatrix}$$

Lời giải: $\boldsymbol{w} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$

Hồi quy đa thức

Một mô hình (model) hồi quy đa thức (polynomial regression) thực hiện dự đoán đầu ra (output) với một hàm đa thức (polynomial function) của các đặc trưng đầu vào (input features). Với một đặc trưng x, ta có thể có mô hình sau:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_D x^D$$

trong đó, $\phi(x) = [1, x, x^2, \ldots, x^D]^T$.

Hoặc, ví dụ với các đặc trưng $\boldsymbol{x} = (x_1, x_2)$, ta có thể có mô hình sau:

$$f(\boldsymbol{x}; \boldsymbol{w}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

trong đó, $\phi(\boldsymbol{x}) = [1, x_1, x_2, x_1 x_2, x_1^2, x_2^2]^T$. Đây là một ví dụ *chế tác đặc trưng (feature engineering)*.

Lưu ý:

- Mô hình hồi quy đa thức là phi tuyến tính theo các đặc trưng đầu vào $\boldsymbol{x}$.
- Mô hình hồi quy đa thức là tuyến tính theo các trọng số $\boldsymbol{w}$ của mô hình. Do đó, ta giải quyết bài toán hồi quy đa thức tương tự như hồi quy tuyến tính đa biến (multiple linear regression).

$$\boldsymbol{\Phi} = \begin{bmatrix} -- \phi(x^{(1)})^T -- \\ -- \phi(x^{(2)})^T -- \\ \vdots \\ -- \phi(x^{(N)})^T -- \end{bmatrix}. \text{ Ta có: } \hat{\boldsymbol{w}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}.$$

Hồi quy đa thức – Ví dụ

Tập dữ liệu (Dataset)

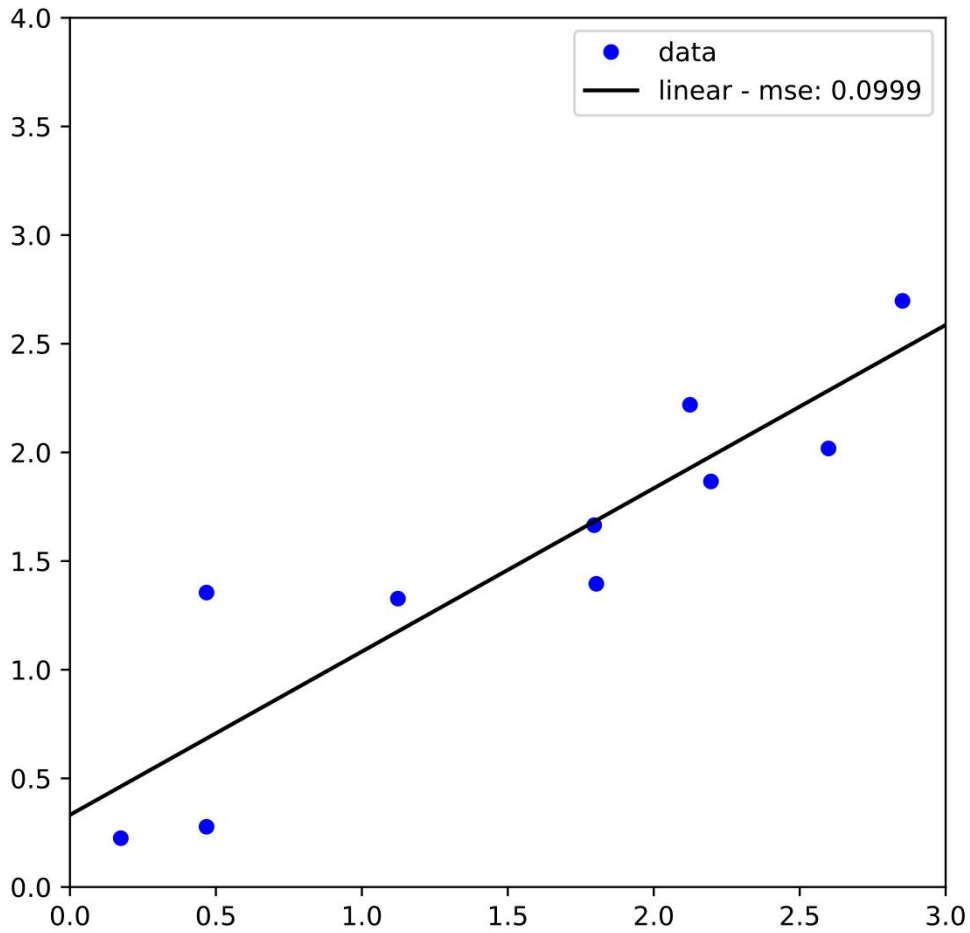
Phát sinh một tập dữ liệu (dataset) gồm N điểm dữ liệu (data points) $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ với:

