IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

Code Development in Octave for Optimization Using Logistic Regression

D. Lavanya[1], Dr. Vivek Pandey[2]

Department of Mechanical Engineering, Raghu Engineering College Vishakhapatnam[1]

Associate Professor, Department of Mechanical Engineering, Raghu Engineering College Vishakhapatnam[2]

Abstract The problem of optimization is fundamental to various branches of Science and Engineering. In CAD and CAM, a present optimization problem is one of searching for patterns in given data which is necessary for recognizing shapes, assessing quality of manufactured goods etc. for the purpose of advanced robotic assisted Manufacturing. In this work, we have developed optimization code using logistic regression. This code can be very useful for manufacturing processes for separating manufactured goods into acceptable and non-acceptable classes. Machine Learning (ML) is a developing branch of science and will be widely applied in Manufacturing in the years to come. The classical optimization problems and their solutions using linear and logistic regression are well suited for ML. In this project we will built a regularized logistic regression optimization problem model for the ML in manufacturing process. We will implement regularized logistic regression to predict whether microchips from a fabrication plant pass Quality Assurance (QA). During QA , each microchip goes through various tests to ensure it is functioning correctly. To make decision, we have data set of test results of past microchips, from which we built a logistic regression model.

Keywords: Optimization, Manufacturing, Machine Learning (ML), logistic regression

I. INTRODUCTION

Machine learning is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed.[1] The name machine learning was coined in 1959 by Arthur Samuel. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions,[through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR) learning to rank, and computer vision. Don't get confused by its name! It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as **logistic regression**. Since, it predicts the probability, its output values lies between 0 and 1 (as expected). Again, let us try and understand this through a simple example. Let's say your friend gives you a puzzle to solve. There are only 2 outcome scenarios – either you solve it or you don't. Now imagine that you are being given wide range of puzzles / quizzes in an attempt to understand which subjects you are good at. The outcome to this study would be something like this – if you are given a trigonometry based tenth grade problem, you are 70% likely to solve it. On the other hand, if it is grade fifth history question, the probability of getting an answer is only 30%. This is what Logistic Regression provides you. Coming to the math, the log odds of the outcome is modeled as a linear combination of the predictor variables.

II. LOGISTIC REGRESSIONS

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes

2.1 Comparison to linear regression: Given data on time spent studying and exam scores. Linear Regression and logistic regression can predict different things:

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

- **Linear Regression** could help us predict the student's test score on a scale of 0 - 100. Linear regression predictions are continuous (numbers in a range).
- **Logistic Regression** could help use predict whether the student passed or failed. Logistic regression predictions are discrete (only specific values or categories are allowed). We can also view probability scores underlying the model's classifications.

2.2 Types of logistic regression

- Binary (Pass/Fail)
- Multi (Cats, Dogs, Sheep)
- Ordinal (Low, Medium, High)

2.3 Binary logistic regression: Say we're given data on student exam results and our goal is to predict whether a student will pass or fail based on number of hours slept and hours spent studying. We have two features (hours slept, hours studied) and two classes: passed (1) and failed (0).

Table: 1

Studied	Slept	Passed
4.85	9.63	1
8.62	3.23	0
5.43	8.23	1
9.21	6.34	0

Graphically we could represent our data with a scatter plot.

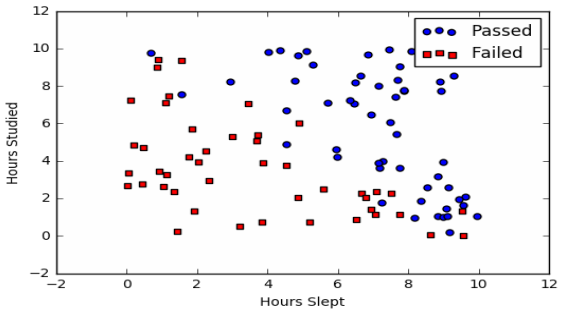

Figure: 1 represent of data

2.4 Sigmoid activation: In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

Math

$S(z)=11+e-z$

- s(z) = output between 0 and 1 (probability estimate)
- z = input to the function (your algorithm's prediction e.g. mx + b)
- e = base of natural log

