Classification

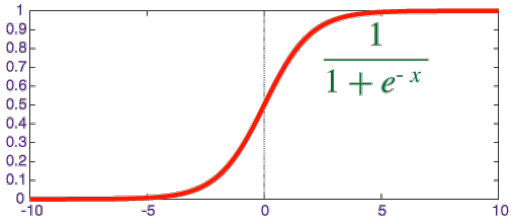
In classification problem, we want to predict based on discrete values. Let's start with simple version of problem, binary classification in which y can take value of 0 or 1 (later we can generalize it to more classes). For example, in spam classifier problem y=1 (positive classes) when it's spam and y=0 for noon-spam emails (it's called negative class).

Hypothesis Representation

Logistic regression is one of the algorithm used for classification:

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

The logistic function is called sigmoid function. It maps g(z) to values between (0,1) interval shown in image on the right. $h_\theta(x)$ gives probability. *$h_\theta(x)$ gives probability that our output is 1.* For example, $h_\theta(x) = 75$ means with probability of 75%, the output is 1. We can calculate the prediction of 0 by 1-$h_\theta(x)$.

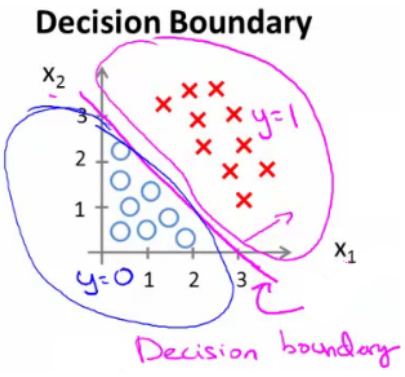


Decision Boundary

In order to get the discrete value of 0 or 1 for classification, we translate the hypothesis as follow:

$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$
$$h_\theta(x) < 0.5 \rightarrow y = 0$$

The decision boundary is the line that separates the area where y = 0 and where y = 1. It is created by our hypothesis function. $g(z) \geq 0.5$ when z≥ 0, when z=0, $e^{-z} = 1$ therefore g(z)=1/2. Decision boundary is a property of hypothesis, not the properties of training set.



For example, in the plot here, the classifier is drawn when two class are separated with probability of 0.5. Another example is:

$\theta = \begin{bmatrix} 5 \\ -1 \\ 0 \end{bmatrix}$, y=1 if

$5 + (-1)x_1 + 0x_2 \geq 0, \qquad x_1 \leq 5$

The decision line is $x_1 = 5$, and anything on the left of line is y=1 and else its negative class.

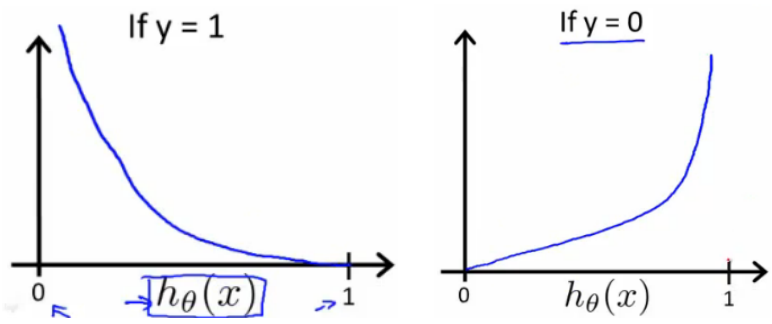
Classification

Cost Function

We define the cost function as follow for the logit:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(\, h_\theta(x^i), y^i)$$
$$Cost(h_\theta(x), y) = -\log(h_\theta(x)) \quad if y = 1$$
$$Cost(h_\theta(x), y) = -\log(1 - h_\theta(x)) \quad if y = 0$$

When y is 1, we get cost function J as shown on the left. And if y=0, the cost function is similar to the plot on the right.



The cost function is 0 when $h_\theta(x) = y$. In summary:

- If our correct answer 'y' is 0, then the cost function will be 0 if our hypothesis function also outputs 0. If our hypothesis approaches 1, then the cost function will approach infinity.
- If our correct answer 'y' is 1, then the cost function will be 0 if our hypothesis function outputs 1. If our hypothesis approaches 0, then the cost function will approach infinity.

Simplified Cost Function and Gradient Descent

We can combine the cost function and rewrite it as:

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

Or rewrite it in vectorize format:

Classification

$$h = g(X\theta),\ J(\theta) = \frac{1}{m}.(-y^T \log(h) - (1-y)^T \log(1-h)$$

We may use gradient descent for logit cost function as well, where we get the derivative if J and plug it into the cost function equation:

Repeat {

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

}

Instead of gradient descent we may use other methods such as BFGS, L-BFGS to do the minimization, however the gradient descent is the one that's mostly common used.

One Vs All (OVA)

If we have more than two categories and classes such 0,1, 2..., n, OVA are the solution. In this case we divide the classes to n+1 problem. In each one we predict of probability of being in each other classes and assign it to the class with highest probability.

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

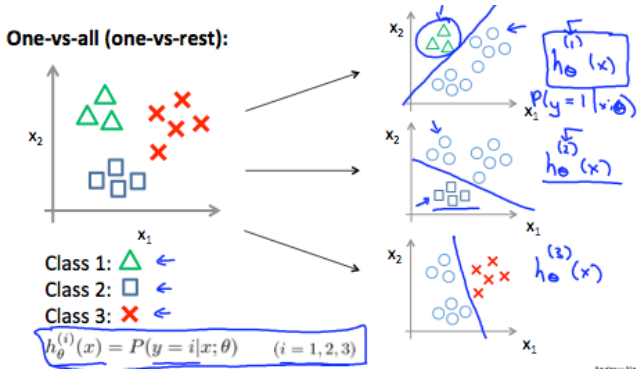
$h_\theta^{(0)}(x) = P(y = 0|x;\theta)$

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

....

$h_\theta^{(n)}(x) = P(y = n|x;\theta)$

prediction=$max_i h_\theta(x^i)$



In summary:

- Train a logistic regression classifier $h_\theta(x)$ for each class to predict the probability that y=i
- To make a prediction on a new x, pick the class that maximizes $h_\theta(x)$