CSE 574 INTRODUCTION TO MACHINE LEARNING

PROGRAMMING ASSIGNMENT 3

Classification and Regression

By Group 38-

Tulika Sengupta (tulikase@buffalo.edu)

Darshan Prakash Mhatre(dmhatre@buffalo.edu)

Divyansh Bhatnagar(dbhatnag@buffalo.edu)

LOGISTIC REGRESSION

Classification Model	**Dataset Accuracy**		
	Training Set	**Validation Set**	**Testing Set**
Logistic Regression	**84.96%**	**83.74%**	**84.11%**
Multi-Class Logistic Regression	**93.09%**	**92.50%**	**92.52%**

Table-1 Logistic Regression VS Softmax Regression

Logistic Regression is a linear model for classification of binary data. The classification is done by projecting the data points on a pre-specified set of features and then finding separating hyperplanes in this feature-space.

For the given set of data we employ one-vs-all strategy, i.e. we build 10 binary-classifiers to distinguish the given class from all other classes.

The accuracy for this model is not very high because the binary-classifiers work best in datasets which are not mutually exclusive.

On the other hand a Softmax classifier or a Multi Class Logistic Regression is a generalized form of Logistic Regression, where the features can take more than two possible values. For $k = 2$, this works the same as Logistic Regression.

For the given dataset we don't have to build 10 classifiers for each class, instead, we make just 1 classifier which can classify 10 classes at the same time.

As seen from the above table the accuracy for Logistic Regression is not very high, this is due to the fact that logistic regression works best with dataset which are not mutually exclusive, on the contrary, Multiclass regression or softmax regression works well with mutually exclusive dataset such as this one.

SUPPORT VECTOR MACHINE

SVM is a classifier which is defined by Discriminating hyperplanes which we are using here to perform classification on Handwritten dataset.

We computed the accuracy of the training, validation and testing data using score function for various parameters of the SVM.

- **Using linear kernel (all other parameters are kept default)**

Dataset	**Accuracy**
Training	**97.286**
Validation	**93.64**
Testing	**93.78**

Table-2 SVM using linear kernel

Linear kernels are always faster comparing to non linear kernels like radial basis function kernels. And most of the time, for high dimensional data like ours handwritten dataset,simply using linear kernel is sufficient for obtaining desired prediction accuracy

- **Using radial basis function**
- with value of gamma setting to 1 (all other parameters are kept default)

Dataset	**Accuracy**
Training	**100**
Validation	**15.48**
Testing	**17.14**

Table-3 Using radial basis function

-with value of gamma setting to default (all other parameters are kept default)

Dataset	**Accuracy**
Training	**94.294**
Validation	**94.02**
Testing	**94.42**

Table-4 Default gamma value

Radial basis function is the non linear kernel which is resource exhaustive and more time consuming, but gives more accurate prediction than linear kernels.

A well tuned RBG kernel will always give better accuracy than linear SVM. But, we have to adjust different parameters properly.

For example, when we fit the value of gamma, we should know that gamma defines the level of influence of training data which we used for fitting model, onto the final prediction. Thus, when we put gamma =1 , it meant complete influence of our training data on the prediction, which led to the problem of overfitting. And that is evident from the values we obtained which are tabulated above.

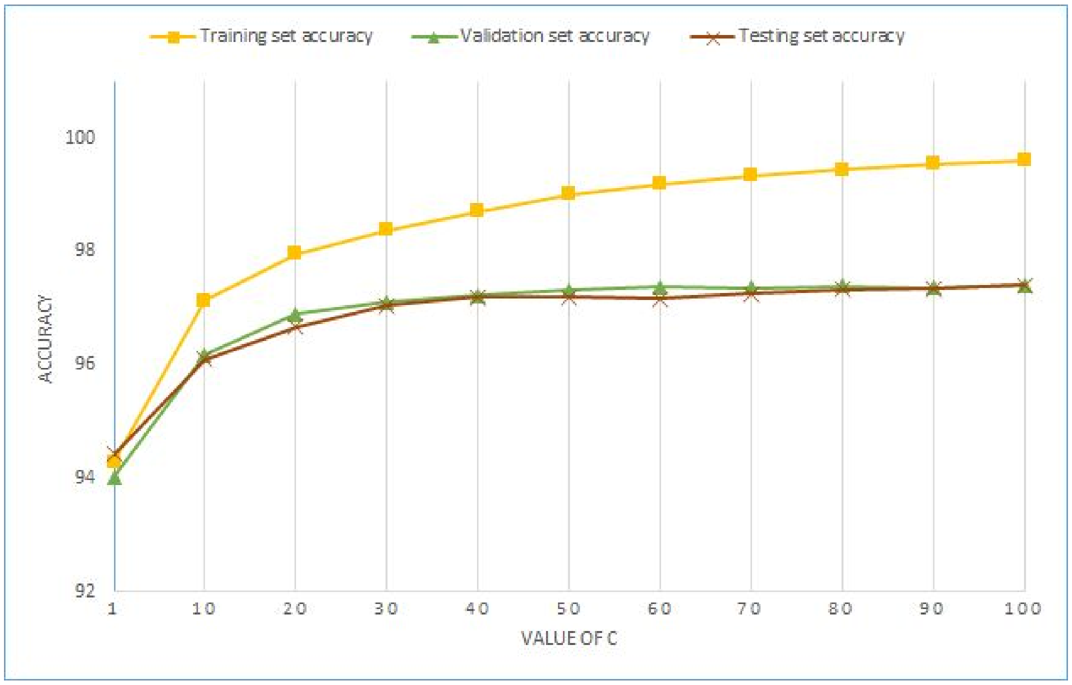
For gamma=1, we got 100% accuracy i.e. we overfitted the model, because of which we got very poor accuracies in validation and testing data.

But, when we fitted the gamma parameter properly we observed better accuracy in RBG than linear kernel.

- **Using radial basis function with value of gamma setting to default and varying value of C (1, 10, 20, 30, · · · , 100)**

Value of C	**Training set**	**Validation set**	**Testing set**
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	99.19	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

Table-5 Varying value of C



C parameter is basically used to control degree of avoiding misclassification rate. It is adjusted in order to achieve the desired balance between low training error vs low testing error. Value of C determines the size of hyperplane margin that will be chosen by optimization.

We varied the values of C regularization parameters in order to avoid misclassification rate and ultimately to improve the accuracy. The graph is plotted as a function of various accuracies against values of C’s in the range(1,10,20,30….,100)