Types of Learning

Instructors: Dr. Durgesh Singh

CSE Discipline, PDPM IIITDM, Jabalpur -482005

Logistic Regression

- Logistic regression is a classifier.
- Classification is supervised learning problem.
  - ✓ Tumor: malignant or benign cancer
  - ✓ Email: Spam or Not spam
- In logistic regression the prediction values always between 0 and 1.

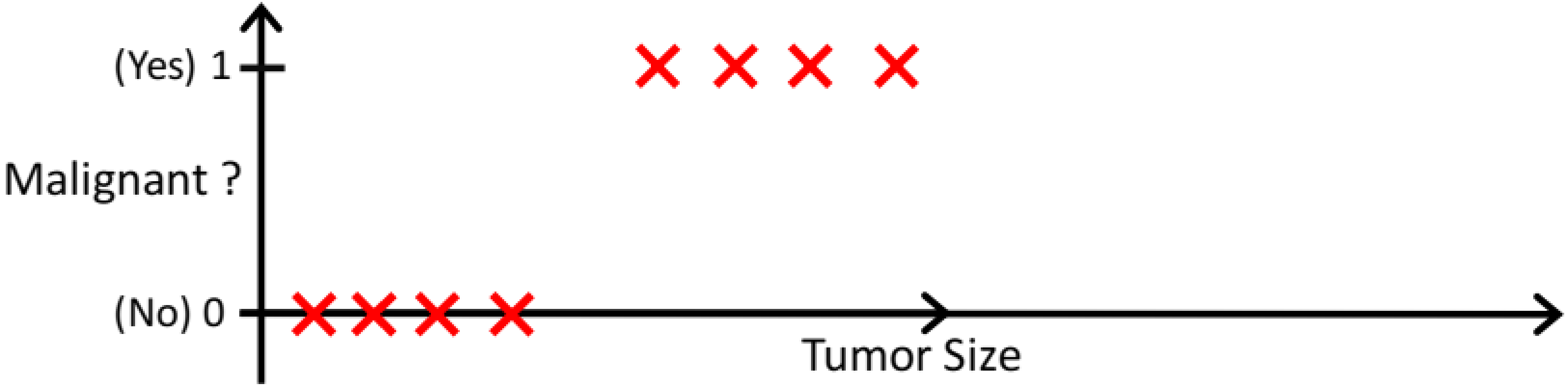
$$0 \leq h_\theta(x) \leq 1$$

Classification Problem

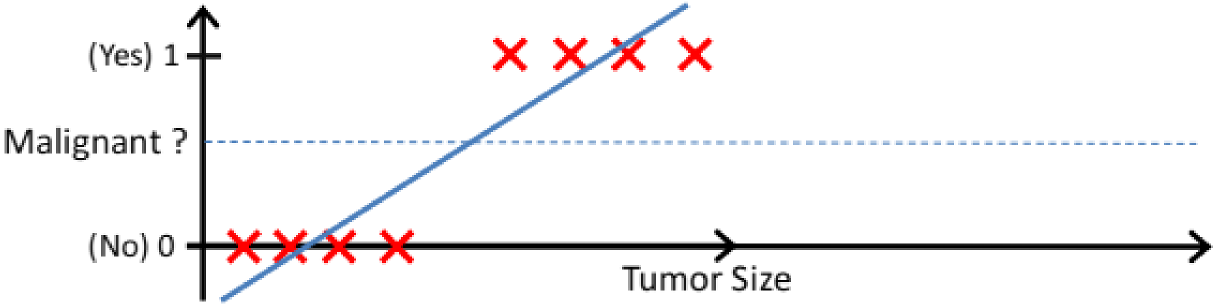
- Classification problem just like the regression problem, except that the values y. We now want to predict take only a small number of discrete values.
- For simplicity, we will focus here on the binary classification problem in which y can take only two values, 0 and 1.
- For instance, if we are trying to build a classifier for tumor, then y may be 1 if it is a malignant, and 0 otherwise.
- 0 is also called the **negative class**, and 1 the **positive class.**
- **Can we solve this using linear regression ?**

Classification Problem

- We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given x.
- However, it is easy to construct examples where this method performs very poorly.



Can we solve the problem using linear regression?

