Logistic Regression

- is a Supervised machine learning classification model use to predict the probability of a class (usally a binary outcome like yes/no, spam/not spam, disease/no disease, churn/not churn).

- even the name say regression it is acctually a classification, not prediction of continuous values.

- This converts values into probability between 0 & 1

When NOT to use

- The relationship is non-linear (deep patterns, image classification, NLP -> use NN, CNN)
- Data has too many features without regularization (Suffers from Overfitting)
- We need class prediction with complex decision boundaries.

  (KNN, Trees, SVM, Random Forest may work better)

The Cost Function $J(\theta)$

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

Instead of MSE like linear Regression), logistic regression use Log loss $J(\theta)$

- This ensures model penalizes wrong confident predictions heavily.

Note ⇒ This is also called **Binary Cross_ Entropy Loss** or (Log loss or Negative Log-likelihood).

- Going forward every formula we drive will be 1st & 2nd derivative of this Cost function $J(\theta)$

Now let see the derivative.

data $(x^{(i)}, y^{(i)})$ for $i = 1, \cdots$

$x^{(i)} \in \mathbb{R}^n$, $y^{(i)} \in 0, 1$

- parameters (weights): $\theta \in \mathbb{R}^n$

our logistic regression model :→

$$y^{(i)} = h_\theta(x^{(i)}) = \sigma(\theta^T x^{(i)})$$

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

interpetation:

$$P(y^{(i)} = 1 \mid x^{(i)}; \theta) = h_\theta(x^{(i)})$$

$$P(y^{(i)} = 0 \mid x^{(i)}; \theta) = 1 - h_\theta(x^{(1)})$$

1. Likelihood & Cost function.

$$P(y^{(i)} \mid x^{(i)}; \theta) = \begin{cases} h_\theta(x^{(i)}) & y^{(i)} = 1 \\ 1 - h_\theta(x^{(i)}) & y^{(i)} = 0 \end{cases}$$

we can combine this to

$$P(y^{(i)} \mid x^{(i)}; \theta) =$$

$$\left[h_\theta(x^{(i)})\right]^{y^{(i)}} \cdot \left[1 - h_\theta(x^{(i)})\right]^{1-y^{(i)}}$$

For all m sample (assuming independence)

$$\mathcal{L}(\theta) = \prod_{i=1}^{m} P(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \underline{\left[h_\theta(x^{(i)})\right]^{y^{(i)}} \cdot \left[1 - h_\theta(x^{(i)})\right]^{1-y^{(i)}}}$$

$\prod$ = Product

$\sum$ = Sum

to convert the $\prod$ to $\sum$ we have to use log

-> we usually maximize log-likelihood:

$$\underbrace{\left[h_\theta(x^{(i)})\right]}\ ^{y^{(i)}} \cdot \underbrace{\left[1 - h_\theta(x^{(i)})\right]}\ ^{1-y^{(i)}}$$

$$\ell(\theta) = \log \mathcal{L}(\theta) =$$

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

- In ML style, we define cost function as negative average log-likelihood:

means we will multiply $-\frac{1}{m}$ in the function

$$J(\theta) =$$

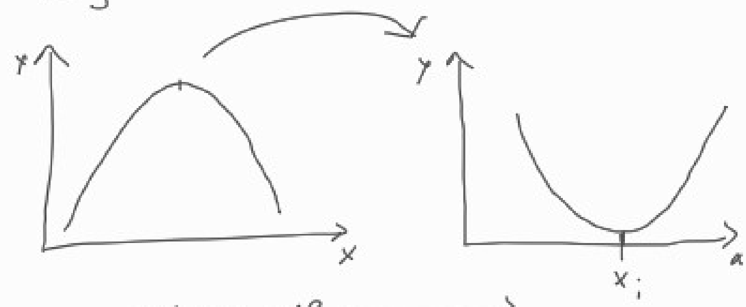
$$-\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log (1-h_\theta(x^{(i)}))\right]$$

why we put $-\frac{1}{m}$

their are 2 reason.

1. The -ve sign make it minimization proble

if you remember in Linear Regression we were reducing the cost function gradually but now the log-likelihood is a maximizing function



by putting -ve ⟶

so just by putting -ve we are converting maximizing problem to minimizing problem making the math simpler.

→ -ve ⇒ Make minimizing equivalent to maximum-Likelihood.

2. The $1/m$ makes the cost scale-independent (average)

. if we don't divide by m, the loss would grow simply because we have more data - not because the model is better or worse.

Example:-

with 10 sample cost $\approx$ 50

with 10000 identical sample cost $\approx$ 50000 ← missleading

→ By dividing with $1/m$

we take the mean loss per sample, so:

- the cost remains consistent no matter how many sample we have.
- The gradient become stable and easier to tune with learning rate.
- Training behave predictable across data size.

