UNIVERSITY AT BUFFALO

CSE – 574 INTRODCUTION TO MACHINE LEARNING

CLASSIFICATION AND REGRESSION

GROUP 11

GROUP MEMBERS:

SIDDHESH CHOURASIA : 50415033

TANUSH TRIPATHI: 50411177

SARVESHWAR SINGHAL: 50418642

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.

Binary Logistic Regression:

Set	Accuracy	Error
Training Set	92.706%	7.294%
Validation Set	91.44%	8.56%
Testing Set	91.93%	8.07%

From the above table, we can infer that the training set accuracy, which is 92.706% is greater than the testing dataset. This means that the linear model performs better on the seen data than the unseen data, and when it comes to the unseen data, it works less accurately.

Problem 2: Multi-Class Logistic Regression:

In Multi-Class Logistic Regression, new data is fed to all the output classes and posterior probability is calculated for each class for given input. The outcome is then decided by selecting the class with highest posterior probability.

Set	Accuracy	Error
Training Set	93.358%	6.642%
Validation Set	92.26%	7.74%
Testing Set	92.46%	7.54%

From the above table is the result table for multiclass logistic regression on Training, Validation and Testing data. We can clearly see that the training error is slightly less than the testing error. From this, we can conclude that that linear model performs better on the seen data than the unseen data, and when it comes to the unseen data, it gives more error as compared to that of training set.

Comparison between Multi-class strategy and one-vs-all strategy

Set	MLR Accuracy	BLR Accuracy
Training Set	93.358%	92.706%
Validation Set	92.26%	91.44%
Testing Set	92.46%	91.93%

- 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 digits. One-VS-Rest performs at par with One-V-Other in cases where data is discriminable with larger probabilities between each possible outcome.

- Also in multiclass logistic regression, we classify all 10 of the classes of MNIST dataset at once, whereas in one-vs-all strategy, we only classify one class with respect to all others at a time, and therefore multi-class logistic regression has less time complexity and less chances of overlapping.

- From the above observations, we can see that the accuracy of the multiclass logistic regression is better than the binary logistic regression. The reason behind this is that the parameters are estimated independently in multiclass which helps to prevent the faulty classification.

Problem 3: Support Vector Machines(SVM):

Support Vector Machines are supervised learning models with associated learning algorithms that analyze the data for classification and regression analysis. Here we trained the SVM model on MNIST dataset and computed the accuracy of prediction by using following parameters.

SVM or Maximum Margin Classifier is a quadratic optimizer that tries to minimize the error through better generalizability and by increasing Margin. SVM is used to improve the accuracy of classification 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.

When training an SVM, we need to make several 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 nonlinear 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 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 investigated at the cross-validations core to avoid over-fitting.

- **SVM With linear kernel:**

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

From the above table, we can infer that the results of Linear kernel works exactly like a linear model since the results are like that of the previous linear models that were trained.

- **Radial Basis Function:**
  - **SVM with radial basis kernel and gamma = 1**

Set	**Accuracy**
Training Set	100%
Validation Set	15.480%
Testing Set	17.14%

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.

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%

- **SVM with radial basis kernel and default gamma value**

Set	**Accuracy**
Training Set	98.982%
Validation Set	97.89%
Testing Set	97.87%

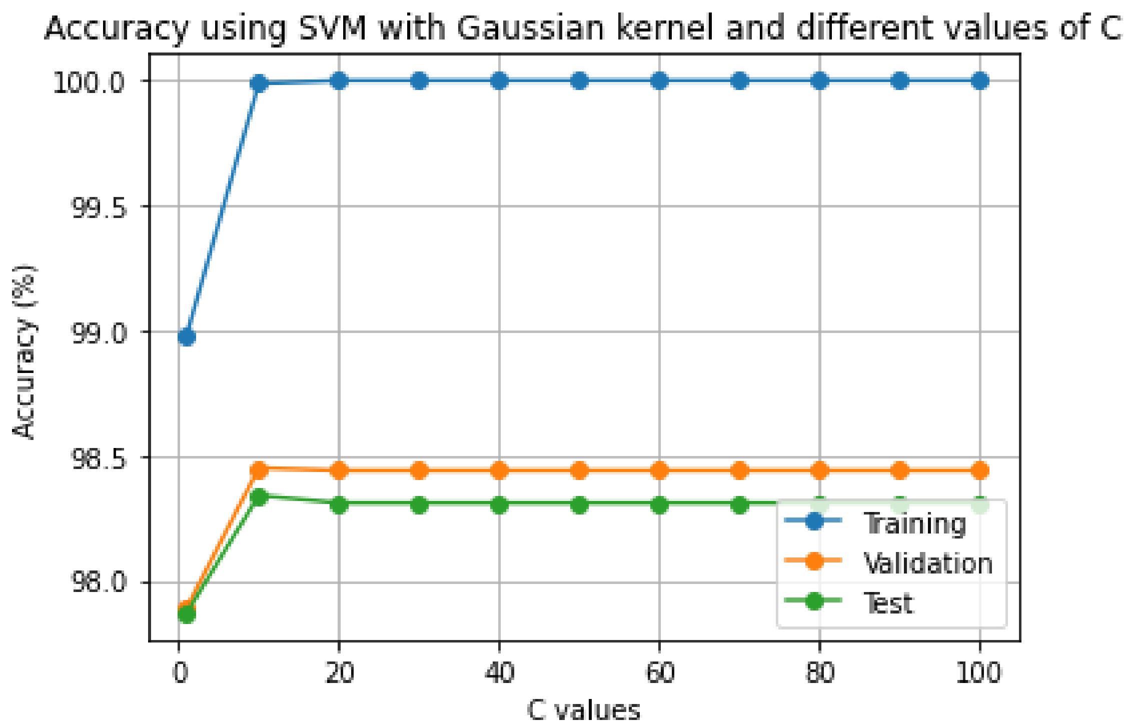
For default value of gamma, SVM dynamically changes gamma as to reduce error function values as per training data set and hence makes less mistakes in classifying validation and test data by choosing appropriate step size values.

As seen from the data above, RBF kernel performs better than linear with default Gamma value for validation and testing data. 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.

- **SVM with radial basis kernel and default gamma value and varying the values of C from 1 to 100**

C	Training Accuracy	Validation Accuracy	Test Accuracy
1	98.982	97.89	97.87
10	99.988	98.45	98.34
20	100	98.44	98.31
30	100	98.44	98.31
40	100	98.44	98.31
50	100	98.44	98.31
60	100	98.44	98.31
70	100	98.44	98.31
80	100	98.44	98.31
90	100	98.44	98.31
100	100	98.44	98.31

C controls the complexity of the hyperplane. This is reflected in below 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, at a certain point , the Validation and Test Accuracy drops by a negligible amount (0.03%) and after that validation and testing accuracy remain constant as the C increases as shown in below plot



So, we can also infer that our dataset is non-linear as it gives better result on this non-linear model.

The SVM Model with C=10, Kernel=rbf and gamma=default gives the best accuracy. The Training , Test and Validation Accuracy is given below:-

Kernel	Gamma	C	Training Accuracy	Validation Accuracy	Test Accuracy
rbf	default	10	99.988	98.45	98.34