

Intro to ML Assignment 3 Report

Te Shi 50608573 Yujie Zhu 50592341

Problem 1: Implementation of Logistic Regression

- Training, Validation and Testing Accuracy

Training set Accuracy:92.674%

Validation set Accuracy:91.39%

Testing set Accuracy:91.96%

As shown above, the accuracy of the logistic regression model on the three dataset is above 90%, suggesting a relatively good performance.

- Total Error

```
Training Error for each class: [np.float64(0.020402631778056814), np.float64(0.020808617698235637), np.float64(0.06277891597655558), np.float64(0.07525138652074126), np.float64(0.044105667080154974), np.float64(0.08343474751348183), np.float64(0.03397972479524274), np.float64(0.04294115367739344), np.float64(0.11027233115512301), np.float64(0.09679697801218122)]

Testing Error for each class: [np.float64(0.02612121553738191), np.float64(0.023193333585749893), np.float64(0.07322832878786498), np.float64(0.07164806145472477), np.float64(0.05096692057982986), np.float64(0.08392585079415814), np.float64(0.03814195835761067), np.float64(0.055318122771812195), np.float64(0.11290150458606056), np.float64(0.10235608642266104)]
```



The above are the error outputs for both training error and testing error. The result showed that the error for both training and testing are relatively low, suggesting good overall performance of the model in general. As expected, for most classes, the testing error is slightly higher than the training one. This is understandable since testing data consists of unseen samples, making generalization more challenging.

The results also show that the error is relatively higher for complicated digits like 8 and 9 but lower for simpler digits like 0 and 1. This makes sense since digits 0 and 1 have a simpler pattern so easier to identify. Digits 8 and 9 by contrast are relatively complex and share some similar patterns with other digits (for example 8 can be treated as two digit 3 combined together), therefore bring more difficulties for the model.

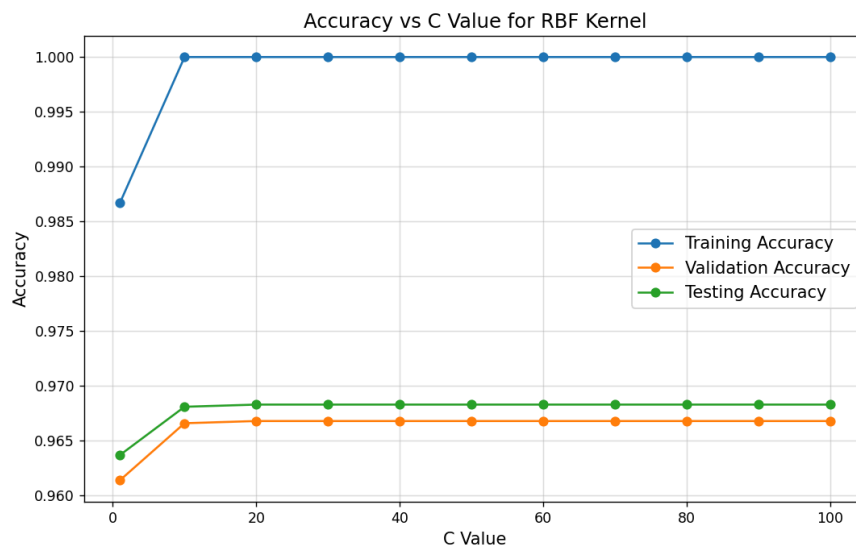
The reason why there is a difference between training error and test error is logistic regression may overfit the training data, so the training error is relatively low.

Problem 2: Support Vector Machine

- **Accuracy of SVM with Different Settings**

```
-----Linear Kernel-----  
training accuracy for linear kernel is 0.9952  
validation accuracy for linear kernel is 0.9173  
Testing accuracy for linear kernel is 0.9167  
-----RBF Kernel with Gamma = 1-----  
training accuracy for RBF Kernel with Gamma = 1 is 1.0  
validation accuracy for RBF Kernel with Gamma = 1 is 0.1702  
Testing accuracy for RBF Kernel with Gamma = 1 is 0.1865  
-----RBF Kernel with Default Gamma-----  
training accuracy for RBF Kernel with default Gamma is 0.9856  
validation accuracy for RBF Kernel with default Gamma is 0.9625  
Testing accuracy for RBF Kernel with default Gamma is 0.9641
```

- **Accuracy of SVM with RBF Kernel and Different C values**



- **Discussion**

The linear kernel SVM achieves nearly perfect accuracy on the training dataset, but the accuracy on validation and testing are 8% lower. Also, the accuracy can still be considered good, but it also indicates slight overfitting.

