Week3-Logistic regression

Used for classification problem, where y is discrete value or y $\in$ {0,1}

- 0: “Negative class” - NO
- 1: “Positive class” - YES

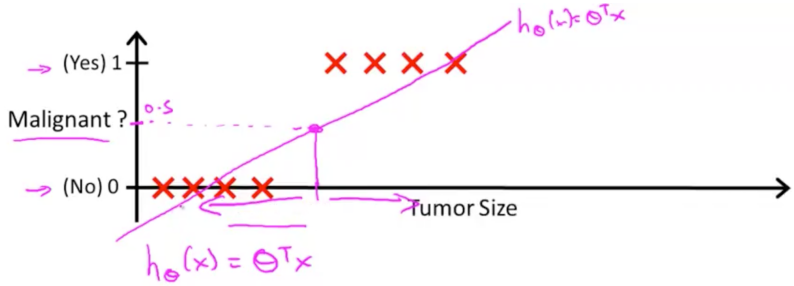
Classification problem examples:

- Email: spam or not
- Online transaction: fraudulent or not
- Tumor: Malignant or Benign
  - Benign is negative class
  - Malignant is positive class

Classification problem can also be extended to multi-value-classification, for example y $\in$ {0,1,2,3}

Why not use linear regression?

Take tumor problem for example



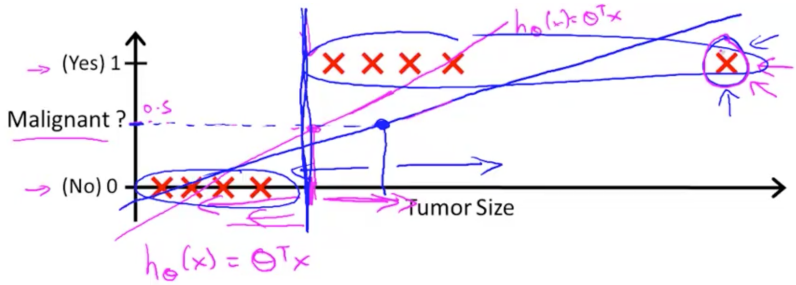
→ Threshold classifier output $h_\theta(x)$ at 0.5:

→ If $h_\theta(x) \geq 0.5$, predict “y = 1”

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

- It looks good at this particular example, we draw a straight line that fits the dataset.
- Select a threshold value in the middle, tumors smaller than that is benign and tumors larger than that is malignant.

But when we add an extra value to it, things get worse



→ 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”

- We added one extra data to the dataset.
- The straight line gets much less steep, threshold moved right
- Some of the tumors that used to be classified as malignant now being classified as benign.

Also…

In classification problem, y can only be 0 or 1
In linear regression, $h_\theta(x)$ can be larger than 1 or smaller than 0
However, with Logistic Regression, $h_\theta(x)$ will always be between 0 and 1

Classification: y = 0 or 1

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

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

Hypothesis Representation

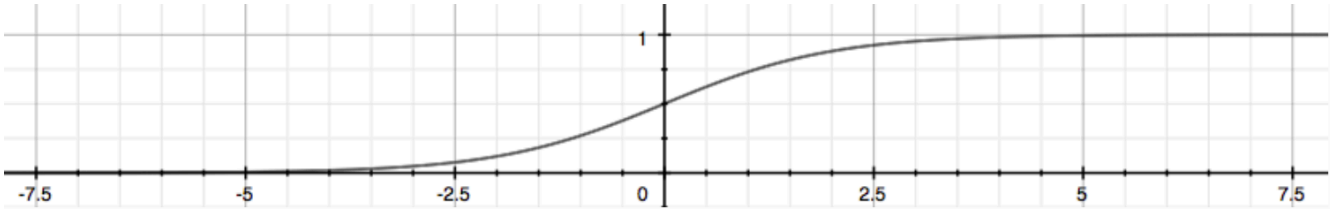
Sigmoid Function (Logistic Function)

