Logistic Regression

Deepali Jain
Department of Computer Science and Engineering
University of Buffalo
Buffalo, NY 14212
deepalij@buffalo.edu

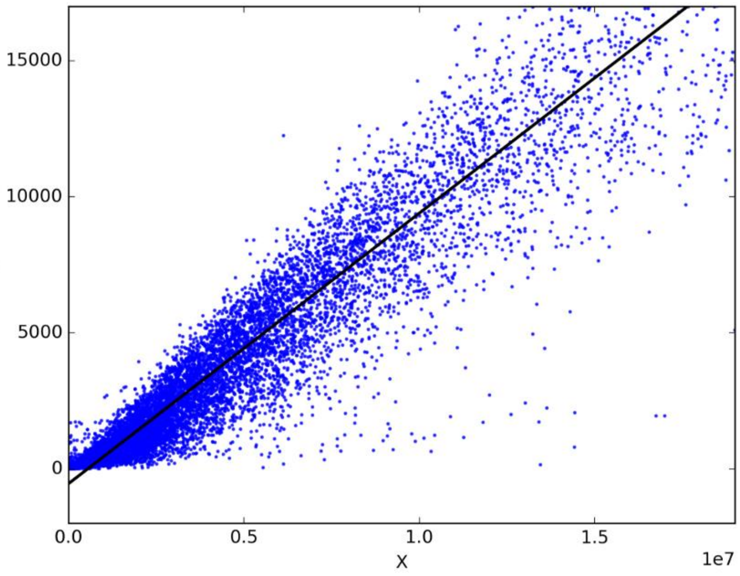
Abstract

This project performs classification using machine learning. It aims at the implementation of Linear Regression and Logistic Regression from the basic level. The dataset used is Wisconsin Diagnostic Breast Cancer (wdbc.dataset). Fine-needle aspiration (FNA) is a diagnostic procedure used to investigate lumps or masses. We use FNA to classify cells of breast mass as Benign (class 0) or Malignant (class 1) using logistic regression as the classifier.

1 Introduction

1.1 Linear Regression

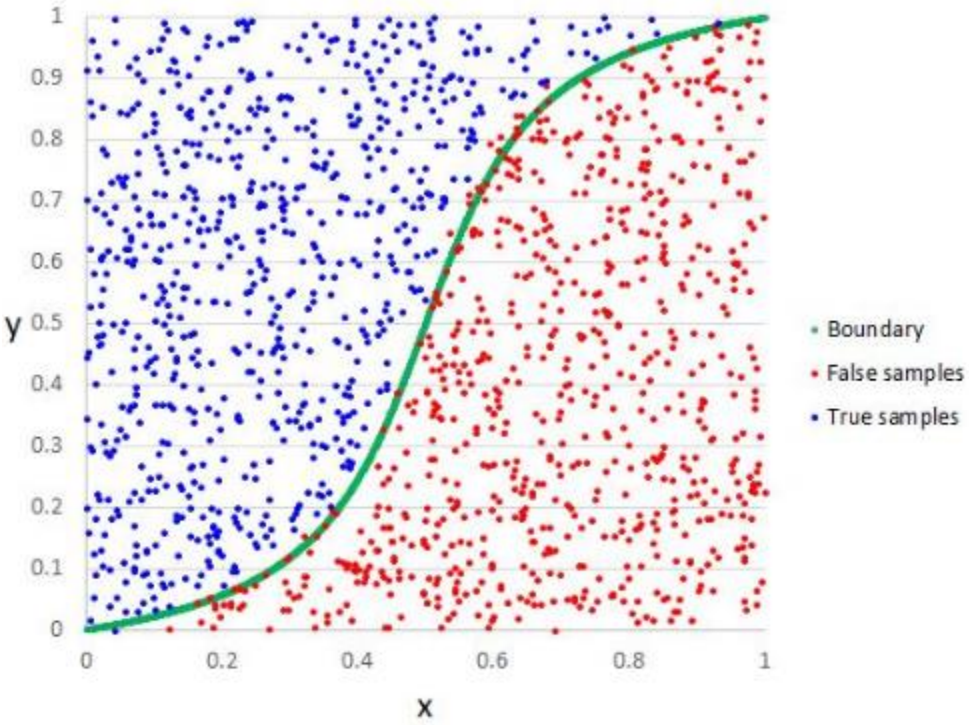
Linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable. [1]



