Question 6 – multi-class classification

For this question I have decided to use LDA as linear classifier and k-nearest neighbor as non-linear classifier, just as it is suggested. Since the scatter plot and clustering from previous exercises shown good split of the two types of patterns, I guess it will be almost the same at this dataset and then these two classifiers should perform well. On top of that, I already had most of the code written since the assignments and wanted to solve the task easier. I have also done these implementations in Matlab as I did for the rest of the exam.

# LDA – Linear classifier

I have implemented this classifier just as I did for the assignment without using many Matlab built-in functions.

I have the function *my\_lda* which takes as input the train and test data and returns error rate on both. Unfortunately, the function is not well parameterized and only built to work for this exercise, so I calculated into separate variables the means and covariance for each the five types of patterns. The model is built and the prediction values are calculated for each class of patterns, put together and the accuracy for train and test set is returned.

The train error obtained is 0.0719 and the test error is 0.0825 which are quite low. As expected, it looks like even when the patterns are divided into these 5 classes, they are well grouped and provide good prediction accuracy using linear classifier. It could also happen that to be harder to identify 5 types of users instead of 2 but still in this application, with this dataset the performance is still good.

# K-nearest neighbor – Non-linear classifier

Since the 5 types of patterns are well grouped, I used the knn being quite confident that I will still obtain good performance. For this I also used most of my code from the assignment so I did not use many Matlab built-in functions.

I have started by reading data and normalizing it because unlike the linear classifier, a higher mean or standard deviation for some parameters can lead to bad results of the algorithm. Since I used 5-fold cross validation technique I shuffled the train set in order to find the k that provides the best accuracy. I had to do that because the patterns are sorted according to the label number and I wanted to have almost equal number of types of patterns at each split during cross-validation. For the cross validation I have decided to use 5-fold cross validation and obtained the best performance for k = 3.

After performing the cross validation, I used the value for k that provided the best results and obtain error rate of 0.0175 for train set and 0.0400 for test set which is better performance than for using linear classifier, perhaps because at the border between the groups of patterns the k-nearest neighbor does better classification of patterns and the separation created by the linear classifier does not split very well the patterns with different labels.