Logistic Regression

Logistic regression is used when the target variable is categorical.

ex: to predict email is spam (1) or ham (1).

to predict tumor is ~~cat~~ malignant (1) or benign (0).

If we use linear regression, there is a need for setting up a threshold based on which the classification could be done.

It can be observed that having a fixed threshold is not a good way to make predictions in real time.

Linear regression is unbounded and this brings logistic regression into picture, their value ranges from 0 to 1.

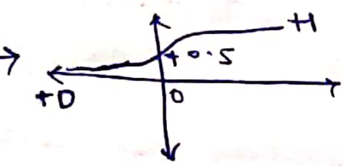
Model structure:

Output = 0 or 1

Hypothesis : $z = w \cdot x + B$

$h\theta(x) = \text{sigmoid}(z)$

sigmoid $\Rightarrow \dfrac{1}{e^{-x}+1} \rightarrow$



if $z \to \infty$, $\hat{y} = 1 \to c_1$

$z \to -\infty$, $\hat{y} = 0 \to c_2$

Predicted value.

we can see that we first fit into linear regression model, which is acted upon by an activation function | logistic function predicting the target categorical dependent variable.

Types of Logistic regression:

① Binary logistic regression – only 2 possible outcomes.

② Multinomial logistic regression – 3 or more categories without ordering. [vegan, non-vegan...]

③ ordinal logistic " – 3 or more categories with ordering [movie rating 1-10].

Decision Boundary:

To predict which class, a data belongs to, a threshold can be set. Based upon this threshold, the obtained estimated probability is classified into classes.

Note: Decision boundary can be linear | non-linear. The polynomial order can be increased to get complex decision boundary.

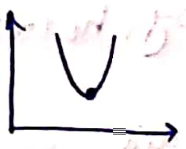
cost function:

$$\text{cost}(h_\theta(x), y_{actual}) = -\log(h_\theta(x)) \text{ if } y=1$$
$$= -\log(1-h_\theta(x)) \text{ if } y=0$$

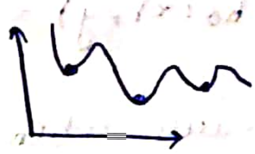
(Q) why can't we use the cost function used for Linear regression?

~~Ans~~ Linear regression uses MSE as its cost function. If this is used for Logistic regression, then it will be a non-convex function.

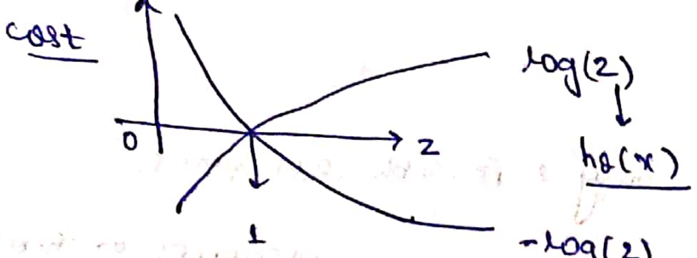
Gradient descent will converge into global min. only if the function is convex.



Convex



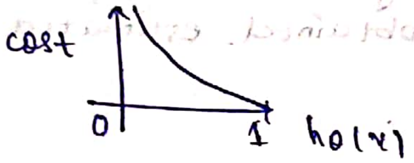
non-convex

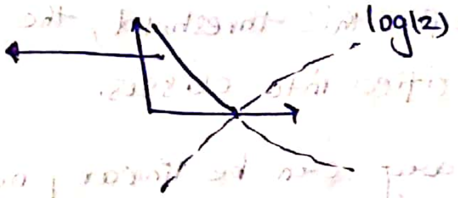


$$\text{cost}(h_\theta(x), y) = -\log(h_\theta(x)) \Rightarrow \text{if } y=1$$
$$= -\log(1-h_\theta(x)) \Rightarrow \text{if } y=0$$

---

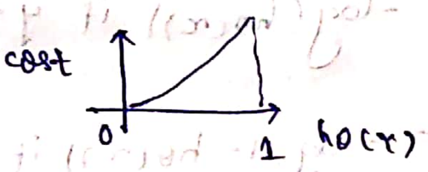
if $y = 1$: $\text{cost}(h_\theta(x), y) = -\log(h_\theta(x))$

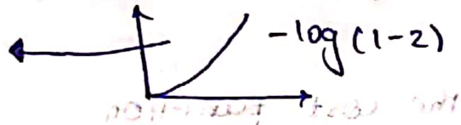




if cost $= 0 \Rightarrow y=1 \Rightarrow h_\theta(x) = 1$

cost $= \infty \Rightarrow h_\theta(x) = 0$





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

cost $= \infty \Rightarrow y \to 1 \Rightarrow h_\theta(x) = 1$

if $h_\theta(x) = 1$, it is similar to predicting $P(y=0 \mid x; \theta) = 0$.

