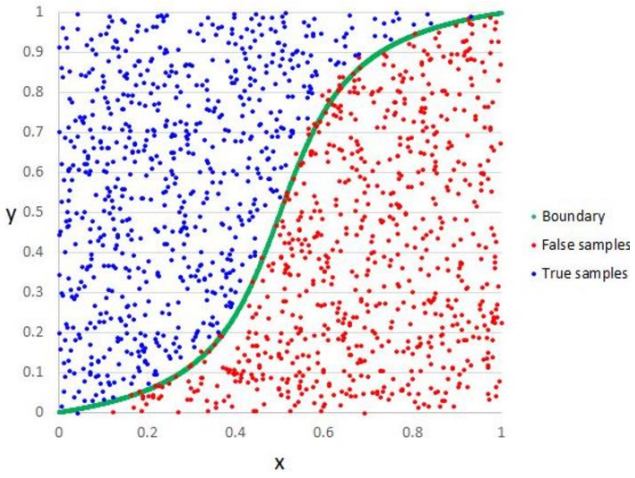
CSE474/574 Introduction to Machine Learning
Programming Assignment 3

Classification and Regression

Group 35

TARIQ SIDDIQUI KAUSHIK RAMASUBRAMANIAN JUNAID SHAIKH

Problem 1: Logistic Regression



Logistic regression is a discriminative classifier which tries to learn linear boundary. Hence, it is a model for classification rather than for regression. This model is based on Log Loss where it trains data based on Maximum Likelihood Approach.

Logistic regression is better than linear regression in way that it can handle classification of multiple classes by directly calculating posterior probability for each outcome. Hence, it's also known as MaxEnt Classifier or Maximum entropy Classifier. To achieve multiclass classification, logistic model uses One-vs-Rest approach or One-VS-Other approach as discussed later.

Below table shows accuracy % for different data in absence of any regularization.

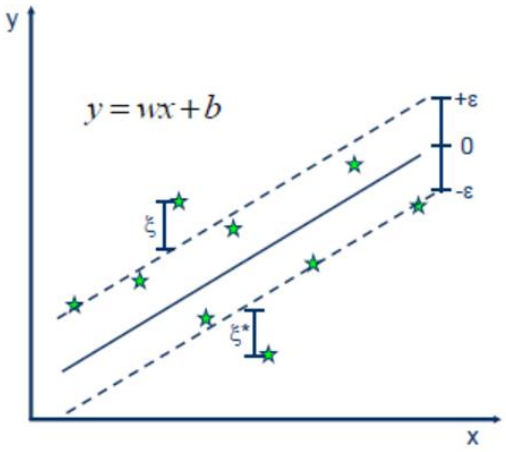
Type Of Data	Accuracy %
Training Data	84.908 %
Validation Data	83.74 %
Testing Data	84.18 %

Table 1: Different Accuracies for MNIST Data with Logistic regression with Gradient descent approach

Problem 2: Support Vector Machines(SVM)

SVM or Maximum Margin Classifier is a quadratic optimizer that tries to minimizes error through better generalizability and by increasing Margin.

SVM is used to improve the accuracy of classification and because of its ability to deal with high dimensional data. Also, unlike Logistic regression, SVM classifies two classes by finding the hyper-plane for the data and maximizing the margin between the two different classes.



- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)$$

- Constraints:

$$y_i - wx_i - b \le \varepsilon + \xi_i$$

$$wx_i + b - y_i \le \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \ge 0$$

When training an SVM, we need to make a number of decisions: how to process the data, what kernel to use and finally setting the parameters of SVM and the kernel.

We will be experimenting with three different parameters for this assignment, the kernel, gamma and C. The theoretical explanations behind these choices are explained as below:

Kernel:
Kernel provides various options such as Linear, RBF, poly and others. RBF and poly are useful for non-linear hyper-plane. Linear kernel is used in cases where we have large number of features because it is more likely that data is linearly separable in high dimensional space. When using RBF, care should be taken to cross-validate the parameters so as to avoid over-fitting.

Gamma:
It is the kernel co-efficient for RBF kernel. For higher values of gamma, it will try to better fit as per training data set and make mistakes in classifying validation and test data due to over-fitting.

C:
Penalty parameter C of the error term, it controls the trade-off between a smooth decision boundary and classifying the training points correctly. A low C makes the decision surface smooth whereas a high C classifies the training samples giving the model freedom to select more samples as support vectors. As the C value is increased, the margin between the hyper-planes around the decision boundary decreases.

An effective combination of these parameters should always be looked into at the cross-validations core to avoid over-fitting.