[2] Figure: Linear Regression

1.2 Logistic Regression

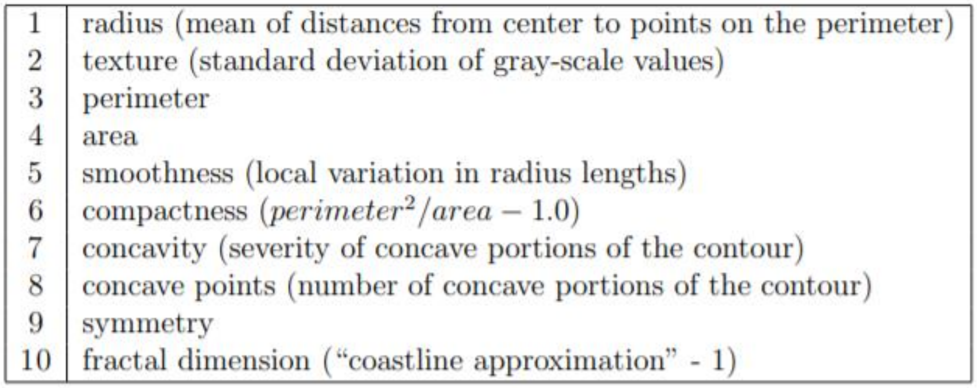
logistic model (or logit model) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc... Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. [3]



[4] Figure: Logistic Regression

2 Data Set

Wisconsin Diagnostic Breast Cancer (WDBC) dataset will be used for training, validation and testing. The dataset contains 569 instances with 32 attributes (ID, diagnosis (B/M), 30 real-valued input features). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Computed features describes the following characteristics of the cell nuclei present in the image:



1	radius (mean of distances from center to points on the perimeter)
2	texture (standard deviation of gray-scale values)
3	perimeter
4	area
5	smoothness (local variation in radius lengths)
6	compactness ($perimeter^2/area - 1.0$)
7	concavity (severity of concave portions of the contour)
8	concave points (number of concave portions of the contour)
9	symmetry
10	fractal dimension ("coastline approximation" - 1)

The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

3 Pre-processing

The data is collected in the raw form from the sources. This data is not suitable for feeding it to the machine learning algorithms. Therefore, different techniques are applied to clean the data. Some of them are Rescaling data, Binarizing data and Normalizing data.

3.1 Step 1: *Read data file (pandas)*

The first step includes reading the csv file through the library "pandas". For this, initially we need to import the pandas library.

```
import pandas as pd
```

Then we need to use command ***read_csv*** to read the data file and load

```
data = pd.read_csv("filename.csv")
```

3.2 Step 2: *Process the data file*

3.2.1 Drop column Id

3.2.2 Map label column (target values) to 0 and 1(numbers)

3.3 Step 3: *Splitting the data frame*

i) Training data (80%)

ii) Validation data (10%)

iii) Testing data (10%)

3.4 Step 4: *Normalization*

Normalization of data means reorganizing the data values to a common scale. Here normalization was done to bring the values in the range between 0 and 1.

3.5 Step 5: *Initialization of weights, biases and learning rate*

3.6 Step 6: *Update values of weights after every iteration*

```
for epoch in Range (10000)
    $z = \theta^{T}X + b$
    $a = \sigma(z)$
    $L = -\frac{(y \log(a) + (1-y) \log (1-a))}{m}$
       $= -\frac{1}{m} (y \log(a) + (1-y) \log (1-a))$
       $= -\frac{1}{m} (y \log(\sigma(z)) + (1-y) \log (1- \sigma(z)))$
```

$\theta = \theta - \eta \Delta \theta$

$b = b - \eta \Delta b$

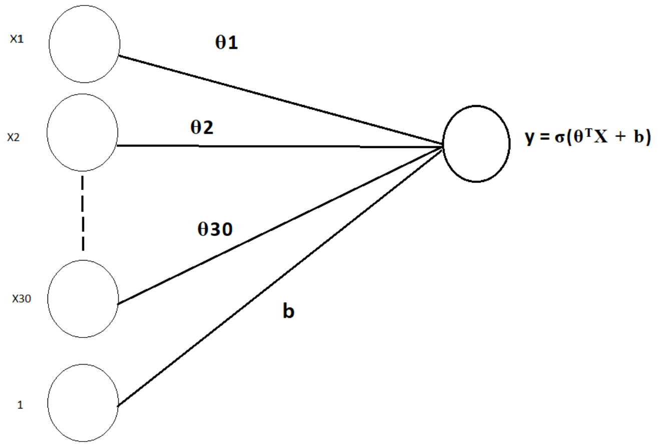
$$L = -\frac{1}{m} (y \log(a) + (1-y) \log (1-a))$$

