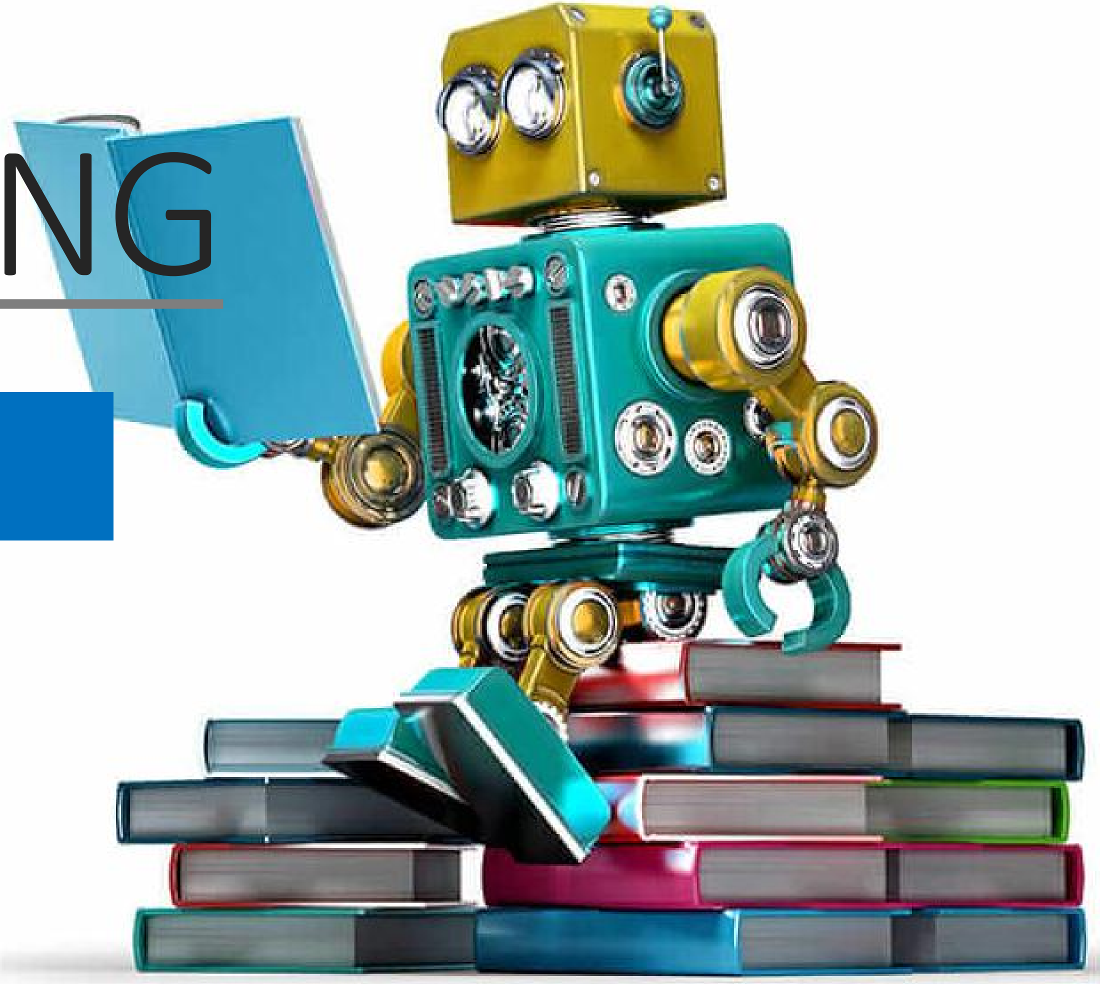
MACHINE LEARNING

LAB4 Linear Regression



贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn

OBJECTIVES

01 Simple Linear Regression

02 Evaluation

03 Polynomial Regression

Outline

- Simple Linear Regression:
  - ✓ attributes and target, least square method
- Evaluation
  - ✓ MSE, RMSE, MAE, R-Squared.
- Polynomial regression
  - ✓ pipeline, bias & variance, regularization

What is regression?

- A method to determine the statistical relationship between a dependent variable and one or more independent variables.

- When two or more independent variables are used to predict or explain the outcome of the dependent variable, this is known as multiple regression.

