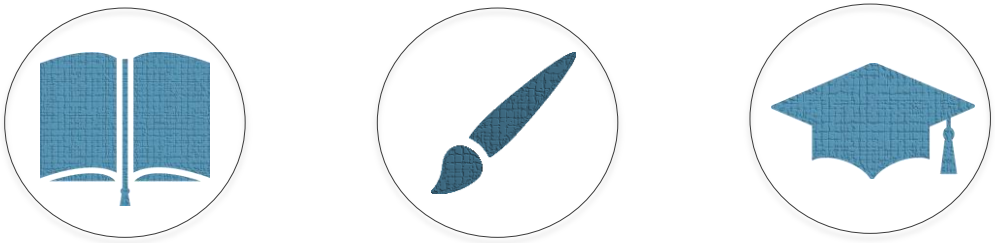


Data Science Advanced

Lesson02–Regression using Gradient Descent

Objective

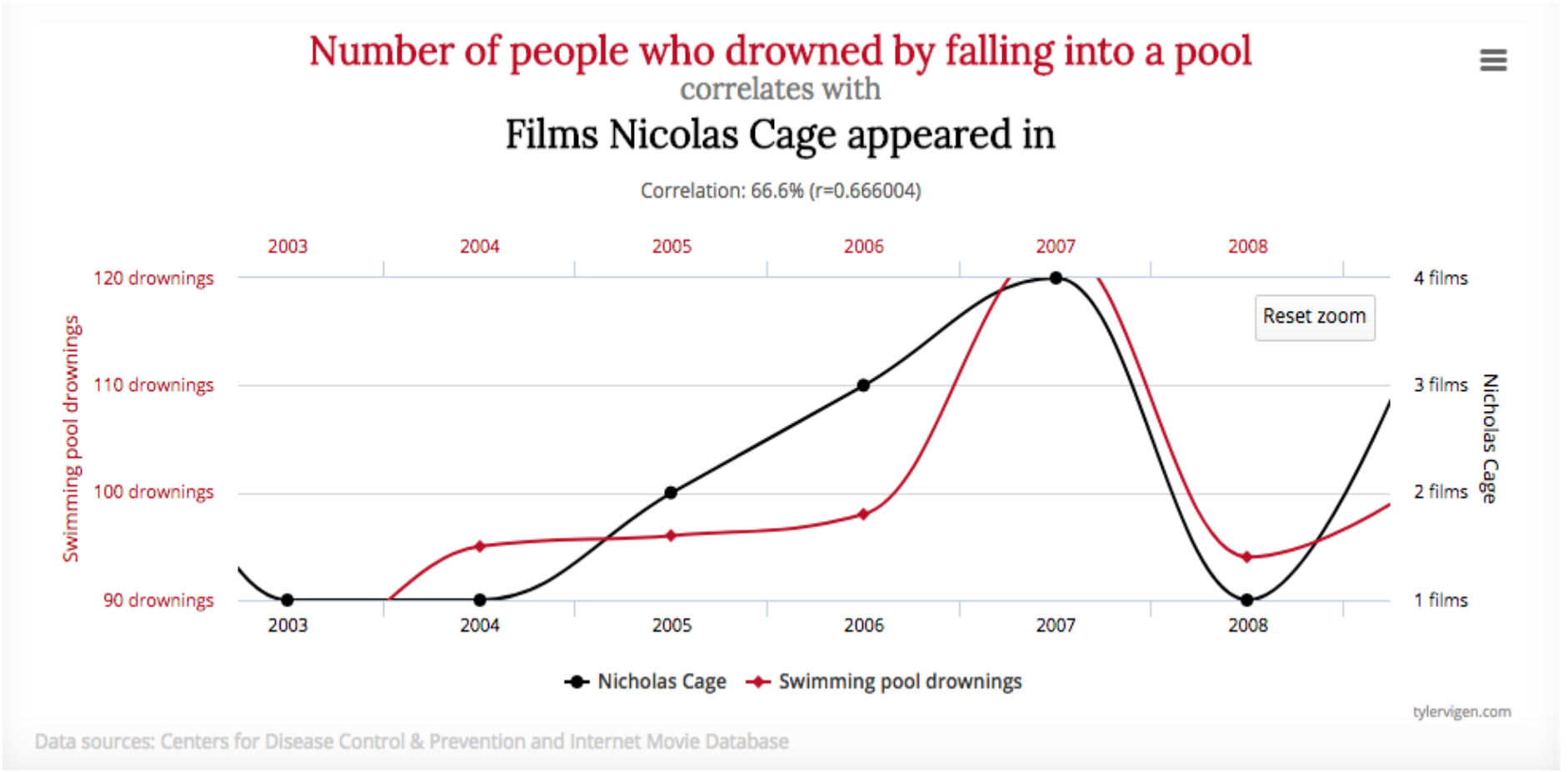
After completing this lesson you will be able to:

Linear Regression:

