CSE574 Introduction to Machine Learning

Programming Assignment 3

Classification and Regression

Group - 25

Harika Reddy Kumbum – 50248882

Sai Nikhila Muthyala – 50245184

Implementation of Logistic Regression

One-vs-All Strategy:

On training the preprocessed MNIST data with logistic regressor, the observed accuracies are:

Training set Accuracy: 86.176%

Validation set Accuracy: 85.25%

Testing set Accuracy: 85.41%

Category wise error representation for Train and Test data:

Confusion matrix for Training data:	**Confusion matrix for Test data:**
[[**4826** 1 14 7 9 21 27 7 7 4] [2 **5650** 32 12 3 17 4 11 4 7] [34 42 **4582** 70 55 24 56 64 14 17] [22 25 130 **4641** 9 162 22 43 11 66] [8 20 26 6 **4555** 13 28 12 4 170] [43 18 32 135 50 **3956** 85 16 30 56] [23 13 29 2 19 69 **4743** 3 14 3] [13 22 48 11 44 11 3 **4970** 3 140] [118 272 642 895 176 1079 105 48 **742** 774] [28 27 16 87 163 41 1 154 9 **4423**]]	[[**961** 0 1 2 0 4 5 3 4 0] [0 **1121** 4 1 0 2 5 1 1 0] [9 9 **944** 20 9 4 13 14 5 5] [4 1 22 **925** 2 26 4 13 0 13] [1 3 5 2 **915** 0 11 2 1 42] [10 4 4 37 12 **780** 20 10 6 9] [9 3 6 1 6 20 **910** 0 3 0] [2 9 19 6 8 2 1 **950** 3 28] [33 37 116 175 47 225 35 21 **130** 155] [9 8 1 17 35 11 1 22 0 **905**]]

Analysis:

Since the training model is developed on the training data, we get less error for train data as it over fits the model. Whereas, training the test data with the above model may be less accurate in this case. Also the size of train data is always more than test data in our experiment, hence the test data has less chances to be more accurate with the true labels.

Multi-class Logistic Regression

Multi-class Classification

On training the preprocessed MNIST data with multi-class logistic regressor, the observed accuracies are:

Training set Accuracy: 93.39%

Validation set Accuracy: 92.43%

Testing set Accuracy: 92.67%

Category wise error representation for Train and Test data:

Confusion matrix for Training data:	**Confusion matrix for Test data:**
[[**4789** 1 11 8 11 28 30 8 33 4] [1 **5612** 27 14 5 18 2 12 42 9] [19 43 **4533** 75 56 18 51 54 94 15] [16 19 99 **4673** 4 132 14 41 92 41] [7 22 23 5 **4553** 6 43 15 28 140] [40 15 41 134 35 **3935** 63 17 106 35] [23 13 34 1 30 50 **4745** 2 18 2] [6 17 48 19 34 9 4 **4979** 11 138] [20 87 62 122 16 109 24 16 **4341** 54] [18 22 10 60 115 30 2 123 34 **4535**]]	[[**959** 0 0 3 1 7 5 4 1 0] [0 **1110** 4 2 0 1 4 2 12 0] [4 8 **938** 14 10 3 13 9 30 3] [4 1 20 **919** 1 21 3 12 23 6] [1 2 7 2 **921** 0 8 4 7 30] [10 2 4 43 10 **769** 13 6 28 7] [9 3 5 3 7 16 **912** 2 1 0] [1 9 20 7 7 1 0 **950** 2 31] [9 8 7 27 9 23 10 8 **862** 11] [11 8 1 9 23 5 0 18 7 **927**]]

Analysis:

Since the training model is developed on the training data, we get less error for train data as it over fits the model. Whereas, training the test data with the above model may be less accurate in this case. Also the size of train data is always more than test data in our experiment, hence the test data has less chances to be more accurate with the true labels.

Since in multi-class regression, we use one classifier to classify all 10 classes, the accuracy is reported to be more for train and test data compared to One-vs-All strategy. One-vs-all binary classifier is inefficient if we have to classify with many classes and when there are class imbalances.

