TRƯỜNG ĐẠI HỌC THỦY LỢI

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: HỌC MÁY

ĐỀ TÀI: Sử dụng một số phương pháp phân lớp trong học có giám sát để dự đoán nguy cơ đột quỵ não

Giáo viên hướng dẫn: TS. Nguyễn Thị Kim Ngân

Sinh viên/nhóm sinh viên thực hiện:

1. Vũ Anh Xuân, lớp 63TTNT
2. Đào Thị Trang, lớp 63TTNT
3. Khổng Quốc Trung, lớp 63TTNT
4. Nguyễn Quang Việt, lớp 63TTNT

Hà Nội, năm 2023

I) Lý thuyết

***) Mô tả bài toán**

- Bài toán nhằm dự đoán nguy cơ đột quỵ não. Sử dụng các đặc trưng của dataset brain_stroke.csv

1) Phương pháp Perceptron

a) Định nghĩa

- Perceptron là một thuật toán Classification cho trường hợp đơn giản nhất:
  - \+ chỉ có hai class (binary classification)
  - \+ cũng chỉ hoạt động được trong một trường hợp rất cụ thể Giả sử chúng ta có hai tập hợp dữ liệu đã được gán nhãn

b) Đầu vào

Cho tập dữ liệu huấn luyện gồm N mẫu. Mỗi mẫu là một cặp (x_i,y_i):

- x_i: vector đặc trưng
- y_i: giá trị của vector đặc trưng x

c) Đầu ra

G/S, tại một thời điểm, ta tìm được đường boundary là đường phẳng có phương trình:

$$f\mathrm{w}\ \mathrm{x} = w1x1 + w2x2 + \cdots + wdxd + w0 = 0$$

Hay $\mathrm{w}T\ \mathrm{x} = 0$

w=(w0 , w1 , w2 , …, wd)

x=(x1, x1 , x2 , …, xd)

d) Định nghĩa hàm mất mát

- Xây dựng hàm mất mát:

  Hàm mất mát đơn giản nhất chúng ta nghĩ đến là hàm đếm số lượng các điểm bị misclassied và tìm cách tối thiểu hàm số này:

  $$J1\ \mathrm{w} = \square\ \mathrm{x}i\in M\ (-yisgn(\ \mathrm{w}T\ \mathrm{x}i))$$

  M: số điểm phân lớp lỗi

  Hạn chế quan trọng: hàm số này là rời rạc, không tính được đạo hàm theo w nên rất khó tôi ưu

- Xét hàm mất mát:

  $J\ \mathrm{w} = \square\ \mathrm{x}i\in M\ (-yi\ \mathrm{w}T\ \mathrm{x}i)$ Khi một x*i* (bị phân lớp sai) nằm càng xa boundary thì giá trị $-yi\ \mathrm{w}T\ \mathrm{x}i$ càng lớn Giá trị nhỏ nhất của hàm mất mát này bằng 0 nếu không có điểm nào bị phân lớp sai Hàm này trừng phạt nặng những điểm lấn sâu sang lãnh thổ của lớp khác

- Tại một thời điểm, nếu ta chỉ quan tâm đến điểm bị phân lớp sai, hàm *J*(w): tính được đạo hàm Ta có thể sử dụng giảm Gradient để tìm w Với một điểm x*i*bị phân lớp sai, hàm mất mát trở thành: $J\ \mathrm{w}, \mathrm{x}i\ , yi = -yi\ \mathrm{w}T\ \mathrm{x}i$ Đạo hàm: $\nabla \mathrm{w}J\ \mathrm{w}, \mathrm{x}i\ , yi = -yi\ \mathrm{x}i$ aw=> (aw)'=a

- Quy tắc cập nhật:

  $w = w + \eta y_i x_i$

  $w_{t+1} = w_t + \eta f'(w) = w_t + \eta y_i x_i$

Ta có một quan sát nhỏ

$w_{t+1}^T x_i = (w_t + y_i x_i)^T x_i = w_t^T x_i + y_i \|x_i\|_2^2$

Nếu $y_i = 1$: vì x_i bị phân lớp sai nên $w_t^T x_i < 0$

vì $y_i = 1$ nên $y_i \|x_i\|_2^2 = \|x_i\|_2^2 \geq 1$

do đó, $w(t+1)^T x_i > w(t)^T x_i$

w_{t+1} tiến về phía làm cho x_i được phân lớp đúng

e) Phương pháp

1. Tại thời điểm t = 0, chọn ngẫu nhiên một vector hệ số w0
2. Tại thời điểm t, nếu không có điểm dữ liệu nào bị phân lớp lỗi, dừng thuật toán.
3. Giả sử xi là một điểm bị phân lớp lỗi. Cập nhật wt+1 = wt + yi xi
4. Thay đổi t = t + 1 rồi quay lại Bước 2

2) Phương pháp SVM

a) Định nghĩa

- Bài toán tối ưu trong Support Vector Machine (SVM) là tìm đường phân chia sao cho margin là lớn nhất (Maximum Margin Classifier)
- Ý tưởng của SVM: Margin của một siêu phẳng được định nghĩa là khoảng cách từ các điểm gần nhất của lớp đó tới mặt phân chia. Margin của hai lớp phải bằng nhau và lớn nhất có thể
- Cần tìm một đường phân chia sao cho:
  - Khoảng cách từ điểm gần nhất của mỗi lớp tới đường phân chia là như nhau (margin, lề).
  - Margin này phải là cực đại

b) Đầu vào

Giả sử rằng các cặp dữ liệu trong tập huấn luyện là

(x1, y1),(x2, y2), . . . ,(xN , yN) với vector xi ∈ R d thể hiện đầu vào của một điểm dữ liệu và yi là nhãn của điểm dữ liệu đó, d là số chiều của dữ liệu và N là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi yi = 1 hoặc yi = −1 giống như trong PLA

c) Đầu ra

Mặt phẳng phân chia giữa hai classes là

$$\mathbf{w}^T\mathbf{x} + b = w_1 x_1 + w_2 x_2 + b = 0$$

d) Định nghĩa hàm mất mát

-Xét trong không gian hai chiều

Các điểm vuông xanh thuộc class 1

Các điểm tròn đỏ thuộc class -1

Mặt phẳng phân chia giữa hai classes là

wTx +b= $w1x1 + w2x2 + b=0$

Class 1 nằm về phía dương, class -1 nằm về phía âm của mặt phân chia

-Với cặp dữ liệu (xn ,yn) bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||_2}$$

-Vì *n* luôn cùng dấu với phía của *n* , nên *n* cùng dấu với *T n* , tử số luôn là 1 số không âm.

-Với mặt phân chia như trên, margin được tính là khoảng cách gần nhất từ 1 điểm tới mặt đó:

$$\text{margin} = \min_n \frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||_2}$$

-Bài toán tối ưu trong SVM chính là bài toán tìm và b sao cho margin này đạt giá trị lớn nhất:

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \left\{ \min_n \frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||_2} \right\} = \arg\max_{\mathbf{w},b} \left\{ \frac{1}{||\mathbf{w}||_2} \min_n y_n(\mathbf{w}^T\mathbf{x}_n + b) \right\}$$

-Việc giải trực tiếp bài toán này sẽ rất phức tạp, nhưng ta có cách để đưa nó về bài toán đơn giản hơn

-Nhận xét quan trọng nhất là nếu ta thay vector hệ số bởi và bởi trong đó là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức margin không đổi

-Vậy bài toán tối ưu

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \left\{ \min_n \frac{y_n(\mathbf{w}^T\mathbf{x}_n + b)}{||\mathbf{w}||_2} \right\} = \arg\max_{\mathbf{w},b} \left\{ \frac{1}{||\mathbf{w}||_2} \min_n y_n(\mathbf{w}^T\mathbf{x}_n + b) \right\}$$

-Có thể đưa về bài toán tối ưu có ràng buộc sau đây

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \frac{1}{||\mathbf{w}||_2}$$

$$\text{subject to:} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \forall n = 1, 2, \ldots, N$$

-Bằng phép lấy nghịch đảo, bài toán trên chuyển thành

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$

thoả mãn: $1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 0, \forall n = 1, 2, \ldots, N$

e) Phương pháp

- Sau khi đã tìm được mặt phân cách wTx + b = 0
- Nhãn của bất kỳ một điểm được xác định bằng

class(x) = sgn(wTx + b)