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

$$z = \theta^T x$$

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

Graph of Sigmoid Function



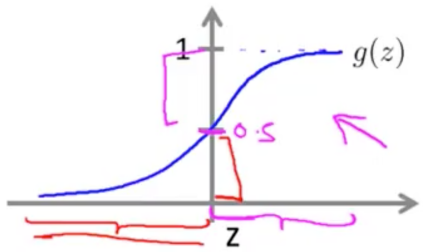
- The function g(z), shown here, maps any real number to the (0, 1) interval, making it useful for transforming an arbitrary-valued function into a function better suited for classification.
- hθ(x) will give us the probability that our output is 1.
    - For example, hθ(x)=0.7 gives us a probability of 70% that our output is 1.
- $$h_\theta(x) = P(y = 1|x; \theta) = 1 - P(y = 0|x; \theta)$$
  $$P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$$

Decision Boundary

Logistic regression

$h_\theta(x) = g(\theta^T x) = P(y=1 \mid x; \theta)$

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



Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

$\theta^T x \geq 0$

$g(z) \geq 0.5$ when $z \geq 0$

$h_\theta(x) = g(\theta^T x) \geq 0.5$ whenever $\theta^T x \geq 0$

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

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

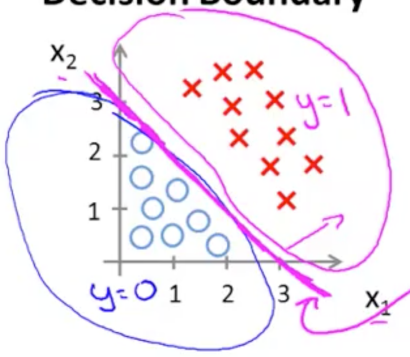
$\theta^T x < 0$

$g(z) < 0.5$

- h θ (x)≥0.5→y=1; h θ (x)<0.5→y=0
- g(z)≥0.5whenz≥0
- h θ (x)=g($\theta^T x$)≥0.5when$\theta^T x \geq 0$
- $\theta^T x \geq 0 \Rightarrow$y=1; $\theta^T x$ <0⇒y=0
- **The decision boundary is the line that separates the area where y = 0 and where y = 1**

Decision Boundary

$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

-3 1 1

Decision boundary

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

$\theta^T x$

$x_1 + x_2 \geq 3$

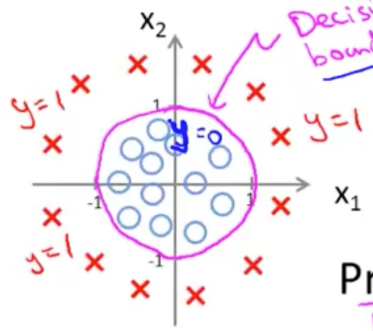
$x_1, x_2 \rightarrow h_\theta(x) = 0.5$

