Machine Learning

Dr. Shahid Mahmood Awan
Assistant Professor
School of Systems and Technology, University of Management and Technology
shahid.awan@umt.edu.pk

Umer Saeed(MS Data Science, BSc Telecommunication Engineering)
Sr. RF Optimization & Planning Engineer
f2017313017@umt.edu.pk

Classification

Representation

Classification

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

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

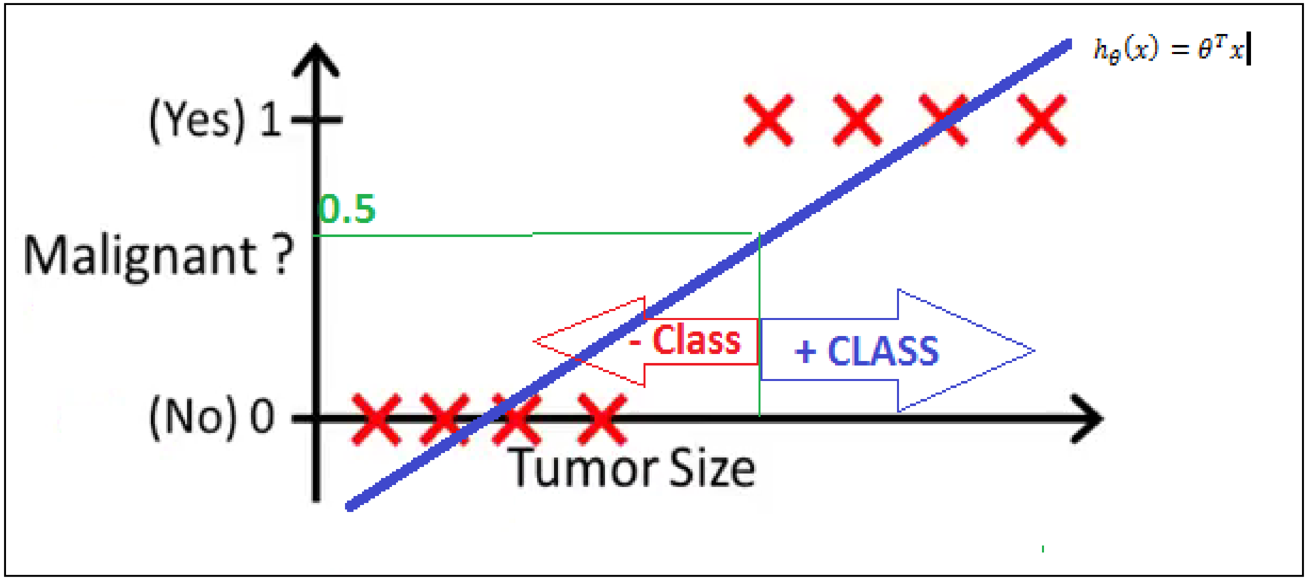
- **0:”** Negative Class**”** (e.g. benign tumor)
- **1:** ”Positive Class” (e.g. malignant tumor)

- and they are sometimes also denoted by the symbols “-” and “+.”

- Such problem also know as “Binary Classification”.

- Predict value y is discrete in classification.

Classification

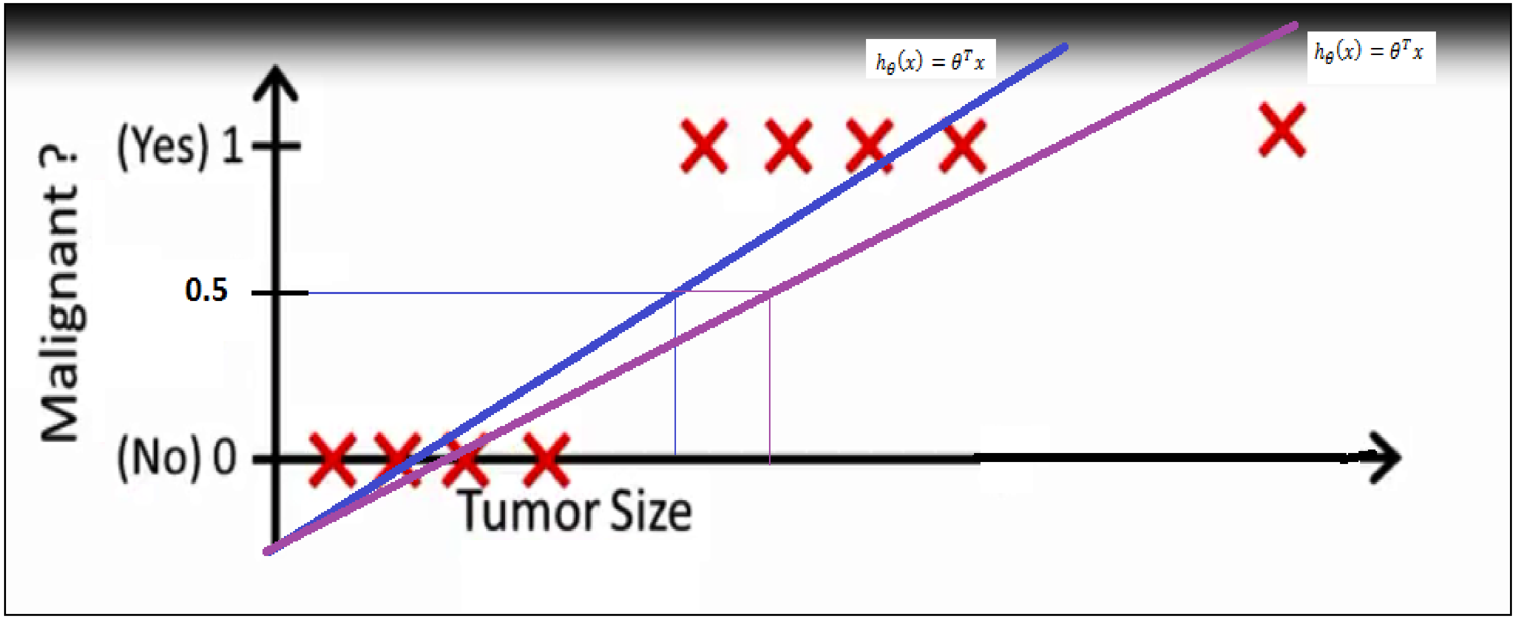


- Threshold classifier output $h_\theta(x)$ at 0.5

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

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

Classification



- To attempt classification, one method is to use linear regression and map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0. However, this method doesn't work well because classification is not actually a linear function.

Classification

$$Classification: y = 0 \ or \ 1$$

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

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

- Logistic Regression is also know as Classification Algorithm.

Question

- Which of the following statements is true?

- (a) If linear regression doesn't work on a classification task as in the previous example shown in the video, applying feature scaling may help.

- (b) If the training set satisfies $0 \le y^{(i)} \le 1$ for every training example $(x^{(i)}, y^{(i)})$, then linear regression's prediction will also satisfy $0 \le h(x) \le 1$ for all values of x.

- (c) If there is a feature x that perfectly predicts y, i.e. if $y=1$ when $x \ge c$ and $y=0$ whenever $x < c$ (for some constant c), then linear regression will obtain zero classification error.

- **(d) None of the above statements are true. (Answer)**

Classification

Hypothesis Representation

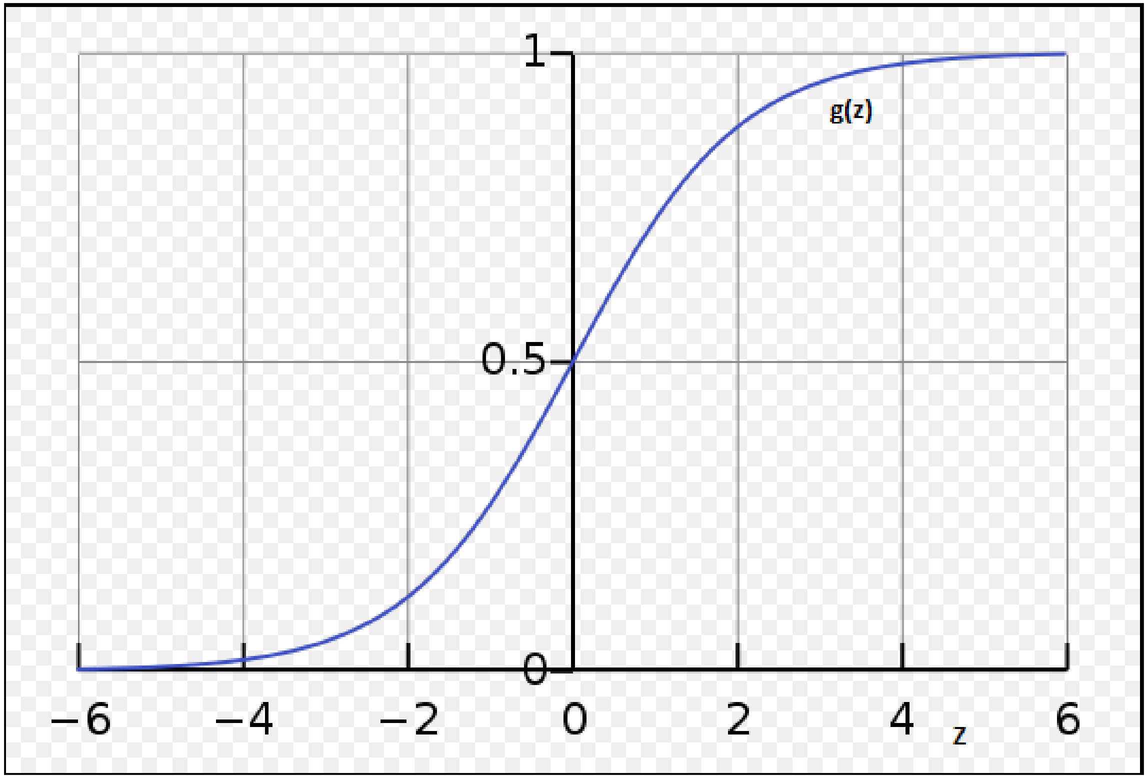
Hypothesis Representation

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

$$h_\theta(x) = \mathrm{g}(\theta^T \mathrm{x}) \qquad g(z) = \frac{1}{1+e^{-z}}\ ;\ \text{Logistic/Sigmoid function}$$

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

- 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.



