IIT PALAKKAD

CS4801: Principles of Machine Learning

Under the guidance of Dr. Sahely Bhadra

Linear algebra, probability, regression assignment

Date of submission : Aug 28 2017

Submitted by

Jayaprakash A

111501010

Contents

Chapter 1

Problem 1

1.1 Problem Statement

Write a program (without using any machine learning related in built library) to create another file named as iris-svm-input.txt which contain the same data but in the given format

1.2 Methodology

The problem has been solved in python. The file was read line by line. The line was stripped of any trailing white spaces or new line character. Later it was split based on the delimiter character ','. Necessary processing of data was done and written to the the required file `iris-svm-input.txt` in the requested format.

1.3 Observations and results

The output file was checked thoroughly and no errors were reported.

Chapter 2

Problem 2

2.1 Question 2.a

2.1.1 Problem Statement

Implement least square regression with help of matrix inversion

- w = LeastSquares(Featurematrix, y):
  - input: Feature matrix matrix $\phi \in \mathbb{R}^{nxp}$ and the outputs y $\in \mathbb{R}^n$ (column vector)
  - output: weight vector w of least squares regression as column vector

2.1.2 Methodology

The program written takes the following as input from the input.

- The dimensions of feature matrix
- The feature matrix
- The output matrix

To find out the weight vector corresponding to the given matrices. We calculate it by minimizing the sum of squares of residual, S(β) = (Y - Zβ)^t(Y - Zβ). On partially differentiating the residual function w.r.t β we get, Z^tZβ = Z^tY. The above equation reduces to $\beta = (Z^tZ)^{-1}Z^tY$.

Various `numpy` functions were used in the implementation in order to transpose a matrix, to invert a matrix and else where.

2.1.3 Results

The code was run on an example from the INTERNET. It gave results very to the one given there.

2.2 Question 2.b

2.2.1 Problem Statement

Implement stochastic gradient descent algorithm with step size 0.1 to solve ridge regression.

- w = RidgeRegression(Featurematrix, y, λ):
  - input: Feature matrix matrix $\phi \in \mathbb{R}^{nxp}$ and the outputs y $\in \mathbb{R}^n$ (column vector) and the regularization parameter $\lambda \in \mathbb{R}^+$
  - output: weight vector w of least squares regression as column vector

2.2.2 Methodology

The program written takes the following as input from the input.

- The dimensions of feature matrix
- The regularization parameter
- The feature matrix
- The output matrix

The residual function in case of ridge regression S(β) = $\frac{1}{2}$(Y - Zβ)^t(Y - Zβ) + $\frac{\lambda}{2}\beta^t\beta$.
We initialise $\beta = \beta^0$. Repeat until convergence :

$$\beta = \beta - \alpha\frac{\partial S}{\partial \beta}$$
$$\beta = \beta - \frac{1}{2}\alpha(Z^t(\text{Z}\beta \text{ - Y}) + \lambda\beta)$$

Various `numpy` functions were used in the implementation in order to transpose a matrix, to invert a matrix and else where. The number of iterations is just the number of example data provided. According to some articles running the iteration on all the example data once in a random manner is a very good approximate to the answer.

2.2.3 Results

The code was tested by comparing the answers obtained by Ridge regression (without using gradient descent) and the one with it. The answers were close enough. Error was just of the $\mathbb{O}(0.1)$ decimal.

2.3 Question 2.c

2.3.1 Problem Statement

Find optimal hyper-parameter λ for ridge regression using 5 fold cross validation on training data. Find out the optimal λ from $\lambda \in \{2^{-10}, 2^{-9}, 2^{-8}, \ldots, 2^{0}, 2^{1}, \ldots, 2^{10}\}$. The relation between x and y can be non-linear. Hence to catch non-linear relationship we will generate Featurematrix=[1, x, x^2 , . . . ,x^{10}]

2.3.2 Methodology

The functions written earlier were used in the calculation. Ridge regression has been used without any gradient descent algorithm. For all the k folds in the cross validation section errors have been calculated and stored in a dictionary. Later after all the k iterations, the minimum among them was found.