$$L = -\frac{1}{m} (y \log(\sigma(z)) + (1-y) \log (1- \sigma(z)))$$

$$\Delta \theta 1 = -\frac{1}{m} (y - \sigma(z)) X1$$

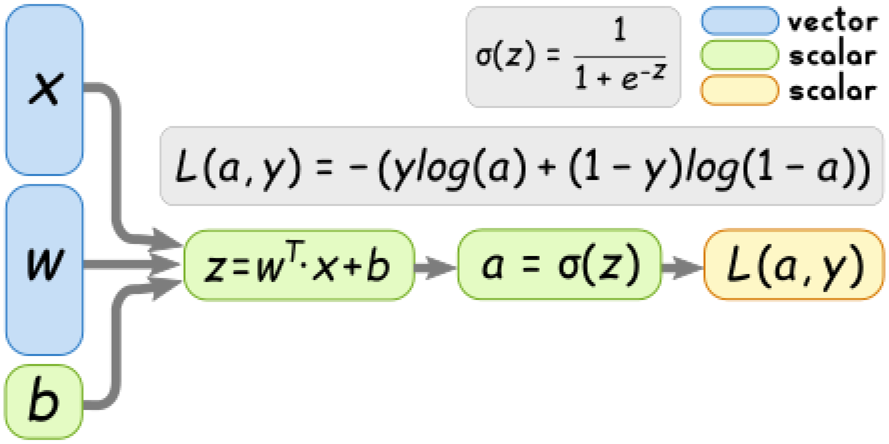
For updating Bias,

$$\Delta b = -\frac{1}{m} (y - \sigma(z)) .1$$



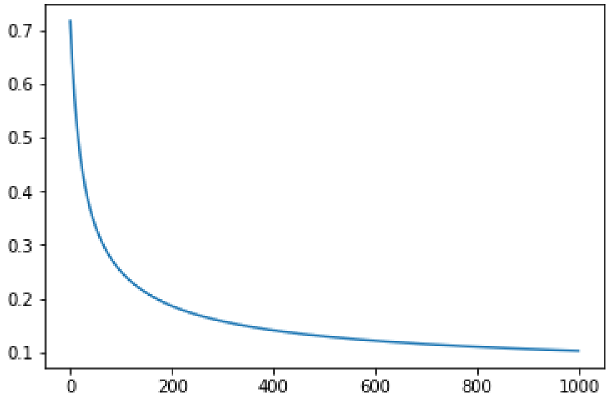
Genesis Equation representation with 30 different features and basis

4 Architecture


x
w
b
$\sigma(z) = \frac{1}{1+e^{-z}}$
vector
scalar
scalar
$L(a,y) = -(y\log(a) + (1-y)\log(1-a))$
$z=w^T \cdot x+b$
$a = \sigma(z)$
$L(a,y)$

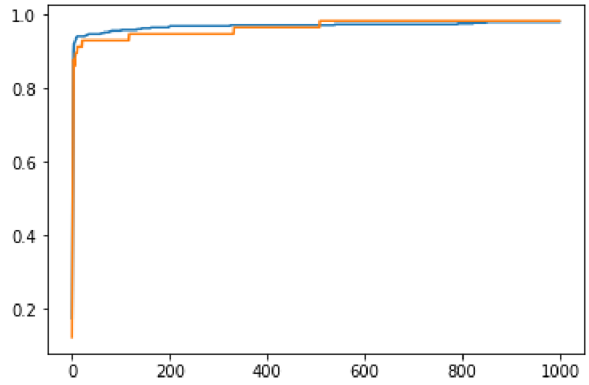
5 Result

We can see, with increasing epoch(X), the losstrack(Y) was initially decreasing thereafter reaching a constant value. The graph then tends to infinity.



Graph for Losstrack(Y) versus each epoch(X)

In this graph, both validation and training data are almost overlapping each other showing that the training data was trained effectively and it worked well on validation data as well.



Graph for accuracy score of training and validation data

$$\text{Accuracy score} = \frac{TP + TN}{TP+TN+FP+FN}$$

Accuracy score for testing data = 0.9824561403508771

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall score for testing data = 1.0

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision score for testing data = 0.9705882352941176

6 Conclusion

The experiment performed to classify breast cells as malignant or benign was successfully carried out using Logistic Regression. It was observed that training of 455 samples allowed us to correctly predict the classification. The accuracy was rightly achieved as 98% with a precision of 97% and recall of 100%.

7 Some Questions

Ques: How do you derive cost function?
Ans: To derive cost function, differentiate the loss function w.r.t weight(s).

Ques: Why do we need validation?
Ans: To change hyper parameters. Hyperparameters are multiple parameters on model which

can be trained by try and test and it can be changed during test process.

Ques: What is Normalized data?
Ans: Arranging data in some order.

Ques: Why normalized data?
Ans: To keep related features in same range.
Ques: How to decide number of loops/iterations?

Ans: If the losses are decreasing, increase the number of epochs and see where is it back at the original value.

References

[1] https://en.wikipedia.org/wiki/Linear_regression

[2] https://medium.com/@amarbudhiraja/ml-101-linear-regression-tutorial-1e40e29f1934

[3] https://en.wikipedia.org/wiki/Logistic_regression

[4] https://st3.ning.com/topology/rest/1.0/file/get/2808358994?profile=o-----riginal