Hypothesis Representation

- **Remember.**

$$
\begin{aligned}
z = 0, e^0 = 1 &\Rightarrow g(z) = 1/2 \\
z \to \infty, e^{-\infty} \to 0 &\Rightarrow g(z) = 1 \\
z \to -\infty, e^{\infty} \to \infty &\Rightarrow g(z) = 0
\end{aligned}
$$

Hypothesis Representation

- **Interpretation of Hypothesis Output**

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

- **Example**

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

- Tell patient that 70% chance of tumor being malignant.

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

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

Question

- Suppose we want to predict, from data *x* about a tumor, whether it is malignant (*y*=1) or benign (*y*=0). Our logistic regression classifier outputs, for a specific tumor, $h(x)=P(y=1|x;\theta)=0.7$, so we estimate that there is a 70% chance of this tumor being malignant. What should be our estimate for $P(y=0|x;\theta)$, the probability the tumor is benign?

$(a) P(y = 0|x;\theta) = 0.3$ Answer

$(b) P(y = 0|x;\theta) = 0.7$

$(c) P(y = 0|x;\theta) = 0.7^2$

$(d) P(y = 0|x;\theta) = 0.3 \times 0.7$

Classification

Decision Boundary

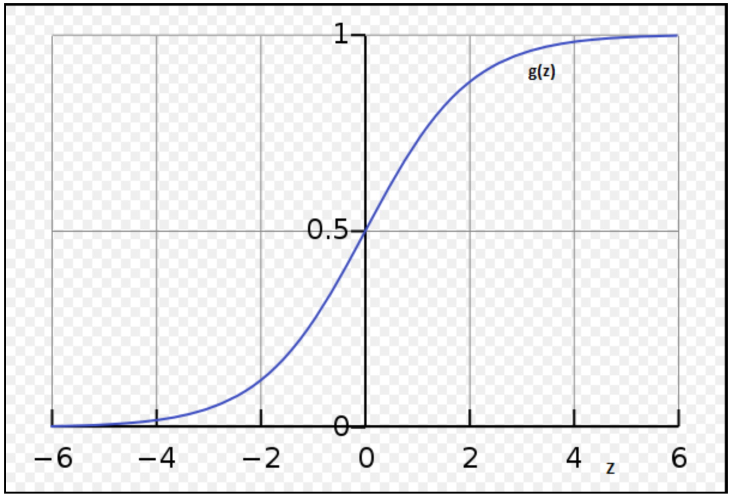
Decision Boundary

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

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

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

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

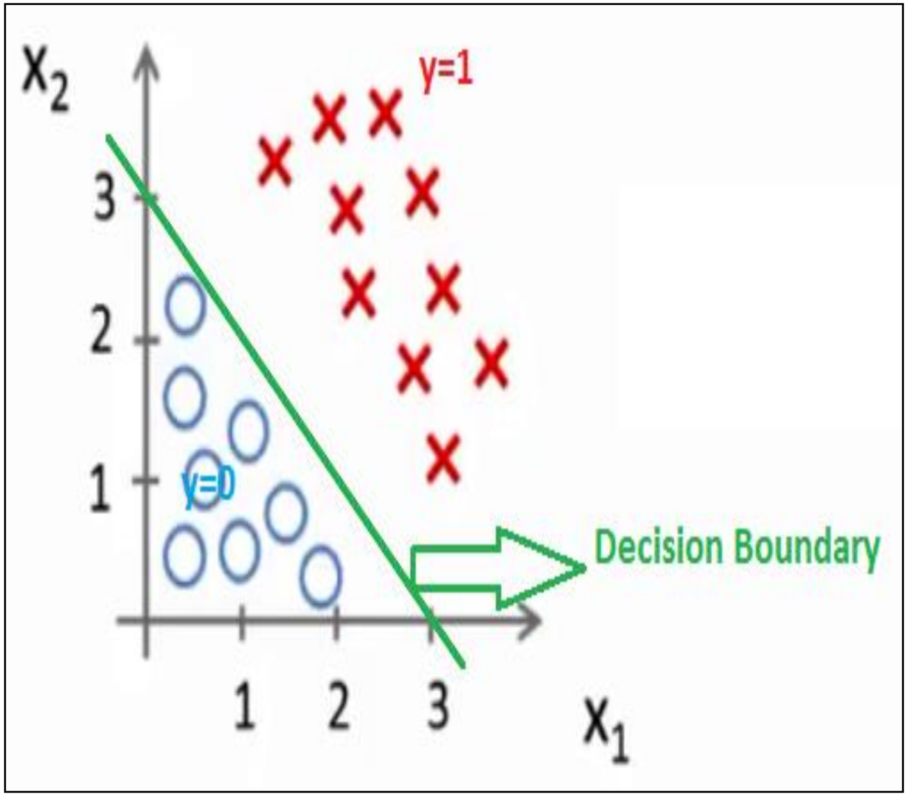


$\boldsymbol{Predict}$ "**y=0**"$if\ h_\theta(x) < 0.5$

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

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

Decision Boundary



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

$$Let\ \theta_0 = -3, \theta_1 = 1, \theta_2 = 1$$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

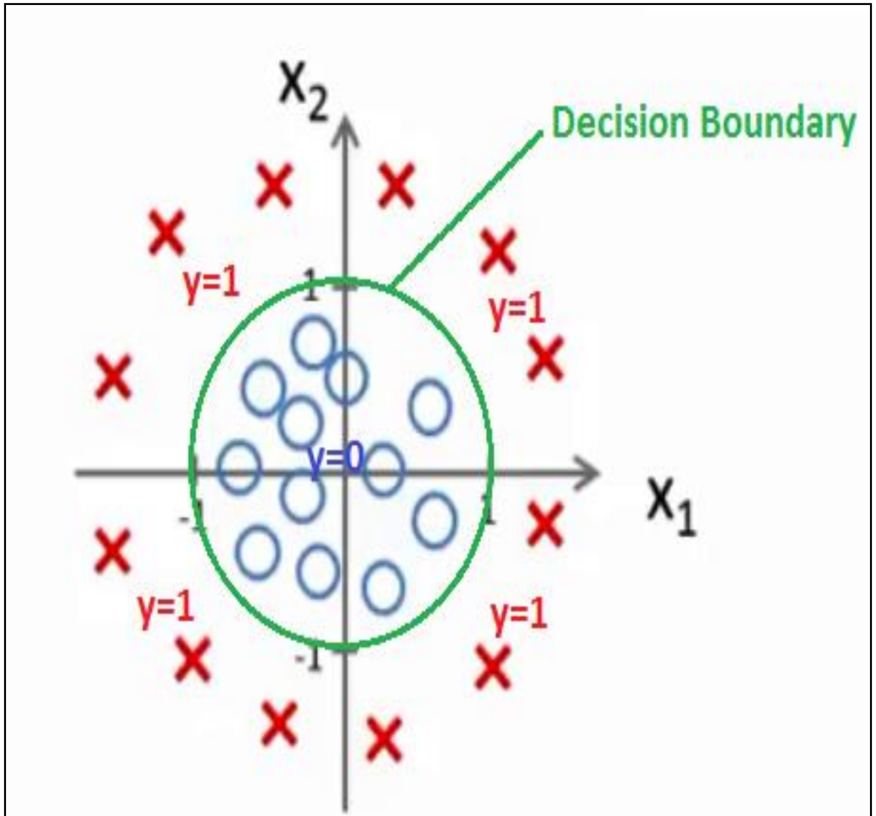
$Predict$ y=1 $if\ -3 + x_1 + x_2 \geq 0$

$$x_1 + x_2 \geq 3$$

y=0 if $x_1 + x_2 < 3$

- The **decision boundary** is the line that separates the area where y = 0 and where y = 1. It is created by our hypothesis function.

Non-Linear Decision Boundaries



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

$$Let\ \theta_0 = -1, \theta_1 = 0, \theta_2 = 0, \theta_3 = 1, \theta_4 = 1$$

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

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

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

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

- Decision Boundary is the property of the hypothesis and parameter not a training set.