2.3.3 Observations and results

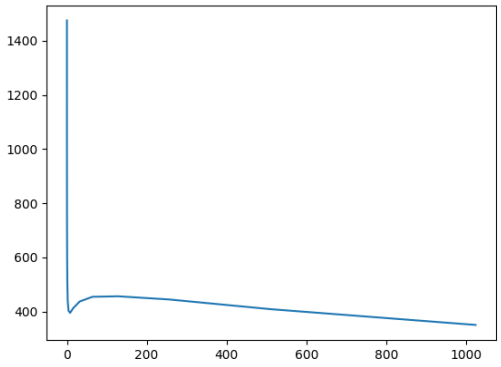


Figure 2.1: The error percentage vs λ cross validation

We can clearly see from the above plot that the optimal value of hyper parameter is 2^{10}.

2.4 Question 2.d

2.4.1 Problem Statement

Discuss with increase of regularization parameter how the training error, validation error and test error change. Also report test error for least square regression. Plot three graph showing these three kinds of error where x-axis will show λ and y axis will show corresponding error. [for better visualization use log scale along x axis].

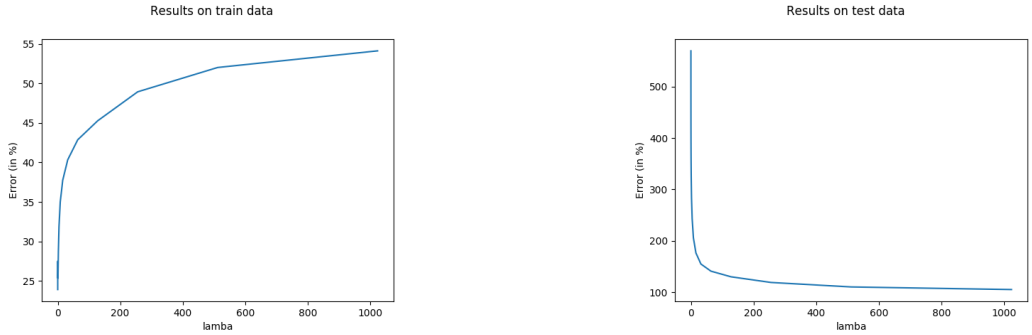
2.4.2 Methodology

The functions written earlier were used in the calculation. The weight vector calculated was applied on training data and test data. The difference of calculated data and actual data were logged to various files. The error in measuring is taken to the mean of absolute values of percentage error between each calculated and expected value. The errors were plotted along with its corresponding hyper parameter.

2.4.3 Observations and results

In case where the weight vector was applied on training data itself, the error percentages were less for smaller values of λ and it kept increasing with increase in λ. However, this was not the case when results were tested on the given test data. Results obtained from test data had very high errors for lesser values of λ and vice versa.

The test error for linear regression model was very high and very similar to the case of ridge regression with very small values of lambda.



(a) Training data set (b) Test data set

Figure 2.2: The error percentage vs λ for different data sets by ridge regression

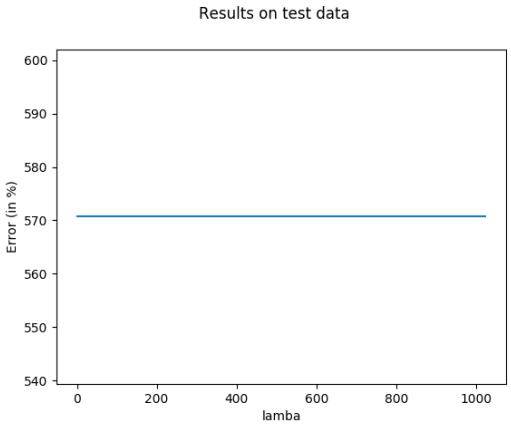


Figure 2.3: The error percentage vs λ for different data sets by linear regression