Introduction to Intelligent Systems
Lab 1

Team 15:
Sara Bardají Serra and Albert Sallés Torruella

September 2019

1 Assignment 1

Given are two sets of measured lengths (in cm) of men (length men)and women (length women) in the file lab1 1.mat.

Exercise 1. Plot histograms of both sets in one figure.

```
histogram(length_men)
hold on
histogram(length_women)
title("Men and women's lengths")
xlabel('Length(cm)')
legend('Men', 'Women')
hold off
```

Code from exercise 1

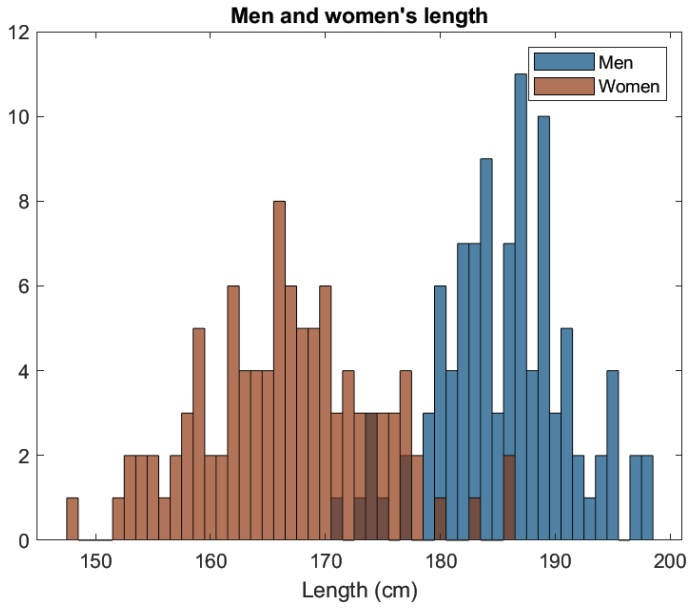

Figure 1: Histogram for both men and women's lengths.

Exercise 2. Now choose the decision criterion at 170 cm. How many men are classified incorrect? And how many women?

```
hold on

% Draw a vertical line at 170 cm
line([170, 170], ylim, 'LineWidth', 3, 'Color', 'r')
legend('Men', 'Women', 'Decision criterion')

hold off

% Misclassified men are shorter or equal than 170 cm
nIncMen = length(length_men(length_men <= 170));
fprintf('Men classified incorrect: %i', nIncMen)

% Misclassified women are taller than 170 cm
nIncWomen = length(length_women(length_women > 170));
fprintf('Women classified incorrect: %i', nIncWomen)
```

Code from exercise 2

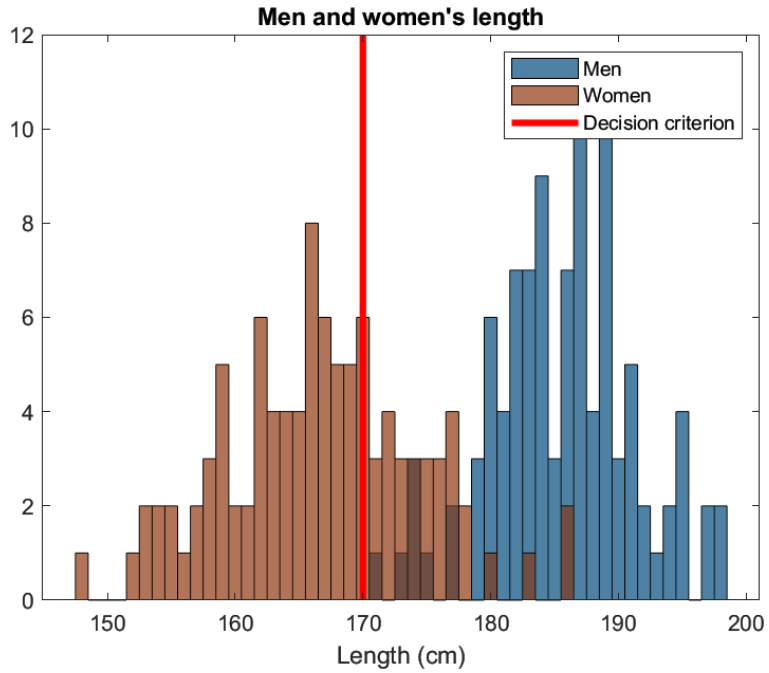

Figure 2: Histogram for both men and women's lengths with the decision criterion set at 170cm.