Logistic Regression Model

Cost Function

Logistic Regression Cost Function

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

$$m\ examples: x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

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

$$How\ to\ choose\ parameters\ \theta?$$

Logistic Regression Cost Function

- **Linear regression:**

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

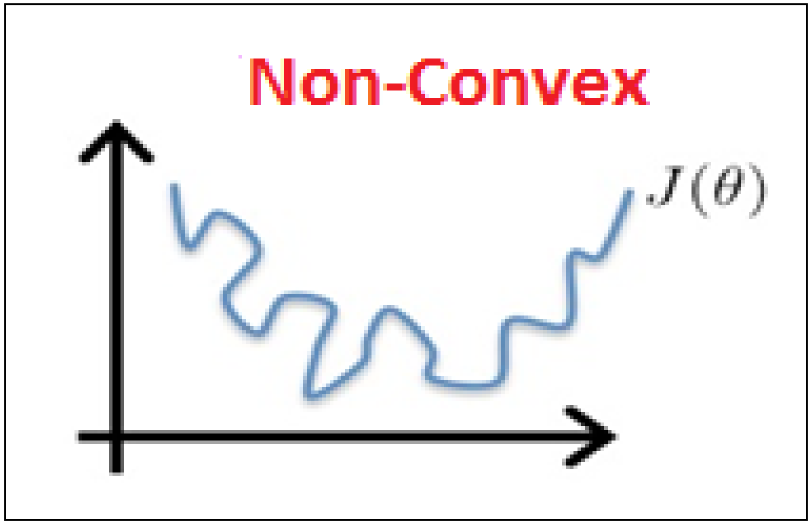
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta\left(x^i\right) - y^{(i)}\right)^2$$

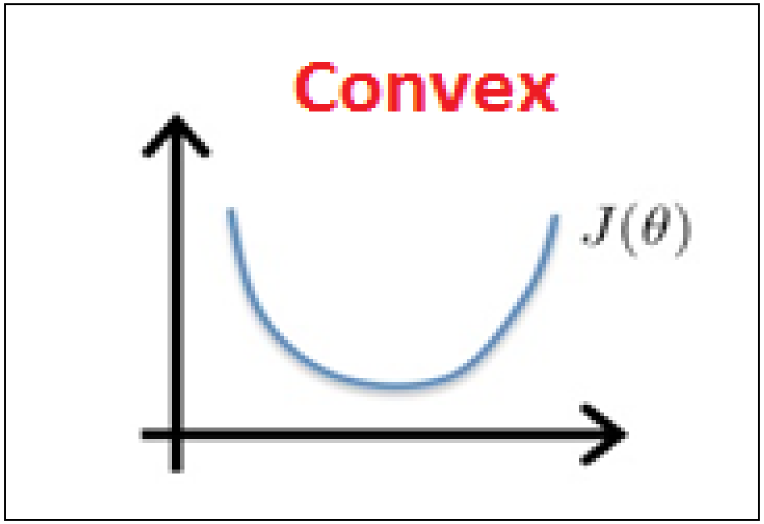
$$Cost\left(h_\theta\left(x^i\right), y^{(i)}\right) = \frac{1}{2}\left(h_\theta\left(x^i\right) - y^{(i)}\right)^2$$

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

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

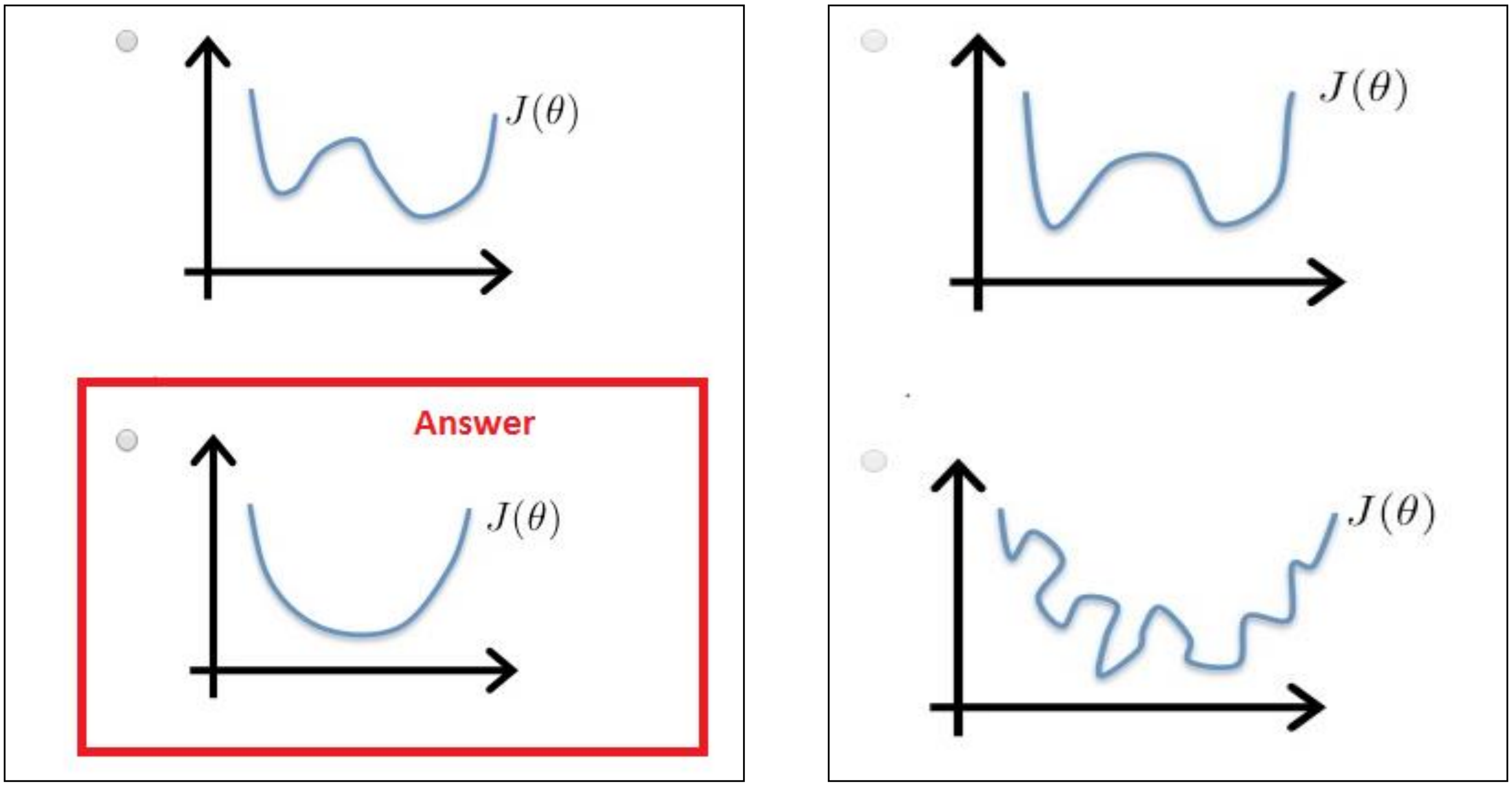
Logistic Regression Cost Function





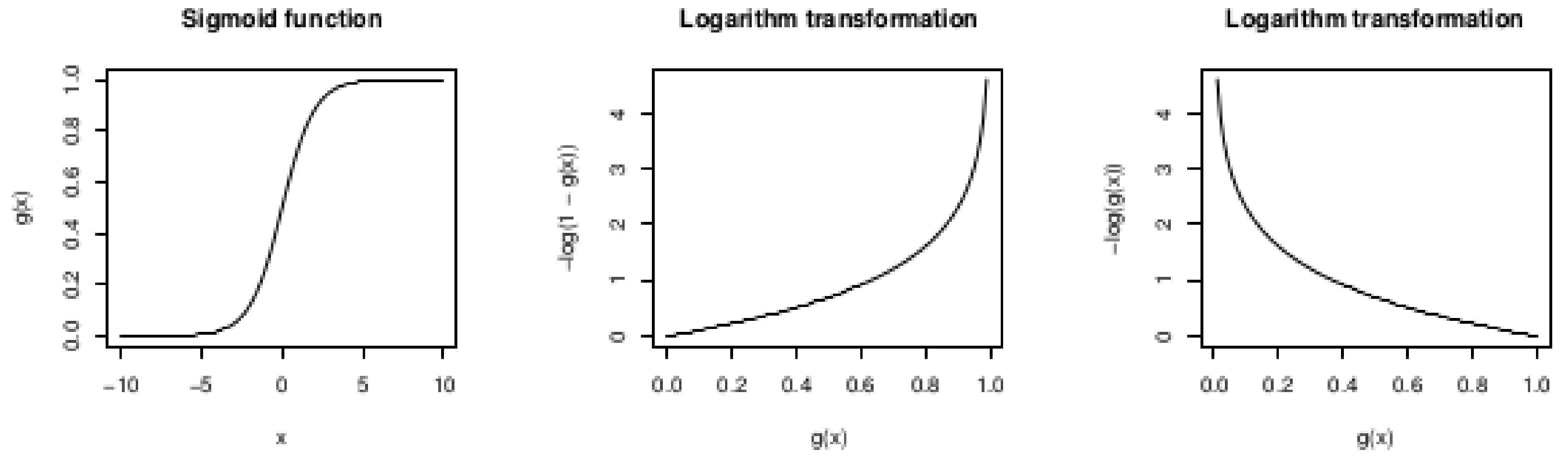
Logistic Regression Cost Function

- **Question**
- Consider minimizing a cost function $J(\theta)$. Which one of these functions is convex?



Logistic Regression Cost Function

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



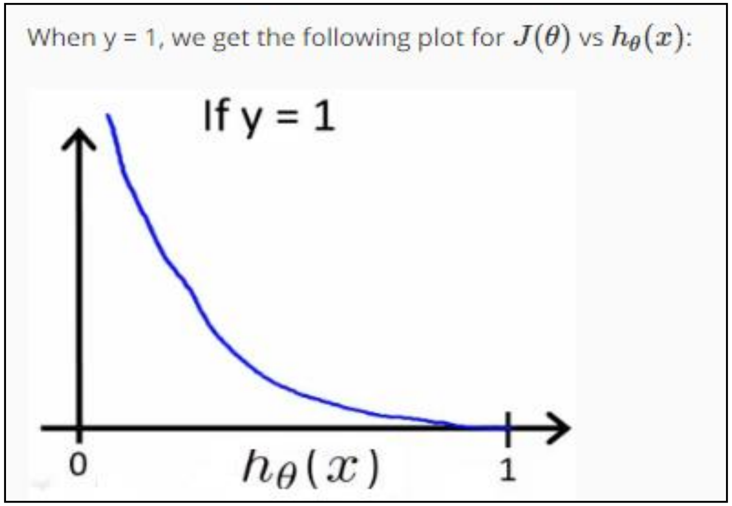
(a) *Sigmoid function.* (b) *Cost for* $y = 0$. (c) *Cost for* $y = 1$.