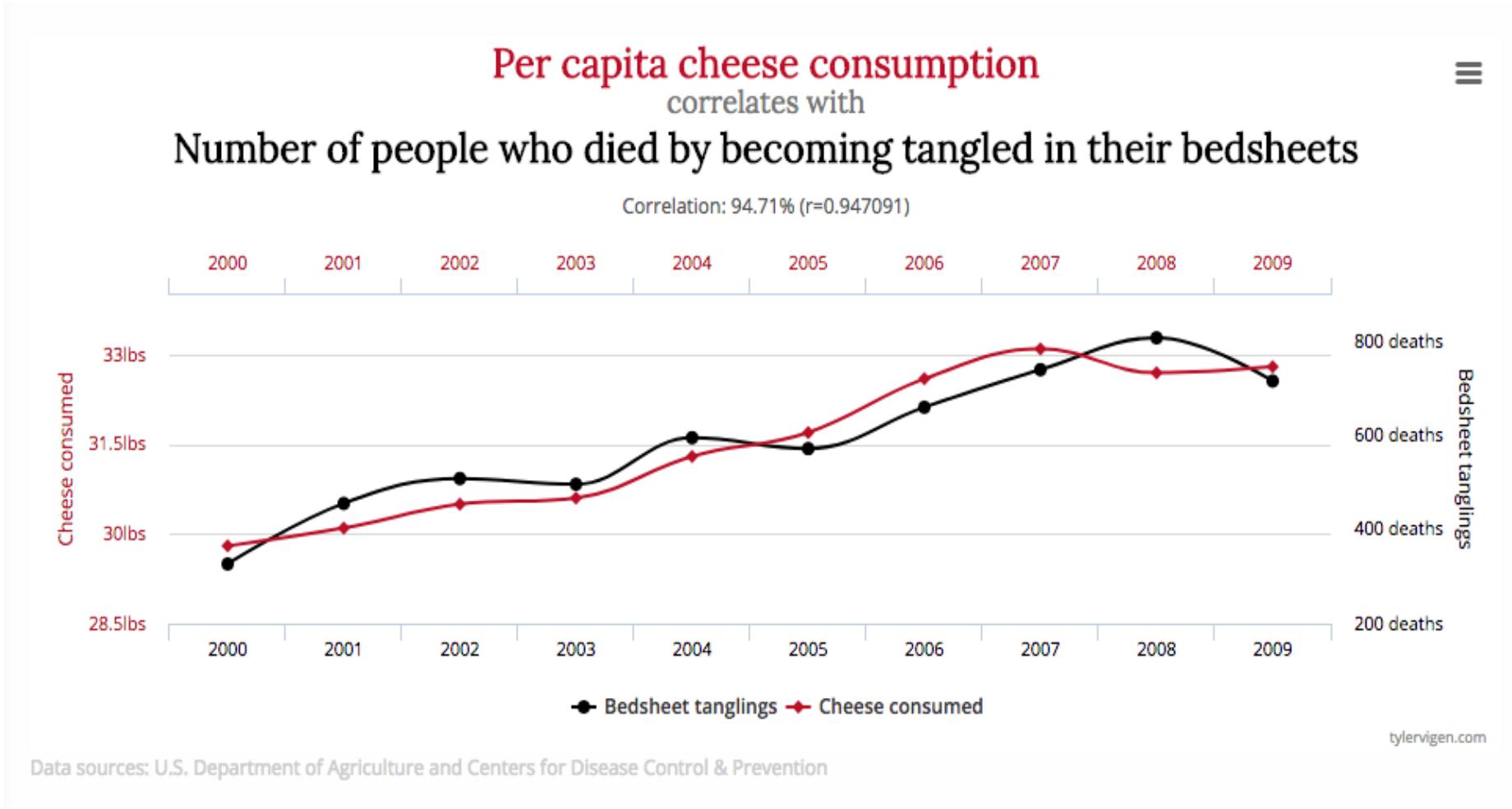
- Describe hypothesis for a linear regression
- Understand cost function as a measure to derive regression equation.
- Understand gradient descent algorithm and its working to minimize the cost function.
- Understand Bias and Variance Concept
- Apply Regularization for Model generalization

Married men earn more money

Which is a dependent variable and which one an independent variable?



Source : http://www.tylervigen.com/spurious-correlations



Source : http://www.tylervigen.com/spurious-correlations

CRIME A BLOG ABOUT MURDER, THEFT, AND OTHER WICKEDNESS. | JULY 9 2013 2:59 PM

When Ice Cream Sales Rise, So Do Homicides. Coincidence, or Will Your Next Cone Murder You?

By Justin Peters

*Crime is **Slate**'s crime blog. Like us on **Facebook**, and follow us on Twitter **@slatecrime**.*

JUSTIN PETERS

97% Correlation between these two. But correlation is not causation!!!

"The England tour has been an eye-opener for everyone. From whatever information we have gathered, it's been seen that even if players wanted to focus on their cricket, their wives were being a big distraction.

- Indian official, after test series loss to England 2014

The essence of regression analysis is to be use the information available about surrounding (independent variables) to better predict an outcome (dependent variable).

Regression–Building the Concept

	Weekly family income X (Rs.)									
X	**800**	**1000**	**1200**	**1400**	**1600**	**1800**	**2000**	**2200**	**2400**	**2600**
Weekly expenditure (Rs.) Y	550	650	790	800	1020	1100	1200	1350	1370	1500
	600	700	840	930	1070	1150	1360	1370	1450	1520
	650	740	900	950	1100	1200	1400	1400	1550	1750
	700	800	940	1030	1160	1300	1450	1520	1650	1780
	750	850	980	1080	1180	1350	-	1570	1750	1800
	-	880	-	1130	1250	1400	-	1600	1890	1850
	-		-	1150	-	-	-	1620	-	1910
Total	3250	4620	4450	7070	6780	7500	6850	10430	9660	12110
E(Y|X)	650	770	890	1010	1130	1250	1370	1490	1610	1730

- The unconditional mean i.e. E(Y) = 72720/60 = 1212.
- The essence of regression analysis is to be use the knowledge of income level to better predict the weekly expenditure.

