COL 341 – ASSIGNMENT 1

Name: Priyadarshini Radhakrishnan

Entry Number: 2021CS50614

3.1 Basic Implementation

I used the values 1000 and 1e-3 for “maxit” and “reltol”. For the learning rates 0.1 and 0.01, there was a divergence and MSE was calculated as inf since it was a very large value. But for values less than or equal to 0.001 there was a convergence. So, I reduced the value of such losses by a factor of ten so that I can get an output.

For the train data, the values of MSE and MAE for different learning rates:

For learning rate = 0.1:

MSE = 6.944444444444444e+305

MAE = 1.5449001417019539e+156

For learning rate = 0.01:

MSE = 6.944444444444444e+305

MAE = 3.626526509015206e+154

For learning rate = 0.001:

MSE = 0.3930611129008407

MAE = 0.5033635665832086

For learning rate = 0.0001:

MSE = 0.6415094978623606

MAE = 0.6248683262173887

For the validation data, the values of MSE and MAE for different learning rates:

For learning rate = 0.1:

MSE = 4.761904761904762e+306

MAE = 2.1228620412335e+156

For learning rate = 0.01:

MSE = 4.761904761904762e+306

MAE = 5.2725937368771787e+154

For learning rate = 0.001:

MSE = 0.12110561196045729

MAE = 0.2462006213500875

For learning rate = 0.0001:

MSE = 0.7445861234453187

MAE = 0.6613757786988239

3.2 Ridge Regression

I used the values 1000 and 1e-3 for "maxit" and "reltol". For the learning rates 0.1 and 0.01, there was a divergence and MSE was calculated as inf since it was a very large value. But for values less than or equal to 0.001 there was a convergence.

For ridge regression the equation of the in-sample error is different. The vector form of the cost function is,

$$E_{in}(W) = \frac{1}{N}\left(\left|\left|\mathrm{XW} - \mathrm{Y}\right|\right|^2 + \lambda * \mathrm{W^T W}\right).$$

Hence, the gradient of this function is given by,

$$\nabla E_{in}(W) = \frac{2}{N}(X^T XW - X^T Y + \lambda * W).$$

The parameter λ is added in the total cost function as a tuning parameter to balance the fit of data and magnitude of coefficients.

The rest all equations is same as in the case of linear regression.

For the train data, the values of MSE and MAE for different learning rates and for different λ:

For $\lambda = 5$:

For learning rate = 0.001:

MSE = 0.398595201802582

MAE = 0.5045876541253621

For learning rate = 0.0001:

MSE = 0.6435793661817213

MAE = 0.6248770672319246

For $\lambda = 25$:

For learning rate = 0.001:

MSE = 0.41838611106693346

MAE = 0.5084604593421398

For learning rate = 0.0001:

MSE = 0.6518555475325011

MAE = 0.6249119688369733

For the validation data, the values of MSE and MAE for different learning rates and for different λ:

For $\lambda = 5$:

For learning rate = 0.001:

MSE = 0.6642383722265102

MAE = 0.43642009939082727

For learning rate = 0.0001:

MSE = 0.8249461109761681

MAE = 0.6826838556964657

For $\lambda = 25$:

For learning rate = 0.001:

MSE = 0.6642383722265102

MAE = 0.43642009939082727

For learning rate = 0.0001:

MSE = 1.0883530919535076

MAE = 0.7478141098843094

3.3 Using Scikit-Learn Library

MSE and MAE on validation data using the corresponding weights received from the linear regression of scikit library:

MSE = 4.904761904761905

MAE = 1.476190476190478

3.4 Feature Selection

Note: For selectKbest I gave the score function as f_regression.

