Data Science Advanced

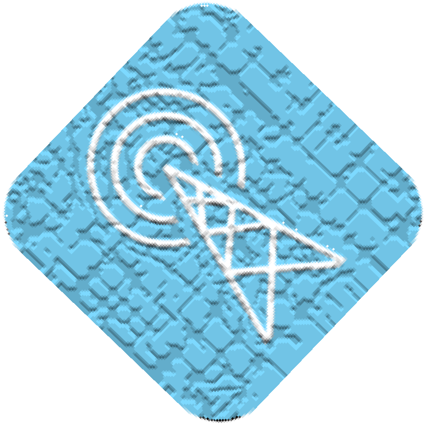
Lesson02–Regression, Logistic Regression using Gradient Descent

[Credit: Machine Learning Course, Andrew Ng

© Copyright 2015 All rights reserved.

Objective

After completing this lesson you will be able to:



Linear Regression:

- Describe hypothesis for a linear regression
- Understand cost function as a measure to derive regression equation.
- Understand gradient descent algorithm and its working to minimize the cost function.

Logistic Regression:

- Describe hypothesis for a logistic regression
- Understand cost function as a measure to derive logistic regression equation.
- Understand gradient descent algorithm and its working to minimize the cost function for a logistic regression

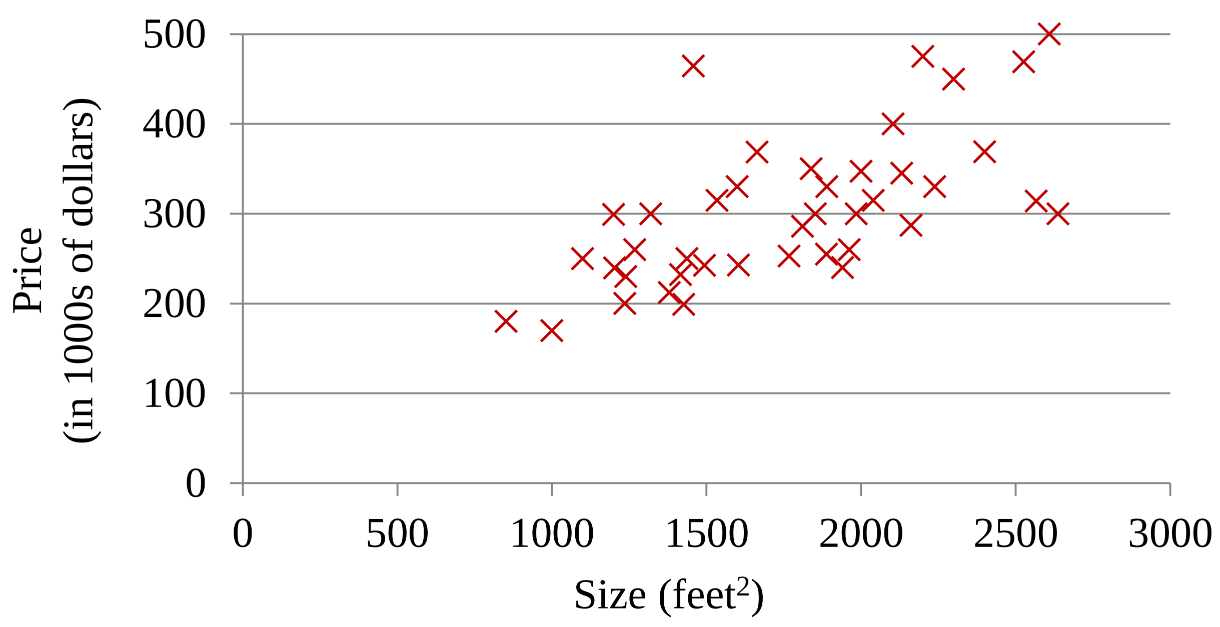
K Means Clustering:

- Understand cost function as a measure to derive K Means Cluster
- Understand Optimization objective for K means cluster

© Copyright 2015 All rights reserved.

Linear Regression

Linear Regression to predict housing prices for a given size.



- It is a supervised learning problem as the “right answer” for each example in given in the dataset.
- A regression based supervised learning to predict real valued output.

© Copyright 2015 All rights reserved.

