Logistic Regression for Classification

KAUSHIK ROY

SRC JUMP

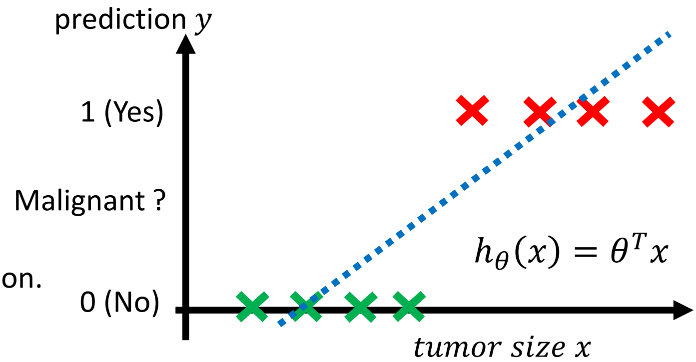
C-BRIC
Center for Brain Inspired Computing

Logistic Regression

➢ Motivation

E.g. Predict tumor type (Malignant or Benign) based on tumor size.

- Feature:
  $x = tumor\ size\ (mm)$
- Prediction:
  $y = 1$, tumor is malignant
  $y = 0$, tumor is benign



A straight line $h_\theta(x)$ is used to fit the data using linear regression.

Recall linear regression for classification.

- Hypothesis:
  $h_\theta(x) = \theta^T x$, in which,
  $\theta = \begin{bmatrix}\theta_0\\ \theta_1\end{bmatrix}$, and $x = \begin{bmatrix}x_0\\ x_1\end{bmatrix}$, $(x_0 = 1)$
- Prediction:
  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 $y = 0$

Prediction result of linear regression hypothesis is $-\infty < h_\theta(x) < \infty$, but most of the time, the desired output y is in the range $-1 \leq y \leq 1$

Logistic Regression

➢ Hypothesis and representation

We want the classifier to output values in range $0 \leq h_\theta(x) \leq 1$

A new hypothesis satisfying this requirement:

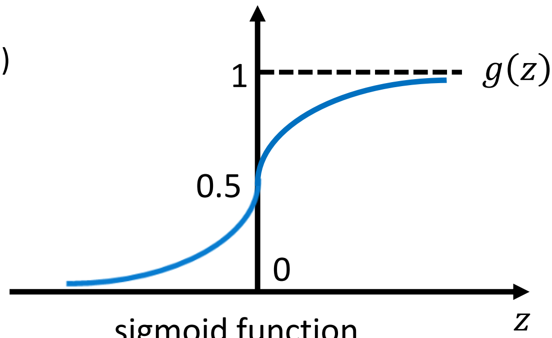
- Logistic function (also called sigmoid function)

$$h_\theta(x) = g(\theta^T x),$$

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

or in a more compact format:

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



sigmoid function

- The prediction result of logistic regression is between -1 and 1.

Similar to linear regression, after defining the logistic regression hypothesis, we need a learning algorithm to find the proper parameter θ, so that the model can predict desirable outputs. How to compute parameter θ?

Logistic Regression

➢ Model interpretation

E.g. Predict tumor type (Malignant or Benign) based on tumor size.

Logistic regression model:

Features: $x = tumor\ size\ (mm)$

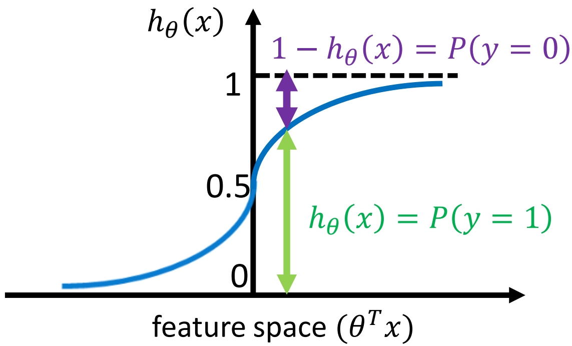
Prediction:

$y = 1$, tumor is malignant

$y = 0$, tumor is benign

Logistic regression hypothesis:

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



Interpretation

- Predicted output $h_\theta(x)$ equals the estimated probability that $y = 1$ for the given input x and parameter θ:

  $P(y = 1|x; \theta) = h_\theta(x)$
- The probability that $y = 0$ on the same x and θ:

  $P(y = 0|x; \theta) = 1 - h_\theta(x)$
- Summation of probability that $y = 1$ and $y = 0$ equals 1.

