PROGRAMMING ASSIGNMENT 3

Utkarsh Pratap Singh Jadon, jadon.1

CSE 5526: PA-3: Report

1. INTRODUCTION

Programming Assignment 3 required the training and implementation of a Suppot Vector Machine (SVM) to determine if a genomic sequence is an Non-Coding RNAs (ncRNAs). An 8-dimensional feature input is provided to the classifier. Training and test data were provided in the form of 'ncRNA_s.train.txt' and 'ncRNA_s.test.txt' files. Each row in the file represents an instance, first column represents the class label, and remaining eight columns corresponds to eight feature values.

Part 1 required **Classification using linear SVMs**, wherein a linear SVM was trained for different values of C, ranging from $C = (2^{-4}, 2^{-3}, 2^{-2}, ..., 2^{7}, 2^{8})$. Classification accuracy was plotted against different values of C.

Part 2 required **Classification using RBF kernel SVMs**. Subpart (a) involved 5-fold cross validation to choose best C and γ values. 50% (1000 instances) of training set were used as cross validation set, which was subsequently divided into 5 subsets of equal size (200 instances). 5 SVM models were trained using these 5 validation subsets, and the cross-validation accuracy was calculated as the average over 5 validation accuracies. Both C and γ were varied in the range of $(2^{-4}, 2^{-3}, 2^{-2}, ..., 2^{7}, 2^{8})$, and thus, a total of $13x13 = 169$ different models were trained. Output is provided in the form of a matrix of cross-validation results, where (i, j) entry of the matrix corresponds to classification accuracy on the cross validation set with i^{th} value of C and j^{th} value of γ. Subpart (b) involved training an SVM using whole data set with the best C and γ values found in subpart (a)

2. METHODOLOGY

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm used for classification tasks. Linear SVMs are a variant of SVMs that use a linear decision boundary to separate data points into different classes. In a linear SVM, the decision boundary is a hyperplane that separates the data points into different classes. Mathematically, a hyperplane is represented as a linear equation:

$$g(x) = w^T x + b = 0 \tag{1}$$

where w is the normal vector to the hyperplane, b is the bias term, and x is the input feature vector. The sign of the expression $w^T x + b$ determines which side of the hyperplane a data point belongs to.

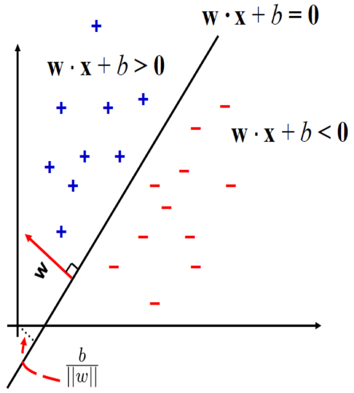

$\mathbf{w} \cdot \mathbf{x} + b = 0$
$\mathbf{w} \cdot \mathbf{x} + b > 0$
$\mathbf{w} \cdot \mathbf{x} + b < 0$
$\mathbf{w}$
$\frac{b}{||w||}$

Fig. 1. Linear SVM Boundary

SVMs aim to find the hyperplane that has the maximum margin between the classes. The margin is the perpendicular distance between the hyperplane and the closest data points from each class. The goal is to find the hyperplane that maximizes this margin.

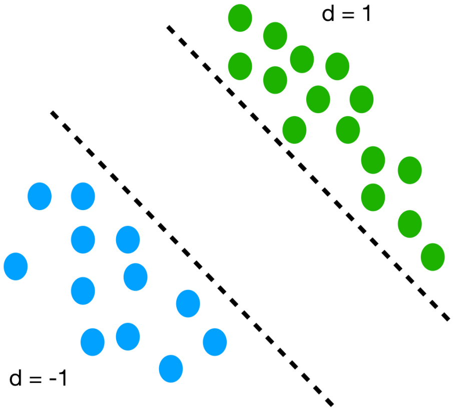


Fig. 2. Linear SVM Margin

$$w^Tx + b \geq 1 \text{ for } d_i = 1$$
$$w^Tx + b \leq -1 \text{ for } d_i = -1$$

Optimization problem can be formulated as:

$$min\frac{1}{2}\|w\|^2$$
$$\text{subject to } d_i(w^Tx_i + b) \ \geq 1 \, \forall \, i \ \in [1, N]$$

