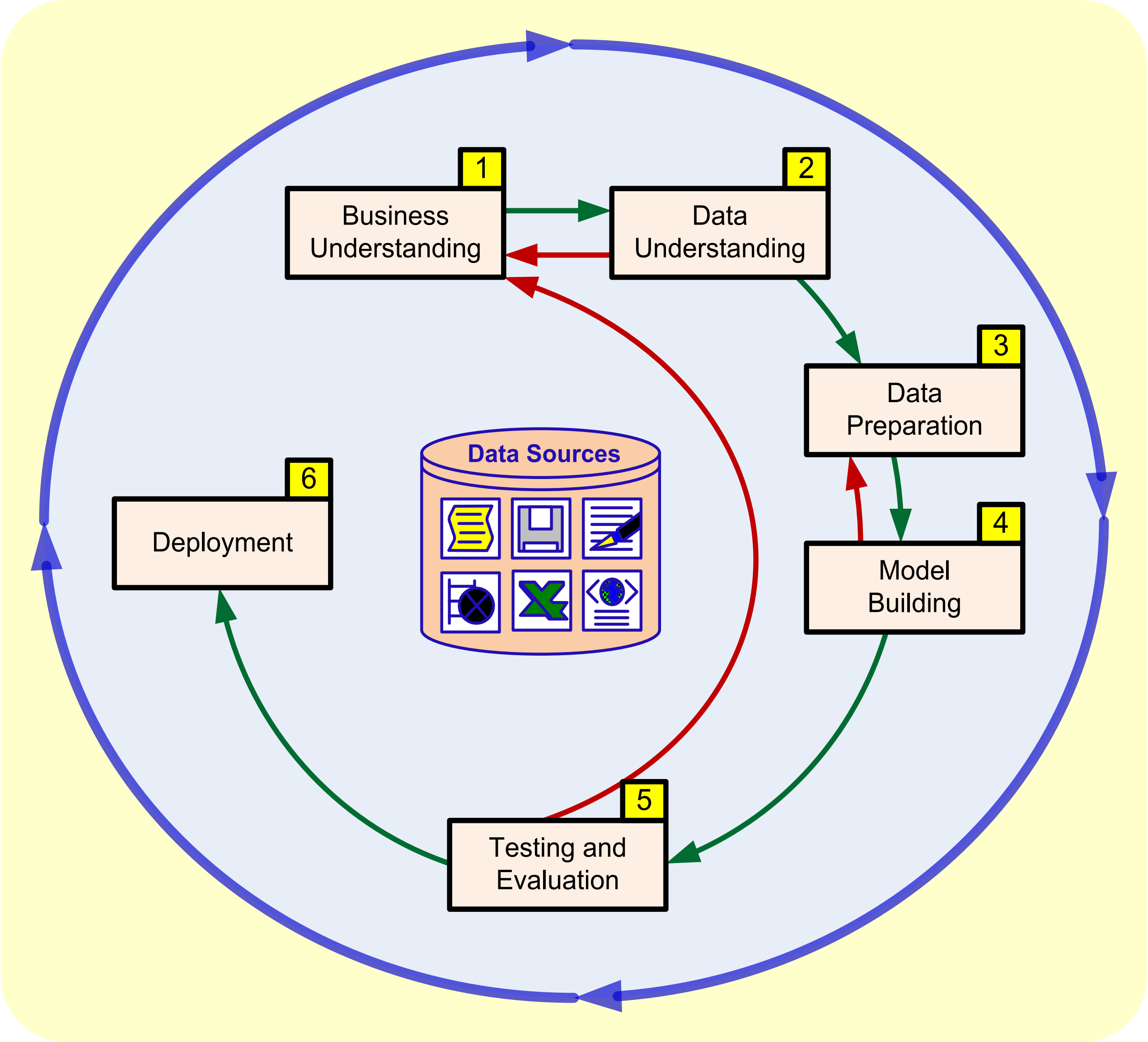
Chapter 3: Linear Regression and Logistic Regression

Yunming Ye, Baoquan Zhang
School of Computer Science
Harbin Institute of Technology, Shenzhen

Data Mining Process Model



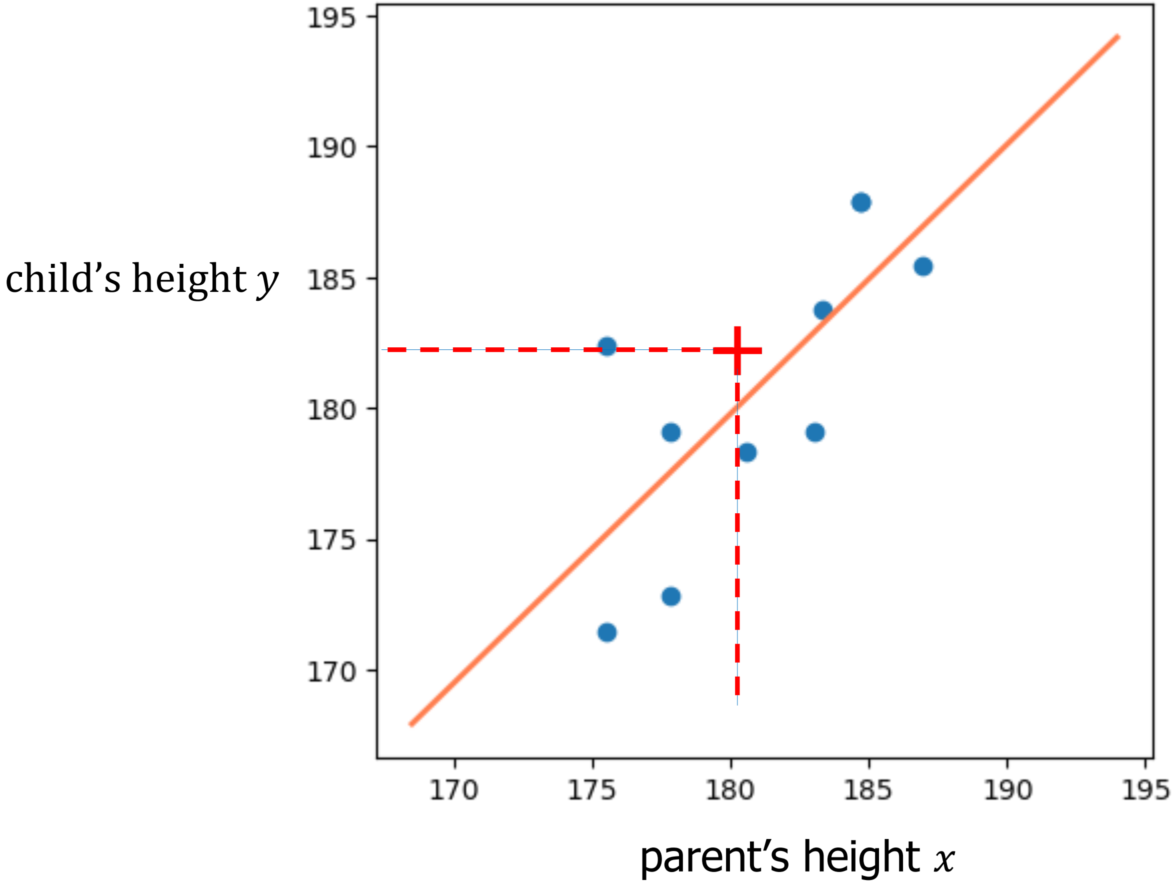
Agenda

- Basic Concept of Regression
- Linear Regression
- Least Square Method
- Gradient Descent Method
- Logistic Regression

3.1: Basic Concept of Regression

Regression Task

- regression : predicting the target value of a given object of data (corresponding to the category of the classification)



when x = 180 , y = ?

Application of regression prediction

- Almost every AI application involves the problem of prediction
  - Stock forecasts
  - Loan amount estimate
  - Video predictions
  - Sales performance forecasts
  - Medical diagnosis
  - Fraud detection
  - ……

Definition of the regression task

- The regression task can be represented by a function:

$$y = f(\boldsymbol{x}),$$

$$\text{in which } \boldsymbol{x} \in \mathbb{D}, y \in \mathbb{R}$$

- The regreesion function f(x), also called “regression model” , outputs a continuous real value y by calculation.

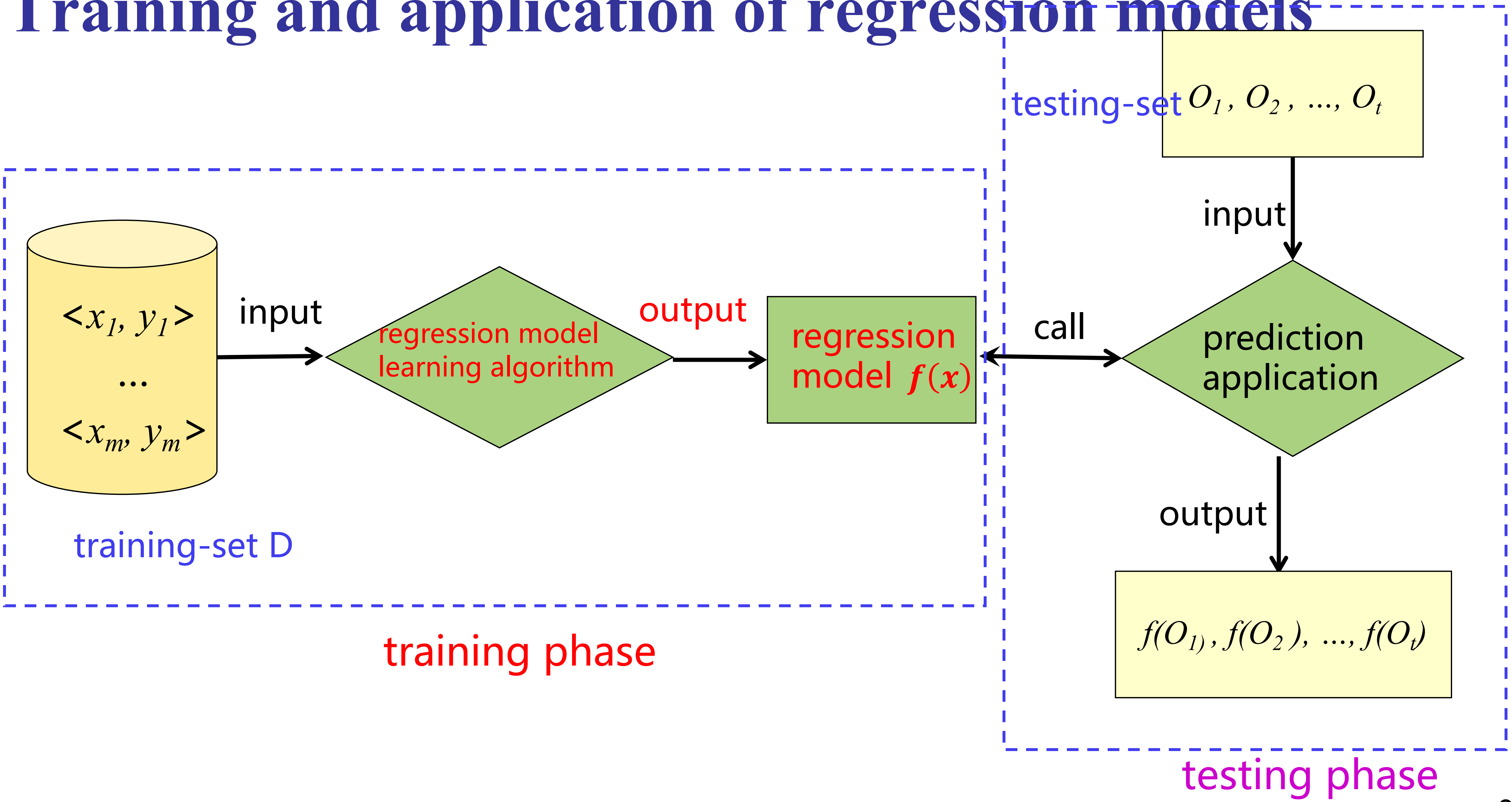
how to construct the regression function f(x)?

the "two-phase" process of the regression task

- **training the regression model (training phase)**
  - Learn from a training dataset where the target value is known to generate a regression model f(x).
  - Regression models can be represented as linear functions, hyperplanes, regression trees, and so on.

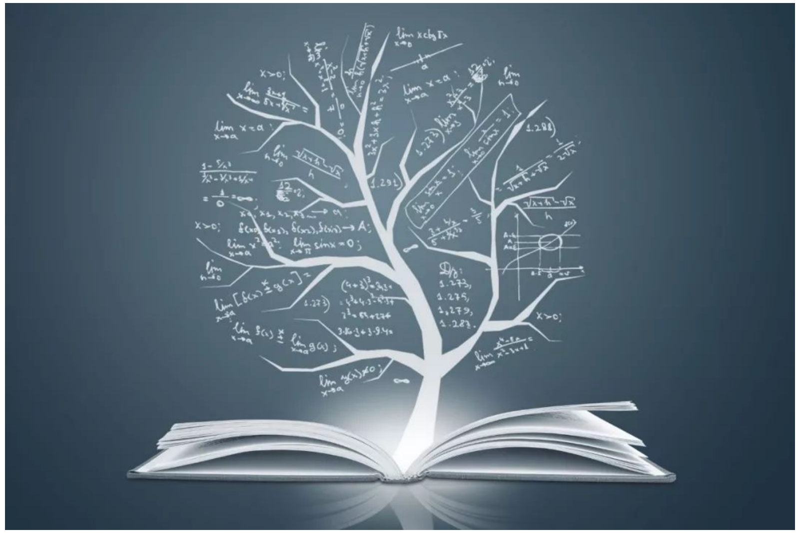
- **applying the regression model (testing phase):**
  - Use the regression model f(x) to predict the target value of a new data object.