Logarithmic transformation of the sigmoid function.

Logistic Regression Cost Function

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

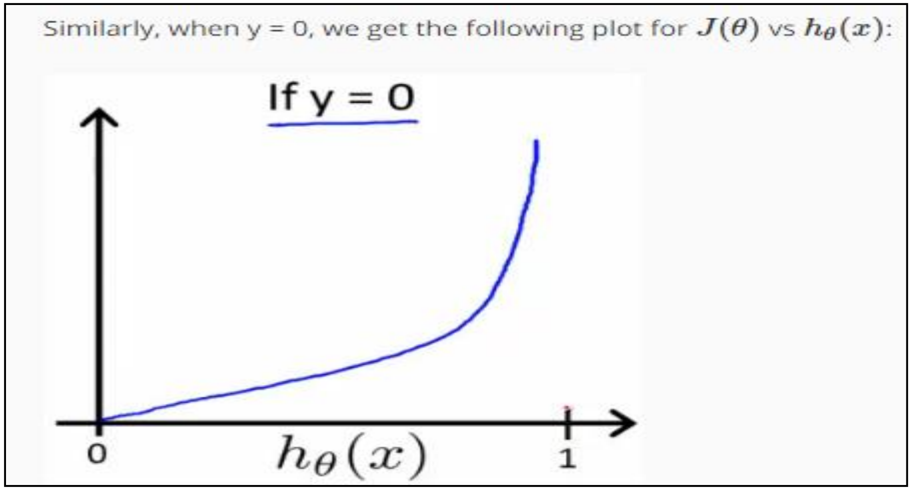
- **Case-1: If $(h_\theta(x))$=0** $\quad Cost(h_\theta(x), y) = -log(0) = \infty$
- **Case-2: If $(h_\theta(x))$=1** $\quad Cost(h_\theta(x), y) = -log(1) = 0$
- If our correct answer 'y' is 1, then the cost function will be 0 if our hypothesis function outputs 1. If our hypothesis approaches 0, then the cost function will approach infinity.
- Captures intuition that if $(h_\theta(x))$=0, (predict P(y=1|x;θ), but y=1 we'll penalize learning algorithm by a very large cost.



Logistic Regression Cost Function

$\underline{\boldsymbol{Cost(h_\theta(x), y) = -log(1 - h_\theta(x)) \;\; if \; y = 0}}$

- **Case-1:** If $(h_\theta(x))$=0 $\quad Cost(h_\theta(x), y) = -log(1-0) = -\log(1) = 0$
- **Case-2:** If $(h_\theta(x))$=1 $\quad Cost(h_\theta(x), y) = -log(1-1) = -\log(0) = \infty$
- 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.



Question

- In logistic regression, the cost function for our hypothesis outputting (predicting) *h* (*x*) on a training example that has label $y \in \{0,1\}$ is;

- [x] If $h_\theta(x) = y$, then $\text{cost}(h_\theta(x), y) = 0$ (for $y = 0$ and $y = 1$).

  Correct

- [x] If $y = 0$, then $\text{cost}(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 1$.

  Correct

- [ ] If $y = 0$, then $\text{cost}(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 0$.

  Un-selected is correct

- [x] Regardless of whether $y = 0$ or $y = 1$, if $h_\theta(x) = 0.5$, then $\text{cost}(h_\theta(x), y) > 0$.

  Correct

Logistic Regression Model

Simplified Cost Function and gradient descent

Logistic Regression Cost Function

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

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

- Note: y=0 or y=1 always.

$$Cost(h_\theta(x), y) = -y(log(h_\theta(x)) - (1 - \; y) \; log(1 - h_\theta(x))$$
$$Cost(h_\theta(x), y) = -[y\left(log(h_\theta(x)\right) + (1 - \; y) \; log(1 - h_\theta(x))]$$

$$\boldsymbol{Case - 1 \; (y = 0)}$$
$$Cost(h_\theta(x), y) = -(0)(log(h_\theta(x)) - (1 - \; 0) \; log(1 - h_\theta(x)) = -log(1 - h_\theta(x))$$
$$\boldsymbol{Case - 1 \; (y = 1)}$$
$$Cost(h_\theta(x), y) = -(1)(log(h_\theta(x)) - (1 - 1) \; log(1 - h_\theta(x)) = -log(h_\theta(x)$$

Logistic Regression Cost Function

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

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

- This cost function is drives from statistics using the principle max likelihood estimation.
- The property of max likelihood estimation that is “convex”.

$$\boldsymbol{To\ fit\ parameters\ \theta}\text{:}\ \underset{\theta}{min}(J(\theta))$$

- To make a prediction given new x;

$$Logistic\ Regression\ Hypothesis\text{:}\ h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

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

Logistic Regression Cost Function

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

- $We\ want\ \ \begin{matrix} min \\ \theta \end{matrix}\big(J(\theta)\big)$.
- To do so, we use gradient descent, but we first need to find the partial derivatives $\frac{\partial}{\partial \theta_j}$

Logistic Regression Cost Function

- We're going to make use of a neat property of the logistic functions:

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

$$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{v\,\frac{du}{dx} - u\,\frac{dv}{dx}}{v^2}$$

$$\frac{d}{dx} e^x = e^x$$

- By taking the derivatives on the both sides, we have

$$h_\theta(\acute{x}) = \frac{d}{dx}\left[\frac{1}{1 + e^{-\theta^T x}}\right]$$

$$h_\theta(\acute{x}) = \frac{\left(1 + e^{-\theta^T x}\right)\frac{d}{dx}(1) - (1)\frac{d}{dx}\left(1 + e^{-\theta^T x}\right)}{(1 + e^{-\theta^T x})^2}$$

$$h_\theta(\acute{x}) = \frac{\left(1 + e^{-\theta^T x}\right)(0) - \left[\frac{d}{dx}(1) - \frac{d}{dx}(e^{-\theta^T x})\right]}{(1 + e^{-\theta^T x})^2}$$

Logistic Regression Cost Function

$$h_\theta(\acute{x}) = \frac{e^{-\theta^T x}}{(1 + e^{-\theta^T x})^2}$$

$$h_\theta(\acute{x}) = \frac{1 + e^{-\theta^T x} - 1}{(1 + e^{-\theta^T x})^2}$$

$$h_\theta(\acute{x}) = \frac{1}{1 + e^{-\theta^T x}} - \frac{1}{(1 + e^{-\theta^T x})^2}$$

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

$$h_\theta(\acute{x}) = h_\theta(x)[1 - h_\theta(x)]$$

Logistic Regression Cost Function

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

- By taking the derivatives on the both sides, we have

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

$$\begin{aligned}\frac{\partial}{\partial\theta_j}J(\theta) &= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\frac{x_j^{(i)}}{h_\theta(x^{(i)})}h_\theta(x^{(i)})(1-h_\theta(x^{(i)})) - (1-y^{(i)})\frac{x_j^{(i)}}{1-h_\theta(x^{(i)})}h_\theta(x)(1-h_\theta(x^{(i)}))\right]\\ &= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}x_j^{(i)}(1-h_\theta(x^{(i)})) - (1-y^{(i)})x_j^{(i)}h_\theta(x^{(i)}))\right]\\ &= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}x_j^{(i)} - x_j h_\theta(x^{(i)})\right]\end{aligned}$$

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

Gradient Descent

Repeat {

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

(Simultaneously update all θ_j)

}

Repeat {

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

(Simultaneously update all θ_j)

}

- Algorithm looks identical to linear regression!

Logistic Regression Model

Advanced Optimization

Advanced Optimization

- 1- "Conjugate gradient"
- 2- "BFGS"
- 3- "L-BFGS"
- are more sophisticated, faster ways to optimize θ that can be used instead of gradient descent.

