CSE 574 Introduction to Machine Learning

Programming Assignment 3

Classification and Regression

Members of GROUP 43

1) Arundhati Ravikumar Rao
2) Cuurie Athiyaman
3) Prithvi Ganesan

Instructor: Prof.Varun Chandola

1. Logistic Regression

Logistic Regression is a predictive analysis method where the dependent (class) variable is binomial or multinomial. It aims to find the best fitting that describes the relationship between the dependent and independent variable. Unlike normal regression, logistic regression tries to maximize the likelihood of the sample values. In our assignment we try to classify the given data point 'x' into class C1 or C2.

We use the following formula to find the error by taking negative log of the log likelihood.

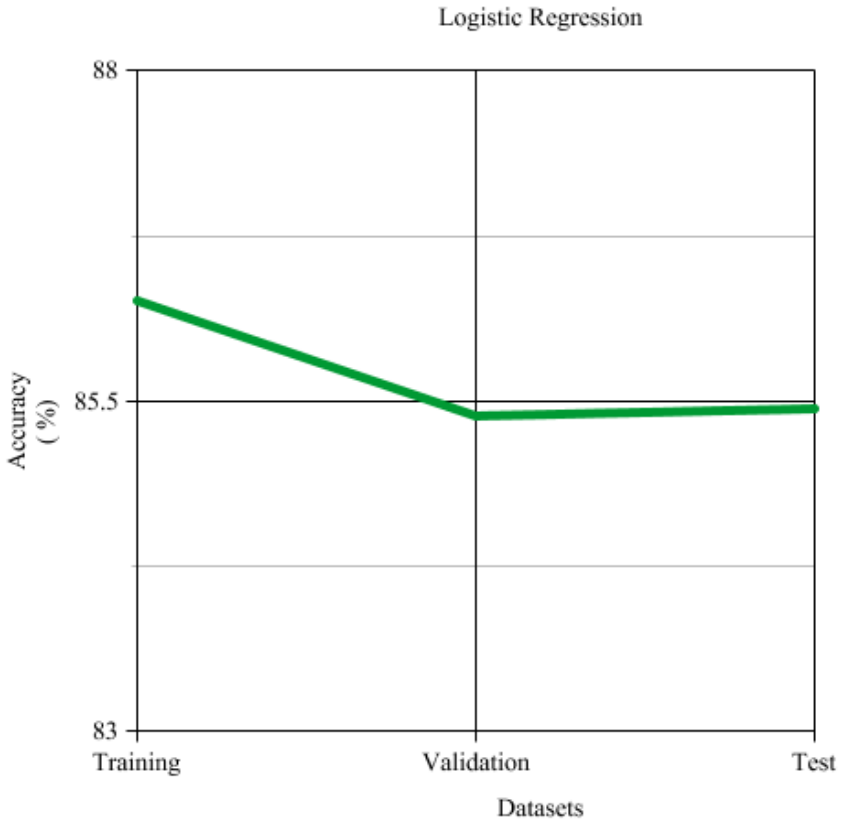
$$E(\mathbf{w}) = -\frac{1}{N}\ln p(\mathbf{y}|\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\{y_n \ln \theta_n + (1-y_n)\ln(1-\theta_n)\}$$

And the following expression is to find the gradient of the error function with respect to weight to minimize the classification error.

$$\nabla E(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(\theta_n - y_n)\mathbf{x}_n$$

Results for Logistic Regression:

Dataset	Accuracy
Train	86.26%
Validation	85.39%
Test	85.45%



Direct Multi-class Logistic Regression – Extra Credit

In the above regression method, we are building 10 separate binary classifiers to classify each data point. But it can also be implemented by building single multinomial classifier that is able to sort the data point accordingly to its specific class. Like binary logistic regression, we find the posterior probability then the likelihood.