Linear Regression–Data Representation

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

(x^i, y^i) $-$ *represents* i^{th} *training example*

$x^1 = 2104$

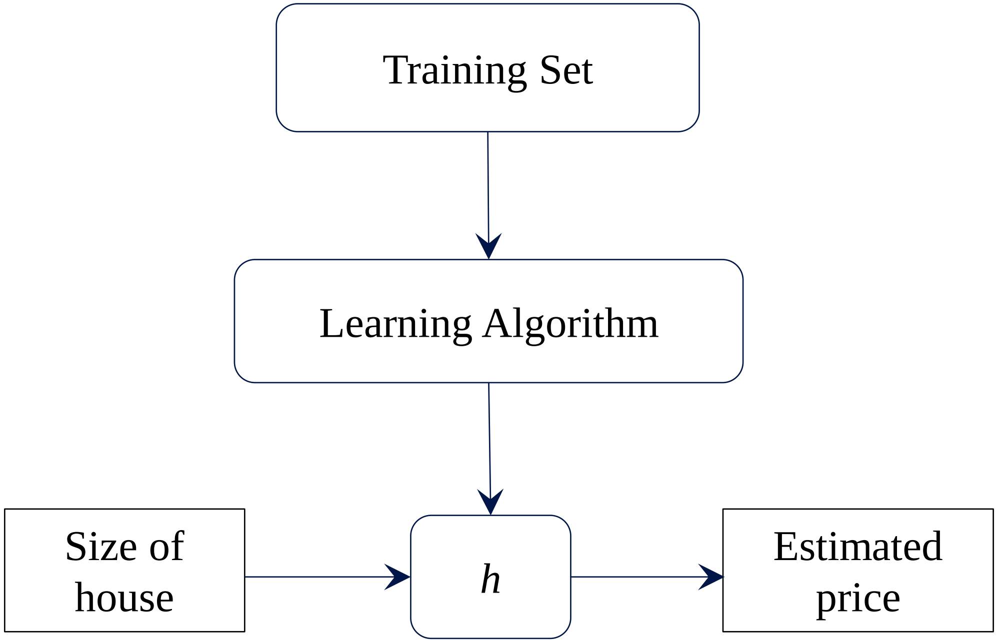
$y^1 = 460$

$(x^1, y^1) = (2104,460)$

Notions: **m** = Number of training examples
x's = "input" variable / features
y's = "output" variable / "target" variable

© Copyright 2015 All rights reserved.

Regression–Hypothesis Formulation



- In case of Linear Regression, the hypothesis function is:

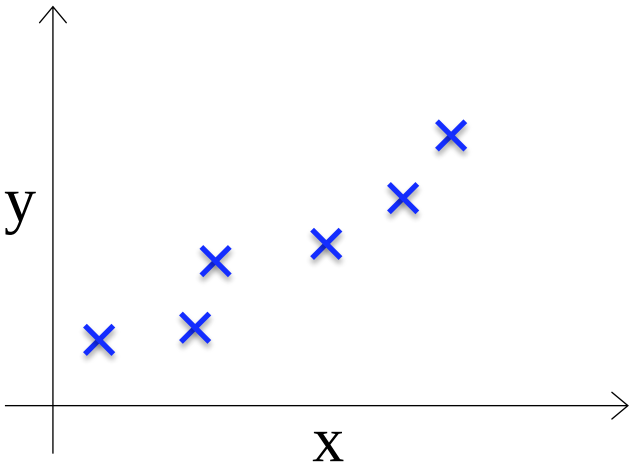
$$h_\theta(x) = \theta_0 + \theta_1 * x$$

- How to choose θs?

© Copyright 2015 All rights reserved.

Regression–Cost Function

Choose θ_0, θ_1 so that $h_\theta(x)$ is close to $\boldsymbol{y}$ for the training examples $(\boldsymbol{x}, \boldsymbol{y})$.



The cost function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^{(i)}\right)^2$$