Logistic Regression Model

Multi-Class Classification: One-Vs-all

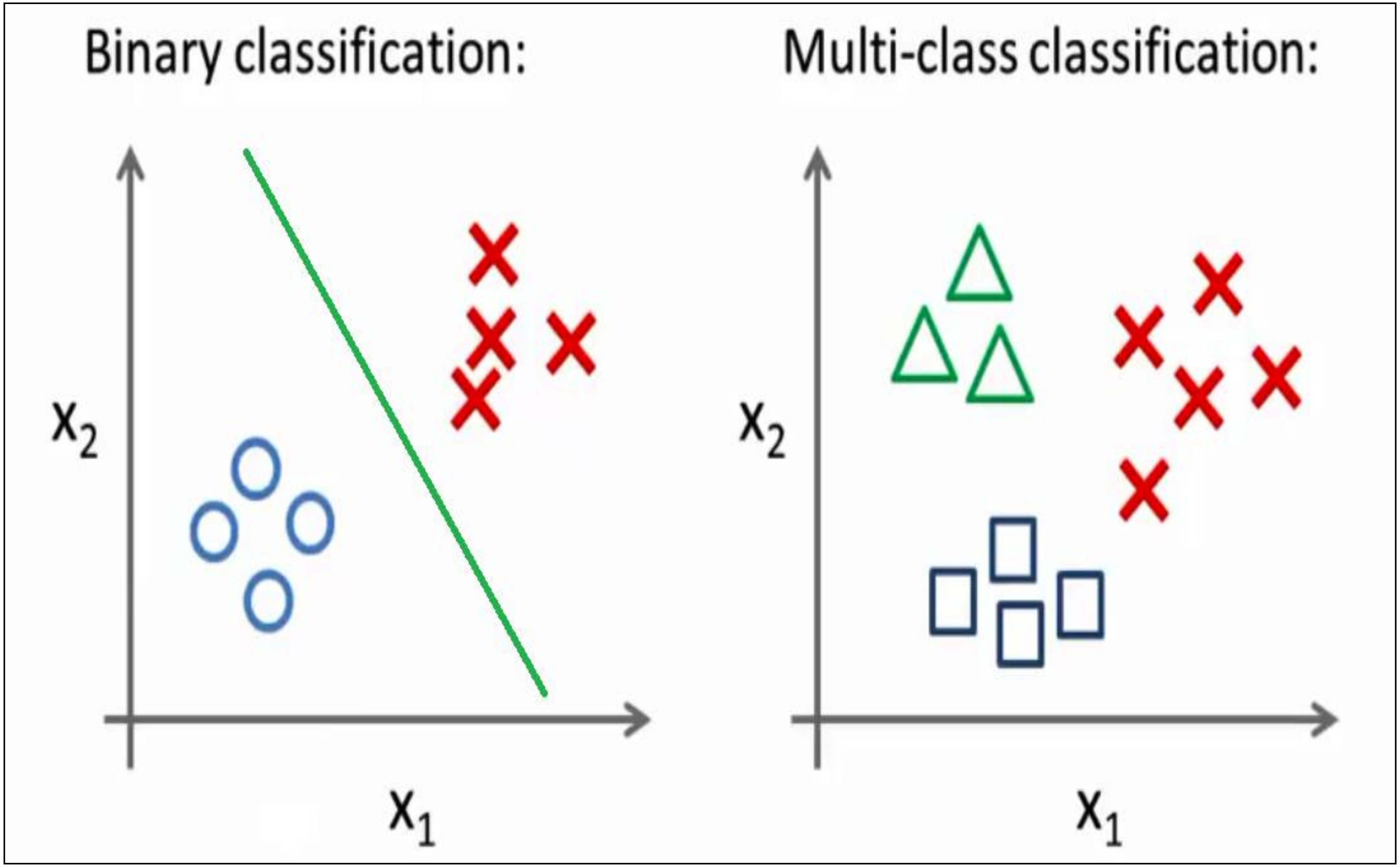
Multiclass Classification: One-vs-all

- Now we will approach the classification of data when we have more than two categories. Instead of y = {0,1} we will expand our definition so that y = {0,1...n}.

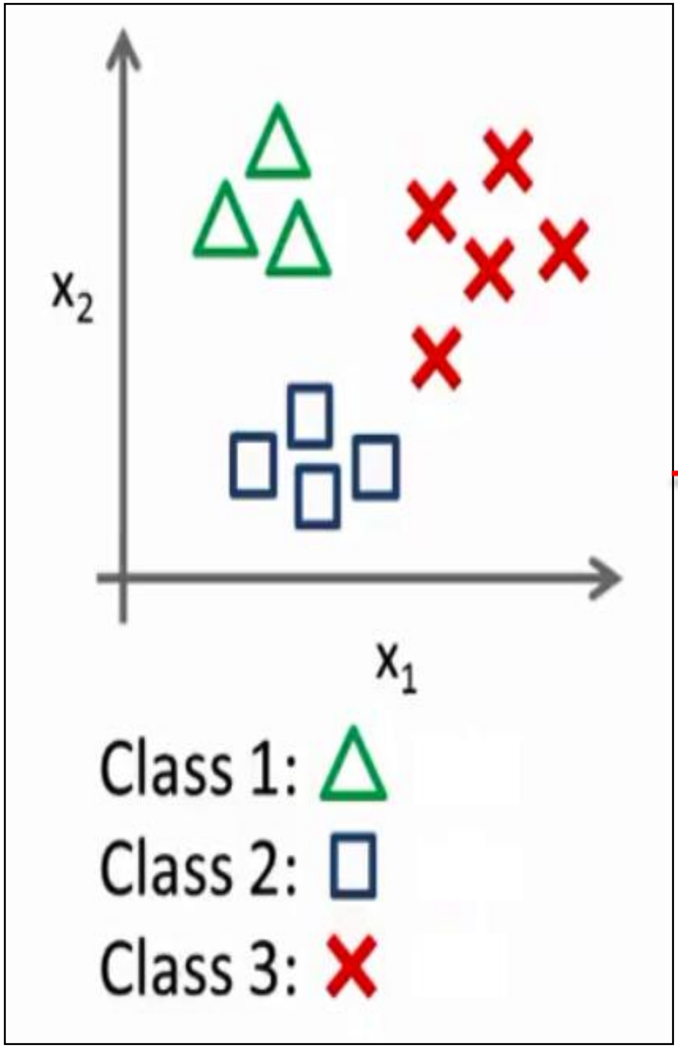
- **Examples:**

Code(format-1)	Code(format-2)	Email Foldering/Tagging	Medical Diagrams	Weather
0	1	Work	Not Till	Sunny
1	2	Friends	Cold	Cloudy
2	3	Family	Flu	Rain
3	4	Hobby		Snow

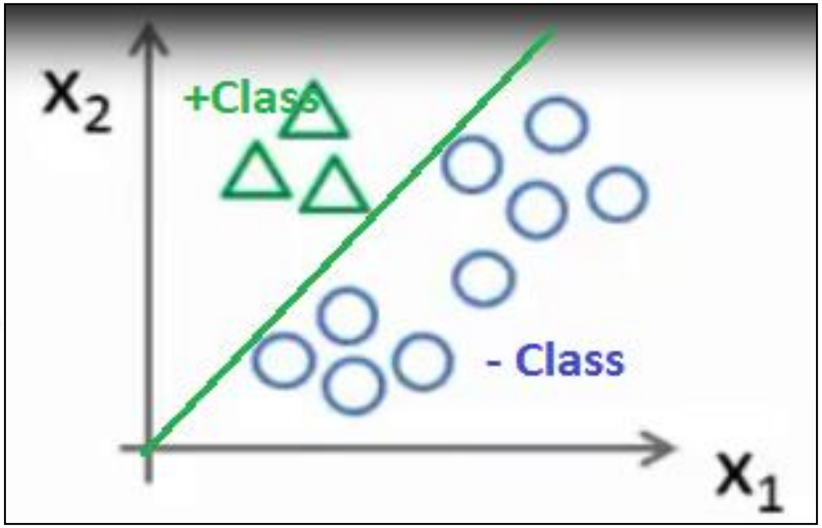
Multiclass Classification: One-vs-all



Multiclass Classification: One-vs-all (one-vs-Rest)

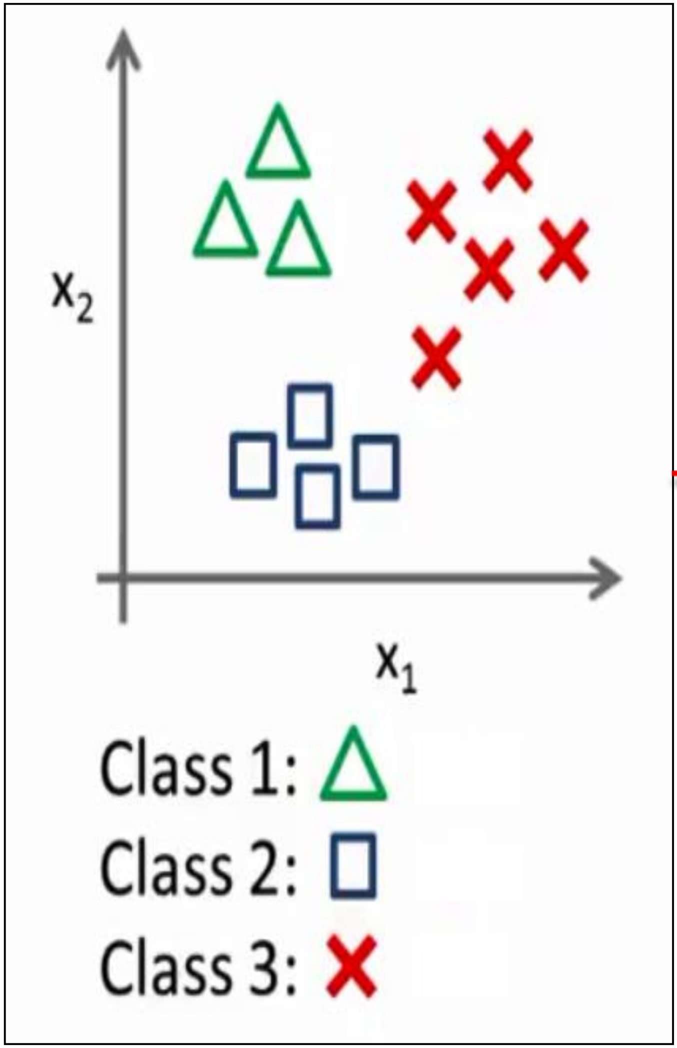


$h_\theta^{(1)}$: *Superscript 1 is stands for class – "1"*

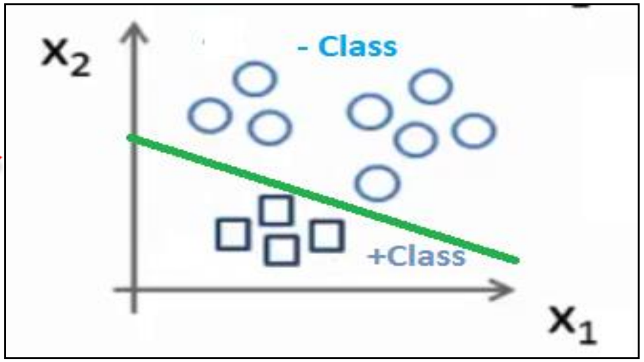


$$P(y = 1|x; \theta)$$

Multiclass Classification: One-vs-all (one-vs-Rest)

