Project Report

Multinomial Logistic Regression

Logistic Regression is a kind of statistical modeling which is often used for classification and predictive analysis .This is a supervised machine learning technique which is also considered a discriminative model as it classifies between various classes. Logistic Regression uses Log Odds for it's calculation . Logistic Regression are of various types :-

1) Binary Logistic Regression – Used when there can be two outcomes wither 0 or 1 .
2) Multinomial Logistic Regression – Used when there are more than two target variables and they are generally accompanied by Softmax Activation Function .
3) Ordinal Logistic Regression – This is used when there are more than two targets or outcomes but they do not have a defined order .

Here in this report we will be mainly looking at Multinomial Logistic Regression , where we can have multiple outcomes and on using Softmax activation function we will be having probabilities in the range of 0 to 1.

Here we use Log Odds as the measuring parameter .**Log Odds** means calculating the probability of the **Event's success** over the **Event's failure.**

$$\log [p/(1-p)]$$

where ,

p →probability of an event happening

1-p → probability of the event **not** happening

When a function's variable represents a probability p , like the above example it is known as **Logit Function**.

Why we use Log Odds ?

Probability of any function or variable gives the result ranging from [0 , 1] . But in complex problems specially in those of Multinomial Logistic Regression where there can be multiple outcomes with **multiple predictors** , this range of [0 , 1] does not do justice to the **effects of all the predictors to the targets**.

Thus we use Log Odds as it ranges from $-\infty$ *to* ∞ . Log Odds can effectively capture all the positive and the negative effects of all the predictors towards the target outcomes.

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$$

The beta parameter or coefficient is calculated using Maximum Likelihood Estimation(MLE) .

The model tests with various values of Beta through multiple iterations to find the most **optimized value of beta** . Once the best and the most optimized value of all the beta coefficients are found (**more than one coefficient for multinomial logistic regression , as more than two independent variables**) , the conditional probabilities of each observation can be calculated , logged and summed up to yield the predicted probable class as the outcome .

According to the Problem Statement …..

The problem statement also aimed at doing Multinomial Logistic Regression but in a more generalized manner. It mentions the finding of **Deterministic Utility(V)** which is nothing but the aforementioned Log odds .

$$V1 = \beta 01 + \beta 1X1 + \beta 2X2$$

$$V2 = \beta 02 + \beta 1X1 + \beta 2X2$$

$$V3 = \beta 03 + \beta 1Sero + \beta 2Sero$$

Here ,

X1 , X2 and Sero are the Independent variables or the predictors whose values are going to affect the outcomes either in a positive or a negative manner.

There can be many more independent variables or predictors and with each independent variable the amount of coefficients will also increase proportionately .

Assumptions : *According to the problem statement in the code we have taken Sero to be a default independent variable which will be always present and as it's values are mostly seen zero thus it is thought to be like that . A variable which does not impact the prediction or classification in any way as it being 0 with it's coefficient , as seen in the sample data provided.*

The probabilities of each independent variable's contribution towards the final outcome is calculated by dividing the exponential of each individual Deterministic Utility to that of the sum of all the exponential of all the Deterministic Utilities .

