

CSE: Faculty of Computer Science and Engineering  
Thuyloi University

# Hồi quy logistic

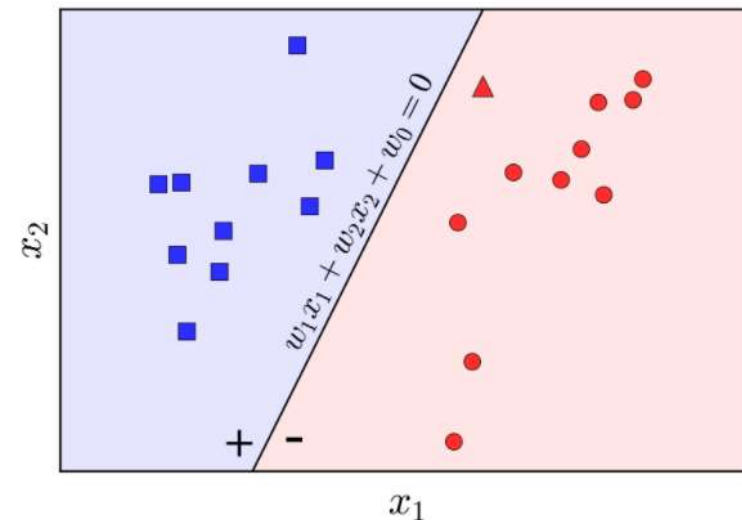
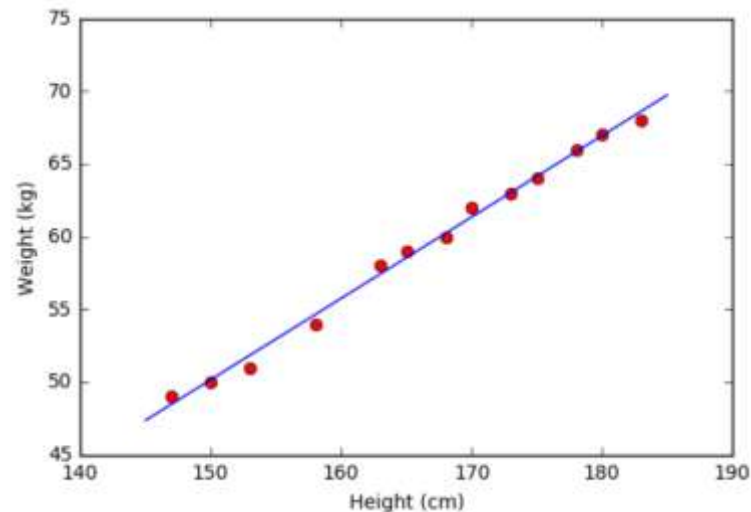
Trình bày: PGS.TS Nguyễn Hữu Quỳnh

# Giới thiệu

- Hai mô hình tuyến tính: Linear Regression và Perceptron Learning Algorithm (PLA) đều có chung một dạng:

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

- trong đó  $f(\mathbf{w}^T \mathbf{x})$  là hàm kích hoạt (*activation function*)
- Với linear regression thì  $f(\mathbf{w}^T \mathbf{x}) = \mathbf{w}^T \mathbf{x}$ , với PLA thì  $f(\mathbf{w}^T \mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x})$



## Giới thiệu

- Trong linear regression, tích vô hướng  $\mathbf{w}^T \mathbf{x}$  được trực tiếp sử dụng để dự đoán output  $y$ , loại này phù hợp khi dự đoán một giá trị thực của đầu ra không bị chặn trên và dưới.
- Trong PLA, đầu ra chỉ nhận một trong hai giá trị 1 hoặc  $-1$ , phù hợp với các bài toán *binary classification*.

# Giới thiệu

- Trong phần này, ta sẽ giới thiệu mô hình có tên là *logistic regression*:
  - Đầu ra có thể được thể hiện dưới dạng xác suất (probability). Ví dụ: xác suất thi đỗ nếu biết thời gian ôn thi, xác suất ngày mai có mưa dựa trên những thông tin đo được trong ngày hôm nay,...
  - *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.

