Logistic Regression and Support Vector Machines

Programming Assignment No. 3
CSE574 Introduction to Machine Learning

Ashwini Aher, Pratik Patil

Department of Electrical Engineering
{aaher, ppatil2}@buffalo.edu
State University of New York at Buffalo

May, 10 2018

Contents

Chapter 1

Logistic Regression

1.1 Logistic Regression

In logistic regression output is binary i.e. it has only two outcome either true or false. We had to implement logistic regression to classify the handwritten digit images into respective predicted digit label for that image. We are using the same dataset as that of we used in neural network assignment. Logistic regression is mainly used for the binary classification which identify that whether given input belongs to particular class or not. In this case, we have to classify the digit in 10 different classes. So, simple logistic regression will not work here and hence we have to implement 10 classifier for each digit. Each classifier will convey that whether the given image belongs to particular digit or not.

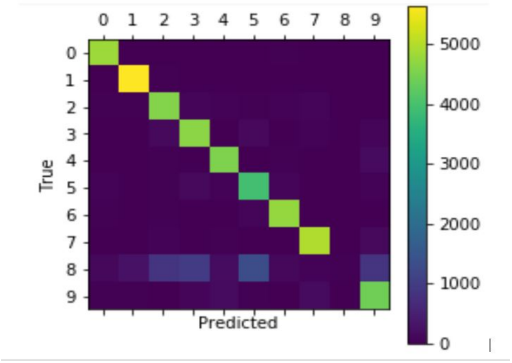
1.1.1 Results

Digit Wise Classifier Accuracy

As shown in figure and and matrix graph we can clearly see that the digit one is predicted with highest percentage of 98.43% and digit eight images were predicted with lowest 0.7% in case of training dataset. The best classifier was for digit 0 with 94.1% success rate over all digit images and worst for digit 8 with 69.3%, this is also in case of training data set. Most of the digit 8 images were predicted as digit 5 in case of training dataset. A similar kind of scenario can be found in testing dataset as well. Highest prediction success rate is for digits 1 labels with 98.8%, and lowest for digit 8 images with 0.1%. Digit one classifier worked best among all classifiers with 93.2%. While digit eight classifier worked worst with 50% success rate. Most of the digit 8 images were predicted as either digit five, three or nine, in ascending order.

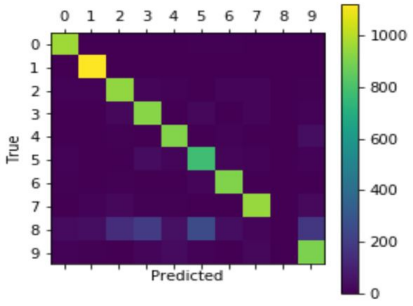
Confusion Matrix and Matrix graph for training data.

4830	1	14	9	10	20	27	6	2	4
2	5652	33	13	3	18	4	11	0	6
33	39	4596	72	48	29	56	67	1	17
19	21	137	4664	9	148	19	46	1	67
6	20	27	6	4565	12	26	15	0	165
47	19	38	139	45	3959	90	17	8	59
25	11	32	3	19	68	4751	3	2	4
9	21	47	10	42	15	4	4981	0	136
128	282	848	999	187	1344	117	51	34	861
30	24	16	90	164	44	1	155	1	4424



Confusion Matrix and Matrix graph for test data.

961	0	1	3	1	5	6	2	0	1
0	1123	4	1	0	2	4	1	0	0
10	11	941	20	10	4	14	16	0	6
4	1	23	924	2	25	4	14	0	13
1	2	6	3	915	0	11	2	1	41
10	4	4	39	14	779	21	9	0	12
8	4	8	2	4	20	911	1	0	0
2	10	23	5	7	1	1	951	0	28
35	41	148	198	51	260	40	20	1	180
10	8	1	18	34	12	1	23	0	902



Click here to access digit wise percentage for all datasets.

