Logistic Regression
AI4003-Applied Machine Learning

Dr. Mohsin Kamal

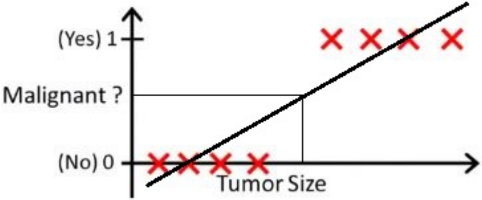
Department of Electrical Engineering
National University of Computer and Emerging Sciences, Lahore, Pakistan

- Email: Spam / Not Spam?
- Online Transactions: Fraudulent (Yes / No)?
- Tumor: Malignant / Benign?

In all these examples,

$$y \in \{0, 1\}$$

0: "Negative Class" (e.g., benign tumor)
1: "Positive Class" (e.g., malignant tumor)

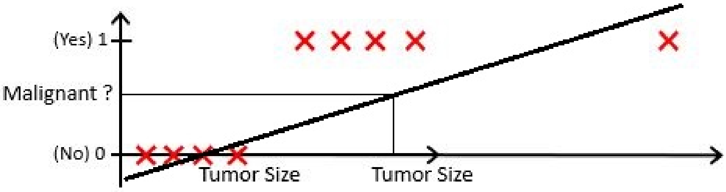


If we apply linear regression algorithm i.e.,

$$h_\theta = \theta^T x$$

then,
Threshold classifier output h_θ at 0.5:

$$\text{If } h_\theta \geq 0.5\text{, predict "}y = 1\text{"}$$
$$\text{If } h_\theta < 0.5\text{, predict "}y = 0\text{"}$$



Will linear regression algorithm apply??

Problems:

- Extreme cases can not be handled
- Bad error function

Classification predicts: $y = 0$ *or* 1
But,
h_θ can be > 1 *or* < 0 when applying linear regression.

Solution:
Logistic Regression gives: $0 \leq h_\theta \leq 1$

Logistic regression model

We want $0 \leq h_\theta \leq 1$
Previously, from linear regression model, we know that

$$h_\theta = \theta^T x \tag{1}$$

modifying equation 1 by introducing **Sigmoid function** or **Logistic function**

$$h_\theta = g(\theta^T x) \tag{2}$$

where,

$$g(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

Equation 2 becomes

$$h_\theta = \frac{1}{1 + e^{-\theta^T x}} \tag{4}$$

Interpretation of Hypothesis Output

$h_\theta(x)$ = estimated probability that $y = 1$ on input x

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ tumorSize \end{bmatrix}$

and we get $h_\theta(x) = 0.7$

Tell patient that 70% chance of tumor being malignant

Hypothesis equation can be represented as:

$$h_\theta(x) = P(y = 1|x;\theta) \tag{5}$$

Equation 5 translates as "probability that $y = 1$, given x, parameterized by θ"

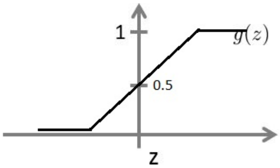
$$P(y = 0|x;\theta) + P(y = 1|x;\theta) = 1 \tag{6}$$

$$P(y = 0|x;\theta) = 1 - P(y = 1|x;\theta) \tag{7}$$

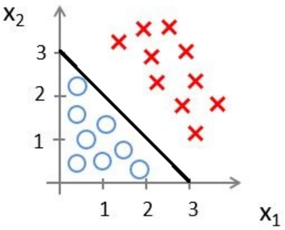
Logistic regression

Referring to equations 2 and 3.

Predict "$y = 1$" if $h_\theta(x) \geq 0.5$	Predict "$y = 0$" if $h_\theta(x) < 0.5$
$g(z) \geq 0.5$ when $z \geq 0$ $h_\theta(x) = g(\theta^T x) \geq 0.5$ whenever $\theta^T x \geq 0$	$g(z) < 0.5$ when $z < 0$ $h_\theta(x) = g(\theta^T x) < 0.5$ whenever $\theta^T x < 0$



Decision boundary



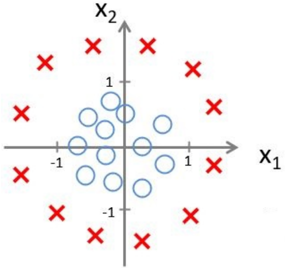
If we have $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
having $\theta_0 = -3,\ \theta_1 = 1\ and\ \theta_2 = 1$ then,

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

$$\implies \quad x_1 + x_2 = 3$$

Also, $x_1 + x_2 < 3$

Non-linear decision boundaries



If we have $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$
having $\theta_0 = -1,\ \theta_1 = 0,\ \theta_2 = 0,\ \theta_3 = 1\ and\ \theta_4 = 1$ then,

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

$$\implies \quad x_1^2 + x_2^2 \geq 1$$

So, $x_1^2 + x_2^2 = 1$ represents the equation on circle with radius 1.

Another equation could be:

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \cdots)$$

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

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

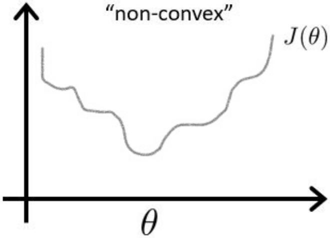
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

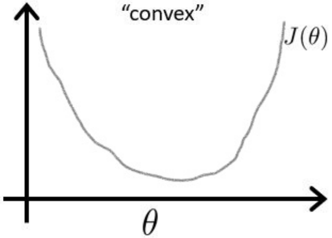
How to choose parameters θ ?