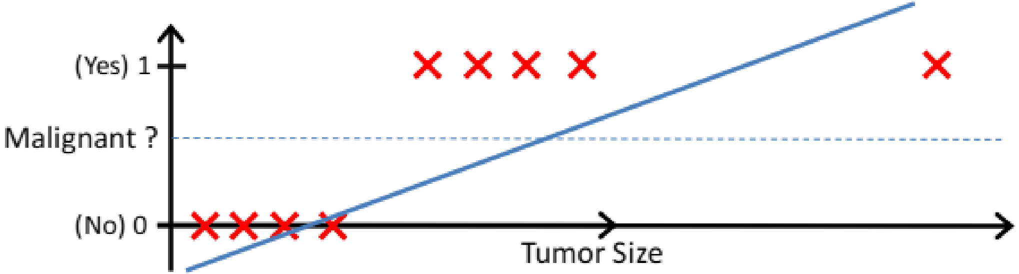


Can we solve the problem using linear regression? E.g., fit a straight line and define a threshold at 0.5

Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"



Can we solve the problem using linear regression? E.g., fit a straight line and define a threshold at 0.5

Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Failure due to adding a new point

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

Another drawback of using linear regression for this problem

What we need:

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Logistic Regression

- To fix this, let's change the form for our linear regression hypotheses $h_\Theta(x)$ as

$$h_\theta(x) = g(\theta^T x)$$
$$= \frac{1}{1 + e^{-\theta^T x}}$$

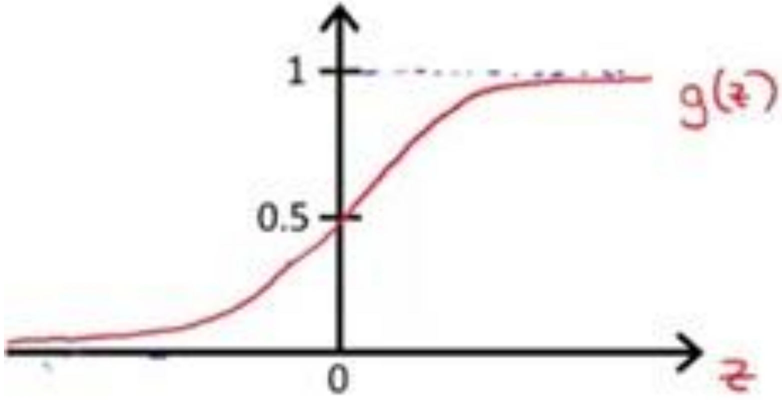
where

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

is called the logistic function or the sigmoid function.

Sigmoid Function

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



Interpretation of Hypothesis Output

- In logistic regression, we do not just predict +ve or –ve.
- We predict probability, how likely this is to be +ve or –ve.

$$h_\Theta(x) = P(y=1|x; \theta)$$

- $h_\Theta(x)$ = estimated probability that y = 1 on input x Example: If classifier for tumor gives $h_\Theta(x)$ =0.7 means 70% chance of tumor being malignant.

Cost function for Logistic Regression

- Training Set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $x \in \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix}$ $x_0 = 1, y \in \{0, 1\}$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

How to choose parameters θ ?

Cost function

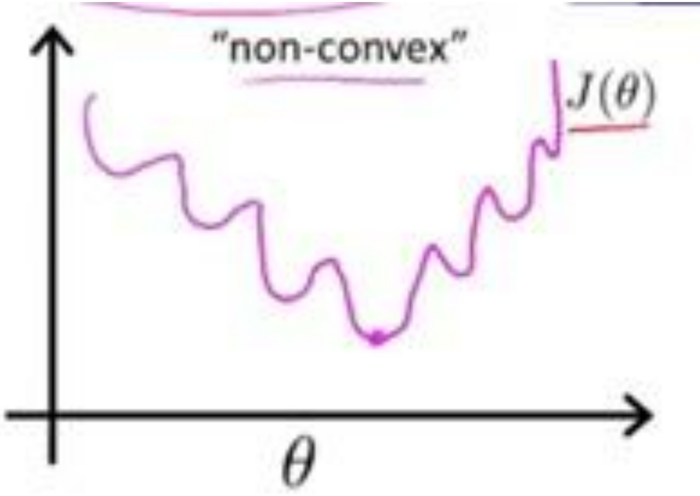
Linear Regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Squared error cost function:

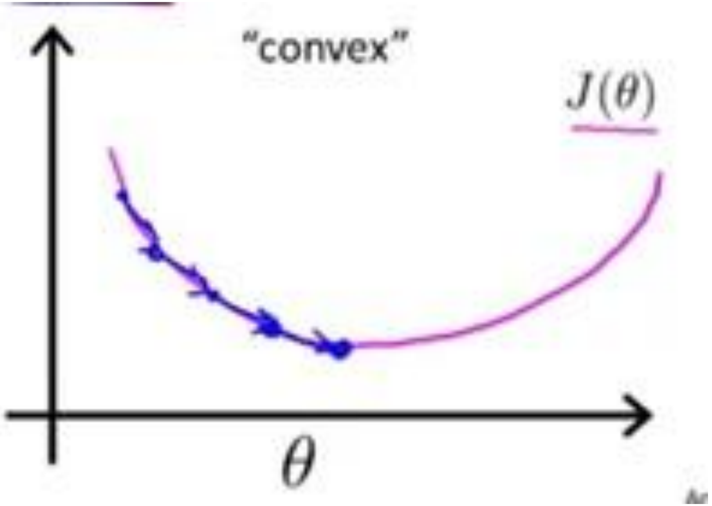
$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

However, this cost function is non-convex for the hypothesis of logistic regression.

Cost function



Logistic regression

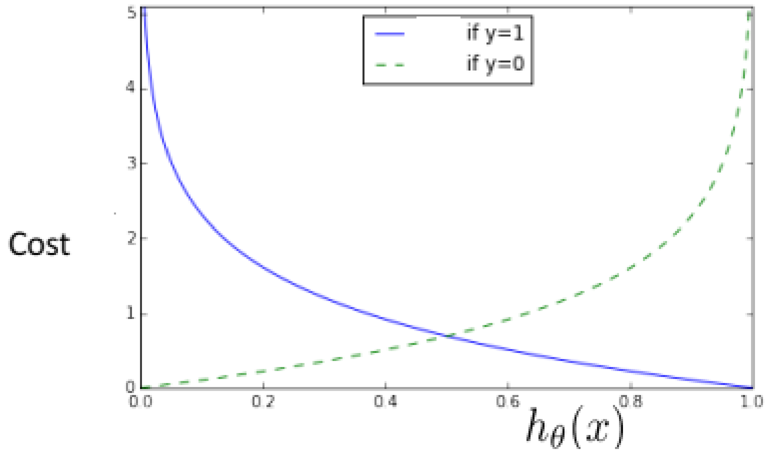


Linear regression

Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



Logistic regression cost function

This cost function is convex

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

To fit parameters θ :

$$\min_\theta J(\theta)$$

Gradient Descent:

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

Want $\min_\theta J(\theta)$

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

$$\frac{\partial}{\partial z} g(z) = \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right) = \frac{-1}{(1+e^{-z})^2}\frac{\partial}{\partial z} e^{-z}$$

$$= \frac{+1}{(1+e^{-z})}\left(\frac{e^{-z}}{1+e^{-z}}\right) = \left(\frac{1}{1+e^{-z}}\right)\left(1 - \frac{1}{1+e^{-z}}\right)$$

$$\boxed{\frac{\partial}{\partial z} g(z) = g(z)\left(1 - g(z)\right)}$$

$$\frac{\partial}{\partial \theta_j} j(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{\partial}{\partial \theta_j}\log(h_\theta x^{(i)})\right.$$

$$\left. + (1-y^{(i)})\frac{\partial}{\partial \theta_j}\log(1-h_\theta x^{(i)})\right)$$

Apply $\frac{\partial}{\partial x}\log x = \frac{1}{x}$ and $\frac{\partial}{\partial z} g(z) = g(z)(1-g(z))$

$$\boxed{\frac{\partial}{\partial \theta_j} j(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta x^{(i)} - y^{(i)}\right)x_j^{(i)}}$$

Gradient Descent

Repeat $\{$

$$\theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression, but the hypothesis function is different for logistic regression.


Thank You
Gracias
Danke
Merci
Grazie