$\frac{1}{M}$ ⇒ Make the Cost an average loss per sample.

2. Derive Gradient $\nabla_\theta J(\theta)$

we will differentiate $J(\theta)$ w.r.t each component θ_j.

- 1st write for a single-sample loss:

$$J^{(i)}(\theta) =$$

$$\underbrace{-\left[y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right]}$$

then with multiple Sample it will be come

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} J^{(i)}(\theta)$$

- The derivative will be.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial J^{(i)}(\theta)}{\partial \theta_j}$$

So now let focus on one sample $J^{(i)}$

2.1 Derivative of Single-Sample loss

Recall

$$h_\theta(x^{(i)}) = \sigma(\theta^T x^{(i)}) = \sigma(z^{(i)})$$

$$z^{(i)} = \theta^T x^{(i)} = \sum_{k=1}^{n} \theta_k x_k^{(i)}$$

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

using the chain rule

$$\frac{d}{du}\log(u) = \frac{1}{u} \qquad \frac{d}{dx}\log(f(x)) = \frac{1}{f(x)}\frac{d}{dx}(f(x))$$

$$= \frac{f'(x)}{f(x)}$$

if $h = h_\theta$

$$\frac{\partial}{\partial\theta_j}(y\log(h)) = y \cdot \frac{1}{h} \cdot \frac{\partial h}{\partial\theta_j} = \frac{y}{h}\frac{\partial h}{\partial\theta_j}$$

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

Simplify signs:

$$\frac{\partial J^{(i)}}{\partial\theta_j} = \underline{-\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j}\left[\frac{y^{(i)}}{h_\theta(x^{(i)})} - \frac{1-y^{(i)}}{1-h_\theta(x^{(i)})}\right]}$$

Now we need $\dfrac{\partial h_\theta(x^{(i)})}{\partial \theta_j}$

2.2 Derivative of sigmoid

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

we know:

$$\frac{d\sigma(z)}{dz} = \sigma(z)\,(1-\sigma(z))$$

So

$$\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} = \frac{d\sigma(z^{(i)})}{dz^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial \theta_j} =$$

$$= \sigma(z^{(i)})\,(1-\sigma(z^{(i)})) \cdot x_j^{(i)}$$

$$= h_\theta(x^{(i)})\,(1-h_\theta(x^{(i)}))\, x_j^{(i)}$$

2.3 plug back

$$\frac{\partial J^{(i)}}{\partial \theta_j} = -h(x^{(i)})(1-h_\theta(x^{(i)}))\,x_j^{(i)}\underbrace{\left[\frac{y^{(i)}}{h(x^{(i)})} - \frac{1-y^{(i)}}{1-h(x^{(i)})}\right]}$$

now simplify inside the bracket:

$$\frac{y^{(i)}}{h(x^{(i)})} - \frac{1-y^{(i)}}{1-h(x^{(i)})} = \frac{y^{(i)}(1-h) - h(1-y^{(i)})}{h(1-h)}$$

$$= \frac{y^{(i)} - \cancel{y^{(i)}h} - h + \cancel{y^{(i)}h}}{h - h^2}$$

$$\frac{y^i}{h} - \frac{1-y^i}{1-h} = \boxed{\frac{y^{(i)} - h}{h(1-h)}}$$

Now pluging this back

$$\frac{\partial J^{(i)}}{\partial \theta_j} = -\cancel{h(x^{(i)})(1-h(x^{(i)}))}\,x_j^{(i)} \cdot \frac{y^{(i)} - h(x^{(i)})}{\cancel{h(x^{(i)})(1-h(x^{(i)}))}}$$

$$= -x_j^{(i)}(y^{(i)} - h(x^{(i)}))$$

$$\frac{\partial J^{(i)}}{\partial \theta_j} = -(y^{(i)} - h(x^{(i)}))x_j^{(i)} = \underline{(h(x^{(i)}) - y^{(i)})x_j^{(i)}}$$

2.A Average over all sample.

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

vector form

let

$X \in \mathbb{R}^{m \times n}$ (rows = Sample)

$\hat{y} = h(x) = \sigma(X\theta) \in \mathbb{R}^m$

$y \in \mathbb{R}^m$

then

$$\boxed{\nabla_\theta J(\theta) = \frac{1}{m} X^T (\hat{y} - y)}$$

This is the core gradient formula used in

- Batch gradient decent/accent
- mini-batch/SGD
- Newton's method / IRLS as the first order ter.

3. Update the Cost function

Now in this step we need to 1st decide which cost function is best suited for us or our model.