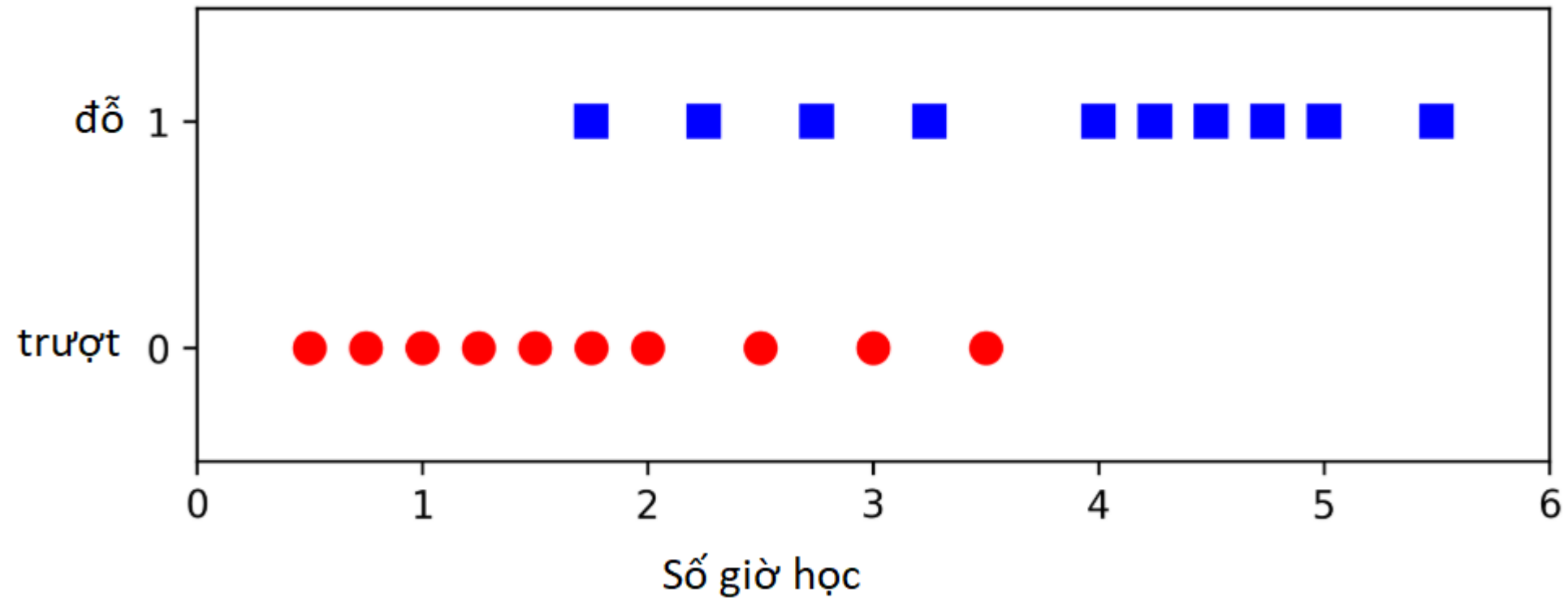
# Giới thiệu

Ví dụ: Một nhóm 20 sinh viên dành thời gian trong khoảng từ 0 đến 6 giờ cho việc ôn thi và có kết quả như bảng. Hỏi rằng, em sinh viên ôn x giờ là đỗ hay trượt.

Tập huấn luyện

Hours	Pass	Hours	Pass
.5	0	2.75	1
.75	0	3	0
1	0	3.25	1
1.25	0	3.5	0
1.5	0	4	1
1.75	0	4.25	1
1.75	1	4.5	1
2	0	4.75	1
2.25	1	5	1
2.5	0	5.5	1

# Giới thiệu



- Cả linear regression và PLA đều không phù hợp với bài toán này, chúng ta cần một mô hình *flexible* hơn.

# Mô hình Logistic Regression

Đầu ra dự đoán của:

- Linear Regression:

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

- PLA:

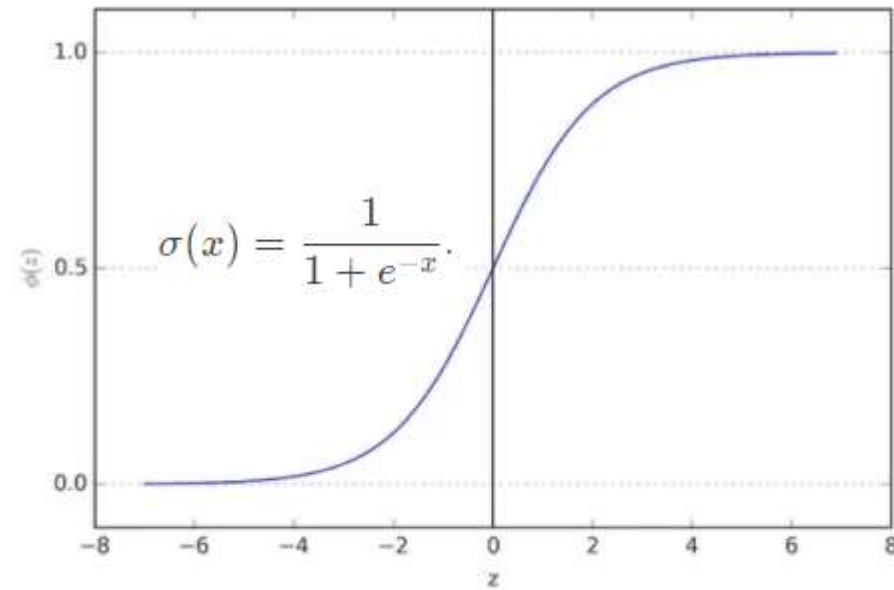
$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x})$$