Regression–Population Regression Function

E(Y|X) is called the population regression function and tells how the mean response of Y varies with X.

The first assumption of PRF is a linear function of X:

$$E(Y|X_i) = \theta_0 + \theta_1 * X_i$$

- θ_0 is the estimated average value of Y when the value of X is zero. More often than not it does not have a physical interpretation
- θ_1 is the estimated change in the average value of Y as a result of a one-unit change in X.

Linearity for regression assumes linearity in beta values and not in X variables. Example of non linear form: Y = β0 + 1/ (β1 + β2X1) + X2β3 + ε.

Regression–Sample Regression Function

Generally the information available will be a randomly selected sample of Y values for fixed X values.

Y (Exp)	X (Inc)
700	800
650	1000
900	1200
950	1400
1100	1600
1150	1800
1200	2000
1400	2200
1550	2400
1500	2600

Y (Exp)	X (Inc)
550	800
880	1000
900	1200
800	1400
1180	1600
1200	1800
1450	2000
1350	2200
1450	2400
1750	2600

Sample regression function (SRF) takes the form:

$$\widehat{Y_i} = \widehat{\theta_0} + \widehat{\theta_1} * \widehat{X_i}$$

where

- $\widehat{Y_i}$ = estimator of $E(Y|X_i)$
- $\widehat{\theta_0}$= estimator of θ_0
- $\widehat{\theta_1}$= estimator of θ_1

Regression–Sample Regression Function

Method of ordinary least squared (OLS) is used to choose SRF in such a way that $\sum \hat{u}_i^2 = \sum (Y_i - \widehat{Y_i})^2$ is minimized.

The equation obtained

$$\widehat{Y_i} = \widehat{\theta_0} + \widehat{\theta_1} * \hat{X}_i$$

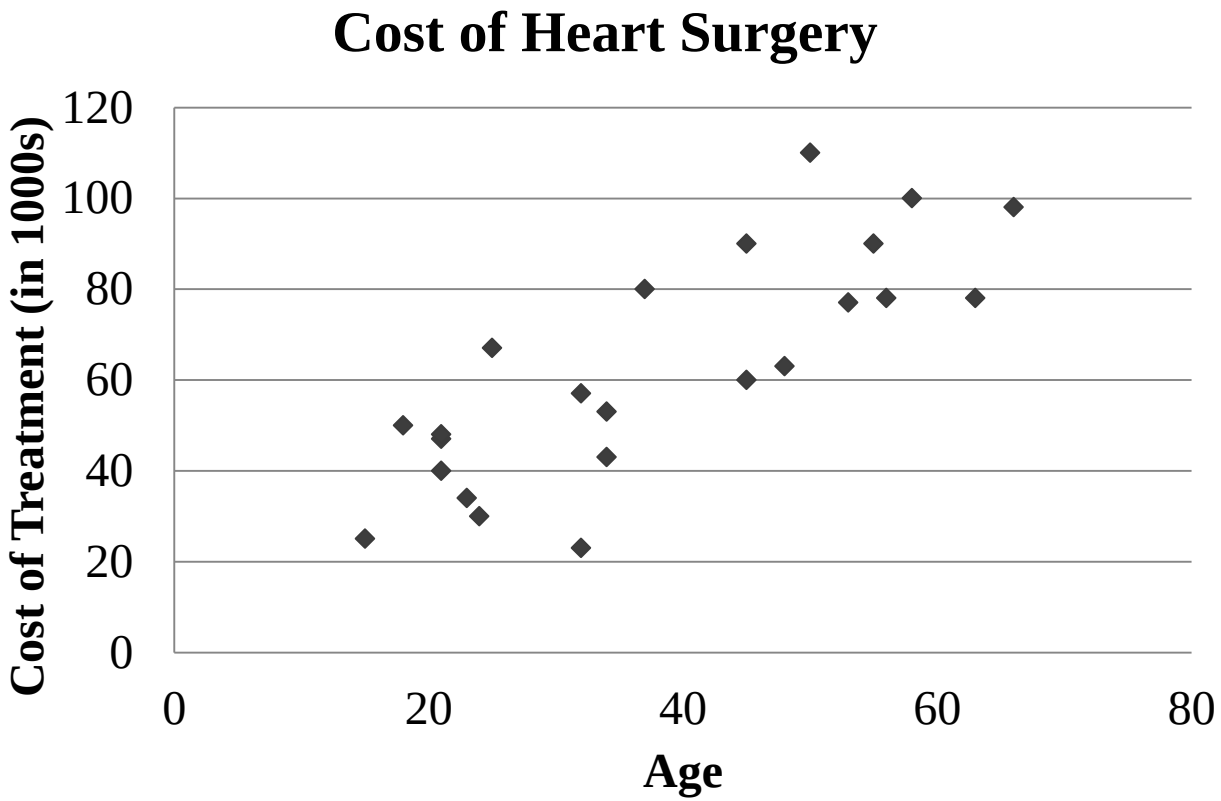
will have following properties:

- The sum of the squared residuals is a minimum.
- The sum of the residuals from the least squares regression line is 0.
- The simple regression line always passes through the sample mean of the Y and X variable.

