Linear Regression

Hoàng-Nguyên Vũ

1 Lý thuyết về Linear Regression

1.1 Giới thiệu về Linear Regression

Hồi quy tuyến tính (**Linear Regression**) là một thuật toán trong Machine Learning được sử dụng để dự đoán một biến phụ thuộc (y) dựa trên một hoặc nhiều biến độc lập (X). Đây là một trong những mô hình đơn giản nhưng mạnh mẽ để phân tích dữ liệu.

Có hai loại chính:

- **Hồi quy tuyến tính đơn giản** (*Simple Linear Regression*) – chỉ có một biến đầu vào.
- **Hồi quy tuyến tính bội** (*Multiple Linear Regression*) – có nhiều biến đầu vào.

1.2 Công thức toán học

1.2.1 Hồi quy tuyến tính đơn giản

Công thức tổng quát của hồi quy tuyến tính đơn giản:

$$y = wX + b$$

Trong đó:

- y là giá trị đầu ra cần dự đoán.
- X là biến đầu vào.
- w là hệ số hồi quy.
- b là hằng số (bias).

Ví dụ: Dự đoán giá nhà dựa trên diện tích:

$$\text{Giá nhà} = w \times \text{Diện tích} + b$$

1.2.2 Hồi quy tuyến tính bội

Khi có nhiều biến đầu vào, công thức tổng quát là:

$$y = w_1X_1 + w_2X_2 + \cdots + w_nX_n + b$$

Hoặc viết dưới dạng ma trận:

$$Y = XW + b$$

1.3 Tìm tham số w và b

Để tìm giá trị tối ưu của w và b, ta sử dụng phương pháp **Gradient Descent** để giảm sai số.

1.3.1 Hàm mất mát (Loss Function)

Hàm mất mát phổ biến nhất là **Mean Squared Error (MSE)**:

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

1.3.2 Gradient Descent

Gradient Descent là thuật toán tối ưu hóa giúp cập nhật w và b sao cho giảm MSE:

$$w = w - \alpha \cdot \frac{1}{m}\sum(y_i - \hat{y}_i) \cdot X_i$$

$$b = b - \alpha \cdot \frac{1}{m}\sum(y_i - \hat{y}_i)$$

1.4 Đánh giá mô hình

Sau khi huấn luyện mô hình hồi quy tuyến tính, ta cần đánh giá hiệu suất của nó để đảm bảo độ chính xác khi áp dụng vào thực tế. Có nhiều chỉ số được sử dụng để đánh giá một mô hình hồi quy tuyến tính, bao gồm:

- **Mean Squared Error (MSE)**
- **Root Mean Squared Error (RMSE)**
- **Mean Absolute Error (MAE)**
- **R-squared Score (R^2)**

Dưới đây là chi tiết về từng phương pháp đánh giá.

1.4.1 Mean Squared Error (MSE)

MSE đo lường độ lệch trung bình bình phương giữa giá trị dự đoán và giá trị thực tế. Công thức tính MSE như sau:

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

Trong đó:

- m là số lượng mẫu trong tập dữ liệu.

- y_i là giá trị thực tế của mẫu i.
- $\hat{y}_i$ là giá trị dự đoán của mẫu i.

Ý nghĩa:

- MSE càng nhỏ chứng tỏ mô hình dự đoán càng chính xác.
- Do lỗi được bình phương, MSE nhạy cảm với outliers (các giá trị ngoại lai).

1.4.2 Root Mean Squared Error (RMSE)

RMSE là căn bậc hai của MSE, giúp biểu diễn sai số theo cùng đơn vị với dữ liệu gốc:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$$

Ý nghĩa:

- RMSE có cùng đơn vị với đầu ra y, dễ diễn giải hơn MSE.
- Giá trị RMSE thấp chứng tỏ mô hình có độ chính xác cao.

