CSE 574 Introduction to Machine Learning
Programming Assignment 3
Classification and Regression

Group Number: 37
Arvind Srinivass Ramanathan (arvindsr)
Barath Eswer Nagasubramaniyan (barathes)
Senthil Kumar Laguduva Yadindra Kumar (laguduva)

Introduction

In the third programming assignment, we will be extending the first programming assignment in solving the problem of handwritten digit classification. Our focus in this assignment is to implement Logistic Regression and use Support Vector Machine tool in **sklearn.svm.SVC** to classify handwritten digit images and compare the performance of these methods.

Logistic Regression

Logistic regression is a type of probabilistic statistical classification model. Logistic regression can be broadly of two types, binary and multi-class. In the case that a dependent variable can have only two possible outputs it is a binary logistic regression or binary logistic regression. Multi-class regression is when the dependent variable can assume more than two values.

Results Binary Logistic Regression:

Training Data Accuracy	86.21%
Validation Data Accuracy	85.31%
Test Data Accuracy	85.39%

The binary logistic regression upon classification generates an accuracy of **85.39%** on the test data.

Results Multi-class Logistic Regression:

Training Data Accuracy	93.39%
Validation Data Accuracy	92.43%
Test Data Accuracy	92.67%

The multi-class logistic regression upon classification generates an accuracy of **92.67%** for the test data.

Multi-class Logistic Regression vs Logistic Regression using One-vs-all Strategy

The performance of multi-class logistic regression or the **one-vs-all strategy** will be better because within the space, the classification of the different images can be noted to be within independent regions. In the case of binary distribution, or **one-vs-rest strategy** we cannot obtain this accuracy because the space will not be clearly differentiated for all observations. There will be a class overlap thus some observation can fall under more classifications. The class definition cannot clearly indicate for all observations in the space for the classification in case of binary logistic regression. However, multi-class logistic regression will have all have clear boundaries defining the observations into distinct regions. **Thus one-vs-all strategy performs better than one-vs-rest strategy.**

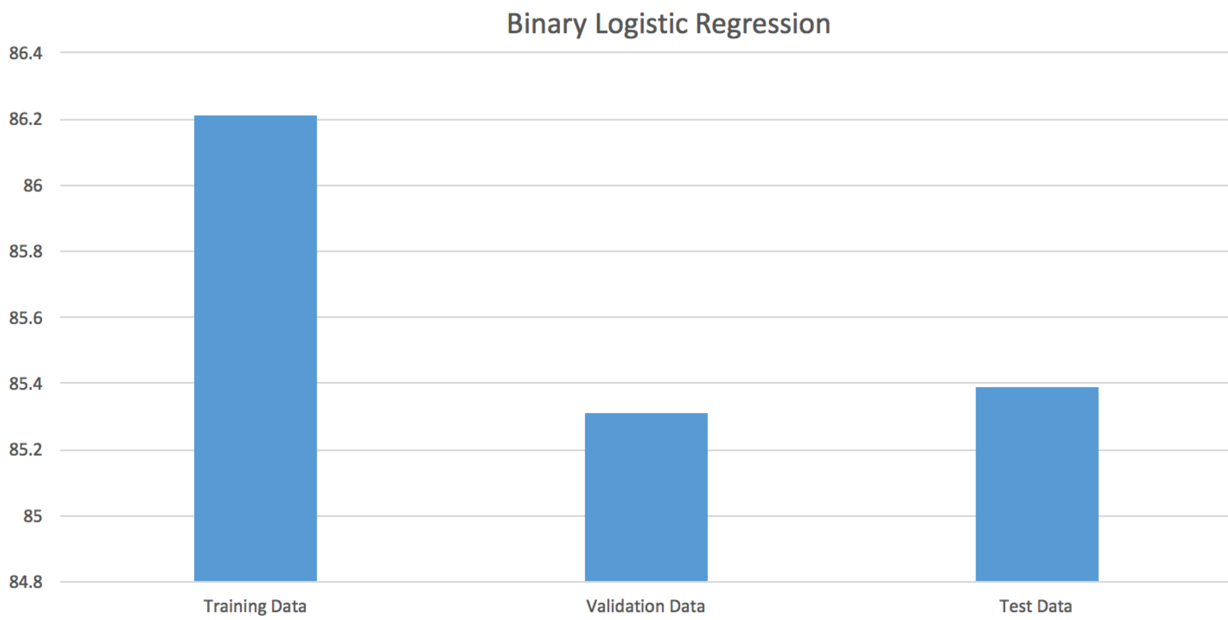

Figure 1 Binary Logistic Regression Accuracies for Training, Validation and Test Data

Figure 2 Multi-class Logistic Regression Accuracies for Training, Validation and Test Data

<u>Support Vector Machines:</u>

Support vector machines constructs hyper plane or set hyper planes in a high dimensional space. This can be used to perform classification and regression. A good separation is achieved for a hyper plane that has the largest distance the nearest training point any class since in general larger the margin lower the generalization error of the classifier. However, the sets discriminate are not linearly separable in that space. So, the original finite space mapped into a much higher dimensional space, presumably making the separation easier in space.