The output we obtained from exercise 2 is as follows:

```
Men classified incorrect: 0
Women classified incorrect: 29
```

As we can see from the output, we have found that by setting the decision criterion at 170cm we misclassify 0 men and 29 women.

Exercise 3. What decision criterion should be used to minimize total number of mis-classifications (sum over men and women)?

```
    min = nIncMen + nIncWomen;
    d_criterion = 170;

    % The optimal decision criterion is between 150 cm and 200 cm
    for i = 150:200

        % Firstly, calculate misclassifications
        nIncMenAux = length(length_men(length_men <= i));
        nIncWomenAux = length(length_women(length_women > i));

        % If it's lower than the minimum, then it becomes the new minimum
        if min > nIncMenAux + nIncWomenAux
            min = nIncMenAux + nIncWomenAux;
            d_criterion = i;
            nIncMen = nIncMenAux;
            nIncWomen = nIncWomenAux;
        end
    end

    fprintf('Best decision criterion: %i', d_criterion)
    fprintf('Men classified incorrect: %i', nIncMen)
    fprintf('Women classified incorrect: %i', nIncWomen)
```

Code from exercise 3

The output we obtained for exercise 3 is as follows:

```
Best decision criterion: 178
Men classified incorrect: 8
Women classified incorrect: 4
```

Therefore, we have found that the decision criterion that should be used is 178cm as it minimizes the total number of mis-classifications.

2 Assignment 2

In the file lab1 2.mat a two dimensional array is given, consisting of measurements of the length (in cm) and the hair length (in cm) of 200 people.

Exercise 1. Plot the length versus the hair length.

```
    people_length = measurements(:,1);
    hair_length = measurements(:,2);
```

```
3
4       plot(people_length, hair_length, 'r.')
5       title('People length vs hair length')
6       ylabel('Hair length (cm)')
7       xlabel('People length (cm)')
```

Code from exercise 1

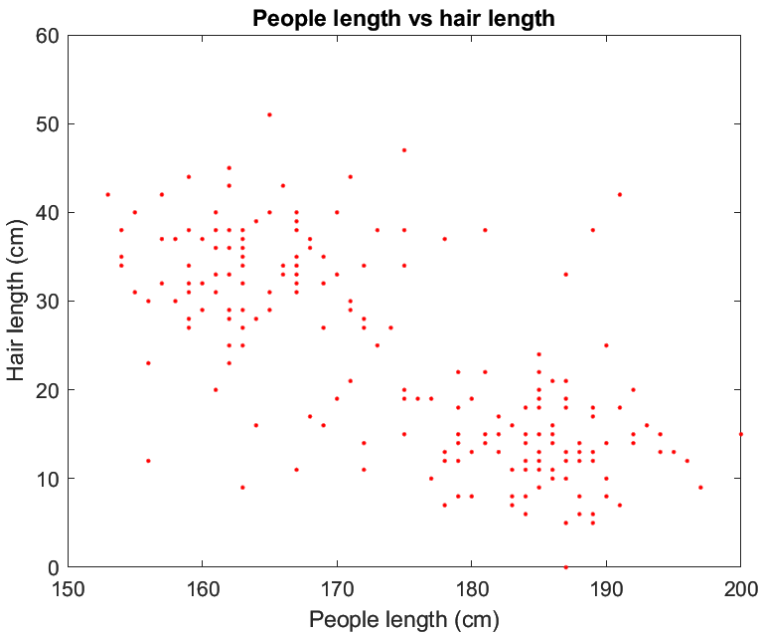

Figure 3: Graph showing people's length versus their hair length

Exercise 2. The measurements originate from 100 women and 100 men. Given the fact that in general men have shorter hair than women and men are taller, where would you draw the decision boundary (for example use a simple graphics editor to sketch it, or just plot a line between two points on top of your plot in matlab) and why?

```
1       line(xlim, ylim)
```

Code from exercise 2

In order to draw our decision boundary, we've decided to cut the plot in half following its diagonal from its bottom-left hand corner to it top-right hand corner. We've decided to use this decision boundary because men tend to be taller and have shorter hair, so they are more likely to be found in the lower right-hand corner of the plot. On the other side, women tend to be shorter and have longer hair, therefore they are more likely to be found in the upper left-hand corner of the plot.

