

# SPS : CW2 Report

Matthew Duffin

April 19, 2016

## 1 Introduction

This report describes the process of designing a classifier for a set of characters. By extracting features from the fourier space images of these three types of character (S, T and V) I have been able to build a reasonably accurate classifier.

## 2 Feature Selection

The first step in the process is to find the features that clearly separate the three types of character. We were asked in this task to extract features from the fourier space. As a starting point I created the fourier space images for each of the characters and then took the average of each class. The results of this are shown below.

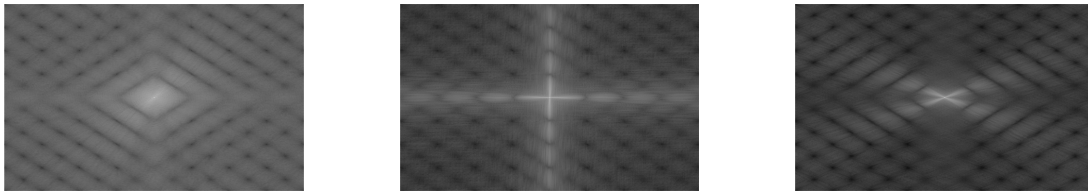


Figure 1: Average fourier spaces of S (left) T (centre) and V (right).

You can see that the fourier space for the letter T has very strong spectral peaks in the horizontal and vertical directions. The vertical line tells us that in the original image there is a high rate of change of intensity in the horizontal direction. The horizontal line shows the same for the vertical direction. Looking at the letter T you can indeed see that the two lines that make it up correspond to the strong bright vertical and horizontal lines in the fourier space.

At this point I looked at the three images and tried to find regions where the intensity differed. For example, the strong horizontal and vertical lines that are present in the T fourier space are not present in the others. Likewise, the diagonal lines in V's fourier space are not present in the others. Using this information I selected five different regions to take forward and test, which can be seen in Figure 2.

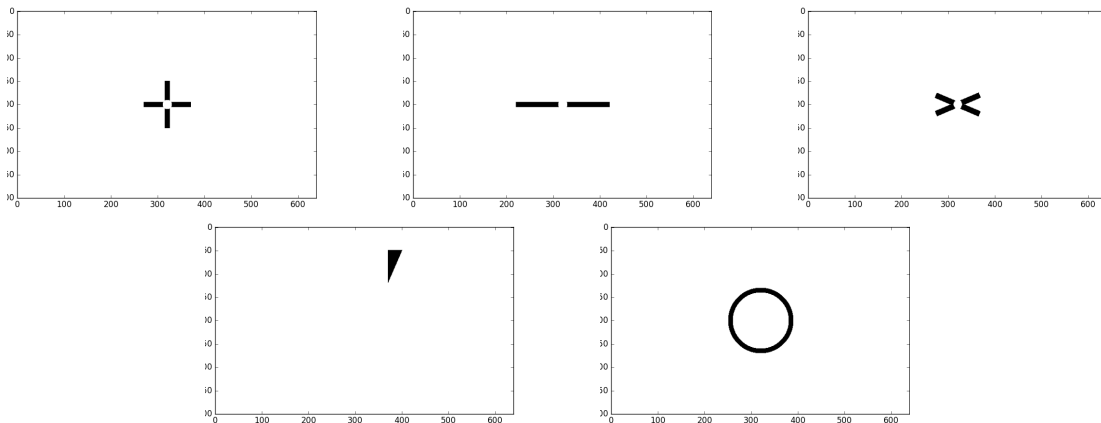


Figure 2: Features selected to take forward to testing.

You will notice that a lot of the features have a "hole" in the middle. This is because I wanted to exclude the DC component from the features. The DC component is the average brightness of the image. It does not give any information about direction. Therefore its inclusion would be likely to skew my results if a certain image were brighter than the others.

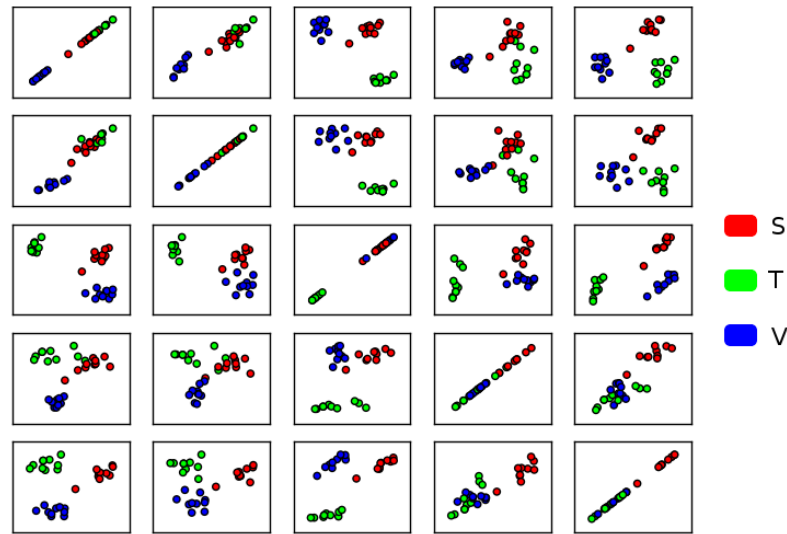


Figure 3: A grid of plots, each of which plots one feature against another.