Goal: $\min_{\theta_0, \theta_1} (J(\theta_0, \theta_1))$

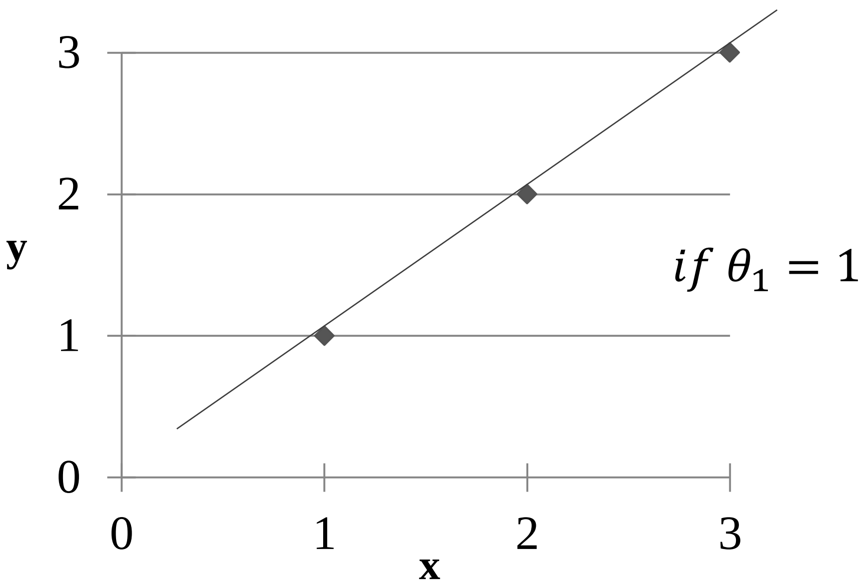
- Cost function $J(\theta_0, \theta_1)$ for regression is also called squared error function.
- The mean is halved (1/(2*m)) as a convenience for the computation of the gradient descent. The derivative term of the square function will cancel out the 1/2 term.

© Copyright 2015 All rights reserved.

Regression–Cost Function Intuition with one parameter

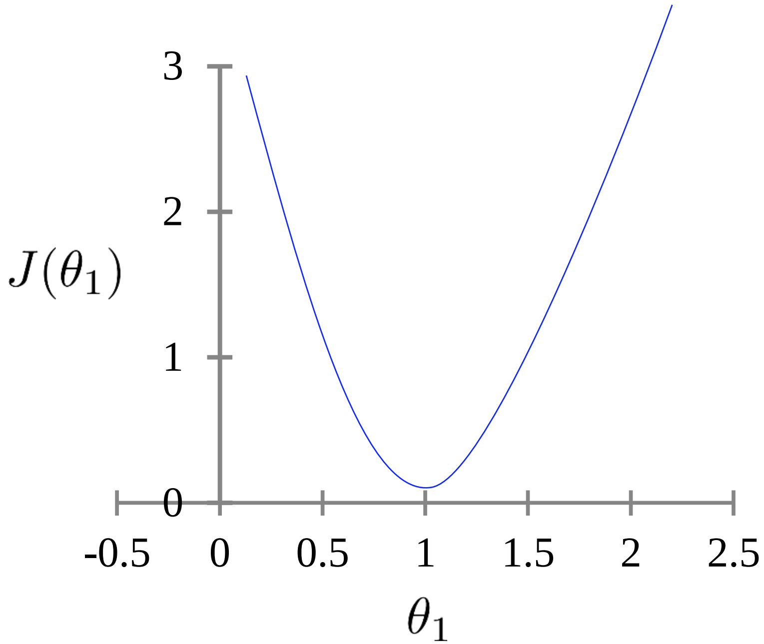
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

(function of the parameter θ_1)



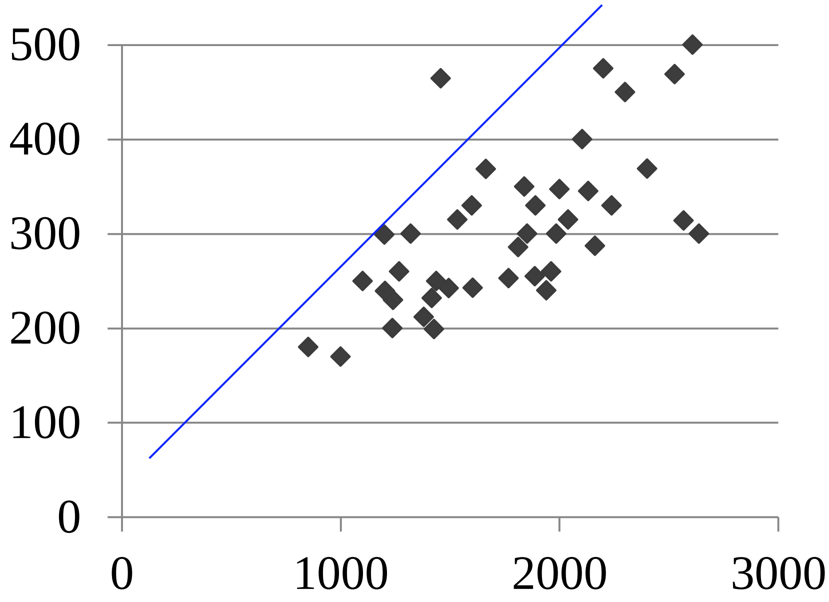
A simplified case where $\theta_0 = 0$. The cost function will be minimum for theta equal to zero. This will always be a convex shaped function.