  $P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$

Decision Boundary

- Linear decision boundary

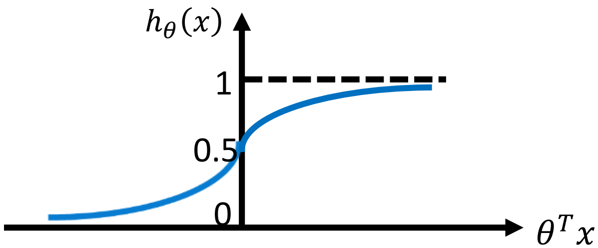
Recall the logistic regression model

- Model expression:

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

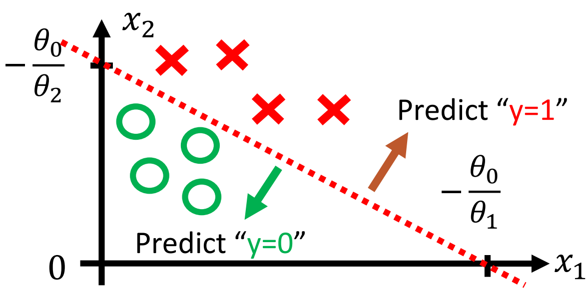
- Model prediction:

Predict "$y = 1$" if $\theta^T x \geq 0$
Predict "$y = 0$" if $\theta^T x < 0$



Note: To better illustrate decision boundary, in the following example, the parameter θ is assumed to be known. How to compute parameter θ will be introduced later.

Suppose $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$,
in which $(\theta_0, \theta_1, \theta_2 \in \mathbb{R},\ \theta_1 \neq 0 \ and \ \theta_2 \neq 0)$
Assume parameters $\theta_0, \theta_1, \theta_2$ are already known.



Decision boundary in the feature space

- Decision boundary

Predict "y=1" if $\theta_0 + \theta_1 x_1 + \theta_2 x_2 \geq 0$

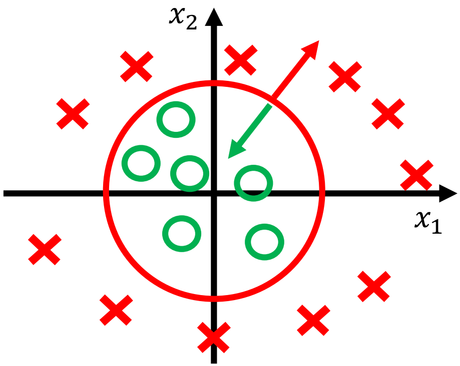
$$\rightarrow x_1 + \frac{\theta_2}{\theta_1} x_2 \geq -\frac{\theta_0}{\theta_1}$$

Predict "y=0" if $\theta_0 + \theta_1 x_1 + \theta_2 x_2 < 0$

$$\rightarrow x_1 + \frac{\theta_2}{\theta_1} x_2 < -\frac{\theta_0}{\theta_1}$$

Decision Boundary

➢ Non-linear decision boundary

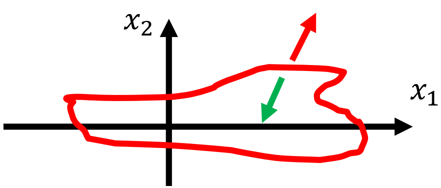


- Model expression

  $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$, $(\theta_0, \cdots, \theta_4 \in \mathbb{R})$

  Suppose parameter $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

  The original hypothesis $h_\theta(x)$ becomes:
  $h_\theta(x) = g(-1 + \theta_3 x_1^2 + \theta_4 x_2^2)$



An example of more complex non-linear decision boundary

- Model prediction

  Predict "y=1" if $x_1^2 + x_2^2 \geq 1$
  Predict "y=0" if $x_1^2 + x_2^2 < 1$

- Decision boundary

  $x_1^2 + x_2^2 = 1$

More complex decision boundary is possible by using higher order polynomial, i.e. $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1^3 + \theta_6 x_2^3 + \cdots)$

Cost Function and Gradient Descent

➢ Linear regression cost function

Recall the cost function of linear regression:

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

For simplicity we use the following notation:

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

Original cost function is simplified to:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)$$

in which $h_\theta\left(x^{(i)}\right)$ indicates predicted output of the ith example in dataset, and $y^{(i)}$ indicates desired output of the ith example in dataset.

Cost function computes the summation of "cost" of all examples divided by the number of examples in the dataset.

