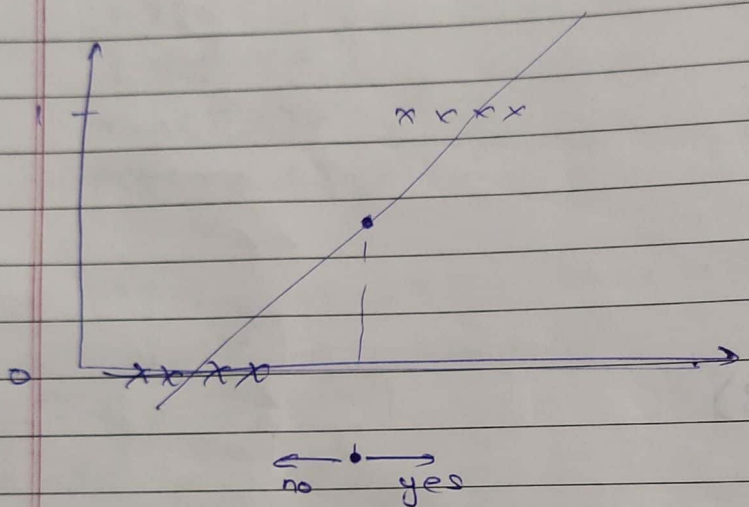
☆ Week 3 ☆

--@ Classification — — —

★ Classification.

$y \in \{0, 1\}$ binary classification

$y \in \{0, 1, 2, 3\}$ latter



Threshold classifier output $h_\theta(x)$ at 0.5:

→ If $h_\theta(x) \geq 0.5$, predict $y = 1$

→ else $y = 0$.

In classification

$y = 0$ or $y = 1$

$h_\theta(x)$ can be (≥ 1) / (< 0)

In logistic regression

$0 \leq h_\theta(x) \leq 1$

Hypothesis Representation

1. Logistic regression Model

Want $0 \le h_\theta(x) \le 1$

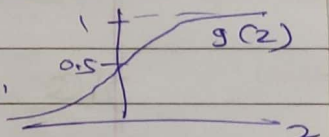
$h_\theta(x) = \theta^T x$ previously

$h_\theta(x) = g(\theta^T x)$ now

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

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}}$$

sigmoid function
logistic function.



- Interpretation of Hypothesis output.

$h_\theta(x)$ = estimated probablity that $y=1$ on input x.

eg. If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

$h_\theta(x) = 0.7$ $\quad y=1$

↳ 70% chance of being melignant

$h_\theta(x) = P(y=1 \mid x; \theta)$

(given x; parameterised by θ)

$P(y=0 \mid x;\theta) + P(y=1 \mid x;\theta) = 1$

i) i) $g(z) \ge 0.5$ when $z \ge 0$

from graph $h_\theta(x) = g(\theta^T x) \ge 0.5$ whenever $\theta^T x \ge 0$

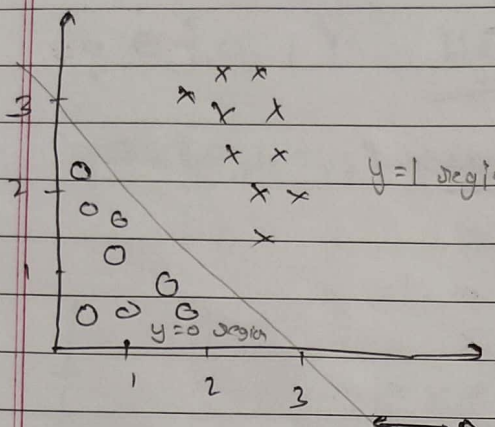
ii) $h_\theta(x) = g(\theta^T x)$, $g(z) < 0.5$, $\theta^T x \le 0$.

i) • predict $y=1$ if $h_\theta(x) \ge 0.5 \rightarrow \theta^T x \ge 0$

ii) • predict $y=0$ if $h_\theta(x) < 0.5 \rightarrow \theta^T x \le 0$

Decision Boundary — — —

- Decision Boundary.



Decision Boundary.

