Introduction to Intelligent Systems
Lab Session 2

Group 30
Sergio Calogero Catalano (s3540294) & Emmanouil Gionanidis (s3542068)

September 25, 2017

Overview

To complete this exercises, as with the previous assignment, we did a lot of work together in the lab session. However, this time the third exercise was a little bit longer, forcing us to meet after the lab session to write it down together. After that we just wrote the latex document, working separately. This time the Matlab language has been easier to use, having already had contact with it, even if it still took a lot of time to make it work properly.

Assignment 1

- **Figure 1** shows the histograms of the heights in the same figure. The height of men is in blue, the height of women is in green. They partially overlap because, as expected, there are some men who are as tall as some women.

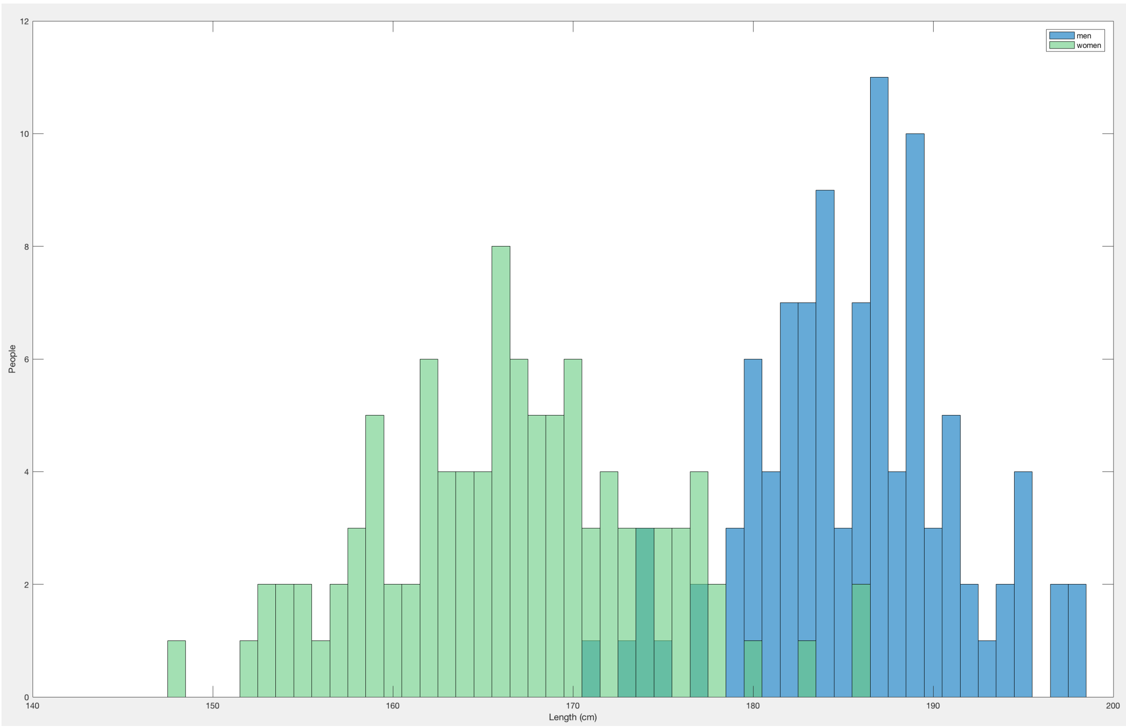

Figure 1: *Histogram for Assignment 1: Shows the height of men (blue) and of women (green)*

- If we choose a decision criterion of 170cm (we classify a person who is 170cm tall as a man), all the men in the provided dataset are classified correctly, but 35 women are misclassified as men. This means that men have an error rate of 0% while women have an error rate of 35%

- To minimize the total number of misclassifications, the best decision criterion is 179cm, as with that value only 12 people are misclassified. **Figure 2** shows the number of misclassifications depending on the decision criterion. From the plot it is obvious that the best choice is 179cm.

