CSE 574 Introduction to Machine Learning

Programming Assignment 3

TEAM 18

Team Members:

Name	Email	UB ID
Akash Yeleswarapu	akashyel@buffalo.edu	50207826
Maheedhara Achalla	maheedha@buffalo.edu	50207395
Phani Ram Sayapaneni	phaniram@buffalo.edu	50166685

1: BINARY LOGISTIC CLASSIFIER

This technique assumes only two classes for the output. That is, it assumes Bernoulli distribution for the output data. Here we initially train the weights using the training data for each class(10-digits) by minimizing error using the gradient descent technique. Next we apply the testing data to this model and find out which class gives maximum classification against all other classes. Thus, even though our model is only a binary classifier we are using it classify each digit against all other digits and find the most probable digit.

RESULTS:

Training set Accuracy: **84.904**%
Validation set Accuracy: **83.68**%
Testing set Accuracy: **84.15**%

2: DIRECT MULTI-CLASS LOGISTIC REGRESSION:

This technique is used for binary classification. Here, we don't need to build 10 classifiers like Binary Logistic Classifier. Instead, we build only one classifier that can classify 10 classes at the same time and we train our sample data by using this one classifier. The following are the results obtained on the training set, validation set and testing set when MLR is used.

RESULTS:

Training set Accuracy: **93.232**%
Validation set Accuracy: **92.47**%
Testing set Accuracy: **92.45**%

CONCLUSION:

We observed that the accuracy of all the three datasets i.e. training, validation and testing set have improved significantly when MLR is used instead of BLR. Also we observed that MLR is quite fast.

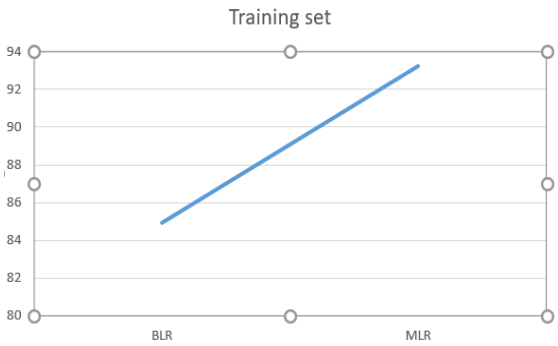
MLR vs BLR - PERFORMANCE:

In Binary Logistic Classifier, we build the classifier for each class in the data set and train them. This process takes a long time since we need to train classifiers for each class. The application of binary logistic regression to classify handwritten digits is interesting firstly. We need to understand the fact that image pixels have dependencies, that is, there would be dependencies within the features. Also when we are trying to classify each number against all other, the two possible classes might not be completely independent. Also there are too many features (716), any outliers might make it difficult to clearly mark boundaries between the classes. This also increases the computation time to minimize the error function. Thus it is not surprising to accuracies around 85% for binary logistic regression. The idea of individual classifier for each digit is itself inefficient and so is the computation time, performance of the technique. Also it is difficult, time consuming for the gradient descent algorithm to use all the 716 dimensions from error vector to minimize the error. And this has to be done for each class(digit), this is very time consuming and little inefficient.

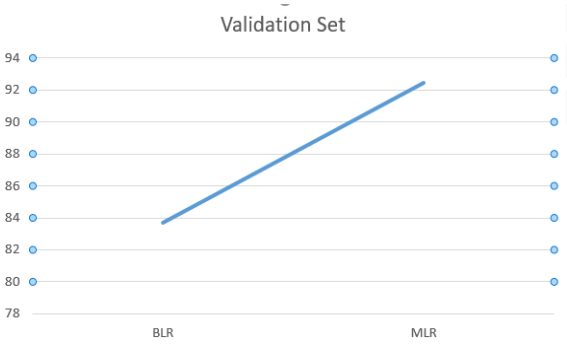
MLR, on the other hand, runs way faster when compared to BLR since we are building a single classifier and training this classifier. This method also provides significant accuracies when compared to BLR since we find the probability of a given sample of a class by computing probabilities of all the classes and taking the maximum one. MLR also showed no signs of overfitting or underfitting.

The following graphs show the difference in accuracies between BLR and MLR for different datasets:

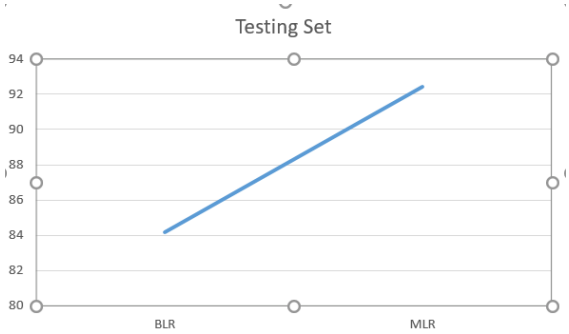
Training Set:



Validation Set:



Testing Set:



Conclusion:

MLR can be preferred over BLR to get better accuracies and faster results. But MLR can’t be used for when the datasets are huge as usage of softmax function in MLR could be expensive.

