Week 3 : Classification

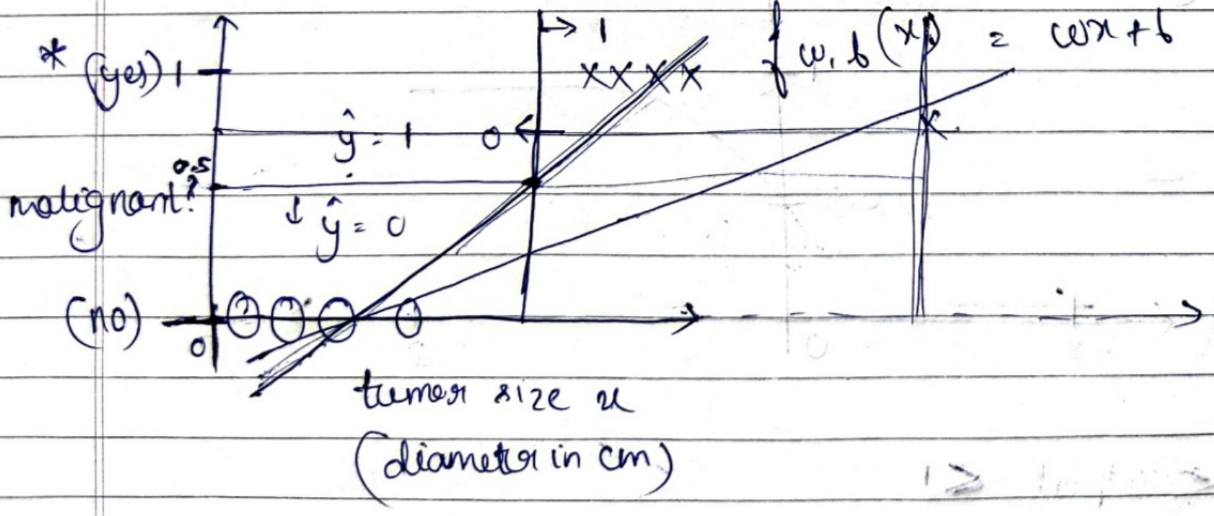
⇒ Classification with logistic regression

Video 1: Motivation

Question	Answer "y"	
Is this email span ?	no	yes
Is the transaction fraudulent ?	no	yes
Is the tumour malignant ?	no	yes
	0 (negative) (absence)	1 (true) (presence)