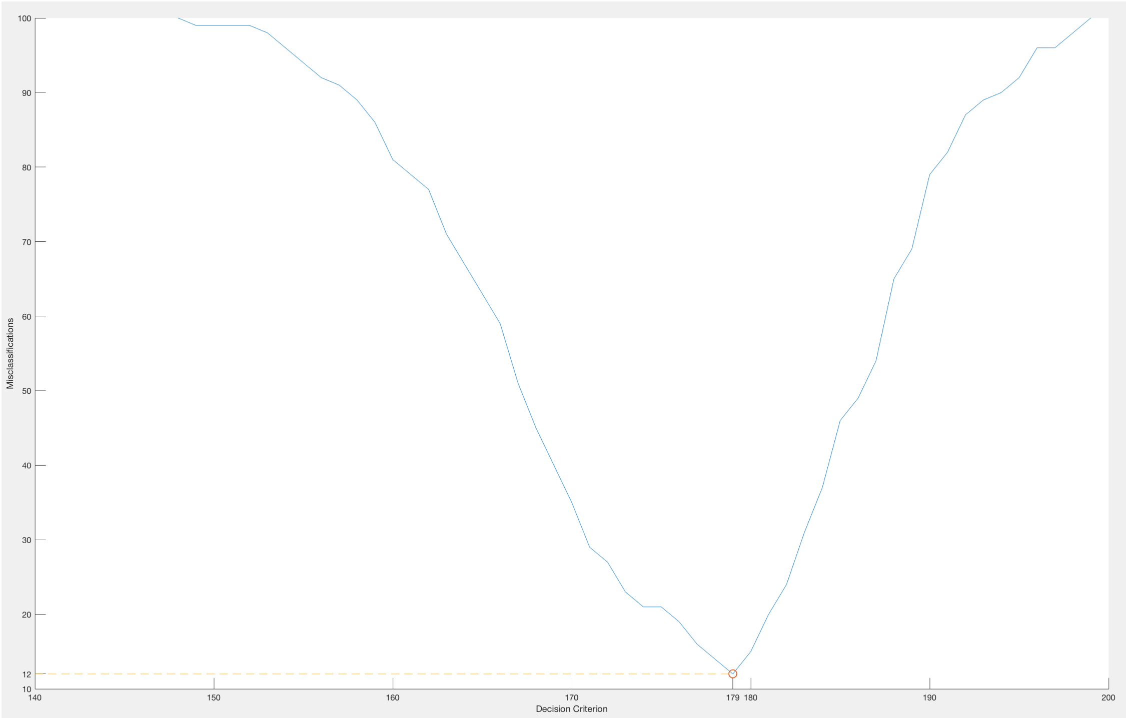

Figure 2: *Plot for Assignment 1: Shows the number of misclassifications depending on the decision criterion*

Assignment 2

- **Figure 3** shows a graph containing a point for each pair (height, hair length).

