Logistic Regression & Assignment 1 Handout

Machine Learning 1 UE (INP.33761UF)

Thomas Wedenig

Apr 17, 2024

Institute of Theoretical Computer Science
Graz University of Technology, Austria

Table of contents

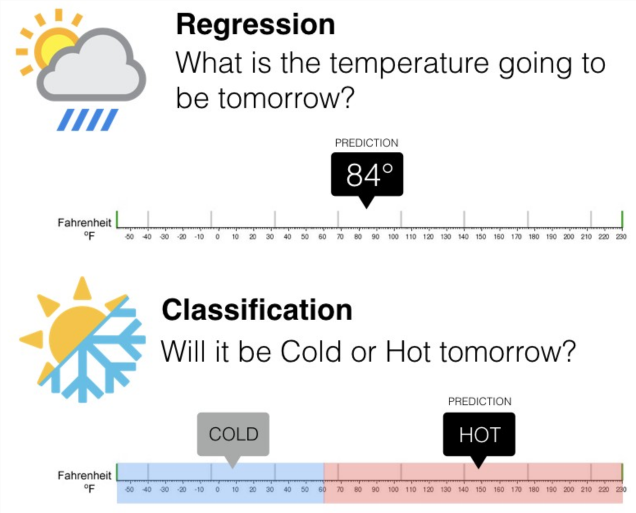
Logistic Regression - Recap

- We have a dataset $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$
- $\mathbf{x}^{(i)} \in \mathbb{R}^D$ and $y^{(i)} \in \mathcal{C}$
- $\mathcal{C}$ contains **discrete classes**, e.g. $\mathcal{C} = \{\texttt{cat}, \texttt{dog}, \texttt{rabbit}\}$
- **Binary Classification**: $\mathcal{C}$ contains **2 elements**

(Informal) Binary Classification Problem

Let $\mathcal{C} = \{-1, 1\}$. Given $\mathcal{D}$, find a function $g : \mathbb{R}^D \to \mathcal{C}$ that "predicts the labels well":

$$g(\mathbf{x}) \approx y$$



https://www.springboard.com/blog/data-science/regression-vs-classification/

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad
0	68.95	35	61833.90	256.09	0	0
1	80.23	31	68441.85	193.77	1	0
2	69.47	26	59785.94	236.50	0	0
3	74.15	29	54806.18	245.89	1	0
4	68.37	35	73889.99	225.58	0	0
5	59.99	23	59761.56	226.74	1	0
6	88.91	33	53852.85	208.36	0	0
7	66.00	48	24593.33	131.76	1	1

- **Logistic Regression**: A particular method for **binary classification problems**
- We want to **model the conditional probability** $p(y = 1 \mid \mathbf{x}) \in [0, 1]$
- Given $\mathbf{x}$, we use a **linear function**

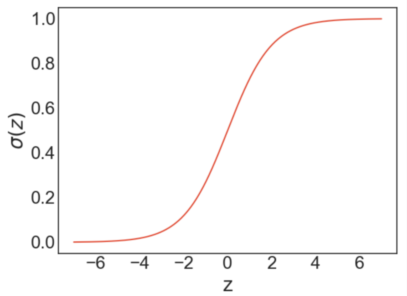
$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \phi(\mathbf{x})$$

  to compute a **logit** $\in (-\infty, \infty)$
- To convert the logit to a probability, we use the **(logistic) sigmoid function**:

$$p(y = 1 \mid \mathbf{x}) = \sigma(f_{\boldsymbol{\theta}}(\mathbf{x})), \qquad \sigma(z) = \frac{1}{1 + \exp(-z)}$$

Logistic Regression (cont.)

$$p(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\theta}^T \phi(\mathbf{x})), \qquad \sigma(z) = \frac{1}{1 + \exp(-z)}$$



- σ maps a real number $\in (-\infty, \infty)$ to the interval $[0, 1]$
- σ is **differentiable everywhere** and its gradients are **informative** 😍
- Finally, use the predicted probability to **make a decision**
    - If $p(y = 1 \mid \mathbf{x}) \geq 0.5$, predict $\hat{y} = +1$
    - If $p(y = 1 \mid \mathbf{x}) < 0.5$, predict $\hat{y} = -1$