Next, I took the log of the magnitude spectrum, squared it to get the power and then summed up the region defined by the feature.

In Figure 3 I have plotted the features against each other to form a grid of plots. The features are numbered 0 to 4 downwards and 0 to 4 from left to right. There are a number of good candidates, such as (0,2) and (2,4) but I decided to take (0,4) forwards (the cross and the ring) because apart from one wayward point it separates the data into three well defined clusters. A more detailed plot of these two features can be seen in Figure ??.

### 3 The Classifier

#### (a) Training The Classifier

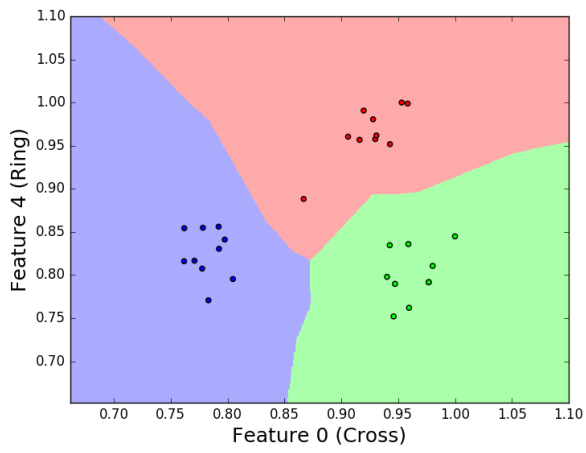
Figure ?? shows that the selection of features 0 and 4 was a good one. The K-Nearest Neighbours algorithm has produced a fairly well-defined decision boundary. Only one point sits right on the decision boundary between S and V. This is fine, as it shows we are avoiding over-fitting the model to the test data.

I must also mention that the values supplied to the K-Nearest Neighbour algorithm were normalised first by dividing each feature by the largest value. This had no effect on the classification but just made the scale easier to interpret.

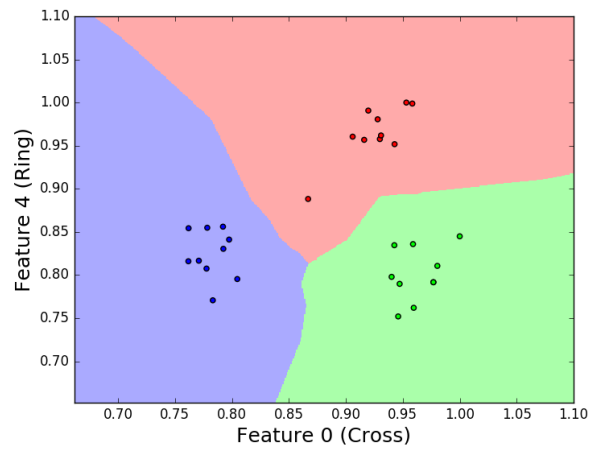
#### (b) Testing The Classifier

### 4 Nearest-Centroid Classification

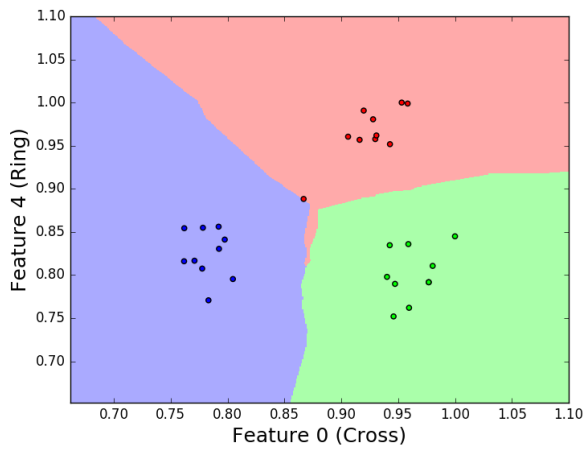
The next stage of our analysis was to start classifying new data. We started by using the centroids that we found using K-Means as a simple nearest-neighbour classifier. You can see the results in Figures ?? and ?. The crosses represent the test data. As our features had high inter-cluster distance the nearest neighbour produced a fairly good assignment of data points. However, we could not test that it was getting the right result every time as we did not know which cluster each test point was *supposed* to belong to.