can be swapped

* y can be only one of 2 values.
* "binary classification"
  class = category

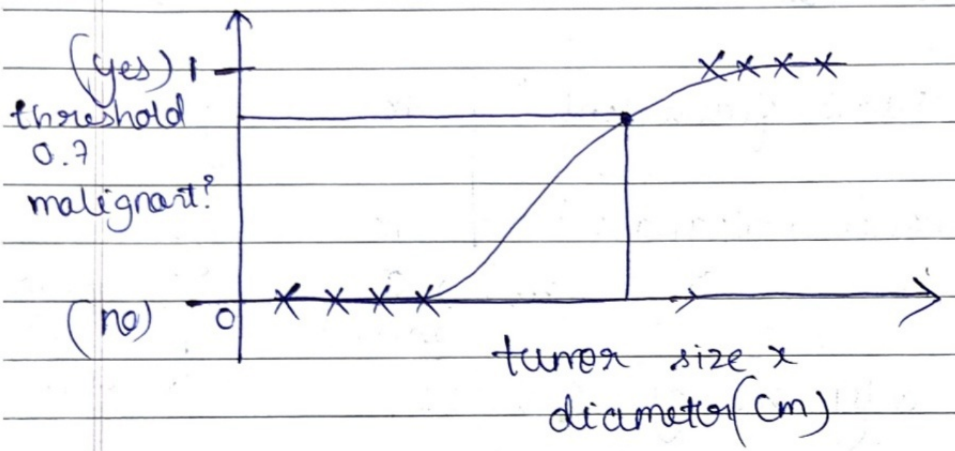


O benign
× malignant

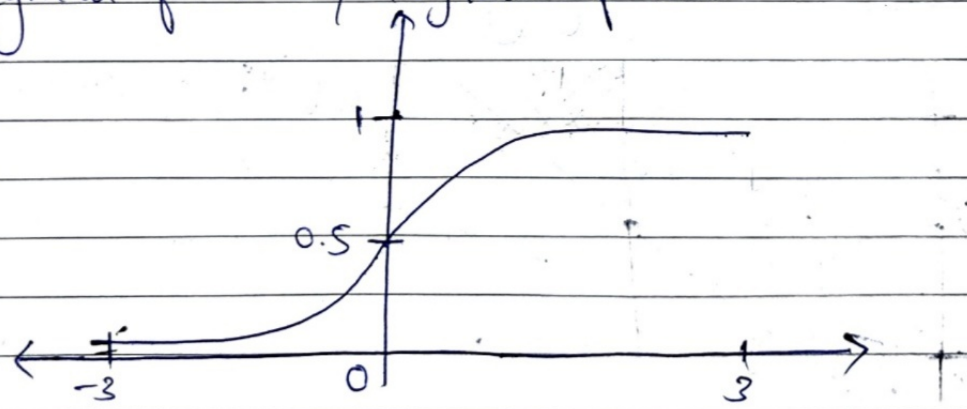
if $f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$
if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$

Logistic ~~regress~~ regression is used to solve binary classification problems where the output is either 0 & 1

Video2: Logistic Regression



Sigmoid function / logistic function



$0 \leq$ output ≤ 1

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

$f_{\vec{w},b}(\vec{x})$

$$z = \vec{w} \cdot \vec{x} + b$$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$f_{\vec{w},b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1+e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Interpretation of logistic regression output.

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w} \cdot \vec{x} + b)}}$$

probability that the class is 1

Example:

x is "tumor size"

