CH5019 Mathematical Foundations of Data Science

Team Project

Team Members:

N Sowmya Manojna	BE17B007
VVS Lalitha	CE17B063
Lakshman Kanth Boyina	ME16B021
S Rahul	ME16B036
Kamesh K	MM16B107

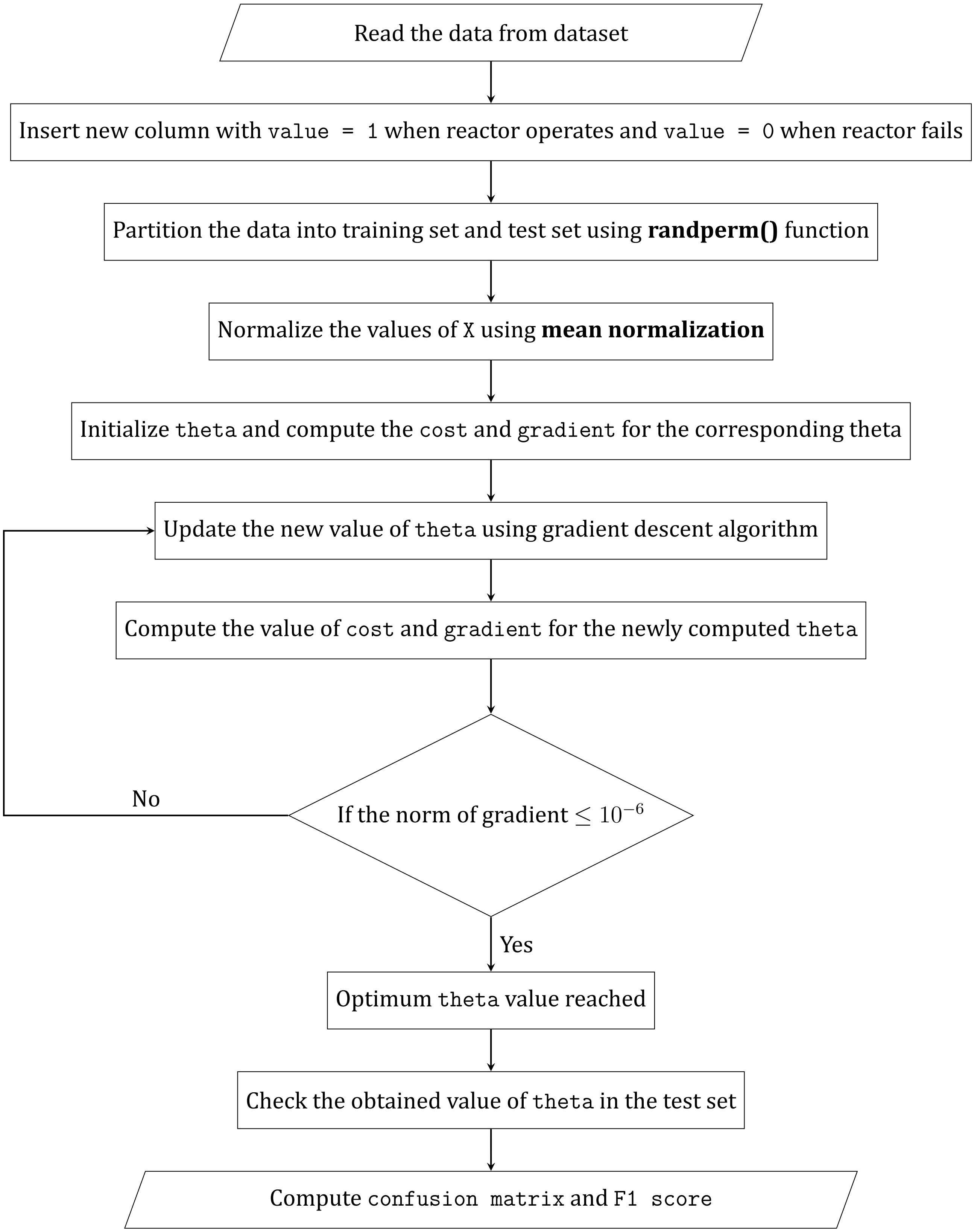
Indian Institute of Technology, Madras

Contents

1 Question 1

2 Question 2

This question was coded using MATLAB.
The flowchart of the approach used is as follows:



2.1 Result Statistics

2.2 Data Partitioning

Data was partitioned using the MATLAB inbuilt function `randperm(n)`. This function returns an array containing the random permutation of integers from 1 to n (in our case 1000) without repeating elements. The first 700 elements of this array and the corresponding entries in the data comprise the training set. The remaining 300 entries were made as the testing set.

