Lab 5: Features

Cristian M. Amaya
Universidad de los Andes
cm.amaya10@uniandes.edu.co

Abstract: In this laboratory, feature based classification will be studied, a database of 25 different features and patters will be used to construct K nearest neighbors and random forest classifiers, for latter classification of test images.

Keywords: Classification, features, textons.

1.. Introduction

Features is one of the most complex information we can get from an image, it's not intuitive how to approach problems with features as is not as simple as extracting pixel values. We can understand features as recurring patterns that have similar characteristics as orientation, gradient changes and colors.

In this laboratory, we intend to classify a database into 25 feature categories training a KNN (K Nearest Neighbor) and Random Forest classifiers with training data. To represent the training data, we create a filter bank of textons to generate histograms and train both classifiers in hopes of generating good predictors for the test data.

2.. Methods and Materials

2.1.. Database

The database has 1000 images, separated in two categories: training and test. The training sets contains 30 images for each of the 25 semantic categories while the test category only has 10 for each, the images are of the same size and are of different ripes of features at different orientations.

2.2.. Textons

Textons are the "atoms" of features, their formal definition is as basic patterns that combined can generate more complex patterns, they are usually lines, dots and spots with different sizes and directions. Using textons we can identity the difference between local patters, using a similar technique of template matching, we can observe the response of filtering and decide which textons can better describe a pattern.

A library of functions for Matlab was used to generate the filter bank and generate the feature representation in the training data.

2.3.. K Nearest neighbor

The nearest neighbor method is one of the most basic method in classification, in this case it takes into account the k nearest neighbors to decide the label of new data that enters the representation space. To compare between the training data, the Chi-square distance between histograms was used.

2.4.. Random Forest

Random forest uses a similar "voting" system as the KNN method, although is based in a collection of single classification trees in which the input data is introduce and each tree produces a classification to the input, the most frequent classification stays. It's a more accurate and efficient method in comparison to the KNN classification, it takes into account the information gain of the different classification distributions that it can produce.[1]

2.5.. Subsampling stategy

2.5.1. KNN

To optimize computation times, both sets were divide in groups and process in different orders. In the training group, 30 groups were formed taking one image of every categories each, with this groups different classifiers were trained. In the other side, the test data was divided in 10 groups of 25 image each, their annotation and predicted categories for each of the 30 classifiers were saved for latter processing. All this sub sampling was necessary, as the servers and computers couldn't take the 750 images of training and assemble the filter banks and a unique classifier, or classify directly the set of test images.

2.5.2. Random Forest

Here subsampling was only used for the filter bank construction, only three images were used per category to create the bank. This was due to the objective of training only one classifier and that using more that 3 images will make Matlab run out of memory. The representaion of training and test data was perform in the servers, but the classifier training and classification was able to run on a laptop.

2.6.. Code strategy

2.6.1. KNN

Three separated codes were develop to satisfy the subsampling problem, one would train the different classifiers, the other would classify the test images with the previous classifiers and filter banks, and lastly the third code would construct the confusion matrix based in the predictions. To allow communication between codes and avoid time loss do to Matlab problems, variables were constantly save in Matlab's own file format (.mat).

2.6.2. Random Forest

As the previous strategy, three different codes were used, one for the training of a classification tree (fitctree) and a random forrest (TreeBagger) classifiers, other for classification with both classifiers and a last one for confusion matrix generation. The codes communication was used with .mat files that containing the filter bank, the classifiers and the prediction results.

3.. Results and Discussions

3.1.. KNN

As it can be appreciated in the figures 1 through 3, the Average Classification Accuracy was of 39,2%, this was probably do to a problem in subsampling the training sets and how it was decided which category each image belong to.

With 30 classifiers, you obtained 30 prediction per test image, for time constraints, the mode of this 30 predictions was used as the label for each image. This didn't take into account the accuracy of each prediction, which could improve the result. Training one classifier with the right parameters could produce a more accurate prediction.

Additionally, there was some parallel work between building the classifiers and using them instantly to classify the test images, for this reason it was necessary to assemble the responds of different groups of classifiers in the final code (Confusion.m) to assemble the final confusion matrix.

3.2.. Random Forest

As one can appreciate in both figures 4 and 5, both random forest classifiers achieve a higher ACA that the KNN general method. Using TreeBager prove to be more effective as the number of trees was specify to 25, the ACA for this method was 66,8% which is to say 2 of every 3 classifications are correct. Using fitctree, which trains a binary classification tree, as a comparison resulted in a 47,6% ACA, this is better that the KNN method but fails in comparison to the TreeBagger method.

This shows the robust power of a random forest which can accomplish more that a single classification tree, as it takes into account the opinion of multiple classifiers.

4.. Conclusions

To construct a filter bank as big as the problem requires, a big computational time has to be used, and based in the used of k means for the nearest neighbor method, it last even more as it tries to converge every time. The random forest classifier provided a more time efficient and accurate method,that with better training data and parameters, could generate a more robust classification method.