© Copyright 2015 All rights reserved.

Regression–Cost Function Intuition with two parameter

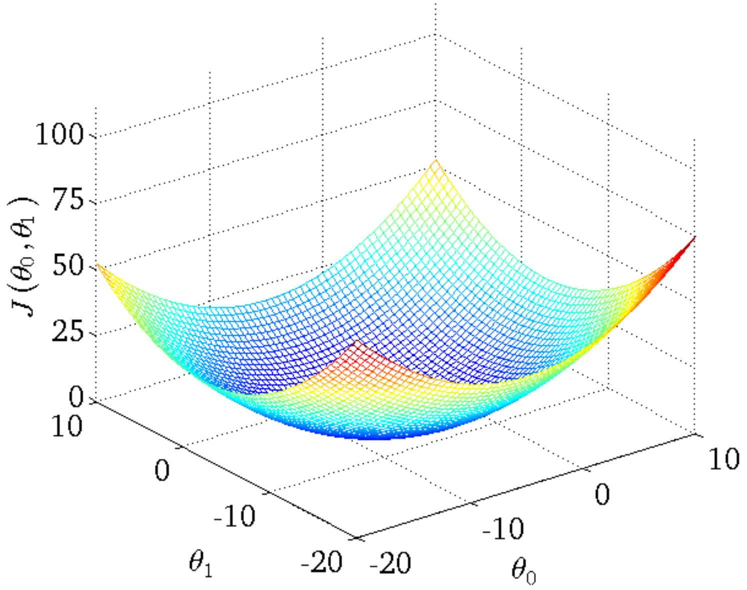
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)

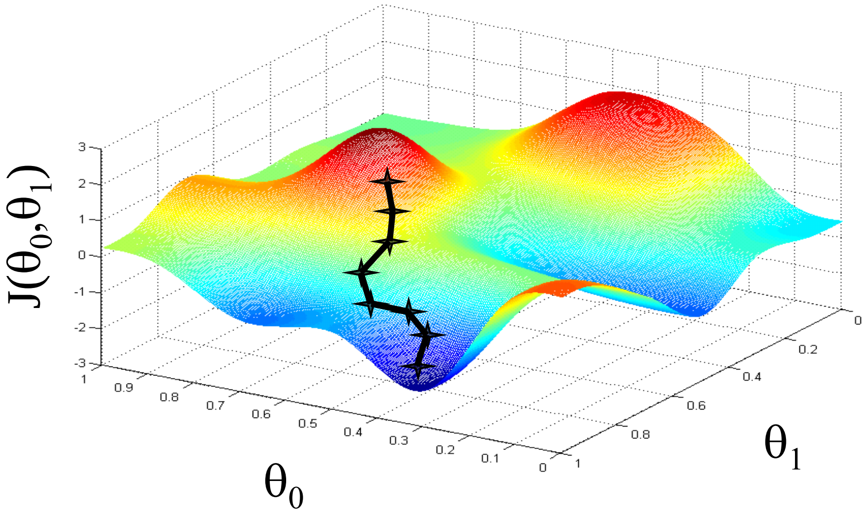


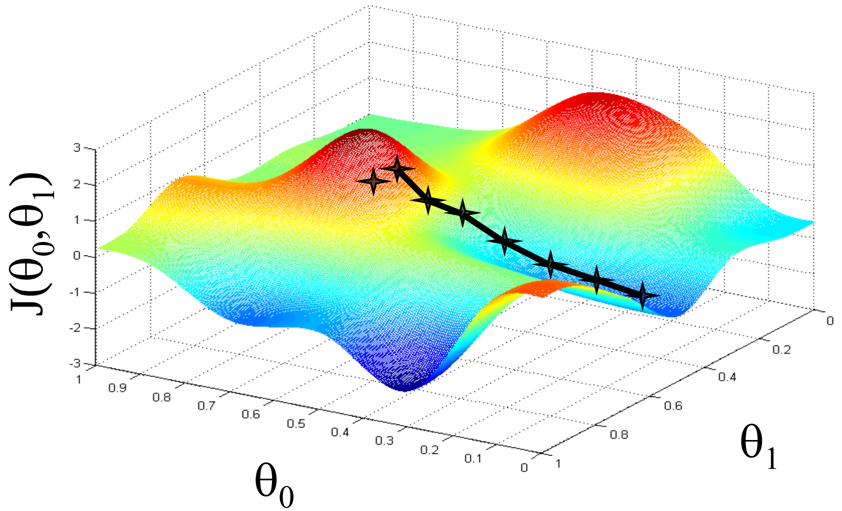
© Copyright 2015 All rights reserved.

