Logistic, Softmax Classifier

FIC - Ta Dang Khoa

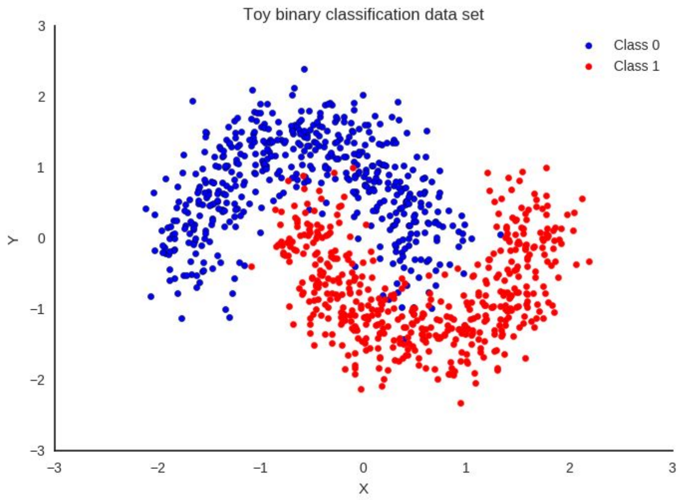
Nội dung

- Giới thiệu về mô hình phân loại
- Phân loại hai lớp
    - Logistic Regression
- Phân loại nhiều lớp
    - Softmax Regression
- Code ví dụ

Giới thiệu

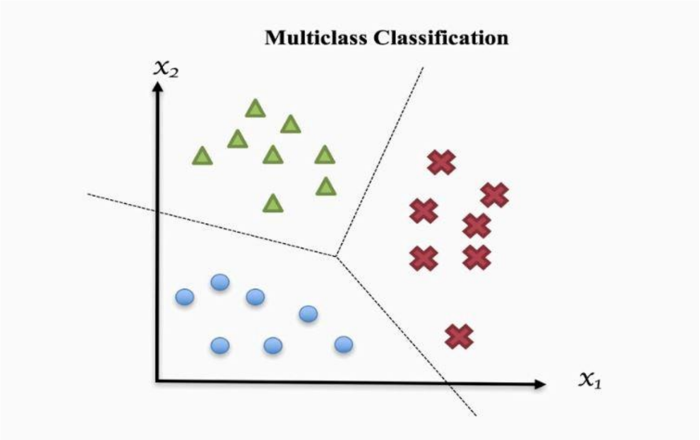
Phân loại hai lớp

- Binary classification: data được chia làm 2 lớp
- Ví dụ: True or False, Cat or Dog, Man Woman, …
- Label được lưu dạng 0, 1



Giới thiệu
Phân loại nhiều lớp

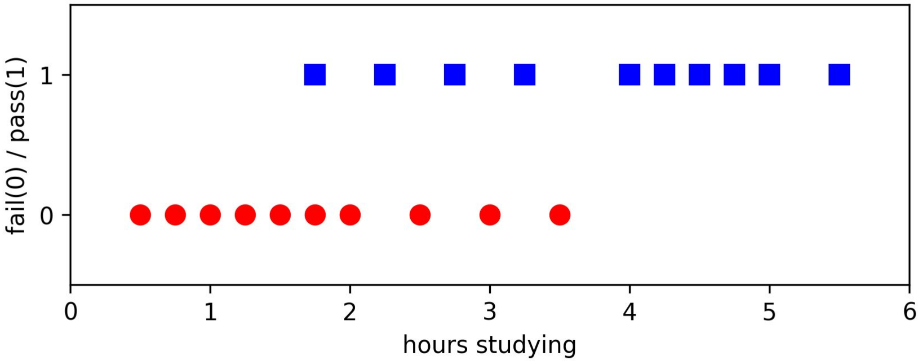
- Multiclass classification: data được chia thành nhiều hơn 2 lớp
- Ví dụ: Phân loại các số từ 0-9
- Label được lưu dưới dạng "one-hot"