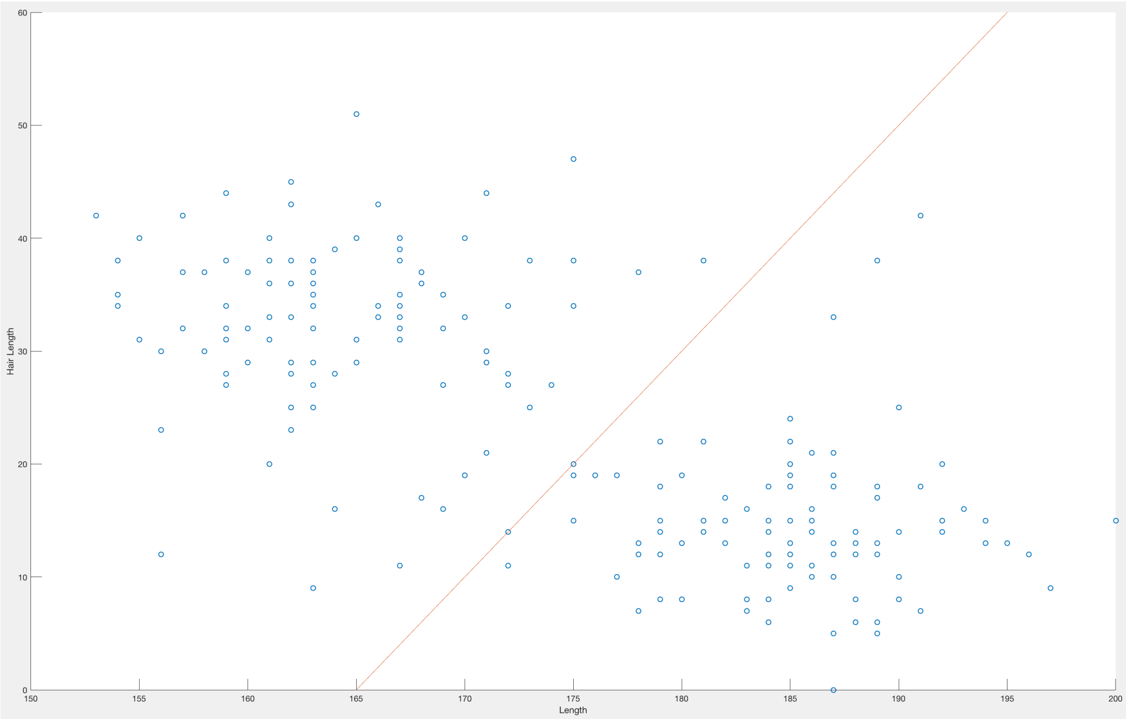

Figure 3: *Plot for Assignment 2: Shows the height versus the hair length*

- The plot in **Figure 3** shows clearly that there are 2 aggregations of points: the first one, most probably representing men, is down on the right, while the second one, probably representing women, is up on the left. Since the plot shows height and hair length, we should probably decide how each of them influences the decision boundary. Considering that hair length is a subjective matter (it is common to see women with short hair or viceversa men with long hair, they just cut it or let it grow as they like) while height is genetic, we chose to give more importance to the height than to the hair length to draw our decision boundary. Thus we are drawing a line, represented by a function which has 2 as slope (as the height influences the decision more than the hair length). To know exactly where to draw the line, we also forced it to divide the dataset in two sets of points, each with cardinality equal to 100, as we know that there are 100 men and 100 women. Putting these 2 conditions together, we plotted in the graph the line y = 2x - 330

Assignment 3

The complete code used for this assignment is in Appendix A.

- Analyzing the given data, it is possible to notice that two iris codes from the same person differ, on average, only in 2.551 bits, with a minimum difference of 0 and a maximum difference of 6; while two iris codes from different people differ, on average, in 14.8202 bits, with a minimum difference of 3 and a maximum difference of 25. Thus it is obvious that, generally, if two iris codes differ only in less than 3 bits, it is very likely that the iris codes come from the same person, while if they differ in more than 6 bits, it is very likely that they are from two different people.

- **Figure 4** shows the histograms obtained by computing the hamming distances for set S and set D. These histograms overlap only around the value 0.2 because, as observed earlier, the only times the hamming distance of two iris codes from the same person is equal to that of iris codes from different people is when the distance is between 3 and 6, and 0.2 is obtained by normalizing 6. The fact that the histograms overlap implies that there is not a simple decision criterion that makes it 100% certain whether two given iris codes are from the same person or from different people.