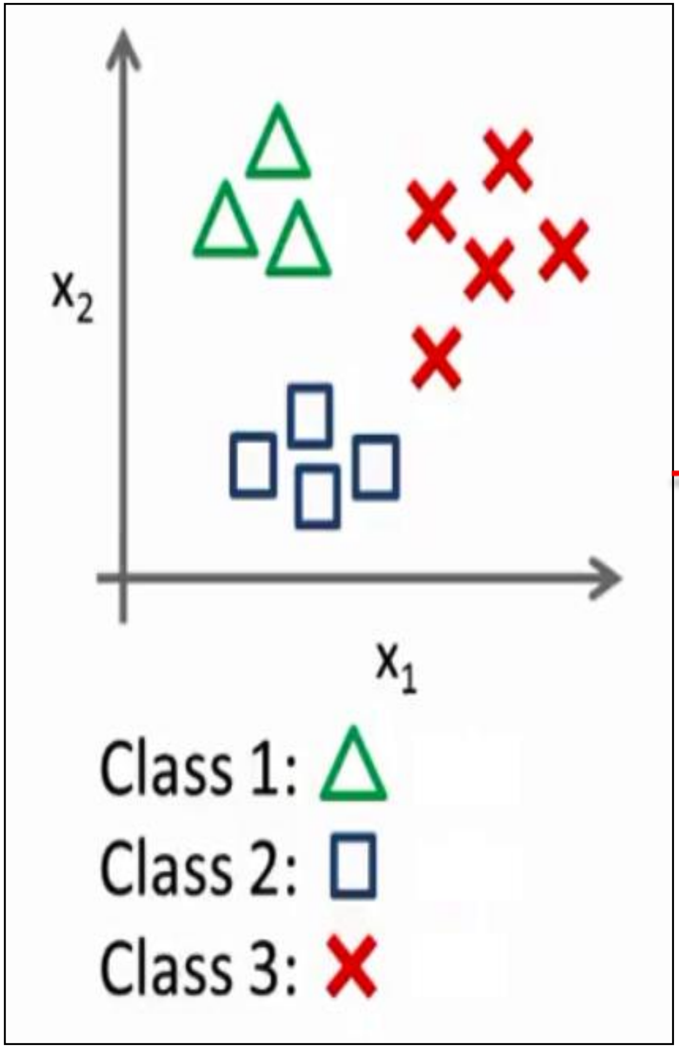


$h_\theta^{(2)}$: *Superscript 2 is stands for class – "2"*

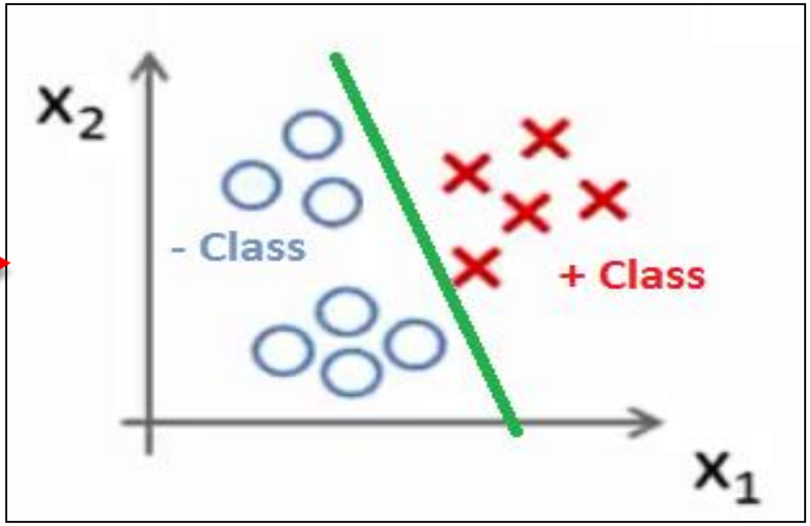


$$P(y = 2|x; \theta)$$

Multiclass Classification: One-vs-all (one-vs-Rest)



$h_{\theta}^{(3)}$: *Superscript 3 is stands for class – "3"*



$$P(y = 3|x; \theta)$$

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

Multiclass Classification: One-vs-all (one-vs-Rest)

- 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.

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

Multiclass Classification: One-vs-all

- Since y = {0,1...n}, we divide our problem into n+1 (+1 because the index starts at 0) binary classification problems; in each one, we predict the probability that 'y' is a member of one of our classes.

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

- We are basically choosing one class and then lumping all the others into a single second class.

- We do this repeatedly, applying binary logistic regression to each case, and then use the hypothesis that returned the highest value as our prediction.

Question

- Suppose you have a multi-class classification problem with *k* classes (so $y \in \{1,2,\ldots,k\}$). Using the 1-vs.-all method, how many different logistic regression classifiers will you end up training?

$$(a)\ k - 1$$

$$(b) k \quad (answer)$$

$$(c) k + 1$$

$$(d) Approximately\ log_2(k)$$

Logistic Regression Model

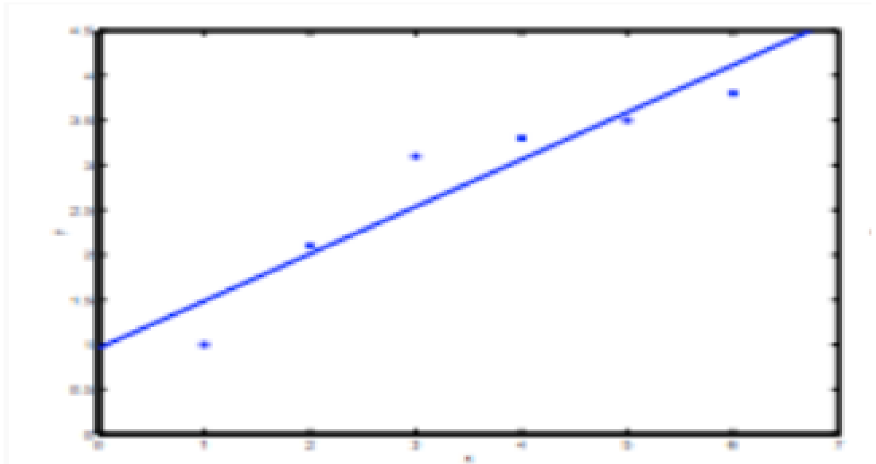
The Problem of Over fitting

The Problem of Over fitting (Linear Regression)

- Consider the problem of predicting y from x ∈ R.

- The figure shows the result of fitting a $y = \theta_0 + \theta_1 x$ to a dataset.

- We see that the data doesn't really lie on straight line, and so the fit is not very good.



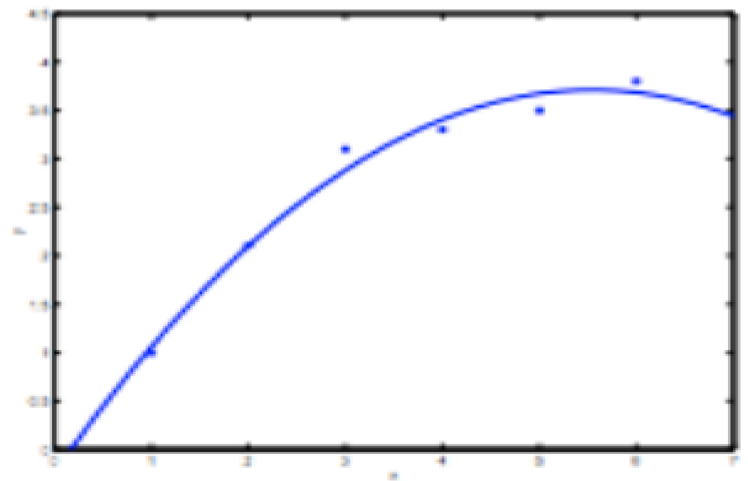
$\theta_0 + \theta_1 x$

underfitting

- we'll say the figure shows an instance of **under-fitting**—in which the data clearly shows structure not captured by the model.

The Problem of Over fitting (Linear Regression)

- Instead, if we had added an extra feature x^2 , and fit $y=\theta_0+\theta_1 x+\theta_2 x^2$, then we obtain a slightly better fit to the data.



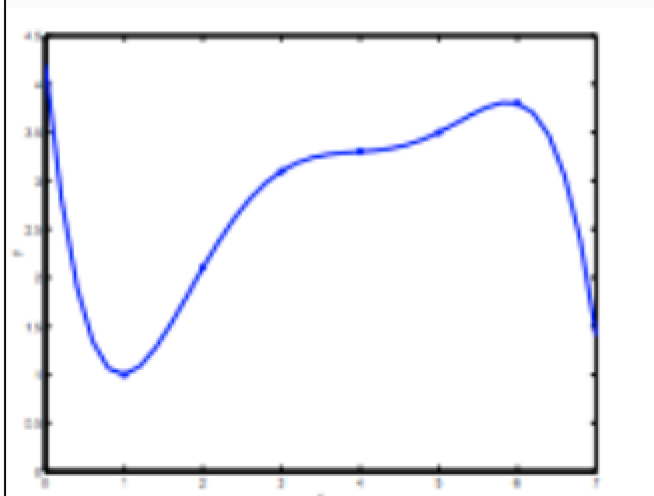
$$\theta_0 + \theta_1 x + \theta_2 x^2$$

- Naively, it might seem that the more features we add, the better.

