ĐỒ ÁN THỰC HÀNH CUỐI KỲ

1 Tối ưu lồi - Data Fitting (5 điểm)

Hồi quy tuyến tính (Linear Regression) là một trong những thuật toán cơ bản nhất của *học máy (Machine Learning)*. Đây là phương pháp thống kê được sử dụng để mô hình hóa mối quan hệ giữa một biến đầu ra (hay còn gọi là biến mục tiêu) với một hoặc nhiều biến đầu vào (hay còn gọi là biến dự đoán). Phương pháp này được áp dụng chủ yếu trong các bài toán *học có giám sát (Supervised Learning)* nhằm dự đoán giá trị của một biến liên tục.

1.1 Mô hình hồi quy tuyến tính

Nguyên lý hoạt động của hồi quy tuyến tính là đi tìm mối quan hệ giữa biến mục tiêu Y và biến dự đoán X thông qua một hàm tuyến tính theo công thức tổng quát như sau:

$$y = f(x_1, x_2, ..., x_n) + \epsilon = w_0 + w_1 x_1 + ... + w_n x_n + \epsilon.$$

Trong đó:

- $\mathbf{w} = (w_0, w_1, ..., w_n)$ là vectơ tham số của mô hình (sẽ đề cập cách tính sau). Từng tham số w_i đại diện cho mức độ phụ thuộc của mô hình vào biến x_i. Giá trị w_i càng cao thì độ quan trọng của biến x_i trong mô hình học càng lớn và ngược lại.

- $\mathbf{x} = (x_1, x_2, ..., x_n)$ là *vectơ đặc trưng (feature vector)* của mô hình. Đối với phương pháp hồi quy tuyến tính, ta thường thêm giá trị $x_0 = 1$ vào vectơ đặc trưng $\mathbf{x}$ để hàm tuyến tính $f(\mathbf{x})$ có dạng:

$$y = f(\mathbf{x}) + \epsilon = w_0 x_0 + w_1 x_1 + ... + w_n x_n + \epsilon = \sum_{i=0}^{n} w_i x_i + \epsilon = \langle \mathbf{w}, \mathbf{x} \rangle + \epsilon = \mathbf{w}^T \mathbf{x} + \epsilon.$$

- ϵ là *sai số (error term)* đại diện cho các tác động bên ngoài khi triển khai mô hình trong thực tế. Ví dụ, một người đang muốn bán gấp căn nhà để có tiền đầu tư vào thương vụ làm ăn lớn nên sẽ hạ giá nhà xuống để xác suất bán được nhà tăng lên, từ đó đem số tiền trên đi đầu tư. Khi huấn luyện mô hình, sai số ϵ thường được sinh ra từ phân phối chuẩn tắc $\mathcal{N}(0, 1)$.

Mục tiêu chính của các phương pháp học máy là tìm vectơ tham số $\mathbf{w} = (w_0, w_1, ..., w_n)$ sao cho mô hình có thể đưa ra dự đoán "đủ tốt" đối với bài toán mà ta đang "huấn luyện". Quá trình "huấn luyện" cho phép mô hình "học" từ các dữ liệu thực tế, từ đó cải thiện khả năng thực hiện tác vụ một cách chính xác nhất. Vì vậy, để xác định được vectơ tham số $\mathbf{w}$ phù hợp, ta cần thu thập một bộ dữ liệu thích hợp với bài toán đang giải quyết. Sau đó, ta thực hiện "huấn luyện" mô hình (lựa chọn vectơ tham số $\mathbf{w}$) sao cho mô hình có thể đưa ra các dự đoán sát nhất với kết quả thực tế.