Objective is to not only estimate $\widehat{\theta_0}$ and $\widehat{\theta_1}$ but also ensure it is close as possible to the true θ_0 and θ_1 (termed as Generalizability)

Linear Regression

Linear Regression to predict cost of treatment for a given age.



- 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.

Linear Regression–Data Representation

Age (x)	Cost of Treatment (y)
15	25
18	50
37	80
21	40
58	100
24	30
25	67

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

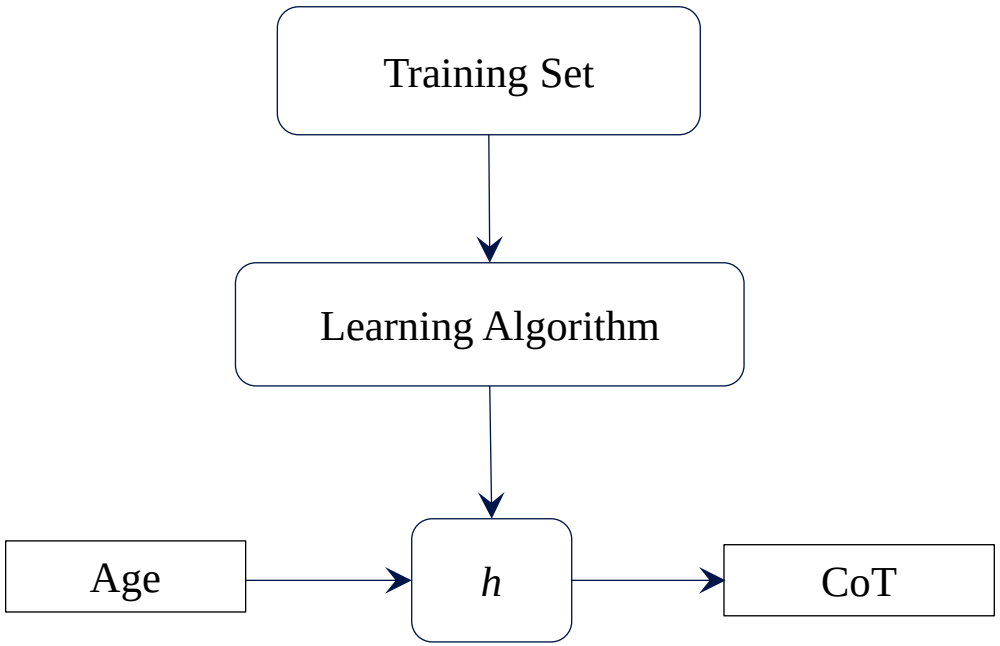
$x^1 = 15$

$y^1 = 25$

$(x^1, y^1) = (15,25)$

Notions: **m** = Number of training examples (7 records in the above table)
x's = "input" variable / features; **y**'s = "output" variable / "target" variable
n = Number of features

Regression–Hypothesis Formulation



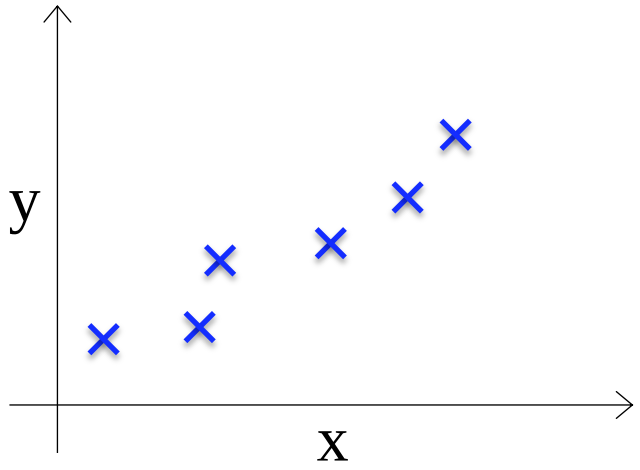
- In case of Linear Regression, the hypothesis function is:

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

- How to choose θs?

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:

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

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.

Regression–Cost Function Intuition with one parameter

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

Parameters: θ_0, θ_1

Cost Function:

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

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Simplified

$$h_\theta(x) = \theta_1 x$$

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

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

Regression–Cost Function Intuition with one parameter

Actual

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

Parameters: θ_0, θ_1

Cost Function:

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

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

Simplified

Hypothesis: $h_\theta(x) = \theta_1 * x$

Parameters: θ_1

Cost Function:

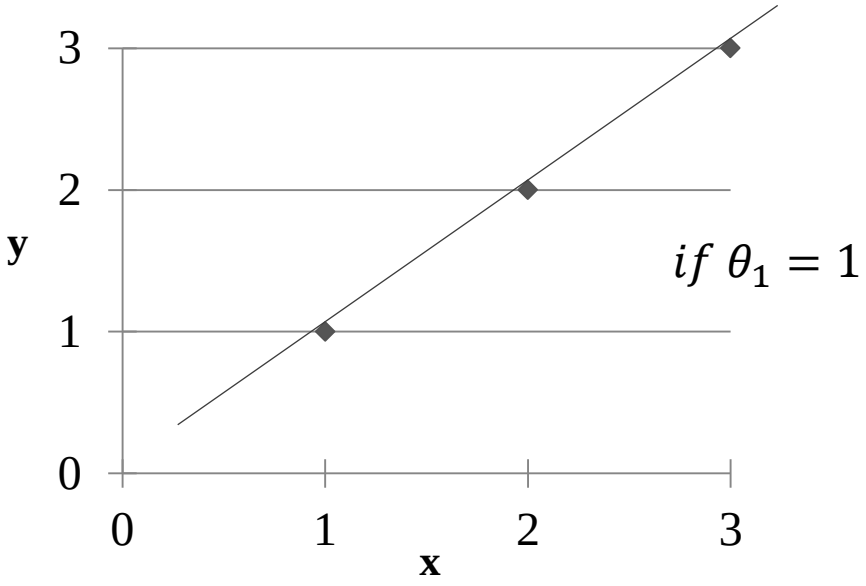
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^{(i)}\right)^2$$

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

Regression–Cost Function Intuition with one parameter

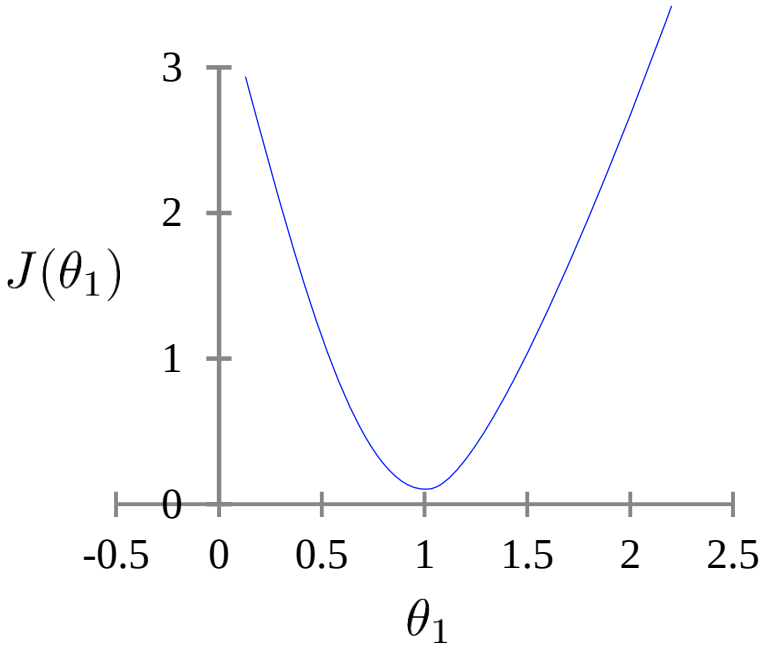
$h_\theta(x)$

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



$J(\theta_1)$

(function of the parameter θ_1)

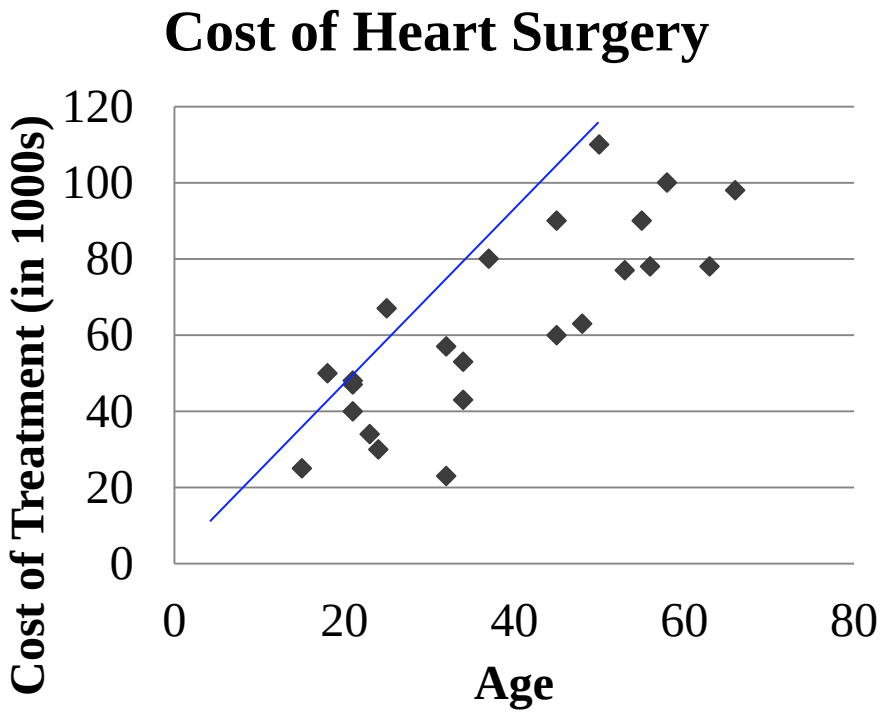


A simplified case where $\theta_0 = 0$. The cost function will be minimum for theta equal to zero. This will always be a convex shaped function.

