IFI 9000 Analytics Methods
Linear and Logistic Regression

by **Houping Xiao**

January 12^{th}, 2021

Introduction

Brief Overview of Maximum Likelihood

- Maximum likelihood is a statistical estimation technique
- The main goal is to estimate the parameters of a statistical model given some sample observations
- Let $x_1, \cdots, x_n$ be samples from a distribution with some unknown parameter θ and joint distribution

$$f(x_1, \cdots, x_n|\theta)$$

- the maximum likelihood estimate of θ based on the observations

$$\hat{\theta} = \arg\max_{\theta} f(x_1, \cdots, x_n|\theta)$$

- When $x_1, \cdots, x_n$ are i.i.d samples from a distribution $f(\cdot)$, then

$$f(x_1, \cdots, x_n|\theta) = f(x_1|\theta) \cdots f(x_n|\theta)$$

Brief Overview of Maximum Likelihood

- **Example**: There is a normal distribution $\mathcal{N}(\mu, 1)$ with unknown μ. Given five samples from this distribution

$$x_1 = 2.5377, x_2 = 3.8339, x_3 = -0.2588, x_4 = 2.8622, x_5 = 2.3188;$$

what is the maximum likelihood estimate of μ?

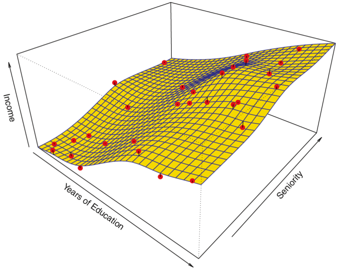
- **Solution**: If we take 5 independent samples x_1, x_2, x_3, x_4, and x_5 from a normal distribution $\mathcal{N}(\mu, 1)$, their joint distribution is

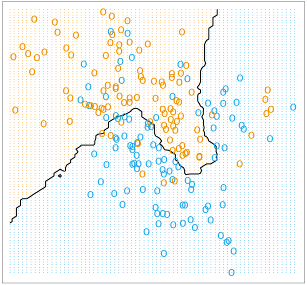
$$f(x_1, x_2, x_3, x_4, x_5|\mu) = \prod_{i=1}^{5} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu)^2}{2}\right),$$

some basic calculus yields $\hat{\mu} = \frac{x_1+x_2+x_3+x_4+x_5}{5} 5 = 2.2587$ (why?)

Classification

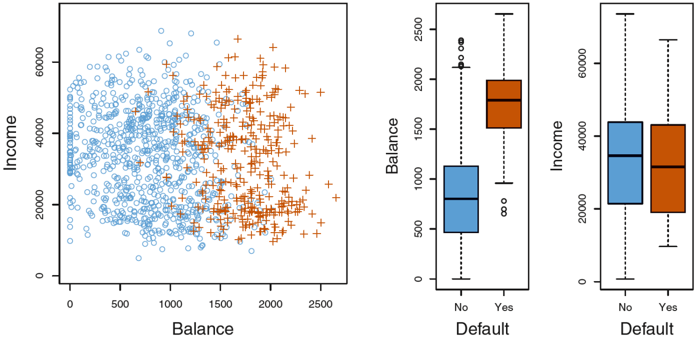
- In many applications, the response is not a quantitative value and instead represents a class, e.g., $y \in \{\text{spam}, \text{email}\}$, $y \in \{0, 1, \cdots, 9\}$
- Yet based on the observation of some features, we would like to predict the class, i.e., what we refer to as the classification
- Regression vs classification





Classification

- **Example**: Predicting default cases on the credit card (unable to pay the credit card), based on the income and current balance



Balance is more useful than the income. (why?)