- Logistic regression

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

# Mô hình Logistic Regression

Hàm sigmoid:





# Mô hình Logistic Regression

Nhận xét hàm sigmoid :

- Nó là liên tục, nhận giá trị thực trong khoảng  $(0,1)$ .
- Nó có đạo hàm tại mọi điểm (để áp dụng gradient descent)

# 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  $\mathbf{x}$  rơi vào class 1 là  $f(\mathbf{w}^T \mathbf{x})$  và rơi vào class 0 là  $1 - f(\mathbf{w}^T \mathbf{x})$ .
- Với mô hình được giả sử này, với một điểm dữ liệu training (đã biết đầu ra  $y$ ), ta có thể viết như sau:

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

- Mục đích là tìm các hệ số  $\mathbf{w}$  sao cho  $f(\mathbf{w}^T \mathbf{x}_i)$  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  $z_i = f(\mathbf{w}^T \mathbf{x}_i)$  và viết gộp lại hai biểu thức

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

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

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.

# Hàm mất mát

- 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  $\mathbf{w}$  để
  - $P(\mathbf{y}|\mathbf{X}; \mathbf{w})$  đạt giá trị lớn nhất hay  $\mathbf{w} = \arg \max_{\mathbf{w}} P(\mathbf{y}|\mathbf{X}; \mathbf{w})$
- Bài toán tìm tham số để mô hình gần với dữ liệu nhất trên đây có tên gọi chung là bài toán *maximum likelihood estimation* với hàm số phía sau argmax được gọi là *likelihood function*.

# Hàm mất mát

- GS các điểm dữ liệu được sinh ra một cách ngẫu nhiên độc lập với nhau (independent), ta có thể viết:

$$\begin{aligned} P(\mathbf{y}|\mathbf{X}; \mathbf{w}) &= \prod_{i=1}^N P(y_i|\mathbf{x}_i; \mathbf{w}) \\ &= \prod_{i=1}^N z_i^{y_i} (1 - z_i)^{1-y_i} \end{aligned}$$

- Trực tiếp tối ưu hàm số này theo  $\mathbf{w}$  khó

# Hàm mất mát

- Một phương pháp thường được sử dụng đó là lấy logarit tự nhiên (cơ số e) của *likelihood function* biến phép nhân thành phép cộng và để tránh việc số quá nhỏ.
- Sau đó lấy ngược dấu để được một hàm và coi nó là hàm mất mát. Lúc này bài toán tìm giá trị lớn nhất (maximum likelihood) trở thành bài toán tìm giá trị nhỏ nhất của hàm mất mát (hàm này còn được gọi là negative log likelihood):

$$\begin{aligned} J(\mathbf{w}) &= -\log P(\mathbf{y}|\mathbf{X}; \mathbf{w}) \\ &= -\sum_{i=1}^N (y_i \log z_i + (1 - y_i) \log(1 - z_i)) \end{aligned}$$

# Tối ưu hàm mất mát

- Chúng ta lại sử dụng phương pháp GD) để tìm  $\mathbf{w}$
- Công thức cập nhật (theo thuật toán GD) cho logistic regression là:

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

# Dự đoán lớp của điểm $\mathbf{x}$

- Sau khi có  $\mathbf{w}$ , ta dự đoán nhãn

$$\hat{y} = \text{sigmoid}(w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d)$$

- Với bài toán phân lớp:

Nếu  $\hat{y} > 0.5$  thì  $\mathbf{x}$  thuộc về lớp 1,

ngược lại ,  $\mathbf{x}$  thuộc về lớp 0



# Ví dụ với Python

- Chạy chương trình với dữ liệu huấn luyện:

Hours	Pass	Hours	Pass
.5	0	2.75	1
.75	0	3	0
1	0	3.25	1
1.25	0	3.5	0
1.5	0	4	1
1.75	0	4.25	1
1.75	1	4.5	1
2	0	4.75	1
2.25	1	5	1
2.5	0	5.5	1