$x_1 + x_2 = 3$

$x_1 + x_2 < 3$

$y = 0$

Non-linear decision boundaries



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

$$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

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

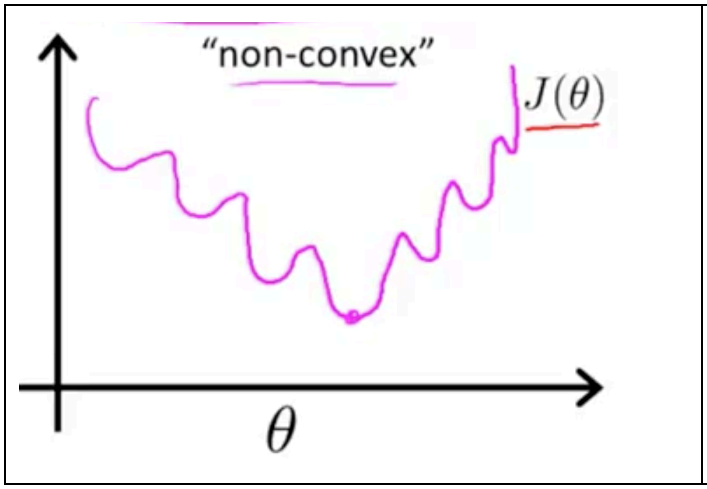
$x_1^2 + x_2^2 = 1$

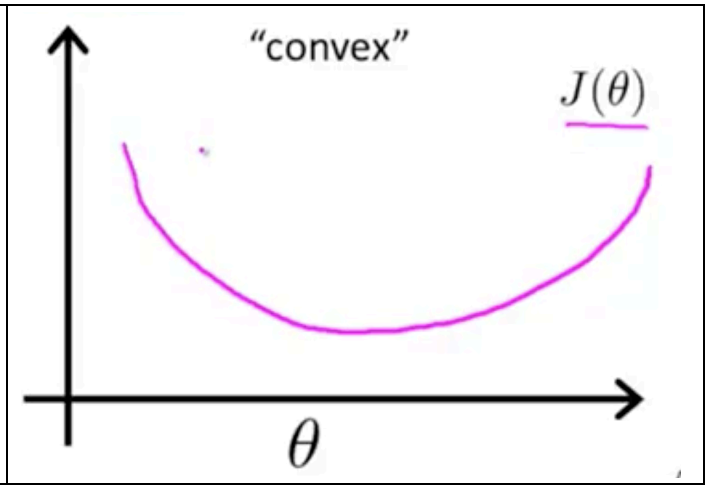
$x_1^2 + x_2^2 \geq 1$

Cost Function

Non-convex VS convex

- Non-convex has more than one local minima
- Convex has one universal minima



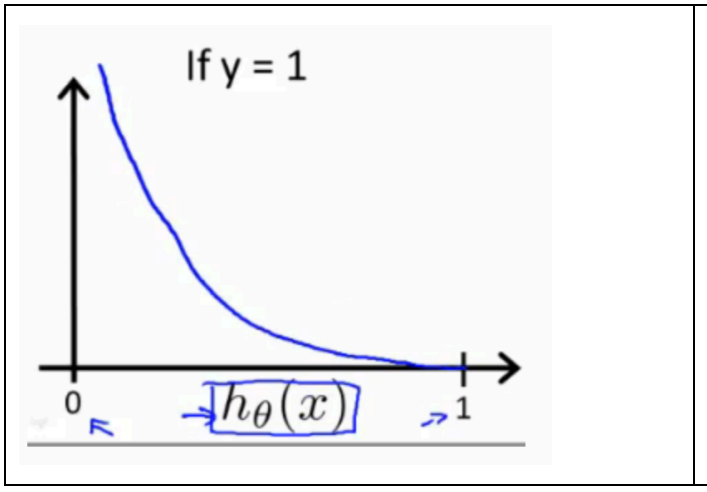


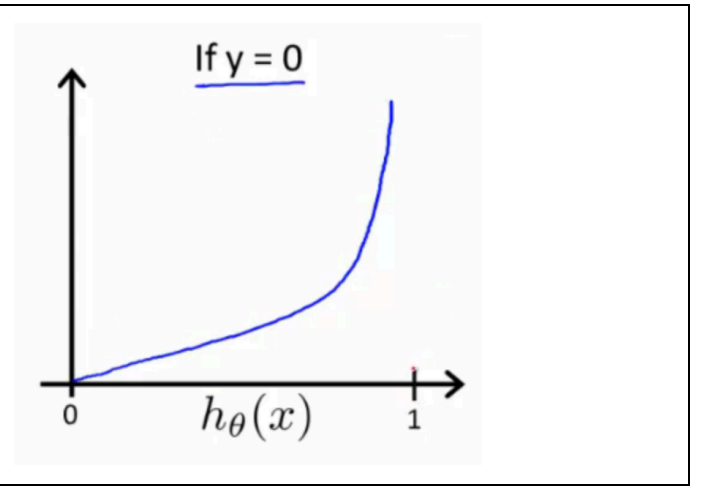
Cost function

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

$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \quad \text{if y = 1}$$

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \quad \text{if y = 0}$$





$$\text{Cost}(h_\theta(x), y) = 0 \text{ if } h_\theta(x) = y$$
$$\text{Cost}(h_\theta(x), y) \to \infty \text{ if } y = 0 \text{ and } h_\theta(x) \to 1$$
$$\text{Cost}(h_\theta(x), y) \to \infty \text{ if } y = 1 \text{ and } h_\theta(x) \to 0$$

- 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.
- Note that writing the cost function in this way guarantees that J(θ) is convex for logistic regression.

Simplified Cost Function and Gradient Descent

Simplified Cost Function

$$\text{Cost}(h_\theta(x), y) = -y \ \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

or

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

Vectorized implementation

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

Gradient Descent

