CSE 574 Introduction to Machine Learning

Programming Assignment – 3

Classification and Regression

Date: May 3, 2017

Group 12

Venkata Saicharan Kolla
Chaitanya Vedurupaka
Anirudh Yellapragada

Binary Logistic Regression:

Logistic regression is a type of probabilistic classification model where we come up with a probability function that can give us the chance, for an input to belong to any one of the various classes we have (classification). Logistic regression can be binomial, ordinal or multinomial.

Binomial or binary logistic regression(BLR) deals with situations in which the observed outcome for a dependent can have only two possible types, "0" and "1". We performed BLR on the digit classification data, and the observed accuracies are as follows:

Data Set	Accuracy
Training Data	84.878%
Validation data	83.7%
Testing Data	84.12%

Even though the accuracies for all the three data sets are a little lower, we can notice that compared with training set accuracy, the test set is almost equal, by which we can say that the model performed accurately for the given data sets and there is no kind of overfitting.

Direct Multi-class Logistic Regression:

Multi-class or Multinomial logistic regression deals with situations where the outcome can have three or more possible types that are not ordered. We extended the BLR to solve the multi-class classification. The accuracies on each data set are as follows:

Data Set	Accuracy
Training Data	93.114%
Validation data	92.39%
Testing Data	92.54%

In this case too, the model performed accurately on the given data set with no overfitting. Compared to binary logistic regression, multi-class regression has high accuracy. And also the time taken for MLR to learn the training data is relatively less compared to BLR.

Performance Comparison between the two strategies:

The *One-Vs-All* strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. In One-Vs-All, we are training the model with C classifiers, where C is the number of unique class labels. We need to train C models in case of binary regression. And we used sigmoid function for classification to find the class probability of the data. But, these class probabilities are not efficient. One more problem with this strategy is even if the class distribution is balanced in the training set, the binary classification learners see unbalanced distributions because typically the set of negatives they see is much larger than the set of positives. But, in multi-class logistic approach, we can obtain the multiclass probabilities at the same time, as we consider all the classes to design a single model. And we have used softmax for classification. Since in the softmax equation, we divide with sum of all the possibilities, we get the efficient class probabilities and can consider the maximum of these for predicted label. Since we consider all the classes at same time to calculate probability, the efficiency in multi regression is higher than that of the binary regression. Also the learning time in multi-class logistic approach is relatively less compared to One-Vs-All strategy. Multi-class approach is much less sensitive to the problems of imbalanced datasets but is much more computationally expensive compared to BLR.

Support Vector Machine

A Support Vector Machine is a hyperplane based classifier which can be used for classification and regression. We used SVM tool in sklearn to perform classification on the data set. We computed the accuracy of prediction with respect to training data, validation data and testing data for the following hyper parameters:

1.Linear kernel:

In general, Linear SVM is less prone to over-fitting than non-linear (e.g. RBF) and this can be observed from the below accuracy values. Linear SVM works best when the data is linearly separable and since our data set is linearly separable it is giving good accuracies. Also linear SVM will be useful if data being tested on is multi-dimensional (E.g. Handwritten digit classification data in our case)

Data Set	Accuracy
Training Data	97.286%
Validation data	93.64%
Testing Data	93.78%

2.Radial Basis Function:

RBF is a non-linear SVM. C and Gamma are the parameters for a nonlinear SVM with a Gaussian radial basis function kernel. The C parameter trades off misclassification of training examples against simplicity of the decision surface. The gamma parameter defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close. We tested with different values of gamma and C observed the following results

a) Gamma=1:

Very low accuracies were observed for Validation and Testing data which is a classic case of overfitting

Data Set	Accuracy
Training Data	100%
Validation data	15.48%
Testing Data	17.14%

b) Gamma=Default:

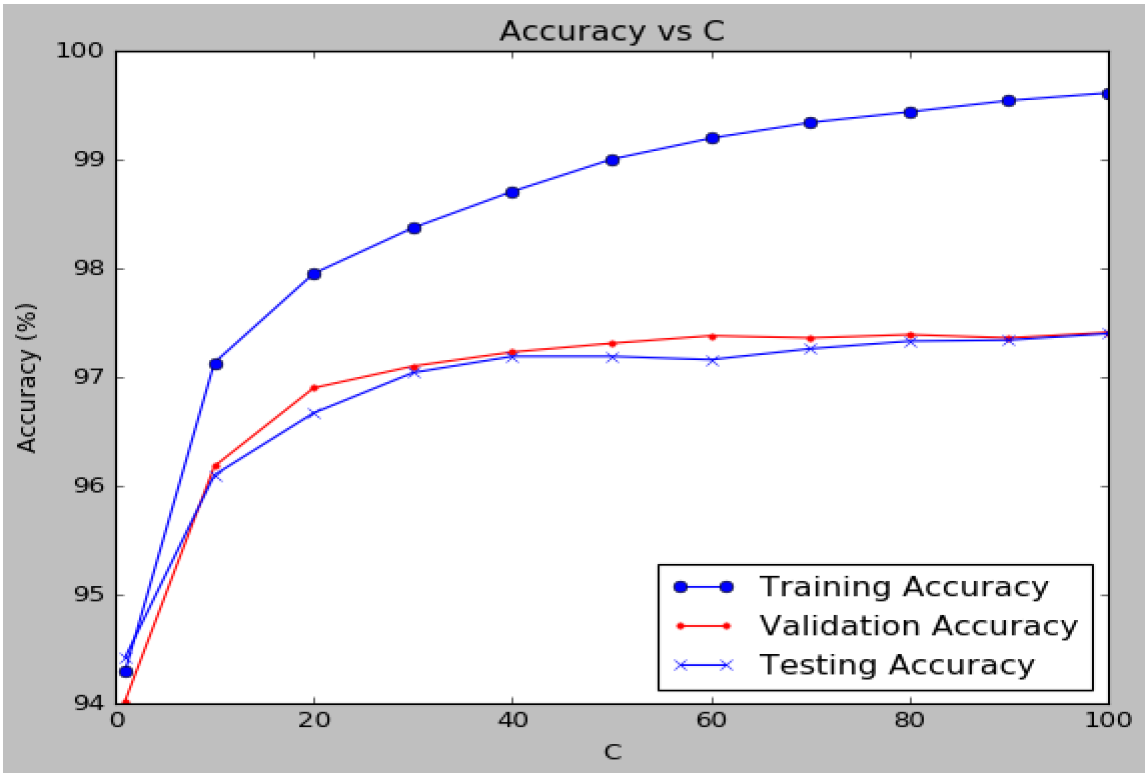
The accuracies with default values of gamma gave better results compared to Gamma=1. Default value of gamma is given by '1/n_features' and for our data set, n_features is 716.

Data Set	Accuracy
Training Data	94.294%
Validation data	94.02%
Testing Data	94.42%

From the above 2 results (gamma = 1 and default), we can deduce that gamma controls the influence of each training example on the learned hyperplane because when gamma is high (gamma = 1.0) we saw overfitting and for low values of gamma, we saw much better results.

c) Gamma=Default and Varying values of C:

From the below plot for Accuracy Vs C, we can see that as C value is increased, the accuracy is also getting increased. That is happening because C controls the cost of misclassification on the training data and it basically determines the impact of error on the training examples. When the C value is low, the corresponding error of each error term is low as well which means that a larger error value can be accepted during the training phase. Therefore, a larger margin hyperplane is created which results in more samples being misclassified. So we can say that a small value of C gives you higher bias and lower variance.



Based on analysis for smaller values of C, we can say that for higher values of C there will be overfitting i.e. a large C gives you low bias and high variance. Low bias because you penalize the cost of misclassification a lot. But from the plot, we can see that the value of C is increased for all the three data sets. If we observe closely, we can see that there is a slight drop in validation and test data sets for values of C between 40 and 70. This is an indication of overfitting. And even for larger values of C (> 100) the accuracy might actually drop because of overfitting. Therefore, the value of C should be neither too large nor small, but somewhere in between where we can find the right balance.

Linear Kernel vs RBF Kernel:

From experimental analysis, we found that the training time for Linear Kernel is faster compared to non-linear kernel(RBF) which is a really a good advantage. Linear SVM is less prone to overfitting than non-linear which is quite evident from the results (very low accuracies are reported for gamma=1). But RBF can give better accuracy results than linear kernel if we tune the value of C and gamma properly (gamma = default and C = varying values from 1 to 100).