- $x^{(i)} \in (0,1)$ là đặc trưng đầu vào (input feature) của điểm dữ liệu thứ i.
- $y^{)i(} \in \mathbb{R}$ là giá trị đích (target value) của điểm dữ liệu thứ i. Ta mô phỏng giá trị $y^{)i(}$ thu thập được có một lượng nhiễu nhỏ cho mỗi điểm dữ liệu.

Hồi quy tuyến tính đơn giản

Sử dụng một mô hình hồi quy tuyến tính đơn giản, ta có:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x$$




```
from sklearn.linear_model import LinearRegression
np.random.seed(42)
N = 10
X_train = 3.0*np.random.rand(N, 1)
y_train = 1.0 + 0.5*X_train + np.random.randn(N,1)/2.0
lin_reg = LinearRegression()
lin_reg.fit(X_train,y_train)
```

Hồi quy đa thức – Ví dụ

Hồi quy tuyến tính đơn giản

- Sử dụng một mô hình hồi quy tuyến tính đơn giản:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x$$

Hồi quy đa thức

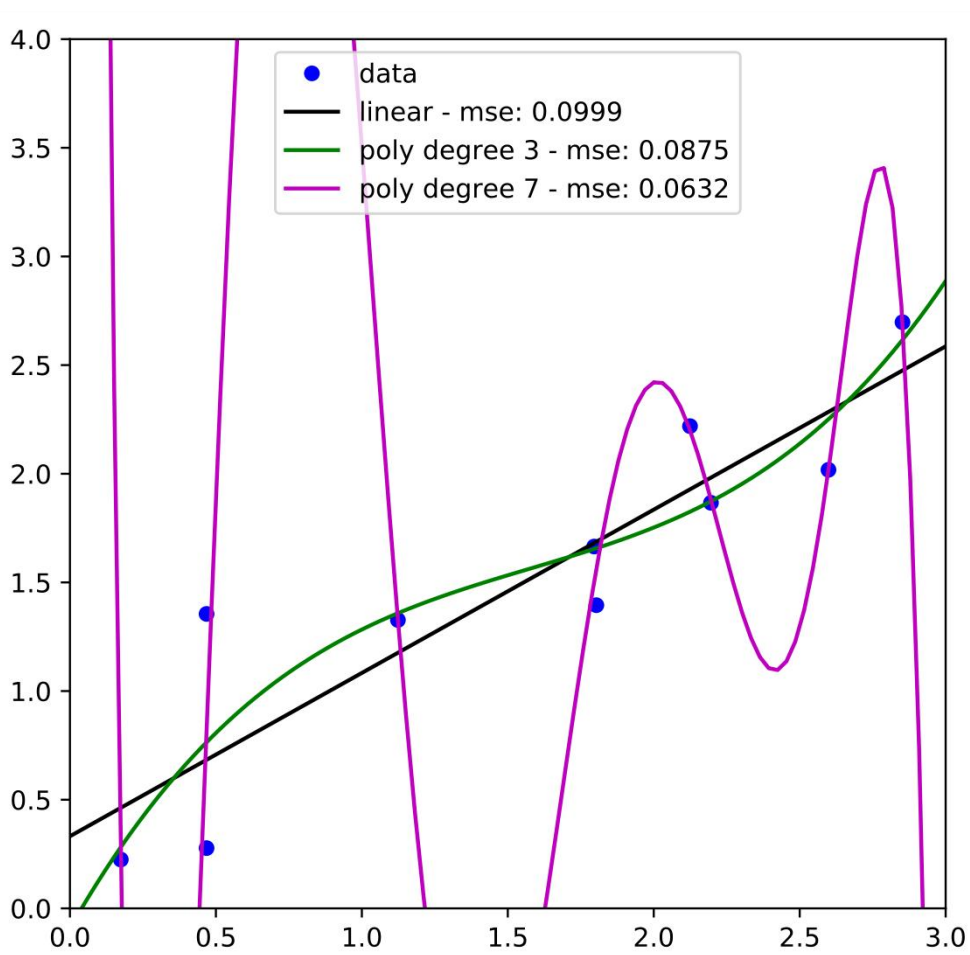
- Sử dụng một mô hình hồi quy đa thức bậc 3:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Sử dụng một mô hình hồi quy đa thức bậc 7:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_7 x^7$$