Results:

Type of Kernel	Training Accuracy(%)	Validation Accuracy(%)	Test Accuracy(%)
Linear	97.286	93.64	93.78
RBF(with Gamma=1)	100.0	15.48	17.14
RBF(with Gamma=auto)	94.294	94.02	94.42

Table 2 Accuracy for SVM with Linear Kernel vs RBF with Gamma=1 and Gamma=auto

As seen above, with gamma set to 1, the validation and test accuracies drop significantly. This is due to **over-fitting** and trying to over-accurately classify each training example.

Observation1:
For higher values of gamma, SVM tries to better fit as per training data set and make mistakes in classifying validation and test data due to **over-fitting due to larger step sizes**. Though, since we over fit to training data, training accuracy is 100%

Observation2:
For automatic setting of gamma, SVM dynamically changes gamma as to reduce error function values as per training data set and hence makes less mistakes in classifying validation and test data by choosing **appropriate step size** values.

Observation3:
SVM always performs better than logistic regression because it not only tries to find a line/plane which helps in classification but also tries to **maximizes marginal Value.**

Observation4: Logistic regression performs better for data with lesser number of features as lesser data has more probability of being cleanly linearly separable. Whereas, SVM performs better than logistic regression for more number of features as it not only finds classifying plane/hyperplane but also one with maximum margin

Observation5: A linear kernel performs worse than non-linear kernel though training time for linear kernel is small. This is expected with the complexity in curve that non-linear kernel brings to table.

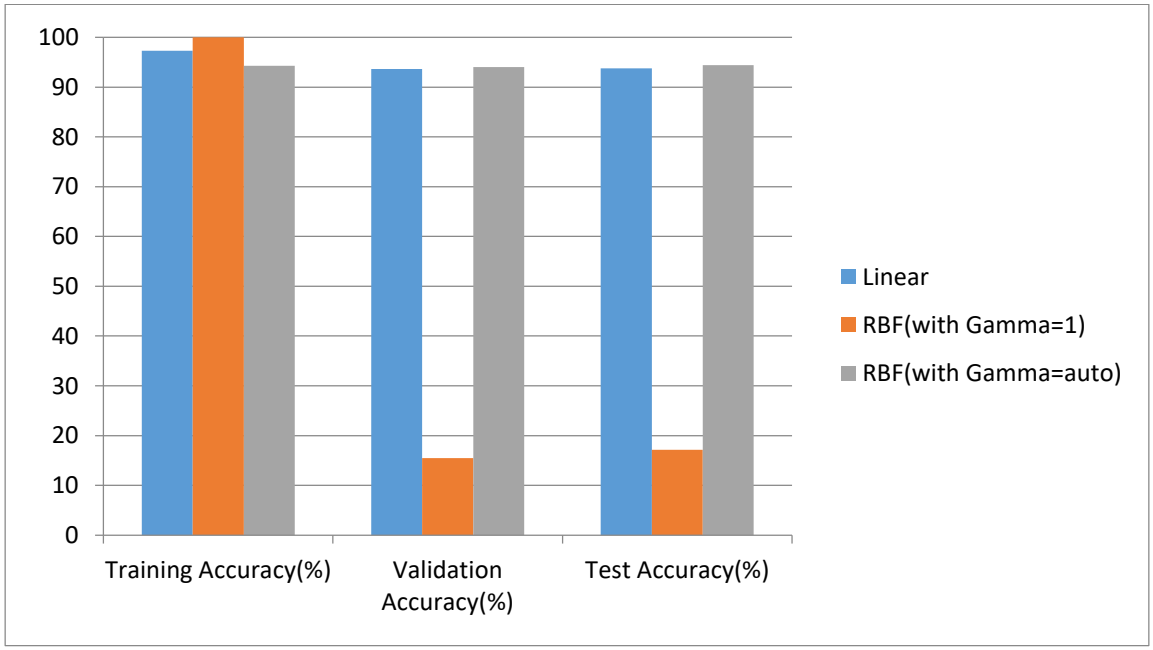


Figure 1: Train, Validation and Test Accuracies for Linear Kernel vs RBF Kernel with Gamma=1 and Gamma=Auto

As can be seen from the plot above, RBF kernel performs better than linear with Gamma=auto for validation and testing data. In this case, RBF is better than Linear kernel. However, that will not always