Regression–Cost Function Intuition with two parameter

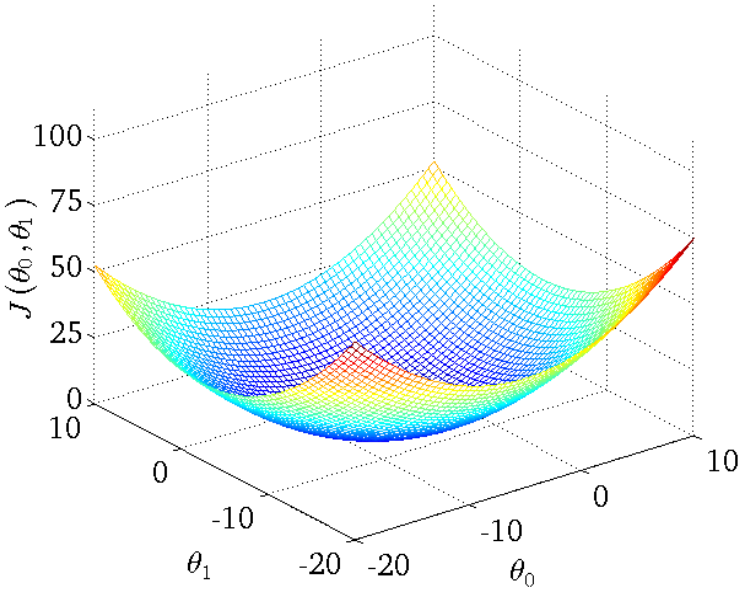
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

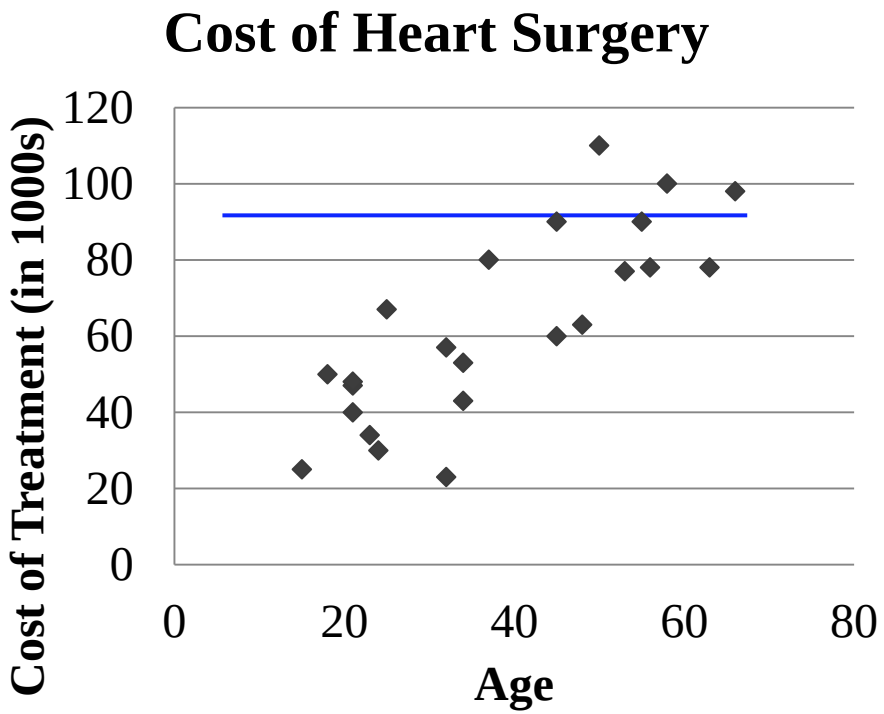
(function of the parameters θ_0, θ_1)



