Logistic Regression

YuHua Li(李玉华)

Intelligent and Distributed Computing Lab,
Huazhong University of Science & Technology

idcliyuhua@hust.edu.cn

2019年04月14日

Introduction

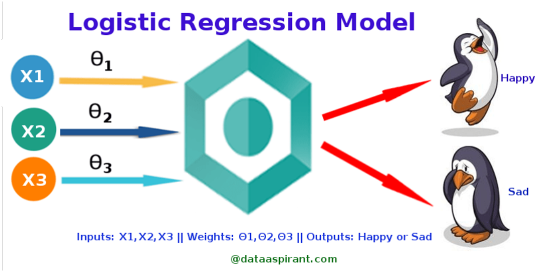


Figure: Logistic regression is a classic machine learning algorithom used for calssification task. As shown in the picture, we first feed the data, the inputs to the model, and the model gives us its classification result.

Introduction

Basic idea:

- $f(Z) = h_\theta(x) = sigmoid(Z) = \frac{1}{1+e^{-Z}}$
- Output $=$ 0 or 1
- $Hypothesis \longrightarrow Z = W_x + B$

We usually attach the sigmod function to the end of the model, it gives us the probaility of the label being 1

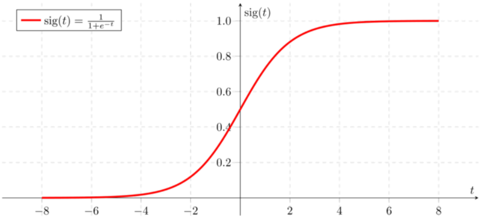


Figure: If ‘Z’ goes to infinity, Y(predicted) will become 1 and if ‘Z’ goes to negative infinity, Y(predicted) will become 0.

Loss Function

Predicted Probability:

$$P(y=1|x) = h\theta(x) = \frac{1}{1+e^(-\theta^T x)} = \delta(\theta^T x) \tag{3}$$

$$P(y=0|x) = 1 - P(y=1|x) = 1 - h_\theta(x) \tag{4}$$

Combined (3).(4):

$$P(y|x;0) = (h_\theta(x))^y (1-h_\theta(x))^{1-y^{y \text{ stands for the labels, being 0 or 1}}} \tag{5}$$

Using maximum likelihood estimation(MLE) according to the m given samples:

$$L(\theta) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) = \prod_{i=1}^{m} (h_\theta(x^{(i)}))y^{(i)}(1-h_\theta(x^{(i)}))^{1-y^{(i)}} \tag{6}$$

$$\longrightarrow l(\theta) = \log L(\theta) = \sum_{i=1}^{m} (y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(y^{(i)}))) \tag{7}$$

$$J(\theta) = -\frac{1}{m} l(\theta)$$

$J(\theta)$ is the desired loss function

Basic idea

- L2

$$J(\theta) = \frac{1}{2m}[\sum_{i=1}^{m}(h_\theta(x^{(i)}))^2 + \lambda \sum_{j=1}^{n} \theta_j^2] \tag{8}$$

- Regularization : prevent the weights from getting too large
- Regularization can avoid overfitting to some extent through the restraint of weights

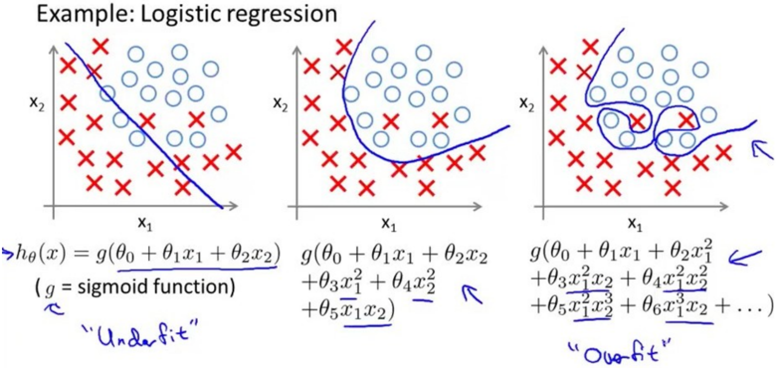


Figure: Underfitting/overfitting

Classification task through Logistic Regression

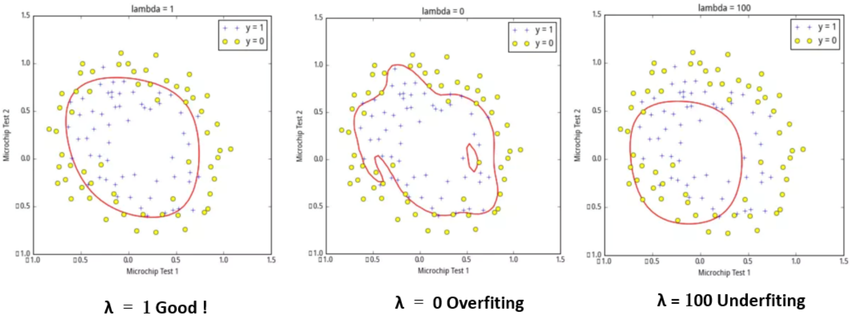
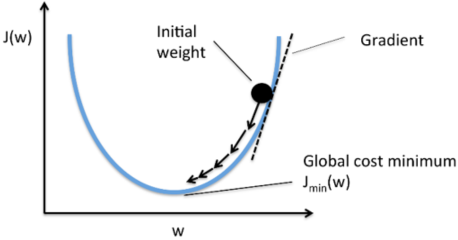
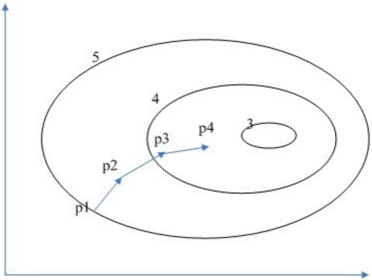


Figure: Two-class classification when λ has different values

Gradient Descent





Gradient Descent

$$J(\theta) = -\sum_{i=1}^{m}(y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(y^{(i)}))) \tag{9}$$

$$\downarrow$$

Partial derivative:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_i x_j^{(i)}(h_\theta(x^{(i)}) - y^{(i)})$$

$$\downarrow$$

Weights update:

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial J(\theta_j)}$$

Table of Contents

Comparison

Basic idea

- Machine learning algorithms can be (roughly) categorized into two categories:
- Generative algorithms, that estimate $P(x_i, y)$ (often they model $P(x_i|y)$ and $P(y)$ separately).
- Discriminative algorithms, that model $P(y|x_i)$

1. The Naive Bayes algorithm is generative. $(p(y), p(x|y) \rightarrow p(y|x))$

$$P(y|x) = \frac{p(x, y)}{\prod_y p(x, y)} = \frac{p(y)p(x|y)}{\prod_y p(y)p(x|y)}$$

2. Logistic Regression is discriminative. (Gradient descent$\rightarrow$ w)

$$P(y|x_i) = \frac{1}{1 + e^{y(w^T x_i + b)}}$$

The End