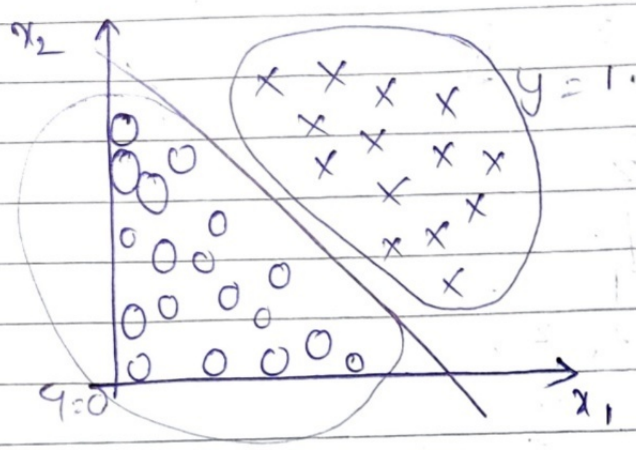
y is 0 or 1

$f_{\vec{w},b}(\vec{x}) = 0.7$

70% chance that y is 1

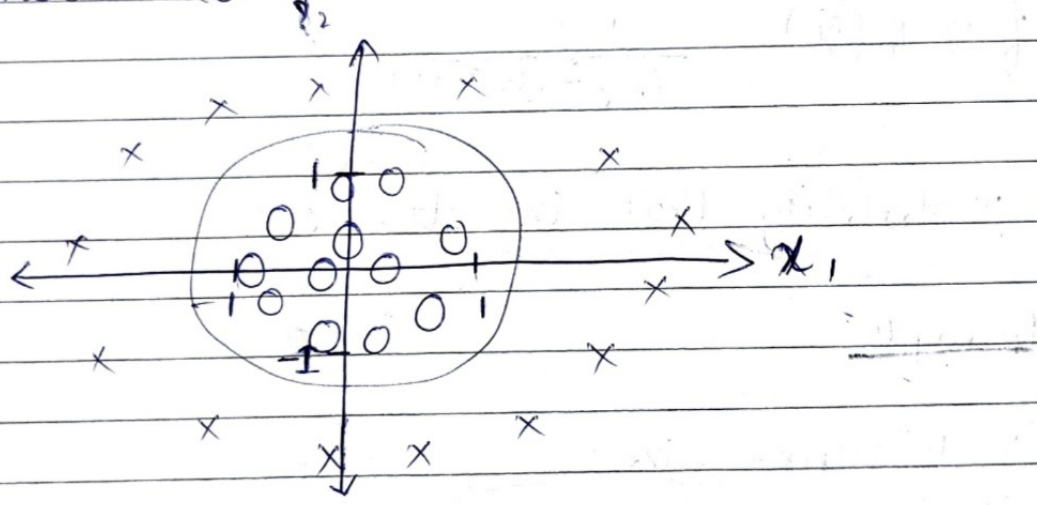
Page No.
Date

Video 5: Decision Boundary



$$f_{(w,b)}(\vec{x}) = g(z) = g(w_1 x_1 + w_2 x_2 + b)$$

Non linear decision boundaries



$$z = w_1 x_1^2 + w_2 x_2^2 + b$$

$$f_{\vec{w},b}(\vec{x}) = g(z) = g(w_1 x_1^2 + w_2 x_2^2 + b)$$

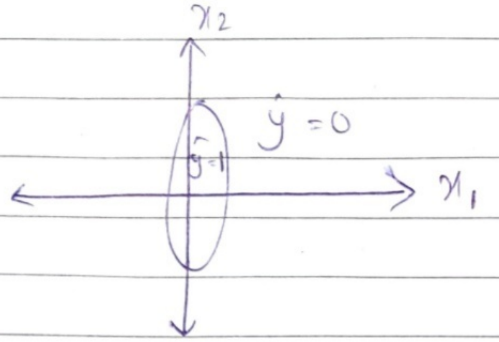
decision boundary: $z = x_1^2 + x_2^2 - 1 = 0$

$$x_1^2 + x_2^2 = 1$$

$$x_1^2 + x_2^2 < 1$$

$$y = 0$$

Non-Linear Decision Boundaries



$$f_{\vec{w},b}(\vec{x}) = g(z)$$
$$= g(w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2)$$

⇒ Cost Function For Logistic Regression

Training Set

tumor size (cm)	. . .	patient's age x_n	malignant
		52	1
10		73	0
2		55	0
5		49	1
12			

⇒ target y is 0 or 1.

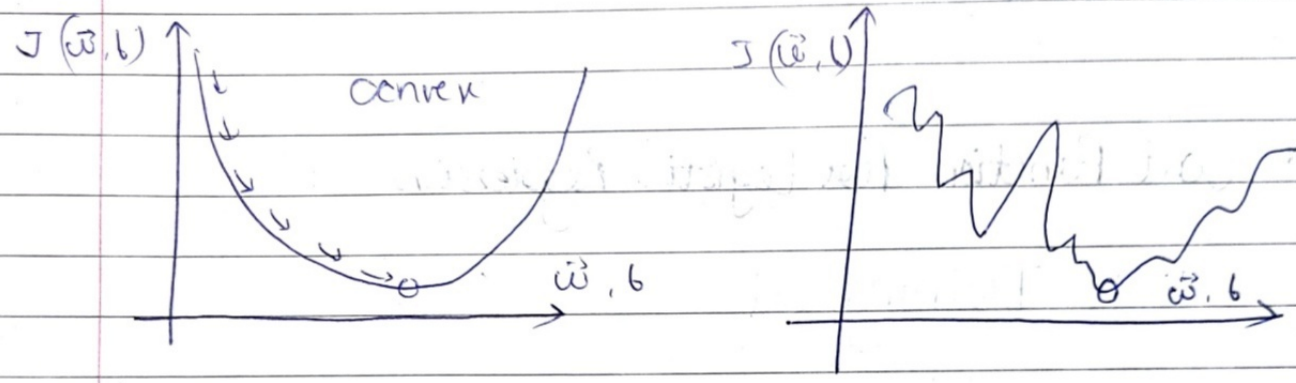
$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w}\cdot\vec{x} + b)}}$$