Logistic Regression

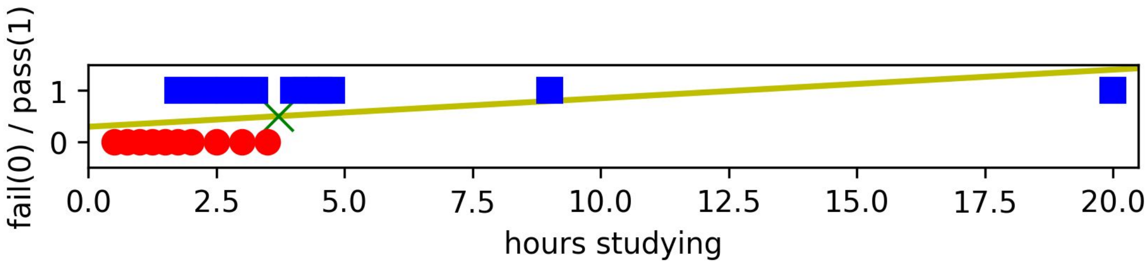
Bài toán (dữ liệu 1 chiều)

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



Logistic Regression

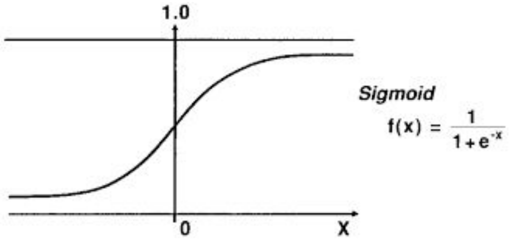
Tại sao Linear Regression không phù hợp?



- Vì là hàm tuyến tính nên khoảng giá trị là vô cùng, trong khi đó ta lại muốn output thuộc khoảng giá trị [0, 1]
- Bị ảnh hưởng bởi nhiễu

Logistic Regression
Mô hình

- Hàm giả thiết của chúng ta sẽ có dạng: f(wT.x). Với f là một hàm phi tuyến có output thuộc khoảng [0, 1]



- Chúng ta sẽ tìm cách tối ưu một hàm "Likelihood" (Đọc thêm về "Maximum a Posteriori" và "Maximum Likelihood Estimation")

$$\theta = \max_{\theta} p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \theta)$$

Logistic Regression
Mô hình - Xây dựng hàm mất mát

- Ta có thể giả sử rằng xác suất để một điểm dữ liệu x rơi vào class 1 là $f(\mathbf{w}^T\mathbf{x})$, rơi vào class 0 là $1 - f(\mathbf{w}^T\mathbf{x})$
- Ta có thể viết ngắn gọn:

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

- Đặt $z_i = f(\mathbf{w}^T\mathbf{x}_i)$ ta viết gộp 2 biểu thức lại thành:

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

- Vậy ta cần tối ưu hàm xác xuất sao cho tỉ lệ đoán đúng là cao nhất có nghĩa là ta cần tìm `w`:

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

$$P(\mathbf{y}|\mathbf{X}; \mathbf{w}) = \prod_{i=1}^{N} z_i^{y_i}(1 - z_i)^{1-y_i}$$

- Tuy nhiên do z_i thuộc [0, 1] nên với số mẫu lớn giá trị sẽ về 0. Do đó ta thêm hàm `-log` để chuyển tích thành tổng
- Ta thu được hàm loss:

$$J(\mathbf{w}) = -\sum_{i=1}^{N}(y_i \log z_i + (1 - y_i)\log(1 - z_i))$$

Logistic Regression

Mô hình - Tối ưu hàm mất mát

- Hàm mất mát với một điểm dữ liệu (x_i, y_i):

$$J(\mathbf{w}; \mathbf{x}_i, y_i) = -(y_i \log z_i + (1 - y_i)\log(1 - z_i))$$

- Với đạo hàm:

$$\frac{\partial J(\mathbf{w}; \mathbf{x}_i, y_i)}{\partial \mathbf{w}} = -(\frac{y_i}{z_i} - \frac{1 - y_i}{1 - z_i})\frac{\partial z_i}{\partial \mathbf{w}} = \frac{z_i - y_i}{z_i(1 - z_i)}\frac{\partial z_i}{\partial \mathbf{w}}$$