Binary Classification

- In simple regression for a single feature x we fitted a line $y = \beta_0 + \beta_1 x$ to the data
- In a binary classification with only one feature, and the corresponding two classes: class 0 and class 1
- **Question**: Can we do the fit in a way that the sign of $\beta_0 + \beta_1 x$ becomes an indicator of the class for us?
- Mathematically, for a given feature x_i, making a decision as follows:

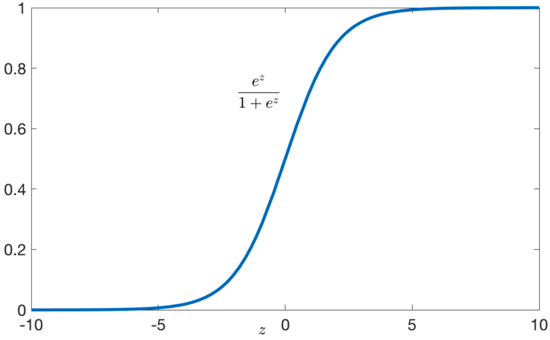
$$y_i = \begin{cases} 1, & \text{if } \beta_0 + \beta_1 x_i \geq 0; \\ 0, & \text{if } \beta_0 + \beta_1 x_i < 0. \end{cases}$$

- A smooth function (i.e., Sigmoid or inverse Logit) that takes almost binary values 0 and 1 based on the sign of the input z

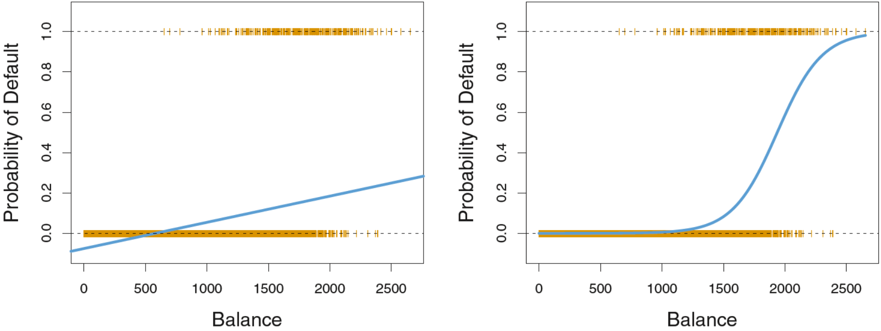
$$\frac{e^z}{1 + e^z} = \begin{cases} 1, & z >> 0; \\ 0, & z << 0. \end{cases}$$

Binary Classification

- When we have a smooth approximation of the sign function, learning the parameters β_0 and β_1 is numerically easier



Binary Classification



- Trying to treat the classification problem as a regression problem does not produce reasonable results!
- Some probability even becomes negative!

How Does Binary Classification Work?

- Learn β_0 and β_1 from the training data
- Given a test point x_i, evaluate $\beta_0 + \beta_1 x_i$
- Pass this quantity to the smooth sign approximation

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

- If $p(x_i)$ was closer to 1 our prediction of the class for x_i is class one (e.g., $p(x_i) = 0.9$) and if $p(x_i)$ was closer to 0 our prediction of the class for x_i is class zero (e.g., $p(x_i) = 0.1$)
- Now that $p(\cdot)$ generates some value between 0 and 1; one immediate interpretation for it is being the probability of label 1

$$p(x_i) = \mathbb{P}(y_i = 1|x_i) = 1 - \mathbb{P}(y_i = 0|x_i)$$

  - If $p(x_i) = 0.9$, then the test label is 1 with probability 0.9 and 0 with 0.1

How to Do the Training for the Simple Logistic Regression?

- Sample observations $(x_1, y_1), \cdots, (x_n, y_n)$ where $y_i \in \{0, 1\}$
- The goal is to determine β_0 and β_1 such that the probability of assigning the right labels is maximized

$$\arg\max_{\beta_0, \beta_1} \mathbb{P}(Y_1 = y_1, \cdots, Y_n = y_n | X_1 = x_1, \cdots, X_n = x_n, \beta_0, \beta_1)$$

We want to find the Maximum likelihood estimates for β_0 and β_1!