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

linear regression → logistic regression

$$f_{\vec{w},b}(\vec{x}) = \vec{w}\cdot\vec{x} + b$$

$$f_{w,b}(\vec{x}) = \frac{1}{1+e^{-(\vec{w}\cdot\vec{x}+b)}}$$

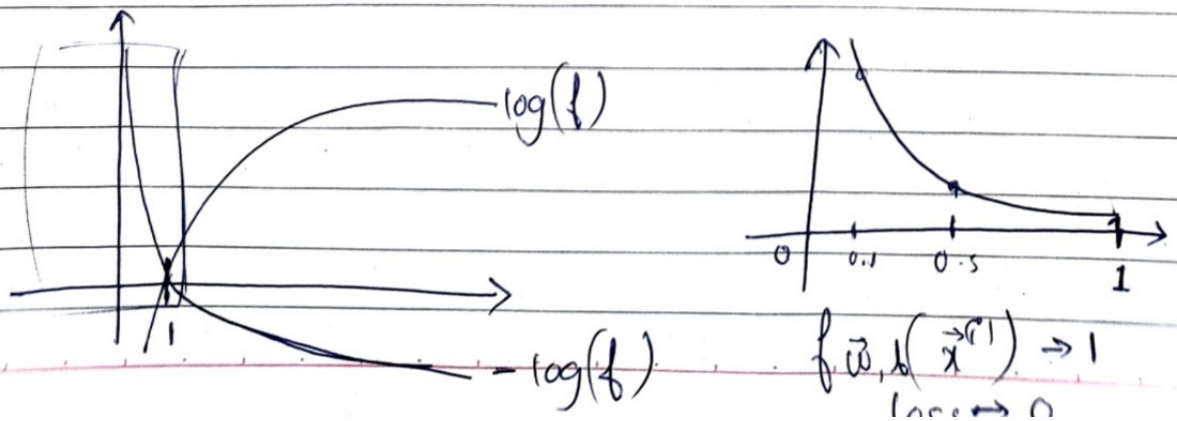


$$J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m} \left[\frac{1}{2} f_{\vec{w},b}(\vec{x}^{(i)} - y^{(i)})^2\right]$$

loss on a single training example
function of x and true label y

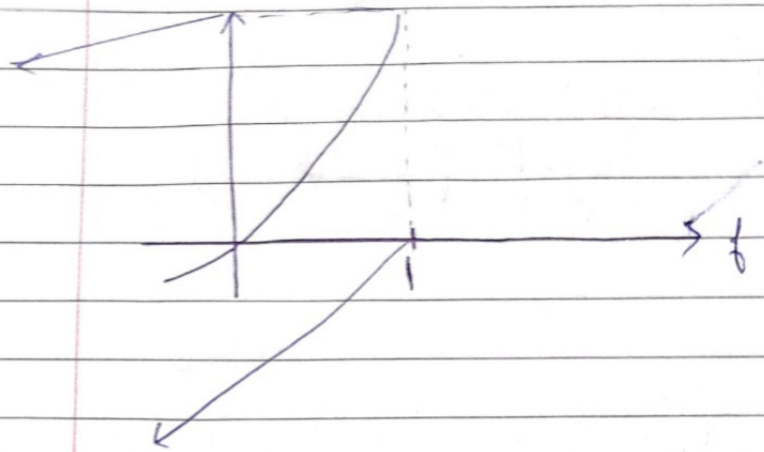
$$L = L\left(f_{\vec{w},b}(\vec{x}^{(i)}, y^{(i)})\right)$$