Tiếp theo, làm sao ta thực hiện "huấn luyện" mô hình từ tập dữ liệu mà ta đã thu thập ở bước trên? Tại bước này, ta mong muốn tìm một đường thẳng trên đồ thị mà khoảng cách từ các điểm dữ liệu đến đường thẳng mục tiêu là nhỏ nhất. Do đó, ta định nghĩa *hàm mất mát (Loss Function)*, thường là *hàm sai số toàn phương trung bình (Mean Squared Error - MSE)* có dạng

$$\mathcal{L}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - (f(\mathbf{x}) + \epsilon))^2$$

với n là kích thước của tập dữ liệu huấn luyện, y_i là giá trị thực tế ở trong tập dữ liệu, và $\hat{y}_i$ là giá trị dự đoán của mô hình hồi quy tuyến tính. Đối với hàm $\mathcal{L}(\mathbf{w})$ vừa được định nghĩa, ta muốn tìm vectơ tham số $\mathbf{w}$ sao cho hàm mất mát đạt giá trị nhỏ nhất.

Một phương pháp quen thuộc để tìm giá trị nhỏ nhất cho hàm $\mathcal{L}(\mathbf{w})$ là thực hiện phương pháp đạo hàm và tìm điểm cực trị, từ đó có thể kết luận được giá trị nhỏ nhất và giá trị lớn nhất. Đối với trường hợp đặc biệt là mô hình hồi quy tuyến tính, sau một loạt các phép biến đổi toán học, ta có được công thức tính vectơ tham số $\mathbf{w}$ để hàm mất mát $\mathcal{L}(\mathbf{w})$ đạt giá trị nhỏ nhất như sau:

$$\mathbf{w} = (X^T X)^{-1}(X^T Y).$$

Trong đó, ma trận X và Y được xây dựng sau khi xử lý bộ dữ liệu ta cần huấn luyện cho mô hình, và giả sử sai số $\epsilon = 0$ trong quá trình huấn luyện.

Đối với sinh viên nào có hứng thú về học máy, các bạn có thể tham khảo Gradient Descent - thuật toán tổng quát để huấn luyện mô hình máy học. Tuy vậy, thuật toán này yêu cầu kích thước tập dữ liệu huấn luyện phải lớn để hàm mất mát đạt giá trị nhỏ nhất, dẫn đến thời gian huấn luyện mô hình lâu.

1.2 Yêu cầu bài toán

Hai điều quan trọng trước khi huấn luyện một mô hình máy học là (i) tìm tập dữ liệu phù hợp với bài toán mà ta đang giải quyết, và (ii) chọn mô hình $f(\mathbf{x})$ phù hợp. Trong phần này, sinh viên sẽ sử dụng phương pháp hồi quy tuyến tính để xây dựng mô hình dự đoán điểm thân thiết của khách hàng thông qua thói quen mua hàng của họ trong siêu thị.

File chứa tập dữ liệu mà ta sử dụng để huấn luyện mô hình là `customer_purchase_behaviors.csv`. Bộ dữ liệu có 238 dòng, mỗi dòng là thông tin của một khách hàng trong siêu thị, và có tổng cộng 7 đặc trưng chính như sau:

- `user_id`: ID của khách hàng (khác nhau với từng khách hàng).
- `age`: Tuổi của khách hàng.
- `annual_income`: Thu nhập của khách hàng theo năm (tính theo đơn vị USD).
- `purchase_amount`: Tổng giá tiền mua hàng của người dùng (tính theo đơn vị USD).
- `purchase_frequency`: Tần suất mua hàng của người dùng (tính theo tổng số lần trong 1 năm).
- `region`: Tên vùng mà khách hàng sinh sống (North, South, East, West).
- `loyalty_score`: Điểm thân thiết của khách hàng trong siêu thị.

Khi xây dựng mô hình hồi quy tuyến tính, sinh viên bỏ qua cột `user_id` và `region` trong file dữ liệu. Sau đó, sinh viên thực hiện những yêu cầu dưới đây.

(a). Mô tả đầu vào (input) và đầu ra (output) của mô hình cần được xây dựng.

(b). Sử dụng thư viện `matplotlib`, để xem mối liên hệ giữa đặc trưng thứ i và đầu ra của tập dữ liệu, vẽ biểu đồ thể hiện các điểm dữ liệu cho từng cặp (X_i, Y), trong đó X_i là đặc trưng thứ i của tập dữ liệu, và Y là đầu ra của tập dữ liệu.

