Linear Regression

07 September 2021 00:12

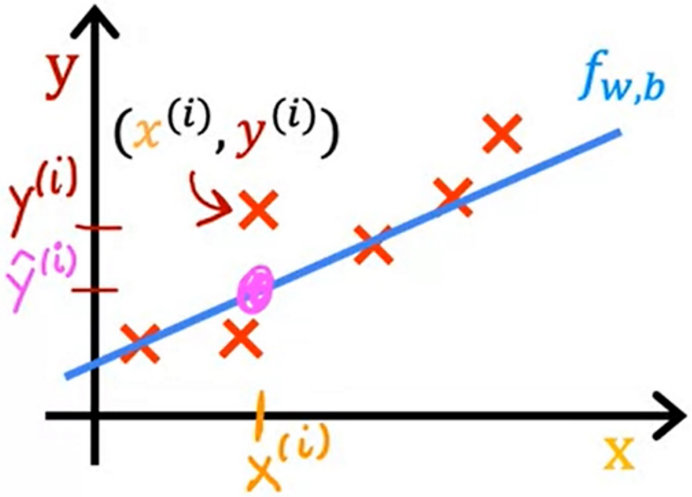
Training set

features: size in feet² (x) | targets: price $1000's ($y$)

size in feet² (x)	price $1000's ($y$)
2104	460
1416	232
1534	315
852	178
...	...

Model: $f_{w,b}(x) = wx + b$

w, b: parameters
coefficients
weights



$$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$$

$$f_{w,b}(x^{(i)}) = wx^{(i)} + b$$

Cost function: Squared error cost function

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(\underbrace{\hat{y}^{(i)} - y^{(i)}}_{\text{error}}\right)^2$$

m = number of training examples

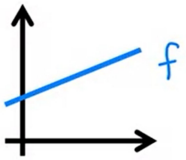
$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}\left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2$$

model:

$f_{w,b}(x) = wx + b$

parameters:

w, b



cost function:

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

goal:

$\underset{w,b}{\text{minimize}}\, J(w,b)$

simplified

$f_w(x) = wx$ $\quad b = \emptyset$

w

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(f_w(x^{(i)}) - y^{(i)})^2$$

$wx^{(i)}$

$\underset{w}{\text{minimize}}\, J(w)$

Have some function $J(w,b)$ for linear regression or any function

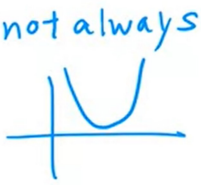
Want $\min_{w,b} J(w,b)$ $\quad \min_{w_1,\ldots,w_n,b} J(w_1, w_2, \ldots, w_n, b)$

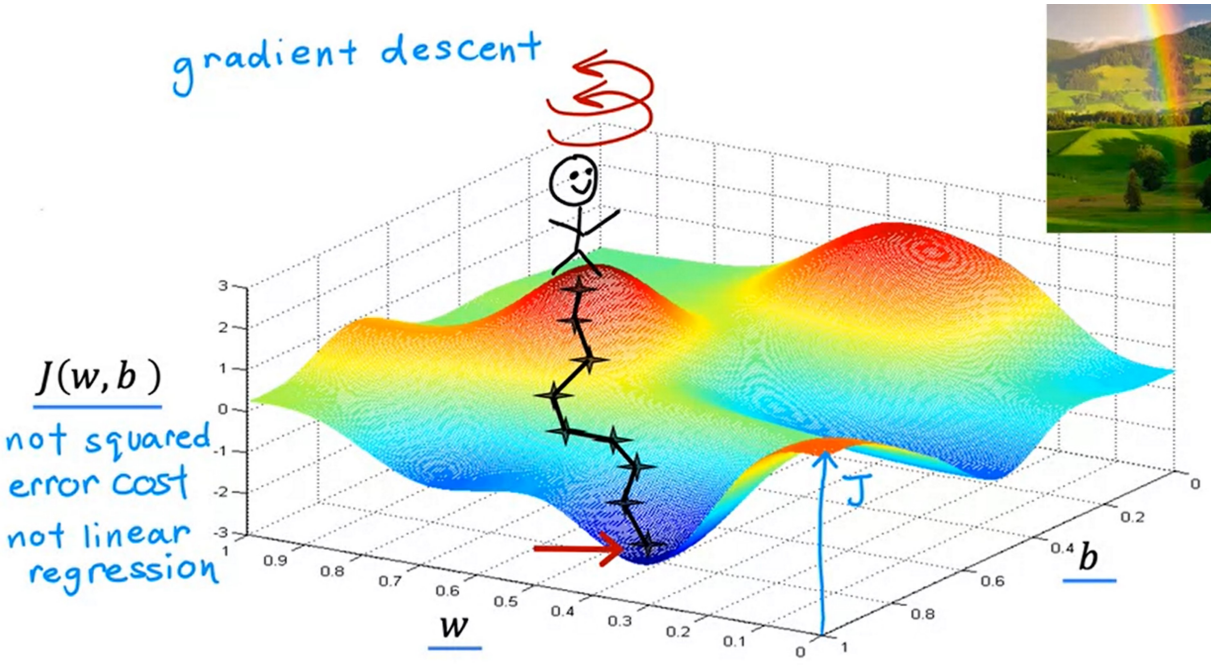
Outline:

- Start with some w, b (set $w=0$, $b=0$)
- Keep changing w, b to reduce $J(w,b)$
- Until we settle at or near a minimum

may have >1 minimum

J not always





Gradient descent algorithm

$$w = w - \alpha \frac{\partial}{\partial w} J(w,b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w,b)$$

Simultaneously update w and b

Correct: Simultaneous update

$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w,b)$

$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w,b)$

$w = tmp\_w$

$b = tmp\_b$

Linear regression model

$$f_{w,b}(x) = wx + b$$

Cost function

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})^2$$

Gradient descent algorithm

repeat until convergence {

$$w = w - \alpha \frac{\partial}{\partial w} J(w,b) \rightarrow \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) x^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w,b) \rightarrow \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$$

}

repeat until convergence {

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)}) \quad x^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})$$

}

Multiple features (variables)

	Size in feet²	Number of bedrooms	Number of floors	Age of home in years	Price ($) in $1000's
	x_1	x_2	x_3	x_4	
	2104	5	1	45	460
$i=2$	1416	3	2	40	232
	1534	3	2	30	315
	852	2	1	36	178
	...	...	...	...	...

$j=1 \dots 4$

$n=4$

x_j = j^{th} feature

n = number of features

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

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

$\vec{x}^{(2)} = [1416 \ 3 \ 2 \ 40]$

$x_3^{(2)} = 2$

Model:

Previously: $f_{w,b}(x) = wx + b$

$$f_{w,b}(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$

example

$$f_{w,b}(x) = 0.1x_1 + 4x_2 + 10x_3 + -2x_4 + 80$$

x_1: size; x_2: #bedrooms; x_3: #floors; x_4: years; 80: base price

$$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

$\vec{w} = [w_1 \ w_2 \ w_3 \dots w_n]$ parameters of the model

b is a number

vector $\vec{x} = [x_1 \ x_2 \ x_3 \dots x_n]$

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b$$

$\cdot$: dot product

multiple linear regression

(not multivariate regression)

	Previous notation	Vector notation
Parameters	$w_1, \cdots, w_n$ b	$\vec{w} = [w_1 \quad \cdots \quad w_n]$ ← vector of length n b still a number
Model	$f_{\vec{w},b}(\vec{x}) = w_1 x_1 + \cdots + w_n x_n + b$	$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$ (dot product)
Cost function	$J(w_1, \cdots, w_n, b)$	$J(\vec{w}, b)$

Gradient descent

repeat {

$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(w_1, \cdots, w_n, b)$

$b = b - \alpha \frac{\partial}{\partial b} J(w_1, \cdots, w_n, b)$

}

repeat {

$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$

$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$

}

Gradient descent

One feature

repeat {

$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right) x^{(i)}$

($\frac{\partial}{\partial w} J(w, b)$)

$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{w,b}(x^{(i)}) - y^{(i)}\right)$

simultaneously update w, b

}

n features ($n \geq 2$)

repeat {

$j = 1$: $w_1 = w_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right) x_1^{(i)}$

($\frac{\partial}{\partial w_1} J(\vec{w}, b)$)

$\vdots$

$j = n$: $w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right) x_n^{(i)}$