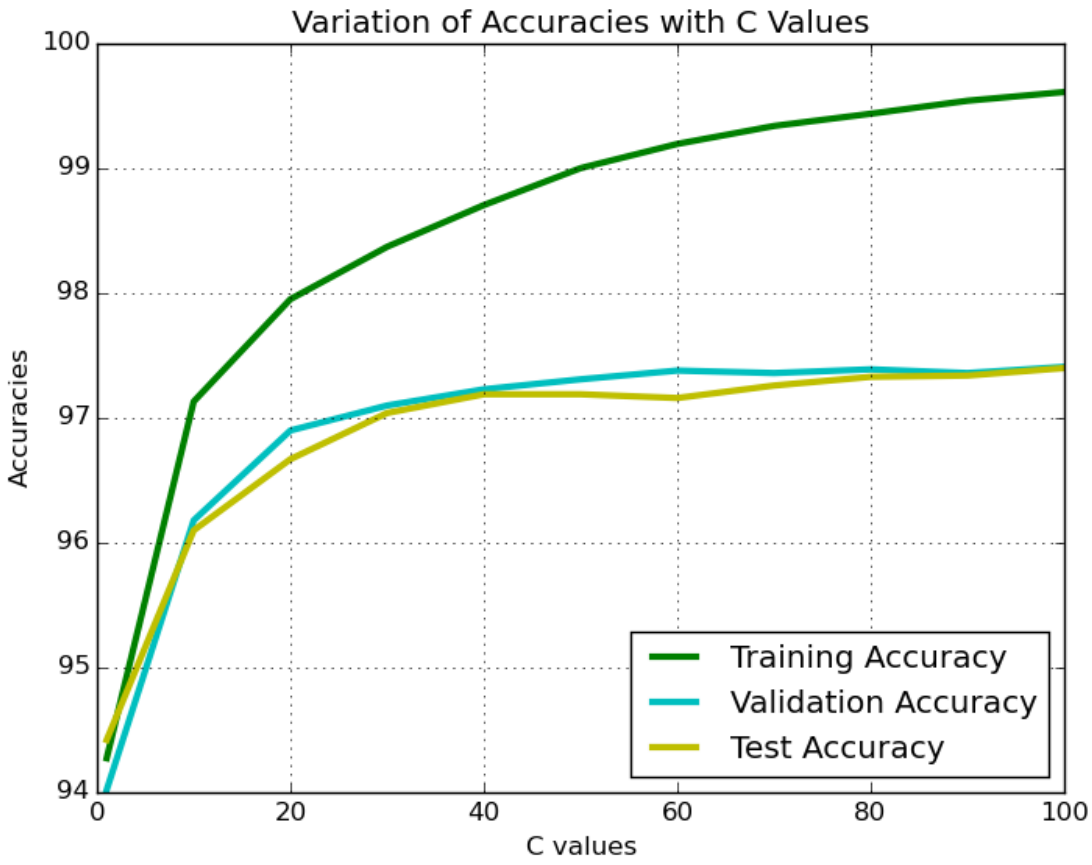
be the case. **In a scenario where the number of dimensions (or features) is high, the advantages of RBF over Linear kernel will diminish.**

C values	Training Accuracy(%)	Validation Accuracy(%)	Test Accuracy(%)
1	94.274	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

Table 3 Accuracy for SVM with RBF kernel for different values of C

Observation6: For higher values of C, even lower margin is accepted and hence less points lie in unacceptable region and hence overall accuracy for all above 3 data increases.

Observation 7: C controls the complexity of the hyperplane. This is reflected in above plot where training data has steeper slope than validation and testing data. Since higher C values mean a lower margin between the hyper-planes, the training accuracy goes on increasing as C value is increased. However, validation and testing accuracy remain more or less constant after a certain point as shown in below plot.



Problem3 (Extra Credit): Gradient Descent minimization of multi-class Logistic Regression

Logistic regression is better than linear regression in way that it can handle classification of multiple classes by directly calculating posterior probability for each outcome. Hence, its also known as MaxEnt Classifier or Maximum Entropy Classifier. To achieve multiclass classification, logistic model uses :

1> One-vs-Rest approach : In this classification, we create artificial class and compare one possible output class to rest of the possible outputs. In summary, this is achieved by repeating binary classifier for all the outputs.

2> One-VS-Other approach: In this K-Classifier, new data is fed to all the output classes and posterior probability is calculated for each class for given input. The final outcome is then decided by selecting the class with highest posterior probability.

Type Of Data	Accuracy %
Training Data	93.212%
Validation Data	92.56%
Testing Data	92.45%

3> Table 4 Different Accuracies for MultiClass Logistic Regression

Obsevation8: Multiclass logistic regression performs better than one-vs-all strategy. This is expected since with given small size data of MNIST Handwritten digits and input digits are **not closely co-related**. Hence, In One-VS-Other approach, true probability for a digit clearly overpowers other digits in most cases rather than the case where a single digit is compared against all the digists.One-VS-Rest performs at par with One-V-Other in cases where data is discriminable with larger probabilities between each possible outcome.

THANKS