For the train data, the values of MSE and MAE for different learning rates:

For learning rate = 0.1:

MSE = 1.2527252901651933

MAE = 0.8824734996550964

For learning rate = 0.01:

MSE = 1.4516210469363395

MAE = 0.9653966247427874

For learning rate = 0.001:

MSE = 1.6369766432963808

MAE = 1.0218970632766058

For learning rate = 0.0001:

MSE = 5.92719916367684

MAE = 2.2137737589259543

For the validation data, the values of MSE and MAE for different learning rates:

For learning rate = 0.1:

MSE = 1.2219812619814123

MAE = 0.9015827351092958

For learning rate = 0.01:

MSE = 1.7446720862534815

MAE = 1.0718493735447243

For learning rate = 0.001:

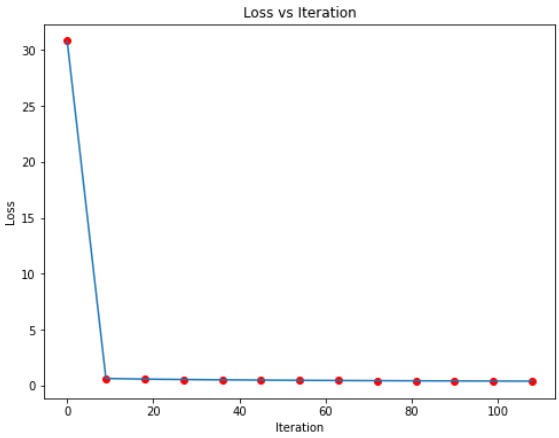
MSE = 1.9841162260741019

MAE = 1.1595125225760432

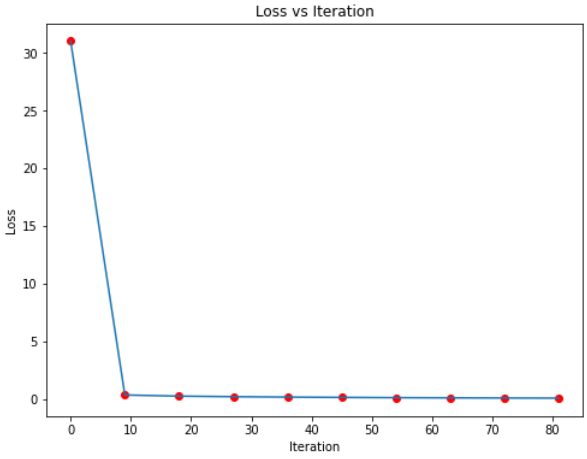
3.6 Visualization

For section 3.1:

For the train data the loss vs iteration plot is:

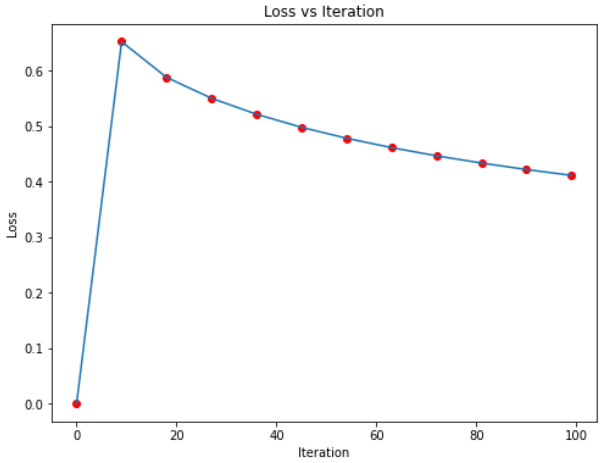


For the validation data the loss vs iteration plot is:

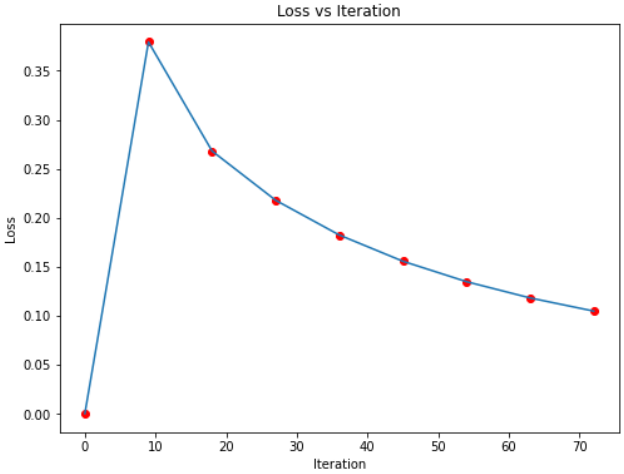


For section 3.2:

For the train data the loss vs iteration plot is:

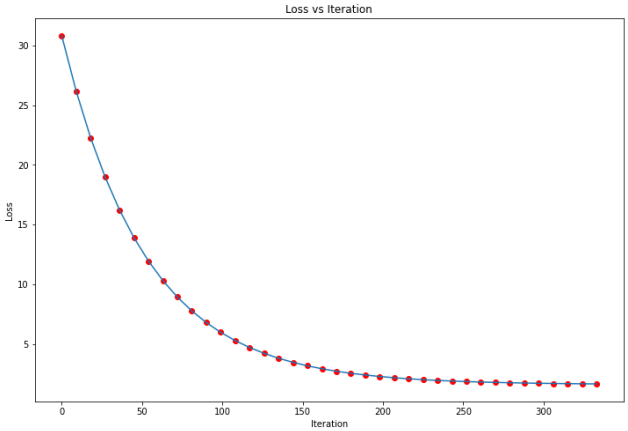


For the validation data the loss vs iteration plot is:

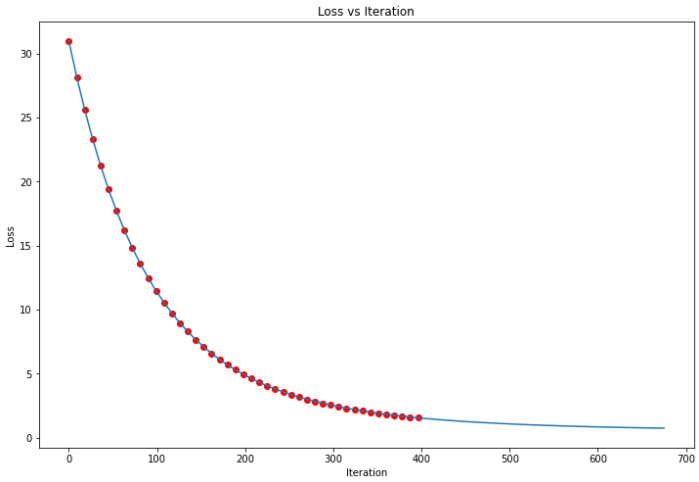


For section 3.4:

For the train data the loss vs iteration plot is:



For the validation data the loss vs iteration plot is:



After splitting the data into 2 halves:

For learning rate = 0.001 with section 3.1:

For the first half:

MSE = 0.19373053029377024

MAE = 0.3630722607449653

For the second half:

MSE = 0.2627004820547451

MAE = 0.4174302738174253

For learning rate = 0.001 and param 5 with section 3.2:

For the first half:

MSE = 0.19373053029377024

MAE = 0.3630722607449653

For the second half:

MSE = 0.2627004820547451

MAE = 0.4174302738174253

3.7 Generalization

For the 1st plot I took learning rate as 0.01, maxit = 1000 and reltol = 1e-4

For 2_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.018169539042977138

E_{out} = 0.1159150561724617

For 5_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.0268255344216416

E_{out} = 0.1124970945615535

For 10_d model, the vales of E_{in} and E_{out} are:

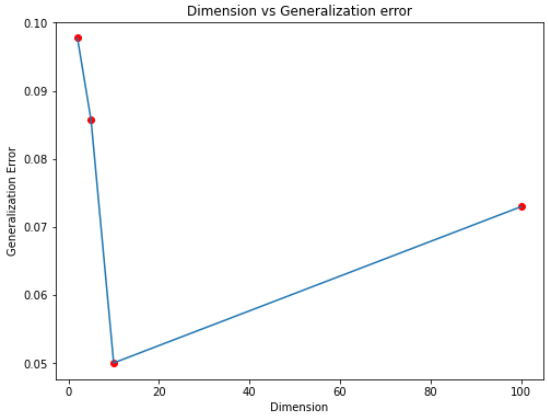
E_{in} = 0.029648958372494923

E_{out} = 0.07964717637771195

For 100_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.023874547297993494

E_{out} = 0.09683849903708203



For the 1st plot I took learning rate as 0.0001, maxit = 1000 and reltol = 1e-4

For 2_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.2898646096856307

E_{out} =0.2061306192130888

For 5_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.22946217669674468

E_{out} = 0.30255557894356894

For 10_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.167478396948282

E_{out} = 0.1042061767072468

For 100_d model, the vales of E_{in} and E_{out} are:

E_{in} = 0.09007788215240871

E_{out} = 0.099297956024897