Training and application of regression models



Commonly used regression models

- linear regression
  - Lasso Regression
  - Ridge Regression
  - ElasticNet Regression



- non-linear regression
  - K Neighbors Regression
  - Decision Tree Regression
  - Support Vector Regression，SVR
  - Ensemble regression：Random Forest、AdaBoost、XGBoost、LightGBM
  - Deep Learning

3.2 Linear regression

Application cases

$$y = f(\boldsymbol{x}), \text{in which } \boldsymbol{x} \in \mathbb{D}, y \in \mathbb{R}$$

- predicting the average temperature for the next day
  - dataset from Kaggle (daily climate time series data)

日期 (date)	日均气温 (mean temp)	相对湿度 (humidity)	风速 (wind speed)	气压 (pressure)
…	…	…	…	…
2017-01-02	7.40	92.00	2.980	1017.80
2017-01-03	7.17	87.00	4.63	1018.67

$\boldsymbol{x}$

f

y

https://www.kaggle.com/sumanthvrao/daily-climate-time-series-data

linear regression model

- given dataset $\mathbb{X} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_n}, y_n)\}$, in which there are d features of each sample $\boldsymbol{x_i}$:

$$\boldsymbol{x_i} = \left(x_{i,1}; x_{i,2}; \dots; x_{i,d}\right)^T, \quad y_i \in \mathbb{R}.$$

- The purpose of a linear regression model is to learn a linear function $f(\boldsymbol{x})$ about $\boldsymbol{x}$ to predict y as accurately as possible.

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b \qquad \boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_d \end{pmatrix} \qquad y \approx f(\boldsymbol{x})$$

  - that is: the smaller the bias between $f(\boldsymbol{x})$ and y, the better.
  - $\boldsymbol{w}$ and b are the parameters that need to be learned.

Evaluation of linear regression model

- Before solving $\boldsymbol{w}$ and b, a loss function measuring the error between $f(\boldsymbol{x})$ and y needs to be given.
- In regression methods, mean squared error is a commonly used loss function, which is defined as follows:

$$L(\boldsymbol{w}, b) = \frac{1}{2n}\sum_{i=1}^{n}(f(\boldsymbol{x_i}) - y_i)^2$$

$$(\boldsymbol{x_i}, y_i)$$

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$

Linear regression of individual variables

- Step 1: Determine the model space
  - an intuitive idea: The daily average temperature of the next day is likely to be related to the daily average temperature of the previous day.

regression model

$$y = w \cdot x_t + b$$

w :weight, b : bias

$\Rightarrow$

$y = 0.9 \cdot x_t + 2.0$
$y = 1.0 \cdot x_t + 0.1$
$y = 1.1 \cdot x_t + 1.8$
… …

model space
(function set)

loss function

- Step 2: Evaluation criteria for the merits of the model

$$L(w,b)=\sum_{i=1}^{n}\underbrace{\left[y_i-\boxed{\left(w\cdot x_{i,t}+b\right)}\right]^2}$$

prediction error

prediction result

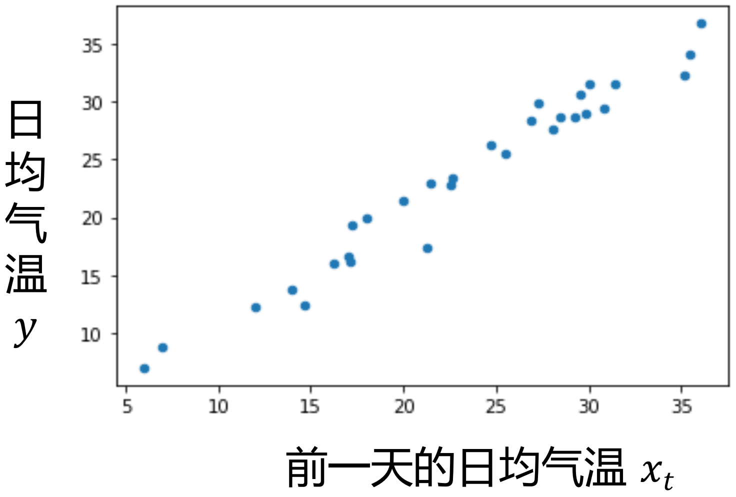
$y = 0.9 \cdot x_t + 2.0$
$y = 1.0 \cdot x_t + 0.1$
$y = 1.1 \cdot x_t - 1.8$
… …

model space
(function set)

- For ease of calculation, we randomly selected thirty days as the training set (units, °C)

$(x_{1,t}\ ,\ y_1)$
$(x_{2,t}\ ,\ y_2)$
⋮
$(x_{30,t}, y_{30})$

Scatter plot



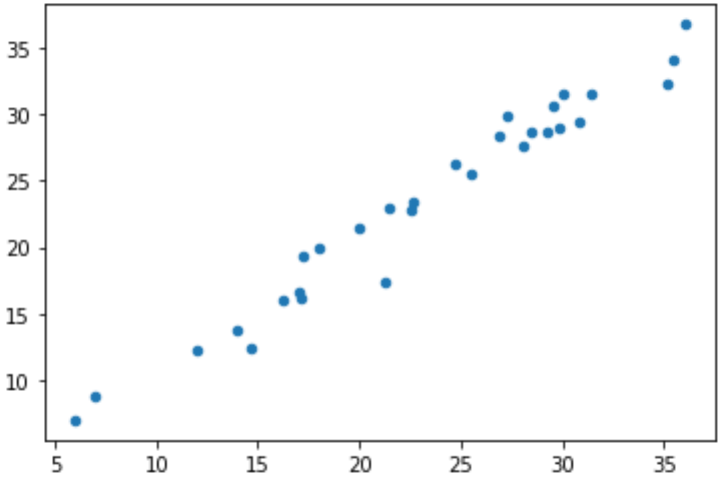
loss function

- Contour plot of the loss function

$$L(w,b)=\sum_{i=1}^{n}\left(y_i - w\cdot x_{i,t} - b\right)^2$$

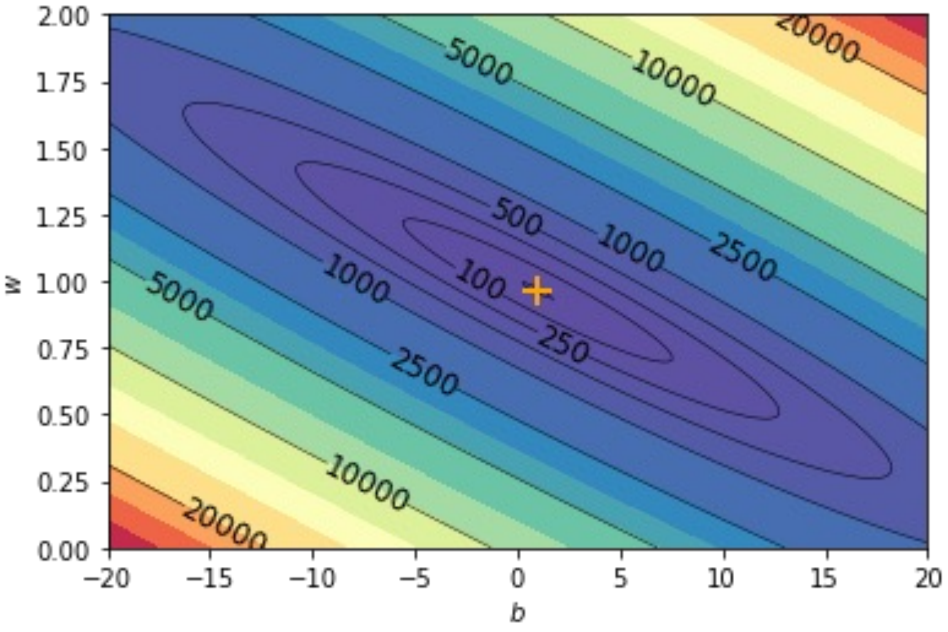
How do I find the minimum point?

daily temperature of the next day y



daily temperature of the previous day x_t

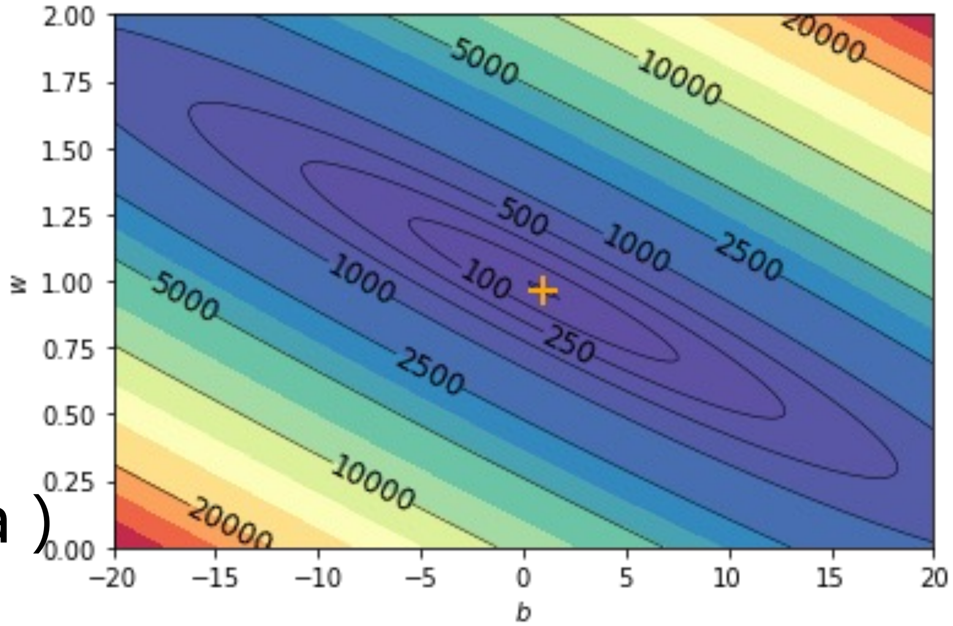
model space



Optimization algorithm

- Step 3: Find the "optimal" model
- find the optimal parameter (w^*, b^*) that minimize the loss function

$$w^*, b^* = \arg\min_{b,w} L(w,b)$$
$$= \arg\min_{b,w} \sum_{i=1}^{n} \left[y_i - \left(w \cdot x_{i,t} + b\right)\right]^2$$



- commonly used algorithm :
  - Least squares method (small-scale data)
  - Gradient descent

3.3 Linear regression model based on least squares method

Linear regression model based on least squares

- The data sample is represented by a matrix X with size of $n \times (d+1)$, each row is a sample, and each column is a feature of the sample, that is :

$$\boldsymbol{x_i} = (x_{i,1}; x_{i,2}; \dots; x_{i,d})$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ 1 & x_{2,1} & \dots & x_{2,d} \\ 1 & \dots & \dots & \dots \\ 1 & x_{n,1} & \dots & x_{n,d} \end{pmatrix}$$

$$f(\boldsymbol{x}) = \boldsymbol{wx} + b$$

- in which the first column is constantly 1, corresponding to the constant term b in the case of vector multiplication.

Vector form of the loss function

- The target values of the dataset can also be written as vectors $\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$
- the parameters $\boldsymbol{w}$ and b can be merged as $\boldsymbol{W} = \binom{b}{\boldsymbol{w}}$
- Thus, the loss function of linear regression can be rewritten as follows:

$$L(\boldsymbol{W}) = \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{f}(\boldsymbol{x}_i) - y_i)^2 = \frac{1}{2}(\boldsymbol{XW} - \boldsymbol{y})^T(\boldsymbol{XW} - \boldsymbol{y}) \qquad f(\boldsymbol{x}) = \boldsymbol{wx} + b$$

process of least squares

- Our goal is to find an optimal set of parameters$\boldsymbol{W}^* = \binom{b^*}{\boldsymbol{w}^*}$, which is able to minimize the loss ：

$$\underset{\boldsymbol{W}}{\operatorname{argmin}}\, L(\boldsymbol{W}) \qquad L(\boldsymbol{W}) = \frac{1}{2}(\boldsymbol{XW} - \boldsymbol{y})^T(\boldsymbol{XW} - \boldsymbol{y})$$

- If $X^T X$ is reversible, the optimal solution of W can be obtained directly by deriving and settling on $X^T X$:

$$\begin{aligned}
\frac{\partial L}{\partial \boldsymbol{W}} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{W}}(\boldsymbol{W}^T X^T X \boldsymbol{W} - \boldsymbol{W}^T X^T \boldsymbol{y} - \boldsymbol{y}^T X \boldsymbol{W} + \boldsymbol{y}^T \boldsymbol{y}) \\
&= \frac{1}{2}(2X^T X \boldsymbol{W} - X^T \boldsymbol{y} - X^T \boldsymbol{y}) \\
&= X^T X \boldsymbol{W} - X^T \boldsymbol{y}
\end{aligned}$$

process of least squares

- we have

$$\frac{\partial L}{\partial \boldsymbol{W}} = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{W}}(\boldsymbol{W}^T X^T X \boldsymbol{W} - \boldsymbol{W}^T X^T y - y^T X \boldsymbol{W} + y^T y)$$

$$= \frac{1}{2}(2X^T X \boldsymbol{W} - X^T y - X^T y)$$

$$= X^T X \boldsymbol{W} - X^T y$$

- let

$$X^T X \boldsymbol{W} - X^T y = 0$$

so that we can get the analytical solution of $\boldsymbol{W}$:

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

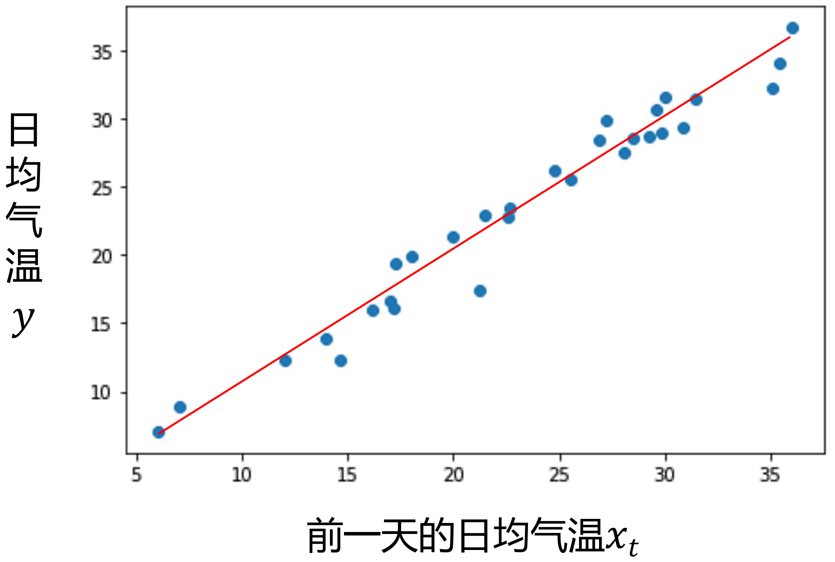
Example of an application of least squares

- Linear regression model for a single variable: $y = w \cdot x_t + b$
- The data is expressed as :

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{1,t} \\ 1 & x_{2,t} \\ \dots & \dots \\ 1 & x_{30,t} \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_{30} \end{pmatrix} \qquad \boldsymbol{W} = \begin{pmatrix} b \\ w \end{pmatrix}$$

- we can get :

$$\boldsymbol{W} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y = \begin{pmatrix} 0.980 \\ 0.965 \end{pmatrix}$$



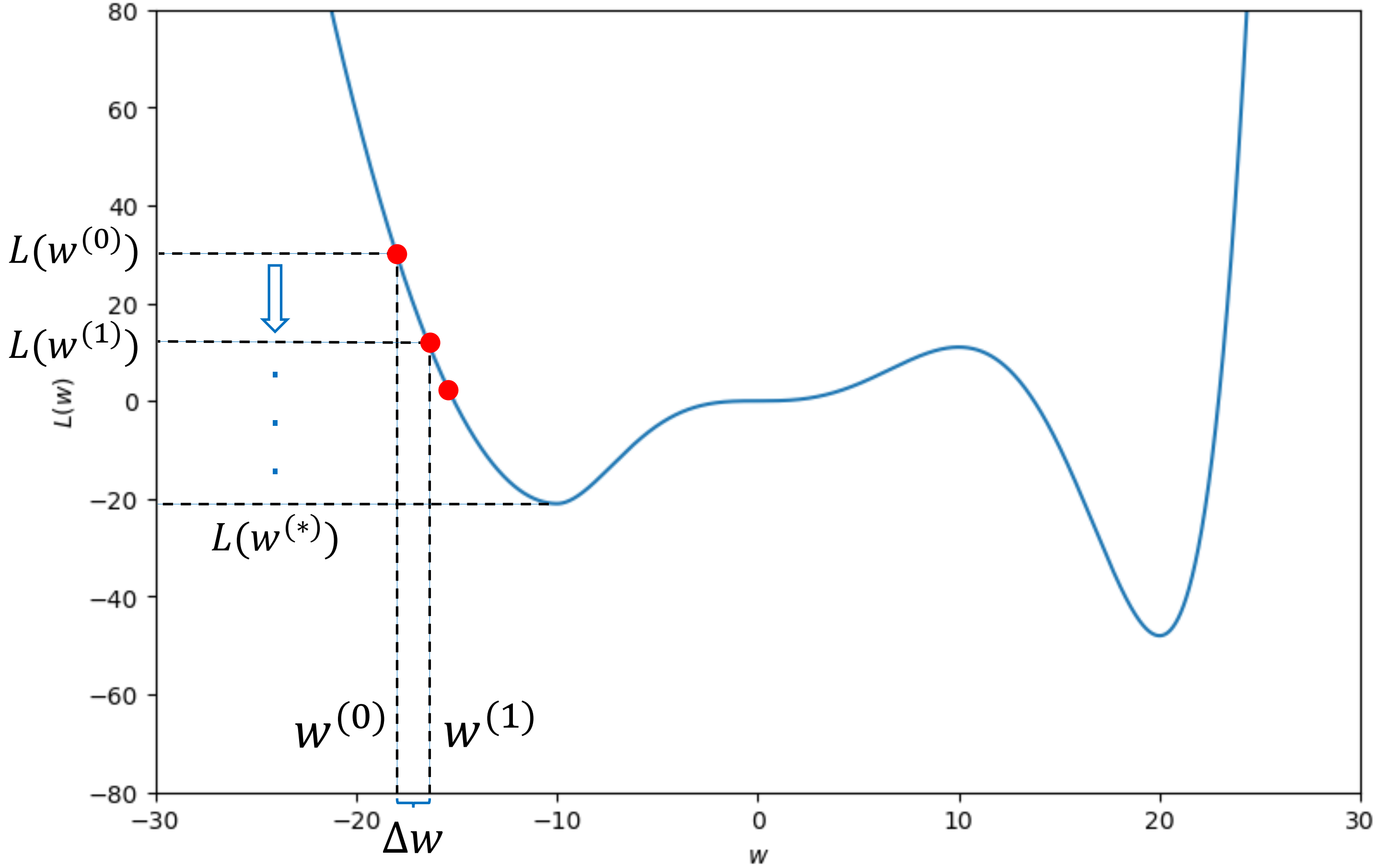
The problem of least squares

- When these n independent variables are not independent of each other, but with some linear relationship, $X^T X$ will be irreversible, and the resulting solution is a pathological solution so that cannot be used as the optimal parameter learned

- When $X^T X$ is irreversible, it can be solved by gradient descent

3.4 Linear regression model based on gradient descent

The basic idea of gradient descent



Mathematical principles of gradient descent (unary function)

- The unary function Taylor formula
  - if the unary function $L(w)$ is derivable in neighborhood of point $w^{(0)}$, then we have:

$$L(w) = L\left(w^{(0)}\right) + L'\left(w^{(0)}\right)\left(w - w^{(0)}\right) + o\left(w - w^{(0)}\right)$$

- if the variety $\Delta w = w - w^{(0)} = -\eta L'\left(w^{(0)}\right)$, and the learing rate η is a small positive number, then:

"Negative Gradient Direction"

$$\begin{aligned} L(w) &\approx L\left(w^{(0)}\right) - L'\left(w^{(0)}\right) \cdot \eta L'\left(w^{(0)}\right) \\ &= L\left(w^{(0)}\right) - \eta\left(L'\left(w^{(0)}\right)\right)^2 \\ &< L\left(w^{(0)}\right) \end{aligned}$$

$$L'\left(w^{(0)}\right) = \frac{dL}{dw}\Big|_{w=w^{(0)}}$$

Mathematical principles of gradient descent (multivariate functions)

- thinking of linear regression model $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b, \boldsymbol{w} = \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_d \end{pmatrix}$, loss fuction $L(\boldsymbol{w}, b)$ is a multivariate function
- For brevity, the parameters $\boldsymbol{w}$ and b to be solved are represented as W , $W = \begin{pmatrix} b \\ \boldsymbol{w} \end{pmatrix}$
- Generalize the Unary Function Taylor Formula $L(w) = L\left(w^{(0)}\right) + L'\left(w^{(0)}\right)\left(w - w^{(0)}\right) + o\left(w - w^{(0)}\right)$, we can get :

$$L(W) = L\left(W^{(0)}\right) + \nabla L\left(W^{(0)}\right)^T\left(W - W^{(0)}\right) + o\left(W - W^{(0)}\right),$$

$$\nabla L\left(W^{(0)}\right) = \left(\frac{\partial L}{\partial b}|_{W=W^{(0)}}, \quad \frac{\partial L}{\partial w_1}|_{W=W^{(0)}}, \quad \frac{\partial L}{\partial w_2}|_{W=W^{(0)}}, \dots, \frac{\partial L}{\partial w_d}|_{W=W^{(0)}}\right)^T$$

Mathematical principles of gradient descent (multivariate functions)

- The gradient $\nabla L(W^{(0)})$ of the multivariate function $L(W)$ at the point $W^{(0)}$ is a vector :
  - The direction of the gradient is the same as the direction in which the maximum square number o guides is obtained ,
  - The modulus of the gradient is the maximum number of square wizards。
- according to the formular $L(W) = L(W^{(0)}) + \nabla L(W^{(0)})^T (W - W^{(0)}) + o(W - W^{(0)})$,

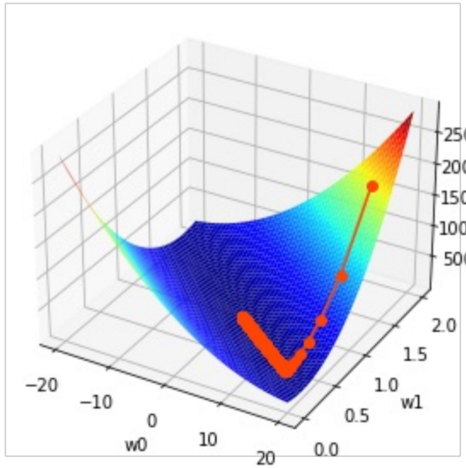
if $\Delta W = W - W^{(0)} = -\eta \nabla L(W^{(0)})$, learining rate η is a small positive number, then

"Negative Gradient Direction"

$\Downarrow$

The function value drops

$$L(W) \approx L(W^{(0)}) - \eta \nabla L(W^{(0)})^T \nabla L(W^{(0)})$$
$$= L(W^{(0)}) - \eta \left\| \nabla L(W^{(0)}) \right\|^2$$
$$< L(W^{(0)})$$



Process of gradient descent (single parameter)

- Take l(w), a smooth loss function with only a single argument w, as an example, gradient descent

randomly choose init value $w^{(0)}$

$$w^{(1)} \leftarrow w^{(0)} - \eta \frac{dL}{dw}|_{w=w^{(0)}}$$

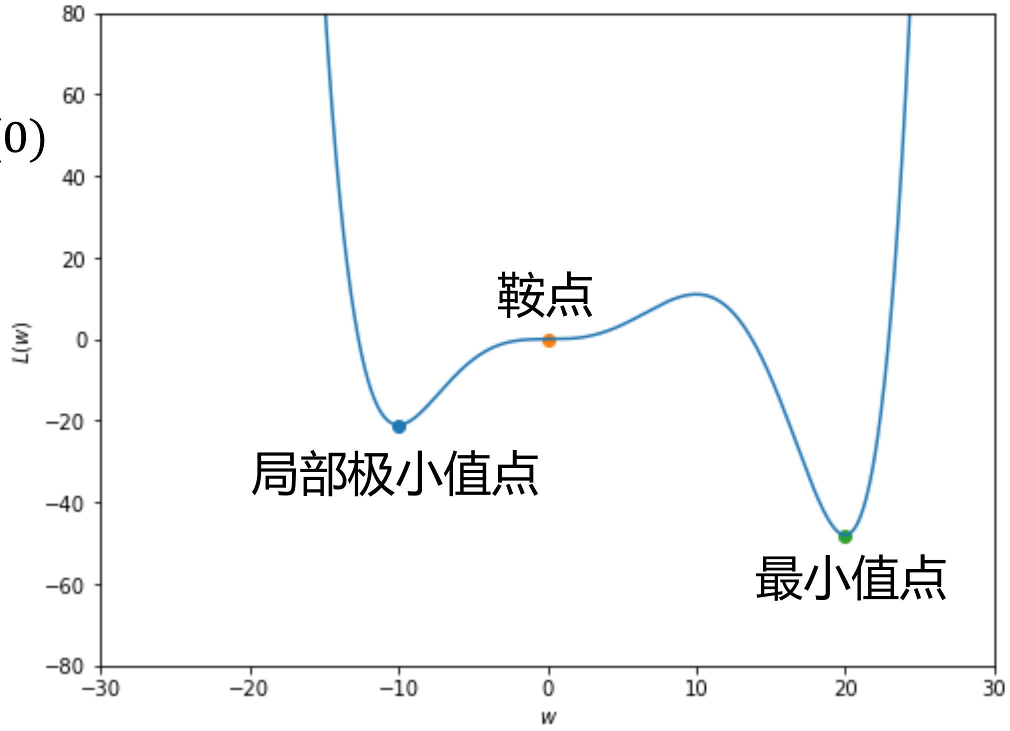
$$w^{(2)} \leftarrow w^{(1)} - \eta \frac{dL}{dw}|_{w=w^{(1)}}$$

$$\dots\dots$$

$$w^{(j+1)} \leftarrow w^{(j)} - \eta \frac{dL}{dw}|_{w=w^{(j)}}$$

$$\dots\dots$$

until $|w^{(n+1)} - w^{(n)}| < \varepsilon$, ε is called termination condition



Process of gradient descent (two parameters)