3) Phương pháp ID3

a) Định nghĩa

- Cây quyết định là một mô hình có giám sát (supervised learning), có thể được áp dụng cho cả bài toán phân lớp (classification) và hồi quy (regression)

- Các thuộc tính của cây quyết định có thể là:
  - Thuộc tính rời rạc và không có thứ tự (categorical). Ví dụ, mưa, nắng hay xanh, đỏ, …
  - Thuộc tính liên tục (numeric)
- Cây quyết định
  - Cấu trúc cây giống như biểu đồ luồng
  - Mỗi nút trong thể hiện một sự kiểm tra trên một thuộc tính
  - Mỗi nhánh đại diện cho một kết quả của sự kiểm tra
  - Các nút lá đại diện cho các nhãn lớp hoặc phân phối lớp
- Việc tạo cây quyết định gồm 2 giai đoạn
  - Xây dựng cây quyết định
    - Tại bước khởi tạo, nút gốc bao gồm tất cả các mẫu huấn luyện
    - Các mẫu được phân chia đệ quy dựa trên thuộc tính được chọn
  - Tỉa cây Phát hiện và loại bỏ những nhánh nhiễu hoặc ngoại lệ

b) Đầu vào

Cho tập dữ liệu huấn luyện gồm N mẫu. Mỗi mẫu là một cặp (x_i,y_i):

• x_i: vector đặc trưng

• y_i: giá trị của vector đặc trưng x

$$x^* = \arg\max_x G(x, \mathcal{S}) = \arg\min_x H(x, \mathcal{S})$$

c) Đầu ra

- Cây quyết định .Tại mỗi node, thuộc tính được chọn được xác định dựa trên:

d) Phương pháp

Hàm số entropy

Cho một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau $x1, x2,\ldots,xn$.

Giả sử rằng xác suất để x nhận các giá trị này là $pi = p\ x = xi$, $với 0 \leq pi \leq 1$, $\sigma i=1\ n\ pi = 1$ Ký hiệu phân phối này là $p = (p1,p2,\ldots,pn)$. Entropy của phân phối này được định nghĩa là

$$H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log(p_i)$$

Thuật toán ID3

- Tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt.
    + việc này đạt được bằng cách chọn ra thuộc tính sao cho nếu dùng thuộc tính đó để phân chia, entropy tại mỗi bước giảm đi một lượng lớn nhất.

- Xét một bài toán với C class khác nhau:
    + GS ta đang làm việc với một non-leaf node với các điểm dữ liệu tạo thành một tập S với số phần tử là $|S| = N$.
    + Giả sử thêm rằng trong số N điểm dữ liệu này, $Nc, c = 1,2,\ldots,C$ điểm thuộc vào class c. Xác suất để mỗi điểm dữ liệu rơi vào một class c được xấp xỉ bằng Nc/N (maximum likelihood estimation).
    +entropy tại:

$$H(\mathcal{S}) = -\sum_{c=1}^{C} \frac{N_c}{N} \log\left(\frac{N_c}{N}\right)$$

- GS thuộc tính được chọn là x. Dựa trên x, các điểm dữ liệu trong S được phân ra thành K child node $S1,S2,\ldots,SK$ với số điểm trong mỗi child node lần lượt là $m1,m2,\ldots,mK$. Ta định nghĩa

