Probability & Statistics for DS & AI

Logistic Regression

Michele Guindani

Summer

Types of Regression Analyses

- An appropriate regression model for statistical analysis depends on the nature of the response variable:
- continuous/normally distributed responses: linear regression
- binary responses (disease/no disease) or binomial responses (number of successful outcomes out of a fixed number of patients): binomial regression
- counts: Poisson and negative binomial regression
- time-to-event data (e.g, time until your car breaks down): survival analysis
- Dataset: vector of outcomes (responses)

$$y = (y_1, \ldots, y_n)$$

Logistic Regression

For binary data, there may be a function $g(\cdot)$ such that

$$g(X;\boldsymbol{\beta}) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}$$

so that we can describe

$$P(Y = 1|\boldsymbol{X}) = \theta(x) = g(X;\boldsymbol{\beta})$$

Logistic Regression

For binary data, there may be a function $g(\cdot)$ such that

$$g(X;\boldsymbol{\beta}) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}$$

so that we can describe

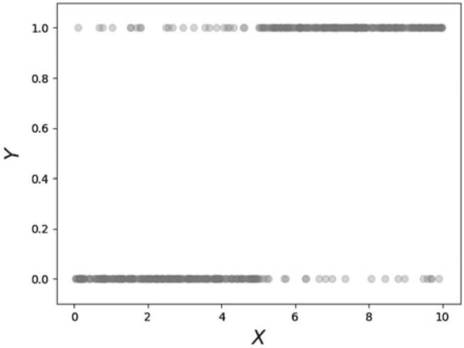
$$P(Y = 1|\boldsymbol{X}) = \theta(x) = g(X;\boldsymbol{\beta})$$

- In the logistic regression, we assume that the probability of success is represented by the logistic (or sigmoid) function:

$$g(X;\boldsymbol{\beta}) = \frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}}{1 + e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}}$$

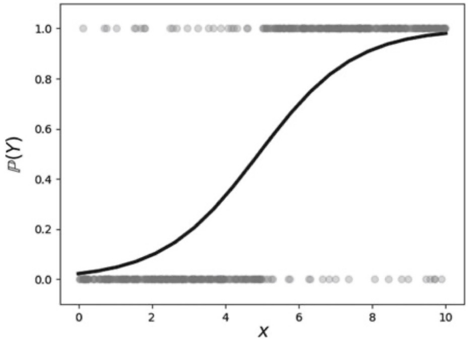
In general, appropriate choices for g depend on the nature of the response variable

This scatterplot shows the binary Y variables and the corresponding x data for each category



Larger values of X seem to lead to larger values of Y

This shows the fitted logistic regression on the data shown
The points along the curve are the probabilities that each point lies in either of the two categories



For large values of x the curve is near one, meaning that the probability that the associated Y value is equal to one. On the other extreme, small values of x mean that this probability is close to zero. Because there are only two possible categories, this means that the probability of $Y = 0$ is thereby higher. The region in the middle corresponding to the middle probabilities reflect the ambiguity between the two categories because of the overlap in the data for this region. Thus, logistic regression cannot make a strong case for one category here.

- So, logistic regression assumes that

$$P(Y = 1 \mid \boldsymbol{X}) = \theta(x) = \frac{e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}}}$$

👉 It follows that

$$P(Y = 0 \mid \boldsymbol{X}) = 1 - \theta(x) = 1 - \frac{e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}}} = \frac{1}{1 + e^{\beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}}}$$

❍ So, logistic regression assumes that

$$P(Y = 1 \mid \boldsymbol{X}) = \theta(x) = \frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}}{1+e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}}$$

👉 It follows that

$$P(Y = 0 \mid \boldsymbol{X}) = 1-\theta(x) = 1-\frac{e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}}{1+e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}} = \frac{1}{1+e^{\beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}}}$$

👉 Moreover,

$$\log(\text{Odds}) = \log\frac{P(Y=1|X)}{P(Y=0|X)} = \beta_0+\beta_1 x_{i1}+\ldots+\beta_k x_{ik}$$

which provides an immediate interpretation of the coefficients in terms of log-odds. The last equation is also called logistic transformation.

Use in Machine Learning

- In ML, logistic regression is used to predict the probability of a categorical dependent variable. The logistic regression model predicts $P(Y = 1)$ as a function of X.

- Binary logistic regression requires the dependent variable to be binary.

- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

- Other than that, everything proceeds as before (e.g. only the meaningful variables should be included, the model should have little or no multicollinearity etc).

- See examples in code (lab).