CS589 Machine Learning

Homework 1

Submitted by- Ravi Agrawal Due on- October 2nd, 2017

1.a: - Given feature fi to split the data into the subset D0 and D1:

From the Note we know the fraction of data in the region m with label k to be

$$p\_hat_{mk} = (1/Nm)\ sum(I[yi = k])$$

Expression from the information gain in this case is The fraction of data in region Di with label fi is p_hat_{Difi}:-

$$p\_hat_{Difi} = (1/\ N) sum(I[yi = fi])$$

And more intuitively, we want p_hat_{mk} to have low randomness, and the criterion we can have to choose p_hat_{mk} can be using Misclassification Error 1 - p_hat_{mk}. But this is hard to optimize.

Another, Criterion, Can be Gini Index

$$Sum(p\_hat_{mk}\,(1 - p\_hatmk)) \text{ For K 1 … K}$$

Or we can use cross entropy:

$$-Sum(p\_hat_{mk}\, log(p\_hatmk))$$

1.b: - The out of sample error is reported bellow in the table 1 below. It is clear from the sample error that Decision Tree with **max depth 9** outperforms other trees.

The Max Depth does significantly affects the accuracy of the Tree. As we can see in the table below as the Tree max depth increases the Accuracy is also increasing. Before the max depth of the tree is 9 the tree is facing high bias problem (under Fitting). At max depth equals 9 the Tree is performing just right and then as the max depth is increasing the Tree start facing the problem of high variance (Over Fitting) and sample error again start increasing.

Kaggle: - The Predicted output with Decision Tree **max depth 9** is tested in the Kaggle and the predicted output gives higher accuracy In **Kaggle output (0.59150)** as compared to the validation set accuracy of 0.518.

Max Depth	Accuracy	Sample Error
3	0.448	0.552
6	0.499	0.501
9	0.518	0.482
12	0.513	0.487
14	0.51	0.49

Table 1 Decision Tree Sample Error for Different Depth

2 a: - Given the data is D Contains N samples with F features and we need to find test complexity of X query points.

So, the time complexity for single query point is O(NF).

And for X query points the test complexity will be **O(NFX)**.

2 b: The out of sample error for the five different KNN model with neighbours [3, 5, 7, 9, 11] is shown in the table below:

Neighbors	Accuracy	Sample Error
11	0.53425	0.46575
3	0.55285	0.44715
5	0.54785	0.45215
7	0.54165	0.45835
9	0.53800	0.462

The K Nearest neighbor neighbors with **3 neighbors** model is trained in the full training data and tested on the Kaggle. This model gives accuracy score of **0.61722** on Kaggle on the test data.

3 a: The out of Sample error for the 10 different models is reported below in the table.

The **Hinge loss** Model with **alpha 0.01** gives minimum out of sample error in the validation set. The Sample model is trained on the full dataset and used to generate the prediction for the test set. Kaggle Sample error on the predicted output is **0.57834.**

Loss	Alpha	Accuracy	Sample Error
Hinge Loss	1.00E-06	0.41895	0.58105
	0.0001	0.44655	0.55345
	0.01	0.5287	0.4713
	1	0.44265	0.55735
	10	0.457	0.543
Logistic Regression Loss	1.00E-06	0.23885	0.76115
	0.0001	0.26485	0.73515
	0.01	0.23855	0.76145
	1	0.23015	0.76985
	10	0.26505	0.73495

Table 2 Linear Model with L2 regularization

4 a.1 : derivative is listed below:

h = b + np.dot(w, X)

dLdc = dldf

dLdV = np.dot(dldf, np.transpose(sig(h)))

dLdb = np.multiply(sigp(h), np.dot(np.transpose(V), dldf))

dLdW = np.multiply(sigp(h), np.dot(np.dot(np.transpose(V), dldf), np.transpose(x)))

4.a.2: The Output of the derivative as listed below:

Loss = 4.8145

dLdc, Autograd

[[0.917135 0.045394 -0.991889 0.02936]]

dLdc, partial derivative

[[0.917135 0.045394 -0.991889 0.02936]]

dLdV, Autograd

[[-0.908328 0.848131 0.87481 -0.869308 -0.784553]

[-0.044958 0.041979 0.043299 -0.043027 -0.038832]

[0.982365 -0.917261 -0.946115 0.940164 0.848501]

[-0.029078 0.027151 0.028005 -0.027829 -0.025116]]

dLdV, partial derivative

[[-0.908328 0.848131 0.87481 -0.869308 -0.784553]

[-0.044958 0.041979 0.043299 -0.043027 -0.038832]

[0.982365 -0.917261 -0.946115 0.940164 0.848501]

[-0.029078 0.027151 0.028005 -0.027829 -0.025116]]

dLdb, Autograd

[[0.049305 0.189112 0.115923 -0.175011 -0.786228]]

dLdb, partial derivative

[[0.049305 0.189112 0.115923 -0.175011 -0.786228]]

dLdW, Autograd

[0.004256 0.016843 0.007743 -0.007793 -0.037776 0.003351 0.008865

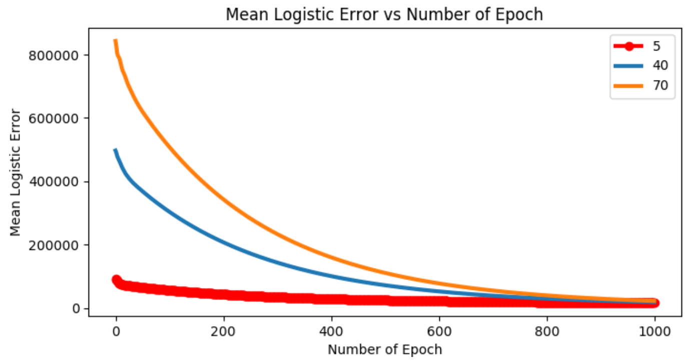
0.003396 -0.006905 -0.024877]

dLdW, partial derivative

[0.004256 0.016843 0.007743 -0.007793 -0.037776 0.003351 0.008865

0.003396 -0.006905 -0.024877]

4.b.1: The Mean Logistic Error is plotted against the number Epoch.



4.b.2: The Training Time for the All three-neural network is shown below:

Hidden Units	Training Time (Milliseconds)
5	43888.22
40	133807.23
70	218470.64

4.b.3: Validation Error for three neural networks with hidden units [5, 40, 70]. It is clear from the table below that as the number of hidden units increases, the accuracy is also increasing. The Neural Network with 5 hidden units is suffering from underfitting. The 40 Hidden unit neural is performing better if compared to the neural network with 5 hidden units. The 70-hidden unit is outperforming other two neural networks.

The Neural Network with 70 Hidden unit is selected because it gives lowest validation error. The Model when trained in the full training set. The predicted output from the model, when tested in the Kaggle generates the Accuracy score of 0.79126. At the time of testing my **Kaggle standing was 5.**

Hidden Unit	Validation Error	Validation Accuracy
5	0.441	0.559
40	0.332	0.668
70	0.3292	0.670