- Mô hình nào là tốt nhất đối với tập dữ liệu đang xét?




```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
deg = 7
np.random.seed(42)
poly = PolynomialFeatures(degree=deg,
include_bias=False)
X_train = 3.0*np.random.rand(N, 1)
X_poly = poly.fit_transform(X_train)
y_train = 1.0 + 0.5*X_train + np.random.randn(N,1)/2.0
poly_reg = LinearRegression()
poly_reg.fit(X_poly,y_train)
lin_reg = LinearRegression()
lin_reg.fit(X_train,y_train)
```

Hồi quy đa thức – Ví dụ

- Hồi quy tuyến tính đơn giản:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x$$

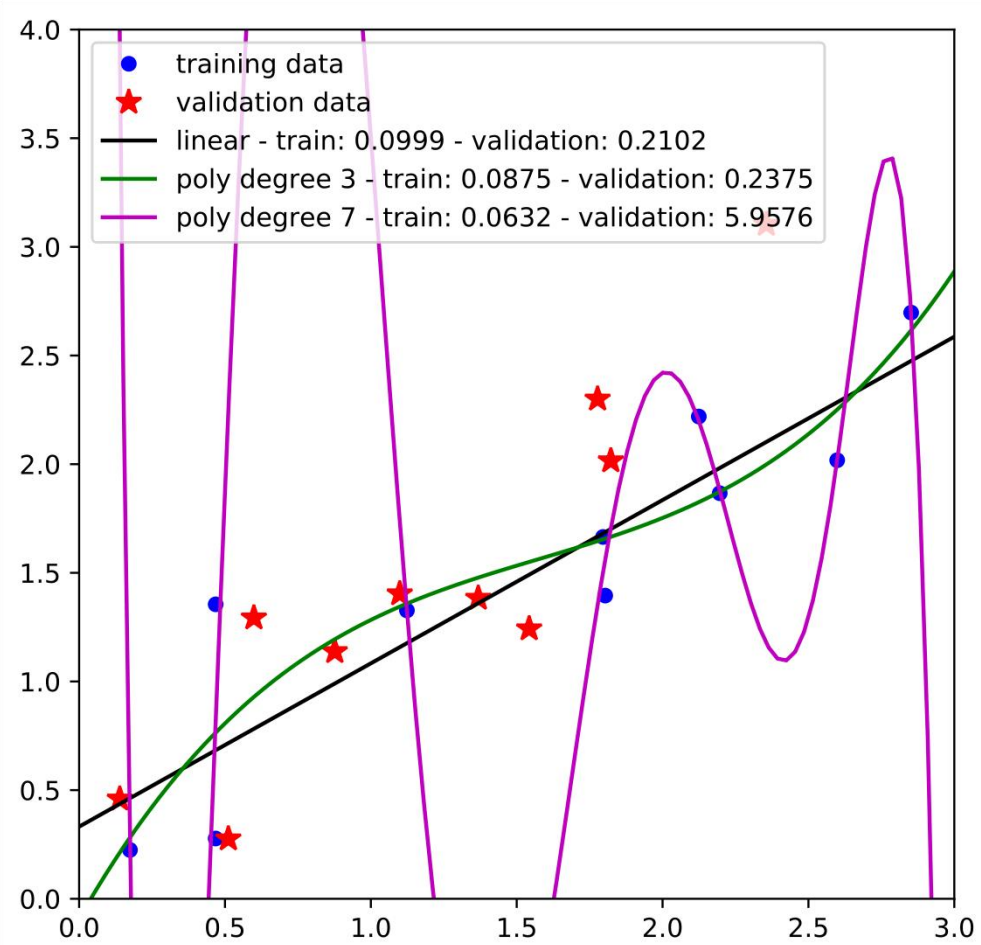
- Hồi quy đa thức bậc 3:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