The Problem of Over fitting (Linear Regression)

- There is also a danger in adding too many features: The figure is the result of fitting a 5^{th} order polynomial

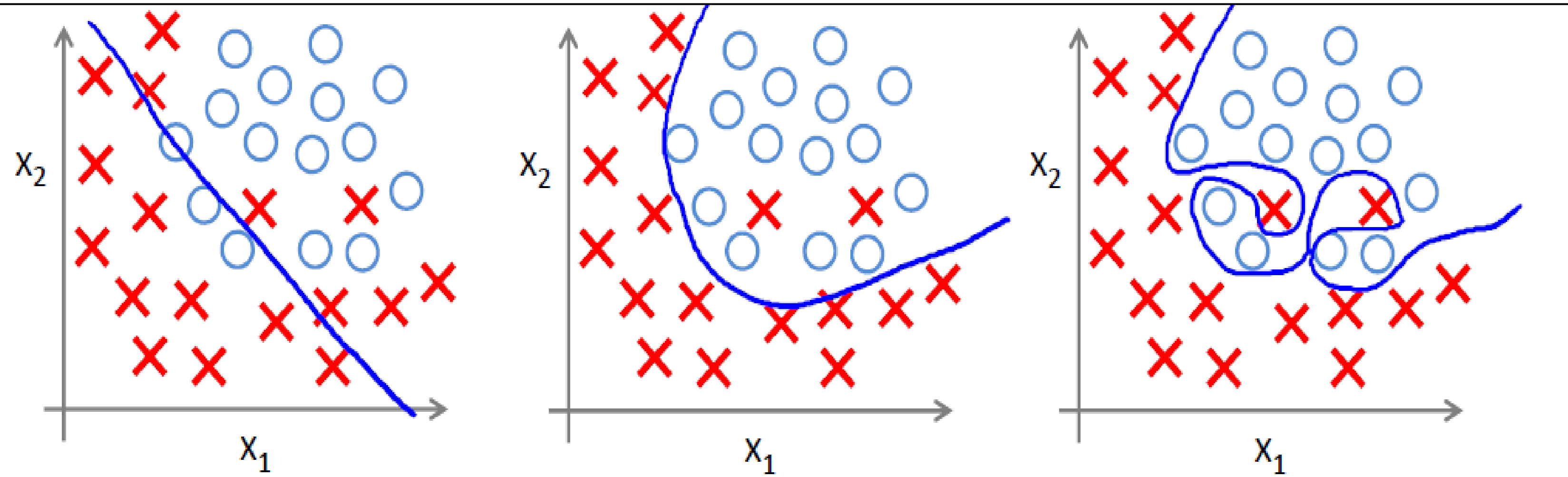
- We see that even though the fitted curve passes through the data perfectly, we would not expect to be a very good predictor.



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

- we'll say the figure shows an instance of **over-fitting**—in which the data clearly shows structure not captured by the model.

The Problem of Over fitting (Logistic Regression)



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

(g = sigmoid function)

underfitting

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

overfitting

- This terminology is applied to both linear and logistic regression

The Problem of Over fitting

- **Under-fitting, or high bias**, is when the form of our hypothesis function h maps poorly to the trend of the data.
- It is usually caused by a function that is too simple or uses too few features.

- At the other extreme, **over-fitting, or high variance**, is caused by a hypothesis function that fits the available data but does not generalize well to predict new data.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h(x)^{(i)} - y^{(i)})^2 \approx 0$$

- It is usually caused by a complicated function that creates a lot of unnecessary curves and angles unrelated to the data.

Addressing Over-fitting

- There are two main options to address the issue of over-fitting:
- **1- Reduce number of features.**
- - Manually select which features to keep.
- - *Model section algorithm*

- **2- Regularization**
- **-** Keep all the features, but reduce the magnitude/values of parameters θ_j
- **-** Works well when we have a lot of features, each of which contributes a bit to predicting y.

Question

- Consider the medical diagnosis problem of classifying tumors as malignant or benign. If a hypothesis $h(x)$ has overfit the training set, it means that:

- (a) It makes accurate predictions for examples in the training set and generalizes well to make accurate predictions on new, previously unseen examples.

- (b) It does not make accurate predictions for examples in the training set, but it does generalize well to make accurate predictions on new, previously unseen examples.

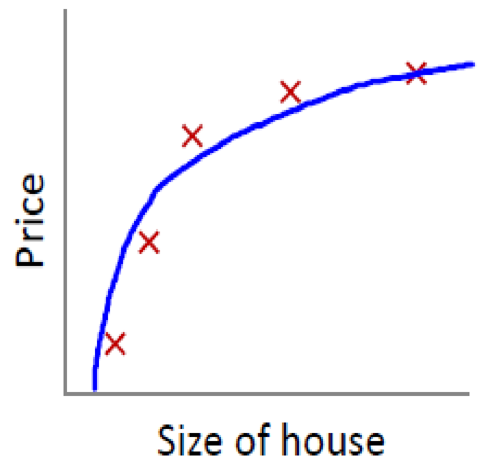
- (c) It makes accurate predictions for examples in the training set, but it does not generalize well to make accurate predictions on new, previously unseen examples. (Answer)

- (d) It does not make accurate predictions for examples in the training set and does not generalize well to make accurate predictions on new, previously unseen examples.

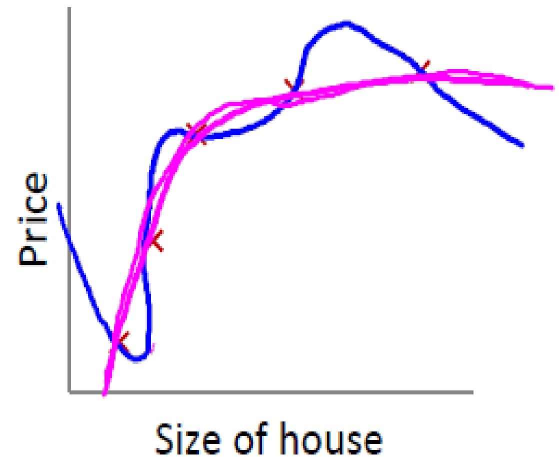
Linear Regression Model

Regularization-Cost Function

Regularization



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

Suppose we penalize and make θ_3, θ_4 really small.

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + 1000\,\theta_3^2 + 1000\,\theta_4^2$$

$$\theta_3 \approx 0 \qquad \theta_4 \approx 0$$

Regularization

- Small values of parameters $\theta_0, \theta_1, \ldots \theta_n$
- - "Simples" hypothesis
- - Less prone to over-fitting

- **Example (Housing)**

$$Features: x_1, x_2 \ldots, x_{100}$$

$$Parameters: \theta_0, \theta_1, \ldots, \theta_{100}$$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} \theta_j^2\right]$$

Regularization

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} \theta_j^2\right]$$

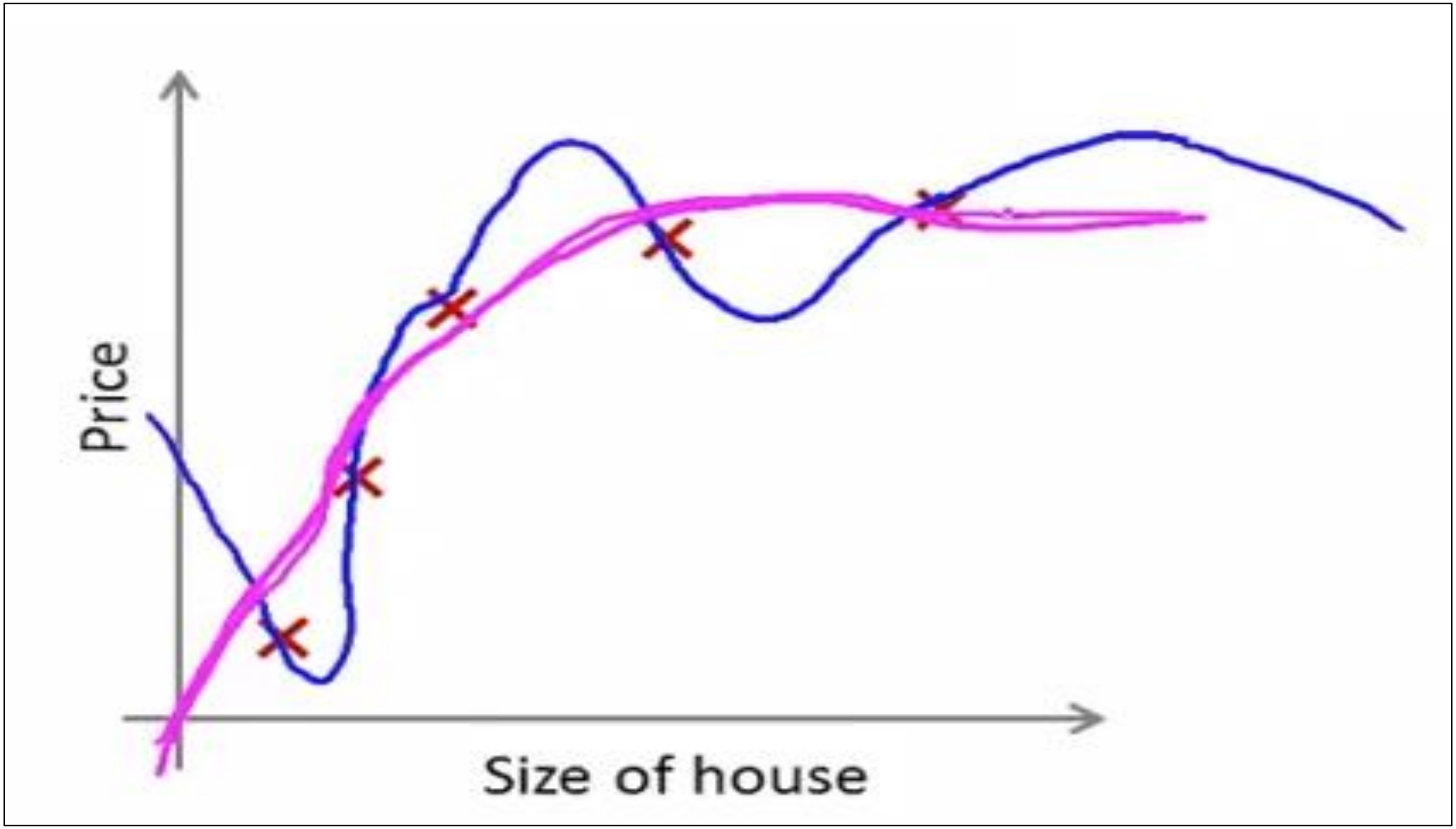
- The λ, or lambda, is the **regularization parameter**. It determines how much the costs of our theta parameters are inflated.