- This can be broadly classified into two major types.
  ① Linear Regression
  ② Logistic Regression

Linear regression

The general form of regression is:

- Simple linear regression:

$$y = \theta_0 + \theta_1 x$$

- Multiple linear regression:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$

Where,

x: Independent Variable

y: Dependent Variable

θ_1 : Slope of Line

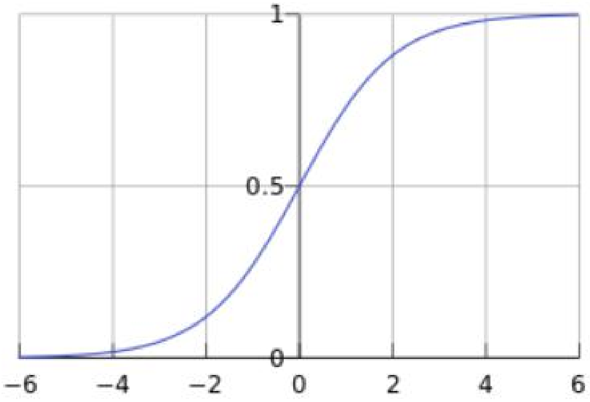
θ_0 : y Intercept

Logistic regression

Logistic regression is a classification algorithm. It is used to predict a binary outcome based on a set of independent variables.

Logistic\Sigmoid function:

$$h_\theta(y) = \frac{1}{1 + e^{-\theta \quad y}}$$



Linear Regression vs Logistic regression

Logistic Regression(这里特指回归操作):

$$f(x) = sigmoid(w^T x + b)$$

Logistic Classifier:

$$y = \begin{cases} 1 & f(x) \geq 0.5 \\ -1 & f(x) < 0.5 \end{cases}$$

Linear Regression:

$$f(x) = w^T x + b$$

Linear Classifier:

$$y = \begin{cases} 1 & f(x) \geq 0 \\ -1 & f(x) < 0 \end{cases}$$

Attributes and Targets（Linear Regression）

- Attributes: measuring aspects of a sample instance
- Targets: **continuous** value of the indicator we concern
  - ✓ example: Boston house prices (embedded dataset in scikit-learn)

	Attributes													Target
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	Price
0	9.91655	0.0	18.10	0.0	0.6930	5.852	77.8	1.5004	24.0	666.0	20.2	338.16	29.97	6.3
1	3.83684	0.0	18.10	0.0	0.7700	6.251	91.1	2.2955	24.0	666.0	20.2	350.65	14.19	19.9
2	0.43571	0.0	10.59	1.0	0.4890	5.344	100.0	3.8750	4.0	277.0	18.6	396.90	23.09	20.0
3	0.03150	95.0	1.47	0.0	0.4030	6.975	15.3	7.6534	3.0	402.0	17.0	396.90	4.56	34.9
4	0.62976	0.0	8.14	0.0	0.5380	5.949	61.8	4.7075	4.0	307.0	21.0	396.90	8.26	20.4
5	5.87205	0.0	18.10	0.0	0.6930	6.405	96.0	1.6768	24.0	666.0	20.2	396.90	19.37	12.5
6	2.37857	0.0	18.10	0.0	0.5830	5.871	41.9	3.7240	24.0	666.0	20.2	370.73	13.34	20.6
7	0.03578	20.0	3.33	0.0	0.4429	7.820	64.5	4.6947	5.0	216.0	14.9	387.31	3.76	45.4
8	5.70818	0.0	18.10	0.0	0.5320	6.750	74.9	3.3317	24.0	666.0	20.2	393.07	7.74	23.7
9	0.46296	0.0	6.20	0.0	0.5040	7.412	76.9	3.6715	8.0	307.0	17.4	376.14	5.25	31.7