- there are two params in loss function $L(w, b)$
  - (randomly choose two init values) $b^{(0)}$, $w^{(0)}$

$$L(w, b) = \sum_{i=1}^{N} \left(y_i - w \cdot x_{i,t} - b\right)^2$$

$$\nabla L(b, w) = \begin{pmatrix} \frac{\partial L}{\partial b} \\ \frac{\partial L}{\partial w} \end{pmatrix}$$

  - compute $\frac{\partial L}{\partial w}|_{w=w^{(0)},b=b^{(0)}}$, $\frac{\partial L}{\partial b}|_{w=w^{(0)},b=b^{(0)}}$,
    update b, w

$$w^{(1)} \leftarrow w^{(0)} - \eta \frac{\partial L}{\partial w}|_{w=w^{(0)},b=b^{(0)}} \quad , \quad b^{(1)} \leftarrow b^{(0)} - \eta \frac{\partial L}{\partial b}|_{w=w^{(0)},b=b^{(0)}}$$

  - compute $\frac{\partial L}{\partial w}|_{w=w^{(1)},b=b^{(1)}}$, $\frac{\partial L}{\partial b}|_{w=w^{(1)},b=b^{(1)}}$,
    update b, w

$$w^{(2)} \leftarrow w^{(1)} - \eta \frac{\partial L}{\partial w}|_{w=w^{(1)},b=b^{(1)}} \quad , \quad b^{(2)} \leftarrow b^{(1)} - \eta \frac{\partial L}{\partial b}|_{w=w^{(1)},b=b^{(1)}}$$

…… ……

Example of gradient descent

- Calculate the gradient

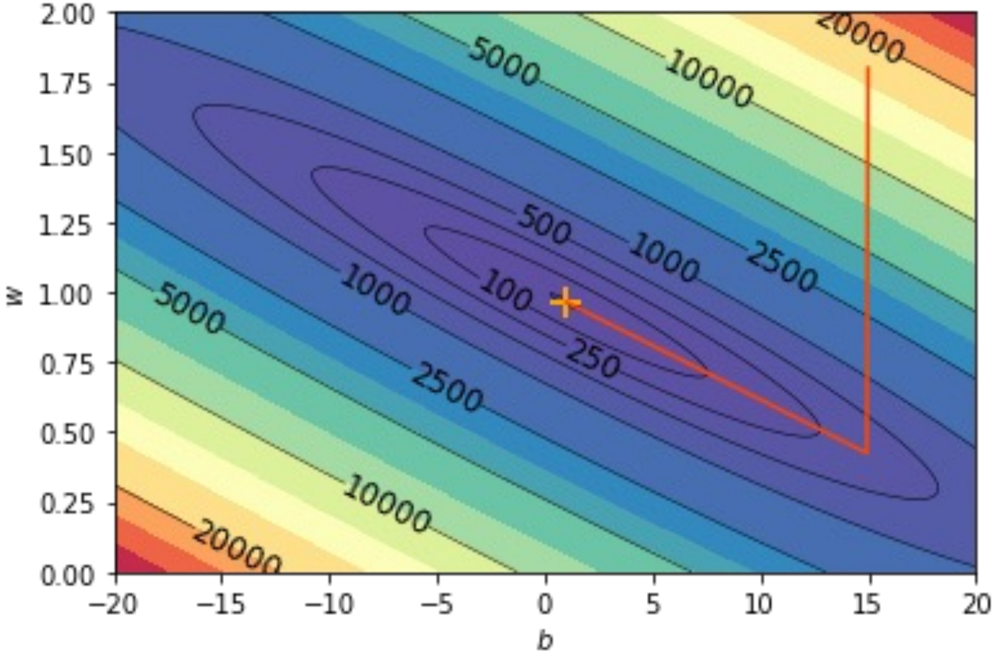
$$L(w,b) = \sum_{i=1}^{n} \left(y_i \underline{- w \cdot x_{i,t}} \underline{- b}\right)^2$$

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{n} 2\left(y_i - w \cdot x_{i,t} - b\right)\left(-x_{i,t}\right)$$

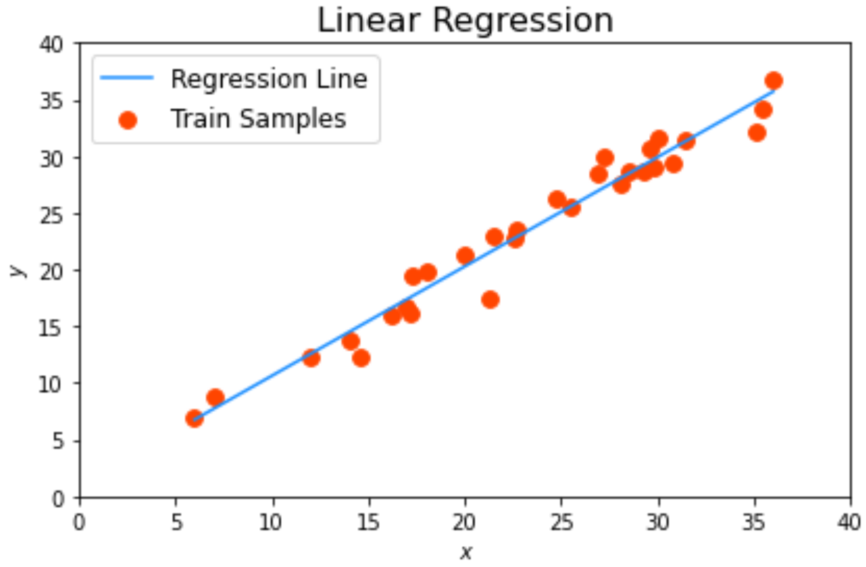
$$\frac{\partial L}{\partial b} = \sum_{i=1}^{n} 2\left(y_i - w \cdot x_{i,t} - b\right)(-1)$$

Example of gradient descent

- Look for minima values in the direction of gradient descent



$w^* = 0.965, b^* = 0.980$

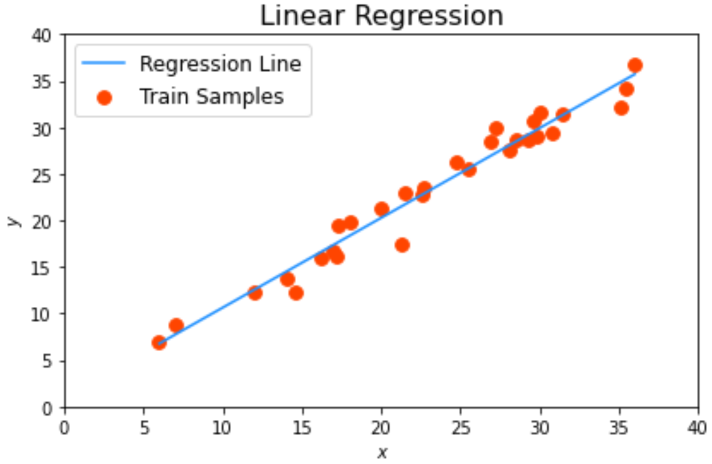


$y = 0.965x_t + 0.980$

Test model effects (univariate linear regression)

- MSE of training set

$$\frac{1}{30}\sum_{i=1}^{30}\left(y_i - w^{*}\cdot x_{i,t} - b^{*}\right)^2 = 2.134$$



- randomly select ten days of data from testing dataas a testing set to test the generalization performance of the model

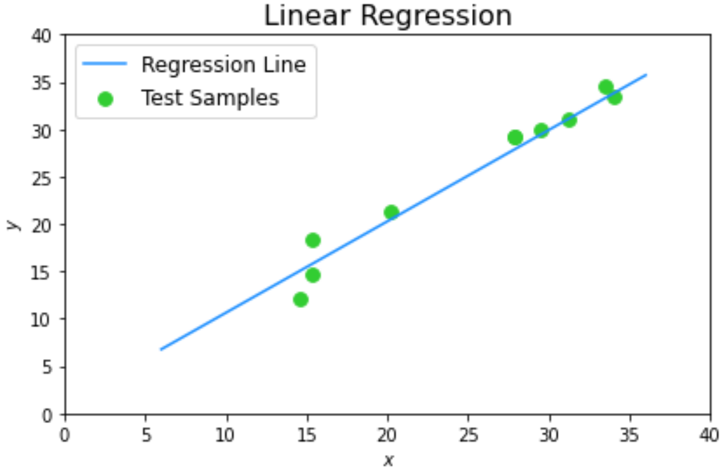
$$w^{*} = 0.965, b^{*} = 0.980$$

  - MSE of testing set

$$\frac{1}{10}\sum_{i=1}^{10}\left(y_i - w^{*}\cdot x_{i,t} - b^{*}\right)^2 = 2.294$$

  - mean error of testing set

$$\frac{1}{10}\sum_{i=1}^{10}\left|y_i - w^{*}\cdot x_{i,t} - b^{*}\right| = 1.229$$



Improved model: add quadratic term features

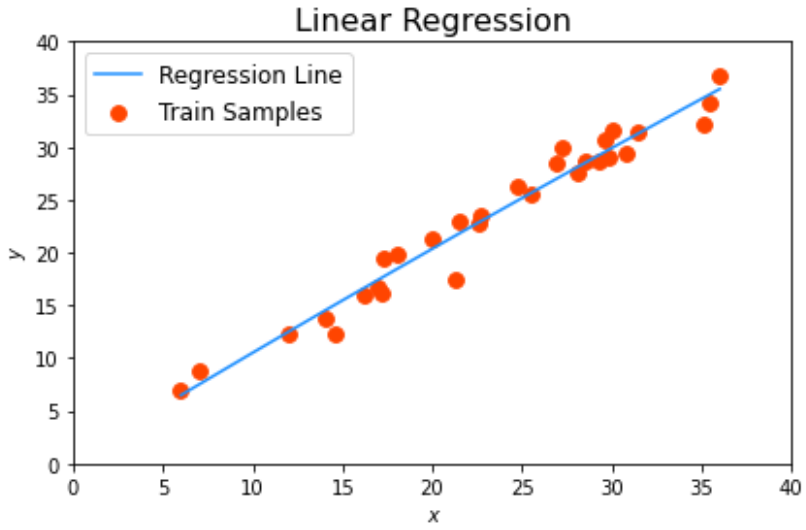
- regression model :

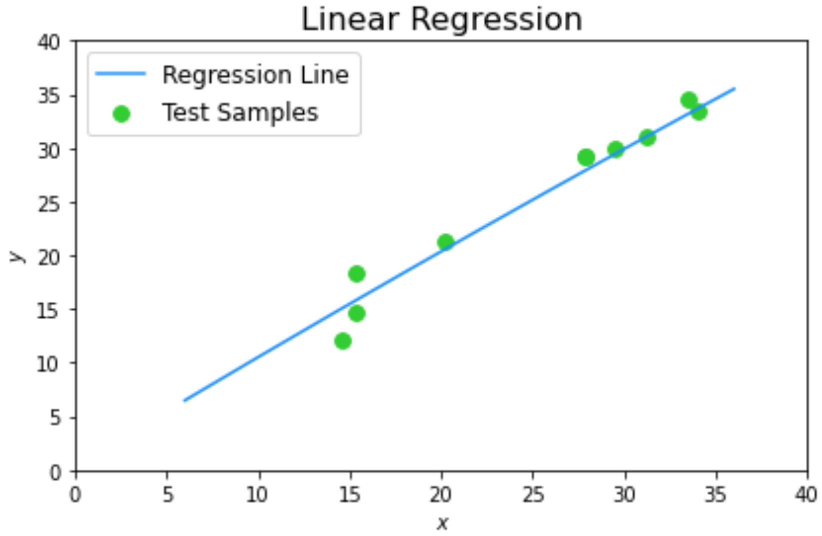
$$y = w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

- calculate the params, we get:

$$w_2^* = -1.50\times10^{-3}, w_1^* = 1.030, b^* = 0.361$$

- error :
  - MSE of training set : 2.123<2.134
  - MSE of testing set : 2.278<2.294





Improved model: add cubic term feature

- regression model :

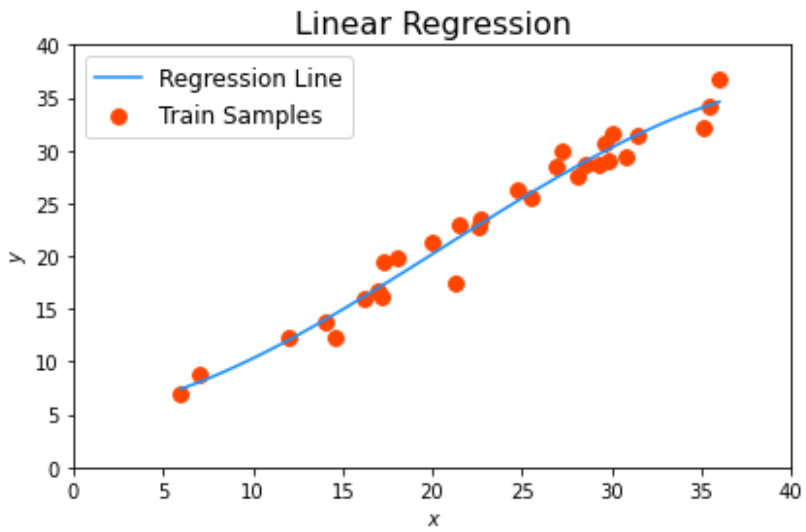
$$y = w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