- Hồi quy đa thức bậc 7:

$$f(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_7 x^7$$

- Mô hình nào là tốt nhất đối với tập dữ liệu đang xét?
- Ta cần đánh giá các mô hình sau khi huấn luyện trên một tập dữ liệu khác với tập dữ liệu đã dùng để huấn luyện các mô hình.
- Mô hình phức tạp có thể có độ lỗi nhỏ trên tập huấn luyện nhưng có độ lỗi lớn trên tập dữ liệu đánh giá.




```
N=10
Xtrain= 3.0*np.random.rand(N, 1)
ytrain= 1.0 + 0.5*Xtrain+ np.random.randn(N,1)/2.0
N_test = 10
Xtest= 3.0*np.random.rand(N_test, 1)
ytest= 1.0 + 0.5*Xtest+ np.random.randn(N_test,1)/2.0
```

```
from sklearn.preprocessing import PolynomialFeatures
deg=7
poly = PolynomialFeatures(degree=deg,
include_bias=False)
X_poly = poly.fit_transform(X_train)
poly_reg = LinearRegression()
poly_reg.fit(X_poly, y_train)
```

QUÁ KHỚP VÀ KỸ THUẬT ĐIỀU CHUẨN

OVERFITTING & REGULARIZATION

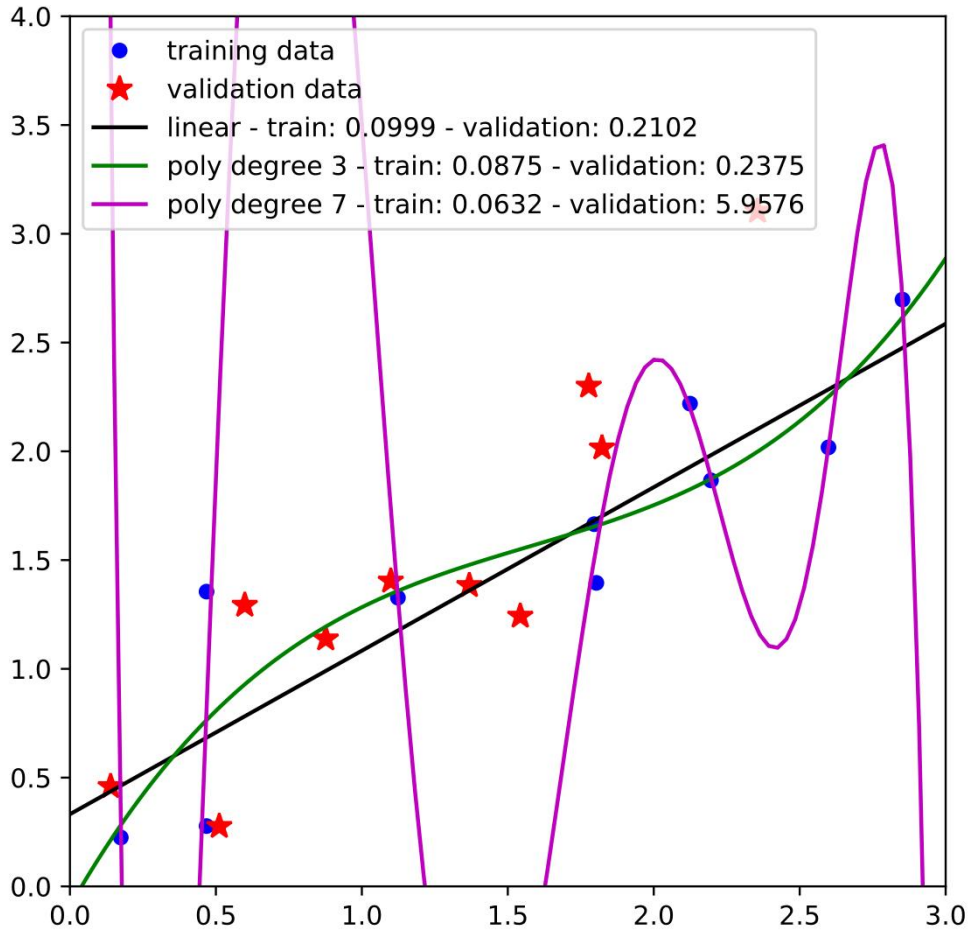
Hồi quy đa thức – Quá khớp

- Mô hình nào là tốt nhất đối với tập dữ liệu đang xét?

```
---Simple Linear Regression---
[0.33181066] [[0.75155622]]
Training Loss:  0.09991531905633781
Validation Loss:  0.21020128681013045

---Polynomial Regression degree = 3---
[-0.0916451] [[ 2.35120734 -1.23811156  0.26177074]]
Training Loss:  0.08750190066671261
Validation Loss:  0.23747048298743456

---Polynomial Regression degree = 7---
[ 3 8 . 0 6 9 8 7 1 0 1 ]  [ [  - 3 9 9 . 6 0 8 7 3 8 9 3   1 3 6 4 . 6 9 5 7 9 4 8 8  -