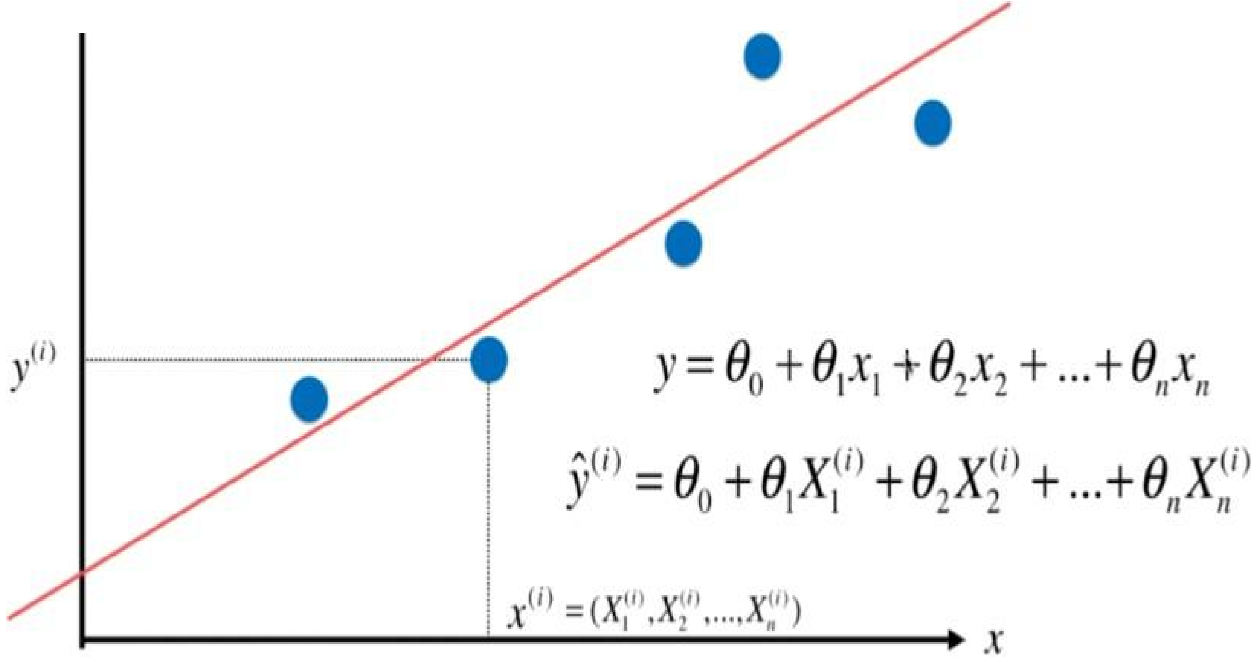
Least square method（Linear Regression）

- Consider the model: $y=\theta_0+\theta_1 x_1+\theta_2 x_2+...+\theta_n x_n$
- Given a random sample of observations, the population regression line is estimated by:

$$\hat{y}^{(i)}=\theta_0+\theta_1 X_1^{(i)}+\theta_2 X_2^{(i)}+...+\theta_n X_n^{(i)}$$

$$\theta=(\theta_0,\theta_1,\theta_2,...,\theta_n)^T$$

- $\boldsymbol{\theta}$ is the parameters we want



Least square method（Linear Regression）

- $\boldsymbol{X}$ is attribute values of all samples, $X^{(i)}$ denotes the i-th sample, X_n denotes the n-th attribute

$$\hat{y}^{(i)} = \theta_0 X_0^{(i)} + \theta_1 X_1^{(i)} + \theta_2 X_2^{(i)} + \ldots + \theta_n X_n^{(i)} \quad , X_0^{(i)} \equiv 1$$

$$X^{(i)} = (X_0^{(i)}, X_1^{(i)}, X_2^{(i)}, \ldots, X_n^{(i)})$$

$$\hat{y}^{(i)} = X^{(i)} \cdot \theta$$

Least square method

$$X_b = \begin{pmatrix} 1 & X_1^{(1)} & X_2^{(1)} & \dots & X_n^{(1)} \\ 1 & X_1^{(2)} & X_2^{(2)} & \dots & X_n^{(2)} \\ \dots & & & & \dots \\ 1 & X_1^{(m)} & X_2^{(m)} & \dots & X_n^{(m)} \end{pmatrix} \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \dots \\ \theta_n \end{pmatrix} \qquad \hat{y} = X_b \cdot \theta$$

$$e = \left\| y - \hat{y} \right\| = \left\| y - X_b \theta \right\|$$

Least square method

- To make $||y - \hat{y}||$ as small as possible, the solution is

$$\theta = (X_b^T X_b)^{-1} X_b^T y$$

- This means $\boldsymbol{\theta}$ is the orthogonal projection of y into $\mathrm{R}(X_b)$