$$\begin{aligned} &Repeat\ \{ \\ &\quad \theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \\ &\} \end{aligned}$$

Vectorized implementation

$$\theta := \theta - \frac{\alpha}{m} X^T (g(X\theta) - \vec{y})$$

Advanced Optimization

Optimization algorithms: - Gradient descent - Conjugate gradient - BFGS - L-BFGS	Advantages: - No need to manually pick α - Often faster than gradient descent. Disadvantages: - More complex

Function calculate J value and gradient

```
function [jVal, gradient] = costFunction(theta)
  jVal = [...code to compute J(theta)...];
  gradient = [...code to compute derivative of J(theta)...];
end
```

Then we can use octave's "fminunc()" optimization algorithm along with the "optimset()" function that creates an object containing the options we want to send to "fminunc()"

```
options = optimset('GradObj', 'on', 'MaxIter', 100);
initialTheta = zeros(2,1);
   [optTheta, functionVal, exitFlag] = fminunc(@costFunction, initialTheta, options);
```

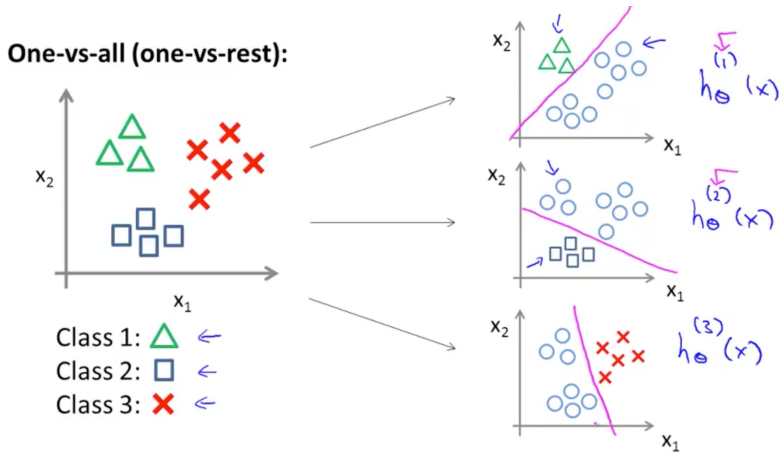
Multi-class classification

Examples:

- E-mail foldering/tagging: work, friends, family, hobby
- Medical disgrams: Not ill, cold, flu
- Weather: Sunny, Cloudy, Rain, Snow

One-vs-all

$$
\begin{aligned}
&y \in \{0, 1 \ldots n\} \\
&h_\theta^{(0)}(x) = P(y = 0 | x; \theta) \\
&h_\theta^{(1)}(x) = P(y = 1 | x; \theta) \\
&\cdots \\
&h_\theta^{(n)}(x) = P(y = n | x; \theta) \\
&\text{prediction} = \max_i (h_\theta^{(i)}(x))
\end{aligned}
$$



- Note: for n features, you need to train n logistic regression classifiers