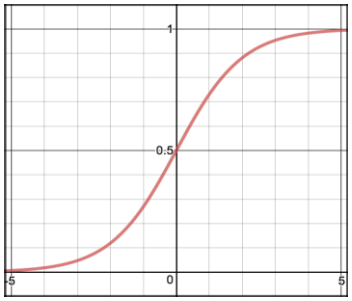

Figure 2: Sigmoid activation

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

2.5 Decision boundary

Our current prediction function returns a probability score between 0 and 1. In order to map this to a discrete class (true/false, cat/dog), we select a threshold value or tipping point above which we will classify values into class 1 and below which we classify values into class 2.

$$p \geq 0.5, \text{class} = 1 p < 0.5, \text{class} = 0$$

For example, if our threshold was .5 and our prediction function returned .7, we would classify this observation as positive. If our prediction was .2 we would classify the observation as negative. For logistic regression with multiple classes we could select the class with the highest predicted probability.

2.6 Making predictions: Using our knowledge of sigmoid functions and decision boundaries, we can now write a prediction function. A prediction function in logistic regression returns the probability of our observation being positive, True, or "Yes". We call this class 1 and its notation is P(class=1). As the probability gets closer to 1, our model is more confident that the observation is in class 1.

Math

Let's use the same multiple linear regression equation from our linear regression tutorial.
z=W0+W1Studied+W2Slept
This time however we will transform the output using the sigmoid function to return a probability value between 0 and 1.
P(class=1)=11+e−z
If the model returns .4 it believes there is only a 40% chance of passing. If our decision boundary was .5, we would categorize this observation as "Fail."

2.7 Cost function: Unfortunately we can't (or at least shouldn't) use the same cost function MSE (L2) as we did for linear regression. Why? There is a great math explanation in chapter 3 of Michael Neilson's deep learning book, but for now I'll simply say it's because our prediction function is non-linear (due to sigmoid transform). Squaring this prediction as we do in MSE results in a non-convex function with many local minimums if our cost function has many local minimums, gradient descent may not find the optimal global minimum

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

Math

Instead of Mean Squared Error, we use a cost function called Cross-Entropy, also known as Log Loss. Cross-entropy loss can be divided into two separate cost functions: one for y=1 and one for y=0

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

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

The benefits of taking the logarithm reveal themselves when you look at the cost function graphs for y=1 and y=0. These smooth monotonic functions (always increasing or always decreasing) make it easy to calculate the gradient and minimize cos

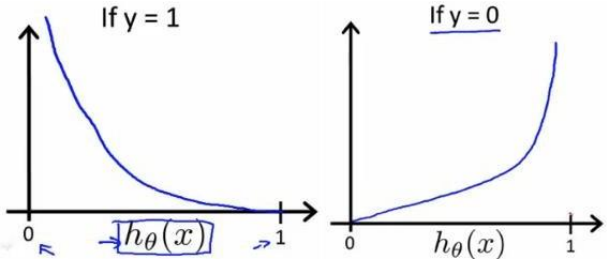

Figure3: Cost function

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

The key thing to note is the cost function penalizes confident and wrong predictions more than it rewards confident and right predictions! The corollary is increasing prediction accuracy (closer to 0 or 1) has diminishing returns on reducing cost due to the logistic nature of our cost function.

Above functions compressed into one: Multiplying by y and (1−y) in the above equation is a sneaky trick that lets us use the same equation to solve for both y=1 and y=0 cases. If y=0, the first side cancels out. If y=1, the second side cancels out. In both cases we only perform the operation we need to perform.