Overall Accuracy

Dataset	Accuracy
Training Set	0.84912
Validation Set	0.837
Test Set	0.8408

Table 1.1: Accuracy for Logistic Regression over all data sets

1.1.2 Observations

1. The training set has the highest accuracy as we have used the same data to train the algorithm and weights are optimized using this dataset.
2. Validation set has the lowest accuracy among all datasets, but not too small.
3. Test dataset has the accuracy lower than the training set but not by huge difference.

Chapter 2

Support Vector Machines

We had to implement Support Vector Machines (SVM) by using the inbuilt function of sklearn libraries. We had to learn the model and fit the learned SVM model on training, validation and test dataset. While implementing the SVM we considered the following cases.

1. Linear Kernel Model
2. RBF model with Gamma set to 1
3. RBF model with default gamma
4. RBF model with C ranging from (1, 10, 20, ..., 100)

2.1 Linear Kernel Model

The linear kernel SVM are useful when the data points are linearly separable and accuracy of model increases if the data is linearly separated. For the MINST dataset we got the following results.

2.1.1 Results

Dataset	Accuracy
Training Set	0.97286
Validation Set	0.9364
Test Set	0.9378

Table 2.1: Accuracy for Linear SVM Kernel over all data sets

2.1.2 Observations

1. From the above results we can infer that the gievn training dataset is most linearly separable among training, validation and test dataset.

2. Validation and test dataset are not quite as linearly separable as training dataset.

3. The accuracy for the training dataset is highest while accuracy for validation and test dataset is lower but not worst.

4. Thus, we can say that MINST data set is very well linearly separable with the results that we have observed.

2.2 RBF model with Gamma = 1

We have implemented RBF kernel with the parameter Gamma equal to 1. In RBF kernel gamma and C are the two main important parameters. The default value of C is 1 while the default value of Gamma is inverse of number of features. In the MINST dataset number of features are very large hence Gamma has to be very low but we are setting Gamma equal to 1 which is very large value. Gamma decides the similarity between the two points. Similarity between points decreases, as gamma increases. The curve falls sharply and the similarity distance decreases and vice versa for small gamma values.

2.2.1 Results

Dataset	Accuracy
Training Set	1
Validation Set	0.1714
Test Set	0.1548

Table 2.2: Accuracy for RBF Kernel over all data sets

2.2.2 Observations

1. As we have mentioned above the number of features for the given dataset are very high. Thus, the gamma value has to be small but we have to set it to 1 which is very large as compare to the required gamma value.

2. This large gamma value led to the over fitting issue on the training dataset. Hence the accuracy with respect to training dataset is 1.

3. But over fitting results into poor accuracy for other datasets, namely validation and test dataset.

2.3 RBF model with default gamma

RBF function is a Gaussian function which maps the data from current feature dimension to infinite dimensions. Sometimes it is difficult to find a linear separable boundaries in the current feature dimension of the data. In such case, RBF can be used for mapping into infinite dimension so that we can find a linear

line or a hyper plane to separate the data into different classes. We get nonlinear boundaries around different classes of the dataset when we bring back the separation line or the hyper plane to original dimension from infinite dimension. Here we have used the default values of C and gamma i.e. C and gamma equal to 1.