But this method works only for linearly separable data. Classification using SVMs with a Radial Basis Function (RBF) kernel is a popular technique for non-linear classification tasks. The RBF kernel allows SVMs to capture non-linear relationships between data points by transforming the input data into a higher-dimensional space. RBF kernel is defined as:

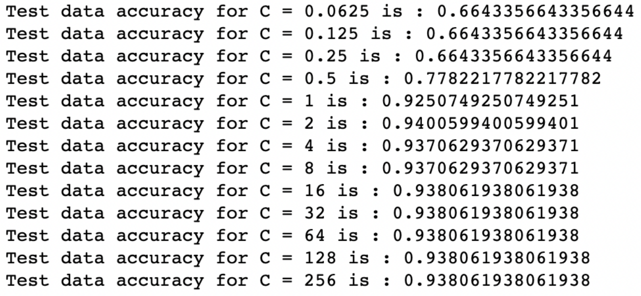
$$k(x, x_i) = e^{-\gamma\|x-x_i\|^2}$$

First step was to load the data, and create a dataframe object for both training and testing data by segregating the columns in text file. All the N/A values were replaced by 0. Post that, the dataframe were converted to numpy arrays for better handling. For part 1, classification using linear SVM was performed by providing different values of C to the SVC class from Scikit-Learn toolbox. 13 different models were trained for 13 different C values, and classification accuracy was plotted with respect to different C values.

For part 2, subpart (a), 5-fold cross validation was implemented manually to get best C and γ values. For 13 C values, and 13 γ values, 169 different models were trained with 169 different classification accuracy values. All these values were stored in a 13 x 13 matrix, where each entry (i, j) corresponds to classification accuracy obtained using i^{th} value of C and j^{th} value of γ. Maximum classification accuracy was achieved at the matrix location (8, 1). Finally for subpart (b), an SVM was trained on whole data set with best C and γ values obtained.

3. RESULTS

3.1. Classification using Linear SVMs



```
Test data accuracy for C = 0.0625 is : 0.6643356643356644
Test data accuracy for C = 0.125 is : 0.6643356643356644
Test data accuracy for C = 0.25 is : 0.6643356643356644
Test data accuracy for C = 0.5 is : 0.7782217782217782
Test data accuracy for C = 1 is : 0.9250749250749251
Test data accuracy for C = 2 is : 0.9400599400599401
Test data accuracy for C = 4 is : 0.9370629370629371
Test data accuracy for C = 8 is : 0.9370629370629371
Test data accuracy for C = 16 is : 0.938061938061938
Test data accuracy for C = 32 is : 0.938061938061938
Test data accuracy for C = 64 is : 0.938061938061938
Test data accuracy for C = 128 is : 0.938061938061938
Test data accuracy for C = 256 is : 0.938061938061938
```

Fig. 3. Classification Accuracy Values

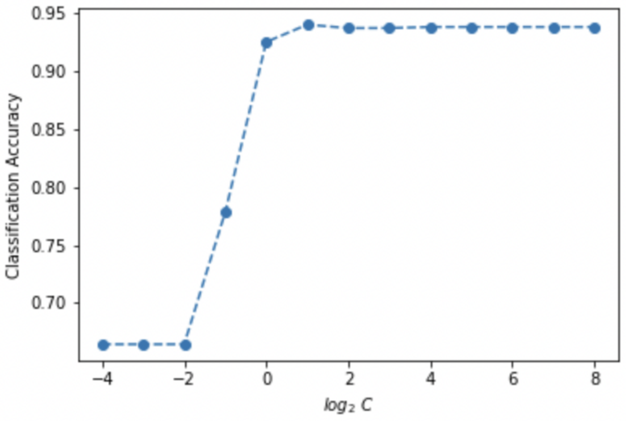
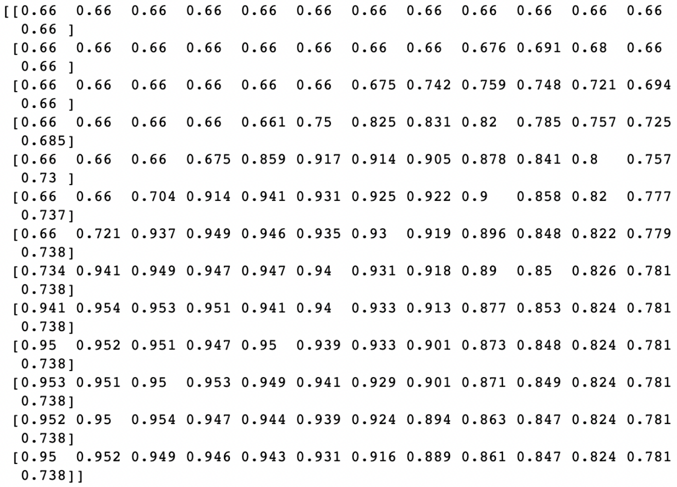