2093.21939536  1665.82047801
   -716.24860925   158.03695712   -14.0390417 ]]
Training Loss:  0.06322418953671403
Validation Loss:  5.9575569104420065
```



- Quá khớp (overfitting): hiện tượng mô hình dự đoán rất chính xác trên dữ liệu huấn luyện, nhưng lại không thể dự đoán tốt trên dữ liệu mới (không tổng quát hóa tốt).

Hiện tượng quá khớp (Overfitting)

Giả sử ta sử dụng các kỹ thuật chế tác đặc trưng (feature engineering) với các phép biến đổi $\boldsymbol{\Phi}$ cho mô hình hồi quy tuyến tính. Giả sử số điểm dữ liệu huấn luyện $N = 10$.

Với tập dữ liệu trên, ta huấn luyện mô hình để tìm bộ giá trị trọng số $\boldsymbol{w}$ sao cho mô hình dự đoán chính xác nhất có thể: $\boldsymbol{y} \approx \boldsymbol{\Phi w}$.

- Nếu ta sử dụng $\boldsymbol{\Phi}$ với 2 đặc trưng thì ta có: $\tilde{\boldsymbol{y}}^{10\times1} \approx \tilde{\boldsymbol{\Phi}}^{10\times2}\tilde{\boldsymbol{w}}_{2\times1}$.

- Nếu ta sử dụng $\boldsymbol{\Phi}$ với 10 đặc trưng thì ta có: $\tilde{\boldsymbol{y}}^{10\times1} \approx \tilde{\boldsymbol{\Phi}}^{10\times10}\tilde{\boldsymbol{w}}_{10\times1}$.

- Ta có thể giải chính xác giá trị tối ưu của $\boldsymbol{w}$ nếu $\boldsymbol{\Phi}$ khả nghịch: $\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{y}$.

➢ Giá trị của hàm mất mát trên tập dữ liệu huấn luyện là bao nhiêu?

➢ Mô hình này liệu có hoạt động tốt khi dự đoán trên các mẫu dữ liệu mới không nằm trong số N điểm dữ liệu huấn luyện?