Cost function

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

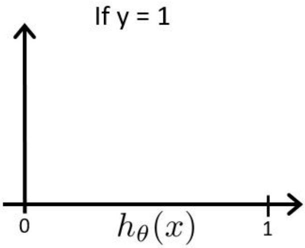
$$\mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$





Logistic regression cost function

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



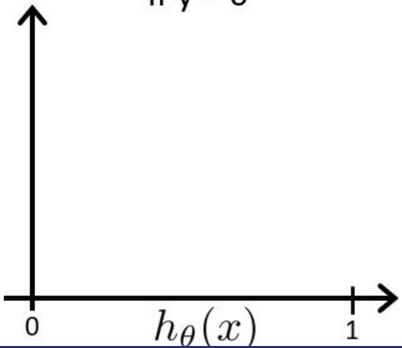
Cost $= 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x;\theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

Logistic regression cost function

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

If y = 0



Logistic regression cost function

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

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

Note: $y = 0$ or 1 always

Logistic regression cost function

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

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

To fit parameters θ:

$$\min_{\theta} J(\theta)$$

To make a prediction given new x:

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

Gradient Descent

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

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

Repeat $\{$

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

$\}$ (simultaneously update all θ_j)

Gradient Descent

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

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

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression!

Advanced optimization

Optimization algorithm

Given θ, we have code that can compute

- $J(\theta)$
- $\frac{\partial}{\partial \theta_j} J(\theta)$ (for $j = 0, 1, \ldots, n$)

Optimization algorithms:

- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

Advantages:

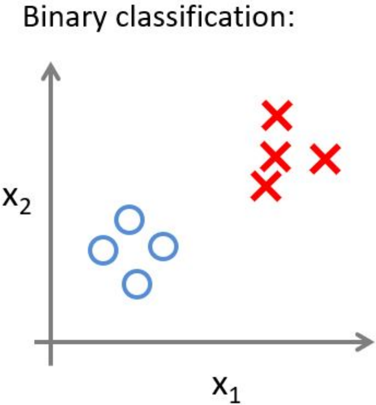
- No need to manually pick α
- Often faster than gradient descent.

Disadvantages:

- More complex

Multiclass classification:

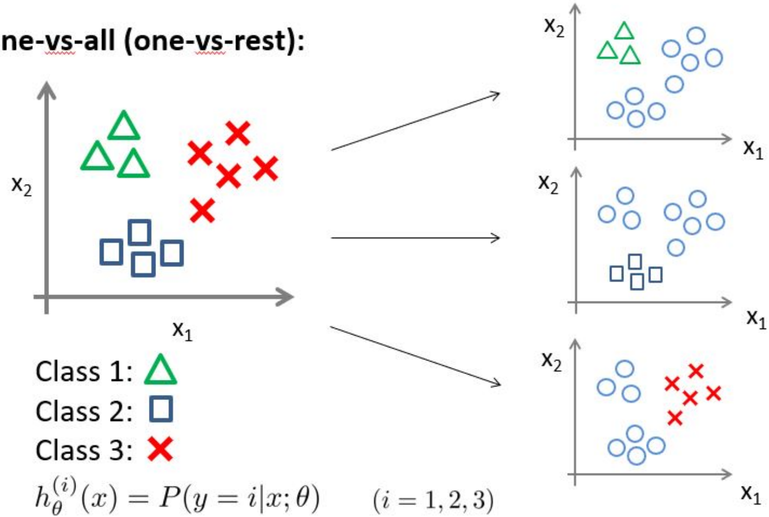
- A classification task with more than two classes.
- Each sample can only be labeled as one class.
- Example 1: Tumor - Benign, stage1, stage2, stage3, stage4.
- Example 2: Weather - Sunny, Cloudy, Rain, Snow.


Binary classification:
x_2
x_1
Multi-class classification:
x_2
x_1

- Some algorithms (such as Random Forest classifiers or naive Bayes classifiers) are capable of handling multiple classes directly.
- Others (such as Support Vector Machine classifiers or Linear classifiers) are strictly binary classifiers.
- However, there are various strategies that you can use to perform multiclass classification using multiple binary classifiers.

- One way to create a system that can classify the digit images into 10 classes (from 0 to 9) is to train 10 binary classifiers, one for each digit (a 0-detector, a 1-detector, a 2-detector, and so on). Then when you want to classify an image, you get the decision score from each classifier for that image and you select the class whose classifier outputs the highest score. This is called the **one-versus-all (OvA)** strategy (also called **one-versus-the-rest**).
- Another strategy is to train a binary classifier for every pair of digits: one to distinguish 0s and 1s, another to distinguish 0s and 2s, another for 1s and 2s, and so on. This is called the **one-versus-one (OvO)** strategy. If there are N classes, you need to train $N \times (N-1)/2$ classifiers.
- Some algorithms (such as Support Vector Machine classifiers) scale poorly with the size of the training set, so for these algorithms OvO is preferred since it is faster to train many classifiers on small training sets than training few classifiers on large training sets. For most binary classification algorithms, however, OvA is preferred.

One-vs-all (one-vs-rest):



Class 1: △
Class 2: □
Class 3: ✕

$$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$$

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

On a new input x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)}(x)$$

Another method:

- Softmax

Multilabel Classification

Until now each instance has always been assigned to just one class. In some cases you may want your classifier to output multiple classes for each instance. For example, consider a face-recognition classifier: what should it do if it recognizes several people on the same picture? Of course it should attach one tag per person it recognizes. Say the classifier has been trained to recognize three faces, Alice, Bob, and Charlie; then when it is shown a picture of Alice and Charlie, it should output [1, 0, 1] (meaning "Alice yes, Bob no, Charlie yes"). Such a classification system that outputs multiple binary tags is called a multilabel classification system.