$$P_1 = \frac{exp(V_1)}{exp(V_1) + exp(V_2) + exp(V_3)}$$

$$P_2 = \frac{\exp\,(V_2)}{\exp\,(V_1) + \exp\,(V_2) + \exp\,(V_3)}$$

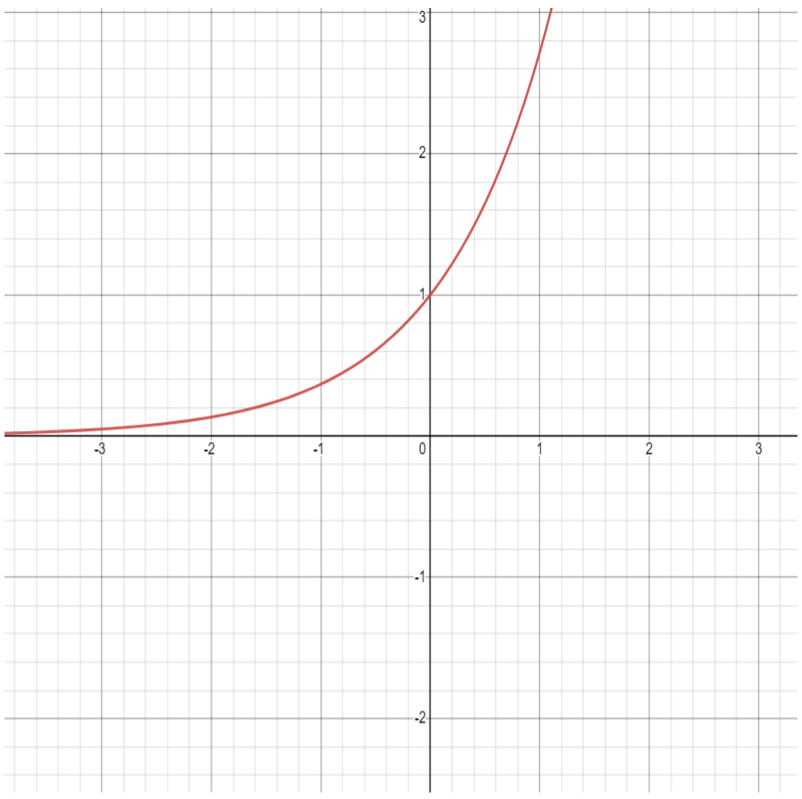
$$P_3 = \frac{exp(V_3)}{exp(V_1) + exp(V_2) + exp(V_3)}$$

Why take Exponential?

- **Normalization –** The data points are normalized to bring them into a particular shape or area in the graph instead of they being scattered.

- **Differentiable –** In various other concepts like that of Machine Learning which uses these accompanied by Cross Entropy loss , it is advised to make the curve differentiable for algorithms like gradient descent to function .

- **High Probability –** Exponential functions' main advantage lies in the fact that high scoring dependent variables is provided higher probability than the low scoring ones. This leads to easy segregation or classification of data.



Trick used for fast and efficient calculation

Vectorization – This is the most important aspect of the code used where we have used **Numpy** array to convert the list into arrays. In vectorization multiple multiplications and additions are done **parallelly** by ***matrix multiplication and dot product .***

The independent variables' array is made to do a dot product with the array of coefficients and the constant coefficients are summed to individual elements using the basic individual element addition and multiplication feature provided by Numpy arrays.

Similarly for the Sero variable the same kind of operation is conducted with it's set of coefficients .

Exception Handling has been incorporated in the program to avoid complete termination of the program due to input error with cautionary instructions being shown to the user ton avoid such exceptions .

The final probability for each data point in each independent variable is given in the form of a dictionary .

Enhancements : *Further to speed up the calculations we can segregate the process into various* ***Threads by importing the threading class in Python .*** *But threading is advised to be done in multicore systems and in higher levels the working of threads parallelly depends on the capability of the processor and the ease of managing threads directly in Operating System .*

System Configurations

Since System Configurations plays a very important roles in the efficient and fast calculation of the matrices ,thus providing the system used.

A very basic system is used so that it can give the minimal result or the worst possible result, thus the most accurate form of time complexity.

- *Processor : Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz   1.80 GHz*
- *Installed RAM : 8.00 GB (7.89 GB usable)*
- *System Type : 64-bit operating system, x64-based processor*
- *Graphics : No External Graphics used .*

IDE used : Spyder

Conclusion

This project shows the computation of Logistic Regression which uses the Logit function as the base. The Logit function is explained and why exponential is used is mentioned for the reader's proper understanding . Larger chunks of data can be efficiently calculated with better systems (preferably with external graphics using the CUDA toolkit which is specifically designed to undertake such huge calculations) . The time taken in this process to complete is 0.0010008811950683594 seconds .