---

simplified cost function:

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

Notes

we use negation because when we train, we need to maximize the prob. by min. the loss function.

Scanned by CamScanner

for classification into more than 2 classes, softmax activation function can be used.

Note: $\boxed{\text{cost vs Loss}}$

The terms cost and loss functions refer to the same meaning. Loss function mainly applies for a single training set as compared to the cost function which deals with a penalty for a number of training sets or a complete batch. Also known as error function.

so loss function is ~~calle~~ calculated many times in a single training cycle but the cost function is calculated once.

$$\boxed{\text{cost} = \Sigma \text{ loss}}$$

Logit function:

$$\log\left(\frac{P(x)}{1-P(x)}\right) = \beta_0 + \beta_1 . x$$

The odds specify the ratio of prob. of success to that of failure.

logit or odds or log-odd

∴ in logistic regression, linear combination of inputs are mapped to the log (odds).

$$\text{sig}(x) = \frac{1}{1+e^{-x}}$$

$$= \frac{1}{1+ 1/e^{x}}$$

$$P(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

$$\boxed{\text{sig}(x) = \frac{e^{x}}{1+e^{x}}}$$

→ sigmoid function → gives a S-shaped curve. It always give a value of prob. ranging from (0,1).

Scanned by CamScanner

Estimation of parameters:

Unlike linear regression that use OLS, we use MLE.

There can be infinite sets of regression coeff. The Max. Likelihood Estimate is that set of regression coeff. for which the output prob. is max.

$$L(\beta, y) = \prod_{i=1}^{N} \left(\frac{p_i}{(1-p_i)} \right)^{y_i} \times (1-p_i)$$

success

For simplicity we use log likelihood.

Performance:

Confusion matrix

		Predicted	
		P	N
actual	P	TP	FN
	N	FP	TN

Accuracy: $\dfrac{TP + TN}{TP + TN + FP + FN}$

Linear regression $\longrightarrow$ $y = mx + c$

slope. intercept.

input data points.

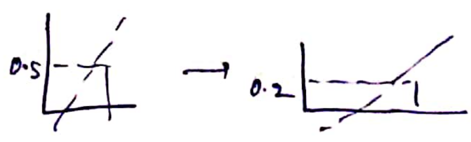
$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$= \beta_0 + \beta_1 x$$

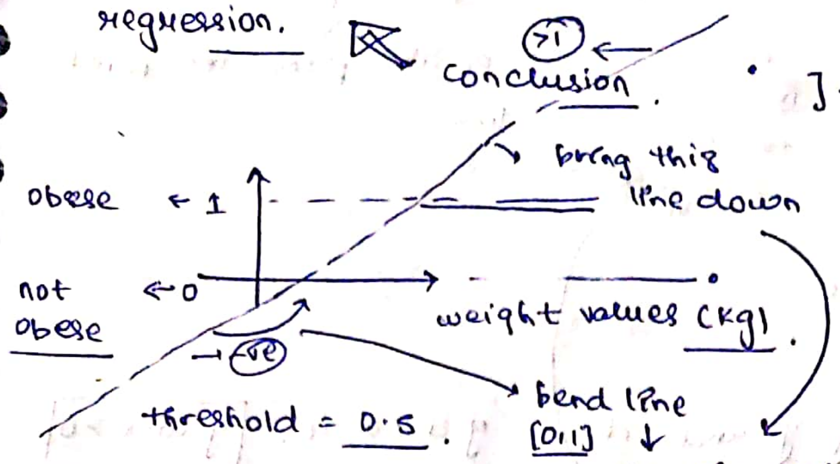
$$y = w^T x + \theta$$

Linear regression is susceptible to outliers.

This will lead to changing thresholds.



Hence, we can't make use of MSE or OLS and also linear regression.



conclusion.

•] → will always be classified as obese.

↓

line keeps growing.

↓

This fails miserably.

bend line [0,1] ↓ use sigmoid activation for the same. → squashed the straight line.

also used in deep learning too in the output dense layer.

$$\text{sigmoid}(x) = \frac{1}{1+e^{x}}$$

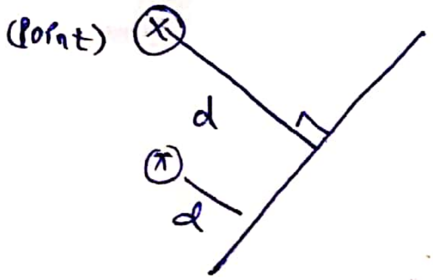
---

assumptions:

all points → +ve class or $c_1 \rightarrow +1$
" " → -ve " or $c_2 \rightarrow -1$] initialization

$y = w^Tx + b = 0$ → (passing through origin).