Gradient Descent

Gradient Descent algorithm:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until a minimum is reached.





Gradient descent is a generic algorithm which can be used to minimize any type of cost function.

The initiation of θ_0, θ_1 can lead to a different local optima.

© Copyright 2015 All rights reserved.

Gradient Descent

- The gradient descent algorithm is:

$$
\begin{aligned}
&repeat \text{ until convergence}\{\\
&\quad \theta_j := \theta_j - \alpha * \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \;\; (for\ j = 0\ and\ j = 1)\\
&\}
\end{aligned}
$$

- Simultaneous update of θ_0, θ_1 is needed:

$$
\begin{gathered}
temp0 := \theta_0 - \alpha * \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)\\
temp1 := \theta_1 - \alpha * \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)\\
\theta_0 := temp0\\
\theta_1 := temp1
\end{gathered}
$$

- α is the learning rate which decides who big or small the steps of descent will be.
- Near to the local minimum, the gradient descent will automatically take smaller steps.
- At the local optima, the θ_0, θ_1 does not change as derivative term will equal to zero.

© Copyright 2015 All rights reserved.

Linear Regression–Gradient Descent

The gradient descent algorithm is:

$repeat$ until convergence{

$$\theta_j := \theta_j - \alpha * \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \;\; (for\ j = 0\ and\ j = 1)$$

}

The cost function:

$$h_\theta(x) = \theta_0 + \theta_1 * x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^{(i)}\right)^2$$

Goal: $\min_{\theta_0, \theta_1}\left(J(\theta_0, \theta_1)\right)$

The derivate term for linear regression will be:

$$(\theta_0)\ J = 0: \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^{(i)}\right)$$

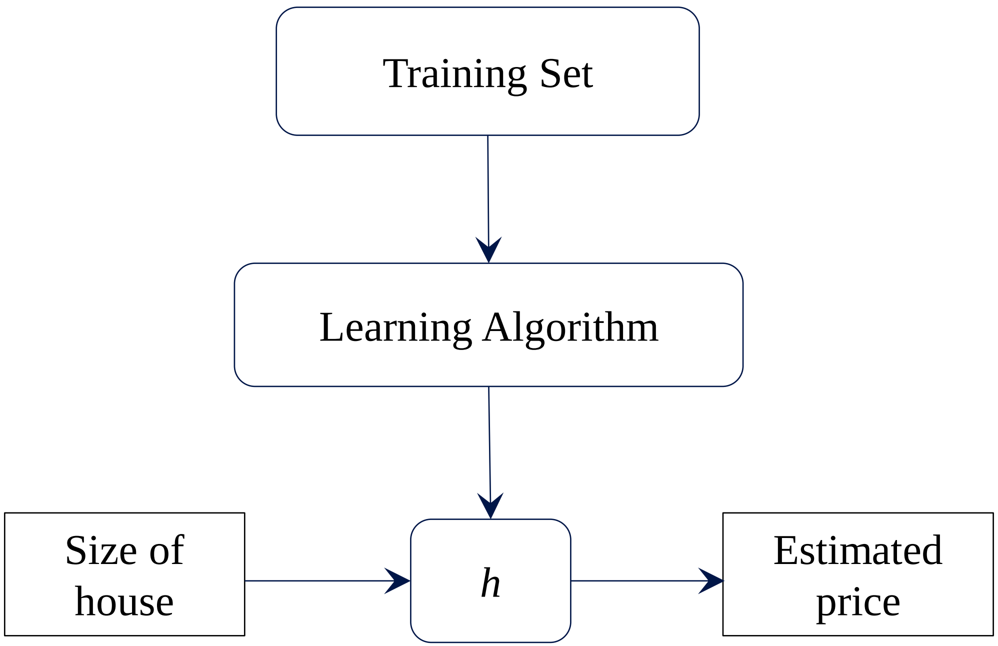
$$(\theta_1)\ J = 1: \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^{(i)}\right) * x^i$$