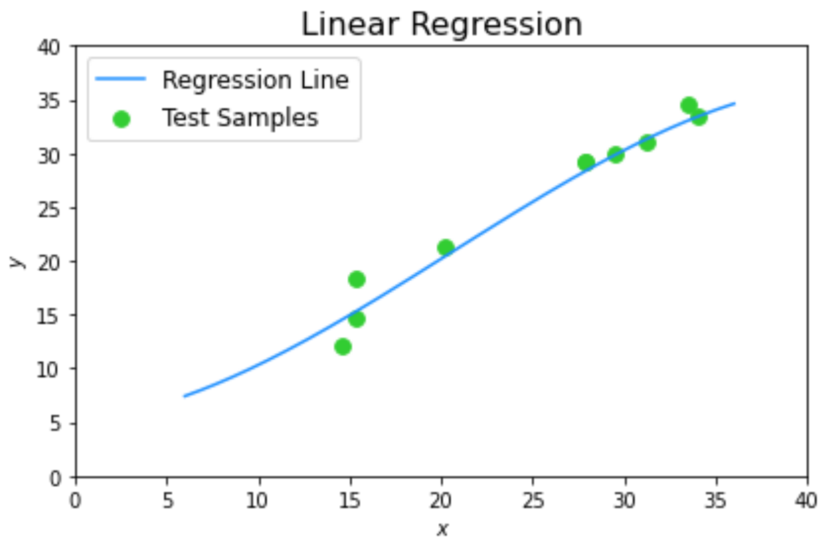
- calculate the params, we get:

$$w_3^* = -7.43\times10^{-4}, w_2^* = 0.046,$$

$$w_1^* = 0.136, b^* = 5.123$$

- error :
  - MSE of training set : 1.913<2.123
  - MSE of testing set : 2.042<2.278





Improved model: add quartic term feature

- regression model :

$$y = w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

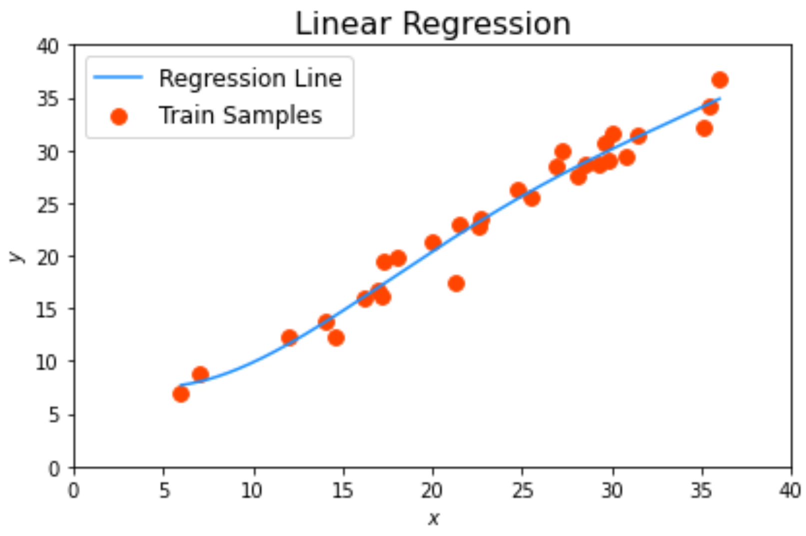
- calculate the params, we get:

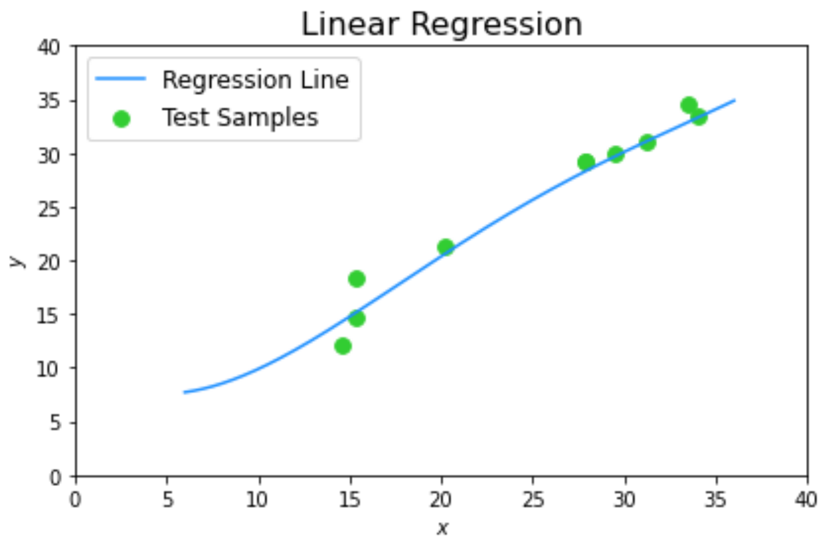
$$w_4^* = 4.75\times10^{-5},\ w_3^* = -4.83\times10^{-3},$$

$$w_2^* = 0.167, w_1^* = -1.290,\ b^* = 10.43$$

- error :
  - MSE of training set : 1.878<1.913
  - MSE of testing set : 2.053>2.042

over-fitting





Improved model: add quintic term feature

- regression model :

$$y = w_5 \cdot x_t^5 + w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

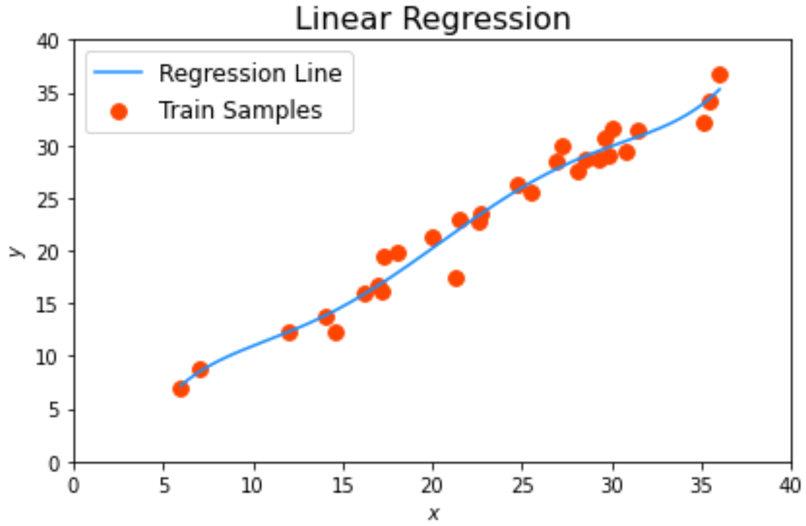
- calculate the params, we get:

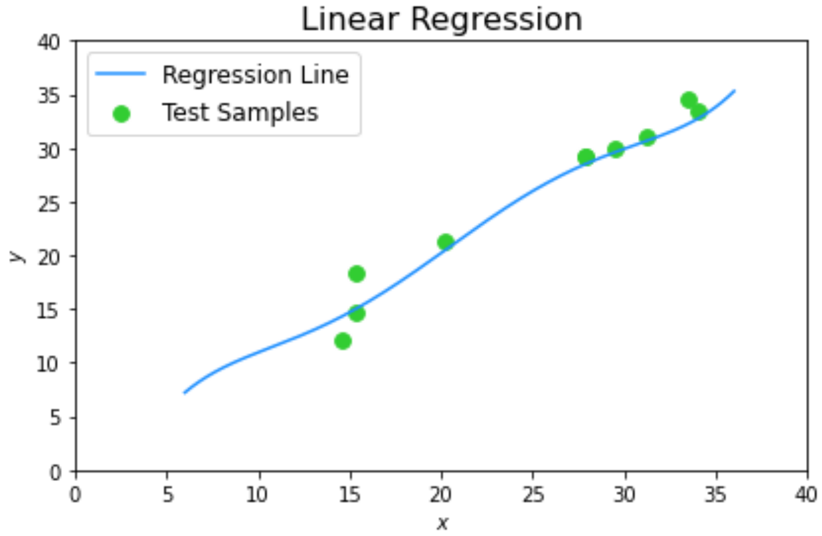
$$w_5^* = 1.184\times10^{-5}, w_4^* = -1.22\times10^{-3}, w_3^* = 4.64\times10^{-2},$$

$$w_2^* = -0.080, w_1^* = 6.948,\ b^* = -14.37$$

- error :
  - MSE of training set : 1.797<1.878
  - MSE of testing set : 2.396>2.053

over-fitting





over-fitting

- complex models can better fit the training data
- However, it may not be better on the test data

$$y = w \cdot x_t + b$$

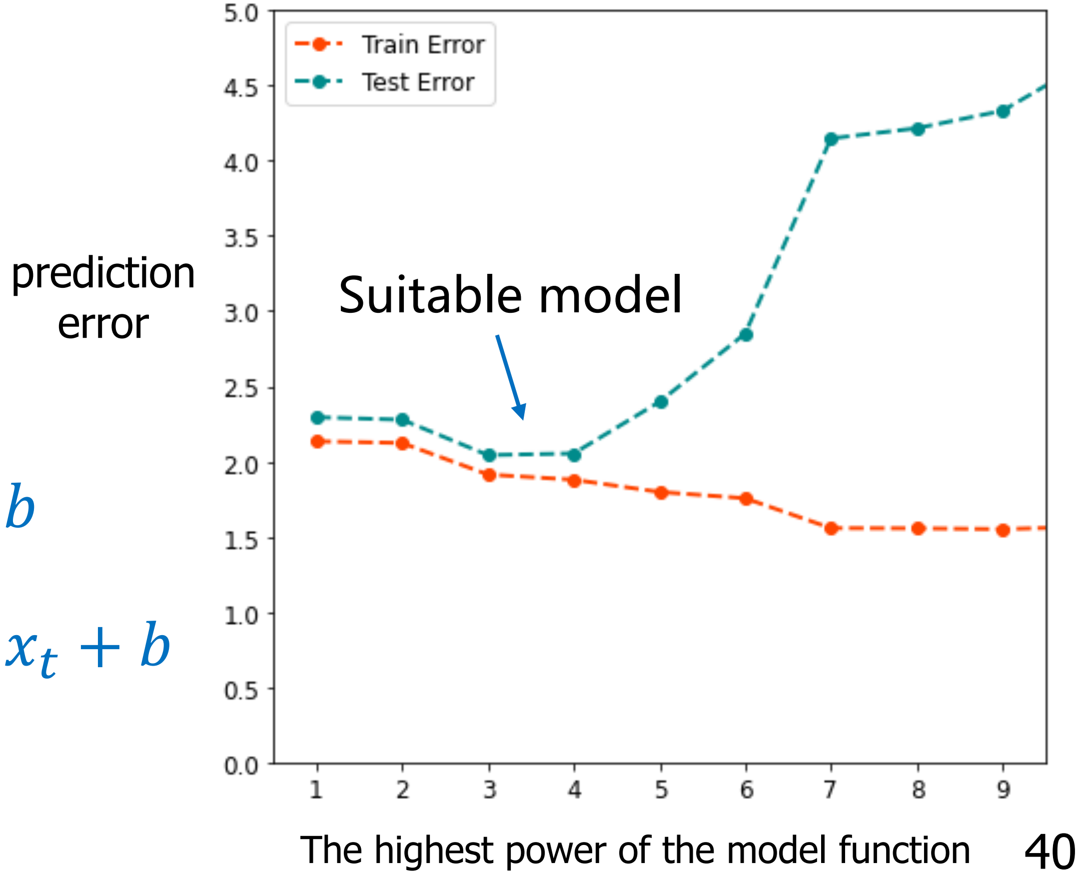
$$y = w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

$$y = w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

$$y = w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

$$y = w_5 \cdot x_t^5 + w_4 \cdot x_t^4 + w_3 \cdot x_t^3 + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

…… ……



Multivariate linear regression

- Go back to step one: determine model space
  - In addition to the average daily temperature of the previous day, consider whether it is related to the relative humidity, wind speed and air pressure of the previous day

$$y = w_1 \cdot x_t + w_2 \cdot x_t^2 + w_3 \cdot x_t^3 + w_4 \cdot x_h + w_5 \cdot x_w + w_6 \cdot x_p + b$$

linear regression model：

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + \mathrm{b}$$

data：$(\boldsymbol{x_i}, y_i)$

日均气温 (mean temp)	相对湿度 (humidity)	风速 (wind speed)	气压 (pressure)
7.40	92.00	2.980	1017.80

$$\boldsymbol{x_i} = (x_{i,t}, x_{i,h}, x_{i,w}, x_{i,p})^T$$

Add additional features

- regression model :