2.3.1 Results

Dataset	Accuracy
Training Set	0.94294
Validation Set	0.9442
Test Set	0.9402

Table 2.3: Accuracy for RBF Kernel over all data sets

2.3.2 Observations

1. The accuracies for training , validation and test data are almost the same.

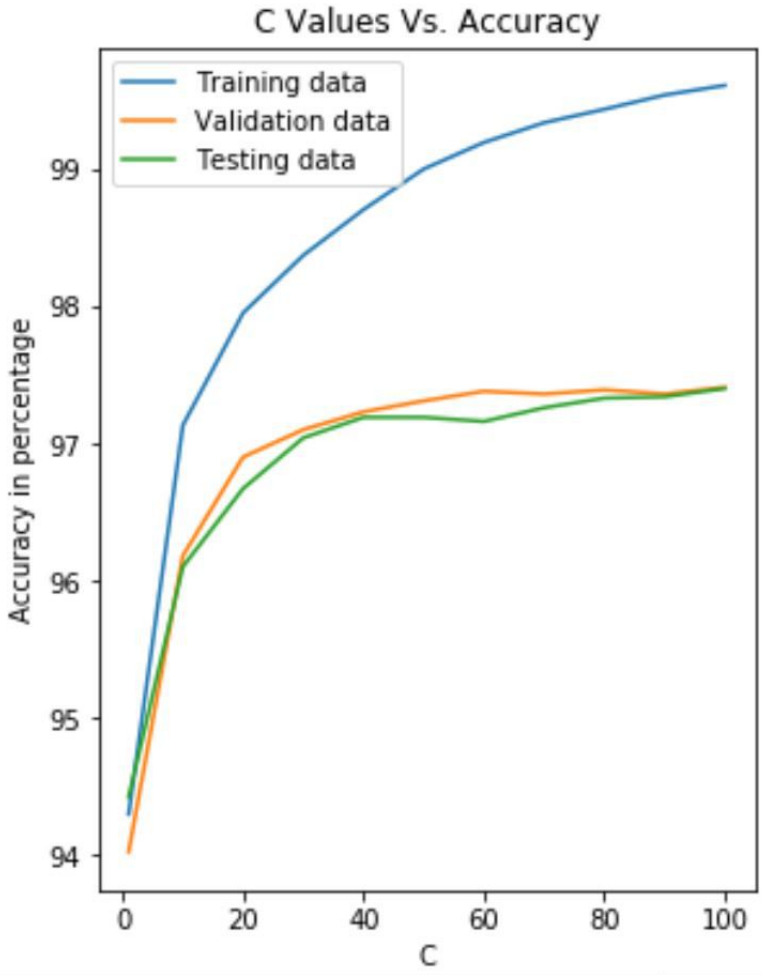
2.4 RBF model with varying values of C

While training the SVM kernel with RBF, we are concentrating on two parameters (namely, C and gamma). C decides that whether the boundaries will be smoother or tighter across training data and gamma decides the similarity between the neighboring data points. As the value of C increases, the boundaries between classes becomes more and more nonlinear where as boundaries are smoother for lower values of C. As the boundaries become more and more nonlinear, the classification accuracy also increases. We have observed this scenario in the results that we have obtained with increasing value of C.

2.4.1 Results

C	Training Data Set	Validation Data Set	Test Data Set
1	0.94294	0.9402	0.9442
10	0.97132	0.9618	0.961
20	0.97952	0.969	0.9667
30	0.98372	0.971	0.9704
40	0.98706	0.9723	0.9719
50	0.99002	0.9731	0.9719
60	0.99196	0.9738	0.9716
70	0.9934	0.9736	0.9726
80	0.99438	0.9739	0.9733
90	0.99542	0.9736	0.9734
100	0.99612	0.9741	0.974

Table 2.4: Accuracy for RBF Kernel with varying values of C over all data sets



2.4.2 Observations

1. If the data is linearly separable then Linear kernels are preferred over RBF kernels. Because linear kernels take less time to train the model as compared to RBF kernel.
2. The accuracy that we have got with linear kernel SVM is not so bad hence; we can say that MINST dataset is somewhat linearly separable with very less error.
3. Accuracy increases, if we run the RBF SVM kernel. However, we have seen that it takes hardly 10 minutes to train linear SVM kernel while it takes around 3 hours to train the RBF SVM kernel.
4. Hence, we can conclude that, RBF SVM kernel is definitely better than linear SVM kernel in terms of accuracy but the time required to train the RBF model is significantly more than the time required to train the linear SVM model.
5. The accuracy is more for training, validation and test data in case of RBF kernel than linear kernel.

Chapter 3

Direct Multi Class Logistic Regression

Logistic Regression performs the binary classification. But sometimes data may belong to one of k classes. For example, given image might represent a dog, a lion or a cat. Thus, for given k classes, multiclass classification problem can be solved by training k different binary classifiers, and predicting that example x belongs to class i which has highest probability. Softmax function can be used to make all values positive and then normalize so that all values sum to 1. This model is also called as Softmax regression or multinomial regression.
We have implemented 100 different classifiers for each digit in logistic regression but in case of multi class logistic regression we will implement only one classifier which will classify image and assign label to to it only in one iteration.