- we have

1. Batch gradient decent / accent.
2. Mini-batch & SGD
3. Newton's method / IRLS as the 1st order $\frac{d}{dx}$

I will only look into

3.1 Batch gradient decent/accent

→ in gradient method we add α learning rate then $\pm$ with calculated gradient

General formula

$$\theta := \theta \pm \alpha \nabla_\theta J(\theta)$$

$$= \theta \pm \alpha \left[\frac{1}{m} X^T (y - \hat{y}) \right.$$

+ → accent

− → decent.

3.2 Newton's Method

(faster, uses Hessian)

input sample

$X_{m \times n}$, $X^T_{n \times m}$

$$\theta = \theta - H^{-1} \nabla J(\theta)$$

$W_{m \times m}$ = diagonal matrix

where

$$H = \frac{1}{m} X^T W X \quad , \nabla J(\theta) = \frac{1}{m} X^T(\bar{y} - y)$$

Let understan hessian

for now let just forget $\frac{1}{m}$ cause it just average out to a indivisual sample.

so $X^T W X$

let say

Hassen

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad W = \begin{bmatrix} \omega_1 & 0 & 0 \\ 0 & \omega_2 & 0 \\ 0 & 0 & \omega_3 \end{bmatrix}$$

$$\omega_i = \bar{y}^{(i)}(1 - y^i) \qquad \bar{y}^{(i)} = h_\theta(x^{(i)})$$

Then $x^T W X =$

$$\begin{bmatrix} x^{(1)} & x^{(2)} & x^{(3)} \end{bmatrix} \begin{bmatrix} \omega^{(1)} & 0 & 0 \\ 0 & \omega^{(2)} & 0 \\ 0 & 0 & \omega^{(3)} \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}$$

$$\begin{bmatrix} x^{(1)} & x^{(2)} & x^{(3)} \end{bmatrix} \begin{bmatrix} \omega^{(1)} x^{(1)} \\ \omega^{(2)} x^{(2)} \\ \omega^{(3)} x^{(3)} \end{bmatrix}$$

$$x^T(DX) = x^{(1)}(\omega^{(1)} x^{(1)}) + x^{(2)}(\omega^{(2)} x^{(2)}) + x^{(3)}(\omega^{(3)} x^{(3)})$$

$$= \omega^{(1)}(x^{(1)})^2 + \omega^{(2)}(x^{(2)})^2 + \omega^{(3)}(x^{(3)})^2$$

$$x^T D x = \sum \omega^{(i)}(x^{(i)})^2$$

$$H = \frac{1}{m}\left[\sum \omega^{(i)} (x^{(i)})^2\right]$$

so this when we have only one feature in our data set now for multiple feature.

- Let more the i down for this

$$\left[\sum w^{(i)}\left(x^{(i)}\right)^2\right] = \sum w_i x_i^2$$

so

$$x^T W x = \begin{bmatrix} \sum w_i x_{i1}^2 & \sum w_i x_{i2}^2 & \cdots & \sum w_m x_{mn}^2 \\ \sum w_i x_{i2} x_{i1} & \sum w_i x_{i2}^2 & \cdots & \sum w_m x \\ \vdots & & & \\ \sum w_m x_{mn} x_{mn}^2 & & & \sum w_m x_m^2 \end{bmatrix}$$

which boil down to

$$H = \frac{1}{m} \sum_{i=1}^{m} w_i x_{ij} x_{ik} \qquad w = \bar{y}(1-\bar{y})$$

j & k = 1 to m

So finally let take a step back and recal

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

where

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

The First derivative $\rightarrow$ Gradient

$$\nabla_\theta J(\theta) = \frac{1}{m}X^T(\bar{y} - y)$$

$$\bar{y} = \sigma(X\theta)$$

& used in gradient decent/accent

$$\theta = \theta \pm \alpha\frac{1}{m}X^T(\bar{y} - y)$$

Taking 2^{nd} derivative $\rightarrow$ Hessian

$$H(\theta) = \nabla_\theta^2 J(\theta) = \frac{1}{m}X^T W(\theta)X$$

where $W(\theta) = \mathrm{diag}(\bar{y}_i(1-\bar{y}_i))$

Newton's Method Update Rule

$$\theta = \theta - H(\theta)^{-1} \nabla_\theta J(\theta)$$

$\Rightarrow$

$$\theta = \theta - \left(\frac{1}{m} X^T W X\right)^{-1} \left(\frac{1}{m} X^T (\bar{y} - y)\right)$$

$$= \theta - \cancel{m} \left(X^T W X\right)^{-1} \frac{1}{\cancel{m}} \left(X^T (\bar{y} - y)\right)$$

$$\boxed{\theta = \theta - \underbrace{\left(X^T W X\right)^{-1}}_{\underline{A}} \cdot \underbrace{X^T (\bar{y} - y)}_{\underline{g}}}$$

$\rightarrow$ gradient

Math Example:→

Manually showing every arithmetic step by using Newton's Method.