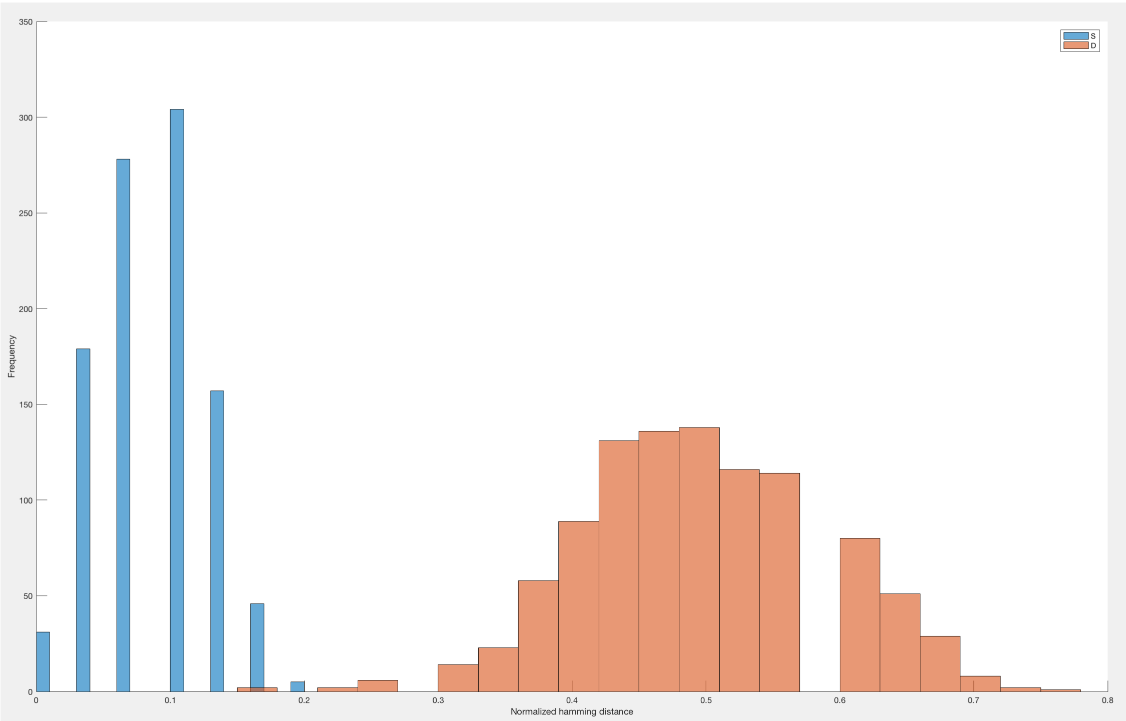

Figure 4: *Histograms for Assignment 3: Shows the sets S and D in the same figure*

- To compute the means and variances and to plot them, the code in **Listing 1** was used:

Listing 1: *Code to plot the normal functions with the means and variances of the sets S and D*

```
mean_S = mean(S); std_S = std(S);
mean_D = mean(D); std_D = std(D);
x = linspace(0, 1, 1000);
histogram(S);
histogram(D);
```

```
    plot(x, normpdf(x, mean_S, std_S), 'LineWidth', 3);
    plot(x, normpdf(x, mean_D, std_D), 'LineWidth', 3, 'Color',
'green');
```

The gaussian functions fit the histograms well enough, since the histograms are taller around their mean and go down as the gaussians do. The normal function is generally a good way to create a mathematical model, given a set of data to analyze. The model itself is really useful, also to predict the outcome for new data. Figure 5 shows the histograms with the gaussian functions.

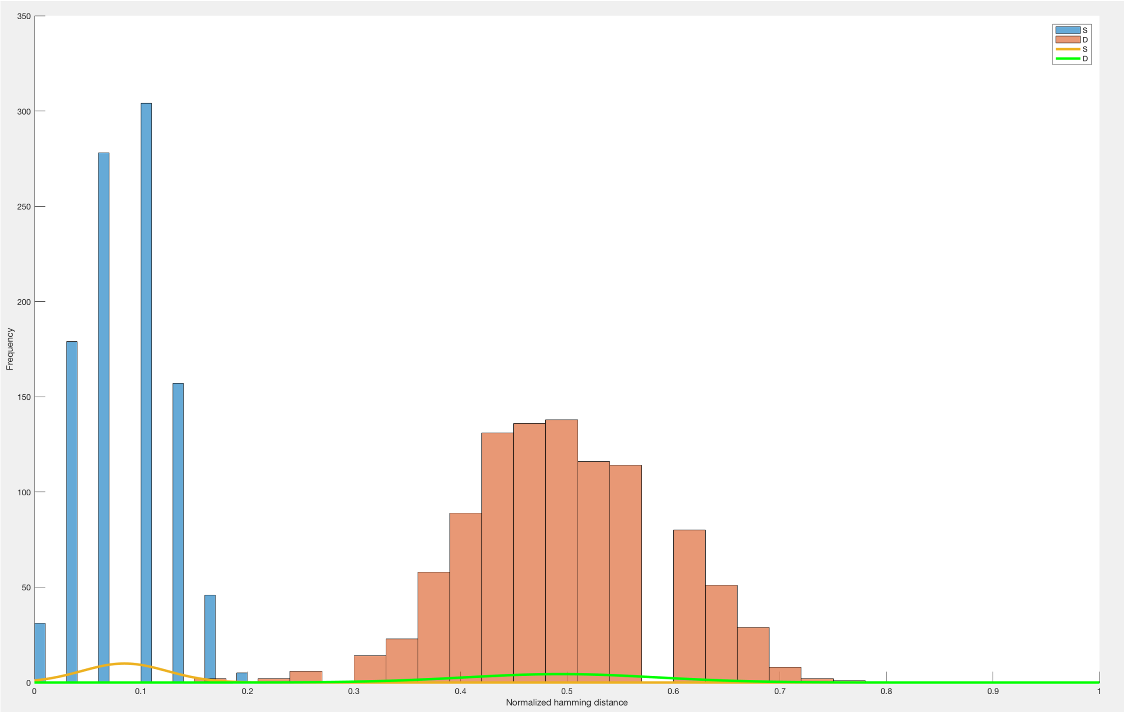

Figure 5: *Histograms and Normal Functions for Assignment 3: Shows the sets S and D in the same figure*