$$H(x, \mathcal{S}) = \sum_{k=1}^{K} \frac{m_k}{N} H(\mathcal{S}_k)$$

- Ta định nghĩa information gain dựa trên thuộc tính x

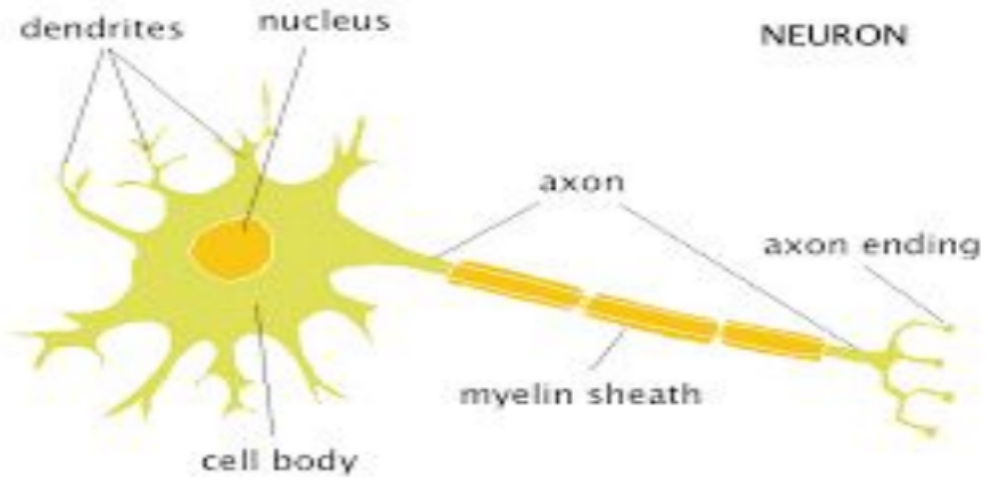
$$G(x, \mathcal{S}) = H(\mathcal{S}) - H(x, \mathcal{S})$$

- Tại mỗi node, thuộc tính được chọn được xác định dựa trên:

$$x^* = \arg\max_x G(x, \mathcal{S}) = \arg\min_x H(x, \mathcal{S})$$

4) Phương pháp Neural Network

a) Định nghĩa



- Soma: thân nơron, tiếp nhận hoặc phát ra các xung động thần kinh
- Dendrites: dây thần kinh vào, đưa tín hiệu tới nơron
- Axon: đầu dây thần kinh ra, nối với dây thần kinh vào hoặc tới nhân tế bào của nơron khác thông qua khớp nối
- Synapse: khớp để kích hoạt hoặc kích thích thông tin

b) Đầu vào

Cho tập dữ liệu huấn luyện gồm N mẫu. Mỗi mẫu là một cặp (x_i,y_i):

• x_i: vector đặc trưng

• y_i: giá trị của vector đặc trưng x

Tập dữ liệu (Xtrain, ytrain), kiến trúc mạng nơ ron (số lớp ẩn, số nơ ron của mỗi lớp ẩn), hàm kích hoạt (activation function), hàm mất mát (loss function)

c) Đầu ra

y = f(Net) = f(∑wi xi)

f được gọi là hàm truyền (transfer function) hay kích hoạt (activation function)
Bộ vector trọng số của các liên kết giữa các nơ ron (W) để hàm mất mát đạt giá trị tối ưu

d) Phương pháp

- Phương pháp phổ biến nhất để tối ưu MLP vẫn là Gradient Descent (GD)
- Để áp dụng GD, chúng ta cần tính được gradient của hàm mất mát theo từng ma trận trọng số W, và vector bias b

5) Phương pháp Logistic Regression

a) Định nghĩa

- Logistic regression:
  - giống với linear regression ở khía cạnh đầu ra là số thực,
  - và giống với PLA ở việc đầu ra bị chặn (trong đoạn [0,1]).
- Mặc dù trong tên có chứa từ regression, logistic regression thường được sử dụng nhiều hơn cho các bài toán classification.

b) Đầu vào

Cho tập dữ liệu huấn luyện gồm N mẫu. Mỗi mẫu là một cặp (x_i,y_i):

- x_i: vector đặc trưng
- y_i: giá trị của vector đặc trưng x

c) Đầu ra

Đầu ra có thể được thể hiện dưới dạng xác suất (probability).

$$f(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x})$$

d) Xây dựng hàm mất mát

- Xây dựng hàm mất mát

+ Với mô hình (các activation màu xanh biển và lá), ta có thể giả sử rằng xác suất để một điểm dữ liệu x rơi vào class 1 là **f(wT x)** và rơi vào class 0 là **1-.f(wT x)**

+ Với mô hình được giả sử này, với một điểm dữ liệu training (đã biết đầu ra viết như sau:

$$P(y_i = 1|\mathbf{x}_i; \mathbf{w}) = f(\mathbf{w}^T\mathbf{x}_i)$$
$$P(y_i = 0|\mathbf{x}_i; \mathbf{w}) = 1 - f(\mathbf{w}^T\mathbf{x}_i)$$