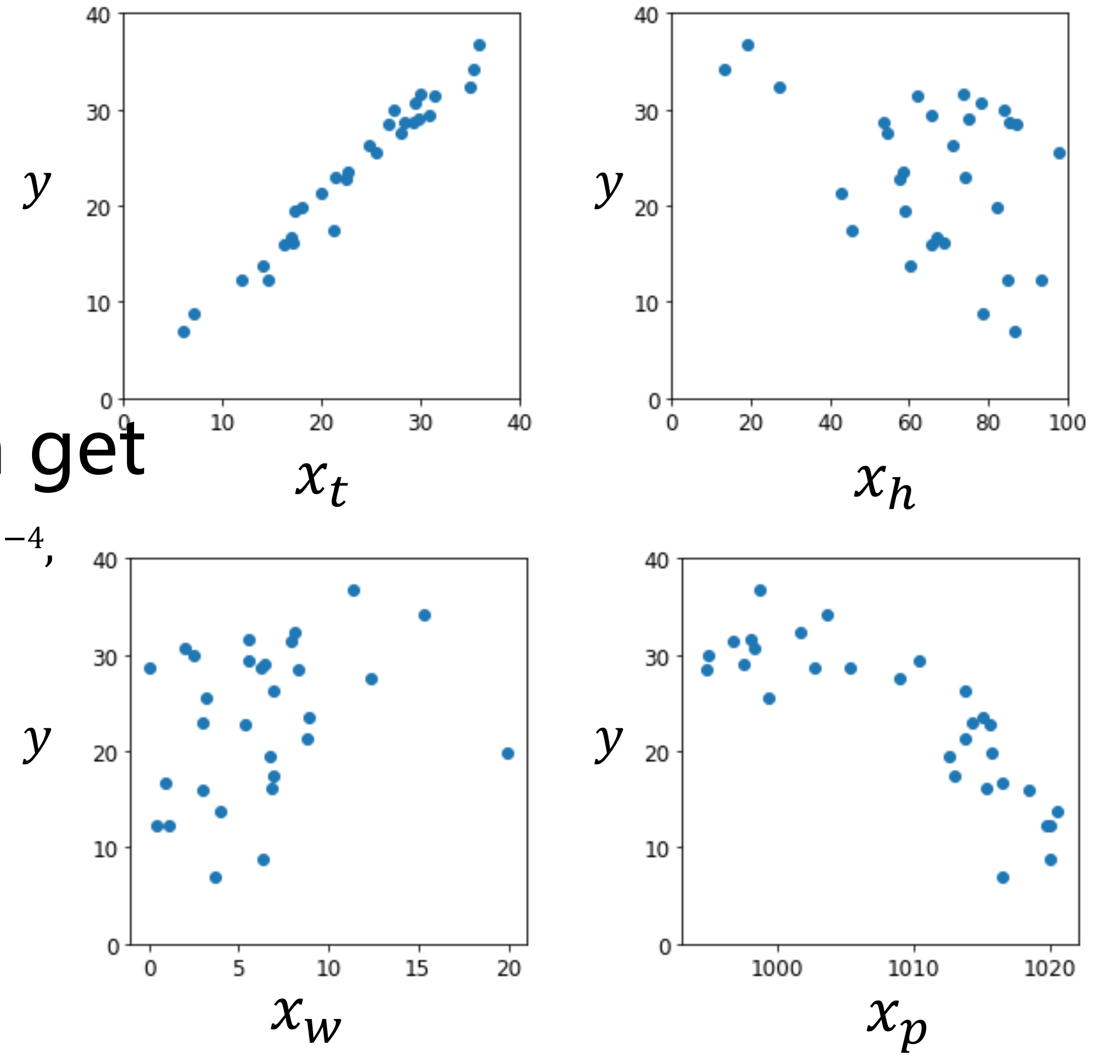
$$y = w_1 \cdot x_t + w_2 \cdot x_t^2 + w_3 \cdot x_t^3$$
$$+ w_4 \cdot x_h + w_5 \cdot x_w + w_6 \cdot x_p + b$$

- calculate the params and we can get

$w_6^* = -0.011, w_5^* = 0.010, w_4^* = 1.18\times10^{-2}, w_3^* = -2.58\times10^{-4},$
$w_2^* = 1.39\times10^{-2}, w_1^* = 0.667,\ b^* = 109.5$

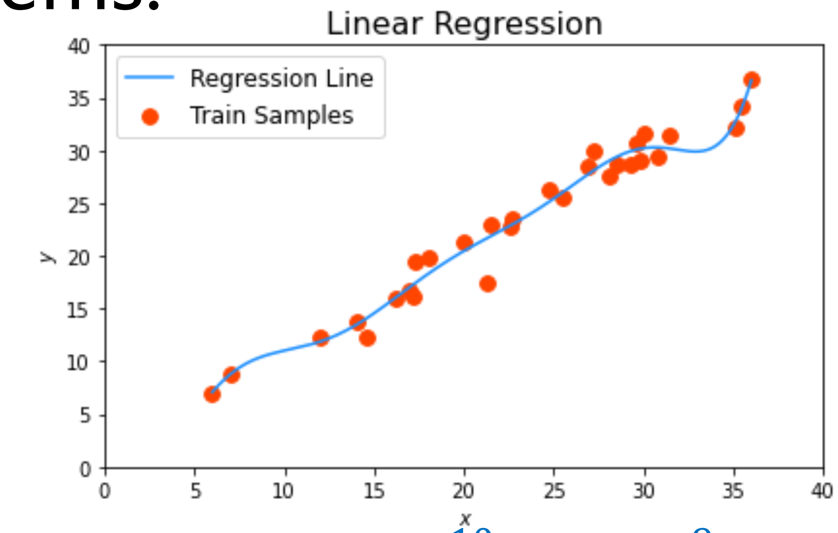
- error :
  - MSE of training set : 1.553<1.913
  - MSE of testing set : 2.278>2.042

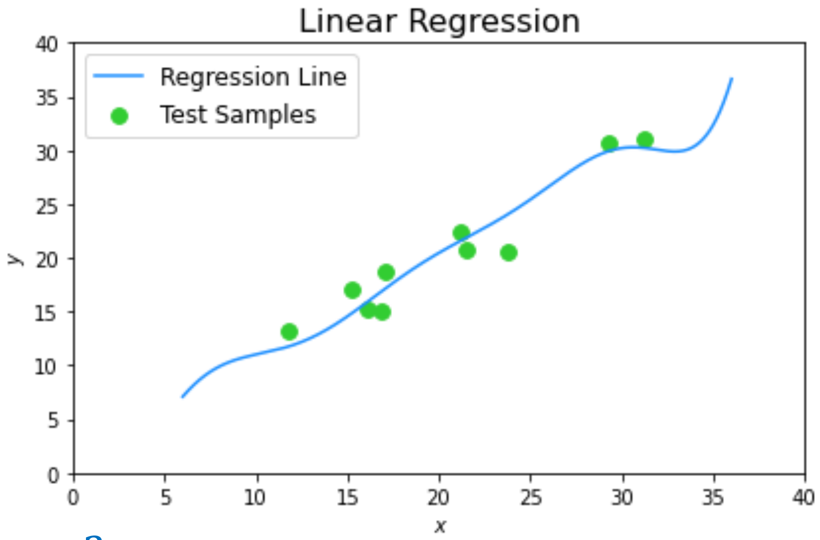
over-fitting



The basic idea of regularization

- Occam's Razor
  - choose a simple model that explains known data well.
- Simple functions are smoother and less prone to fitting problems.





$$y = w_{10} \cdot x_t^{10} + w_9 \cdot x_t^9 + \cdots + w_2 \cdot x_t^2 + w_1 \cdot x_t + b$$

regularization

- thinking of a linear regression model with d features

$$y = \sum_{i=1}^{d} w_j x_j + b$$

- loss function is :

$$L(w,b) = \sum_{i=1}^{n}\left(y_i - \sum_{i=1}^{d} w_j x_{i,j} - b\right)^2$$

λ is a hyperparameter, the larger the value of λ,
The more resistant the model is to disturbances

- loss function with regularization item :
  - L1 regularization

$$L(w,b) = \sum_{i=1}^{n}\left(y_i - \sum_{i=1}^{d} w_j x_{i,j} - b\right)^2 + \boxed{\lambda \sum_{j=1}^{d} \left|w_j\right|}$$

  - L2 regularization

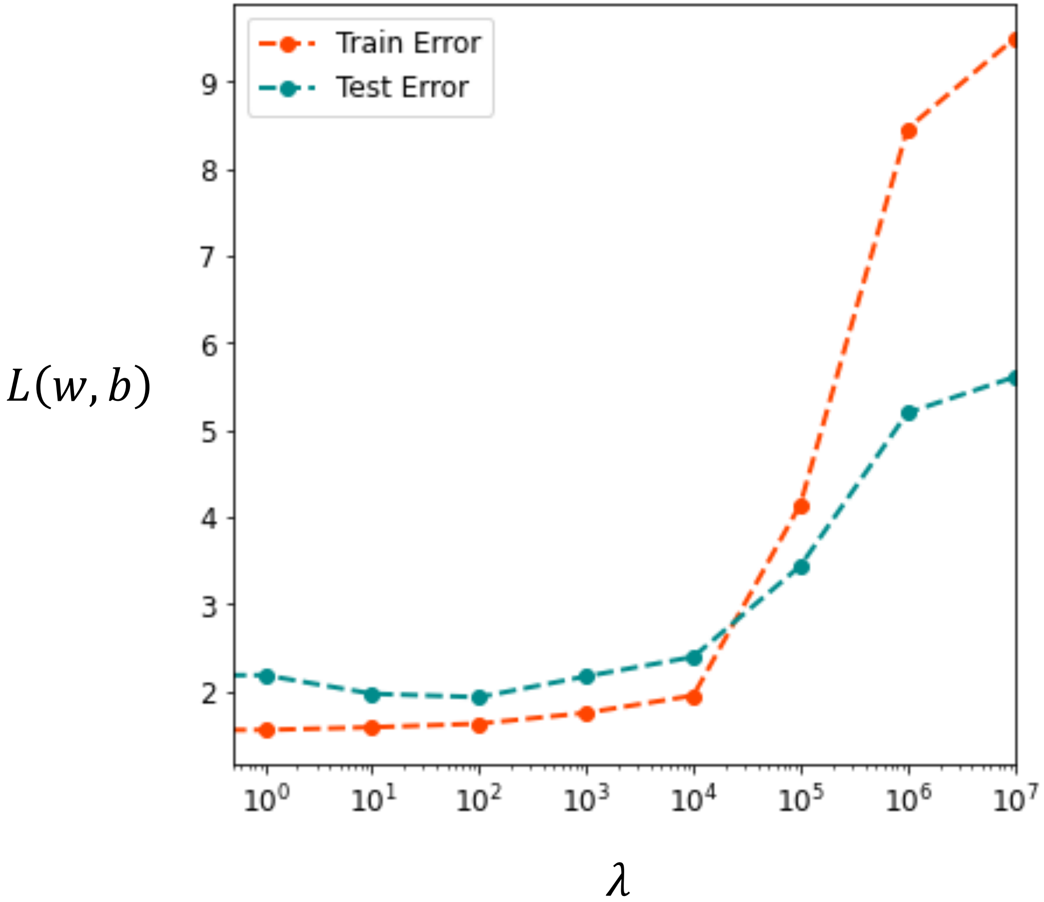
$$L(w,b) = \sum_{i=1}^{n}\left(y_i - \sum_{i=1}^{d} w_j x_{i,j} - b\right)^2 + \boxed{\lambda \sum_{j=1}^{d} \left|w_j\right|^2}$$

regularization

- thinking of model :

$$y = w_1 \cdot x_t + w_2 \cdot x_t^2 + w_3 \cdot x_t^3 + w_4 \cdot x_h + w_5 \cdot x_w + w_6 \cdot x_p + b$$

- add L2 regularization term to loss function
  - when $\lambda = 0$:
    - MSE of training set : 1.553
    - MSE of testing set : 2.278
  - when $\lambda = 100$:
    - MSE of training set : 1.627
    - MSE of testing set : 1.932



3.5 Logistic Regression: a Brief Review

How about Employing Linear Regression for Classification Task?

