Lab Report

1 Raw data and labels

In the very beginning, each person has 3 groups of experiments, each group has 15 matrices which correspond to 15 emotion labels contains "Disgusted", "Sad", "Neutral", "Positive" and "Scared". The matrices are of 33 by n format and each row represents a characteristic of eye movements. We extract every column to be the unit of sample points and the emotion label of the column will depends on which matrix it belongs to.

2 Data preprocessing and training

List the exact path of each file and load it. Use sklearn preprocessing to normalize each column for speeding up the train process.

$$Xscaled = preprocessing.scale(Xtrain)$$

Data selection: According to the choice of train range and test range, join the sample points of each person together.

Regarding to the train process, Liblinear is a simple package for solving large-scale regularized linear classification and regression which is suitable for our case. In addition, it consumes less time than libsvm. Suppose there are k classes to be trained. Libsvm use one vs one strategy for classification, which means it should train a classifier between every two class, totally $k(k-1)/2$ classifiers. While liblinear use one vs all strategy so it just need to train k classifiers.

For each sample point, use liblinear's train function to build the model. For the selection of parameter C, use a loop to change C from 0.001 to 1 and from 1 to 1024 so that we can find the best parameters.

$$model = train(train_label, test_label, params)$$
$$a, b, c = predict(test_label, test_x, model, '-q')$$

Select the best accuracy and keep a record of corresponding prediction.

3 Get confusion matrix

Confusion Matrix is a specific table layout that allows visualization of the performance of an algorithm, it can evaluate the quality of the output of a classifier on the data set.

Each row of the Matrix represents the instances in a predicted class while each column represents the instances in an actual class

In our case, create a $5X5$ matrix CM, for each value in the true label and the predict label,

$$CM[truelabel][predictlabel] += 1$$

Since we use 3-fold crossValidation, we should sum the 3 matrices up to get the value of the CM. Then normalize the CM and use sklearn confusion matrix package to draw it. Figure 1 is the confusion matrix we get.

The darker the color, the higher the accuracy.

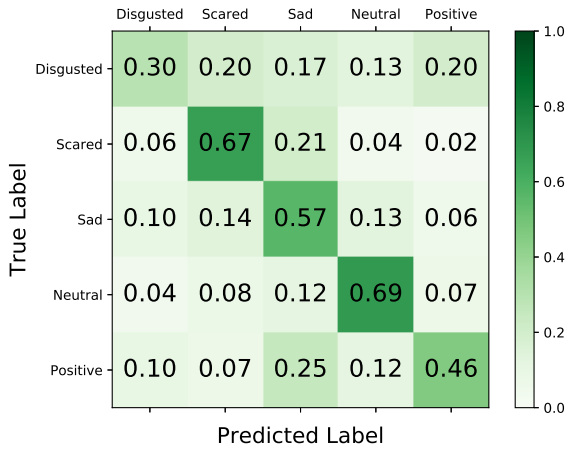

Figure 1: The confusion matrix for eye features predicting emotion

According to Figure 1, the prediction accuracy of "Scared" and "Neutral" are the highest.