$\boxed{y = w^T.x}$ points are linearly separable.



plane (w values) (coeff. values)

$$d = \frac{w^Tx + b}{\|w\|}$$

if w = unit vector
$\|w\| = 1$

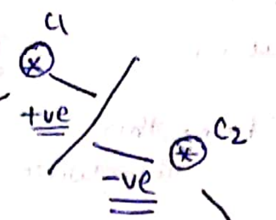
line passing through origin $\Rightarrow b = 0$

$\boxed{d = w^Tx}$

sum of distances.

$$\sum_{i=1}^{n} w_i^T x_i$$

all distances will be +ve or zero. or -ve. depending on the side of the plane the point lies.



① case-1:

$y = +1$

above the plane distance = +ve

$d = w^T x > 0$

multiply y and d $y \cdot d > 0$ ie. $\boxed{y \cdot w^T x > 0}$

y_i → properly classified.
↓
correct.

② case-2:

$y = -1$

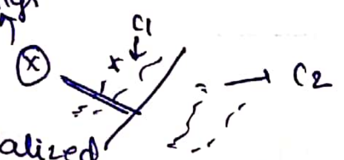
$d = w^T x < 0$

multiply y and d.   -ve × -ve = +ve value

$\boxed{y \cdot w^T \cdot x > 0}$
↓
correctly classified

③ case-3:

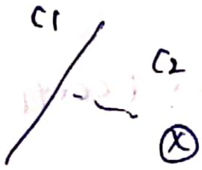
belongs to c_2
↓
initialized as -1.



$y = -1$ $d = w^T x > 0$ (above the plane).

multiply $\boxed{y \cdot w^T x < 0}$ → incorrect classification.

④ case-4:



ⓧ belongs to c_1

$y = +1$

$d = -ve$

$\boxed{y \cdot d < 0}$ → incorrect classification.

cost function or optimizer : $\max \left\{ \sum_{i=1}^{N} y_i \cdot w_i^T \cdot x_i \right\}$ → +ve → correct classif.
−ve → incorrect classif.

y_i — given

w_i^T — compute these coefficients. ← update these coeff. unless I get max. value of cost function.

x_i — given

Note: An outlier can change the entire game

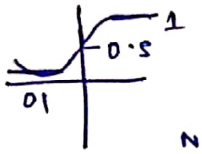
Solution:

let ~~z = wTx~~ $z = y \cdot w^T x$

$\max \left\{ \text{sigmoid} \left(y_i \times w_i^T \times x_i \right) \right\}$

↑ operator

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

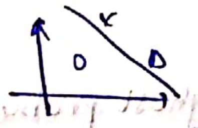


after value calculation → value is transformed b/w 0 ~~-1~~ to +1.

Note: Ill-effect of an outlier is nullified.

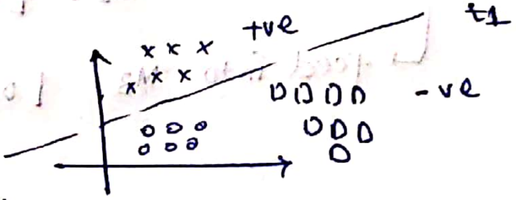
Logistic Regression (One vs Rest) or (One vs All)

Solving multiclass classification.

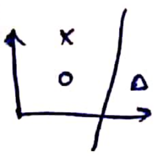


iteration 2

M2 model



iteration 1 — M1 model



iteration 3 → M3 Model

take one out and group rest all as 1.

Note: take combinations.

f1	f2	f3	o/p	O1	O2	O3
I1	I2	I3	O1	+1	-1	-1
I4	I5	I6	O2	-1	+1	-1
I7	I8	I9	O3	-1	-1	+1
I10	I11	I12	O1	+1	-1	-1
I13	I14	I15	O2	-1	+1	-1

+ model M1 → O1 or not

+ Model M2 — O2 or not

+ Model M3 — O3 or not

New point

↳ feed into M1 : [0.20]

↳ feed into M2 : [0.25]] summation = 1

↳ feed into M3 : [0.55]

output Prob.

overall → [0.20, 0.25, 0.55] ← highest probability value.

max.val = 0.55 → ans = $\boxed{O3}$