SVM for Emotion Classification

Abinaya K (Sr No : 06-02-01-10-51-17-1-14956)

1. SVM Implementation

Given subject faces with happy/sad emotions, the aim is to train a 2 class SVM model with the labelled training happy and sad faces. Given a new face image, it is to be classified into happy or sad emotion.

Data

The training data consists of 20 face images showing happy or sad emotions (9 happy faces and 11 sad faces). The test data consisted of 10 images to be classified into either of these classes.

Dimensionality Reduction

Each image is of 101*101 matrix hence each input vector is of dimension 10201. High Dimensional PCA is performed on the input threby reducing the dimension to K ($K \leq 20$).

SVM Classifier Implementation

SVM implementation is done using the LIBSVM package and the functions svmtrain and svmclassify are used for training a svm classifier and predicting a test image respectively.

Figure 1 shows the block diagram illustrating the implementation with and without LDA.

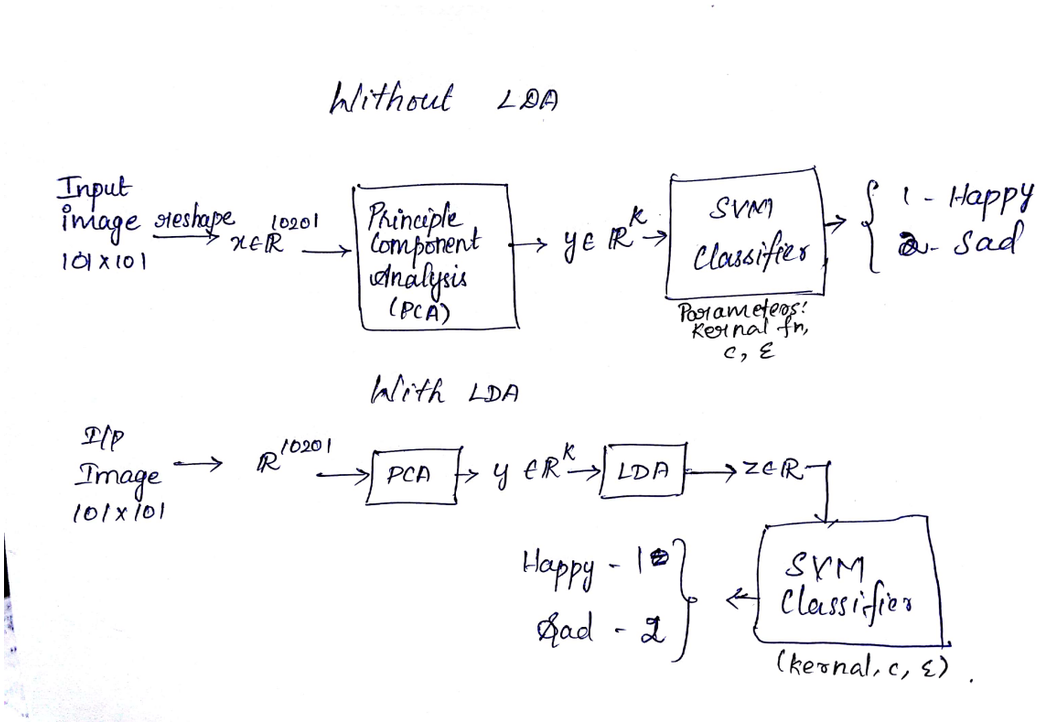

Figure 1: Testing Phase of SVM Implementation - Block Diagram

Classification and Results

There were 10 test face images, that are to be classified as happy or sad emotions.

Accuracy in classification is calculated as percentage of correctly classified samples on test data. Below are the accuracy for different choice of kernal functions, parameter C and tolerance epsilon.

Figure 2 shows the accuracy in classification with different choice of kernal fucntions. Default C and ϵ is used. Linear kernel function is found to perform better with an accuracy of 90%.

