TÌM HIỂU VỀ MÔ HÌNH LINEAR REGRESSION

Nguyen Tien Manh

Tháng 1, năm 2026

Tóm tắt nội dung

Bài viết này tìm hiểu về mô hình hồi quy phổ biến và cơ bản trong học máy là Linear Regression - Hồi quy tuyến tính. Nội dung chính là tìm hiểu mục đích, ý tưởng ban đầu của mô hình, bản chất toán học và cách xây dựng nó.

Mục lục

1 Mở đầu

Trong thực tế có rất nhiều vấn đề cần được dự báo thông qua các thuật toán hay mô hình. Một số bài toán rất phổ biến như dự đoán giá nhà dựa trên diện tích của nó, dự đoán thu nhập cá nhân dựa trên chi tiêu của một người....

Như vậy mục tiêu đặt ra là xây dựng một mối quan hệ giữa đầu vào (Input) và đầu ra (Output), đó cũng là mục đích hay cơ chế của mô hình hồi quy tuyến tính mà ta sẽ tìm hiểu.

2 Cơ sở lý thuyết

Trong phần này, ta sẽ tìm hiểu các lý thuyết nền tảng của mô hình Linear Regression từ bản chất toán học đến xây dựng hàm mất mát, và tìm hiểu về phương pháp tối ưu.

2.1 Mô hình toán học

Từ một mô hình đơn giản là phương trình đường thẳng:

$$y = ax + b$$

với công việc cần làm là tìm hệ số a và b sao cho với đầu vào là x thì đầu ra y đúng theo như ta mong muốn.

Tuy nhiên với bài toán phức tạp với, với điều kiện đầu vào gồm nhiều features, ta cần tổng quát hóa phương trình trên thành:

$$y = w_0 + w^T x$$

ở đây $w = [w_1, w_2, ...w_n]^T$, w_0 là tham số (parameters) của mô hình và $x = [x_1, x_2, ..., x_n]$ là các đặc trưng (features) của mô hình. Đầu ra của mô hình dự đoán chính là y, vì vậy về mặt tính toán và xử lí ta cần tìm các tham số sao cho mô hình dự đoán này có đàu ra gần đáp án đúng nhất với chi phí về mặt tính toán là thấp nhất.

2.2 Hàm mất mát

Để đo được độ chính xác của mô hình dự đoán, trong bài toán này, ta sử dụng hàm mất mát là *Mean Square Error*, có dạng như sau:

$$\mathcal{L}(w, w_0) = \frac{1}{2n}\sum_{i=1}^{n}(\hat{y} - y_i)^2$$

$$= \frac{1}{2n}\sum_{i=1}^{n}\left((w^T x_i + w_0) - y_i\right)^2$$

Trong đó y_i là kết quả mà chúng ta mong muốn, còn $\hat{y}$ là đầu ra của hàm dự đoán khi đầu vào là x_i. Để mô hình có dự đoán chính xác nhất ta cần cực tiểu hóa hàm mất mát, hay chính là đi tìm tham số w và w_0 sao cho hàm $\mathcal{L}(w, w_0)$ đạt giá trị bé nhất.

Hàm mất mát còn có một dạng khác là *RSS* hay *Residual Sum of Square*:

$$\mathcal{R}(w, w_0)\sum_{i=1}^{n}(\hat{y} - y_i)^2$$

Tuy nhiên giá trị của hàm mất mát trên về mặt giá trị là rất lớn với số mẫu cao, nên ta sử dụng *MSE* - chuẩn hóa cho n để giảm giá trị, thuận lợi cho việc tính toán.

