CSE4/574 Introduction to Machine Learning Programming Assignment 3 Classification and Regression

Anand anand6@buffalo.edu
Jing Chen jchen445@buffalo.edu

Classification using Logistic Regression: Many v/s One

We know MNIST dataset contains features and response variable as hand written digits to identify digits ranging from 0 to 9 resulting in 10 classes.

Model 1: Many binary logistic classifiers to build multi class logistic classification

We use sigmoid function to calculate the probability and corresponding binary value and then calculate the log likelihood error, finally using the gradient of error to get the optimal w. Then we use the learnt w of each class to predict the digit label.

Training set Accuracy:92.716%

Validation set Accuracy:91.42%

Testing set Accuracy:91.9%

The training error is lower than the test error because the optimal w (the parameter) is calculated based on the training data.

Model 2: One classifier for multi class logistic classification

We use softmax function to calculate the probability and then calculate the log likelihood error, finally using the gradient of error to get the optimal w. Then we use the learnt w of each class to predict the digit label.

Training set Accuracy:93.122%

Validation set Accuracy:92.57%

Testing set Accuracy:92.55%

The training error is lower than the test error because the optimal w (the parameter) is calculated based on the training data.

Conclusion:

Model 1 builds 10 classifiers with accuracy around 92%, and model 2 builds 1 classifier with accuracy around 93%, so from the accuracy performance, Model 2 might be better. Also, the runtime for model 2 used is much less than model 1, so model 2's total performance is better than model 1.

Support Vector Machines:

SVM kernel as linear: sampled train data of 10000 observations

Training set Accuracy:93.002%

Validation set Accuracy:91.41%

Testing set Accuracy:92.27%

SVM kernel as radial basis function:

Choosing hyper parameters – gamma and cost to tune the svm for randomly sampled 10000 from train data.

Gamma: Default

Training set Accuracy:91.902%

Validation set Accuracy:91.95%

Testing set Accuracy:92.45%

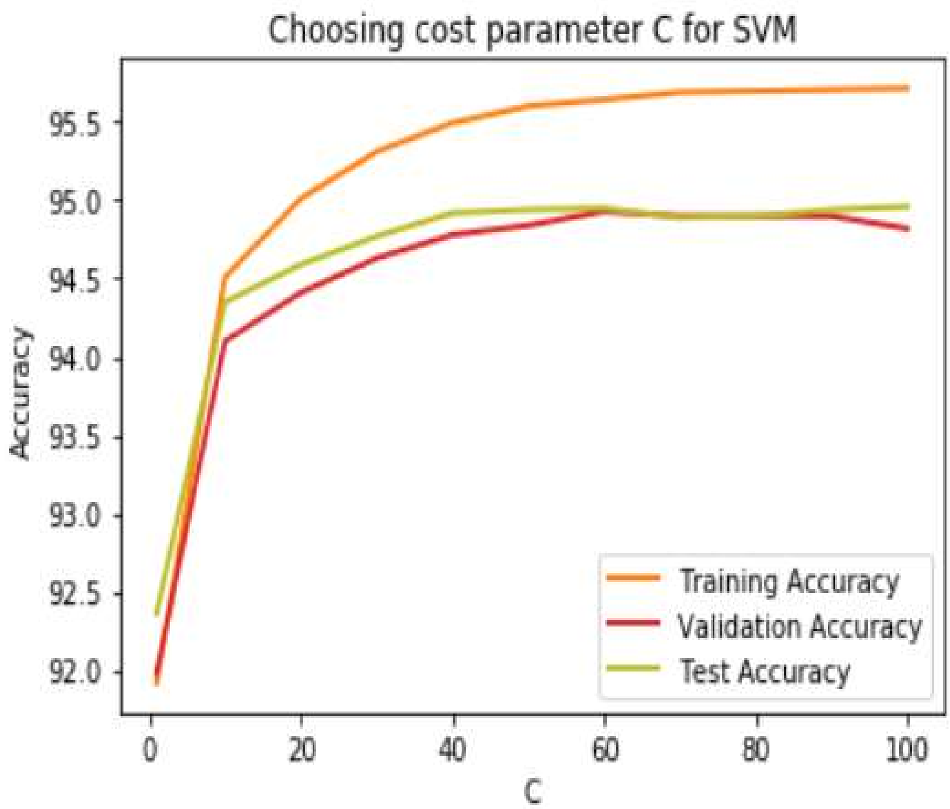
Gamma:1

Training set Accuracy:32.884%

Validation set Accuracy:14.26%

Testing set Accuracy:15.8%

C- Cost Parameter: Gamma default and C = [1,10,20,30,40,50,60,70,80,90,100]



Testing accuracy was high when C=10 and it does not improve much after then. C controls the impact of margin and the margin error, as the value of C get larger, the model allowed less misclassifications and smaller margin, usually a smaller C has larger margin and will allow more support vectors in learning process since inside the margin points both correct and wrong sides points get more. Because only the number of support vectors will affect the accuracy.

Since after c = 10, there is no too significant improvement in the accuracy, so we decide to use c = 10.

Final Model: Gamma default and C=10 trained on full train data.

Training set Accuracy:97.13199999999999%

Validation set Accuracy:96.17999999999999%

Testing set Accuracy:96.1%

Conclusion:

- Assuming separable case SVM:
  For given dataset with linear SVM the test accuracy is around 90% to 93%.

- Assuming non-separable case SVM:
  For radial function with tuning hyper parameters – C and gamma
  When gamma is 1, with default c = 1, the accuracy is very low
  While gamma as default :1 / (n_features * X.var()), the accuracy is around 92%
  When gamma is default, at c = 10, we achieved around 96% of accuracy.

The radial method takes longer time, so if the data is very large and there is no high requirement on accuracy, using linear SVM might be more efficient as its performance is not bad.