CSE 474/574
INTRODUCTION TO MACHINE LEARNING
PROJECT ASSIGNMENT 3 REPORT

Group Number: 86

Members:

Arshabh Semwal(Person No. 50419031)
Utkarsh Bansal(Person No. 50415035)

1. Introduction

As part of the project assignment 3, we are given the MNIST dataset for image classification. Our task is to classify the given data using following techniques:-

1. Training 10 binary logistic regression models.
2. Using Support Vector Machines.
3. Training a multiclass logistic regression model.

2. One vs All Logistic Regression

In this section, we train 10 classifiers where each classifier is responsible to classify 1 digit from the rest. For example, one classifier will distinguish the digit '1' from the rest of the data, similarly for '2' and so on.

The training, validation and test error for each digit is given in the table below:-

Digit	Training Error	Validation Error	Test Error
0	0.0203	0.0257	0.0264
1	0.0200	0.0343	0.0250
2	0.0623	0.0727	0.0734
3	0.0753	0.0878	0.0717
4	0.0444	0.0506	0.0511
5	0.0827	0.0802	0.0839
6	0.0339	0.0400	0.0381
7	0.0436	0.0494	0.0542
8	0.1100	0.1212	0.1127
9	0.0973	0.0958	0.1021

The above data tells us the performance of 10 different classifiers on their respective digit. From this we can infer that the classifier of digits 0 and 1 performed the best as compared to other digit's models (lowest error). Also, the classifier responsible for digit 8 performed the worst out of the bunch (highest error). The overall accuracy of the model is given in the table below.

Dataset	Accuracy
Training Data	92.732%
Validation Data	91.43%
Test Data	91.93%

Another thing we can infer from Table 2.1 and 2.2 is that the model generally does better on the training data as compared to testing or validation data. This is because the training data was used in the first place to train the model, so it is very likely that the model does better on it as compared to the other datasets. If the model had been trained using the testing data, then it would have done better on the testing data as compared to training or validation data.

3. Multiclass Logistic Regression

In this section, instead of using 10 different classifiers, we will use a single multiclass classifier which can classify data in 10 different categories. We do this with the help of the softmax function.

As only one model is trained to classify all the 10 classes, only one error is obtained which is the combined error of all classes. The following table shows the errors and the accuracy on training, validation and test data:-

Dataset	Error	Accuracy
Training Data	0.2378	93.44%
Validation Data	0.2770	92.48%
Test Data	0.2686	92.55%

The error reported above has been normalized by dividing the error by the number of data points. The actual error received after using the softmax function was 11890 for training data, 2770 for validation data and 2686 for test data.

4. Comparison between One vs All Logistic Regression and Multi-class Logistic Regression

When the results achieved by both the methods are compared then we see that Multiclass Logistic Regression has performed better than One vs All binary classifiers. This is because wrong classifications can happen in the one vs all method. This is why overlapping can be seen. Whereas in multiclass classification, overlapping cannot occur.

The one vs all strategy takes a considerable amount of time for training as compared to multiclass classification strategy. This is because in one vs all method, we are training 10 classifiers one by one. Whereas in the multiclass method we only train one classifier.

5. Support Vector Machine (SVM)

Case 1. Using a linear kernel (all other parameters are kept default).

Kernel	C Value	Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
linear	1	99.56%	91.85%	93.35%

Case 2. Using radial basis function with value of gamma setting to 1 (all other parameters are kept default).

Kernel	C Value	Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
rbf	1	100%	15.0499999999 99999%	16.8%

Case 3. Using radial basis function with value of gamma setting to default (all other parameters are kept default).

Kernel	C Value	Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
rbf	1	98.6199999999 9999%	96.15%	96.3%