The following formula is used to find the negative log likelihood

$$E(\mathbf{w}_1, \cdots, \mathbf{w}_K) = -\ln\ P(\mathbf{Y}|\mathbf{w}_1, \cdots, \mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk} \ln\ \theta_{nk}$$

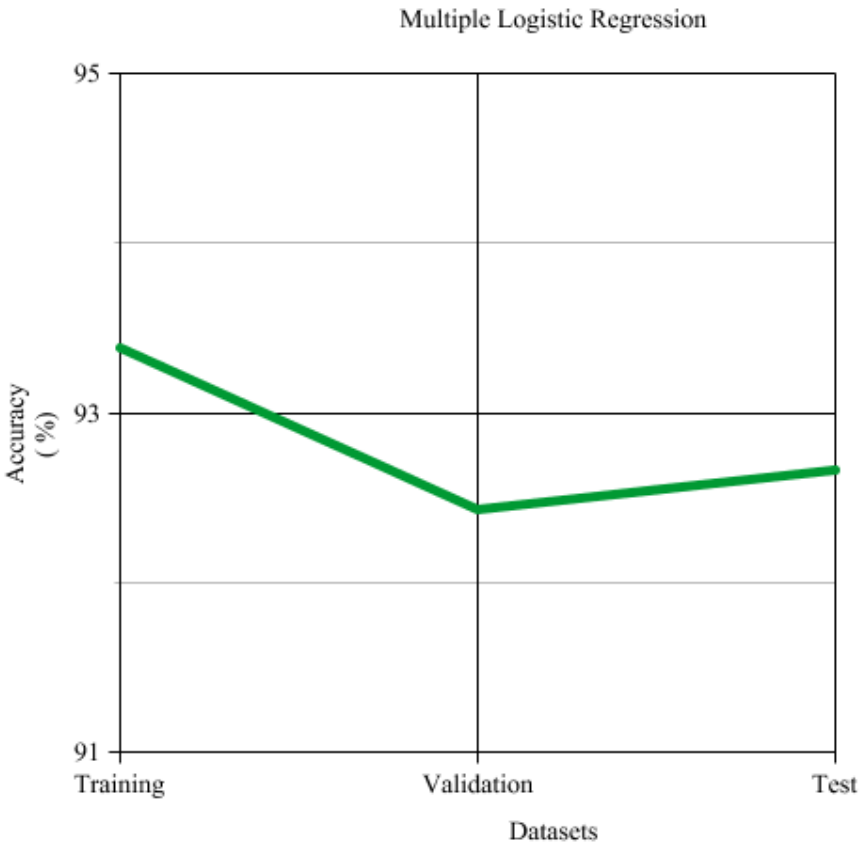
And the error gradient is found with the help of the expression below.

$$\frac{\partial E(\mathbf{w}_1, \cdots, \mathbf{w}_K)}{\partial \mathbf{w}_k} = \sum_{n=1}^{N} (\theta_{nk} - y_{nk})\mathbf{x}_n$$

The results of the above two equations is then divided by the size of training data.

Results for Multi-class Logistic Regression:

Dataset	Accuracy (%)
Train	93.39
Validation	92.43
Test	92.67



This approach has better prediction accuracy for the training data than the validation and testing data. MLR gives better prediction accuracy than BLR. Also, MLR has a lesser runtime than BLR as only one classifier for multiple classes is used, unlike the latter case.

2. Support Vector Machines

SVM, a supervised learning training algorithm, builds a model that will assign new samples to one of the categories (binomial or multinomial) by training the already existing samples. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples. The multiclass support is handled according to a one-vs-one scheme.