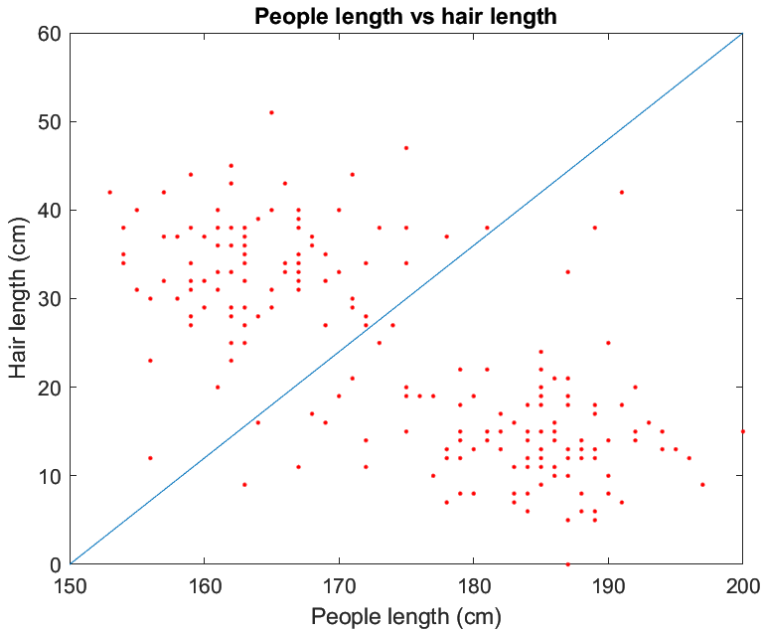

Figure 4: Graph showing people's length versus their hair length with a decision boundary

3 Assignment 3

Consider the two dimensional binary arrays in files person01.mat to person20.mat. Each row of such an array person[i].m is a binary feature vector of 30 elements that is extracted from an iris image of a person that we call here person[i] (i = 1; : : : ; 20). Hence, each row is a 30-dimensional binary iris code of that person. There are 20 such iris codes of each person in the corresponding file person[i]; each row of the array is one such binary iris code.

First of all, in order to load all the files, we've used the following code:

```
% To load files person[i].mat with i = 1..2
for col = 1:20
    % Generate the file name person[i].mat, [i] is a two-digit number
    file = strcat("person" + num2str(col, '%02d') + ".mat");
    % Load each file and save it in an array
    person(col) = load(file);
end
```

Code used to load the files

Also, in order to complete this exercise more easily, we've generated the following function that allows us to calculate the hamming distance between two given arrays.

```
function y = hd(b1, b2)
    y = 0;
    for n = 1:30
        if b1(n) ~= b2(n)
            y = y + 1;
```

```
            end
        end
    end
```

Function to calculate the hamming distance between two arrays

Exercise 1. Take a closer look at the rows of one such array and notice that two rows can differ in only a few positions (bits). Compare now two rows that come from two different files person[i] and person[j]. Notice that two such iris codes differ in about 15 positions.

```
    % Take a random person
    i = randi(20);
    % Take the number of rows in the iriscode
    nrows = numel(person(i).iriscode(:,1));
    % Take the number of columns
    ncols = numel(person(i).iriscode(1,:));
    % Initialize a matrix where all the differences between different rows
        will
    % be stored
    diffRows = zeros(20);

    % Loop from each row to the end, so we compare two different rows only
        once
    for row1 = 1:nrows-1
        for row2 = row1+1:nrows
            % Calculate the differences between both rows
            diff = 0;
            for col = 1:ncols
                if person(i).iriscode(row1,col) ~= person(i).iriscode(row2
                    ,col)
                    diff = diff + 1;
                end
            end

            diffRows(row1, row2) = diff;
            diffRows(row2, row1) = diff;
        end
    end
```