- Given $\mathbb{X} = \{(\boldsymbol{x_1}, c_1), (\boldsymbol{x_2}, c_2), \dots, (\boldsymbol{x_n}, c_n)\}$,

$$c_i \in \{class1,\ class2\}$$

- Translating classification problem into regression problem as:

$$\hat{y}_i = \begin{cases} 1, & if\ \ c_i = class1 \\ -1, & if\ \ c_i = class2 \end{cases}$$

- we get the transformed data set $(\boldsymbol{x_i}, \hat{y}_i)$ for training linear regression model:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + \mathrm{b}$$

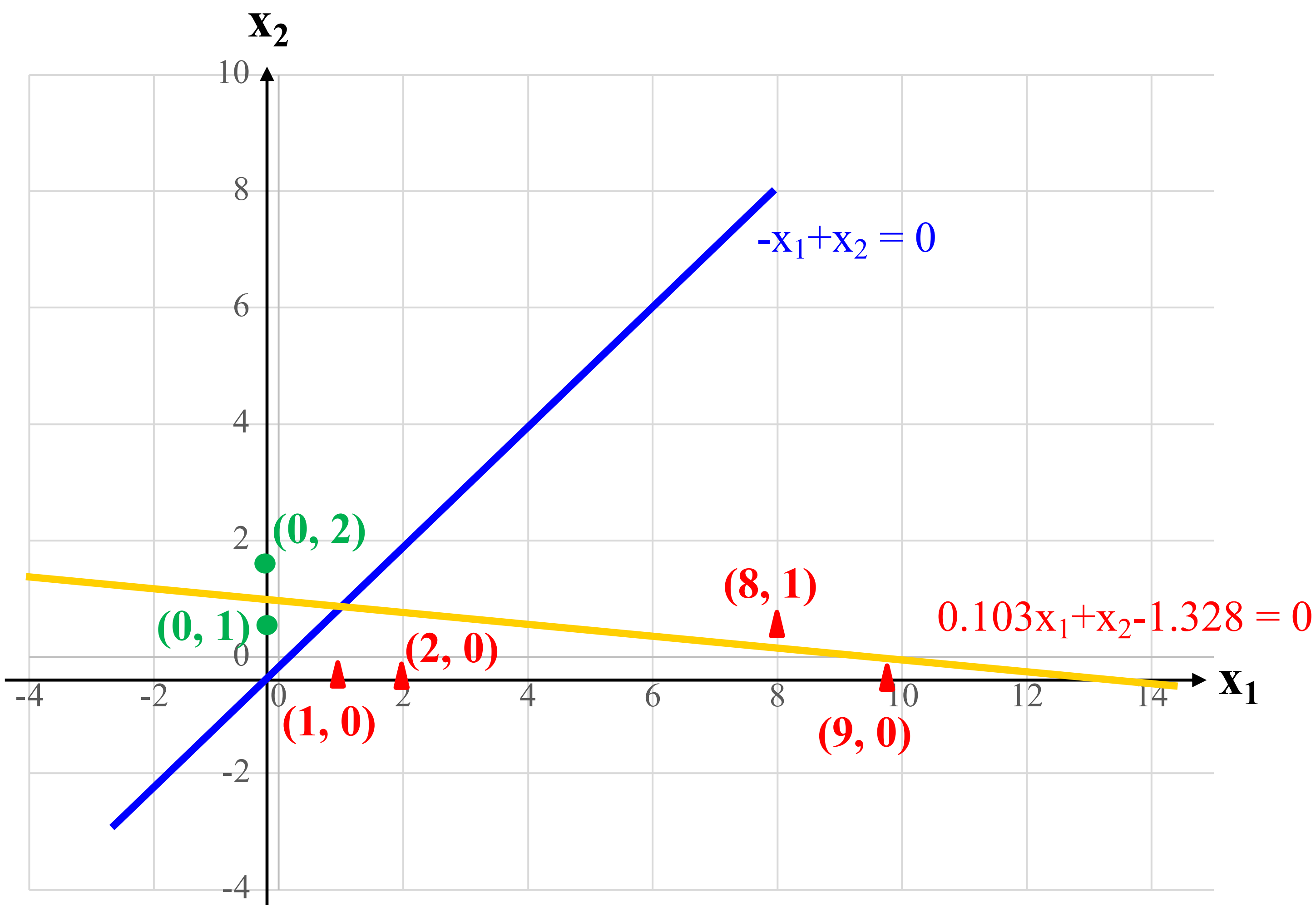
- Using $f(\boldsymbol{x_j}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_j} + \mathrm{b}$ we get the prediction of $\boldsymbol{x_j}$, and do classification:

$$\hat{c}_j = \begin{cases} class1, & if\ \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_j} + \mathrm{b} \geq 0 \\ class2, & if\ \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_j} + \mathrm{b} < 0 \end{cases}$$

How about Employing Linear Regression for Classification Task? (cont.)

$f(\boldsymbol{x_j}) = \boldsymbol{w}^T \boldsymbol{x_j} + \mathrm{b}$

$\boldsymbol{x_j}$	$\hat{y}_i$
(0, 2)	**+1**
(0, 1)	**+1**
(1, 0)	**-1**
(2, 0)	**-1**
(8, 1)	**-1**
(9, 0)	**-1**



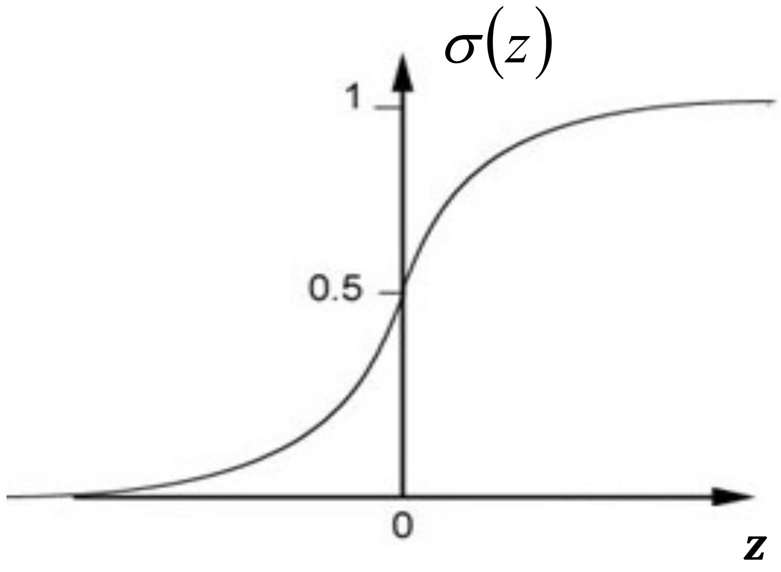
Logistic Regression：model definition

- Two-class classification as probability calculation:

$$P\,(C_i|x),\ \ i = 1{,}2$$

$$P\,(C_1|x) = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$z = w^T x + b\ = \sum_j w_j x_j + b$$



- Given x_j， the classification decision：

$$\hat{c}_j = \begin{cases} class1, & if\ \sigma(z) \geq 0.5 \quad \boldsymbol{z} \geq 0 \\ class2, & if\ \sigma(z) < 0.5 \quad \boldsymbol{z} < 0 \end{cases}$$

Model parameter：**w、b**

Logistic Regression : model goodness

Training data :

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$	……	$\boldsymbol{x_n}$
C_1	C_1	C_2		C_1

Assume the distribution of training data: $f_{w,b}(\boldsymbol{x}) = P(C_1|\boldsymbol{x}) = \frac{1}{1+e^{-z}}$

Given < $\boldsymbol{w}$, b > , the likelihood can be computed as :

$$L(w,b) = f_{w,b}(\boldsymbol{x_1}) f_{w,b}(\boldsymbol{x_2}) \left(1 - f_{w,b}(\boldsymbol{x_3})\right) \cdots f_{w,b}(\boldsymbol{x_n})$$

then , the best < $\boldsymbol{w^*}$, b^* > should be:

$$w^*, b^* = arg \max_{w,b} L(w,b)$$

$$\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \boldsymbol{x}_3 \quad \dots\dots$$
$$C_1 \quad C_1 \quad C_2$$

$$\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \boldsymbol{x}_3 \quad \dots\dots$$
$$\hat{y}_1 = 1 \quad \hat{y}_2 = 1 \quad \hat{y}_3 = 0$$

$\hat{y}_i$: 1 for C_1, 0 for C_2

$$L(w,b) = f_{w,b}(\boldsymbol{x}_1) f_{w,b}(\boldsymbol{x}_2) \left(1 - f_{w,b}(\boldsymbol{x}_3)\right) \cdots$$

$$\boxed{w^*, b^* = arg \max_{w,b} L(w,b)} = \boxed{w^*, b^* = arg \min_{w,b} -lnL(w,b)}$$

$$\begin{aligned}
&-lnL(w,b) \\
&= -lnf_{w,b}(\boldsymbol{x}_1) \Rightarrow -[\boxed{1}\, lnf(\boldsymbol{x}_1) \cancel{+ \boxed{0}\, ln(1 - f(\boldsymbol{x}_1))}] \\
&\quad -lnf_{w,b}(\boldsymbol{x}_2) \Rightarrow -[\boxed{1}\, lnf(\boldsymbol{x}_2) \cancel{+ \boxed{0}\, ln(1 - f(\boldsymbol{x}_2))}] \\
&-ln\left(1 - f_{w,b}(\boldsymbol{x}_3)\right) \Rightarrow -[\cancel{\boxed{0}\, lnf(\boldsymbol{x}_3) +} \boxed{1}\, ln(1 - f(\boldsymbol{x}_3))] \\
&\qquad \vdots \quad \vdots
\end{aligned}$$

$$L(w,b) = f_{w,b}(\boldsymbol{x_1})f_{w,b}(\boldsymbol{x_2})\left(1 - f_{w,b}(\boldsymbol{x_3})\right)\cdots f_{w,b}(\boldsymbol{x_n})$$

$$-lnL(w,b) = lnf_{w,b}(\boldsymbol{x_1}) + lnf_{w,b}(\boldsymbol{x_2}) + ln\left(1 - f_{w,b}(\boldsymbol{x_3})\right)\cdots$$

$\hat{y}_i$: 1 for class 1, 0 for class 2

$$= \sum_i \underbrace{-\left[\hat{y}_i lnf_{w,b}(\boldsymbol{x_i}) + (1-\hat{y}_i)ln\left(1 - f_{w,b}(\boldsymbol{x_i})\right)\right]}_{\text{Cross entropy between two Bernoulli distribution}}$$

Distribution p:

$p(x = 1) = \hat{y}_i$

$p(x = 0) = 1 - \hat{y}_i$

cross entropy

Distribution q:

$q(x = 1) = f(\boldsymbol{x_i})$

$q(x = 0) = 1 - f(\boldsymbol{x_i})$

$$H(p,q) = -\sum_x p(x)ln\left(q(x)\right)$$

Logistic Regression : optimization

$$w^*, b^* = arg\min_{w,b} -lnL(w,b)$$

$$= \sum_i -\left[\hat{y}_i ln f_{w,b}(\boldsymbol{x_i}) + (1-\hat{y}_i) ln\left(1 - f_{w,b}(\boldsymbol{x_i})\right)\right]$$

- **With gradient descend**:

$$f_{w,b} = \frac{1}{1+e^{-z}} \quad z = w^T x + b$$

Gradient update: $w_k \leftarrow w_k - \eta \sum_i -\left(\hat{y}_i - f_{w,b}(\boldsymbol{x_i})\right) x_{ik}$

Compared to linear regression :

$$w_k \leftarrow w_k - \eta \sum_i -\left(\hat{y}_i - f_{w,b}(\boldsymbol{x_i})\right) x_{ik}$$

$$f_{w,b} = w^T x + b$$

Logistic Regression for multi-class : softmax

C_1: w^1, b_1 $\quad z_1 = w^1 \cdot x + b_1$

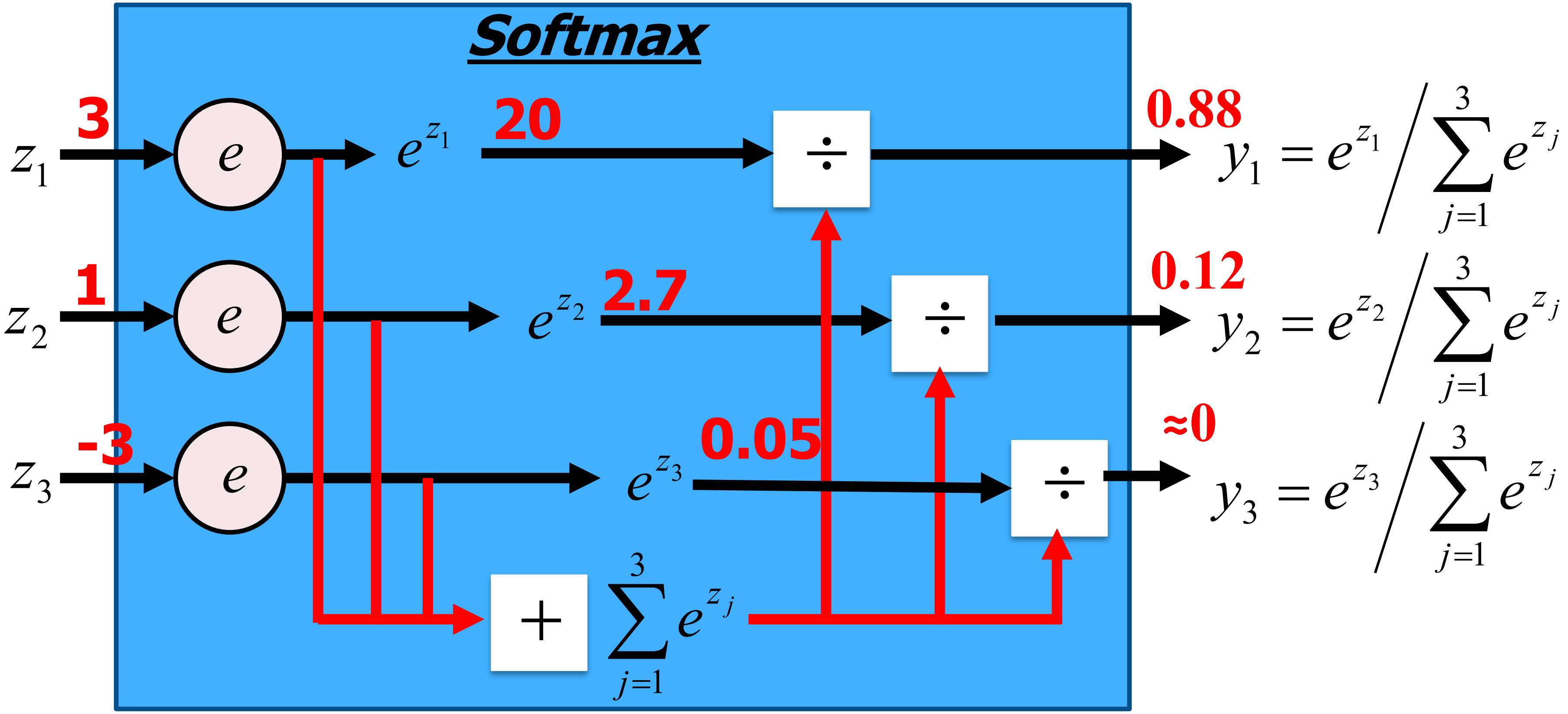
C_2: w^2, b_2 $\quad z_2 = w^2 \cdot x + b_2$

