linear regression with multiple variables

multivariate linear regression

multiple features

multiple variable definitive notation:

$m =$ number of training examples

$n =$ number of features

$x^{(i)} =$ 'input' features of i^{th} training example

$x_j^{(i)} =$ value of feature j in i^{th} training example

Size (feet)^2	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

In the training set above, what is $x_1^{(4)}$?

- The size (in feet^2) of the 1st home in the training set
- The age (in years) of the 1st home in the training set
- The size (in feet^2) of the 4th home in the training set

Correct Response

- The age (in years) of the 4th home in the training set

previously the hypothesis for singlevariate linear regression was denoted as:

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

the hypothesis for **multivariate linear regression** is denoted as:

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

for convenience of notation, define $x_0 = 1$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$\begin{matrix} [\theta_0 & \theta_1 & \theta_2 & \cdots & \theta_n] \\ (n+1) \times 1 \text{ matrix} \end{matrix} \qquad \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

θ^T x

$h_\theta(x) = \theta_0 \boldsymbol{x_0} + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$ which can in turn be simplified as

$= \theta^T x$

gradient descent for multiple variables

hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

parameters: $\theta_0, \theta_1, \ldots, \theta_n$ → denoted simplay as θ; an $n+1$ dimensional vector

cost function:

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

gradient descent:

repeat {

$$\theta_j := \theta_j - a\frac{\partial}{\partial \theta_j}\left(\cancel{J(\theta_0, \theta_1, \ldots, \theta_n)} \quad J(\theta)\right)$$

} (simultaneously update for every $j = 0, \ldots, n$)

When there are n features, we define the cost function as

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

For linear regression, which of the following are also equivalent and correct definitions of $J(\theta)$?

- [x] $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\theta^T x^{(i)} - y^{(i)})^2$

  Correct Response

- [x] $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(\left(\sum_{j=0}^{n}\theta_j x_j^{(i)}\right) - y^{(i)}\right)^2$ (Inner sum starts at 0)

  Correct Response

- [ ] $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(\left(\sum_{j=1}^{n}\theta_j x_j^{(i)}\right) - y^{(i)}\right)^2$ (Inner sum starts at 1)

  Correct Response

- [ ] $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(\left(\sum_{j=0}^{n}\theta_j x_j^{(i)}\right) - \left(\sum_{j=0}^{n} y_j^{(i)}\right)\right)^2$

  Correct Response

previously ($n = 1$):

repeat until convergence {

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

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

(update θ_0 and θ_1 simultaneously)

}

$$\frac{\partial}{\partial \theta_1}J(\theta_0, \theta_1)$$

new algorithm ($n \geq 1$):

repeat until convergence {

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

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

(update θ_j for $j = 0, \ldots, n$ simultaneously)

}

$$\frac{\partial}{\partial \theta_1}J(\theta)$$

gradient descent in practice i – feature scaling

it is important to ensure features used as an input with varying algorithms share similar scales. features with scales differing substantially can skew the model and cause gradient descent to run inefficient; causing many iteration to arrive at the global minimum through continuous iterations to converge

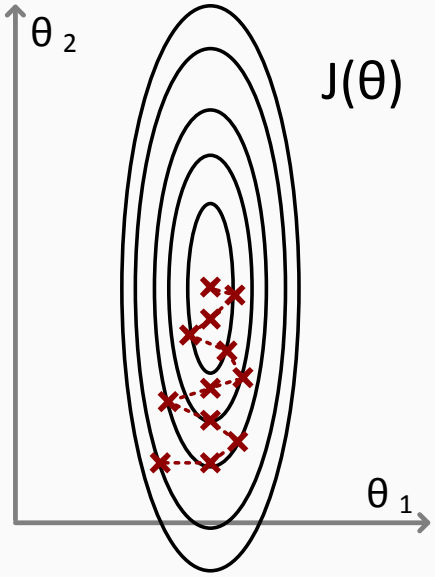
for example:

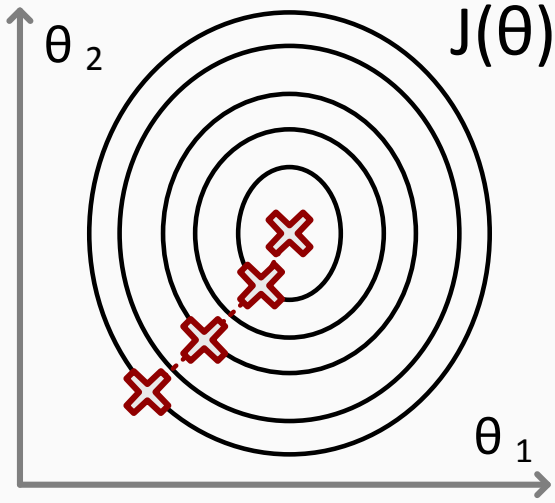
assume x_1 =variable in 10,000s

x_2 =variable in 10s

assume scaled to $x_1 = \frac{x_1}{10{,}000}$

$x_2 = \frac{x_2}{10}$





the general idea is to get each feature into approximately $-1 \leq x_i \leq 1$ range. the rule is subjective and interpretable depending upon the user and data. some data scientists find acceptable ranges as $-3 \leq x_i \leq 3$ or $-\frac{1}{3} \leq x_i \leq \frac{1}{3}$ for larger and smaller feature values, respectively

mean normalization

a prominent method of feature scaling is seen through mean normalization:

replace x_i with $x_i - \mu_i$ to make the features have an approximately zero mean (not applied to $x_0 = 1$):

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

μ_i = average value of x_i in the training set

σ_i = the standard deviation

Suppose you are using a learning algorithm to estimate the price of houses in a city. You want one of your features x_i to capture the age of the house. In your training set, all of your houses have an age between 30 and 50 years, with an average age of 38 years. Which of the following would you use as features, assuming you use feature scaling and mean normalization?

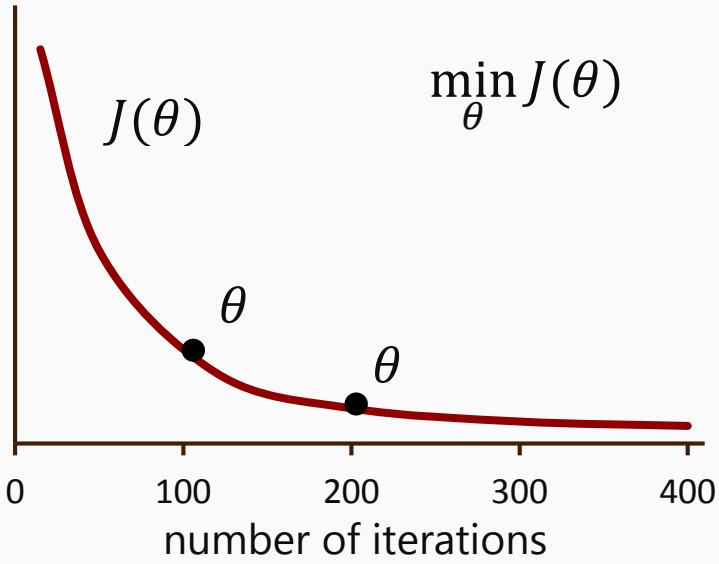
- $x_i = \text{age of house}$
- $x_i = \frac{\text{age of house}}{50}$
- $x_i = \frac{\text{age of house} - 38}{50}$
- $x_i = \frac{\text{age of house} - 38}{20}$

Correct Response

gradient descent in practice ii – learning rate

understanding the learning rate α and how it interacts with the gradient descent update rule $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$ in addition to "debugging" gradient descent for proper function and choosing the value of learning rate α

to determine if gradient descent is working as expected:



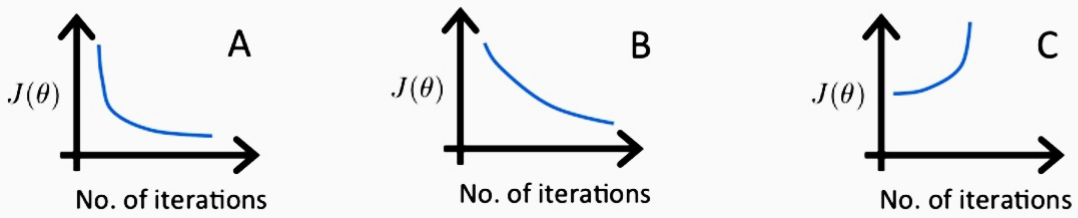
in plotting $J(\theta)$ against the number of gradient descent iterations, the expected behavior is for $J(\theta)$ to effectively decrease after each iteration until convergence

an additional option is to automatically test for convergence through declaring a logical rule; this is difficult (e.g. if $J(\theta)$ decreases by less than 10^{-3} in a single iteration)

gradient descent that experiences a positive slope when plotted against the number of iterations is indicative of divergence likely caused by the learning rate α being too large. a learning rate α too small can cause slow convergence with gradient descent

it is typically to choose α in increments (e.g. , … , 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, …)

Suppose a friend ran gradient descent three times, with $\alpha = 0.01$, $\alpha = 0.1$, and $\alpha = 1$, and got the following three plots (labeled A, B, and C):



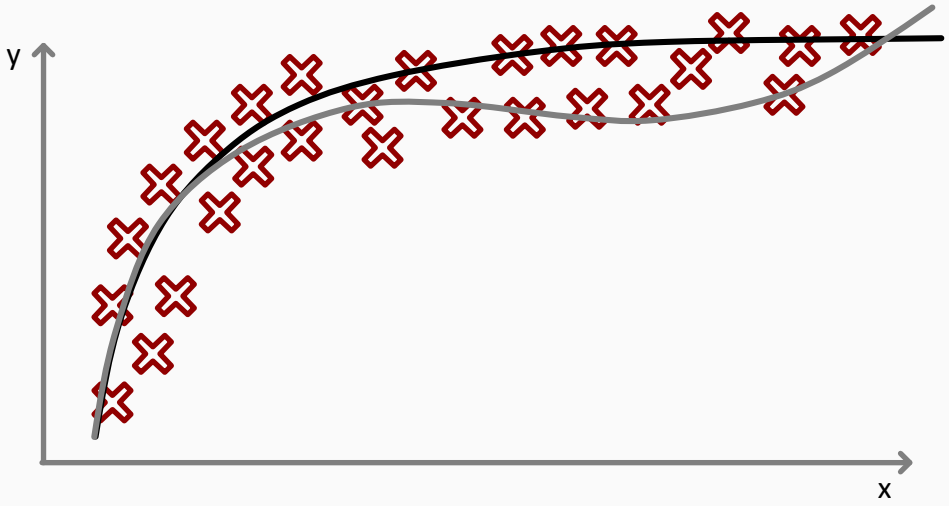
Which plots corresponds to which values of α?

- A is $\alpha = 0.01$, B is $\alpha = 0.1$, C is $\alpha = 1$.
- A is $\alpha = 0.1$, B is $\alpha = 0.01$, C is $\alpha = 1$.

Correct Response
In graph C, the cost function is increasing, so the learning rate is set too high. Both graphs A and B converge to an optimum of the cost function, but graph B does so very slowly, so its learning rate is set too low. Graph A lies between the two.

- A is $\alpha = 1$, B is $\alpha = 0.01$, C is $\alpha = 0.1$.
- A is $\alpha = 1$, B is $\alpha = 0.1$, C is $\alpha = 0.01$.

features and polynomial regression



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

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

the use of polynomial expressions require a greater attention to feature scaling for the effects quadratic equations will have on feature values

Suppose you want to predict a house's price as a function of its size. Your model is

$h_\theta(x) = \theta_0 + \theta_1(\text{size}) + \theta_2\sqrt{(\text{size})}$

Suppose size ranges from 1 to 1000 (feet^2). You will implement this by fitting a model

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

Finally, suppose you want to use feature scaling (without mean normalization).

Which of the following choices for x_1 and x_2 should you use? (Note: $\sqrt{1000} \approx 32$.)

- $x_1 = \text{size},\ x_2 = 32\sqrt{(\text{size})}$
- $x_1 = 32(\text{size}),\ x_2 = \sqrt{(\text{size})}$
- $x_1 = \frac{\text{size}}{1000},\ x_2 = \frac{\sqrt{(\text{size})}}{32}$

Correct Response