1.4.3 Mean Absolute Error (MAE)

MAE đo lường trung bình sai số tuyệt đối giữa giá trị thực tế và giá trị dự đoán:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$$

Ý nghĩa:

- MAE ít bị ảnh hưởng bởi outliers hơn MSE do không có bình phương sai số.
- Giá trị MAE càng nhỏ, mô hình càng tốt.

1.4.4 R-squared Score (R^2)

R^2 (coefficient of determination) đo lường mức độ mô hình có thể giải thích được phương sai của dữ liệu. Công thức tính:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Trong đó:

- $\bar{y}$ là giá trị trung bình của y.
- $\sum(y_i - \hat{y}_i)^2$ là tổng phương sai của sai số dự đoán.

- $\sum(y_i - \bar{y})^2$ là tổng phương sai của dữ liệu thực tế.

Ý nghĩa:

- R^2 nằm trong khoảng $[0, 1]$. Giá trị càng cao chứng tỏ mô hình giải thích tốt dữ liệu.
- $R^2 = 1$ khi mô hình dự đoán hoàn hảo.
- $R^2 = 0$ khi mô hình không dự đoán tốt hơn một đường trung bình ngang.
- $R^2 < 0$ khi mô hình hoạt động tệ hơn cả việc đoán giá trị trung bình của dữ liệu.

1.4.5 So sánh các phương pháp đánh giá

Chỉ số	Công thức	Đặc điểm
MSE	$\frac{1}{m}\sum(y_i - \hat{y}_i)^2$	Nhạy cảm với outliers
RMSE	$\sqrt{MSE}$	Dễ diễn giải, nhạy cảm với outliers
MAE	$\frac{1}{m}\sum\lvert y_i - \hat{y}_i\rvert$	Ít bị ảnh hưởng bởi outliers
R^2 Score	$1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$	Đánh giá mức độ giải thích của mô hình

Bảng 1: So sánh các phương pháp đánh giá hồi quy tuyến tính

2 Bài tập

Cài đặt hồi quy tuyến tính

Hãy đọc dữ liệu tại file sau: advertising.csv. Xây dựng lớp `LinearRegression` để thực hiện hồi quy tuyến tính từ đầu bằng **Gradient Descent**.

Yêu cầu

1. Viết một class `LinearRegression` với các phương thức:
   - `fit(X, y)`: Huấn luyện mô hình bằng Gradient Descent.
   - `predict(X)`: Dự đoán giá trị y từ X.
   - `evaluate(X, y)`: Đánh giá mô hình bằng MSE.
   - `plot_regression_line(X, y)`: Vẽ đường hồi quy.
2. Tạo một bộ dữ liệu giả lập.
3. Huấn luyện mô hình và vẽ đường hồi quy.

Gợi ý cài đặt (Python)

```python
import numpy as np
import matplotlib.pyplot as plt

class LinearRegression:
    def __init__(self, learning_rate=0.01, epochs=1000):
        self.learning_rate = learning_rate
        self.epochs = epochs
        self.w = None
        self.b = None

    def fit(self, X, y):
        m, n = X.shape
        self.w = np.zeros(n)
        self.b = 0

        for _ in range(self.epochs):
            ## Your code here ##

    def predict(self, X):
        ## Your code here ##

    def evaluate(self, X, y):
        ## Your code here ##

    def plot_regression_line(self, X, y):
        plt.scatter(X, y, color="blue", label="Data points")
        y_pred = self.predict(X)
        plt.plot(X, y_pred, color="red", label="Regression line")
        plt.xlabel("X")
        plt.ylabel("y")
        plt.legend()
        plt.show()

# Load data chia tỉ lệ: Train 80% và Test 20%
## Your code here ###

# Huấn luyện mô hình
model = LinearRegression(learning_rate=0.01, epochs=1000)
model.fit(X_train, y_train)

# Dự đoán và vẽ đường hồi quy
print(f"MSE: {model.evaluate(X_test, y_test)}")
model.plot_regression_line(X_test, y_test)
```