For the RBF kernel with Gamma sets to 1, the training accuracy is 100% but the performance on the testing and validation dataset is very poor (around 18% for both of

them), indicating an overfitting. It is expected since the gamma value of 1 is usually considered as too high. A large gamma value limits the influence of individual data points on their immediate surroundings, creating an overly complex decision boundary that remembers the training dataset very well but fails to generalize.^[1]

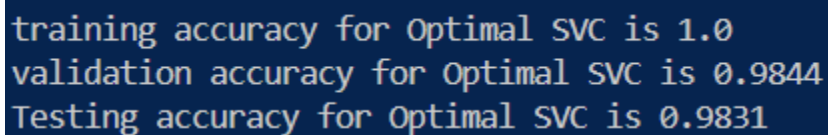
The RBF Kernel with default gamma performs well across all datasets, outperforming the linear kernel and RBF kernel with gamma set to 1. It proves that the RBF kernel with its ability to model non-linear boundaries, is better suited on complex and non-linear tasks like handwritten digit recognition than the simpler linear kernel SVM. Also, the default gamma (around 0.2857)^[2], maintains a balance by avoiding overfitting caused by large gamma values and underfitting caused by very small gamma values. This balance allows this model to capture complicated patterns and have a good generalization, therefore resulting in a better solution.

Then the graph demonstrates how different C values impact the model's accuracy. The C parameters control the trade-off between low training error and a simple decision boundary, ensuring good performance on unseen data. A larger C value will encourage a smaller margin and more complex decision functions and vice versa.^[1] From the graph, the performance plateaus after C=20, and the accuracy is consistently good. It suggests that the C parameter is important in terms of balancing both underfitting and overfitting, leading to the best model among the options. In addition, the result also shows that the default C = 1 does not fit the dataset as well as other choices.

- **Best Choice of Parameters**

From the output and analysis above, the best choice of parameters of SVM is RBF Kernel, default Gamma, and C=20. The reason C should be set to be 20, rather than any larger value is that the result is basically the same but larger C will increase the training and predicting time. Also, a larger C may result in overfitting when the model focuses on the entire training dataset.

- **Accuracy of Optimal SVM On Whole Training Dataset**



```
training accuracy for Optimal SVC is 1.0  
validation accuracy for Optimal SVC is 0.9844  
Testing accuracy for Optimal SVC is 0.9831
```

The result of the SVM with the optimal parameters selected above is nearly perfect for training, validation and testing datasets. It proved the effectiveness and correctness of our choice.

Problem 3: Multi-Class Logistics Regression

```
-----MULTI LOGISTIC-----  
  
Training set Accuracy:93.448%  
  
Validation set Accuracy:92.47999999999999%  
  
Testing set Accuracy:92.55%
```

The overall performance of the multi-class logistic regression is excellent, achieving over 90% accuracy across all three datasets. The training set has a slightly higher accuracy than testing and validation. This is normal because the model is optimized directly on the training dataset. Or since training set accuracy is a little bit high, it may be slightly overfitting. However, the above 90% accuracy on validation and testing proves the model also generalized well to unseen data. Multi-class logistic regression outperforms the one-vs-all in all metrics by a small margin. This slight improvement likely arises from the use of a softmax function, which considers all classes simultaneously, leading to more consistent results. In contrast, the one-vs-all model may have conflicts in prediction since it trains separate binary classifiers for each class, leading to slightly worse performance.

Reference

1. "RBF SVM parameters." *scikit-learn documentation*,
https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.
2. "Default value of gamma - SVC sklearn." *Stack Overflow*, answered by Rishabh Agrahari on Jan 9, 2020,
<https://stackoverflow.com/questions/59660939/default-value-of-gamma-svc-sklearn>.