$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}\right)$

simultaneously update w_j (for $j = 1, \cdots, n$) and b

}

An alternative to gradient descent

→ Normal equation

- Only for linear regression
- Solve for w, b without iterations

Disadvantages

- Doesn't generalize to other learning algorithms.
- Slow when number of features is large (> 10,000)

What you need to know

- Normal equation method may be used in machine learning libraries that implement linear regression.
- Gradient descent is the recommended method for finding parameters w,b

Feature and parameter values

$\widehat{price} = w_1 x_1 + w_2 x_2 + b$

size #bedrooms

x_1: size (feet^2) range: 300 – 2,000 large

x_2: # bedrooms range: 0 – 5 small

House: $x_1 = 2000$, $x_2 = 5$, $price = \$500k$ one training example

size of the parameters w_1, w_2?

$w_1 = 50$, $w_2 = 0.1$, $b = 50$

$\widehat{price} = \underbrace{50 * 2000}_{100{,}000k} + \underbrace{0.1 * 5}_{0.5k} + \underbrace{50}_{50k}$

$\widehat{price} = \$100{,}050.5k = \$100{,}050{,}500$

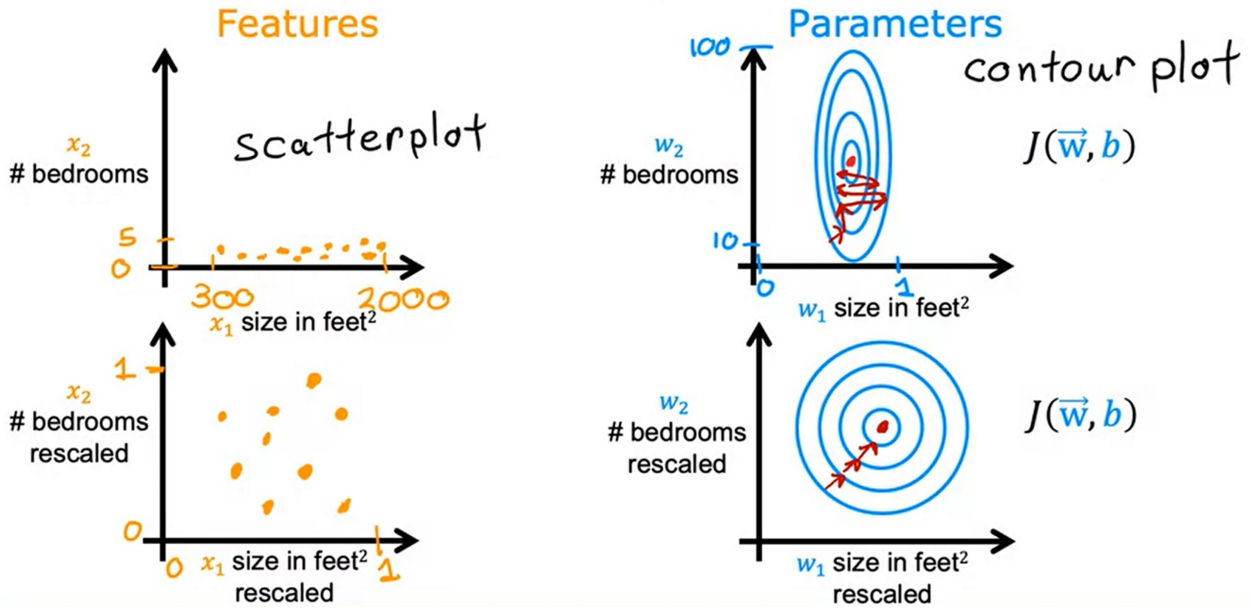
$w_1 = 0.1$, $w_2 = 50$, $b = 50$

small large

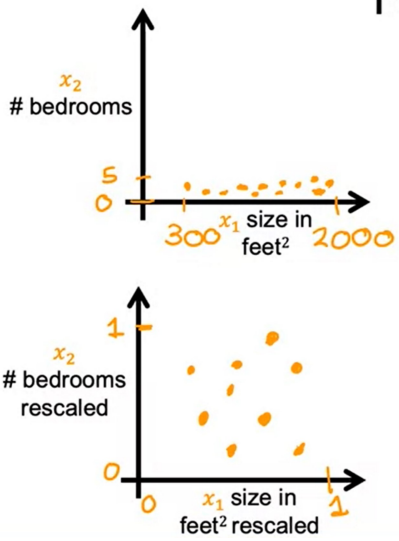
$\widehat{price} = \underbrace{0.1 * 2000k}_{200k} + \underbrace{50 * 5}_{250k} + \underbrace{50}_{50k}$