The SVC function has the following parameters: -

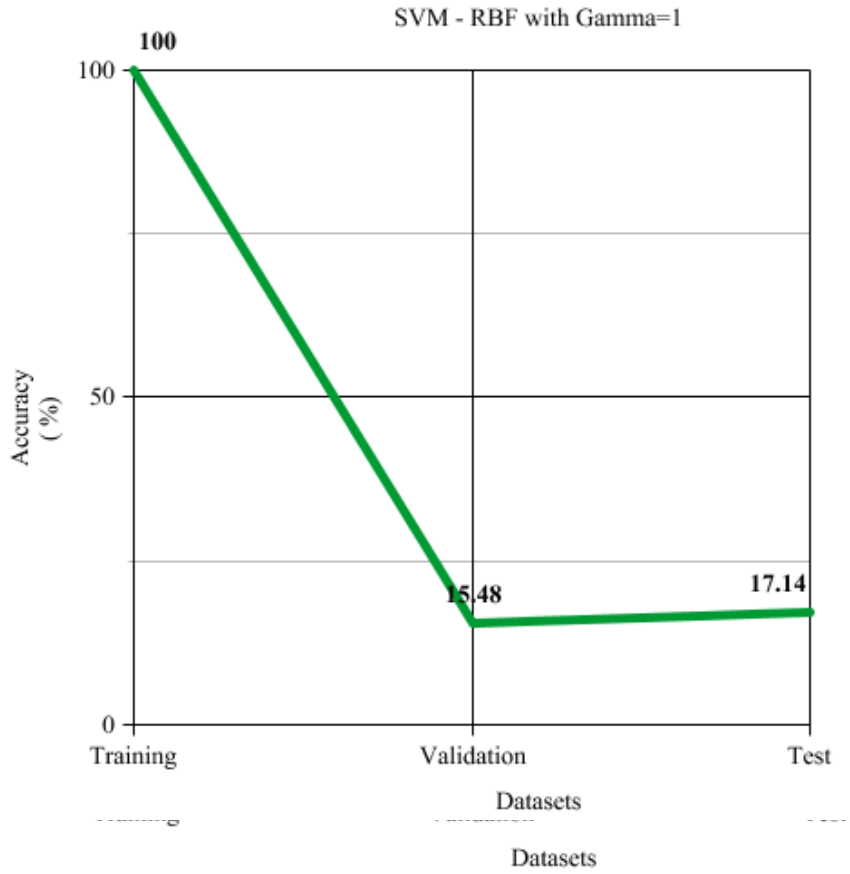
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

SVM with Linear Classifier

Using sklearn.svm.SVM, we have implemented SVM with linear classifier. This is the simplest SVM classifier. A linear classifier is learnt here which would try to correctly classify each data point to one of the 10 classes.

Results for Linear SVM:

Dataset	Accuracy (%)
Train	97.286
Validation	93.64
Test	93.78



SVM with RBF kernel and gamma = 1

To separate data points that are non-linearly separable, we move to learning in a higher dimension space for classifying our data points in a non-linear surface. This is done using the RBF kernel, which maps points to infinite dimension space. The hyper parameter gamma is inversely proportional to the variance. (i.e.) Higher the gamma value, lower the variance and vice versa.

Results for RBF kernel with gamma=1:

Dataset	Accuracy (%)
Train	100
Validation	15.48
Test	17.14

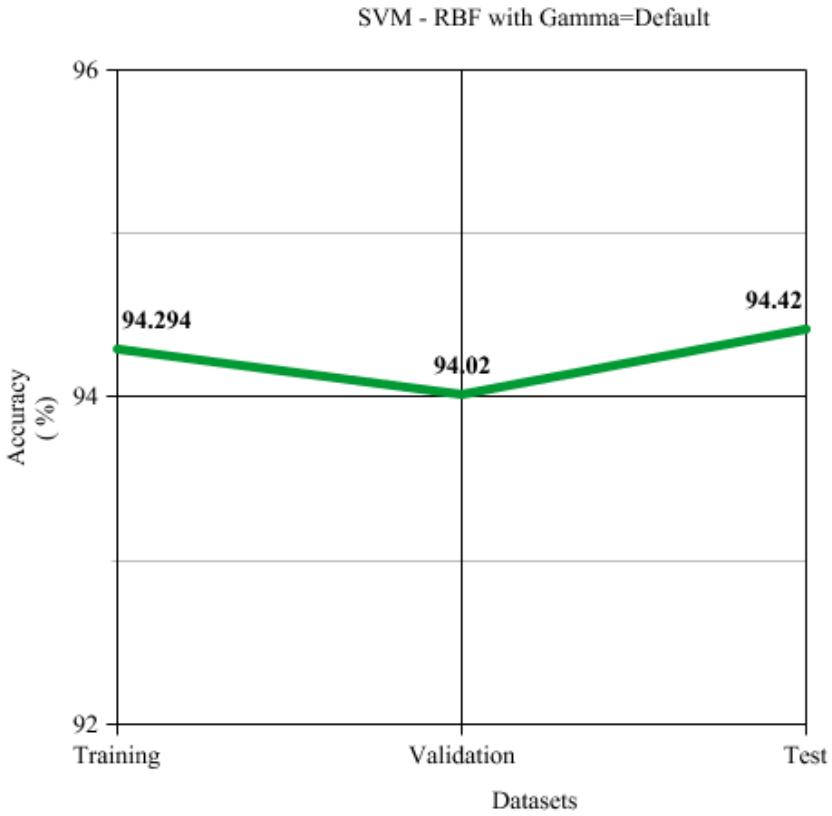
The graph above shows fairly good prediction accuracy for training data at the same time very low accuracy in terms of test and validation data. This is because, the gamma value is inversely proportional to variance as mentioned above. The variance value is assumed to be really low when a higher value for gamma is chosen. The learning model is made very complex when we learn multiple learning boundaries for training data. Poor accuracies for test and validation are seen, as the data points to be classified must lie within the boundaries which is a strict requirement for classification.