$h_\theta(x) = 0.5$

$x_1 + x_2 = 3$

$h_\theta(x) = g(\underset{-3}{\theta_0} + \underset{1}{\theta_1 x_1} + \underset{1}{\theta_2 x_2})$

If θ found out to be $\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

- predict $y=1$ if $\theta^T x \geq 0$

$-3 + x_1 + x_2 \geq 0$

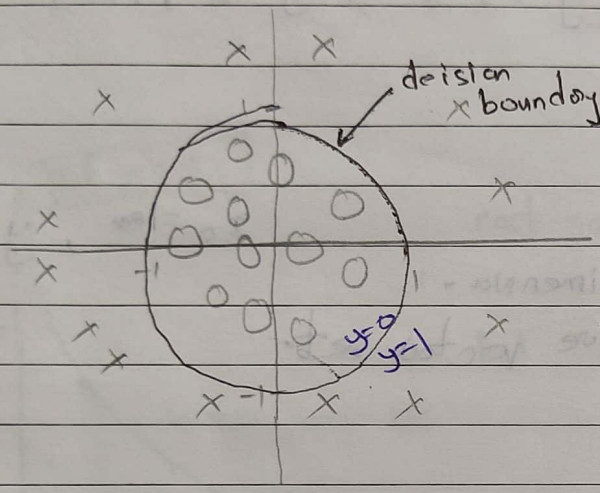
- $y=0$ if $x_1 + x_2 < 3$

Decision Boundary is property of Hypothesis $h_\theta(x) = g(\theta^T x)$ and of parameters of hypothesis

and not property of Data set.

- Non-linear decision boundary

In linear region we added higher degree term. Same here.



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2)$

$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

- predict $y=1$ if $\theta^T x \geq 0$

$-1 + x_1^2 + x_2^2 \geq 0$

$x_1^2 + x_2^2 \geq 1$

- We can add more terms like $x_1^2 x_2$, $x_1^3 x_2$ to get more complex decision boundary.

---② Cost function - - -

- Cost function : logistic regression.

Given:-

* Training set $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

- m examples

- $x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$ dimensional feature vector set. $x_0 = 1$, $y \in \{0, 1\}$

* $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$

- How to choose parameter θ.

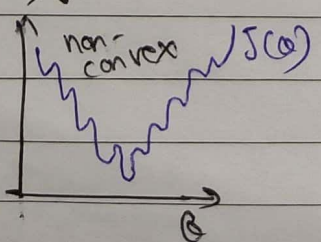
- cost fn before: (Linear regression) $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

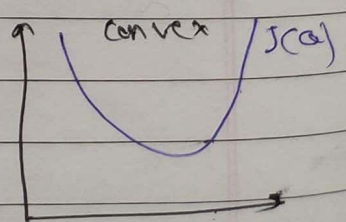
$\text{cost}(h_\theta(x^{(i)}), y)$

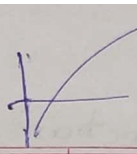
$\text{cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

$h_\theta(x^{(i)}) = \frac{1}{1 + e^{-\theta^T x}}$

making this is FINE, but this gives non-convex fn, & that does not goventee global minima. It could be case we got local minima



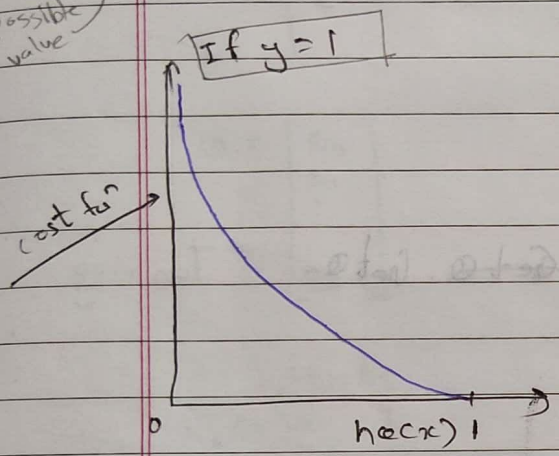




- Cost function: logistic regression:

$$cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0. \end{cases}$$

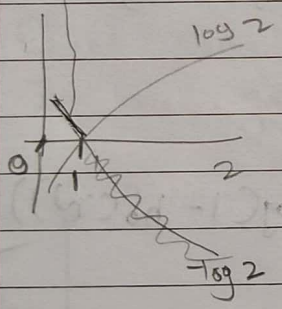
$0 \le h_\theta(x) \le 1$
say it is 0 to 1 possible value