- Đặt $s = \mathbf{w}^T\mathbf{x}$:

$$\frac{\partial z_i}{\partial \mathbf{w}} = \frac{\partial z_i}{\partial s}\frac{\partial s}{\partial \mathbf{w}} = \frac{\partial z_i}{\partial s}\mathbf{x}$$

- Ta muốn rút gọn mẫu số $z(z - 1)$ nên ta sẽ tìm hàm số $z = f(s)$ thỏa mãn:

$$\frac{\partial z}{\partial s} = z(1 - z)$$

- Giải phương trình trên ta thu được $z = f(s)$ sau:

$$f(s) = \frac{1}{1 + e^{-s}}$$

- Đến đây ta đã hiểu tại sao lại sử dụng hàm `signmoid`
- Cuối cùng ta thu được đạo hàm:

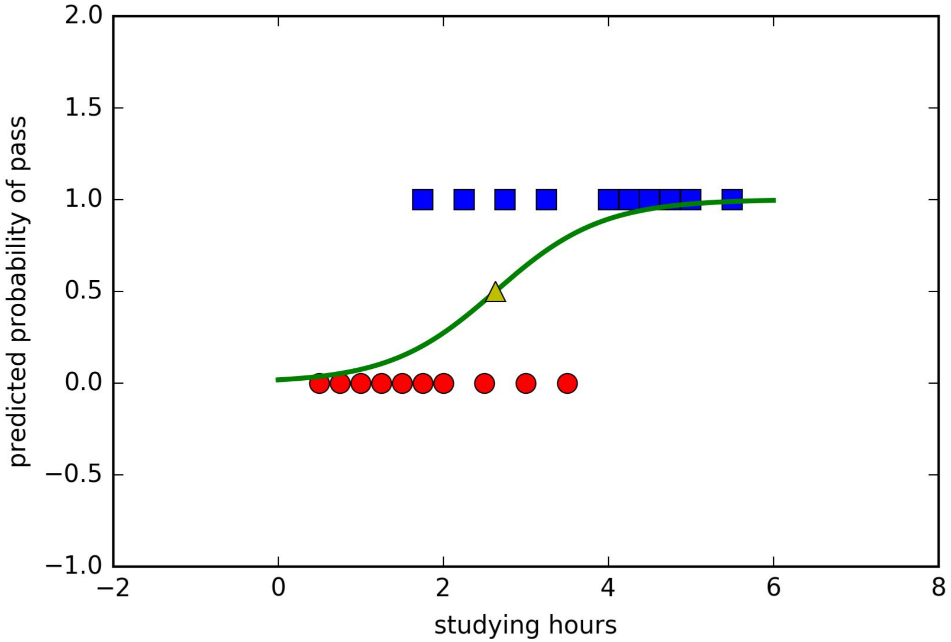
$$\frac{\partial J_i}{\partial \mathbf{w}} = (z_i - y_i)x_i$$

- Công thức cập nhật:

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

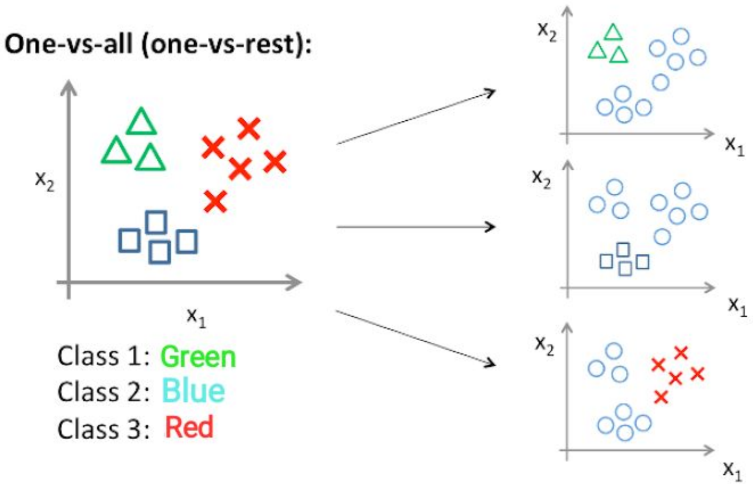
Logistic Regression

Kết quả



Softmax Regression

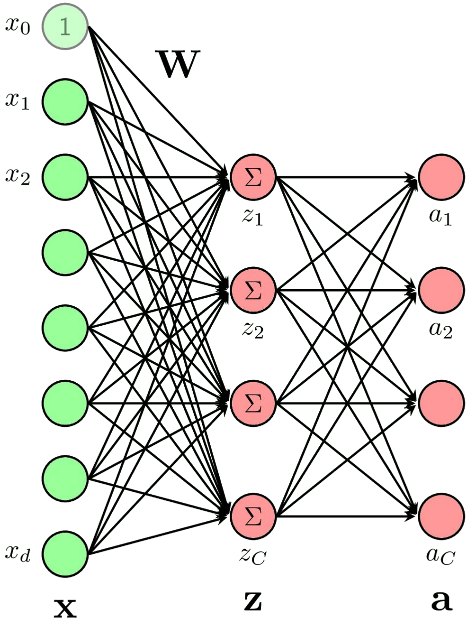
Ý tưởng - Phân tích

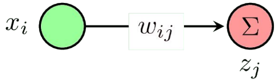


- Xác suất tại từng điểm tổng không bằng 1
- Những điểm tại vùng giữa không phân biệt được

Softmax Regression

Mô hình





w_{0j}: biases, don't forget!

d: data dimension

C: number of classes

$\mathbf{x} \in \mathbb{R}^{d+1}$

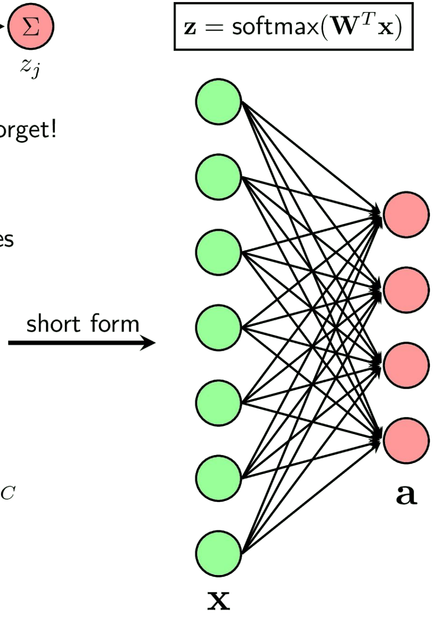
$\mathbf{W} \in \mathbb{R}^{(d+1)\times C}$

$z_i = \mathbf{w}_i^T \mathbf{x}$

$\mathbf{z} = \mathbf{W}^T \mathbf{x} \in \mathbb{R}^C$

$\mathbf{a} = \text{softmax}(\mathbf{z}) \in \mathbb{R}^C$

$a_i > 0, \quad \sum_{i=1}^{C} a_i = 1$



Softmax Regression
Hàm Softmax

Chúng ta cần một mô hình xác suất sao cho với mỗi input x, a_i thể hiện xác suất để input đó rơi vào class i. Vậy điều kiện cần là các a_i phải dương và tổng của chúng bằng 1.
Với $z = \mathbf{w}^T\mathbf{x}$ ta có hàm `softmax` thỏa mãn yêu cầu trên:

$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^{C}\exp(z_j)}, \quad \forall i = 1, 2, \ldots, C$$

Softmax Regression
Hàm mất mát

Ta giả sử rằng công thức thể hiện xác suất để một điểm dữ liệu x rơi vào class thứ i nếu biết tham số mô hình (ma trận trọng số) là W:

$$P(y_k = i | \mathbf{x}_k; \mathbf{W}) = a_i$$

Xây dựng giống như `Logistic Regression`. Ta có hàm mất mát cho Softmax Regression như sau:

$$J(\mathbf{W}; \mathbf{X}, \mathbf{Y}) = -\sum_{i=1}^{N} \sum_{j=1}^{C} y_{ji} \log(a_{ji})$$

$$= -\sum_{i=1}^{N} \sum_{j=1}^{C} y_{ji} \log\left(\frac{\exp(\mathbf{w}_j^T \mathbf{x}_i)}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{x}_i)}\right)$$