SVM with RBF kernel and default gamma

The default gamma value is given by 1/n_features. The data points were classified and the results obtained are.

Results for RBF kernel with default gamma value:

Dataset	Accuracy (%)
Train	94.294
Validation	94.02
Test	94.42



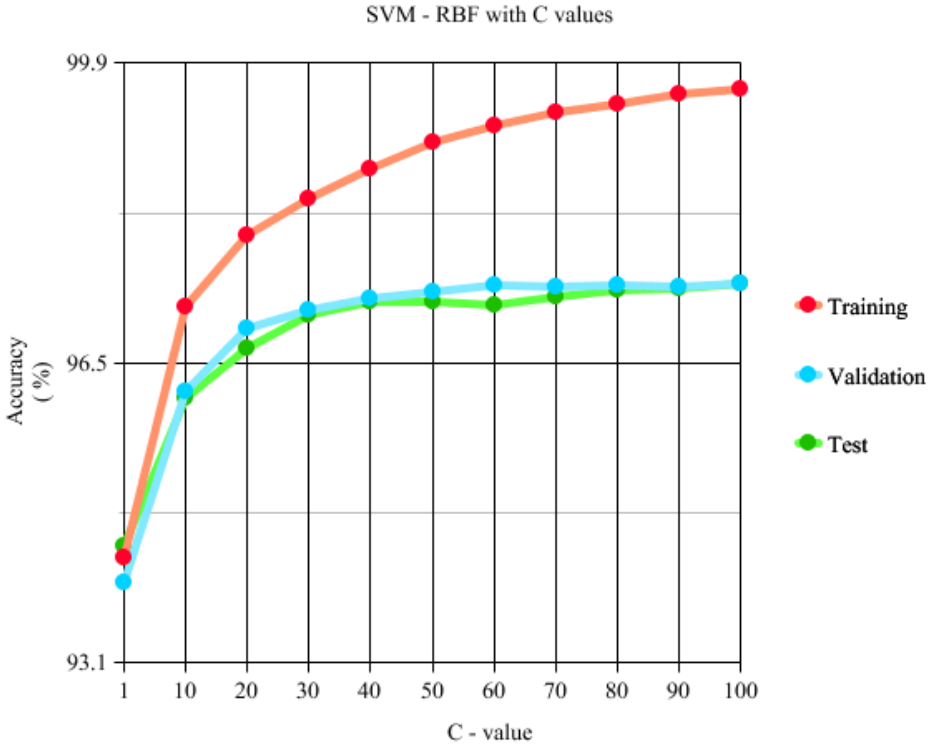
The accuracy is good for all training, validation and test datasets. By choosing a very small value for gamma, the strict classification requirements are relaxed as a higher variance value is set. This results in a less complex learning model than the previous one. As the data points are correctly classified because of high variance, accuracies for validation and test are better.

SVM with RBF kernel and gamma = default & new Hyper-parameter C

The ‘cost’ parameter is another hyper-parameter that is inversely proportional to the ‘slack’ (in SVM) that is the cause for wider margins. This is similar to the regularization parameter.

Results for RBF kernel with varying C (hyper-parameter) values:

Cost	Accuracy (%)		
	Training	Validation	Testing
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



By comparison, we can see that this model gives us the best accuracies than the previous ones. Higher the C value, better the predictions.

Even though large variance is better, this is only true to a certain point. This misclassification is rectified with the help of this hyper parameter - C.

CONCLUSION

The implementation of Logistic Regression and Classification has been used in the classification of hand-written digit images and the performances have been evaluated.