SUPPORT VECTOR MACHINE

In SVM, we chose only the points which are close to the margins and try to provide the maximum possible distance between the points and the hyper-plane. Below is the list of accuracies across the training, validation and the testing data set using both Linear and Radial Basis Function.

Linear Kernel

Linear SVM is very much useful if data under test is high dimensional data and original data is highly informative. It is also less prone to overfitting issues.

Training Set Accuracy	97.28%
Validation Set Accuracy	93.64%
Test Set Accuracy	93.78%

Radial Basis Function

It is a non-linear SVM which is influenced by parameters such as gamma and C. The gamma parameter defines how far the influence of the single training example reaches, with values meaning 'far' and high values meaning 'close'.

With gamma=1

Training Set Accuracy	100%
Validation Set Accuracy	15.48%
Test Set Accuracy	17.14%

With gamma= default value

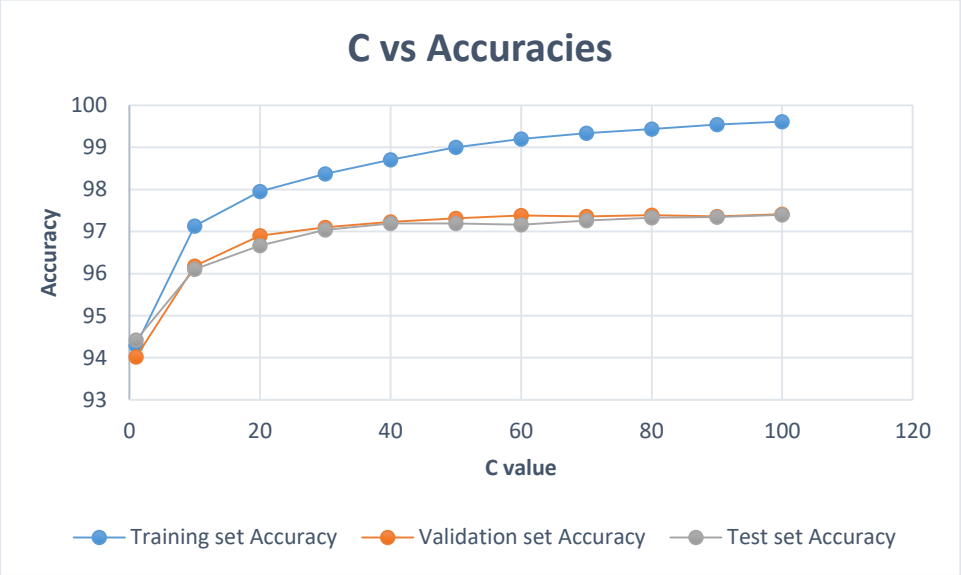
Training Set Accuracy	94.29%
Validation Set Accuracy	94.02%
Test Set Accuracy	94.42%

It can be clearly observed that higher values of gamma cause overfitting problem whereas lower values of gamma give encouraging results.

Varying C values with default gamma

The C parameter governs the impact of error on the training examples which in turn is used to control the complexity of the learned hyper-plane. For Larger values of C, the optimization chooses smaller-margin hyper plane. Conversely, the smaller value of C will cause the optimizer to look for a larger-margin separating hyper plane.

C	Training Accuracy	Validation Accuracy	Test Accuracy
1	94.294	94.02	94.42
10	97.132	96.18	96.1
20	97.952	96.9	96.67
30	98.372	97.1	97.04
40	98.706	97.23	97.19
50	99.002	97.31	97.19
60	99.196	97.38	97.16
70	99.34	97.36	97.26
80	99.438	97.39	97.33
90	99.542	97.36	97.34
100	99.612	97.41	97.4



As it can be seen from the plot, we have higher accuracies for the higher values of C. It is because, C controls the penalty for error term on each training example. When C is low, the weight of each error term is low as well, so larger error value is accepted during training phase. So larger margin hyperplane is created at the expense of more samples misclassified. Conversely, when C increases, the weight of each error term increases, so lower error values will be accepted. So lower margin hyperplane is created which improves the accuracy.

Also there is high risk of overfitting with increase in value of C. If we carefully observe the trend in increment of different data sets, increment in accuracy of training data set is less after C=10 whereas for validation and test data sets, accuracy does not increase much after C=40. This can be attributed to overfitting issue with increase in C.

CONCLUSION:

We conclude that Logistic regression works well with low number of inputs whereas SVM performs well with high dimensional input data. In Logistic regression, all the samples in the data set determine the decision boundary, whereas in SVM only the points near the decision boundary influences it. As our input data has significant features, SVM fares better than logistic regression.