Điều chuẩn (Regularization)

Ta muốn sử dụng các mô hình bậc cao và có tính linh hoạt cao để có kết quả huấn luyện tốt trên tập dữ liệu huấn luyện, nhưng ta cũng muốn kiểm soát "độ phức tạp" của mô hình để giảm thiểu khả năng quá trình huấn luyện dẫn đến hiện tượng quá khớp.

Ta sử dụng hàm mất mát được điều chuẩn (regularized) để huấn luyện mô hình:

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \{L(\boldsymbol{w}) + \text{penalty}(\boldsymbol{w})\}$$

- $L(w)$: hàm mất mát thông thường, ví dụ như hàm mất mát bình phương.
- $\text{penalty}(\boldsymbol{w})$: hàm phạt được sử dụng để giới hạn độ lớn của các tham số $\boldsymbol{w}$. Hai phương pháp phạt phổ biến là:
  - Ridge – L_2 regularization
  - Lasso – L_1 regularization

Điều chuẩn (Regularization)

Ridge – L_2 regularization

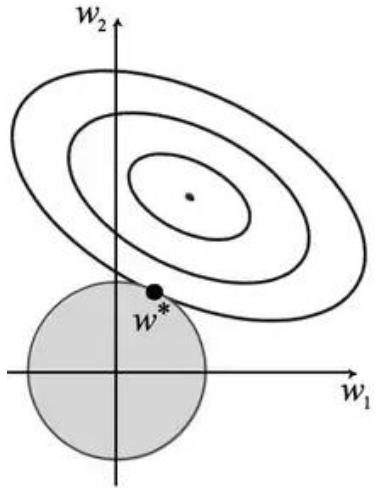
$$L_\lambda(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y^{(i)} - f(\boldsymbol{x}^{(i)}; \boldsymbol{w})\right)^2 + \lambda \sum_{k=1}^{D} (w_k)^2$$

Ta thường không cần phải điều chuẩn w_0. Tại sao?

Ta có thể chuẩn hóa zero-mean dữ liệu trước khi huấn luyện mô hình: các cột của $\boldsymbol{X}$ (hoặc $\boldsymbol{\Phi}$) được chuẩn hóa để có giá trị trung bình là 0.

Lời giải dạng đóng (closed-form solution) của Hồi quy Ridge là:

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \mathbf{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

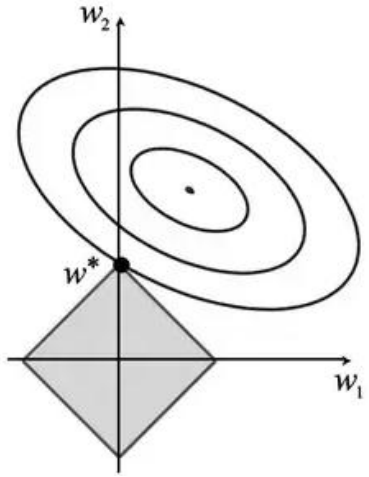


Lasso – L_1 regularization

$$L_\lambda(\boldsymbol{w}) = \sum_{i=1}^{N} \left(y^{(i)} - f(\boldsymbol{x}^{(i)}; \boldsymbol{w})\right)^2 + \lambda \sum_{k=1}^{D} |w_k|$$

L_1 regularization không có lời giải dạng đóng vì có thành phần tính giá trị tuyệt đối $|w_k|$.

