ML Logistic Regression

Thursday, 29 February 2024 09:59

Logistic Regression Machine Learning Example | Simply Explained
Coding Lane

Note: Python versions of formulas can sometimes use vectors more easily than the math notation allows.
Example: NumPy logarithms can take and output a vector

Logistic Regression:

Some notations use z

helps to avoid overflow and divide by zero errors in the computer

Sigmoid: $y = \sigma(\sigma) = \frac{1}{1+e^{-\sigma}} = \frac{e^{\sigma}}{e^{\sigma}+1}$ "Probability function" (gives output between 0 and 1 then set to $= \begin{cases} 1, & \text{if } p \geq 0.5 \\ 0 & \text{if } p < 0.5 \end{cases}$)

$= \frac{1}{1+e^{-(w_1x_1+c_1)}}$ — weights; feature of the dataset (1)

or $= \frac{1}{1+e^{-(w_1x_1+\dots+w_nx_n+b)}}$ — weights; features of the dataset (n)

- $\sigma = W^TX + b$ — dot product (does not always give scalar); Vector

Single observation Vector $= \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} X + b$ — its size can be indicated using a subscript: $\begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}_{(n,1)}$

this is what we are adjusting when we are doing machine learning

OR

Matrix of all observations $= \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} [x_1, \dots x_n] + b$

= Salar (X single vector) or Vector (X matrix of all observations)

we want to MINIMISE the value of the cost function

Cost function (error representation):

Scalar quantity that takes into account all observations

$$Cost = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(\hat{y}_i) + (1-y)\log(1-\hat{y}_i)\right]$$

with $\hat{y}_i = \sigma(W^TX_i + b)$ as the predicted output value for the i-th observation and y_i as the target value or true class (0 or 1) for the i-th observation and m the number of observations.

- Cost or error $= |\underbrace{\hat{y}}_{\text{predicted output}} - \underbrace{y}_{\text{actual output/target output}}|$ (for single observation ∴ one credit card transaction)
- Average Cost $= \frac{1}{m}\sum_{i=0}^{n}|\hat{y}_i - y_i|$ (average of multiple (m) observations)

(works if predictions/outputs are continuous but now our outputs are just 0 or 1 so when uninimising the function, we potentially reach the local minima instead of global minima.) — Ex. in linear regression

Gradient Decent for Cost Minimisation:

we use $\frac{\partial Cost}{\partial w}$ and $\frac{\partial Cost}{\partial b}$

"Loss" or "Error"

- Let $L = -\left[\underbrace{y}_{\text{Scalars}} \log(\hat{y}) + (1-y)\log(1-\hat{y})\right]$ (for a single observation/data point ∴ one credit card transaction NOT the average of multiple observations but there can still be multiple weights)

for a EACH $w \in W$:

- $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w}$ (We use the chain rule)

$= \frac{\partial L}{\partial \hat{y}} \times \left(\frac{\partial \hat{y}}{\partial \sigma} \times \frac{\partial \sigma}{\partial w}\right)$

$= \left(-\frac{y}{\hat{y}} + \frac{(1-y)}{(1-\hat{y})}\right)\left(\hat{y}(1-\hat{y})\right)(x)$ (steps were skipped)

Example: $\frac{\partial \sigma}{\partial w_k} = \frac{\partial}{\partial w}(W^TX + b) = \frac{\partial}{\partial w}(w_1x_1 + \dots + w_kx_k + \dots + w_nx_n + b) = x_k$

$\overset{\text{Simplify}}{=} (\hat{y} - y)\cdot x$

The math for a single observation

$\therefore \frac{\partial L}{\partial w_i} = (\hat{y} - y)x_i$ for $i = 1, \dots, n$ = number of FEATURES

$\therefore \frac{\partial L}{\partial W} = (\hat{y} - y)\cdot X$ (dot product) (vector notation ∴ for all the features, and all the observations)

The math for all observations

- $\boxed{\therefore \frac{\partial Cost}{\partial W} = \frac{1}{m}(\hat{y} - y)\cdot X}$ gives vector

The math for a single observation

Now $\frac{\partial L}{\partial b} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial \sigma} \times \frac{\partial \sigma}{\partial b} = \hat{y} - y$

The math for all observations

- $\boxed{\therefore \frac{\partial Cost}{\partial b} = \frac{1}{m}\sum_{i=0}^{m}(\hat{y}_i - y_i)}$ gives scalar

Machine Learning:

Repeat {

Calculate predicted output (Vector of predicted values)
Calculate cost/loss (Scalar)
Calculate $\frac{\partial Cost}{\partial W}$ (Vector)
Calculate $\frac{\partial Cost}{\partial b}$ (Scalar)

All of these take all observations into account

$W = W - \alpha\left(\frac{\partial Cost}{\partial W}\right)$ — learning rate
$b = b - \alpha\left(\frac{\partial Cost}{\partial b}\right)$

this is where our parameters (weights and baias) get adjusted repeatedly in the Machine Learning process

}

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

with $z = w_1x_1 + \dots + w_ix_i + b$

$= \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} [x_1\ x_2 \dots x_n] + b$

$= W^TX + b$

where $w_1, w_2, \dots, w_n$ are scalar coeficients called weights and each $x_1, x_2, \dots, x_n$ corresponds to a feature in the dataset.

- $Cost = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(\hat{y}_i) + (1-y)\log(1-\hat{y}_i)\right]$

with $\hat{y}_i = \sigma(W^TX_i + b)$ as the predicted output value for the i-th observation and y_i as the target value or true class (0 or 1) for the i-th observation and m the number of observations.

- $\frac{\partial Cost}{\partial W} = \frac{1}{m}(\hat{y} - y)\cdot X$

$\frac{\partial Cost}{\partial b} = \frac{1}{m}\sum_{i=0}^{m}(\hat{y}_i - y_i)$

- Let $L = -\left[y \log(\hat{y}) + (1-y)\log(1-\hat{y})\right]$

for a EACH $w \in W$:

$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w}$ (applying the chain rule)

$= \frac{\partial L}{\partial \hat{y}} \times \left(\frac{\partial \hat{y}}{\partial \sigma} \times \frac{\partial \sigma}{\partial w}\right)$

$= \left(-\frac{y}{\hat{y}} + \frac{(1-y)}{(1-\hat{y})}\right)\left(\hat{y}(1-\hat{y})\right)(x)$ (steps were skipped)

$= (\hat{y} - y)\cdot x$

$\therefore \frac{\partial L}{\partial w_i} = (\hat{y} - y)x_i$ for $i = 1, \dots, n$ = number of features

for multiple observations we have:

$\therefore \frac{\partial L}{\partial W} = \frac{1}{m}(\hat{y} - y)\cdot X$

- $W = W - \alpha\left(\frac{\partial Cost}{\partial W}\right)$

$b = b - \alpha\left(\frac{\partial Cost}{\partial b}\right)$

with α some constant called the learning rate.