Logistic regression

Data science for Engineers

Introduction

- Logistic regression is a classification technique
- Decision boundary (generally linear)derived based on probability interpretation
  - Results in a nonlinear optimization problem for parameter estimation
- Goal: Given a new data point, predict the class from which the data point is likely to have originated

Logistic regression

Binary classification problem

- Classification is the task of identifying a category that a new observation belongs to based on the data with known categories
- When the number of categories is 2, it becomes a binary classification problem
- Binary classification is a simple "Yes" or "No" problem

Input features

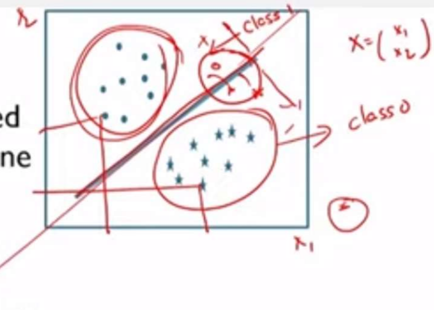
- Input features can be both qualitative and quantitative
- If the inputs are qualitative, then there has to be a systematic way of converting them to quantities
  - For example: A binary input like a "Yes" or "No" can be encoded as "1" and "0"
- Some data analytics approach can handle qualitative variables directly

$X = \begin{pmatrix} x_1 \\ x_2 \\ x_n \end{pmatrix}$

$\begin{pmatrix} \text{Yes} \\ 0.1 \\ 0.2 \end{pmatrix}$ $\begin{pmatrix} \text{No} \\ 0.05 \\ -2 \end{pmatrix}$

Linear classifier

- Decision function is linear
- Binary classification can be performed depending on the side of the half-plane that the data falls in
- We saw this before in the linear algebra module
- However, simply guessing "yes" or "no" is pretty crude
- Can we do something better using probabilities ?



Output

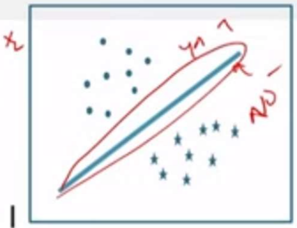
- Why model probabilities ?
  - The probability of a "Yes" or "No" gives a better understanding of the sample's membership to a particular category
  - Estimating the binary outputs from the probabilities is straight forward through simple thresholding
  - How does one model this probability ?

Linear and log models

- Make p(x) a linear function of x

$$p(x) = \beta_0 + \beta_1 X$$

$\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$



  - This makes p(x) unbounded below 0 and above 1
  - Might give nonsensical results making it difficult to interpret them as probabilities

0 and 1

- Make log(p(x)) a linear function of x

$$\log(p(x)) = \beta_0 + \beta_1 X$$

0 and 1

  - Bounded only on one side

Sigmoid function

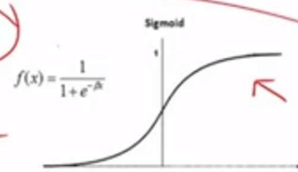
- Make p(x) a sigmoid function of x

$-\infty$ ∞

$$p(X) = \frac{e^{(\beta_0+\beta_1 X)}}{1 + e^{(\beta_0+\beta_1 X)}}$$

$\beta_0 + \beta_1 X$

$$\text{or } \log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$



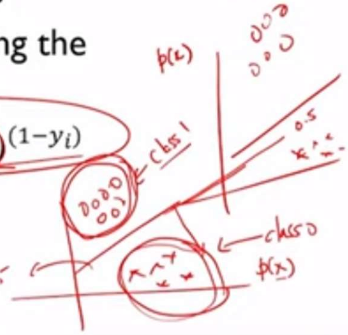
  - p(x) bounded above by 1 and below by 0
  - Good modeling choice for real life scenarios
  - The LHS can be interpreted as the log of odds-ratio in the second equation

Estimation of the parameters

- We find parameters in such a way that plugging these in the model equation should give the best possible classification for the inputs from both the classes
- This can be formalized by maximizing the following likelihood function
  - $L(\beta_0, \beta_1) = \prod_{i=1}^{n} (p(x_i))^{y_i} (1 - p(x_i))^{(1-y_i)}$

  when x_i belongs to class 0, $y_i = 0$

  when x_i belongs to class 1, $y_i = 1$



Log-likelihood function

- The log-likelihood function will become $l(\beta_0, \beta_1) = \sum_{i=1}^{n} y_i \log(p(x_i)) + (1 - y_i)\log(1 - p(x_i))$
- Simplifying this expression and using the definition for p(x) will result in an expression with the parameters of the linear decision boundary
- Now the parameters can be estimated by maximizing the above expression using any nonlinear optimization solver