- False acceptance represents the cases in which the iris codes of two different people are similar enough to be evaluated as coming from the same person. In our gaussian model, this is represented by the area underneath the function of D, for x lower than the decision criterion. What we need is a system that has a very low false acceptance rate, so that it is very rare that an unauthorized user is granted some rights that he should not have. Reducing the false acceptance actually increases the value of the false rejection, since the latter is the area underneath the function of S, for x greater than the decision criterion. So if we move the decision criterion to the left to reduce the false acceptance, we also increase the false rejection. But false rejection is a lesser problem in real situations, since it requires only to gather again the iris code and retry the calculation. Thus in this assignment we are asked to choose a decision criterion which reduces the false acceptance error to 0.0005. This is done with the code in Listing 2:

Listing 2: *Code to compute the value of the decision criterion to meet the required false acceptance rate*

```
    cdf_D = normcdf(x, mean_D, std_D);
    cdf_S = normcdf(x, mean_S, std_S);
```

```
target = 0.0005;
[difference, index] = min(abs(cdf_D - target));
xtarget = x(index);
target_cdf = cdf_S(round(xtarget*1000));
fprintf("Target false acceptance: %f\nResulting decision
criterion: %f\nResulting false rejection: %f\n",target, xtarget
, 1-target_cdf);
```

We can also analyze the current situation with the plot in Figure 6, which shows only the two gaussian functions and the vertical line, which represents the required decision criterion. As shown in Figure 6, the decision criterion required to keep false acceptance at 0.0005 is 0.1972, while the corresponding false rejection rate is: 0.002649. To show the relationship between false acceptance rate and false rejection rate, we also considered another value of false acceptance, to see the outcome. As expected, if we allow a higher false acceptance error, the false rejection rate decreases. To be more easily understandable, we also refer to the value of the second false acceptance tested and to the false rejection obtained with it: with a higher false acceptance of 0.005000, we have a smaller false rejection of 0.000005, with the decision criterion set to 0.261261(higher than the previous value as well).