III. GNU OCTAVE

GNU Octave is software featuring a high-level programming language, primarily intended for numerical computations. Octave helps in solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with MATLAB. It may also be used as a batch-oriented language. Since it is part of the GNU Project, it is free software under the terms of the GNU General Public License.Octave is one of the major free alternatives to MATLAB, others being Scilab and FREEMat. Scilab, however, puts less emphasis on (bidirectional) syntactic compatibility with MATLAB than Octave does

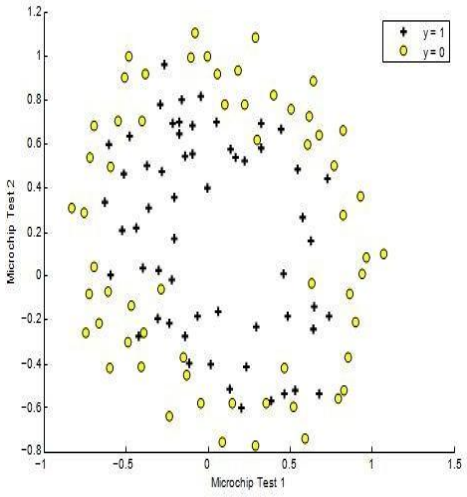

Figure4: Visualizing data

octave:1> x.a = 1; x.b = [1, 2; 3, 4]; x.c = "string"; octave:2> x.a
ans = 1 octave:3> x.b ans =
1 2
3 4
octave:4> x.c ans = string octave:5> x x =
{
a = 1 b =
1 2
3 4
c = string
}

IV. REGULARIZED LOGISTIC REGRESSION

In this part we have implemented regularized logistic regression to predict whether microchips from a fabrication plant pass quality assurance (QA). During QA, each microchip goes through various tests to ensure it is functioning correctly. Suppose we are the product manager of the factory and we have the test results for some microchips on two different tests. From these two tests, we would like to determine whether the microchips should be accepted or rejected. To help us make the decision, we have a dataset of test results on past microchip, from which you can build a logistic regression model

4.1 Visualizing the data: Before starting to implement any learning algorithm, it is always good to visualize the data if possible. In the first part of the code will load the data and display it on a 2-dimensional plot by calling the function plotData. We will now complete the code in plotData so that it displays a figure where

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

the axes are the two test scores, and the positive (y = 1, accepted) and negative (y = 0, rejected) examples are shown with different markers.

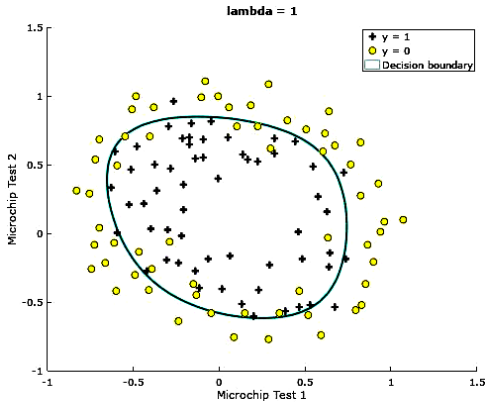

Figure 5: Training data with decision boundary

4.2 PLOTTING THE DECISION BOUNDARY: To visualize the model learned by this classifier, we have provided the function plot Decision Boundary.m which plots the (non-linear) decision boundary that separates the positive and negative examples. In plot Decision Boundary.m, we plot the non-linear decision boundary by computing the classifier's predictions on an evenly spaced grid and then and drew a contour plot of where the predictions change from y = 0 to y = 1 After learning the parameters θ, the next step in code will plot a decision boundary.

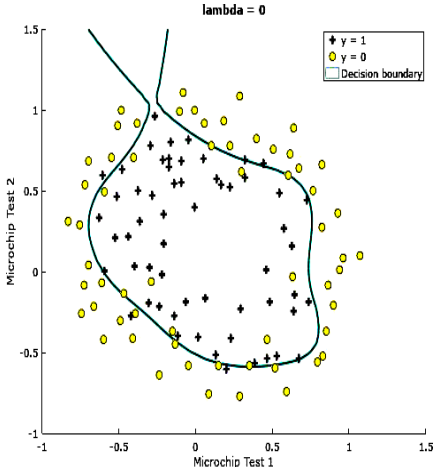

Figure 6: No regularization (Over fitting) (λ = 0)