Softmax Regression
Tối ưu hàm mất mát

Hàm mất mát với một điểm dữ liệu (x_i, y_i):

$$\begin{aligned}
J_i(\mathbf{W}) &= J(\mathbf{W}; \mathbf{x}_i, \mathbf{y}_i) \\
&= -\sum_{j=1}^{C} y_{ji} \log\left(\frac{\exp(\mathbf{w}_j^T \mathbf{x}_i)}{\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{x}_i)}\right) \\
&= -\sum_{j=1}^{C} \left(y_{ji} \mathbf{w}_j^T \mathbf{x}_i - y_{ji} \log\left(\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{x}_i)\right)\right) \\
&= -\sum_{j=1}^{C} y_{ji} \mathbf{w}_j^T \mathbf{x}_i + \log\left(\sum_{k=1}^{C} \exp(\mathbf{w}_k^T \mathbf{x}_i)\right)
\end{aligned}$$

Trong biến đổi cuối cùng ta thấy tổng y bằng 1 do chỉ có một y là 1 còn còn lại là 0

Với đạo hàm:

$$\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}} = \left[\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{w}_1}, \frac{\partial J_i(\mathbf{W})}{\partial \mathbf{w}_2}, \dots, \frac{\partial J_i(\mathbf{W})}{\partial \mathbf{w}_C}\right]$$

Softmax Regression

Tối ưu hàm mất mát

Trong đó, gradient theo từng cột có thể tính được dựa theo:

$$\begin{aligned}\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{w}_j} &= -y_{ji}\mathbf{x}_i + \frac{\exp(\mathbf{w}_j^T\mathbf{x}_i)}{\sum_{k=1}^{C}\exp(\mathbf{w}_k^T\mathbf{x}_i)}\mathbf{x}_i \\ &= -y_{ji}\mathbf{x}_i + a_{ji}\mathbf{x}_i = \mathbf{x}_i(a_{ji} - y_{ji}) \\ &= e_{ji}\mathbf{x}_i \text{ (where } e_{ji} = a_{ji} - y_{ji})\end{aligned}$$

Kết hợp 2 công thức ta được:

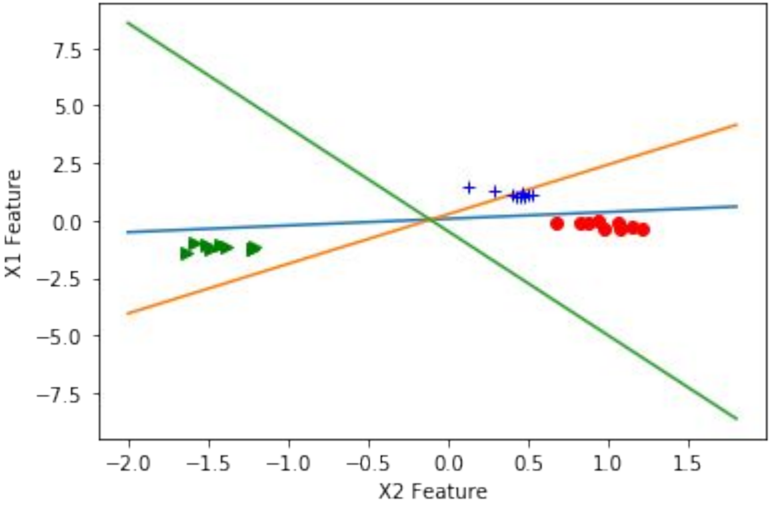
$$\frac{\partial J_i(\mathbf{W})}{\partial \mathbf{W}} = \mathbf{x}_i[e_{1i}, e_{2i}, \dots, e_{Ci}] = \mathbf{x}_i\mathbf{e}_i^T$$

Công thức cập nhật:

$$\mathbf{W} = \mathbf{W} - \alpha\mathbf{x}_i(\mathbf{a}_i - \mathbf{y}_i)^T$$

Có thể thấy được nếu ta có số class là 2 thì công thức a_i sẽ giống với hàm `signmoid`

Softmax Regression
Kết quả



Bài tập

- Đọc hiểu lại về những công thức
- Hiểu cách vector hóa công thức
- Hiểu shape của dữ liệu qua từng bước
- Tìm hiểu các *'metric'* trong bài toán regression và classification
    - MSE
    - ACC
    - F1