1/11/2024 Linear Regression

linear function of input

Predicting a real number

target (y)

Features (d total), each house is feature vector $x \in \mathbb{R}^d$

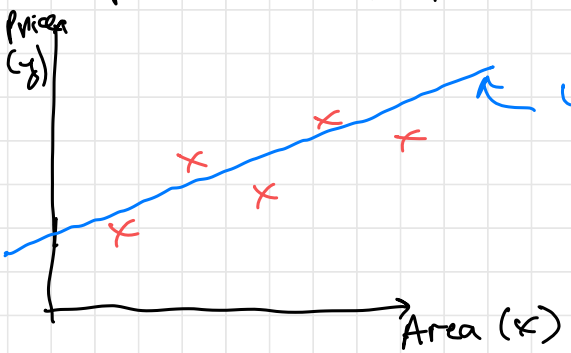
Sale price	Area	#bedrooms	has garage?	...	constant feature
\$500K $= y^{(1)}$	1200	2	0	$= x^{(1)}$	1
\$700K	2000	3	1		1
	⋮				1

↓ makes b param optional

Dataset $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$

of size n

Simplest case: $d = 1$



slope, intercept

$y = wx + b$

parameters (to be learned)

When $d > 1$:

$y = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b$

$= w^T x + b$

Parameters: $w \in \mathbb{R}^d$, $b \in \mathbb{R}$

$1 \times d$ $\cdot$ $d \times 1$ $= 1 \times 1$

Q: How to choose good w & b?

A: Define a loss function

$L(w,b)$ = {How bad do w & b fit our observed data?}

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\underbrace{w^T x^{(i)} + b}_{\text{model prediction}} - \underbrace{y^{(i)}}_{\text{true output}}\right)^2$$

Goal: Find w & b that minimize $L(w,b)$

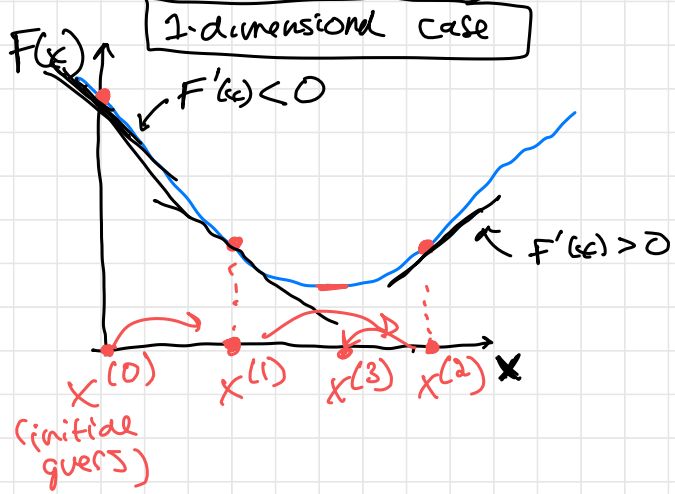
optimization problem?

Gradient Descent General method for minimizing function

Given: Function F from $\mathbb{R}^d \to \mathbb{R}$, differentiable

Gradient will try to find x that minimizes $F(x)$

1-dimensional case



Have current guess $x^{(t)}$

If $F'(x^{(t)}) < 0$, increase $x^{(t)}$ to yield $x^{(t+1)}$

If $F'(x^{(t)}) > 0$ decrease $x^{(t)}$ to yield $x^{(t+1)}$

d-dimensional case optimizing w.r.t. $x \in \mathbb{R}^d$

Partial Derivative: $\frac{\partial F}{\partial x_i}$ ← Take derivative wrt x_i holding all other x_j's constant

New G.D. Rule:

For each $i = 1, \ldots, d$:

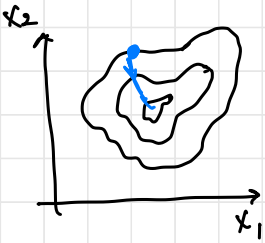
If $\frac{\partial F}{\partial x_i}\Big|_{x = x^{(t)}} < 0$, increase $x_i^{(t)}$

If $\frac{\partial F}{\partial x_i}\Big|_{x = x^{(t)}} > 0$, decrease $x_i^{(t)}$

Gradient $\nabla_x F(x) = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \cdots \frac{\partial F}{\partial x_d}\right]$

Starting at $x^{(t)}$, best direction to go (to minimize F) is direction of negative gradient

Fact: Negative gradient is direction of steepest descent



Gradient Descent Algorithm:

$x^{(0)} \leftarrow [0, 0, \ldots, 0] \in \mathbb{R}^d$

for t in $1, \ldots, T$: ← total # steps

$x^{(t)} \leftarrow x^{(t-1)} - \eta \nabla_x F(x^{(t-1)})$

↖ learning rate (e.g. 0.01)

return $x^{(T)}$

Gradient Descent for Linear Regression

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}\left(w^T x^{(i)} - y^{(i)}\right)^2$$

$$\nabla_w L(w) = \frac{1}{n}\sum_{i=1}^{n} 2\cdot\underbrace{\left(w^T x^{(i)} - y^{(i)}\right)}_{\text{Scalar}}\cdot\underbrace{x^{(i)}}_{\text{Vector}}$$

$\frac{d}{dw} 8w = 8$

GD for Linear Regression:

$w^{(0)} \leftarrow [0, \dots 0] \in \mathbb{R}^d$

for $t = 1, \dots, T$:

$$w^{(t)} \leftarrow w^{(t-1)} - \eta \cdot \frac{1}{n}\sum_{i=1}^{n} 2\cdot\underbrace{\left(w^T x^{(i)} - y^{(i)}\right)}_{\text{Determines if we add or subtract multiple of } x^{(i)}}\cdot x^{(i)}$$

return $w^{(T)}$

If $w^T x^{(i)} - y^{(i)} > 0$: prediction too large

subtract multiple of $x^{(i)}$ from w

$\Rightarrow$ $w^T x^{(i)}$ smaller

If $w^T x^{(i)} - y^{(i)} < 0$: prediction too small,

add multiple of $x^{(i)}$

$\Rightarrow$ $w^T x^{(i)}$ bigger

Want to compute $\sum_{i=1}^{n}\left(w^T x^{(i)} - y^{(i)}\right)\cdot x^{(i)}$

$$X = \begin{bmatrix} -x^{(1)}- \\ -x^{(2)}- \\ \vdots \\ -x^{(n)}- \end{bmatrix}, \text{ so } X^T = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(n)} \\ | & | & & | \end{bmatrix}$$

Want a vector whose i-th entry is $w^T x^{(i)} - y^{(i)}$

Call this v. Then desired quantity is just $X^T v$