Part of the code used for exercise 1

With this code what we've done is created a 20x20 matrix so that for a random person we compare every row with every other row to see in how many bits each row differs with another. The matrix we've obtained is as follows:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	2	1	2	1	1	2	1	1	0	2	2	1	1	2	1	2	3	2	2
2	2	0	3	4	3	3	0	3	3	2	4	4	3	1	4	3	4	3	4	4
3	1	3	0	3	2	2	3	2	2	1	3	3	2	2	3	2	3	4	3	3
4	2	4	3	0	3	3	4	3	3	2	4	2	3	3	4	3	4	3	2	4
5	1	3	2	3	0	2	3	2	2	1	1	3	2	2	3	2	3	2	3	3
6	1	3	2	3	2	0	3	2	2	1	3	3	2	2	3	2	3	4	3	3
7	2	0	3	4	3	3	0	3	3	2	4	4	3	1	4	3	4	3	4	4
8	1	3	2	3	2	2	3	0	2	1	3	3	2	2	3	0	3	4	3	1
9	1	3	2	3	2	2	3	2	0	1	3	3	2	2	3	2	3	4	3	3
10	0	2	1	2	1	1	2	1	1	0	2	2	1	1	2	1	2	3	2	2
11	2	4	3	4	1	3	4	3	3	2	0	4	3	3	4	3	4	3	4	4
12	2	4	3	2	3	3	4	3	3	2	4	0	1	3	4	3	2	3	4	4
13	1	3	2	3	2	2	3	2	2	1	3	1	0	2	3	2	1	4	3	3
14	1	1	2	3	2	2	1	2	2	1	3	3	2	0	3	2	3	4	3	3
15	2	4	3	4	3	3	4	3	3	2	4	4	3	3	0	3	4	5	4	4
16	1	3	2	3	2	2	3	0	2	1	3	3	2	2	3	0	3	4	3	1
17	2	4	3	4	3	3	4	3	3	2	4	2	1	3	4	3	0	5	4	4
18	3	3	4	3	2	4	3	4	4	3	3	3	4	4	5	4	5	0	5	5
19	2	4	3	2	3	3	4	3	3	2	4	4	3	3	4	3	4	5	0	4
20	2	4	3	4	3	3	4	1	3	2	4	4	3	3	4	1	4	5	4	0

Figure 5: Table obtained from comparing every row with every other row, from a random person, to see in how many characters they differ.

As we can see from looking at the table, this person's iris codes differ a maximum of 5 values.

```
% Take two different people
i = 3;
j = 6;
diff = 0;
% Calculate the differences (We chose the first row of each person's
    iriscode)
for col = 1:ncols
    if person(i).iriscode(1,col) ~= person(j).iriscode(1,col)
        diff = diff + 1;
    end
end

fprintf("Person %i and person %i differ in %i positions", i, j, diff)
```

Rest of the code used for exercise 1

For this other part of the exercise, using the code we can see above, we've compared the first row for person 3 and 6, and we've obtained the following result:

Output obtained:

```
Person 3 and person 6 differ in 14 positions.
```

Exercise 2. The Hamming distance (HD) of two binary iris codes is the number of positions (bits) in which the two codes (binary feature vectors) differ. Compute two sets S and D of 1000 HD values each as follows:

1. For set S: Choose randomly one of the files person[i].mat, i = 1; : : : ; 20. Choose randomly two rows in that file. Compute the HD of these two rows. Normalize the HD by dividing it by 30. Repeat this process 1000 times to obtain 1000 HD values.

```
% Initialize an array S with 1000 values
S = zeros(1, 1000);
% Compute 1000 HD values using two random rows of a random person
for n = 1:1000
    pers = randi([1, 20]);
    row1 = randi([1, 20]);
    row2 = randi([1, 20]);
    S(n) = hd(person(pers).iriscode(row1, :), person(pers).
       iriscode(row2, :))/30;
end
```

Code from exercise 2.1

Using the code we can see above, we choose a random person and compare two random rows from its iris code. With this comparison we calculate the Hamming distance between both rows, and we normalize it dividing by 30. We repeat this operation 1000 times.

2. For set D: Choose randomly two different files person[i].mat and person[j].mat, i = 1; : : : ; 20; j = 1; : : : ; 20; i 6= j. Choose randomly one row from each of these two files. Compute the HD of these two rows. Normalize the HD by dividing it by 30. Repeat this process 1000 times to obtain 1000 HD values.

```
% Initialize an array D with 1000 values
D = zeros(1, 1000);
% Compute 1000 HD values using two random rows of two random
   different
% people
for n = 1:1000
    pers1 = randi([1, 20]);
    pers2 = randi([1, 20]);
    while pers1 == pers2
        pers2 = randi([1, 20]);
    end
    row1 = randi([1, 20]);
    row2 = randi([1, 20]);
    D(n) = hd(person(pers1).iriscode(row1, :), person(pers2).
       iriscode(row2,:))/30;
end
```

Code from exercise 2.2

Using the code we can see above we randomly select to person files and a random row number. Then we compare the row, with the row number we've obtained, from each file and calculate the Hamming distance between both rows. In order to normalize the HD, we divide the result we've obtained by 30. We repeat this process 100 times to obtain 1000 values.