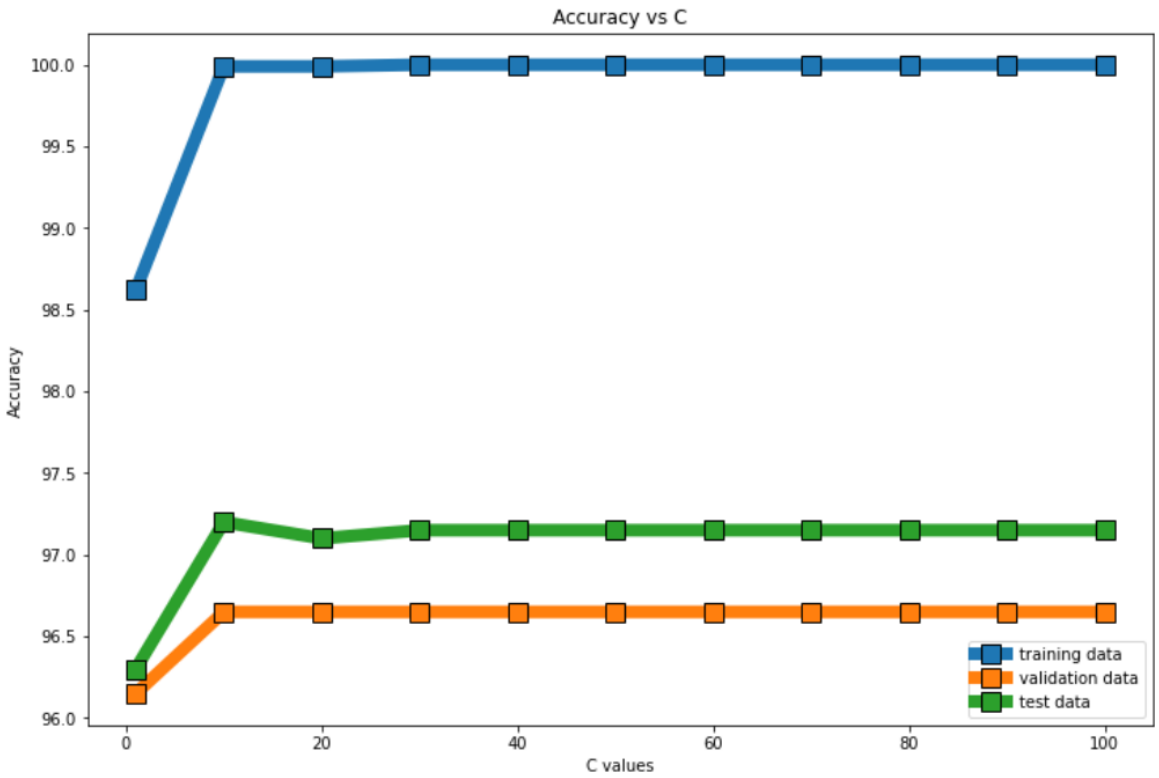
Case 4. Using radial basis function with the value of gamma setting to default and varying value of C (1; 10; 20; 30;...; 100) and plot the graph of accuracy with respect to values of C in the report.

Kernel	C Value	Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy
rbf	1	98.619999999999999	95.15	96.3
rbf	10	99.99	96.65	**97.2**
rbf	20	99.99	96.65	97.1
rbf	30	100	96.65	97.15
rbf	40	100	96.65	97.15
rbf	50	100	96.65	97.15
rbf	60	100	96.65	97.15
rbf	70	100	96.65	97.15
rbf	80	100	96.65	97.15
rbf	90	100	96.65	97.15
rbf	100	100	96.65	97.15

Conclusion:

The linear kernel performs well on the handwritten data since the data can be separated linearly. But using the RBF kernel gives higher accuracy, which can allow us to classify digits more accurately.

The gamma value inversely defines how much the influence of a single training data with low levels signifies a large influence and high value indicates low influences over the neighboring points. A small gamma corresponds to a low bias and high variance. A very high value of gamma can lead to overfitting as is evident in the case when we set the value of gamma as 1 in the rbf kernel.



(Accuracy of SVM with rbf kernel, default value of gamma and value of C in [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100] VS C values.)

The regularization parameter, C, determines the balance between the complexity of the decision rule and the frequency of error. A smaller value of C will lead to the SVM making more mistakes (underfitting) while increasing C will make the SVM perform better at the risk of overfitting. However, the training time as well as the number of support vectors will also increase with increasing C.

On a sample size of 10,000 dataset the accuracy increases from C value 1 to 20 (for validation and test) and then becomes flat as value of C is increased further. This concludes that changing C further doesn't significantly impact the accuracy of SVM with rbf kernel and default value of gamma.

We choose the value C as 20 along with kernel rbs and default value of gamma for training over the entire data as for all the C = 20 along with kernel rbs and default value of gamma the SVM has the highest accuracy (see case 4 table) so it is a good choice to take C = 20 with kernel rbs and default value of gamma to find the accuracy over the entire dataset.

Accuracy on SVM with kernel = rbf, gamma as default and C = 20 is:

Kernel	C Value	Training Data Accuracy	Validation Data Accuracy	Test Data Accuracy

rbf	20	100%	98.44000000000001%	98.31%