Practical Machine Learning

Module 2: Linear Regression with Multiple Variables

intelligent layer

asi

Content and objectives

Content

- Linear Model with Multiple Variables
- Feature Scaling
- Debugging Gradient Descent

Objectives

- Extend linear model to handle multiple features

Content and objectives

Content

- Linear Model with Multiple Variables
- Feature Scaling
- Debugging Gradient Descent

Objectives

- Understand why feature scaling is needed
- Understand how to scale features effectively

Content and objectives

Content

- Linear Model with Multiple Variables
- Feature Scaling
- Debugging Gradient Descent

Objectives

- Understand how to check convergence of Gradient Descent
- Understand how to fix problems

Linear Regression with multiple variables

Linear Model with one variable

Target only a function a single FEATURE

$$f_\theta(x) = \theta_0 + \theta_1 x$$

Size in ft^2 **(x)**	Rent / month (£) **(y)**
3121	450
1746	415
2443	398
...	...

Linear Model with multiple variables

Target a function of several FEATURES

$$f_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots$$

Size in ft^2 **(x_1)**	No. of bedrooms **(x_2)**	No. of floors **(x_3)**	Age of house **(x_4)**	Rent / month (£) **(y)**
3121	4	2	45	450
1746	2	1	24	415
2443	3	3	36	398
…				…

Linear Model with multiple variables

Size in ft^2 (x_1)	No. of bedrooms (x_2)	No. of floors (x_3)	Age of house (x_4)	Rent / month (£) (y)
3121	4	2	45	450
1746	2	1	24	415
2443	3	3	36	398
…				…

Notation

$\mathbf{n}$ = No. of features

$\mathbf{x}^{(i)}$ = features of i^{th} example

$\mathbf{x}_j^{(i)}$ = j^{th} feature in i^{th} example

Applying Gradient Descent

Function: $f_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots$

Parameters: $\theta_0, \theta_1, \theta_2, \cdots$

Cost Function: $C(\theta_0, \theta_1, \cdots) = \frac{1}{2m} \sum_{i=1}^{m} (f_\theta(\mathbf{x}^i) - y^i)^2$

Gradient Descent:

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\qquad \theta_j := \theta_j - \alpha \frac{\partial C}{\partial \theta_j} \qquad \text{for } j = 0, 1, 2, \cdots \\
&\}
\end{aligned}
$$