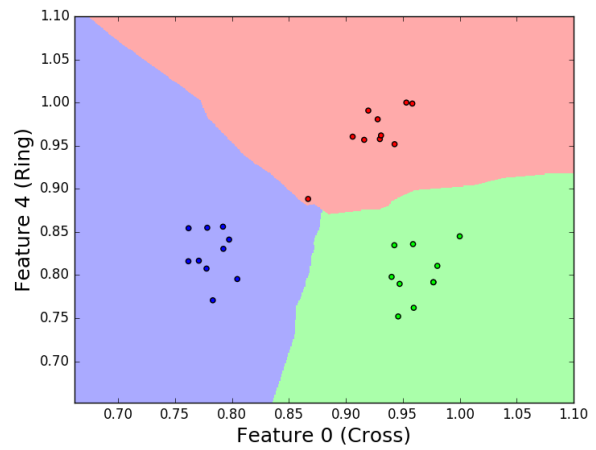
(a) K=1



(b) K=2



(c) K=3



(d) K=4

Figure 4: K-Nearest Neighbour Classifiers with  $K=1..4$

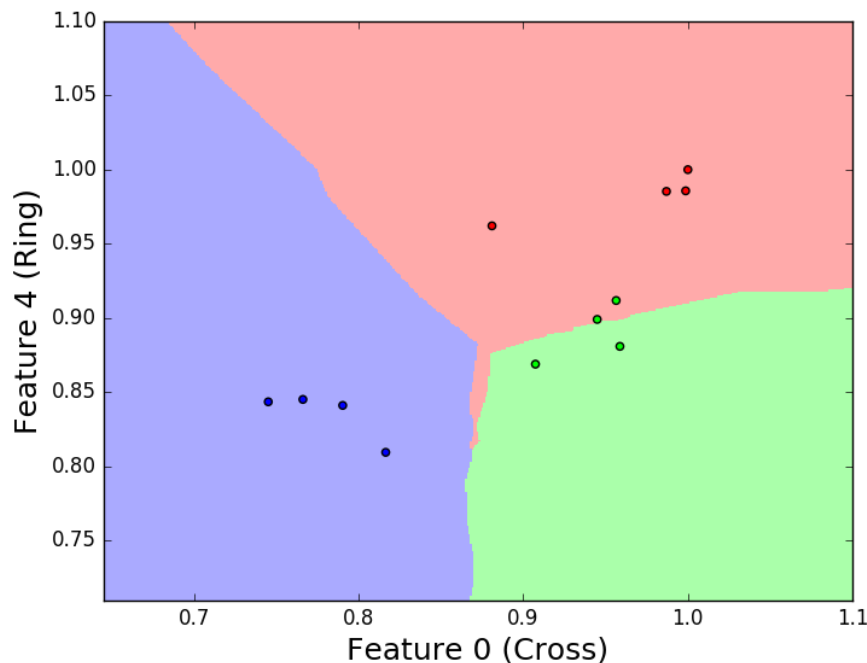


Figure 5: The K-Nearest Neighbour classifier with  $K=3$ . The test data has also been plotted.

## 5 Maximum-Likelihood Classification

Next we used a different classification method: maximum-likelihood classification (with each class modelled as being drawn from a 2-D Normal Distribution). This method has certain advantages over nearest-neighbour classification. In ML classification we need to estimate mean vectors as well as full covariance matrices for each class. This may result in a non-linear decision boundary if the covariance matrices are not the same across classes.

Our first task was to plot contours around each cluster such that 95% of the probability mass was inside the contour. We did this by finding the points where the squared Mahalanobis distance from the mean was 6. The Mahalanobis distance is a multi-dimensional measurement of how many standard deviations a point is from the mean [1].

Next we plotted the decision boundaries. These allow us to classify future data as belonging to one of the three clusters. To generate these boundaries we found the points where the probabilities of belonging to cluster A's distribution and cluster B's distribution were equal.

We then considered how you could change the maximum-likelihood classifier so that its decision boundaries were the same as the ones we found using the nearest-centroid method. We found that you can use the identity matrix as the covariance matrix for the two distributions instead of their true covariances. This way there appears to be no correlation between the features within each cluster. Also, the two covariance matrices are identical. This creates a straight decision boundary that bisects the line between the two centroids.

We also experimented with adding a prior to one of the three classes. Figures ?? and ?? show how the decision boundaries change (dotted line) when one of the classes is twice as likely.

## 6 Discussion

In conclusion, we agreed that the decision boundaries we achieved would be sufficient to assign new data points to our clusters. The use of Maximum-Likelihood Classification would seem more accurate for deciding points which are near the decision boundary, as our clusters had different covariance values from one another.

The data we were given demonstrates how much information you can gain without any prior knowledge of where data was sourced. If we had more knowledge we could have used a MAP estimation for a

final stage allowing for better results. However, we found that even giving one class a prior of 2 times the rest did not move the decision boundaries very far. This indicates that only a strong prior would affect classification greatly.

## 7 Sources

- [1] Mahalanobis Distance Wikipedia, [https://en.wikipedia.org/wiki/Mahalanobis\\_distance](https://en.wikipedia.org/wiki/Mahalanobis_distance), Retrieved 11th March 2016.
- [2] Maximum Likelihood Wikipedia, [https://en.wikipedia.org/wiki/Maximum\\_likelihood](https://en.wikipedia.org/wiki/Maximum_likelihood), Retrieved 11th March 2016.
- [3] Multivariate Normal Wikipedia, [https://en.wikipedia.org/wiki/Multivariate\\_normal\\_distribution](https://en.wikipedia.org/wiki/Multivariate_normal_distribution), Retrieved 11th March 2016.
- [4] K-Means Wikipedia, [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering), Retrieved 11th March 2016.