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

Training Data Accuracy	97.286%
Validation Data Accuracy	93.64%
Test Data Accuracy	93.78%

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

Training Data Accuracy	100%
Validation Data Accuracy	15.48%
Test Data Accuracy	17.14%

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

Training Data Accuracy	94.294%
Validation Data Accuracy	94.02%
Test Data Accuracy	94.42%

Using **radial basis function** with 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 = 'rbf'											
	Gamma=1	Gamma = 'auto'										
	C=1	C=1	C=10	C=20	C=30	C=40	C=50	C=60	C=70	C=80	C=90	C=100
Training	100%	94.29%	97.13%	97.95%	98.37%	98.706%	99.002%	99.19%	99.34%	99.43%	99.54%	99.61%
Validation	15.48%	94.02%	96.18%	96.9%	97.1%	97.23%	97.31%	97.38%	97.36%	97.39%	97.36%	97.41%
Testing	17.14%	94.42%	96.1%	96.67%	97.04%	97.19%	97.19%	97.16%	97.26%	97.33%	97.34%	97.4%

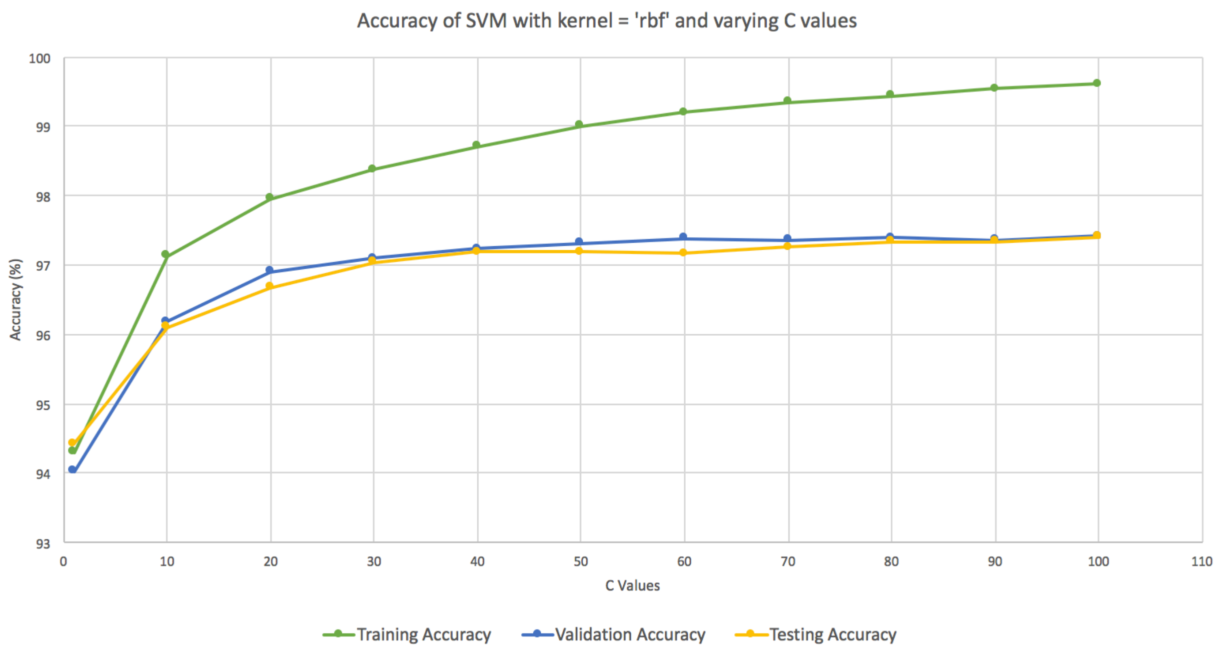

Figure 3 Accuracy of SVM with Kernel as 'rbf' and varying C values

Linear Kernel vs Radial Basis Function Kernel

When the kernel is radial basis function and all the other parameters are default and in another case with the kernel as linear and all other parameters as default, we note that the performance of the test data is better in the case of former.

Kernel	Accuracy
Kernel = ‘linear’	93.78%
Kernel = ‘rbf	**94.42%**

This performance can be justified because in real world scenarios we can note that all data is modeled to fit the Gaussian distribution. Here we note that this accuracy is also seen in the when we use the ‘rbf’ kernel instead of linear kernel. Another reason can be specific to our case where we observe that we have more input data than features where we have 50000 data and 716 features. Thus, with more data than features we see that a ‘rbf’ kernel performs better.

Influence of hyper parameters on Accuracy

Kernel

In all the observations, we can see that the radial basis function performs better than a linear kernel. This is because the real-world data cannot always be differentiated into two. That would work only for any data which answers true or false classification.

C Value

A standard SVM separates all positive and negative examples. However, in case of poorly labelled models or extremely unusual models we will have poorly fit models. This is where a soft margin in SVM is used to ignore some observations. This allows for an overall better fit of the data. Here C is the value for a soft margin cost function. It controls the influence of each individual support vector. This is in essence trading error penalty for stability. A lower value of a C gives a smaller marginal hyper plane. As a result, a smaller margin hyper plane is built and lesser number of points are misclassified, so the accuracy of the data classification increases. So, there is a chance of the overfitting of the data for larger values of C. We can see in the graph that the accuracy of training data increases significantly but the accuracy of validation and test data saturates after increasing initially. Thus, increasing the complexity of the hyper plane gives us the overfitting problem.

Gamma

Gamma has an influence on each training example on the learned hyper plane. For higher values of gamma, we have the over fitting problem and although we have 100% accuracy on the training data we obtain very low accuracy on the validation and testing data, namely 15.48% and 17.14%. When gamma is set to 0, we have better results on the validation and test data set. Thus, we note that large gamma leads to high bias and low variance models and vice versa.

Conclusion:

	Testing Data Accuracy
Binary Logistic Regression	85.39%
Multi-Class Logistic Regression	92.67%

Multi-class logistic regression performs better than binary logistic regression because the space is divided into more distinct parts without any class overlap. Defined boundaries will make the classification of the all inputs more accurate because each region can be matched to a result.

The handwritten digits classification is a real world scenario which can be best related to fit a radial basis function. Also, the input data is more than the features which again reasons the performance of kernel = ‘rbf’ being better than kernel = ‘linear’ with all parameters default.