Applying Gradient Descent

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial C}{\partial \theta_j}$$

}

for $j = 0, 1, 2, \cdots$

Applying Gradient Descent

repeat until convergence {

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

}

for $j = 0, 1, 2, \cdots$

Applying Gradient Descent

repeat until convergence {

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

}

for $j = 0, 1, 2, \cdots$

For θ_0 note that $x_0^i = 1$ always

$$f_\theta(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots$$

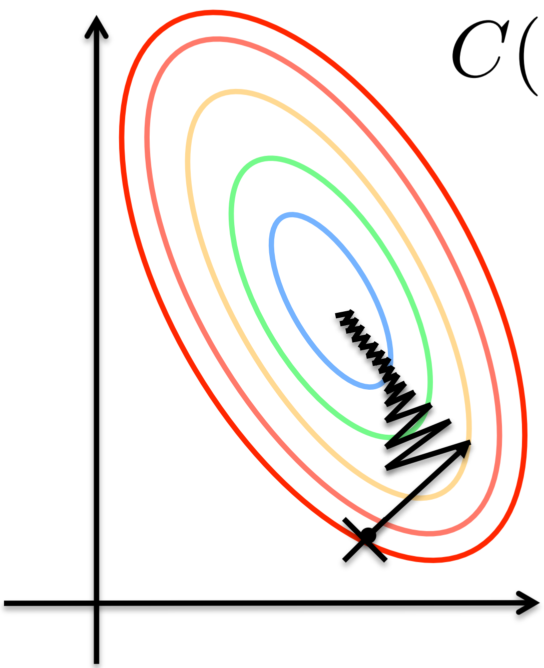
$x_0^i = 1$

Practical tip

- Feature Scaling -

Feature scaling

We want INPUT FEATURES to operate on a SIMILAR SCALE

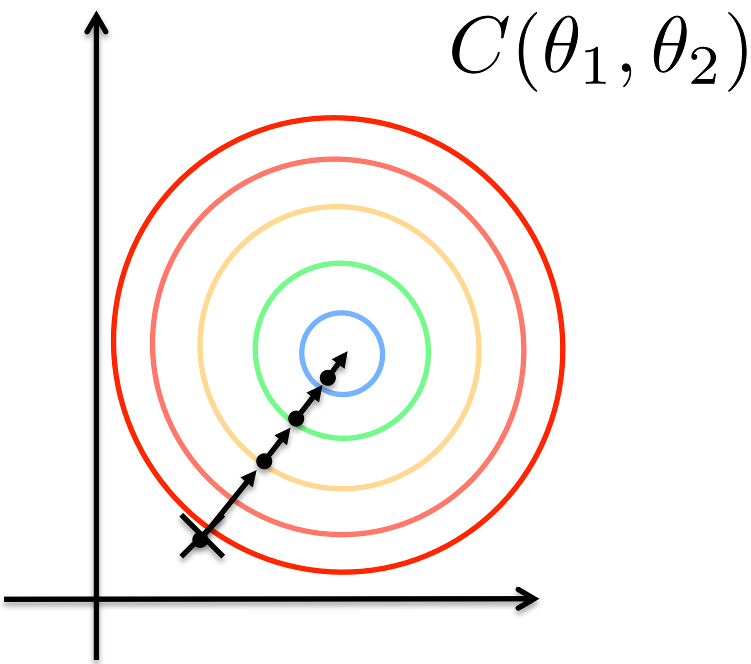


Eg: $x_1 \in (0, 4000)\ \text{ft}^2$

$x_2 \in (1, 5)\ \text{bedrooms}$

Feature scaling

We want INPUT FEATURES to operate on a SIMILAR SCALE



Eg: $x_1 \in (0, 4000)\ \text{ft}^2$

$x_2 \in (1, 5)\ \text{bedrooms}$

$$x_1 = \frac{\text{size ft}^2}{4000}$$

$$x_2 = \frac{\text{no. of bedrooms}}{5}$$

intelligent layer

asi

How to scale features

Get every feature into approximately
$$-1 < x_j < 1$$

centre data

Normalisation:

$$x_i \longrightarrow \frac{x_i - \mu_i}{\sigma_i}$$

normalise data

How to scale features

Get every feature into approximately
$$-1 < x_j < 1$$

Normalisation:

$$x_i \longrightarrow \frac{x_i - \mu_i}{\sigma_i}$$

REMEMBER – new inputs must also be scaled!