Fig. 4. Accuracy vs $\log_2$C

From Fig. 3 and Fig. 4, we can see that maximum classification accuracy is achieved at $C = 2$, with the value of 94%.

3.2. Classification using RBF kernel SVMs



```
[[0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.66
  0.66 ]
 [0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.66  0.676 0.691 0.68  0.66
  0.66 ]
 [0.66  0.66  0.66  0.66  0.66  0.66  0.675 0.742 0.759 0.748 0.721 0.694
  0.66 ]
 [0.66  0.66  0.66  0.66  0.661 0.75  0.825 0.831 0.82  0.785 0.757 0.725
  0.685]
 [0.66  0.66  0.66  0.675 0.859 0.917 0.914 0.905 0.878 0.841 0.8   0.757
  0.73 ]
 [0.66  0.66  0.704 0.914 0.941 0.931 0.925 0.922 0.9   0.858 0.82  0.777
  0.737]
 [0.66  0.721 0.937 0.949 0.946 0.935 0.93  0.919 0.896 0.848 0.822 0.779
  0.738]
 [0.734 0.941 0.949 0.947 0.947 0.94  0.931 0.918 0.89  0.85  0.826 0.781
  0.738]
 [0.941 0.954 0.953 0.951 0.941 0.94  0.933 0.913 0.877 0.853 0.824 0.781
  0.738]
 [0.95  0.952 0.951 0.947 0.95  0.939 0.933 0.901 0.873 0.848 0.824 0.781
  0.738]
 [0.953 0.951 0.95  0.953 0.949 0.941 0.929 0.901 0.871 0.849 0.824 0.781
  0.738]
 [0.952 0.95  0.954 0.947 0.944 0.939 0.924 0.894 0.863 0.847 0.824 0.781
  0.738]
 [0.95  0.952 0.949 0.946 0.943 0.931 0.916 0.889 0.861 0.847 0.824 0.781
  0.738]]
```

Fig. 5. Classification Accuracy Values

Fig. 5 shows the average accuracy values for 13 different values of C and γ. Maximum accuracy was achieved at the index location (8, 1), with the value of 0.954.

Best C value: 16
Best γ value: 0.125

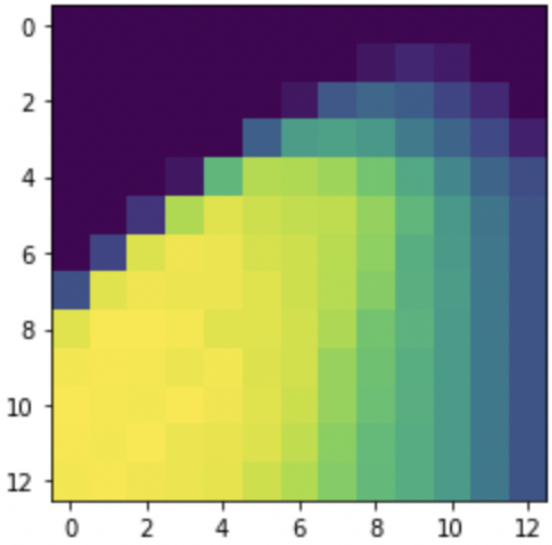


Fig. 6. Classification Accuracy Matrix

Fig. 6 depicts the classification accuracy matrix pictorially.

Finally, an SVM was trained using entire dataset with best C and best γ values, which produced classification accuracy of 93.706%.

4. CONCLUSION

In conclusion, SVM algorithm was developed for two type of kernels, linear and RBF. For linear, entire data set was used whereas for RBF, 5-fold cross validation was performed. Different values of C and γ were given as input to train these models, and results from all these models have been provided in the previous section. Final SVM was trained using the best parameters obtained.