Regression–Cost Function Intuition with two parameter

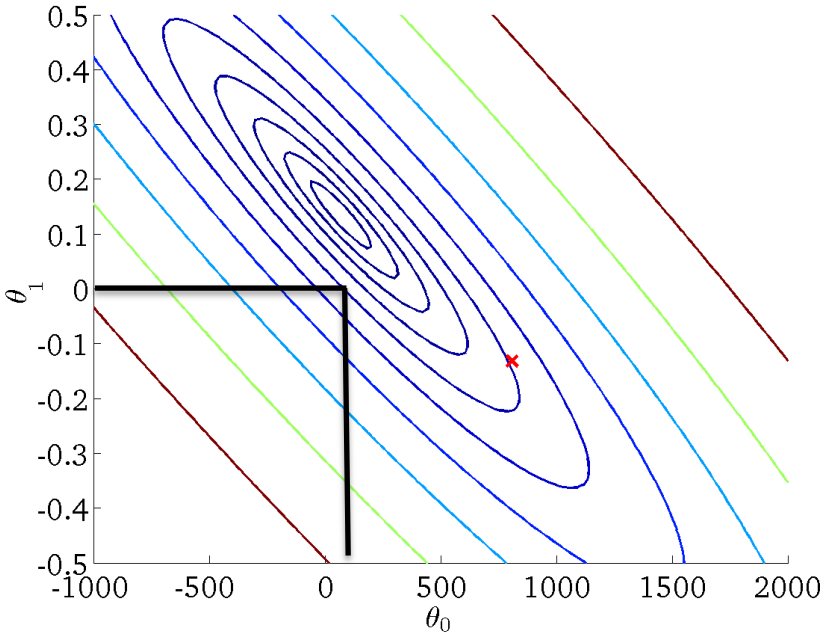
$h_\theta(x)$

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



$J(\theta_0, \theta_1)$

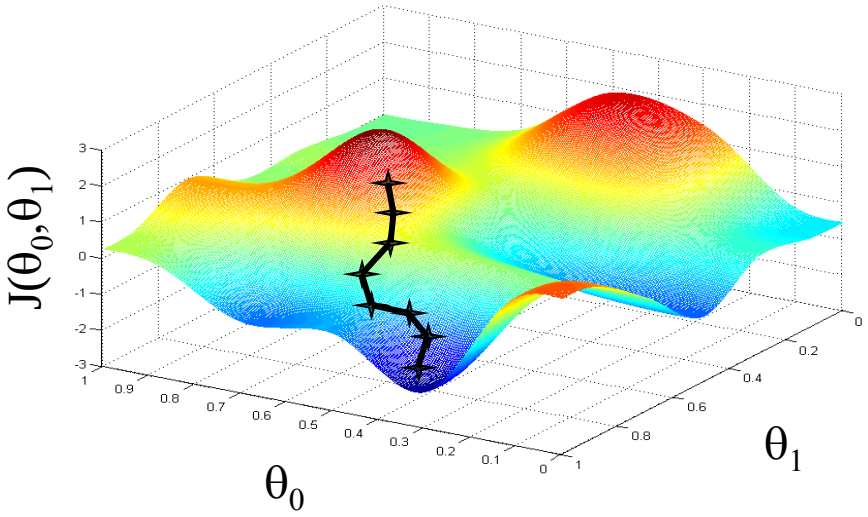
(function of the parameters θ_0, θ_1)

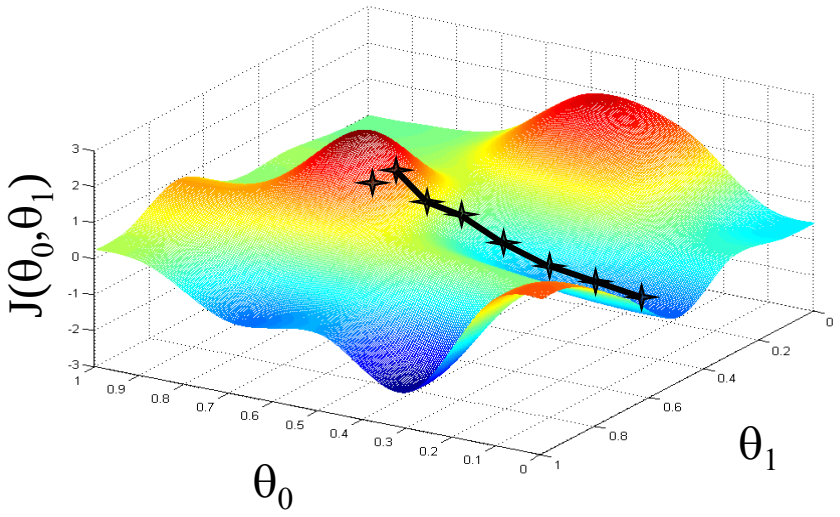


Gradient Descent

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.

Gradient Descent

- The gradient descent algorithm is:

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

- 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_1 does not change as derivative term will equal to zero.

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:

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

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

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

The derivate term for linear regression will be:

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

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

Linear Regression–Gradient Descent

$repeat$ until convergence{

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

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

}

Since for θ_0, x^i feature will be 1 so the gradient descent is written separate. The above can also be written as: $\theta_j := \theta_j - \alpha * \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^i\right) - y^{(i)}\right) * x^i$

Bias Versus Variance

The problem of high bias comes up when the model is miss-specified (simple hypothesis is selected).

The problem of high variance comes up when the model fails to generalize.

If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^{(i)}\right)^2 \approx 0$), but fail to generalize to new examples (predict the cost of treatment for new patients).