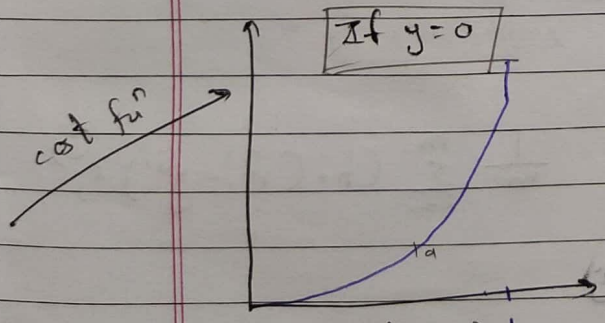
cost = 0, if $y=1$, $h_\theta(x)=1$

But as $h_\theta(x) \longrightarrow 0$
$cost \longrightarrow \infty$

Captures intution that if $h_\theta(x)=0$ (predict $P(y=1|x;\theta)=0$), but $y=1$. we'll penalize learning algorithm by a very large cost.



We are wrong by 1 (in probability) because we told it is begin & found to be melignant, $\therefore$ cost $\longrightarrow \infty$



Same here.

✓ 1) If $h_\theta(x)=y$, then $cost(h_\theta(x), y)=0$ (for $y=0$ & $y=1$)
✓ 2) If $y=0$, then $cost(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 1$
✓ 3) If $y=0$, then $cost(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 0$
✓ Regardless of whether $y=0$ / $y=1$, If $h_\theta(x)=0.5$, then $cost(h_\theta(x), y) > 0$ a case

$$cost(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$$

$\text{cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0. \end{cases}$

y can only be 0/1

---

② Simplified cost function and Gradient Descent.

* logistic regression cost function:-

$$\text{cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y) \log(1-h_\theta(x))$$

→ Principal of maximum likelihood estimation. statics.
That why this is cost fun, other fun can also work as cost fun by combining two fun.

→ convex

- To fit parameters θ:

$$\min_\theta J(\theta) \longrightarrow \text{Get } \theta. \quad \text{Use } \theta. \quad \text{Training}$$

- To make a prediction

$$\text{output } h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \quad P(y=1 \mid x; \theta) \quad \text{Testing.}$$

- Gradient Descent.

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]$$

Want to $\min_\theta J(\theta)$:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \rightarrow \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

simultaneously update all θ_j

}

- Same eqⁿ of linear regⁿ

Repeat {

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

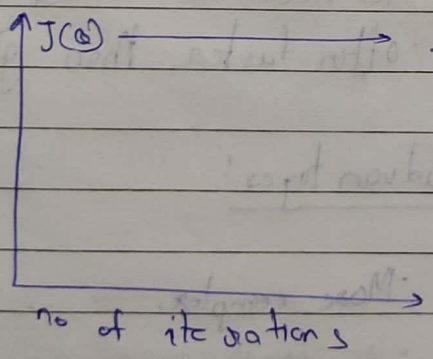
simultaneously update all θ_j)

}

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

logistic	linear
$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$	$h_\theta(x) = \theta^T x$

* Check same way as in linear regⁿ for Is my gradient descent converging.



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

Page No. 27

Date: / /

--(9) Advanced optimization---

* logistic regression : Advanced optimization

cost function $J(\theta)$, Want $\min_{\theta} J(\theta)$.

- Given θ, we have code that can compute.

$J(\theta)$

$\frac{\partial}{\partial \theta_j} J(\theta)$ -- (for $j = 0, 1, 2, \dots$)

Gradient descent

{ repeat

}

optimization algorithm

- Optimization algos:-
  - Gradient descent
  - Conjugate gradient
  - BFGS
  - L-BFGS

Advantages:

- No need to manually pick α.
- often faster than gradient descent.