2.3 Phương pháp tối ưu

Ta có hai sự lựa chọn để tối ưu hàm mất mát như sau:

- Nếu dữ liệu nhỏ, ta có thể sự dụng *Phương pháp tuyến tính - Normal Equation*:

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

- Tuy nhiên với dữ liệu lớn, phương pháp trên sẽ có độ phức tạp rất lớn nên người ta có phương pháp tối ưu hơn là *Gradient Descent*:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{2n}\sum_{i=1}^{n} 2x_i(w^T x_i + w_0 - y_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} x_i(\hat{y} - y_i)$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{1}{2n}\sum_{i=1}^{n} 2(w^T x_i + w_0 - y_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\hat{y} - y_i)$$

Với phương pháp *Gradient Descent*, ở mỗi bước lặp ta cập nhật tham số w và w_0 để có tham só mới khiến cho hàm mất mát tiến gần tới cực tiểu hơn. Tuy nhiên để kiểm soát bước nhảy ta cần một công cụ là *Learning rate*, với công cụ này ta đảm bảo cho số lượng bước nhảy sẽ không quá nhiều, đồng thời đảm bảo cho việc tìm thất điểm gần sát với cực tiểu nhất có thể. Ta đặt giá trị *Learning rate* cho mô hình là α:

$$w \leftarrow w - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w}$$

$$w_0 \leftarrow w_0 - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w_0}$$

Sau khi kết thúc vòng lặp với số lần lặp đủ lớn hoặc độ giảm của hàm chi phí đủ nhỏ, chúng ta sẽ thu được giá trị của 2 tham số w và w_0 rất gần với giá trị nghiệm của bài toán. Đến bước này, mô hình *Linear Regression* cơ bản là đã huấn luyện xong.

3 Huấn luyện

3.1 Thuật toán huấn luyện

Từ mô hình toán học của mô hình tôi đã trình bày ở trên, thuật toán để huấn luyện một mô hình *Linear Regression* gồm các bước như sau:

1. Khởi tạo tham số. Chọn $w = 0, w_0 = 0, \alpha = 0.01, iterations = 2000, \epsilon = 10^{-5}$. Trong đó w, w_0 là các tham số của mô hình, α là **Learning rate**, *iterations* là số lần lặp, ϵ để kiểm tra độ giảm của hàm chi phí.

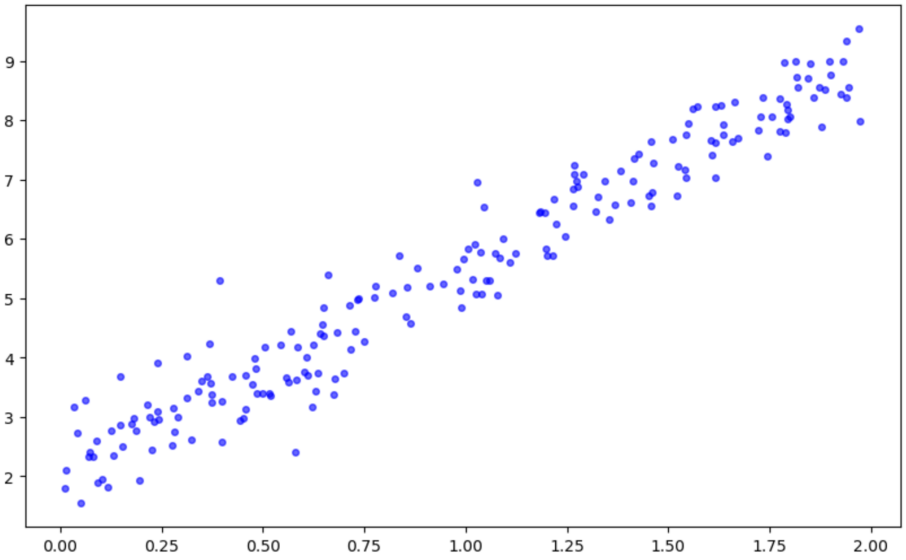
2. Thực hiện phương pháp **Gradient Descent** để tìm tham số tối ưu cho mô hình:

   - Tính đạo hàm: $\mathcal{L}'_w, \mathcal{L}'_{w_0}$.
   - Cập nhật w, w_0:
   $$w \leftarrow w - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w}$$
   $$w_0 \leftarrow w_0 - \alpha \cdot \frac{\partial \mathcal{L}}{\partial w_0}$$
   - Kiểm tra điều kiện lặp với *iterations* và ϵ đã khởi tạo.