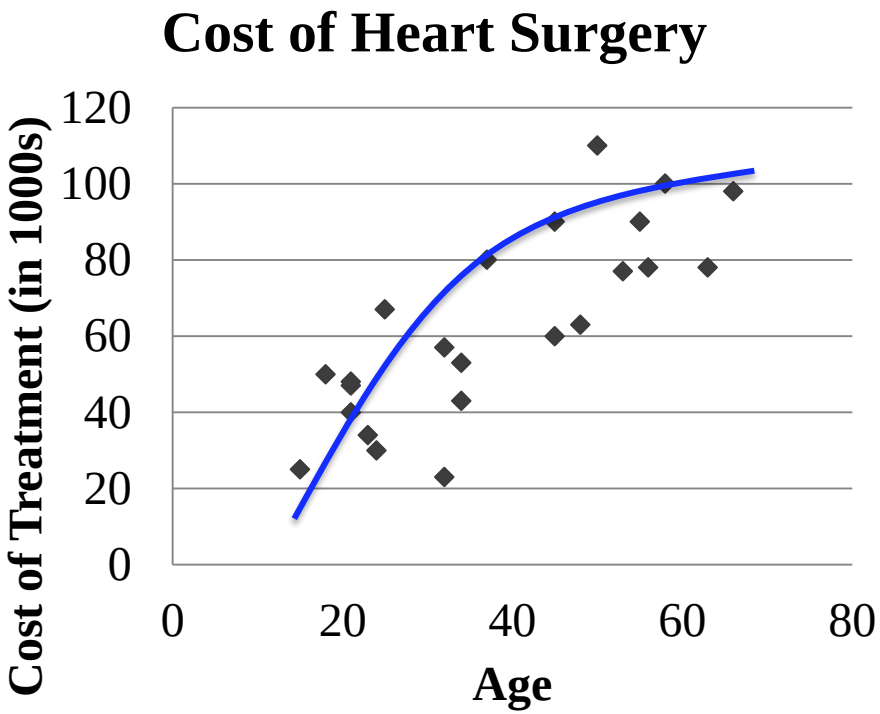
Reduce Overfitting

1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm (discussed later).

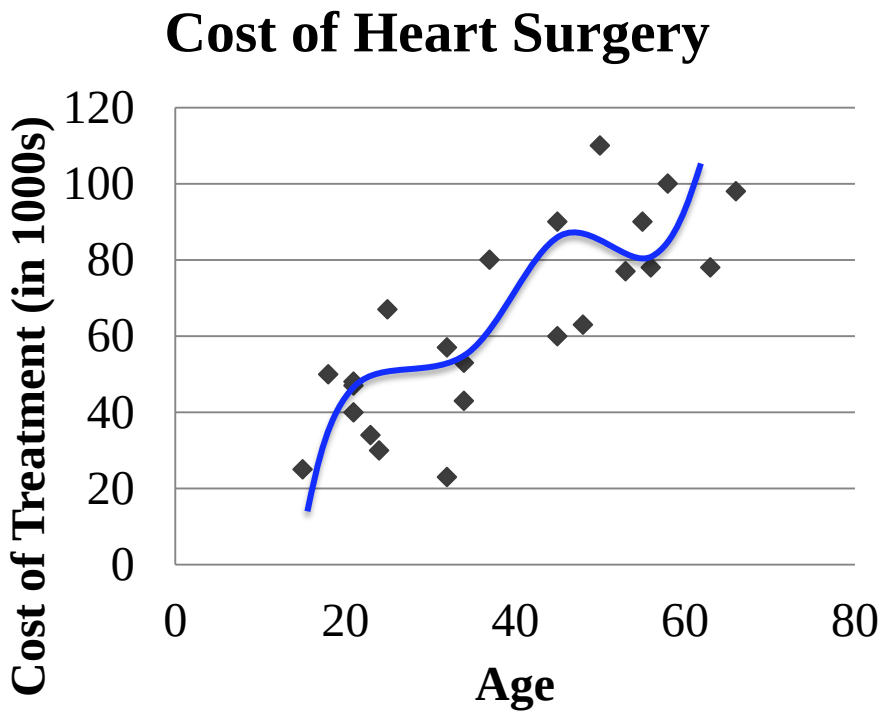
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters .
   - Works well when we have a lot of features, each of which contributes a bit to predicting .

Regularization – Reduce overfitting

Penalize θ_3 and θ_4 to make them really small.



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



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

Modify the cost function to add high weights to θ_3 and θ_4.
The cost function can be minimized only if theθ_3 and θ_4 ≈ 0.

Regularization – Reduce overfitting

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^{(i)}\right)^2 + 1000 * {\theta_3}^2 + 1000 * {\theta_4}^2$$

- Since we do not know which feature to penalize. Small values for parameters $\theta_0, \theta_1, \theta_2, \theta_3, \ldots$.(Penalize all)

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

"Simpler" hypothesis and Less prone to overfitting

The cost function, where the penalty added is sum of θ − squared is know as **Ridge Regression (L2)**.

In case, the penalty added is absolute value of θ then it is **Lasso Regression (L1).**

Regularization – Reduce overfitting

Ridge versus Lasso Regression

Ridge	Lasso
• shrink coefficients towards zero, it can never reduce it to zero. • all features will be included in the model no matter how small the value of the coefficients.	• able to shrink coefficient to exactly zero. • reduces the number of features and serve as a feature selection tools at the same time.

Lasso regression useful in cases with high dimension and helps with model interpretability.

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:

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

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

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

The derivate term for linear regression will be:

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

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial \theta_1}}\boldsymbol{J(\theta_0, \theta_1)} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^{(i)}\right) * x^i + \frac{\lambda}{\mathrm{m}} * \sum_{j=1}^{n}\theta_j$$

Linear Regression–Regularized Gradient Descent

repeat until convergence{

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

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

}

The regularization is for all the features excluding X_0 and thus gradient descent for θ_0 is separate. Note the summation for regularization starts from index 1.

Quiz

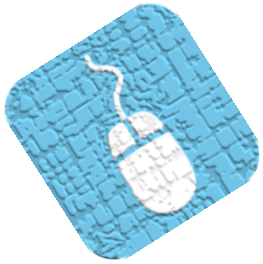
Quiz No:1	What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

a. Algorithm works fine; setting λ to be very large can't hurt it

b. Algorithm fails to eliminate overfitting.

c. Algorithm results in under fitting. (Fails to fit even training data well).

d. Gradient descent will fail to converge.



Use Regression on DAD Hospital Case

End of Lesson02–Regression using Gradient Descent