Perhaps the use of less images, the change and optimization of parameters and a more advance classification method would have produce a better result in the end, but the results acquire with the TreeBagger Function are relativity good for this problem.

References

[1] Breiman, L. & Cutler, A. Random forests - classification description. Stat.berkeley.edu. Retrieved 8 March 2017, from https://www.stat.berkeley.edu/ breiman/RandomForests/cc_home.htm

Images

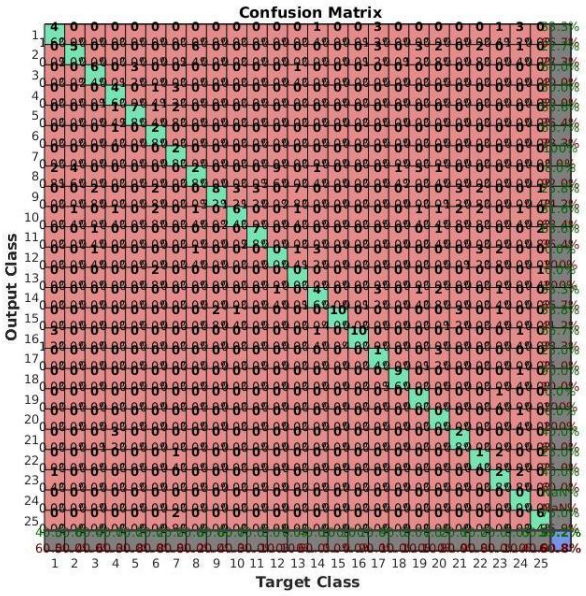

Figure 1. Full confusion matrix for KNN classifier

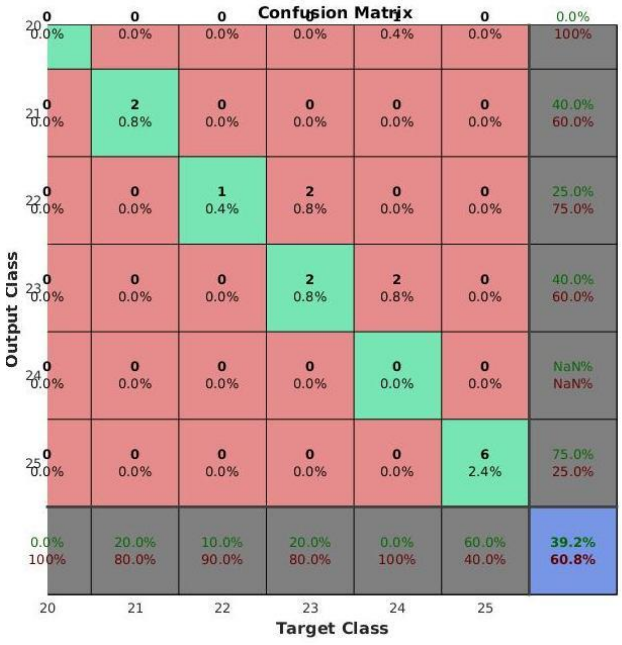

Figure 2. Closer look at the confusion matrix

Figure 3. Average Classification Accuracy for KNN classifier

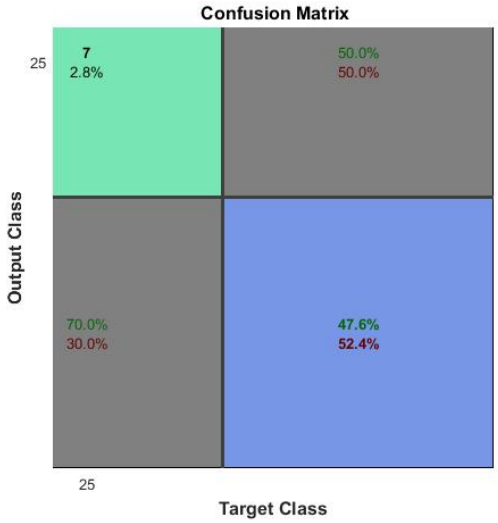

Figure 4. Average Classification Accuracy for the first random forest classifier

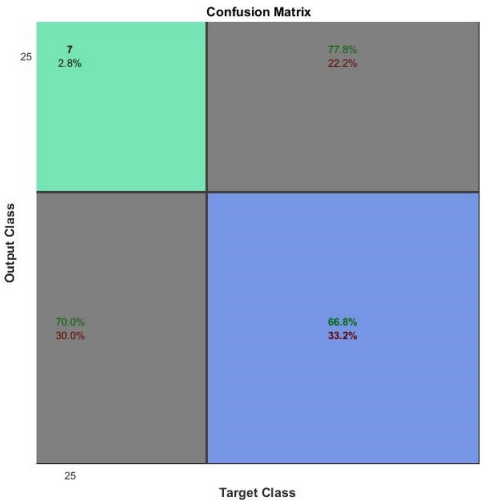

Figure 5. Average Classification Accuracy for the second random forest classifier