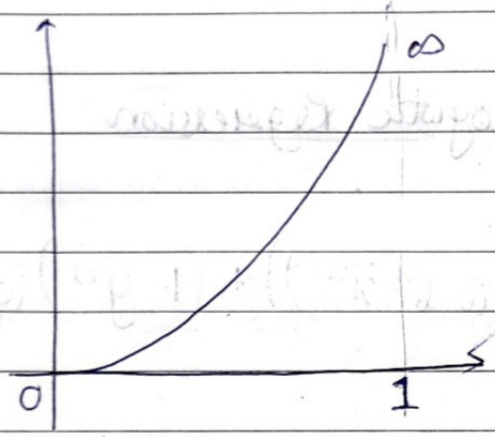
$$L\left(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}\right) = \begin{cases} -\log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) & y^{(i)} = 1 \\ -\log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)})\right) & y^{(i)} = 0 \end{cases}$$



$f_{\vec{w},b}(\vec{x}^{(i)}) \to 1$

loss → 0





as loss $f_{\vec{w},b}(\vec{x}^{(i)}) \to 0$

than loss $\to 0$

$L f_{\vec{w},b}(\vec{x}^{(i)}, y^{(i)})$ if $y^{(i)} = 0$

Due to this cost function will be convex & will reach global minimum.

Video 2: Simplified cost function for Logistic Regression

$$L(f_{\vec{w},b}(\vec{x}^{(i)}, y^{(i)}) = \begin{cases} -\log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{if } y^{(i)} = 0 \end{cases}$$

$$L(f_{\vec{w},b}(\vec{x}^{(i)}, y^{(i)}) = -y^{(i)} \log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1-y^{(i)}) \log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))$$

loss

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)} \log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - (1-y^{(i)}) \log\left(1 - f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)$$

Cost function:

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

$\Rightarrow$ Gradient Descent for Logistic Regression

Video 1:

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

repeat {

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

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

}

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

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

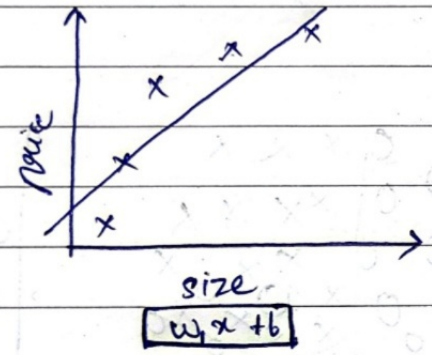
Linear regression : $f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$

Logistic regression : $f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$

⇒ The problem of overfitting

Video 1: The problem of overfitting

* Regularisation - minimises overfitting
* Regression example :

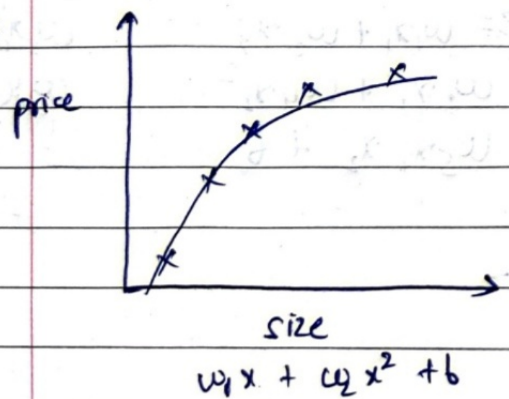


underfit : does not fit the training set well

high bias

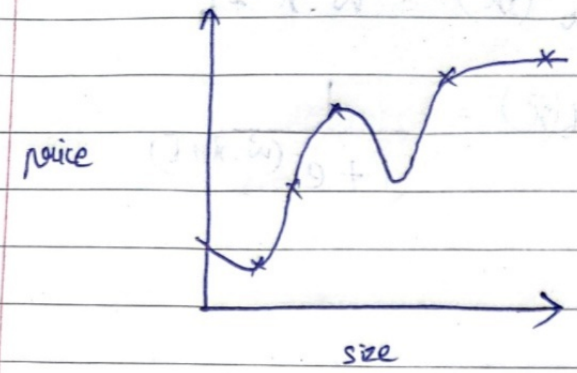
* here there is preconception that the housing prices are going to a completely linear function of size.