Disadvantages:

- More complex.
- We don't ~~come~~ completly know how it is working, hard enough

eg.) $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$

$J(\theta) = (\theta_1 - 5)^2 + (\theta_2 - 5)^2$

$\frac{\partial}{\partial \theta_1} J(\theta) = 2(\theta_1 - 5)$

$\frac{\partial}{\partial \theta_2} J(\theta) = 2(\theta_2 - 5)$

→ for $\min_{\theta} J(\theta)$

$\theta_1 = 5, \theta_2 = 5$

Week 3, LRM → video → Ifdu opt. for coding

--(a) Multiclass classification : one vs all — —

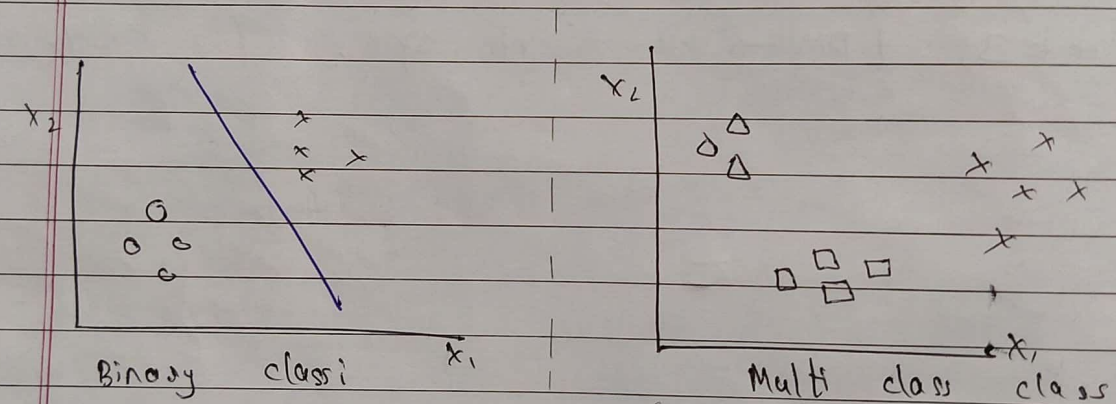
* logistic regon : Multi class classification :
one -vs -all

Multiclass classification :

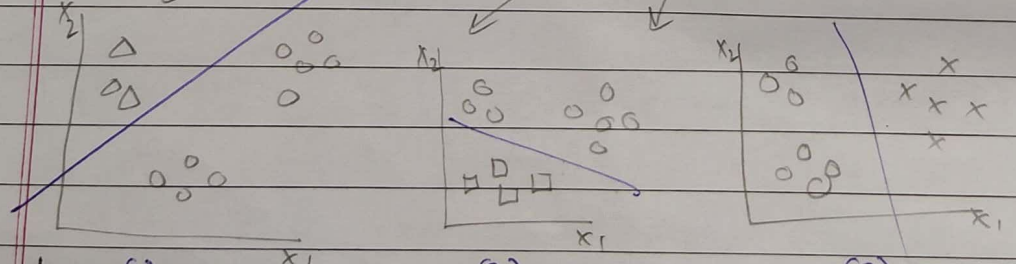
Email folderin tagging : Work, friend, family, Hobby
$y=1$ $y=2$ $y=3$ $y=4$

Disease cold flu Not ill
$y=1$ $y=2$ $y=3$

Whether sunny, cloudy, Rain snow
not matter 1 2 3
1 2 3 4



Binary classi Multi class class



one-vs-rest $h_\theta^{(1)}(x)$ $h_\theta^{(2)}(x)$ $h_\theta^{(3)}(x)$
$P(y=1|x;\theta)$ $P(y=2|x;\theta)$ $P(y=3|x;\theta)$

$$h_\theta^{(i)}(x) = P(y=i|x;\theta) \quad (i=1,2,3)$$

Que k classes ($y \in \{1, 2, \dots, k\}$) using 1 vs all. How many different logistic regn classifiers ~~so~~ will you end up training? ⟶ k.

($i = 1, 2, 3$) we trained 3.

- One- vs- all.