2.3 Logistic Regression

$$\boxed{\boldsymbol{h(X)} = \text{sigmoid}(\boldsymbol{\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5})}$$

Where, $h(X)$ is the probability that y=1 for the value of X

x_1 is Temperature

x_2 is Pressure

x_3 is Feed flow rate

x_4 is Coolant flow rate

x_5 is Inlet reactant concentration

2.3.1 Sigmoid Function

$$\boxed{\text{Sigmoid}(\boldsymbol{z}) = \frac{\boldsymbol{1}}{\boldsymbol{1 + e^{-z}}}}$$

From the fig. we can see that if $z > 0$ output will be greater than 0.5 otherwise, output is less than or equal to 0.5.

So we predict as follows: $y = \begin{cases} 1 & h(X) \leq 0.5 \\ 0 & otherwise \end{cases}$

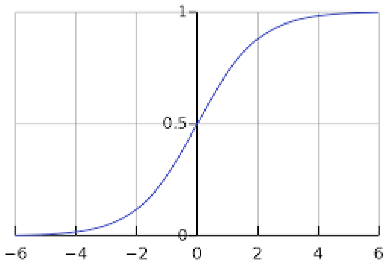


Figure 1: The sigmoid function

2.3.2 Cost function

Since it is a classification problem the cost depends both on the values of $h(X)$ and y. So, we take cost as:

$$\boxed{cost(h(X), y) = \begin{cases} -log(h(X)) & y = 1 \\ -log(1 - h(X)) & y = 0 \end{cases}}$$

By assigning costs like this, we penalize the algorithm if it predicts wrongly (by increasing the cost by a large amount).

In the generalized form, we take the cost function to be:

$$\boxed{J(\theta) = \frac{-1}{m}\Big(\sum_{i=1}^{m}(y_i * log(h(X_i)) + (1 - y_i * log(1 - h(X_i)))\Big)}$$

Where, J is the cost and m is total number of training examples taken.

As y can only take values 0 and 1, the cost in the generalized formulation reduces as below:
When $y = 1$ only the first term is active which implies that $cost = -log(h(X))$
When $y = 0$ only the second term is active which implies that $cost = -log(1 - h(X))$

So we can infer that both formulations are same.

Since each parameter had different ranges, **mean normalization** for each parameter was done before starting Gradient descent. The formula used for the same is as follows:

$$\boxed{X_{norm}(i,j) = \frac{X(i,j) - mean(j)}{Range(j)}}$$

2.3.3 Gradient Descent

Our objective is to minimize the cost function value (J).
So we iterate for the values of theta in the following manner:

$$\theta_{j,new} = \theta_{j,old} - \alpha\frac{\partial J(\theta)}{\partial \theta_j}$$

Since

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\Big(\sum_{i=1}^{m}(h(X_i) - y_i)X_i(j)\Big)$$

we get,

$$\boxed{\theta_{j,new} = \theta_{j,old} - \alpha\frac{1}{m}\Big(\sum_{i=1}^{m}(h(X_i) - y_i)X_i(j)\Big)}$$

The learning rate α was taken to be 0.01.
Convergence condition: Norm of the gradients of $\theta \leq 10^{-5}$

We iterate for the values of theta by using the above condition until the convergence condition is reached.

2.4 Performance

Obtained values of theta are: 1.2369, -0.6470, -1.6551, -2.1753, 11.9063, -0.4493
Obtained Cost = 0.2720

Number of cases predicted correctly = 286
Number of cases predicted wrongly = 14

Accuracy = $\frac{286}{300}$ = 0.9533 = 95.33%

Number of True positive obtained (TP) = 163 ($y_{obtained} = 1$ and $y_{test} = 1$)
Number of False positive obtained (FP) = 9 ($y_{obtained} = 1$ and $y_{test} = 0$)
Number of False negative obtained (FN) = 5 ($y_{obtained} = 0$ and $y_{test} = 1$)
Number of True negative obtained (TN) = 123 ($y_{obtained} = 0$ and $y_{test} = 0$)

2.4.1 Confusion matrix

		Actual Values	
		Positive ($y_{test} = 1$)	Negative ($y_{test} = 0$)
Predicted Values	Positive ($y_{pred} = 1$)	163 *(TP)*	9 *(FP)*
	Negative ($y_{pred} = 0$)	5 *(FN)*	123 *(TN)*

2.4.2 F1 Score

$$\text{Precision(P)} = \frac{TP}{TP + FP} = \frac{163}{172} = 0.9477$$

$$\text{Recall(R)} = \frac{TP}{TP + FN} = \frac{163}{168} = 0.9702$$

$$\text{F1 score} = \frac{2RP}{R + P} = 0.9588$$