*



generalisation

fits training set pretty well

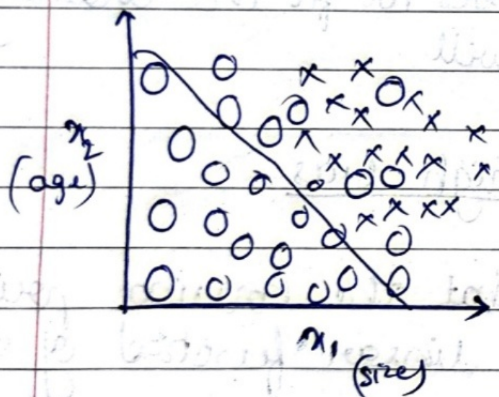


$w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

overfit / high variance

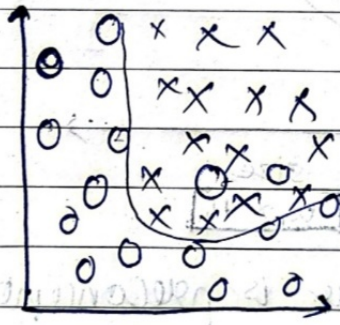
* fits the training data too well.

Classification

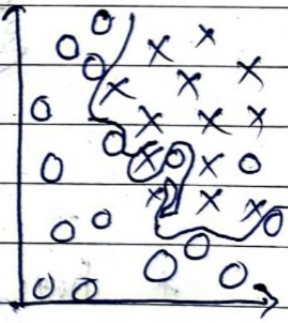


$z = w_1 x_1 + w_2 x_2 + b$

$f_{\vec{w},b}(\vec{x}) = g(z)$



$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 + b$



$z = w_1 x_1 + w_2 x_1^2 x_2 + w_3 x_2 + \dots + b$

g is the sigmaid function.

DATE : / /

Video 2: Addressing Overfitting

* collect more training data
* use fewer features.

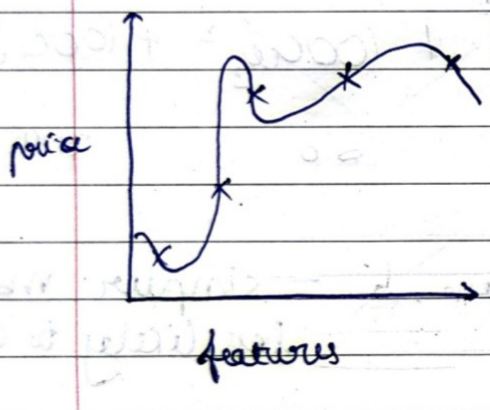
all features
+
insufficient data
↓
overfit