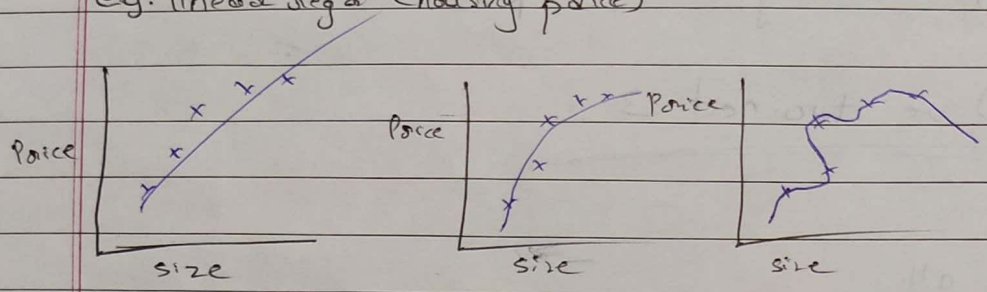
Train a logistic regn classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$

On a new input x, to make a prediction, pick the class i that maximises

$$\max_i h_\theta^{(i)}(x)$$

☆ Regularization : the problem of overfitting

eg. linear regn (housing price)



$\theta_0 + \theta_1 x$ | $\theta_0 + \theta_1 x + \theta_2 x^2$ | $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

'underfit' | 'Just right' | 'overfit'

'High bias' | | 'High variance'.

overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$) but fail to generalize to new examples (not able to predict price of new examples)

- **Addressing overfitting:**

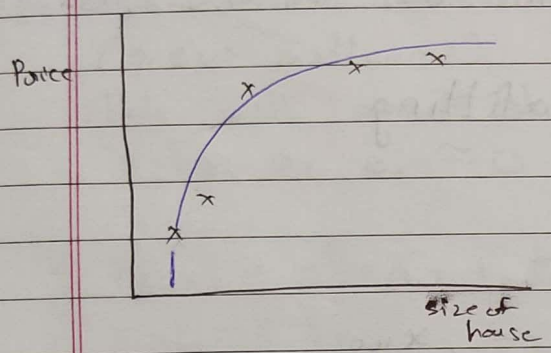
options:

1) Reduce number of features
   - manually select which feature to keep.
   - Model selection algorithm (later in course)

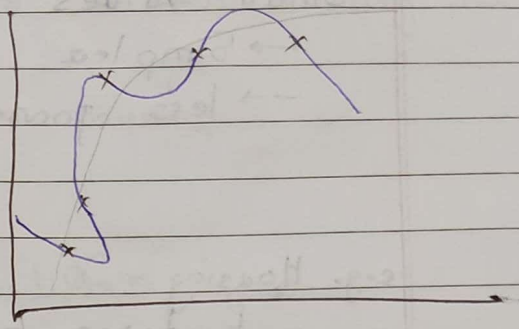
2) Regularization
   - keep all the features, but reduce magnitude/values of parameters θ_j
   - Works well when we have a lot of features, each of which contributes a bit to predicting y.

--@ Cost function ---

- Intuition of regularization :-



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



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Suppose we make penalize and make θ_3, θ_4 really small

→ This can make $\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 \longrightarrow \theta_0 + \theta_1 x + \theta_2 x^2$

→ ∩ → ⌒

Isn't it is like removing x^3, x^4 terms or removing some features. Automatically / manually latter maybe.

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \underbrace{1000\,\theta_3^2 + 1000\,\theta_4^2}_{\text{new terms added}}$$

$\theta_3 \approx 0 \quad \theta_4 \approx 0$

Minimizing this cost fun will surely make θ_3, θ_4

→ 1000, 1000 are just randomly big numbers, as we want to reflect cost caused by θ_3, θ_4, so we choosed big numbers.

· Regularization.

Small values for parameters $\theta_0, \theta_1, \dots, \theta_n$ all θ_j
→ Simpler hypothesis (smoother curve)
→ less prone to overfitting

e.g. Housing.
- Features $x_1, x_2, x_3, \dots, x_{100}$
- Parameters $\theta_0, \theta_1, \dots, \theta_{100}$

· Here we don't know which parameter is to shrink
$\theta_1 \to 0$ / $\theta_2 \to 0$ / $\theta_3 \to 0$ we don't know which $\theta_j \to 0$

· So we shrink all θ_s by modifying cost fn of linear regn.