- **λ is control the trade off between 2 different goals:**
- **1-** $\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$ fit the training data set well.
- 2- $\sum_{i=1}^{n} \theta_j^2$ Keeps the parameter small.

- and therefore keeping hypothesis relatively sample to avoid over-fitting.

Regularization

- **Example**

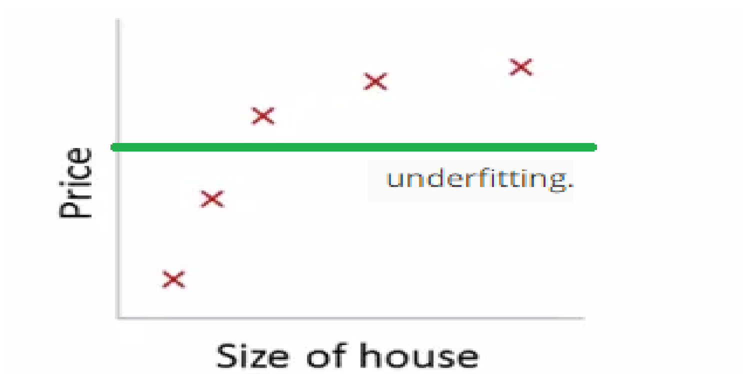


Regularization

- In regularization linear regression, we choose θ to minimize;

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{n}\theta_j^2\right]$$

- What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

$$\theta_1 = 0, \theta_2 = 0, \theta_3 = 0, \theta_4 = 0$$

$$= \theta_0 + \theta_1 (0) + \theta_2 (0)^2 + \theta_3 (0)^3 + \theta_4 (0)^4 = \theta_0$$

Question

In regularized linear regression, we choose θ to minimize:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

What if λ is set to an extremely large value (perhaps too large for our problem, say $\lambda = 10^{10}$)?

- Algorithm works fine; setting λ to be very large can't hurt it.
- Algorithm fails to eliminate overfitting.
- Algorithm results in underfitting (fails to fit even the training set).

Correct

- Gradient descent will fail to converge.

Regularization

$$Repeat$$

$$\{$$

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0) \qquad for\ (j = 0\ )$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J\left(\theta_j\right) \qquad for\ (j = 1,2,3 \ldots, n\ )$$

$$\} \qquad ------------(\boldsymbol{Equation - A})$$

Regularization

Calculation for $\frac{\partial}{\partial\theta_0}J(\theta_0)$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{\partial}{\partial\theta_0}\left[\frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{1}{2m}\frac{\partial}{\partial\theta_0}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

Regularization

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{1}{2m}\left[\frac{\partial}{\partial\theta_0}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2\right] + \frac{\partial}{\partial\theta_0}\left[\lambda\sum_{j=1}^{n}\theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{1}{2m}\left[\frac{\partial}{\partial\theta_0}\left[\sum_{i=1}^{m}(\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2\right] + \frac{\partial}{\partial\theta_0}\left[\lambda\sum_{j=1}^{n}\theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{2}{2m}\left[\left[\sum_{i=1}^{m}(\theta_0 + \theta_1 x^{(i)} - y^{(i)})\frac{\partial}{\partial\theta_0}(\theta_0 + \theta_1 x^{(i)} - y^{(i)})\right] + \frac{\partial}{\partial\theta_0}\left[\lambda\sum_{j=1}^{n}\theta_j^2\right]\right]$$

Regularization

$$\frac{\partial}{\partial\theta_0}J(\theta)$$

$$= \frac{1}{m}\left[\left[\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)\left\{\frac{\partial}{\partial\theta_0}(\theta_0) + \frac{\partial}{\partial\theta_0}(\theta_1 x^{(i)}) - \frac{\partial}{\partial\theta_0}(y^{(i)})\right\}\right]\right.$$

Regularization

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)\{1\}\right]$$

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)\left\{x_0^{(i)}\right\}\right]$$

Regularization

Calculation for $\frac{\partial}{\partial\theta_j} J(\theta_j)$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

$$\frac{\partial}{\partial\theta_j} J(\theta) = \frac{\partial}{\partial\theta_j}\left[\frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_j} J(\theta) = \frac{\partial}{\partial\theta_j}\left[\frac{1}{2m}\left[\sum_{i=1}^{m}(\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]\right]$$

Regularization

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{1}{2m}\left[\frac{\partial}{\partial\theta_j}\left[\sum_{i=1}^{m}(\theta_0+\theta_1x^{(i)}-y^{(i)})^2+\lambda\sum_{j=1}^{n}\theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{1}{2m}\left[\frac{\partial}{\partial\theta_j}\left[\sum_{i=1}^{m}(\theta_0+\theta_1x^{(i)}-y^{(i)})^2\right]+\frac{\partial}{\partial\theta_j}\left[\lambda\sum_{j=1}^{n}\theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{2}{2m}\left[\left[\sum_{i=1}^{m}\left(\theta_0+\theta_1x^{(i)}-y^{(i)}\right)[x_j^{(i)}]\right]+\lambda\theta_j\right]$$

$$\frac{\partial}{\partial\theta_j}J(\theta)=\frac{1}{m}\left[\sum_{i=1}^{m}\left(\theta_0+\theta_1x^{(i)}-y^{(i)}\right)[x_j^{(i)}]+\lambda\theta_j\right]$$

Regularization

By putting the values in equation (A), we have

Repeat

{

$$\theta_0 := \theta_0 - \frac{\alpha}{m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)\left\{x_0^{(i)}\right\}$$

$$\theta_j := \theta_j - \frac{\alpha}{m}\left[\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)[x_j^{(i)}] + \lambda\theta_j\right]$$

}

Logistic Regression Model

Regularization-Cost Function

Regularization

$$Repeat$$
$$\{$$
$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0) \qquad for\ (j = 0\ )$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_j) \qquad for\ (j = 1,2,3 \ldots, n\ )$$
$$\} \qquad ------------(\boldsymbol{Equation - A})$$

Regularization

Calculation for $\frac{\partial}{\partial\theta_0}J(\theta_0)$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(log(h_\theta(x^{(i)})\right) + \left(1 - \ y^{(i)}\right)\log(1-(h_\theta(x^{(i)})\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\frac{\partial}{\partial\theta_0}J(\theta_0) = \frac{\partial}{\partial\theta_0}\left[-\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(log(h_\theta(x^{(i)})\right) + \left(1 - \ y^{(i)}\right)\log(1-(h_\theta(x^{(i)})\right] + \frac{\partial}{\partial\theta_0}\left[\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2\right]\right]$$

$$\frac{\partial}{\partial\theta_0}J(\theta_0) = \frac{1}{m}\left[\sum_{i=1}^{m}\{y^{(i)}(1-h_\theta\left(x^{(i)}\right) - (1-y^{(i)})(h_\theta\left(x^{(i)}\right))\}\right]x^{(i)}$$

Regularization

Calculation for $\frac{\partial}{\partial\theta_j}J(\theta_j)$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(log(h_\theta(x^{(i)})\right) + \left(1 - \ y^{(i)}\right)\log(1 - (h_\theta(x^{(i)})\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\frac{\partial}{\partial\theta_j}J(\theta)$$
$$= \frac{\partial}{\partial\theta_j}\left[-\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(log(h_\theta(x^{(i)})\right) + \left(1 - \ y^{(i)}\right)\log(1 - (h_\theta(x^{(i)})\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2\right]$$

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

Regularization

- By putting the values in equation (A), we have

$$Repeat$$

$$\{$$

$$\theta_0 := \theta_0 - \frac{\alpha}{m}\left[\sum_{i=1}^{m}\{y^{(i)}(1 - h_\theta\left(x^{(i)}\right) - (1 - y^{(i)})(h_\theta\left(x^{(i)}\right))\}\right]x^{(i)}$$

$$\theta_{\ j} := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)} + \frac{\lambda\theta_j}{m}$$

$$\}$$

Reference

- https://www.coursera.org/