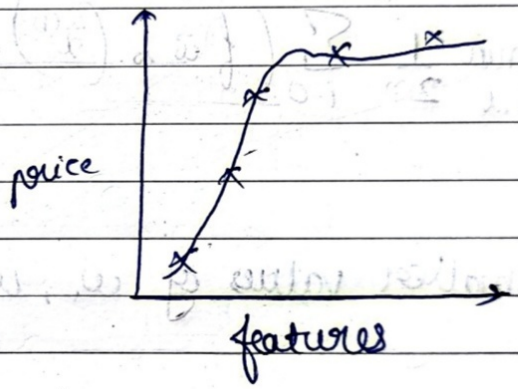
selected features
size
bedrooms
age

just right

feature selection

Disadvantage:
useful features could be lost





$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$

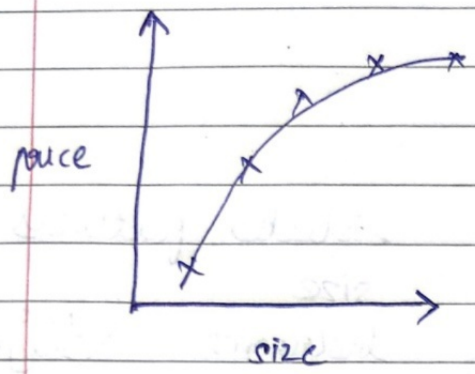
$f(x) = 13x - 0.23x^2 + 0.000014x^3 - 0.0001x^4 + 10$

smaller values

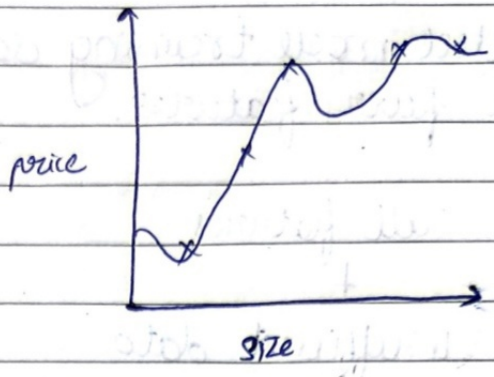
* it makes very less difference if you regularise b or not

PAGE NO.
DATE: / /

Video 8: Cost function with regularisation



$w_1 x + w_2 x^2 + b$



$w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$

make w_3, w_4 really small (≈ 0)

$$\min_{\vec{w},b} \frac{1}{2m} \sum_{i=0}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + 1000 w_3^2 + 1000 w_4^2$$

≈ 0 ≈ 0

* Smaller values of $w_1, w_2, \dots, w_n, b$ — simpler model less likely to overfit

* ~~pen~~ penalise all the parameters

$$J(\vec{w},b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \underbrace{\frac{\lambda}{2m} \sum_{j=1}^{m} w_j^2}_{\text{regularisation term}} + \frac{\lambda}{2m} b^2$$

makes little difference

n = no. of features

λ = regularisation parameter

like α, λ is a hyperparameter

÷ by $2m$ because the terms are scaled uniformly

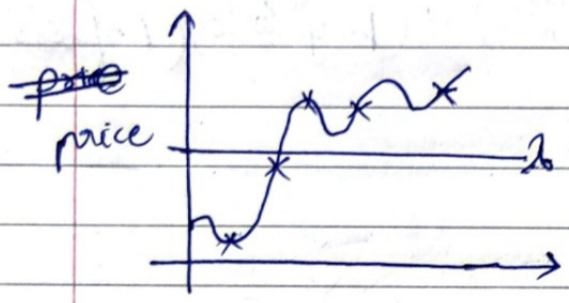
* if $\lambda = 0$, we aren't regularising ... ∴ overfitting

* If $\lambda = 10^{10}$ (large large number)

$$f_{\vec{w},b}(w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b)$$
$$\approx 0 \quad \approx 0 \quad \approx 0 \quad \approx 0$$

$f(x) = b$

does not fit



Video 4: Regularised Linear Regression

$$\min_{\vec{w},b} J(\vec{w},b) = \min_{\vec{w},b} \left[\frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} w_j^2 \right]$$

Gradient Descent

Repeat {

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

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

don't have to regularise b

Video 5: Regularised logistic regression

$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x_1^2 x_2^3 + \ldots$$

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1+e^{-z}}$$

Cost function

$$J(\vec{w},b) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log\left(f_{\vec{w},b}(\vec{x}^{(i)})\right) + (1-y^{(i)}) \log\left(1-f_{\vec{w},b}(\vec{x}^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} w_j^2$$