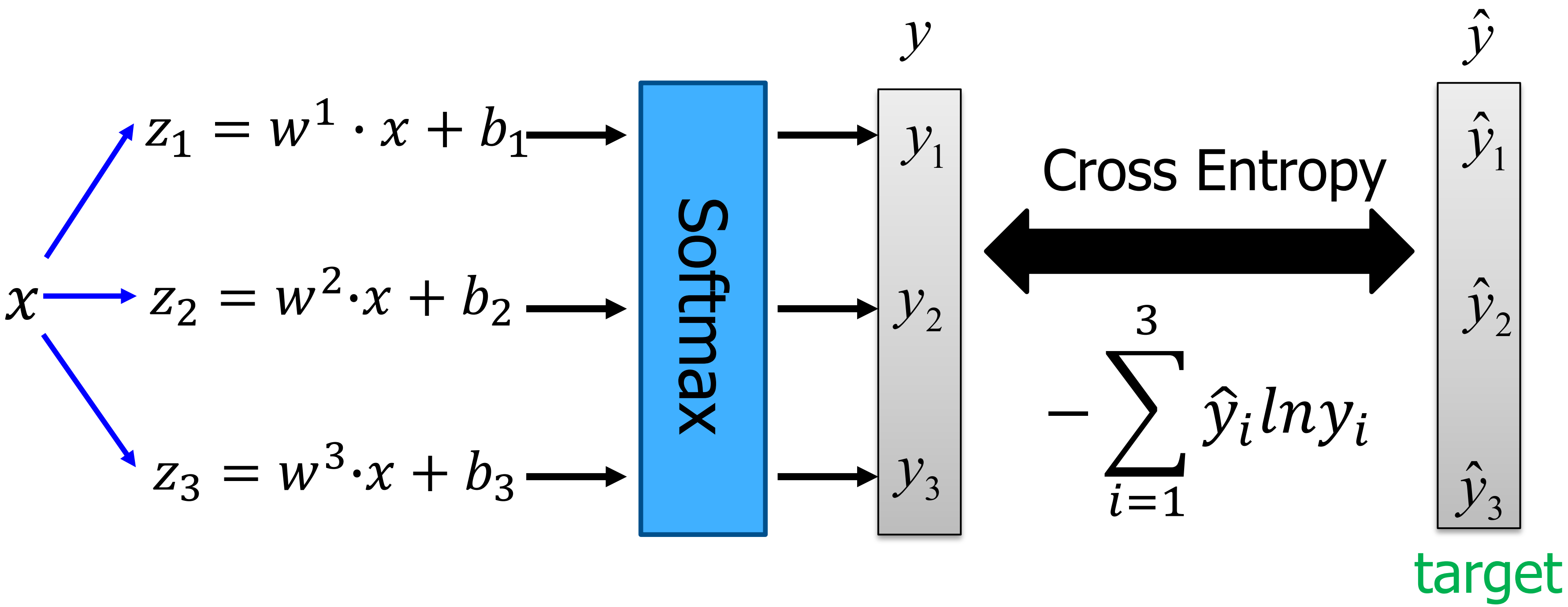
C_3: w^3, b_3 $\quad z_3 = w^3 \cdot x + b_3$

Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$

$$y_i = P(C_i \mid x)$$





If x ∈ class 1

$$\hat{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$-\ln y_1$$

If x ∈ class 2

$$\hat{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$-\ln y_2$$

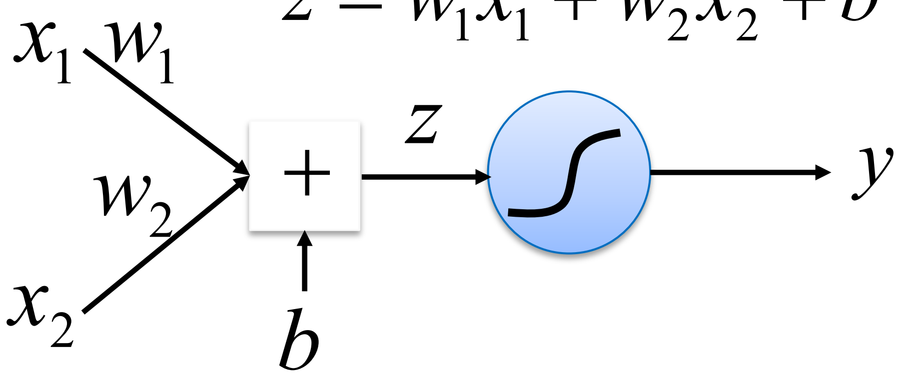
If x ∈ class 3

$$\hat{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$-\ln y_3$$

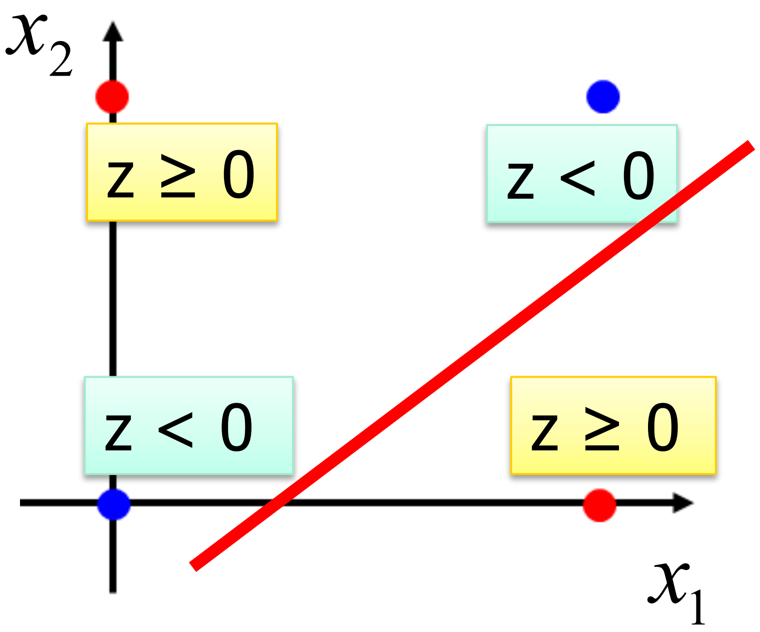
Limitation of Logistic Regression

$$z = w_1 x_1 + w_2 x_2 + b$$



$$\begin{cases} Class1 & y \geq 0.5 \quad (z \geq 0) \\ Class2 & y < 0.5 \quad (z < 0) \end{cases}$$

Input Feature		Label
x_1	x_2	
0	0	Class 2
0	1	Class 1
1	0	Class 1
1	1	Class 2

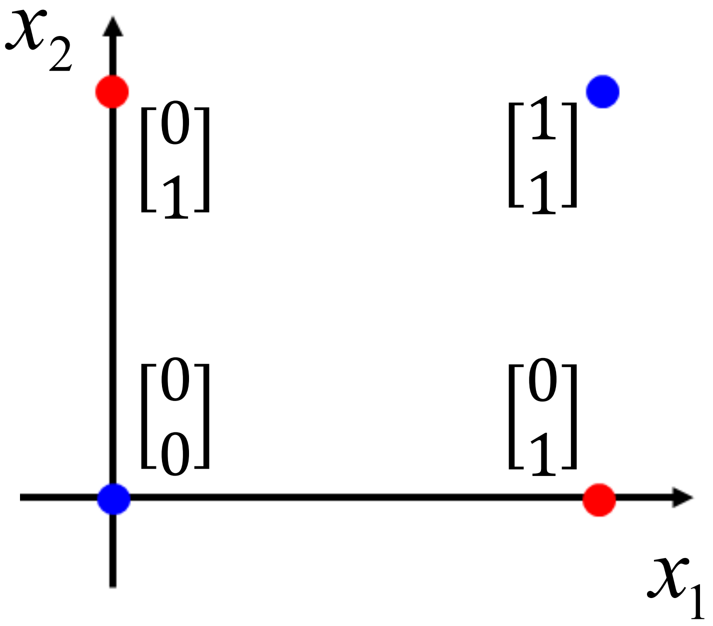


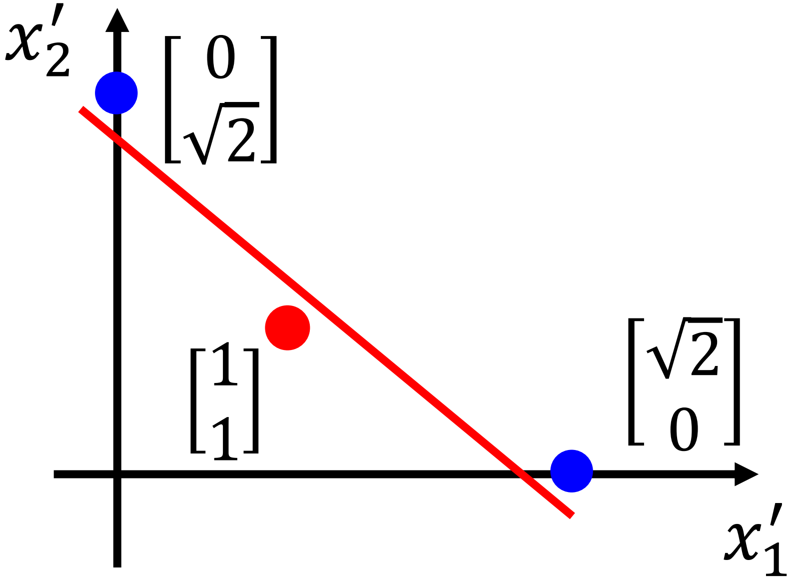
- ***Feature transformation***

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \longrightarrow \begin{bmatrix} x_1' \\ x_2' \end{bmatrix}$$

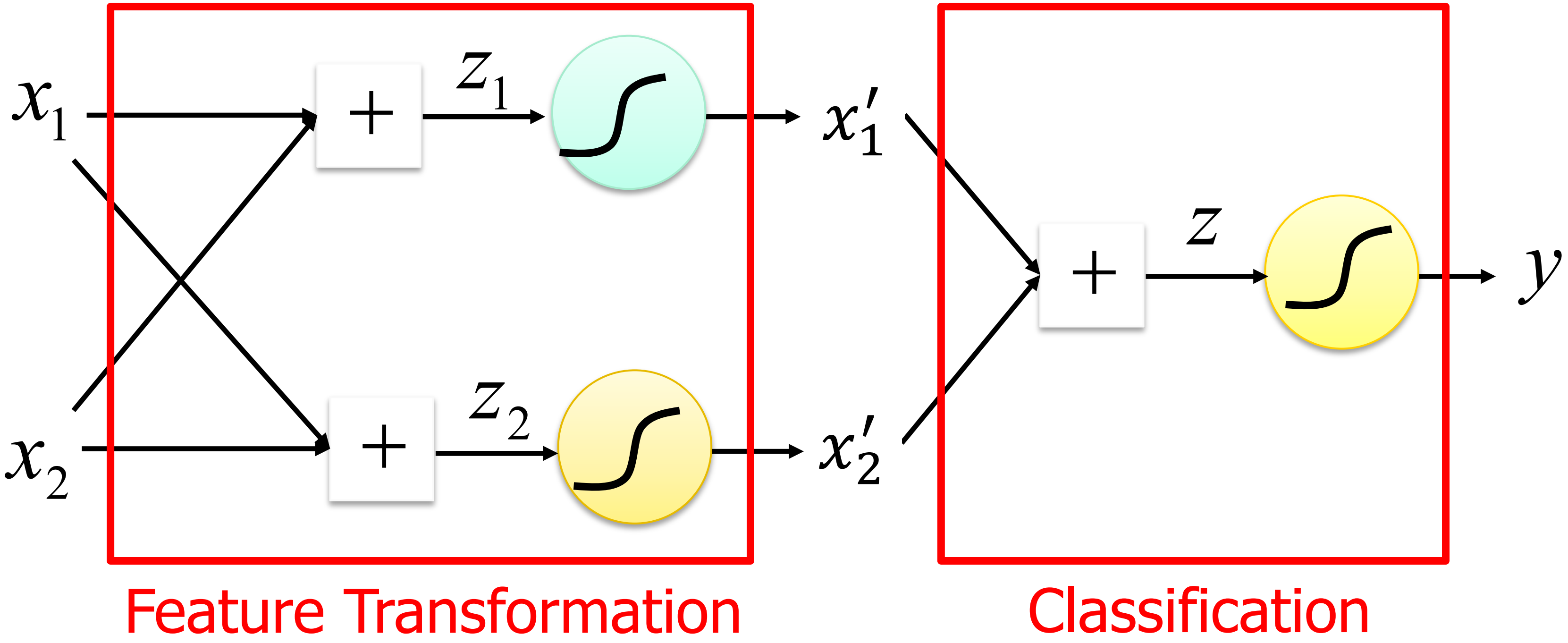
x_1': $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ distance to $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

x_2': $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ distance to $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$





- Cascading logistic regression models



Acknowledgements

- Most slides of this section are from Dr Hongyi Lee and Internet.
- This lecture is distributed for nonprofit purpose.