- Since our samples are independent, we have that

$$\begin{aligned}\mathbb{P}(Y_1 = y_1, \cdots, Y_n = y_n | X_1 = x_1, \cdots, X_n = x_n, \beta_0, \beta_1) \\ = \textstyle\prod_{i=1}^n \mathbb{P}(Y_i = y_i | X_i = x_i, \beta_0, \beta_1) \\ = \textstyle\prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)) \\ = \textstyle\prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}\end{aligned}$$

where

$$p(x_i) = \mathbb{P}(Y_i = 1 | x_i) = 1 - \mathbb{P}(Y_i = 0 | x_i)$$

- Find $\hat{\beta}_0$ and $\hat{\beta}_1$ that maximize

$$\prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i} = \prod_{i=1}^n \left(\frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} \right)^{y_i} \left(\frac{1}{1 + e^{\beta_0 + \beta_1 x}} \right)^{1-y_i}$$

Some Notes on The Logistic Regression

- In logistic regression, we end up with a more complex cost function to optimize

$$\prod_{i=1}^{n} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} = \prod_{i=1}^{n} \left(\frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}\right)^{y_i} \left(\frac{1}{1+e^{\beta_0+\beta_1 x}}\right)^{1-y_i}$$

- Generally speaking, a closed-form solution for the maximizer is not available and often maximization techniques such as gradient ascent (or gradient descent on the negative log-likelihood) are considered
  We will discuss gradient descent soon!

What Happens for More than One Feature?

- In case of multiple features, only minor modification is required
- We still try to maximize $\prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$, but now we have that

$$p(x_i) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

- We run the maximization to obtain the estimates $\hat{\beta}_0, \cdots, \hat{\beta}_p$
- In practice, you never have to do the maximization and most software such as R, Python, and Matlab have packages to dot hat numerically for you

What Happens for More than Two Classes?

- **Example**: Based on some features such as city, year of education, and number of publications, classify the students of a class into undergrads, Master, and PhDs
- Recall our method of classification in the binary case, we evaluated $p(x_i)$ which was technically $\mathbb{P}(Y_i = 1|x_i)$ and if it closer to 1 then our prediction is class one; if it is small, then $\mathbb{P}(Y_i = 0|x_i) = 1 - \mathbb{P}(Y_i = 1|x_i)$ would be large and our prediction is class zero
- One way of interpreting this is evaluating $\mathbb{P}(Y_i = k|x_i)$ for $k = 0$ and 1 and the k that produces the largest value for $\mathbb{P}(Y_i = k|x_i)$ is our predicted label
- Now for K labels, we evaluate $\mathbb{P}(Y_i = k|x_i)$ for $k = 1, 2 \cdots, K$ and the k that produces the largest value for $\mathbb{P}(Y_i = k|x_i)$ is our predicted label

What Happens for More than Two Classes?

- For K labels, we evaluate $\mathbb{P}(Y_i = k|x_i)$ for $k = 1, 2, \cdots, K$ and the k that produces the largest value for $\mathbb{P}(Y_i = k|x_i)$ is our predicted label
- When we have $K > 2$ labels (e.g., $y \in \{0, 1, \cdots, 9\}$) and p features $x_1, \cdots, x_p$, we fit K models parametrized by

$$\begin{gathered}
\text{Label } 1 : \{\beta_0^{(1)}, \beta_1^{(1)}, \cdots, \beta_p^{(1)}\} \\
\text{Label } 2 : \{\beta_0^{(2)}, \beta_1^{(2)}, \cdots, \beta_p^{(2)}\} \\
\vdots \\
\text{Label } K : \{\beta_0^{(K)}, \beta_1^{(K)}, \cdots, \beta_p^{(K)}\}
\end{gathered} \tag{1}$$

- For this problem, we consider the following form,

$$p_k(\boldsymbol{x}) = \mathbb{P}(Y = k|\boldsymbol{x}) = \frac{e^{\beta_0^{(1)} + \cdots + \beta_p^{(1)} x_p}}{e^{\left(\beta_0^{(1)} + \cdots + \beta_p^{(1)} x_p\right)} + \cdots + e^{\left(\beta_0^{(K)} + \cdots + \beta_p^{(K)} x_p\right)}}$$

- What is the sum of all $\mathbb{P}(Y_i = k|x_i)$ for a fixed $\boldsymbol{x}$?

Some Practices

Let's perform some basic classification tasks in Python!

The End