+ Mục đích là tìm các hệ số **w** sao cho **f(wT x)** là càng gần với 1 càng tốt với các điểm dữ liệu thuộc class 1 và càng gần với 0 càng tốt với những điểm thuộc class 0

- Hàm mất mát:

+ Ký hiệu $z_i = f(\mathbf{w}^T\mathbf{x}_i)$ vàviết gộp lại hai biểu thức ta có:

$$P(y_i|\mathbf{x}_i; \mathbf{w}) = z_i^{y_i}(1 - z_i)^{1-y_i}$$

+ Biểu thức ở dưới tương đương với hai biểu thức ở trên vì:

khi $y_i = 1$, phần thứ hai của vế phải sẽ triệt tiêu,

khi $y_i = 0$, phần thứ nhất sẽ bị triệt tiêu!

+ Chúng ta muốn mô hình gần với dữ liệu đã cho nhất, tức xác suất $P(y_i|\mathbf{x}_i; \mathbf{w})$ đạt giá trị cao nhất.

- Xét toàn bộ training set với $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbf{R}^{d\times N}$ và $y = [y_1, y_2, \dots, y_N]$, chúng ta cần tìm **w** để $P(\mathbf{v}|\mathbf{X}; \mathbf{w})$ đạt giá trị lớn nhất hay

$$\mathbf{w} = \arg\max_{\mathbf{w}} P(\mathbf{y}|\mathbf{X}; \mathbf{w})$$

e) Phương pháp

- Sau khi có w, ta dự đoán nhãn

$$\hat{y} = sigmoid(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d)$$

- Vớibài toán phân lớp: Nếu >0.5 thì x thuộc về lớp 1, ngược lại , x thuộc về lớp 0

- **Tối ưu hàm mất mát**
  + Chúng ta lại sử dụng phương pháp GD để tìm w
  + Công thức cập nhật (theo thuật toán GD) cho logistic regression là:

$$\mathbf{w} = \mathbf{w} + \eta(y_i - z_i)\mathbf{x}_i$$

II) Thực nghiệm

a) Hiểu về dữ liệu

+ Hiểu các thuộc tính:

STT	Tên thuộc tính	Ý nghĩa thuộc tính
1	gender	Giới tính của bệnh nhân
2	age	Tuổi của đối tượng
3	hypertension	Tăng huyết áp: + 0: nếu bệnh nhân không bị tăng huyết áp + 1: nếu bệnh nhân tăng bị huyết áp
4	heart_disease	Bệnh tim + 0: Nếu bệnh nhân không mắc bất kì bệnh tim nào. + 1: Nếu bệnh nhân mắc bệnh tim
5	ever_married	Tình trạng hôn nhân + No: Nếu chưa kết hôn + Yes: Nếu chưa kết hôn
6	work_type	Loại công việc
7	Residence_type	Loại nơi cư trú
8	avg_glucose_level	Mức glucose trung bình trong máu
9	bmi	Chỉ số BMI
10	smoking_status	Tình trạng hút thuốc
11	stroke	Nguy cơ đột quỵ + 0: Không có nguy cơ đột quỵ + 1: Có nguy cơ đột quỵ

b) **Kết quả thu được**

	Perceptron	SVM	ID3	Neural Network	Logistic Regression
Tỷ lệ dự đoán đúng	0.9732441471571907	0.9732441471571907	0.9438127090301003	0.9719063545150501	0.9678929765886287
Precision	0.9739600228185367	0.9739600228185367	0.9488003858365327	0.9471692872199584	0.9503597851423938
Recall	0.9732441471571907	0.9732441471571907	0.9438127090301003	0.9719063545150501	0.9678929765886287
F1-score	0.9600476163482795	0.9600476163482795	0.9462892305010184	0.9861342421368381	0.9584246809091529

=> Phương pháp phù hợp nhất cho bài toán là Perceptron, với tỷ lệ dự đoán đúng là: 0.9732441471571907 **%**

TÀI LIỆU THAM KHẢO

- Slide bài giảng

- Ý tưởng và tệp dữ liệu sử dụng:

https://www.kaggle.com/datasets/jillanisofttech/brain-stroke-dataset