UNIVERSITY AT BUFFALO

CSE574 - Introduction to Machine Learning

Programming Assignment 3 – Report (Group 10)

Aayush Shah (50207564)
Siddharth Shah (50205787)
Haril Satra (50208283)
5/3/2017

Experiment with Logistic Regression (One vs All)

Accuracies

Data	Accuracy
Training Set	84.942
Validation Set	83.78
Testing Set	84.26

Analysis

One vs All is one way to classify data with k classes using k binary logistic classifiers. This classifier is better to use when the the data can belong to more than one class.

One issue this classifier faces is that while training a binary classifier for one class the training samples of that class are less as compared to the rest of the samples (if the data are equally sampled in all the classes). This may cause imbalance while training the data.

One way to overcome this issue is to use the One vs Other method where we create a binary classifier for each class with each other class, leading to ${}^{k}C_{2}$ classifiers. However this introduces another issue of training a lot of classifiers if the number of classes is too large leading to inefficiency.

Experiment with Direct Multiclass Logistic Regression

Accuracies

Data	Accuracy
Training Set	93.188
Validation Set	92.5
Testing Set	92.54

Analysis

Multiclass or Multinomial Logistic regression is another way to learn multiple classes using logistic regression. In binary logistic regression the distribution was Bernoulli. Now to learn multiple classes the distribution is now Multinoulli. A softmax function is used to predict the class label.

This type is preferred over One vs All method when the data exclusively belongs to just one class. Our data is a good example to show this. A digit can either be 0,1,2,…….,9. It cannot be part 0 and part 5. Thus for our data Multinomial Logistic Regression provides better accuracy than One vs All Binary Logistic Regression.

In Multinomial Logistic Regression we just have to train one classifier as compared to k binary classifiers in the One vs All method, thus resulting in better efficiency if the number of classes are large.

Experiment with SVM

Using Linear kernel (all other parameters are kept default).

Data	Accuracy
Training Set	97.286
Validation Set	93.64
Testing Set	93.78

Hyper-Parameter Tuning: Linear kernel is usually used when the number of features is large. It is the quickest kernel among all. With the kernel as linear SVM algorithm will try to find the largest possible linear margin that separates the regions of all the classes. Since the accuracy is high it can be claimed that the data is quite linear and does not need to be mapped to a higher dimensional space.

Using Radial basis function with value of gamma setting to 1 (all other parameters are kept default).

Data	Accuracy
Training Set	100.0
Validation Set	15.48
Testing Set	17.14

Hyper-Parameter Tuning: Gamma decides how far the influence of a single training example reaches. The larger gamma is, the closer other examples must be to be affected. The default value of gamma is (1/n_features). Setting gamma as 1 (high) means that we are localizing the reach or in other words it is adjusting too much to the training examples which will result in over fitting and provide poor accuracies for other testing data.