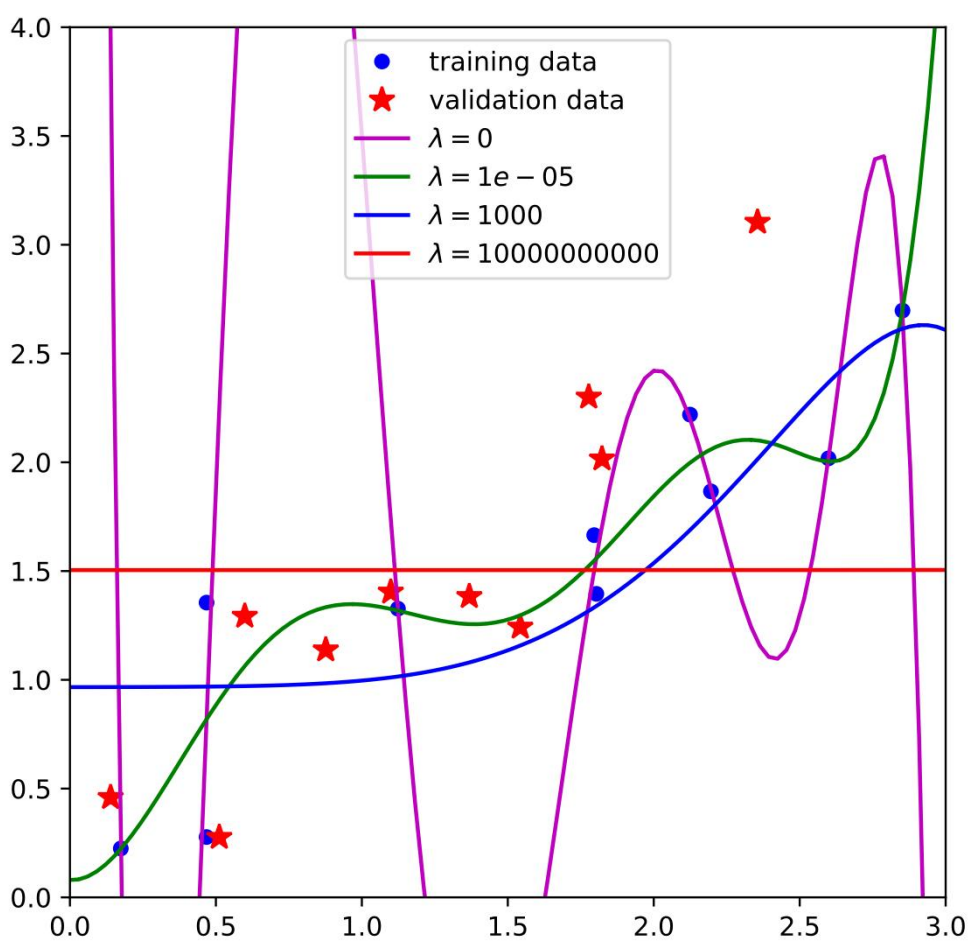
Lasso có thể đưa trọng số w_k của một số đặc trưng x_k về giá trị 0. Ta có mô hình đơn giản hơn, loại bỏ đi những đặc trưng không cần thiết.



Hồi quy đa thức – Quá khớp

```
---Polynomial Regression degree = 7---
[38.06987101] [[ -399.60873893  1364.69579488 -2093.21939536  1665.82047801
   -716.24860925   158.03695712   -14.0390417 ]]
Training Loss:  0.06322418953671403
Validation Loss:  5.9575569104420065


--- lambda = 0 ---
[38.06839355] [ -399.5931998  1364.64294788 -2093.13814505  1665.75551427
  -716.22051288  158.03071754  -14.03848365]
--- lambda = 1e-05 ---
[0.07987999] [-0.09370931  6.14945012 -4.77478402 -3.78021829  5.75140457 -2.28394658
  0.29814782]
--- lambda = 1000 ---
[0.96643294] [ 0.00141423  0.00266913  0.00442294  0.00683627  0.00931761  0.00896911
 -0.00359498]
--- lambda = 10000000000 ---
[1.50446301] [6.07278017e-10 1.77351966e-09 4.72471362e-09 1.25448975e-08
 3.35844395e-08 9.07463757e-08 2.47260904e-07]
```




```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import Ridge
deg = 7
poly = PolynomialFeatures(degree=deg, include_bias=False)
X_poly = poly.fit_transform(X_train)
lambd = 10e-5
ridge_reg = Ridge(alpha=lambd)
ridge_reg.fit(X_poly, y_train)
```

- Giá trị phù hợp cho siêu tham số (hyperparameter) λ = ?