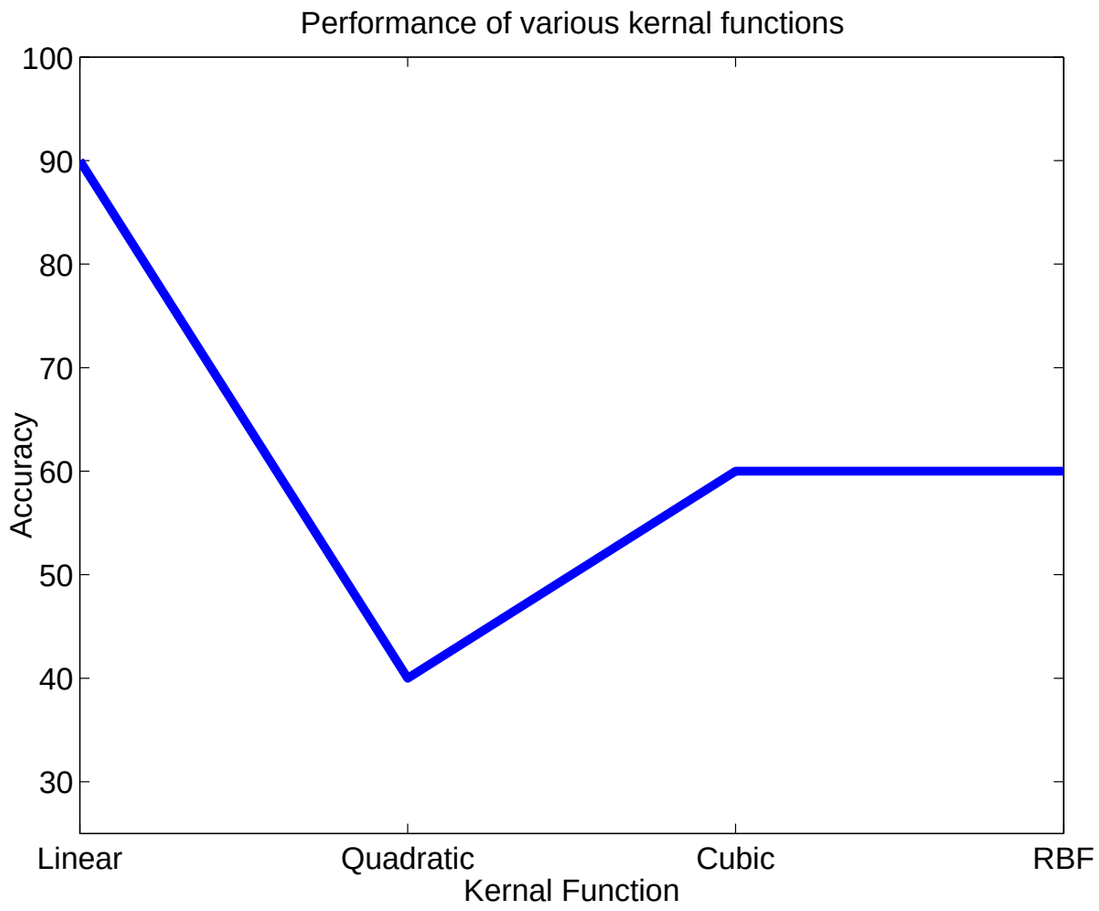

Figure 2: Performance of Different Kernel function

Figure 3 shows the accuracy with Linear kernal as a function of PCA dimension K. We could see that for K values 10, 11, 12, 18, 19 and 20 the accuracy is higher 90%. For rest of the analysis in this report K value is fixed as 20.

Figure 4 shows the accuracy of classification using SVM classifier with RBF kernal on training and test data. Please note that there is a typo in the legend. Red dashed line corresponds to training accuracy while blue plot shows the test accuracy. On Training, we could see the accuray increases with increase in the value of C. This is expected since large value of C tries to fit every data point in training. In the test data, the performance doesn't change for different choices of C.

Figure 5 shows accuracy of classification with RBF kernal for various choice of tolorence ϵ. For lower values of tolerance, we could see that the accuracy is more than the accuracy at higher tolerances.

Figure 6 shows the accuracy of classification with and without LDA. As expected, we could see the impreovement in performance with the use of LDA. As shown in figure 1, output of PCA is reduced to one dimension using LDA and is then classified using SVM.