- If X_b is too large, calculate the inverse is expensive. we often use some optimization methods (like gradient descent)

Evaluation

- How to evaluate the model? We compare the predict $\hat{y}$ with real y

- Mean Squared Error (MSE)

$$\frac{1}{m}\sum_{i=1}^{m}(y_{test}^{(i)}-\hat{y}_{test}^{(i)})^2$$

- Root Mean Squared Error (RMSE)

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{test}^{(i)}-\hat{y}_{test}^{(i)})^2} = \sqrt{MSE_{test}}$$

Evaluation

- Mean Absolute Error (MAE)

$$\frac{1}{m}\sum_{i=1}^{m} | y_{test}^{(i)} - \hat{y}_{test}^{(i)} |$$

- R-Squared (R^2): value closer to 1 is better

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}}$$ (Residual Sum of Squares) (Total Sum of Squares)

$$R^2 = 1 - \frac{\sum_i (\hat{y}^{(i)} - y^{(i)})^2}{\sum_i (\bar{y} - y^{(i)})^2}$$

$$R^2 = 1 - \frac{(\sum_i (\hat{y}_i - y_i)^2)/m}{(\sum_i (\hat{y}_i - y_i)^2)/m}$$

$$= 1 - \frac{MSE(\hat{y}, y)}{Var(y)}$$

Polynomial regression

- Now consider a more complex situation: $\Phi(x) = \prod_{n=0}^{N} x_n^{k_n}$, $\sum_n k_n \leq d$
- It becomes polynomial regression. Theoretically, polynomial can fit any function if the degree (d) were large enough.
- For example, we have 2 attributes, degree is 3

$$x_1, x_2 \quad \Rightarrow \quad \begin{array}{l} 1, x_1, x_2 \\ x_1^2, x_2^2, x_1 x_2 \\ x_1^3, x_2^3, x_1^2 x_2, x_1 x_2^2 \end{array}$$

Standardization

- We have introduced data normalization, **standardization** is another way to regularize data.

$$x_{scale} = \frac{x - x_{mean}}{s} \qquad s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- It makes sure the value of each attribute provided by all samples have a mean of 0, and a standard deviation (s) of 1.

- It is good when the data value has no boundary or has some outliers.

Polynomial Regression Pipeline

- When we performing polynomial regression, 3 steps are required:
  - ✓ Make polynomial features
  - ✓ Standardization (for gradient descent)
  - ✓ Linear regression
- Scikit-learn provides the *Pipeline* class for customizing our own polynomial regression