(c). Xây dựng mô hình hồi quy tuyến tính dạng đơn giản nhất, $y = w_0 + w_1 x_1 + ... + w_n x_n$ với n là số lượng đặc trưng trong tập dữ liệu, trong đó sử dụng toàn bộ tất cả các biến đầu vào được mô tả ở câu (a).

(d). Xét mô hình hồi quy tuyến tính $y = w_0 + w_1 x_1$ chỉ sử dụng 1 đặc trưng duy nhất, hãy tìm đặc trưng mà mô hình hồi quy tuyến tính thể hiện tốt nhất.

(e). Sinh viên hãy thiết kế một mô hình hồi quy tuyến tính khác với những mô hình trên mà cho kết quả tốt nhất. Lưu ý, ta chỉ cần tính chất "tuyến tính" cho các tham số w_i, còn x_i có thể ở bất kì dạng nào. Do đó, các bạn có thể thay đổi x_i tuỳ ý, ví dụ x_1 thành x_1^2 hay $\sqrt{x_1}$, miễn là mô hình các bạn có thể đạt được kết quả tốt hơn những mô hình ở câu (c) và (d).

2 Xích Markov (5 điểm)

2.1 Cơ sở lý thuyết

Trước khi bắt đầu giải bất kỳ bài toán xích Markov nào, ba tham số quan trọng mà ta cần phải xác định là (i) không gian trạng thái, (ii) ma trận chuyển trạng thái, và (iii) phân phối đầu (thường được kí hiệu lần lượt là S, P, π_0).

Gọi hai thời điểm liên tiếp nhau lần lượt là t và $t+1$ ($t \geq 0$). Gọi X_t là biến ngẫu nhiên dùng để mô tả trạng thái tại thời điểm t. Ta định nghĩa ma trận chuyển trạng thái P là ma trận vuông có kích thước $|S| \times |S|$ như sau:

$$P = \begin{bmatrix} Pr[X_{t+1} = 1|X_t = 1] & Pr[X_{t+1} = 1|X_t = 2] & \dots & Pr[X_{t+1} = 1|X_t = |S|] \\ Pr[X_{t+1} = 2|X_t = 1] & Pr[X_{t+1} = 2|X_t = 2] & \dots & Pr[X_{t+1} = 2|X_t = |S|] \\ \vdots & \vdots & \ddots & \vdots \\ Pr[X_{t+1} = |S||X_t = 1] & Pr[X_{t+1} = |S||X_t = 2] & \dots & Pr[X_{t+1} = |S||X_t = |S|] \end{bmatrix}.$$

với mỗi cột đại diện cho xác suất của các trạng thái ở thời điểm $t+1$ nếu cho trước trạng thái ở thời điểm t. Do đó, tổng các xác suất của từng cột trong ma trận chuyển trạng thái P phải có giá trị là 1.

Tuỳ thuộc vào yêu cầu bài toán, ta sẽ định nghĩa được phân phối đầu $\pi_0 = \begin{bmatrix} Pr[X_0 = 1] \\ \vdots \\ Pr[X_0 = |S|] \end{bmatrix}$. Cùng với việc đã định nghĩa ma trận chuyển trạng thái P trước đó, ta có một số công thức sau đây:

- Công thức truy hồi để tính phân phối xác suất π_t dựa trên phân phối xác suất π_{t-1} là $\pi_t = P \times \pi_{t-1}$.
- Từ đó, phân phối xác suất của các trạng thái tại thời điểm t là $\pi_t = \begin{bmatrix} Pr[X_t = 1] \\ \vdots \\ Pr[X_t = |S|] \end{bmatrix} = P^t \times \pi_0$.
- Xích Markov tồn tại phân phối dừng khi tồn tại $t \in \mathbb{N}$ sao cho $\pi_{t+1} \approx \pi_t$. Đồng thời, ta có $\pi_{t+1} = P \times \pi_t$. Suy ra, $P \times \pi_t \approx \pi_t \Leftrightarrow (P - I) \times \pi_t \approx 0$. Để ý rằng ta đã đưa về được thành bài toán giải hệ phương trình tuyến tính sử dụng phương pháp Gauss. Từ đó, giải hệ trên và lời giải của hệ cũng chính là phân phối dừng của xích Markov.
- Phân phối giới hạn π_L có hai tính chất quan trọng là (i) phân phối không phụ thuộc vào phân phối đầu, và (ii) luôn hội tụ về một phân phối duy nhất với mọi phân phối đầu π_0. Theo định nghĩa trên, ta định nghĩa phân phối giới hạn π_L như sau:

$$\pi_L \text{ là phân phối giới hạn} \Leftrightarrow \forall \pi_0 : \lim_{n \to \infty} (P^n \pi_0) = \pi_L.$$