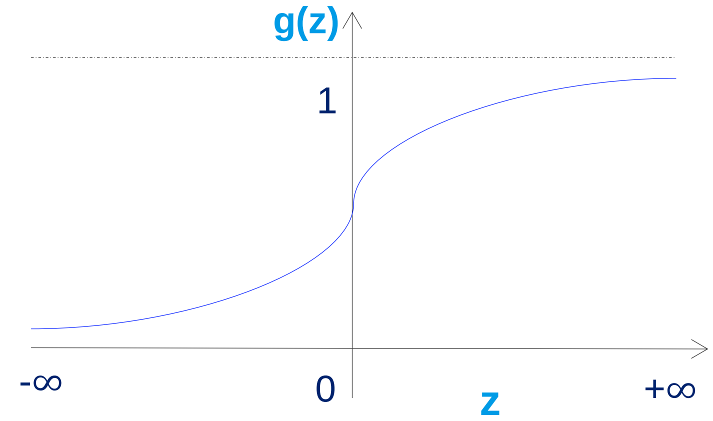
© Copyright 2015 All rights reserved.

Logistic Regression

© Copyright 2015 All rights reserved.

Logistic Regression–Hypothesis Formulation





In case of Logistic Regression, the hypothesis function is:

$$h_\theta(x) = g(\theta_0 + \theta_1 * x)$$

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

Where z = $\theta_0 + \theta_1 * x$

© Copyright 2015 All rights reserved.

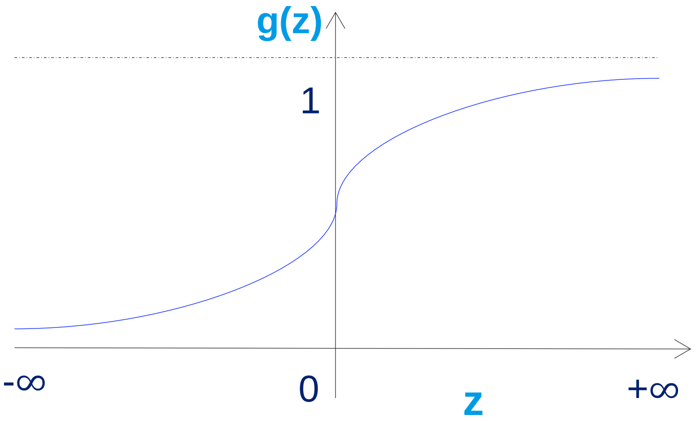
Logistic Regression–Decision Boundary

$P(y=1|x)$: $h_\theta(x) = g(\theta_0 + \theta_1 * x)$

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

Where z = $\theta_0 + \theta_1 * x$

- Predict y =1 if $h_\theta(x) \geq 0.5$
- Predict y =0 if $h_\theta(x) < 0.5$



$$g(z) \geq 0.5 \; whenever \; z \geq 0$$

Since

$$h_\theta(x) = g(\theta_0 + \theta_1 * x)$$

so

$$h_\theta(x) \geq 0.5 \; when (\theta_0 + \theta_1 * x) \geq 0$$

© Copyright 2015 All rights reserved.

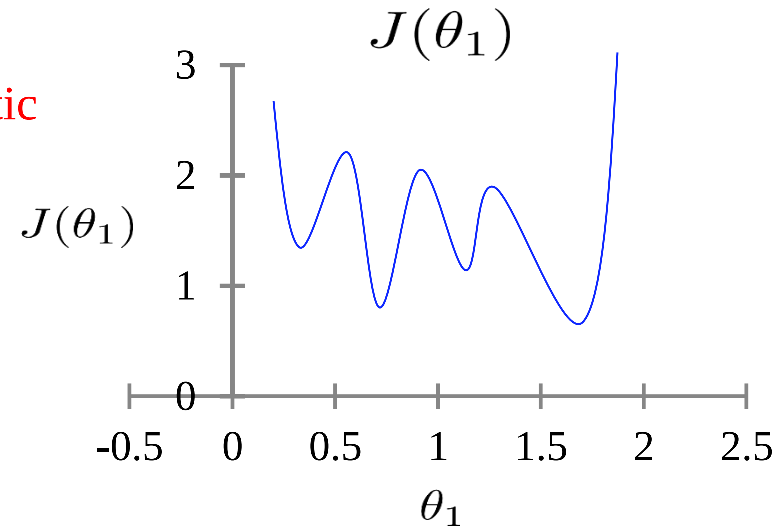
Logistic Regression – Cost Function

- Choose θ_0, θ_1 so that $h_\theta(x)$ is close to $\boldsymbol{y}$ for the training examples $(\boldsymbol{x}, \boldsymbol{y})$. Rewriting the linear regression cost function.

$$\boldsymbol{J(\theta)} = 1/m \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^i)) \quad where\ \boldsymbol{Cost(h_\theta(x^{(i)}), y^i)} = 1/2 \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^i)^2$$

- Cost depicts the penalty learning algorithm has to pay when it outputs $h_\theta\left(x^{(i)}\right)$ when actual label is y

- The above cost function cannot be used for logistic regression as it will be a non-convex function.