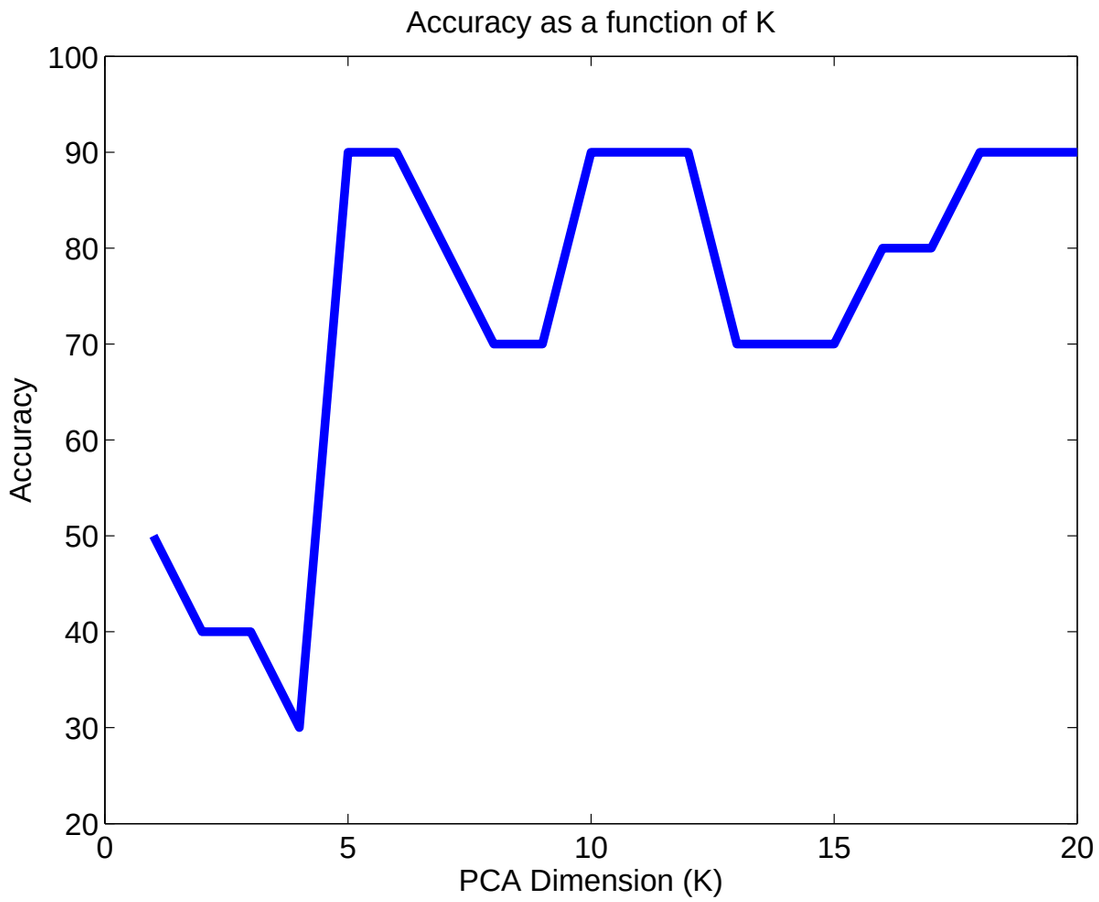

Figure 3: Performance as a function of K

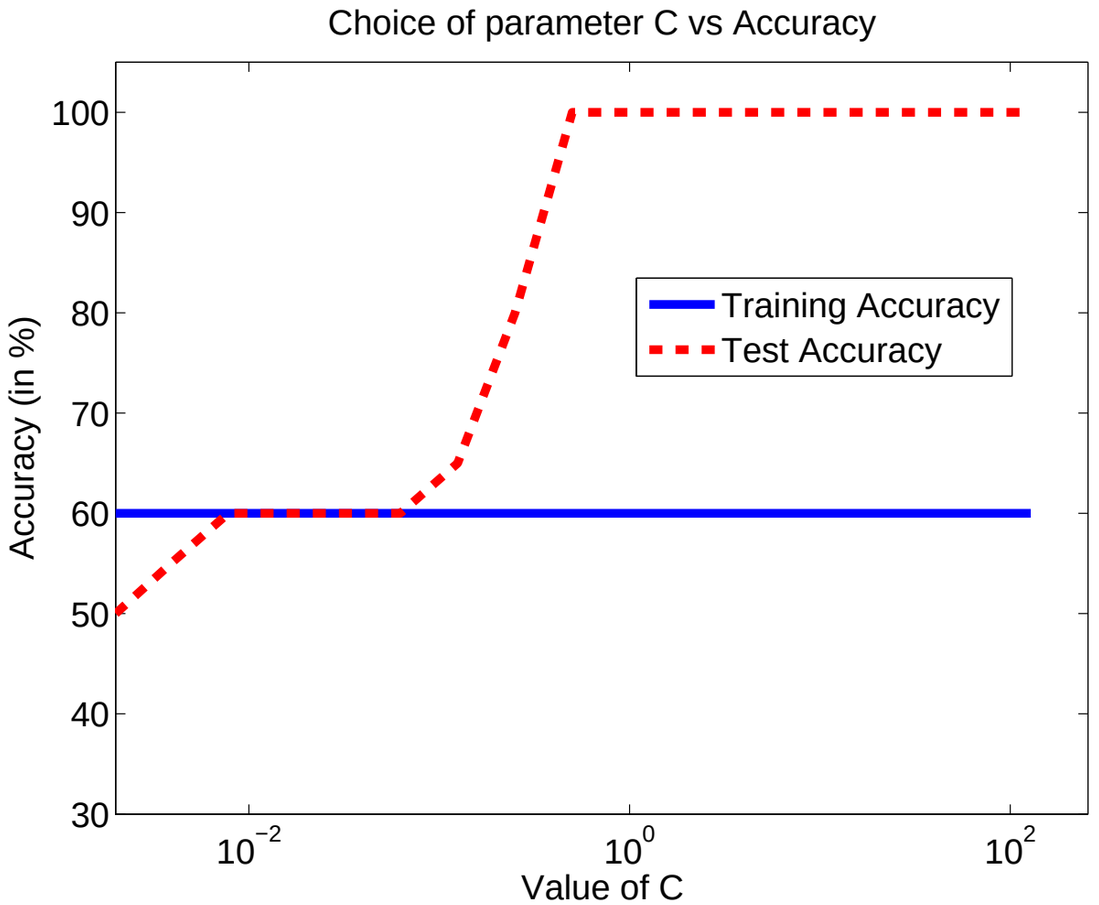

Figure 4: Performance for different choices of C : Typo in the legend: Red dashed line - Training Accuracy, Blue plot - Test accuracy