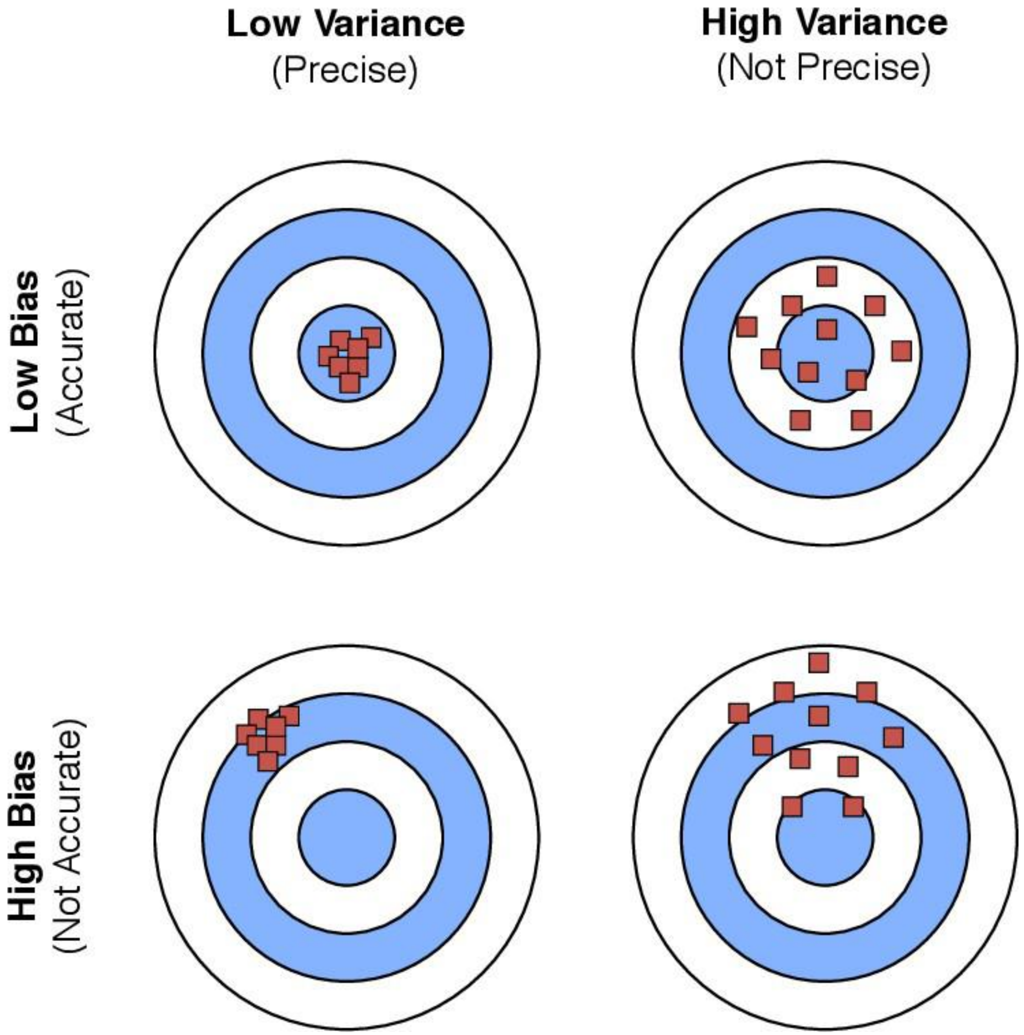
```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PolynomialFeatures

poly_reg = Pipeline([
    ('poly', PolynomialFeatures(degree=2)),
    ('std_scaler', StandardScaler()),
    ('lin_reg', LinearRegression())
])
```

Bias & Variance

- Bias usually caused by underfitting, Variance caused by overfitting

- Choose a better polynomial degree and regularization coefficient

- Use training set, validation set and test set or cross validation to acquire a better model



Regularization

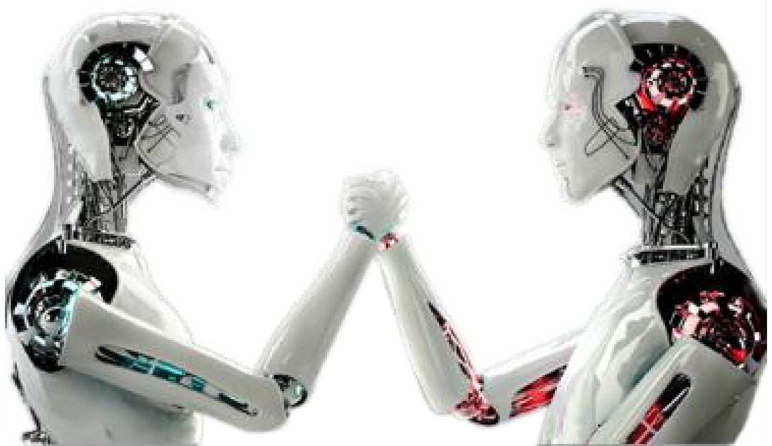
- We can use regularization to restrict the parameters in $\boldsymbol{\theta}$

- In scikit-learn, class Ridge corresponds to

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

```python
from sklearn.linear_model import Ridge

def RidgeRegression(degree, alpha):
    return Pipeline([
        ('poly', PolynomialFeatures(degree=degree))
        ('std_scaler', StandardScaler()),
        ('ridge_reg', Ridge(alpha=alpha))
    ])
```

Lab Task



Lab Task

1. Complete the exercises and questions in the lab03_Bayes.pdf

2. Submit two result files with the same content to bb. The extensions of these two files are **ipynb** and **pdf**, respectively.

Lab1: 周三 上午3-4节 荔园6栋408机房
Lab2: 周三 下午7-8节 荔园6栋406机房
Lab3: 周二下午5-6节 荔园6栋409机房
Lab4: 周二下午7-8节 荔园6栋406机房

Thanks

贾艳红 Jana
Email:jiayh@mail.sustech.edu.cn