$$x^* \longrightarrow \frac{x^* - \mu_i}{\sigma_i}$$

Practical tip
- Debugging Gradient Descent -

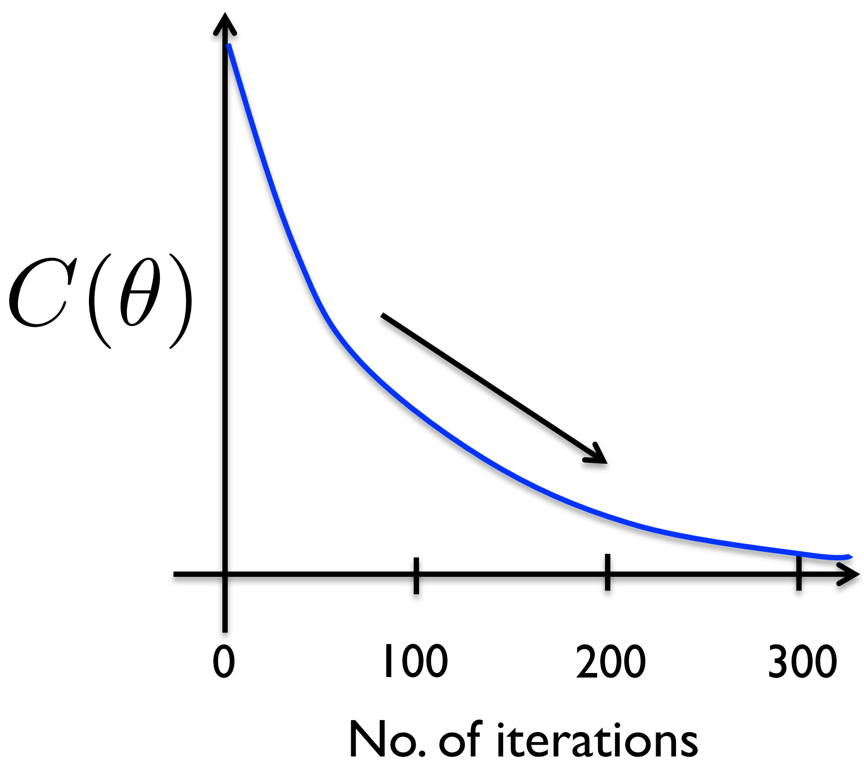
Checking Gradient Descent

Above two parameters we cannot visualise the cost surface effectively

How can we check that Gradient Descent is converging?

Checking Gradient Descent

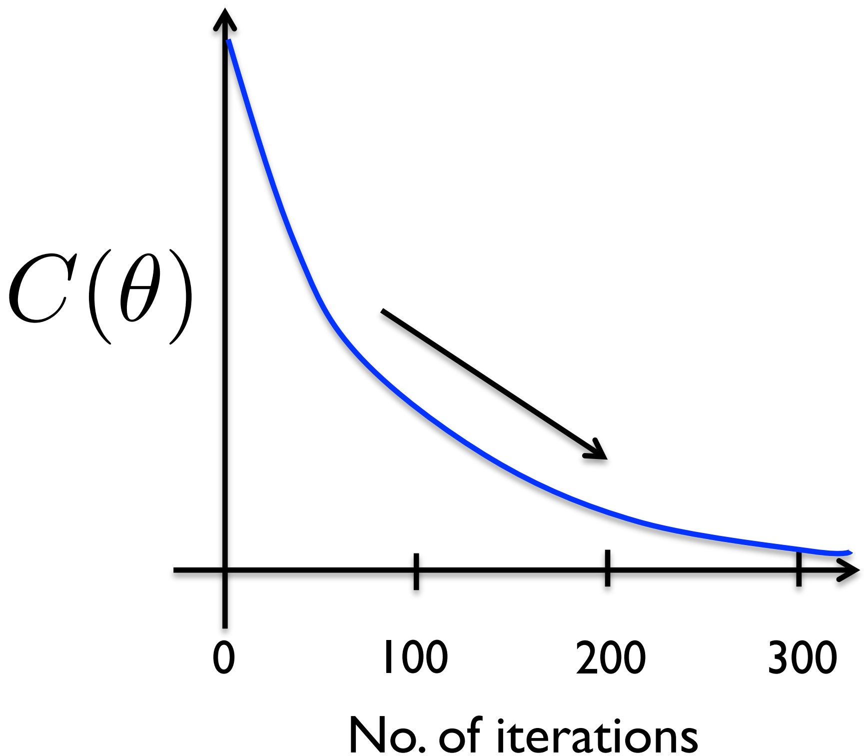
Observe COST FUNCTION against ITERATION



$C(\theta)$ should decrease after every iteration

Checking Gradient Descent

Observe COST FUNCTION against ITERATION



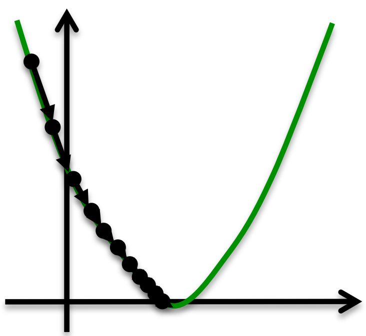
$C(\theta)$ should decrease after every iteration

Evaluating cost function enables us to test when convergence is achieved.