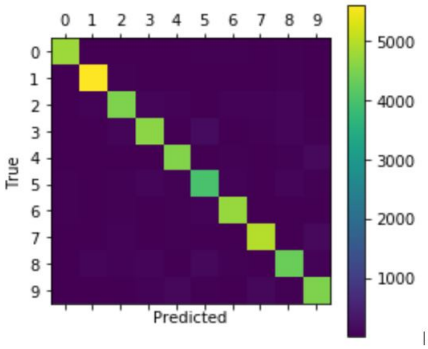
3.1 Results

3.1.1 Digit wise Accuracy

Confusion matrix for training data and test data can be found below. It can be observed that the prediction accuracy is significantly improved over logistic regression. In case of training data digit one has highest prediction rate with 97.4%, since there are not many ways to write digit one, while prediction rate is worst in case of digit 8 images, which is around 89.6%. In case of test data we found the same results that digit one has the highest rate of prediction. Also Over all accuracy of multi class logistic regression is also improved over the binary logistic regression.

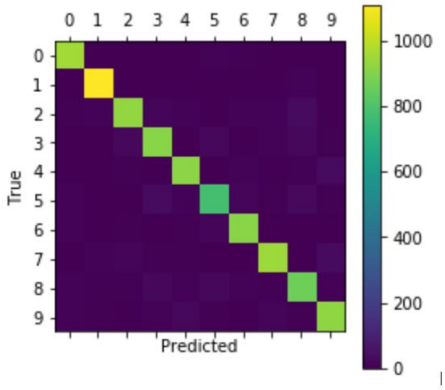
Confusion Matrix and Matrix graph for training data.

4787	1	11	7	11	34	31	8	29	4
1	5612	27	13	6	18	2	12	43	8
21	46	4528	68	56	21	53	53	98	14
15	21	101	4628	4	169	14	37	100	42
9	20	21	4	4562	5	41	14	29	137
36	11	38	100	34	4005	62	14	92	29
23	11	33	1	29	54	4745	3	16	3
6	17	48	18	36	9	4	4981	13	133
21	88	59	102	18	127	24	15	4350	47
17	22	10	50	118	33	2	124	36	4537



Confusion Matrix and Matrix graph for test data.

957	0	0	3	1	9	5	4	1	0
0	1111	4	1	0	1	3	2	13	0
5	9	933	14	10	3	13	9	33	3
4	1	22	914	1	24	3	11	24	6
1	2	6	2	920	0	9	4	7	31
10	2	3	33	9	782	14	5	27	7
9	3	6	1	7	16	912	3	1	0
1	9	19	7	6	2	0	949	2	33
9	8	7	25	9	24	10	8	864	10
11	8	0	9	24	6	0	19	7	925



Click here to access digit wise percentage for all datasets.

3.1.2 Overall Accuracy

Dataset	Accuracy
Training Set	0.9347
Validation Set	0.9234
Test Set	0.9267

Table 3.1: Accuracy for Multi Class Logistic Regression over all data sets

3.2 Logistic Regression Vs. Multiclass Logistic Regression

Dataset	Accuracy	
	Logistic Regression	Multiclass Logistic Regression
Training Set	0.84912	0.9347
Validation Set	0.837	0.9234
Test Set	0.8408	0.9267

Table 3.2: Comparison between Logistic Regression and Multiclass Logistic Regression

3.3 Observations

1. The accuracy of training data set is highest among training, validation and test data set, which is expected since weights will be trained using this data.

2. But accuracy of validation and test data set is also do not much deviate from the accuracy of the training data set. Its slightly lower than training dataset but not much.

3. From the Table 3.2 we can easily conclude that multi class logistic regression gives much better accuracy than logistic regression.

4. On the other hand we can observe that logistic regression takes less time to find train the algorithm than the multi class logistic regression.