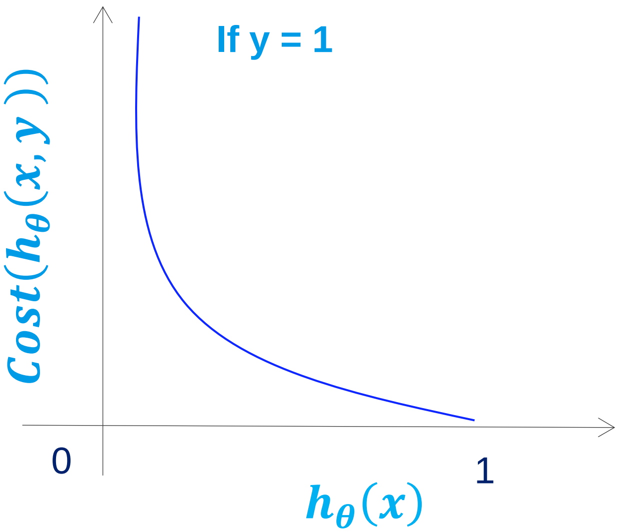


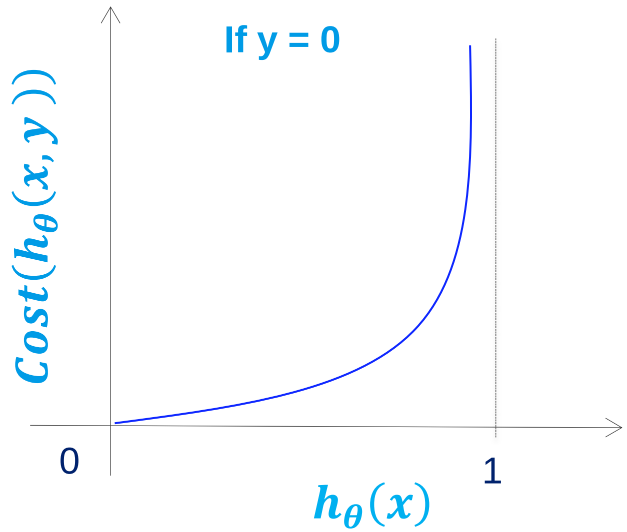
© Copyright 2015 All rights reserved.

Logistic Regression – Cost Function

- The penalty, learning algorithm has to pay when it outputs $h_\theta\left(x^{(i)}\right)$ when actual label is y is

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





© Copyright 2015 All rights reserved.

Logistic Regression – Simplified Cost Function

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

The simplified cost function is

The cost function:

$$J(\theta_0, \theta_1) = -\frac{1}{m}\sum_{i=1}^{m} y^i * \log(h_\theta(x^i)) + (1 - y^{(i)}) * \log(1 - h_\theta(x^i))$$

Goal: $\min_{\theta_0, \theta_1}(J(\theta_0, \theta_1))$

© Copyright 2015 All rights reserved.

Logistic Regression–Gradient Descent

- The gradient descent algorithm is:

$$repeat \text{ until convergence}\{$$

$$\theta_j := \theta_j - \alpha * \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \;\; (for\ j = 0\ and\ j = 1)$$

$$\}$$

- Simultaneous update of θ_0, θ_1 is needed:

$$temp0 := \theta_0 - \alpha * \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$temp1 := \theta_1 - \alpha * \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := temp0$$

$$\theta_1 := temp1$$

- α is the learning rate which decides who big or small the steps of descent will be.
- Near to the local minimum, the gradient descent will automatically take smaller steps.
- At the local optima, the θ_0, θ_1does not change as derivative term will equal to zero.

© Copyright 2015 All rights reserved.

Logistic Regression–Gradient Descent

The gradient descent algorithm is:

$repeat$ until convergence{

$$\theta_j := \theta_j - \alpha * \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$(for\ j = 0\ and\ j = 1)$$

}

The cost function:

$$J(\theta_0, \theta_1) = -\frac{1}{m} \sum_{i=1}^{m} \begin{matrix} y^i * \log(h_\theta(x^i)) \\ +(1 - y^{(i)}) * \log(1 - h_\theta(x^i)) \end{matrix}$$

Goal: $\min_{\theta_0, \theta_1} (J(\theta_0, \theta_1))$

The derivate term for logistic regression will be:

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

© Copyright 2015 All rights reserved.