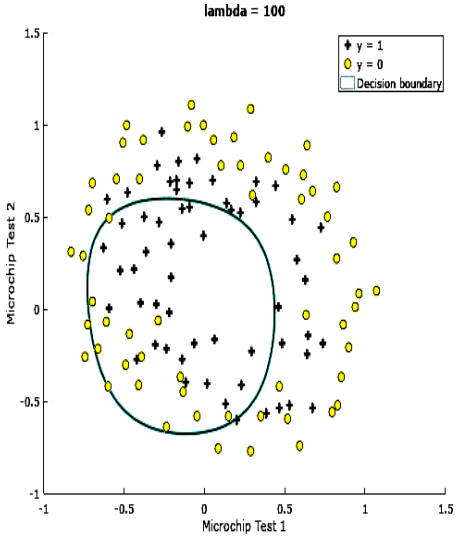

Figure 7: Too much regularization (Under fitting) (λ =100)

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2018

V. APPLICATIONS

I. The code developed in octave for optimization using logistic regression is as fallows. The octave window is shown in below figure

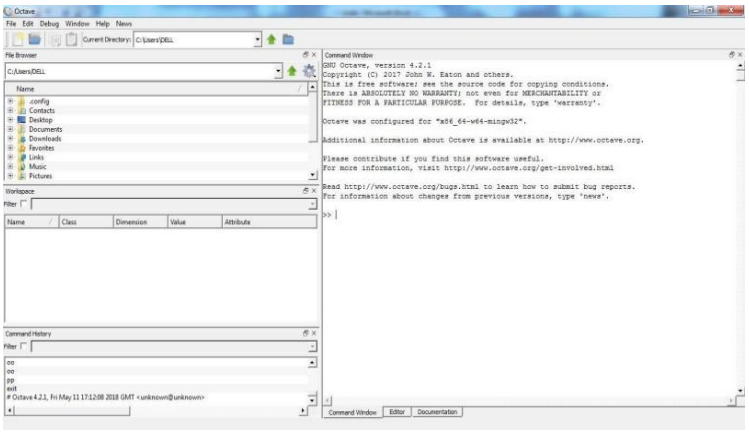

Figure 8: Editor Window

CONCLUSION

Machine learning is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed Logistic regression is one of the technique borrowed by machine learning from the field of statistics. Making predictions with a logistic regression model is as simple as plugging in numbers into the logistic regression equation and calculating a result. Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. It is the go-to method for binary classification problems (problems with two class values). It will predict the probability of an instance belonging to the default class, which can be snapped into a 0 or 1 classification. In this project we had learned the logistic regression algorithm for machine learning. We had implemented regularised logistic regression to predict whether microchips from a fabrication plant passes quality assurance (QA). To help us make decision, we had taken a dataset of test results on past microchip, from which you can built a logistic regression model We had developed a code in OCTAVE to visualize the data, feature mapping, cost function, gradient and also for plotting the decision boundary. And we got results at different values of regularization parameter lambda. Here, we used regularized logistic regression to predict whether microchips from a fabrication plant pass quality assurance. We used optimization function in lieu of gradient descent to get the optimal values of the coefficients. Finally, we got the optimal solutions at various values of lambda, which can be used for good outcomes of an industry.

REFERENCES

[1]https://en.wikipedia.org/wiki/Machine_learning#Journals
[2]http://www.google.co.in/search?dcr=0&ei=ClH1Wsv3NYvXvASvtrfwDw&q=logistic+regression+optimization+model&oq=logistic+regression+optimization+model&gs_l=psy-ab.3..0i22i30k1.4444.4444.0.7792.1.1.0.0.0.0.249.249.2-1.1.0....0...1c.1.64.psy-ab..0.1.2480.f5f9h4TANdY
[3]https://en.wikipedia.org/wiki/Logistic_regression#Setuphttps://en.wikipedia.org/wiki/GNU_Octave
[4]https://en.wikipedia.org/wiki/Mathematical_optimization
[5]https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
[6]http://mlcheatsheet.readthedocs.io/en/latest/logistic_regression.html
[7]Bishop - Pattern Recognition And Machine Learning - Springer 2006