Tùy vào đặc điểm dữ liệu, yêu cầu huấn luyện, ta có thể điều chỉnh các tham số khởi tạo, tốc độ học hay số vòng lặp sao cho phù hợp với như cầu của người huấn luyện mô hình và đảm bảo tối ưu cho quá trình huấn luyện.

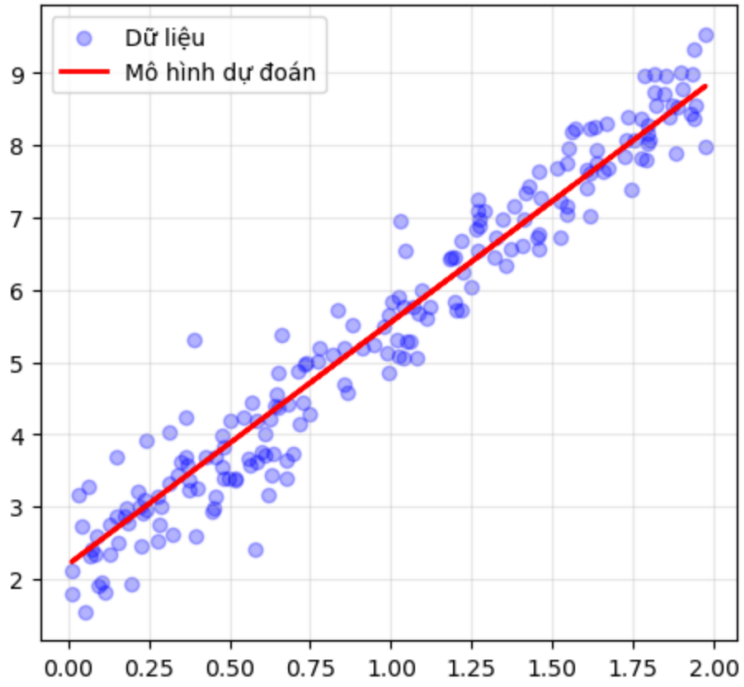
3.2 Cài đăt

Tôi xây dựng Notebook *Linear_Regression_from_scratch.ipynb* để tạo bộ dữ liệu, tạo các hàm cần thiết và huấn luyện mô hình. Hàm **fit(X, y)** là hàm để huấn luyện mô hình. Bộ dữ liệu được tạo ngẫu nhiên và có hình dạng như sau khi được trực quan hóa:



3.3 Kết quả huấn luyện

Sau khi được huấn luyện, mô hình đã tạo ra một đường thẳng như sau:



4 Thảo luận và đề xuất

4.1 Hạn chế của mô hình Linear Regression

- Mô hình rất nhạy cảm với *Outlier*, chỉ cần một điểm dữ liệu sai lệch cũng khiến sai số trở lên rất lớn.
- Mô hình chỉ thực sự phù hợp với dữ liệu tuyến tính, chính vì thế ta cần đảm bảo nắm rõ đặc điểm dữ liệu trước khi xây dựng mô hình này.

4.2 Thảo luận thêm

- Ngoài phương pháp tối ưu chính là *Gradient Descent* tôi đã trình bày ở trên, ta còn một số phương pháp giúp cho *Gradient Descent* hội tụ nhanh hơn như **Feature Scaling** với các kĩ thuật như **Normalization**. Ngoải ra còn một số kĩ thuật tối ưu dữ liệu với **Feature Engineering**.
- Mở rộng hơn của *Linear Regression* còn có *Polynomial regression* giúp tạo ra các đường cong fit với dữ liệu hơn.