Nếu ma trận chuyển trạng thái P là chính quy thì ta có thể kết luận phân phối giới hạn chính là phân phối dừng.

2.2 Yêu cầu bài toán

Cho một con xúc xắc cân bằng có 6 mặt được đánh số từ 1 đến 6. Gọi S_n là tổng các kết quả sau khi tung xúc xắc n lần đầu tiên. Ta muốn khảo sát phân phối của giá trị phần dư của S_n khi chia cho 7.

(a). Hãy mô tả biến ngẫu nhiên X_n phù hợp cho bài toán trên mà có tính chất Markov. Từ đó, xác định ma trận chuyển trạng thái P và vectơ phân phối đầu π_0.

(b). Viết hàm dùng để tính xác suất xuất hiện các giá trị phần dư của S_n khi chia cho 7 theo bảng sau:

	$S_n \ \% \ 7 = 0$	$S_n \ \% \ 7 = 1$	$S_n \ \% \ 7 = 2$	...
$n = 1$				
$n = 2$				
...				
$n = 10$				

(c). Viết hàm dùng để kiểm tra xích Markov đã cho có tồn tại phân phối dừng hay không. Nếu có, hãy tính phân phối dừng và chỉ ra thời điểm $t \in \mathbb{N}$ sao cho phân phối xác suất π_t chính là phân phối dừng.

(d). Quá trình tung xúc xắc được diễn ra cho đến khi tồn tại $i \in \mathbb{N}^*$ sao cho giá trị S_i chia hết cho 7 thì dừng. Viết hàm tính xác suất tung xúc xắc không quá n lần với giá trị n là một trong những đầu vào của hàm.

3 Yêu cầu chung về đồ án

- Đồ án này được thực hiện cá nhân. Nếu phát hiện bất kì hành vi sao chép bài nào của các bạn cùng môn học, toàn bộ phần điểm cho đồ án thực hành cuối kỳ của những sinh viên có liên quan sẽ được đưa về 0.
- Sinh viên chỉ được phép dùng thư viện `pandas` cho việc đọc dữ liệu, thư viện `matplotlib` hoặc `seaborn` để trực quan hoá dữ liệu, và thư viện `math` để thực hiện các tính toán trên số thực.
- Sinh viên nộp 4 file lên Moodle có tên như sau:

  ```
  MSSV_cau1.ipynb
  MSSV_cau1.pdf
  MSSV_cau2.ipynb
  MSSV_cau2.pdf
  ```

  Trong đó:

  - Thay cụm `MSSV` thành mã số sinh viên. Ví dụ, file dùng để trình bày câu 1 có tên là `23120101_cau1.ipynb`.
  - Các file `*.ipynb` là các file chứa toàn bộ báo cáo và mã nguồn ứng với từng yêu cầu bài tập nêu trên.
    - Mở đầu file `*.ipynb` cho từng câu là đoạn văn bản giới thiệu thông tin cá nhân của sinh viên, bao gồm họ và tên, MSSV, và lớp học phần.
    - Sinh viên mô tả ngắn gọn chức năng của các hàm tự viết tại nơi định nghĩa hàm.
  - Các file `*.pdf` được export từ file `*.ipynb` tương ứng.

  Nếu sinh viên nộp sai quy định thì toàn bộ phần đồ án thực hành cuối kỳ sẽ bị điểm 0.

- Nếu chương trình không biên dịch được ở bất kì đoạn mã nguồn nào thì sẽ bị điểm 0 ở bài tập đó.
- Mọi thắc mắc về đồ án này, vui lòng gửi qua email `letronganhtu@gmail.com`.