Exercise 3. Plot the histograms of S and D in one figure with different colors. How much do the two histograms overlap?

```
    SHist = histogram(S, 30, 'BinLimits',[0, 1]);
    hold on
    DHist = histogram(D, 30, 'BinLimits',[0, 1]);
    title('S and D sets histograms');
    legend('S', 'D');
    xlabel('Normalized HD');
    xlim([0 1])
    hold off
```

Code from exercise 3

Using the code above, we plot the histograms of sets S and D using the values we've obtained from exercises 2.1 and 2.2.

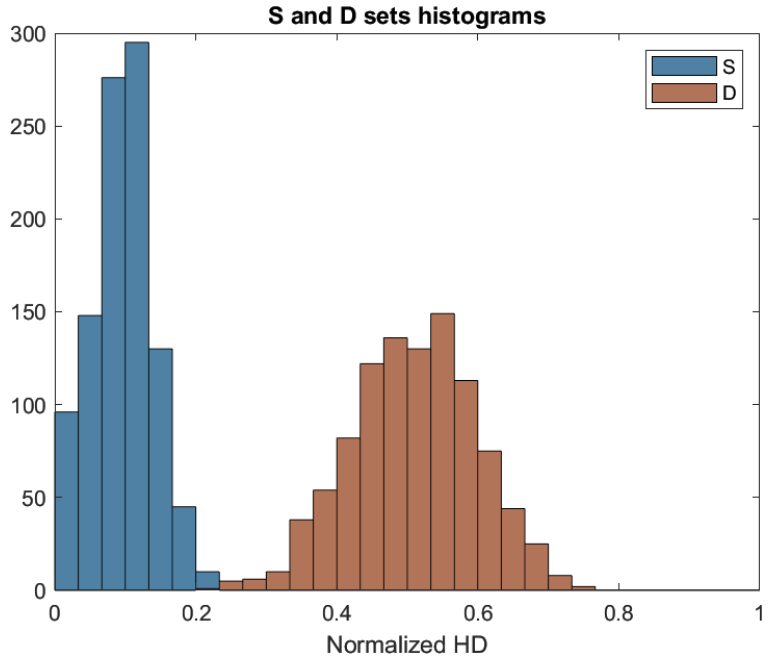

Figure 6: Sets S and D histograms

In order to see how much the two histograms overlap, we've amplified the graph and obtained the following:

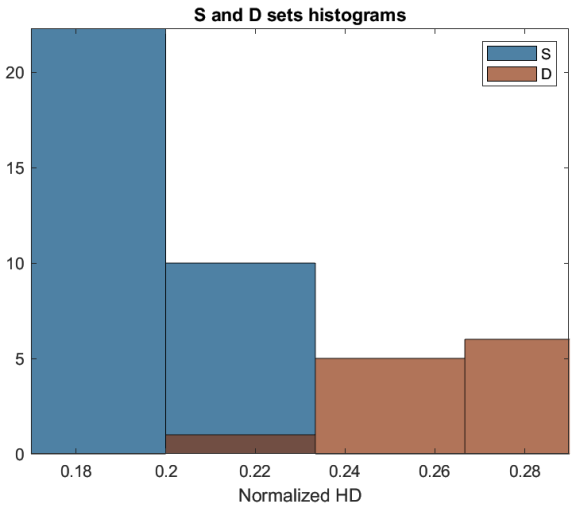

Figure 7: Sets S and D histograms amplified where they overlap

Looking at the previous graph we can conclude the two histograms overlap around the values 0.2 and 0.24 approximately.

Exercise 4. Compute the means and the variances of the sets S and D. Add to the histograms of the previous question (4.3), plots of two normal distributions (Gaussian functions) with these means and variances. How well do the normal distributions fit the histograms?

In order to compute the means and variances of the sets S and D we've generated the following code:

```
% Firstly, compute the mean and variance
SMean = mean(S)
SVar = var(S)

DMean = mean(D)
DVar = var(D)
```

Code used to calculate the means and variances of the sets S and D.