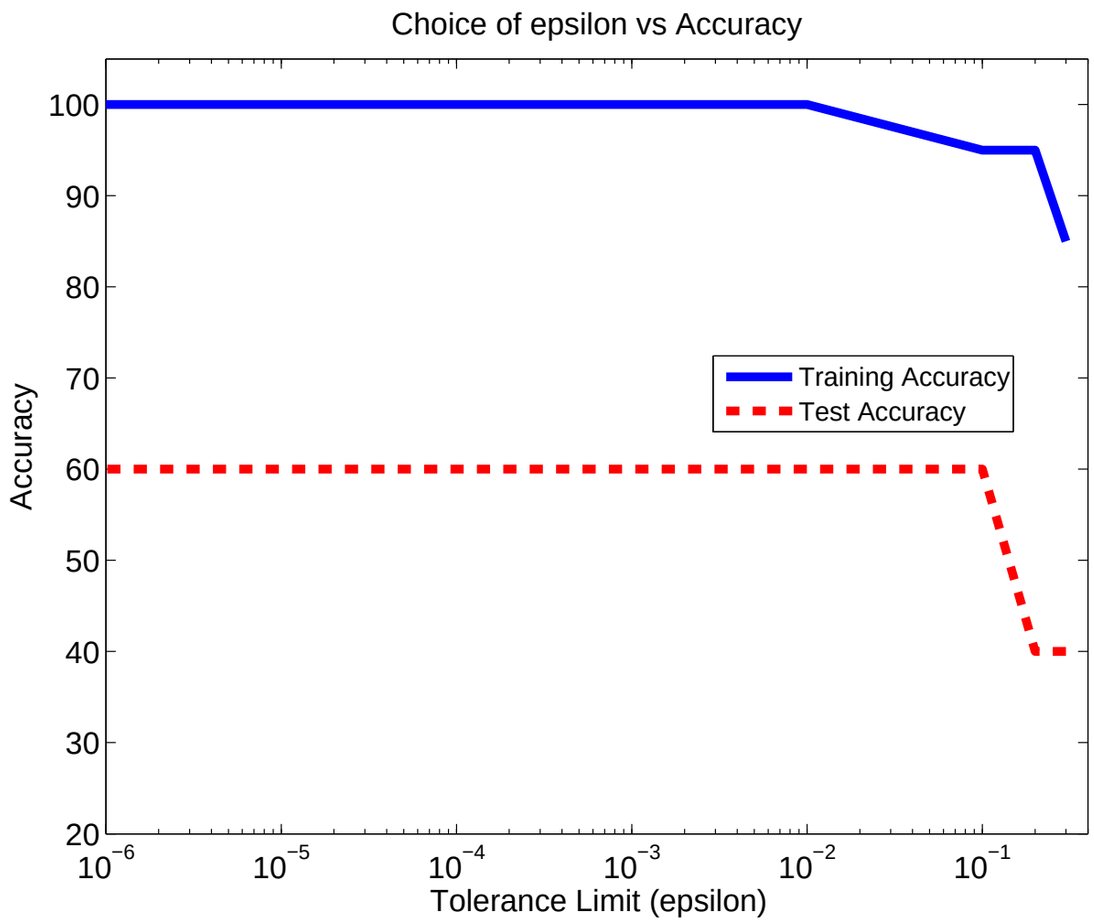

Figure 5: Performance for different choices of ϵ

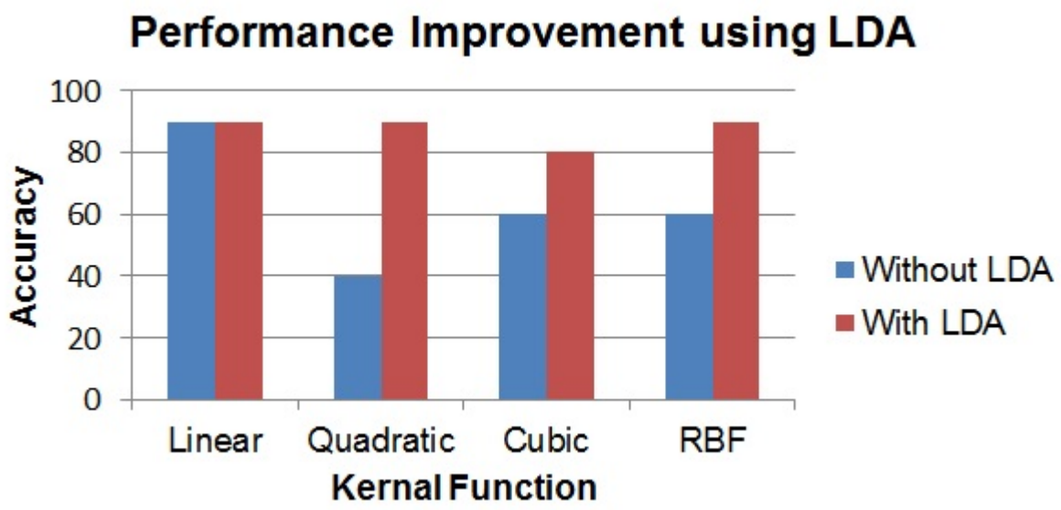

Figure 6: Performance improvement using LDA