- How to interpret the output of the **linear function** $f_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \phi(\mathbf{x})$?
- Given $\mathbf{x}$, let $\pi_{\mathbf{x}} = p(y = 1 \mid \mathbf{x})$ be the **predicted probability of success**
- $f_{\boldsymbol{\theta}}(\mathbf{x})$ models the **log-odds**:

$$\overbrace{f_{\boldsymbol{\theta}}(\mathbf{x})}^{l} = \log\left(\frac{\pi_{\mathbf{x}}}{1 - \pi_{\mathbf{x}}}\right)$$

$$\exp(l) = \frac{\pi_x}{1-\pi_x}$$

$$\exp(l) - \pi_x \cdot \exp(l) = \pi_x$$

$$\exp(l) = \pi_x + \pi_x \cdot \exp(l)$$

$$\exp(l) = \pi_x(1 + \exp(l))$$

$$\boxed{\pi_x = \frac{\exp(l)}{1+\exp(l)}}$$

- From which follows that

$$\pi_{\mathbf{x}} = \sigma(f_{\boldsymbol{\theta}}(\mathbf{x}))$$

divide by exp(l)

$$\pi_x = \frac{\exp(l)}{1+\exp(l)} = \frac{1}{\frac{1}{\exp(l)} + 1} = \frac{1}{\exp(-l)+1} = \sigma(l)$$

- How to interpret the output of the **linear function** $f_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \phi(\mathbf{x})$?
- Given $\mathbf{x}$, let $\pi_{\mathbf{x}} = p(y = 1 \mid \mathbf{x})$ be the **predicted probability of success**
- $f_{\boldsymbol{\theta}}(\mathbf{x})$ models the **log-odds**:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \log\left(\frac{\pi_{\mathbf{x}}}{1 - \pi_{\mathbf{x}}}\right)$$

- From which follows that

$$\pi_{\mathbf{x}} = \sigma(f_{\boldsymbol{\theta}}(\mathbf{x}))$$

Assumption of Logistic Regression !

We assume a **linear** relationship between $\phi(\mathbf{x})$ and the **log-odds** of the event happening.

- (Binary) Cross-Entropy Loss Function:

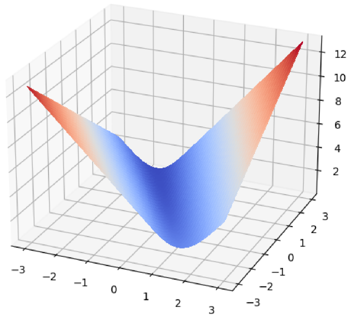
$$\mathcal{L}_{CE}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = 1] \log \pi_{\mathsf{x}^{(i)}} + \mathbb{1}[y^{(i)} = -1] \log (1 - \pi_{\mathsf{x}^{(i)}})$$

with $\pi_{\mathsf{x}^{(i)}} = \sigma(\boldsymbol{\theta}^T \phi(\mathsf{x}^{(i)}))$

- (Binary) Cross-Entropy Loss Function:

$$\mathcal{L}_{CE}(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[y^{(i)} = 1] \log \pi_{\mathbf{x}^{(i)}} + \mathbb{1}[y^{(i)} = -1] \log (1 - \pi_{\mathbf{x}^{(i)}})$$

with $\pi_{\mathbf{x}^{(i)}} = \sigma(\boldsymbol{\theta}^T \phi(\mathbf{x}^{(i)}))$



- No closed form solution for $\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}} \mathcal{L}_{CE}(\boldsymbol{\theta})$
- But $\mathcal{L}_{CE}(\boldsymbol{\theta})$ is **differentiable** and **convex** in $\boldsymbol{\theta}$ 😏
- Use some **optimizer** to solve for $\boldsymbol{\theta}^*$
  - e.g., Gradient Descent

Regularization / Penalty

- In ML, we **usually prefer "simple" models**
- We can encode this into the **loss function**
- **Assumption**: Simple models have parameter values in $\boldsymbol{\theta}$ that **are close to** 0
- Thus, we may **penalize** the model if the following grows large:

$$\sum_{i=1}^{D} \theta_i^2 = \|\boldsymbol{\theta}\|_2^2$$

- We now minimize

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{CE}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2$$

where λ controls the **strength of regularization**

Logistic Regression - Demo

Assignment 1 Handout