K Means

© Copyright 2015 All rights reserved.

K Means Algorithm

Randomly initialize K Cluster Centroids $(\mu_1, \mu_2, \mu_3 \ldots . \mu_k)$

Training set is $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots . x^{(m)}\}$

Cluster assignment step

$Repeat \{ i = 1 \; to \; m$

$\mathrm{c}^{(\mathrm{i})} = \text{index from 1 to K of cluster centroid closet to } \mathrm{x}^{(\mathrm{i})}$

$\}$

Move Centroid

$Repeat \{ k = 1 \; to \; K$

$\mu_\mathrm{k} = \text{average of points assigned to cluster k}$

$\}$

© Copyright 2015 All rights reserved.

Cost function for K Means

$\mathrm{c}^{(\mathrm{i})} =$ index of cluster (1, 2 , ... K) to which $\mathrm{x}^{(\mathrm{i})}$ *is currently assigned*

$\mu_k = cluster\ centroid\ k\ where\ k = \{1, 2, 3, \ldots . K\}$

$\mu_{c^{(i)}} = cluster\ centroid\ of\ the\ cluster\ to\ which\ example\ x^{(i)}\ has\ been\ assigned$

Optimization objective

$$J\left(\mathrm{c}^{(1)}, \ldots . \mathrm{c}^{(\mathrm{m})}, \mu_1, \ldots . \mu_k\right) = 1/\mathrm{m} \sum_{1}^{m} \left\| x^i - \mu_{c^{(i)}} \right\|^2$$

$$\frac{minimize}{\substack{\mathrm{c}^{(1)}, \ldots . \mathrm{c}^{(\mathrm{m})} \\ \mu_1, \ldots . \mu_k}} J\left(\mathrm{c}^{(1)}, \ldots . \mathrm{c}^{(\mathrm{m})}, \mu_1, \ldots . \mu_k\right)$$

© Copyright 2015 All rights reserved.

Random initialization of K clusters

Pick k random training examples and set $\mu_1, \ldots . \mu_k$ equal to these k training examples

for{ i = 1 to 100

Randomly initialize K means

Run K means to get cluster assignment and centroid movement

Compute cost function

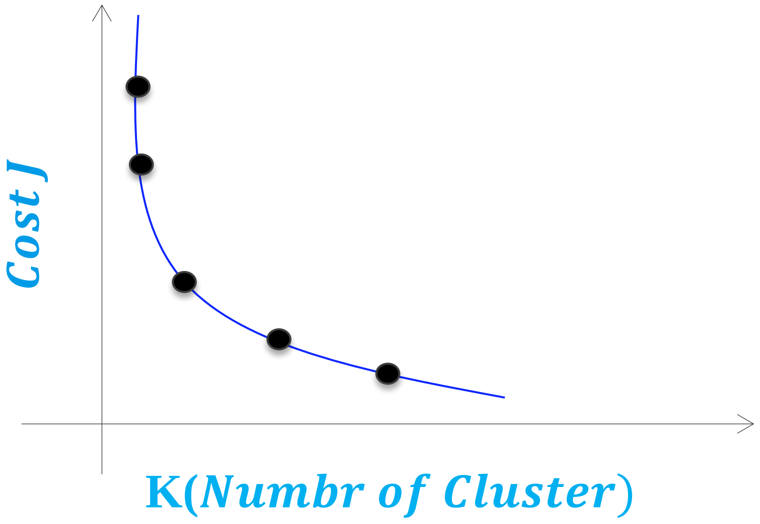
}

Pick the cluster that gives the lowest cost $J\left(c^{(1)}, \ldots . c^{(m)}, \mu_1, \ldots . \mu_k\right)$

© Copyright 2015 All rights reserved.

Optimal Number of Cluster

Elbow Method: Look for the number of cluster after which the delta change in cost function is not significant



Pick the cluster that gives the lowest cost $J\left(\mathrm{c}^{(1)}, \dots .\mathrm{c}^{(\mathrm{m})}, \mu_1, \dots .\mu_k\right)$

© Copyright 2015 All rights reserved.

End of Lesson02–Regression, Logistic Regression using Gradient Descent

© Copyright 2015 All rights reserved.