$\widehat{price} = \$500k$ more reasonable

Feature size and gradient descent



Feature scaling

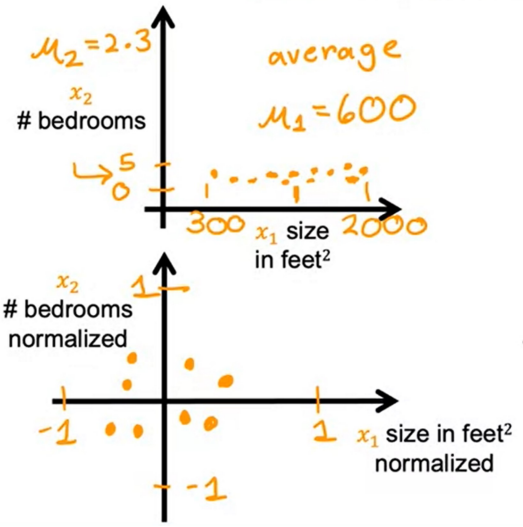


$300 \le x_1 \le 2000$ $\quad$ $0 \le x_2 \le 5$

$x_{1,scaled} = \frac{x_1}{2000}$ max $\quad$ $x_{2,scaled} = \frac{x_2}{5}$ max

$0.15 \le x_{1,scaled} \le 1$ $\quad$ $0 \le x_{2,scaled} \le 1$

Mean normalization



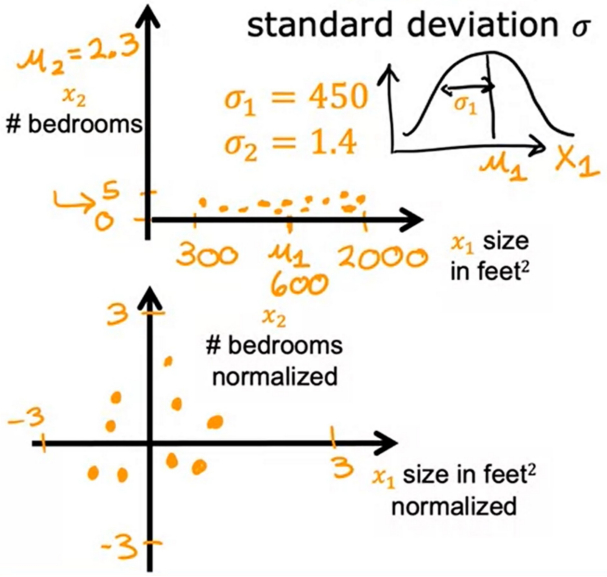
$300 \le x_1 \le 2000$ $\quad$ $0 \le x_2 \le 5$

$$x_1 = \frac{x_1 - \mu_1}{2000 - 300} \qquad x_2 = \frac{x_2 - \mu_2}{5 - 0}$$

max-min $\quad$ max-min

$-0.18 \le x_1 \le 0.82$ $\quad$ $-0.46 \le x_2 \le 0.54$

Z-score normalization



$300 \le x_1 \le 2000$ $\quad$ $0 \le x_2 \le 5$

$$x_1 = \frac{x_1 - \mu_1}{\sigma_1} \qquad x_2 = \frac{x_2 - \mu_2}{\sigma_2}$$

$-0.67 \le x_1 \le 3.1$ $\quad$ $-1.6 \le x_2 \le 1.9$

Feature scaling

aim for about $-1 \le x_j \le 1$ for each feature x_j

$-3 \le x_j \le 3$

$-0.3 \le x_j \le 0.3$

} acceptable ranges

$0 \le x_1 \le 3$	okay, no rescaling
$-2 \le x_2 \le 0.5$	okay, no rescaling
$-100 \le x_3 \le 100$	too large → rescale
$-0.001 \le x_4 \le 0.001$	too small → rescale
$98.6 \le x_5 \le 105$	too large → rescale