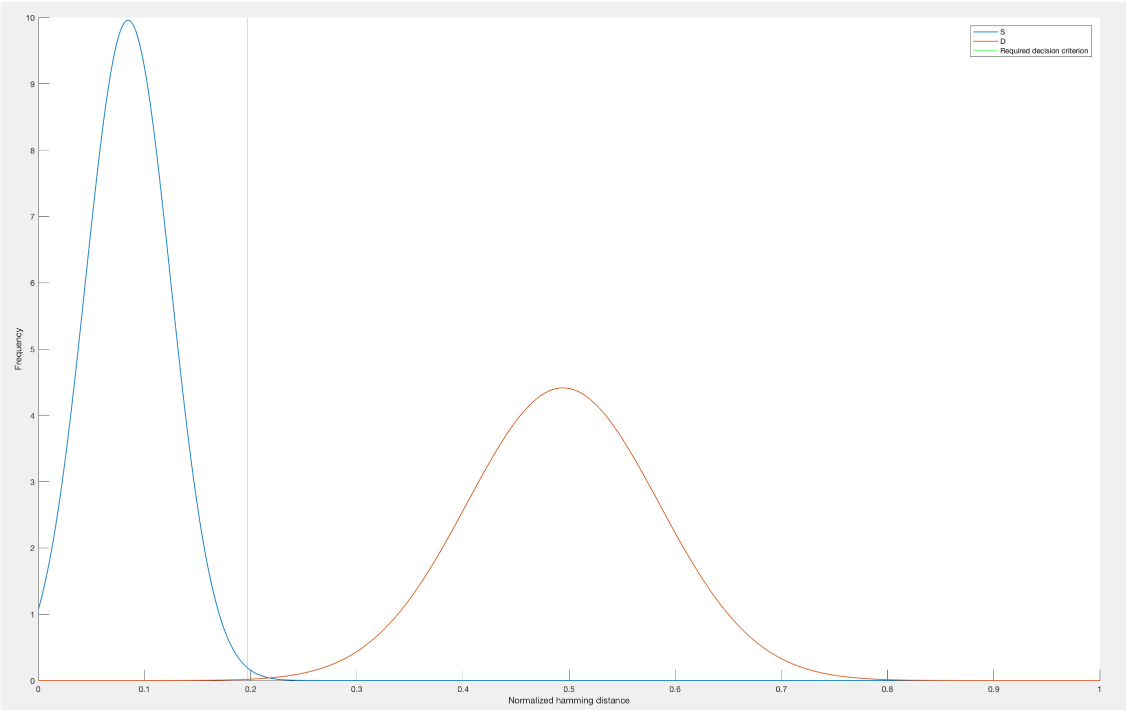

Figure 6: *Normal Functions for Assignment 3: Shows the required decision criterion*

A. Assignment 3

Listing 3: *This is the file `script3.m`*

```
function script3(~)

    %Load data
    codes = zeros(20, 30, 20);
    A = dir('person*.mat');
    for i = 1:20
        S = load(A(i).name);
        codes(:, :, i) = S.iriscode;
    end

    %Compute S and D
    S = zeros(1,1000);
    D = zeros(1,1000);
    for i = 1:1000
        person = randi(20);
        rows = randperm(20, 2);
        S(i) = HD(codes(rows(1), :, person), codes(rows(2), :, person)
   )/30;
        row2 = randi(20);
        D(i) = HD(codes(person, :, rows(1)), codes(row2, :, rows(2)))
   /30;
    end

    %Plot S and D
    figure(4);
    hold on;
    hAx=gca;
    xlabel ( hAx, 'Normalized hamming distance' );
    ylabel ( hAx, 'Frequency' );
    histogram(S);
    histogram(D);
    hold off;

    legend ('S', 'D');

    %Plot the gaussan functions that fit S and D
    figure(5);
    hold on;
    hAx=gca;
    xlabel ( hAx, 'Normalized hamming distance' );
    ylabel ( hAx, 'Frequency' );

    mean_S = mean(S); std_S = std(S);
    mean_D = mean(D); std_D = std(D);
    x = linspace(0, 1, 1000);
    histogram(S);
    histogram(D);
```

```
    plot(x, normpdf(x, mean_S, std_S), 'LineWidth', 3);
    plot(x, normpdf(x, mean_D, std_D), 'LineWidth', 3, 'Color', 'green'
   );
    hold off;

    legend ('S', 'D', 'S', 'D');

    %Compute the decision criterion
    figure(6);

    cdf_D = normcdf(x, mean_D, std_D);
    cdf_S = normcdf(x, mean_S, std_S);

    %Requested target
    target = 0.0005;
    [difference, index] = min(abs(cdf_D - target));
    xtarget = x(index);
    target_cdf = cdf_S(round(xtarget*1000));
    fprintf("Target false acceptance: %f\nResulting decision criterion:
     %f\nResulting false rejection: %f\n",target, xtarget, 1 -
   target_cdf);

    %Show the decision criterion
    hold on;
    hAx=gca;
    xlabel ( hAx, 'Normalized hamming distance' );
    ylabel ( hAx, 'Frequency' );
    plot(x, normpdf(x, mean_S, std_S), 'LineWidth', 1);
    plot(x, normpdf(x, mean_D, std_D), 'LineWidth', 1);
    line([xtarget xtarget], [0 10], 'Color', 'green');
    hold off;

    legend ('S', 'D', 'Required decision criterion');

    %Additional target
    target = 0.005;
    [difference, index] = min(abs(cdf_D - target));
    xtarget = x(index);
    l = cdf_S(ceil(xtarget*1000));
    fprintf("Additional false acceptance: %f\nResulting decision
   criterion: %f\nResulting false rejection: %f\n",target, xtarget, 1-
   l);

end

%Function that calculates Hamming Distance between two iris codes
function ham = HD(r1, r2)
    ham = 0;
    for j = 1:30
        if r1(j) ~= r2(j)
            ham = ham + 1;
```

```
        end
    end
end
```