Regularized linear regn.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2$$

(Regularization parameter; Regularization term)

→ This will shrink all $\theta_s \to \cancel{\theta_0}, \theta_1, \theta_2, \theta_3, \dots, \theta_{100}$

In practice Including θ_0 / not including does not makes any difference in results.

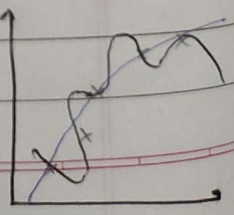
→ This is what we usually do

λ :- Regularization parameter. → 2 goals

Fit training set

shrink parameters

λ controls trade off b/w two goals.



que ~~Regularization~~. $J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1} \theta_j^2\right]$ If $\lambda = 10^{10}$

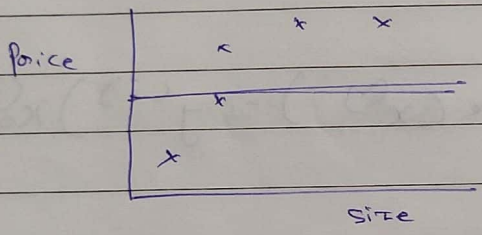
Ans Algo results in underfitting → fails to fit even training set.

This will result shrinking too much,

$\theta_1, \theta_2, \theta_3, \theta_4 \approx 0$ all

$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x + \theta_3 x + \theta_4 x$

$h_\theta(x) = \theta_0$



'underfit'

Lattice → choosing λ automatically in multiselection

— (a) Regularized linear regression — —

★ Regularized linear regⁿ.

• Gradient descent before (for linear regⁿ)

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad --(j=0,1,2,\dots,n)$$

}

Just writing θ_0 saparately. // no change.

updated for regularization.

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)} \rightarrow 1$$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m}\theta_j \quad (j=1,2,\dots,n)$$

}

We penalize $\theta_1, \theta_2, \theta_3, \dots, \theta_n$ only in regularization not θ_0

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

$$1 - \alpha \frac{\lambda}{m} < 1$$

It's like $\theta_j \times 0.99$

- Normal eqⁿ updated for regularization.

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}_{m \times (n+1)} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}_{\mathbb{R}^m}$$

$$\min_{\theta} J(\theta) \qquad \frac{\partial}{\partial \theta_j} J(\theta) \overset{\text{set}}{=} 0$$

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{(n+1) \times (n+1)} \right)^{-1} X^T y$$

e.g. $n = 2$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Non-invertibility (optional / advanced)

suppose $m \leq n$

(#examples) $\leq$ (#features)

$$\theta = (X^T X)^{-1} X^T y$$

non-invertible singular.

If $\lambda > 0$

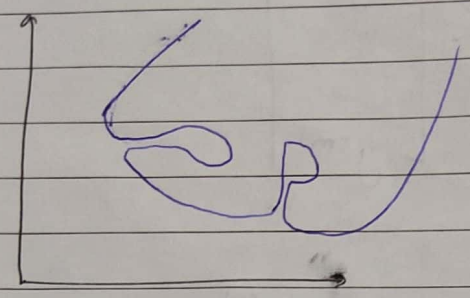
This is invertible matrix anyway.

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & \\ & 1 & \\ & & 1 \end{bmatrix} \right)^{-1} X^T y$$

--@ Regularized logistic regression ----

✱ Regularized logistic regⁿ.

- Regularized logistic regⁿ



sigmoid fⁿ

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^1 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

- cost fⁿ :-

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

$\theta_1, \theta_2, \dots \theta_n$

- Gradient descent.

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\, x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\, x_j^{(i)} + \frac{\lambda}{m}\theta_j\right] \quad j = 1, 2, 3, \dots$$

$\left[\frac{\partial}{\partial \theta_j} J(\theta)\right]$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

}

- Plot $-\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$ should decrease to work properly.