Data set $m = 6$, $d = 3$ with intercept.

for row $i = 1 \ldots 6$, each $x_i = [1, x_{i1}, x_{i2}]$

$$\begin{array}{cccc} x_0 & x_1 & x_2 & \quad y \\ \end{array}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & 3 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & 2 \\ 1 & 6 & 5 \\ 1 & 7 & 8 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

x_0 = intercepter

x_1, x_2 = feature

y = result.

- we run Newton's method with initial $\theta^{(0)} = [0,0,0]^T$
  - $z_i = \theta^T x_i$
  - $h_i = \sigma(z_i) = 1/(1+e^{-z_i}) = \bar{y}$
  - $w_i = h_i(1-h_i)$ → for the diagonals.
  - Gradient: $g = X^T(y - \bar{y})$
  - Hessian: $\nabla^2 \ell(\theta) = -X^T W X$, we form $A = X^T W X$
  - Solve $A \Delta\theta = g$

· update $\theta \leftarrow \theta + \Delta\theta$

Iteration 1 (initial $\theta^0 = [0,0,0]$)

Step-1 - z & h

Since $\theta^{(0)} = 0$,

$z = \sum \theta X$

$z_1 = 0 \times 1 + 0 \times 2 + 0 \times 1 = 0$

$z_2, z_3, z_4, z_5, z_6 = 0$

Now h

$h_i = \sigma(z_i) = \sigma(0) = \frac{1}{1 + e^{-0}} = \frac{1}{1+1} = \frac{1}{2}$

$h_i = \quad = 0.5$

Step-2 ω

$\omega_i = h_i(1 - h_i) = 0.5 \times 0.5 = 0.25$

$(y - h) = [0 - 0.5, 0 - 0.5, 0 - 0.5, 0 - 0.5, 0 - 1, 0 - 1] = [-0.5, -0.5, -0.5, -0.5, 0.5, 0.5]$

step-3 - gradient $g = x^T(y-h) = \nabla_\theta J(\theta)$

$$g_0 = \sum_i \overset{x_0}{1} \cdot (y_i - h_i) = \sum (y-h)$$

$$= (-0.5) + (-0.5) + (-0.5) + (-0.5)$$
$$+ 0.5 + 0.5 = -1.0$$

$$g_1 = \sum_i x_{i1} (y_i - h_i)$$

$$\Rightarrow \quad 2 \cdot (-0.5) = -1.0$$
$$3 \cdot (-0.5) = -1.5$$
$$2 \cdot (-0.5) = -1.0$$
$$3 (-0.5) = -1.5$$
$$6 (0.5) = 3.0$$
$$7 (0.5) = \underline{3.5}$$

$$\text{Sum} = 1.5$$

$$g_2 = 1 \cdot (-0.5) + 1(-0.5) + 2(0.5) + 2(0.5) + 5(0.5) + 8(0.5)$$
$$= 3.5$$

So $\quad g = \nabla_\theta J(\theta) = \begin{bmatrix} -1.0 \\ 1.5 \\ 3.5 \end{bmatrix}$

step - 4 $A = x^T W x$

for W, $\omega_i = h_i(1-h_i)$

$$W = \begin{bmatrix} 0.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.25 \end{bmatrix}$$

$$A_{ij} = \underbrace{\sum_i x_{ij}\, \omega_i\, x_{ik}} \quad \text{where } \omega_i = 0.25$$

$$A = x^T W x = \begin{bmatrix} 1.5 & 5.75 & 4.75 \\ 5.75 & 27.75 & 25.25 \\ 4.75 & 25.25 & 24.75 \end{bmatrix}$$

step 5 $A \Delta\theta = g \Rightarrow \Delta\theta = A^{-1} g$

$$1 \Big/ \begin{bmatrix} 1.5 & 5.75 & 4.75 \\ 5.75 & 27.75 & 25.25 \\ 4.75 & 25.25 & 24.75 \end{bmatrix} \begin{bmatrix} -1.0 \\ 1.5 \\ 3.5 \end{bmatrix} = \begin{bmatrix} -4.1142\cdots \\ 0.82857\cdots \\ 0.08571\cdots \end{bmatrix}$$

Step 6 update θ

$$\theta^{(1)} = \theta^{(0)} + \Delta\theta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -4.1142\cdots \\ 0.82857\cdots \\ 0.08571\cdots \end{bmatrix} = \begin{bmatrix} -4.1142\cdots \\ 0.82857\cdots \\ 0.08571\cdots \end{bmatrix}$$

Now we will repeat the 6 steps again and again