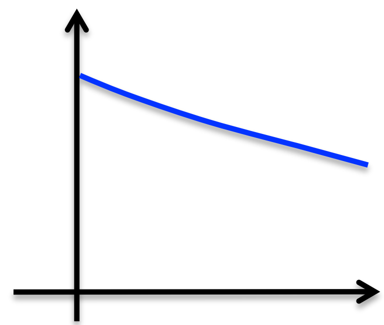
e.g. *difference* < *10^{-3}*

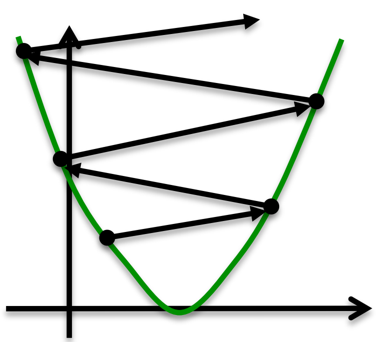
Common problems

SLOW CONVERGENCE and DIVERGENCE can be seen in COST HISTORY

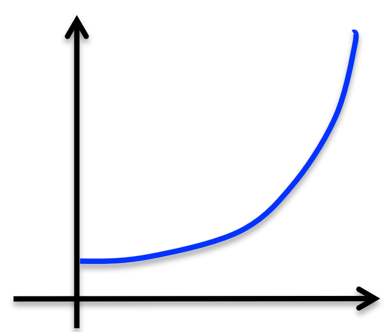


SLOW CONVERGENCE

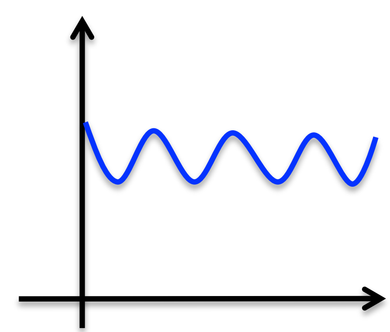




DIVERGENCE

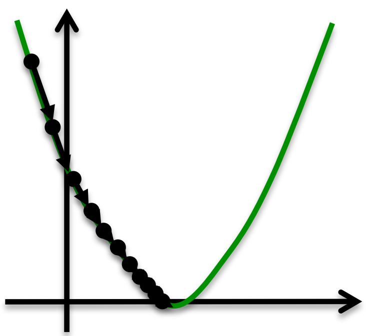


or



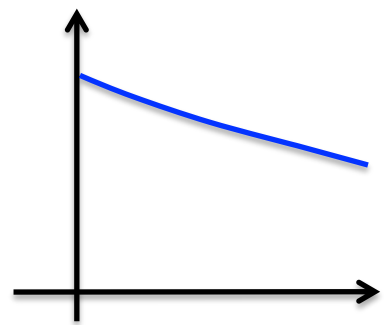
Common problems

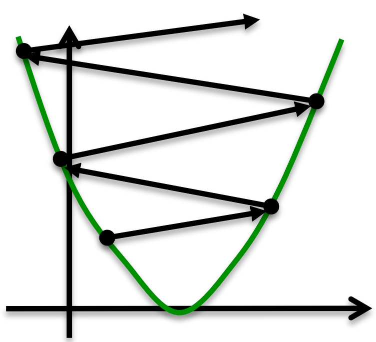
Modify α according to diagnosis of problem



SLOW CONVERGENCE

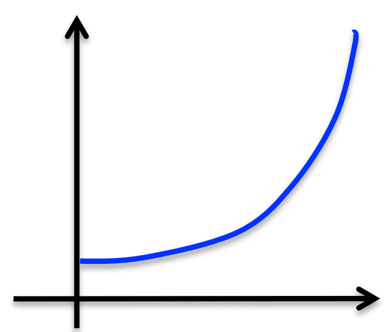
INCREASE α



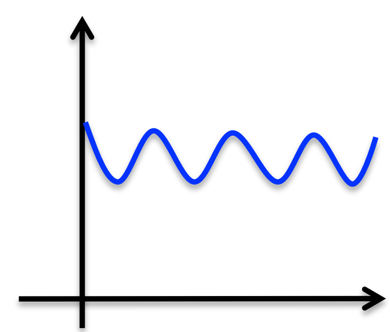


DIVERGENCE

DECREASE α



or



EXERCISE

Time: ? mins

Linear Regression Multiple Variables

- Finish exercises in *Linear Regression with Multiple Variables iPython Notebook*