T05: Multiclass Logistic Regression

MATH 4432 Statistical Machine Learning

WANG Zhiwei

MATH, HKUST

2022-10-11

Let's first recall what we have learned in class!

Logistic regression for classification

- Training set $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, where $y_i \in \{0, 1\}$.
- Probabilistic model

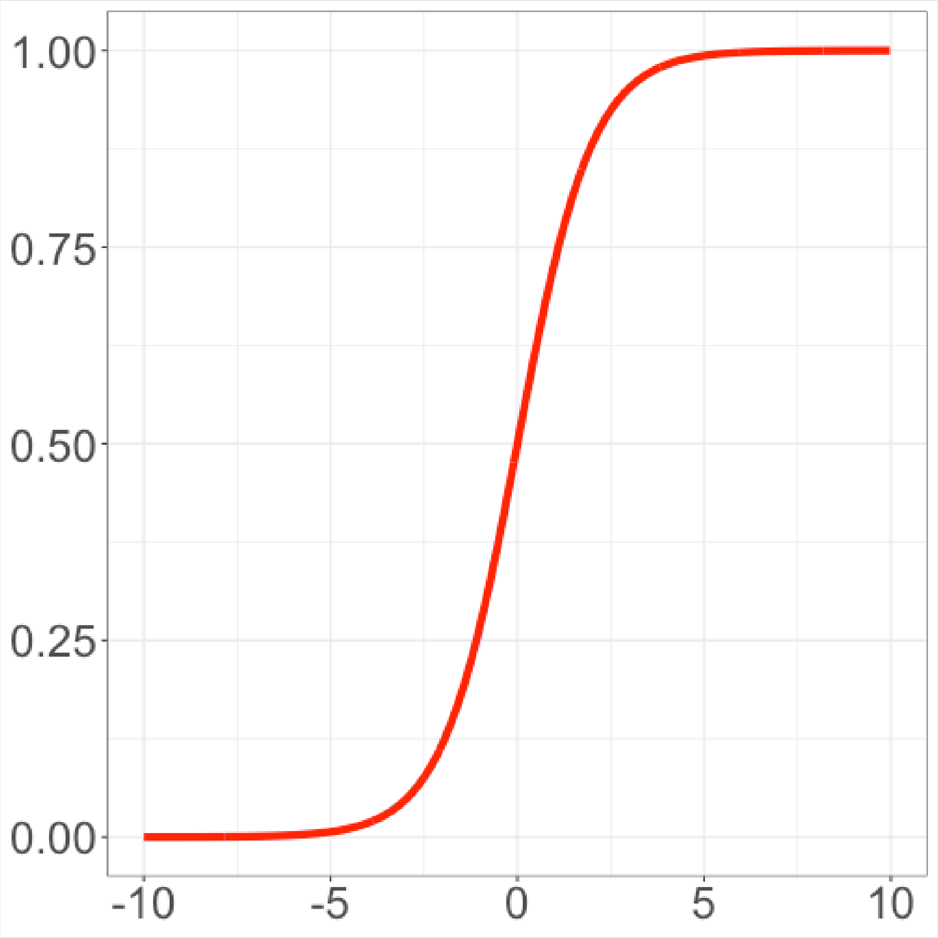
$$p(y \mid \mathbf{x}, \beta) = \mathrm{Ber}\big(y \mid \sigma\left(\beta^\top \mathbf{x}\right)\big)$$

  - $\sigma(z)$ is the sigmoid/logistic/logit function.

$$\sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{e^z}{e^z + 1}$$

  - It maps $\mathbb{R}$ to $(0, 1)$.

Logit function



Joint probability

- Recall that, the likelihood is the joint probability function of joint density function of the data.
- Here, we have independent observations $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, each follows the (conditional) distribution

$$\Pr(y_i = 1 \mid \mathbf{x}_i) = \frac{1}{1 + \exp(-\beta^T \mathbf{x}_i)} = 1 - \Pr(y_i = 0 \mid \mathbf{x}_i)$$

- So, the joint probability function is

$$\prod_{i=1,\ldots,n; y_i=1} \Pr(y_i = 1 \mid \mathbf{x}_i) \prod_{i=1,\ldots,n; y_i=0} \Pr(y_i = 0 \mid \mathbf{x}_i)$$

which can be conveniently written as

$$\prod_{i=1}^{n} \frac{\exp\left(y_i \beta^T \mathbf{x}_i\right)}{1 + \exp(\beta^T \mathbf{x}_i)}$$

The maximum likelihood estimation

- The likelihood function is the same as the joint probability function, but viewed as a function of β. The log-likelihood function is

$$\ell = \sum_{i=1}^{n} \left[y_i \beta^T x_i - \log\big(1 + \exp\big(\beta^T \mathbf{x}_i\big)\big) \right]$$

- Unlike linear regression, we can no longer write down the MLE in closed form. Instead, we need to use optimization algorithms to compute it.
    - Gradient descent
    - Newton's method

Now let's go to multiclass logistic regression!

A set of independent binary regressions

We now extend the two-class logistic regression approach to the setting of $K > 2$ classes. This extension is known as multiclass logistic regression or multinomial logistic regression.

To do this, we first select a single class to serve as the **baseline** (why?); without loss of generality, we select the K-th class for this role. Then

$$\log \frac{\Pr(Y_i = k)}{\Pr(Y_i = K)} = \boldsymbol{\beta}_k^T \mathbf{x}_i,$$

for $k = 1, \ldots, K-1$. **Notice that the log odds between any pair of classes is linear in the features.**

We have introduced separate sets of regression coefficients, one for each possible outcome. If we exponentiate both sides, and solve for the probabilities, we get

$$\Pr(Y_i = k) = \Pr(Y_i = K) e^{\beta_k^T \mathbf{x}_i}.$$

Sum K probabilities

Using the fact that all K of the probabilities must sum to one, we find

$$\Pr(Y_i = K) = 1 - \sum_{k=1}^{K-1} \Pr(Y_i = k) = 1 - \sum_{k=1}^{K-1} \Pr(Y_i = K) e^{\beta_k^T \mathbf{x}_i}$$
$$\Rightarrow \Pr(Y_i = K) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k^T \mathbf{x}_i}}.$$

We can use this to find the other probabilities generally

$$\Pr(Y_i = k) = \frac{e^{\beta_k^T \mathbf{x}_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k^T \mathbf{x}_i}},$$

where β_K is defined to be zero.

Good night!

Slides created via Yihui Xie's R package **xaringan**.

Theme customized via Garrick Aden-Buie's R package **xaringanthemer**.

Tabbed panels created via Garrick Aden-Buie's R package **xaringanExtra**.

The chakra comes from remark.js, **knitr**, and R Markdown.