Support Vector Machines

The following are the accuracies observed from the experiments of svm:

Linear Kernel (with other parameters set to default):

Training set Accuracy: 97.286%

Validation Accuracy: 93.64%

Testing Accuracy: 93.78%

Radial basis function with gamma = 1 and all other parameters set to default:

Training set Accuracy: 100.0%

Validation set Accuracy: 15.48%

Testing set Accuracy: 17.14%

Radial basis function all other parameters set to default(gamma = 0):

Training set Accuracy: 94.294%

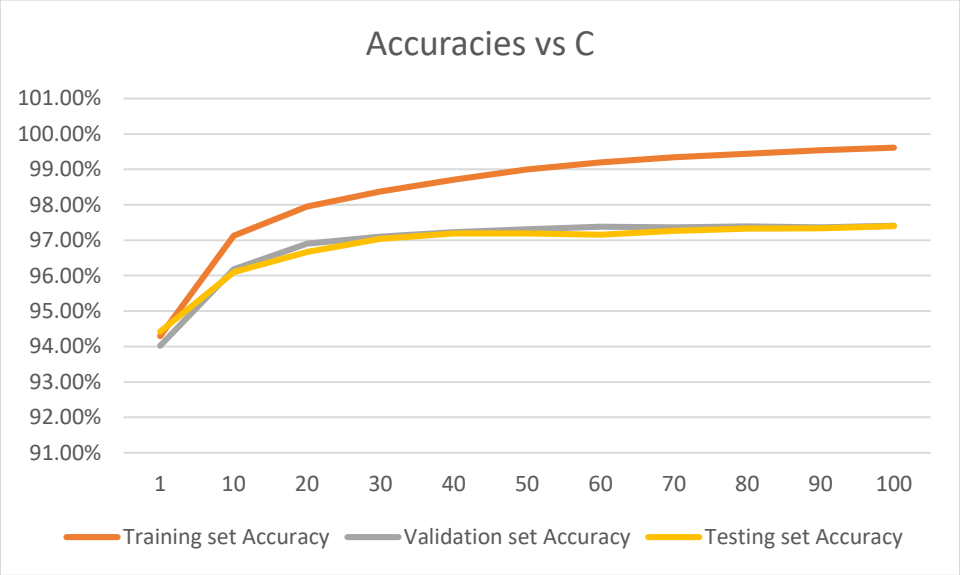
Validation set Accuracy: 94.02%

Testing set Accuracy: 94.42%

Radial basis function with value of gamma set to default and varying value of C (1; 10; 20; 30 ;……; 100)

C	Training set Accuracy	Validation set Accuracy	Testing set 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%



Analysis:

From the plot, as C value increases, accuracies increase. This is because when C increases, the weight of each error term increases and hence smaller error values will be considered. This results in a smaller margin hyper plane and so fewer points will be misclassified which results in higher accuracy.

Linear vs Rbf Kernels

From the above accuracy values, it can be observed that svm gives higher accuracy when a Rbf kernel is used.

Solving an optimization problem for a linear kernel is much faster compared to nonlinear kernel like rbf but the best possible predictive performance is better or at least equal for a nonlinear kernel. Also linear SVM is less prone to overfitting than non-linear.

In our case, since the number of features is large, we need not map data to higher dimensional space. So nonlinear mapping does not improve the performance. Instead we can use a linear kernel as there isn't much difference between accuracies and run time is less for linear kernel.

Gamma effects

Gamma is related to the variance of gaussian in rbf kernel. Higher the value of gamma, lower is the value of variance of gaussian and hence lower is the influence of training samples selected as svms.

From the results, when gamma value is set to 1, the validation and testing accuracies are very low even when training accuracy is 100% because the model has adjusted too much with training data resulting in overfitting.

When gamma value is 0, validation and testing accuracies are much higher. It can also be observed that accuracies obtained from rbf with gamma=0 are higher than accuracies obtained from linear. Whereas they are observed to be much lower for gamma=1 because of over fitting.