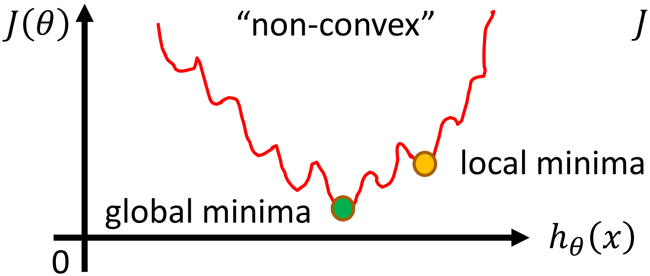
Cost Function and Gradient Descent

➢ Linear regression cost function

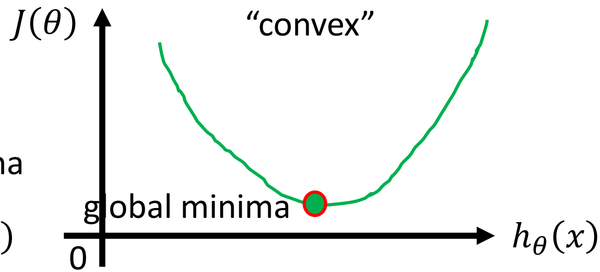
Recall the cost function of linear regression:

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

The cost function of linear regression may not be used for logistic regression because it becomes "non-convex" if the hypothesis $h_\theta(x)$ is non-linear, e.g. sigmoid function.



Cost function becomes "non-convex" if the hypothesis $h_\theta(x)$ is non-linear, e.g. sigmoid function $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$



Cost function is "convex" if the hypothesis $h_\theta(x)$ is linear function, e.g. $h_\theta(x) = \theta^T x$

Gradient descent is not guaranteed to converge at global minima for "non-convex" cost function

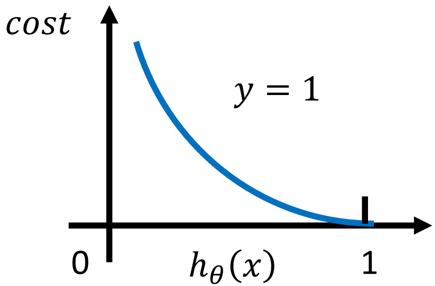
Cost Function and Gradient Descent

➢ Logistic regression cost function

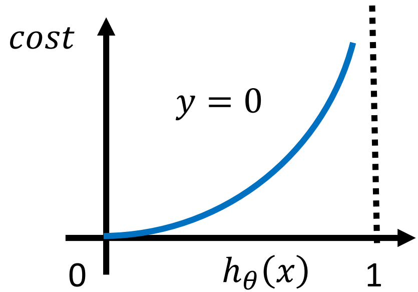
Define a new cost function for logistic regression:

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

and plot the cost function against $h_\theta(x)$ as following:



Plot $cost$ when $y = 1$
$\mathbf{if}\ \boldsymbol{h_\theta(x) = 1, cost = 0}$
$\mathbf{if}\ \boldsymbol{h_\theta(x) = 0, cost \to \infty}$



Plot $cost$ when $y = 0$
$\mathbf{if}\ \boldsymbol{h_\theta(x) = 0, cost = 0}$
$\mathbf{if}\ \boldsymbol{h_\theta(x) = 1, cost \to \infty}$

Cost Function and Gradient Descent

➢ Logistic regression cost function

Logistic regression cost function:

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

Binary Cross-Entropy Loss

Above expression can be written in a more compact but mathematically equivalent way:

$$cost(h_\theta(x), y) = -log(h_\theta(x)) * y - log(1 - h_\theta(x)) * (1 - y),$$

in which $(y = 0\ or\ 1)$

Note that we use above function to compute the "cost" of one example, there are totally m examples in dataset. As a result, a superscript (i) is used to indicate example index in dataset.

$$cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = -log\left(h_\theta\left(x^{(i)}\right)\right) * y^{(i)} - log\left(1 - h_\theta\left(x^{(i)}\right)\right) * \left(1 - y^{(i)}\right),$$

in which $(i = 1,2, \dots, m\ )$

The cost function of the entire dataset equals the average "cost" of all m samples in it.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) \qquad (i = 1,2, \dots, m\ )$$

Cost Function and Gradient Descent

- Gradient descent for logistic regression

How to compute the parameter θ for logistic regression model?

Recall logistic regression cost function:

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

$$(i = 1,2, \dots, m\ )$$

Gradient descent algorithm for logistic regression:

Repeat {

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta_0, \theta_1,\ \theta_2, \cdots \theta_n) \quad (\alpha \text{ is learning rate, } n \text{ is number of features})$$

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

} (simultaneously update for every $j = 0, 1, \dots, n$)

Note that the update rules of gradient descent for linear regression and logistic regression are the same, but the hypothesis function $h_\theta(x)$ and cost function $J(\theta)$, which will be plugged into the gradient descent formula are different.