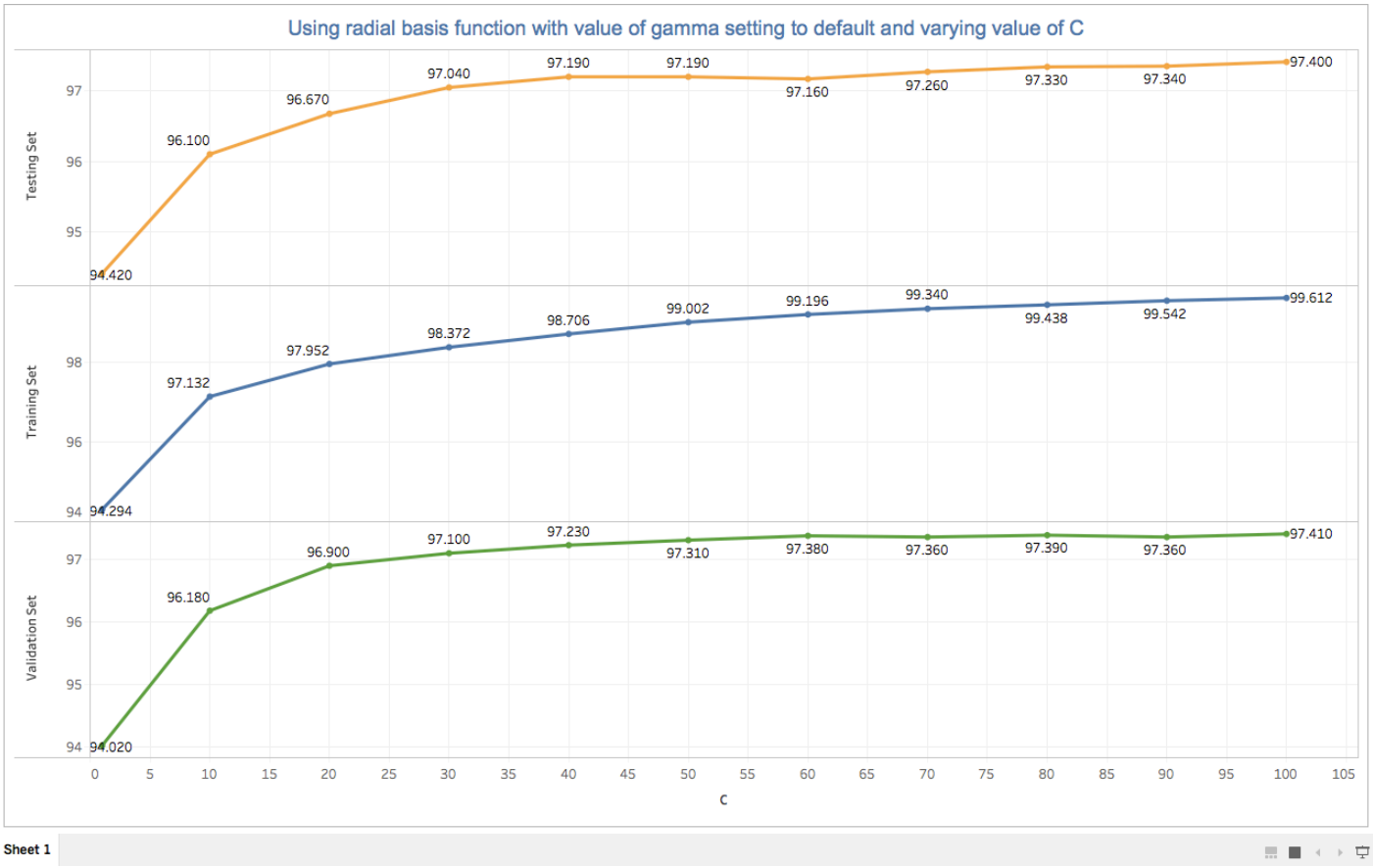
Using Radial basis function with value of gamma setting to default (all other parameters are kept default).

Data	Accuracy
Training Set	94.294
Validation Set	94.02
Testing Set	94.42

Hyper-Parameter Tuning: Gamma decides how far the influence of a single training example reaches. The larger gamma is, the closer other examples must be to be affected. The default value of gamma is (1/n_features) which is considered intermediate given the number of training examples in our data. When the gamma is very small, the model is too constrained and cannot capture the complexity of the data. Intermediate value solves the problem of both small value of gamma and high value of gamma. Thus providing decent accuracy. However this can be further improved by tuning of parameter C.

Using radial basis function with value of gamma setting to default and varying value of C.

C	Training Set	Validation Set	Testing Set
1	94.294	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



Hyper-Parameter Tuning: The parameter C, trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. As mentioned in the above case intermediate value of gamma (smooth models) along with larger values of C can help to make the models more complex and thus provide good accuracy. This can be confirmed from the above provided results and plot. Higher values of C with the default value of gamma gives better accuracy.

Comparison of Linear Kernel and Radial Basis Function.

As mentioned earlier the linear kernel is faster than the RBF kernel. Linear kernel is usually used when the number of features is high whereas the RBF kernel is used when the number of observations is higher than the number of features. Since RBF kernel is a non linear kernel it will always perform better than the linear kernel or at least as good as the linear kernel. The linear kernel only searches for the parameter C, whereas in the RBF kernel along with C the gamma value also plays a crucial role as seen above.

References

[1] http://scikit-learn.org/stable/modules/svm.html#svm

[2] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[3] https://www.quora.com/In-multi-class-classification-what-are-pros-and-cons-of-One-to-Rest-and-One-to-One

[4] https://stats.stackexchange.com/questions/52104/multinomial-logistic-regression-vs-one-vs-rest-binary-logistic-regression