By doing this, we obtain the following results:

Means and Variances the sets S and D

```
SMean = 0.0797
SVar = 0.0020

DMean = 0.4927
DVar = 0.0080
```

Using the following code, we've generated two normal distributions. One normal distribution has S's mean and variance, and the other has D's mean and variance.

```
    % Find the range of the histogram by finding the first and last
    % values non-null values
    SFirstVal = find(SHist.Values, 1, 'first');
    % The [0, 1] interval is divided in 30 spaces, so the histogram has 31
    % edges. That means that the last non-null value's edge has 1 index
    more
    % than the number of the column to which it is edge
    SLastVal = find(SHist.Values, 1, 'last') + 1;
    % Generate linearly spaced vector between both first and last non-null
    % values
    SRange = linspace(SHist.BinEdges(SFirstVal), SHist.BinEdges(SLastVal))
    ;
    % Finally, generate a normal distribution which fits in this range
    SNorm = normpdf(SRange, SMean, sqrt(SVar));
    % Also, amplify the graph so that it has the same height as the
    histogram
    SNorm = SNorm/max(SNorm)*max(SHist.Values);

    % Do the same with the D set
    DMean = mean(D)
    DVar = var(D)
    DFirstVal = find(DHist.Values, 1, 'first');
    DLastVal = find(DHist.Values, 1, 'last') + 1;
    DRange = linspace(DHist.BinEdges(DFirstVal), DHist.BinEdges(DLastVal))
    ;
    DNorm = normpdf(DRange, DMean, sqrt(DVar));
    DNorm = DNorm/max(DNorm)*max(DHist.Values);

    hold on
    plot(SRange, SNorm, 'r')
    plot(DRange, DNorm, 'g')
    legend('S', 'D', 'SNorm', 'DNorm')
    hold off

```

Code used to obtain normal distributions for both S and D sets

If we plot these normal distributions into the histograms from figure 6, we obtain the following plot:

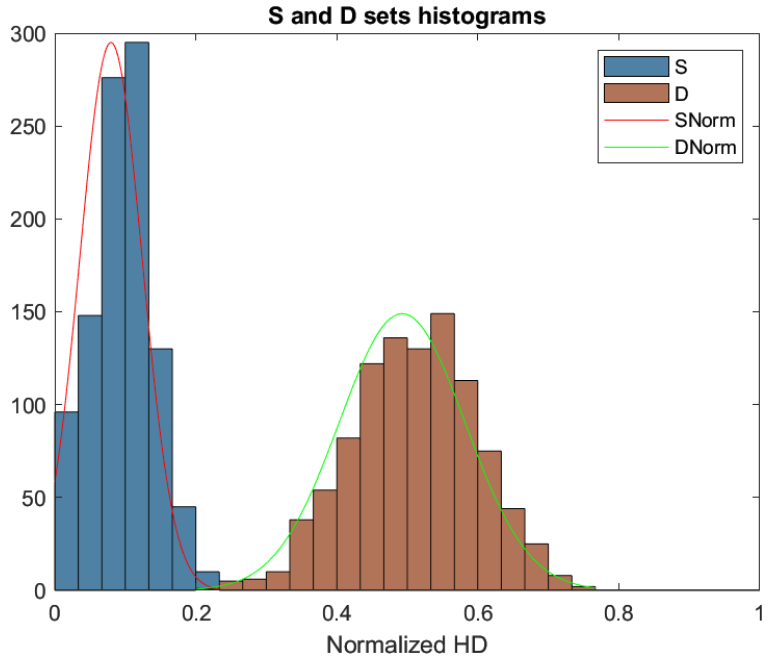

Figure 8: Sets S and D histograms with two normal distributions, both with the same means and variances as S and D

As we can see by looking at the plot, both normal distributions fit the histograms pretty well. This means that both sets follow a normal distribution.

Exercise 5. (BONUS) The distribution associated with the set S is the class-conditional probability density function that we measure a certain HD value for two iris codes of the same person. The distribution associated with the set D is the class-conditional probability density function that we measure a certain HD value for two iris codes of two different persons. Estimate the value of the decision criterion for which the false acceptance error is approximately 0.0005. False acceptance occurs when the iris codes of two different persons are declared to be sufficiently similar so that one can assume that they come from the same person. For that value of the decision criterion, determine the false rejection rate. False rejection occurs when two iris codes of the same person have a HD which is above the decision criterion so that they will wrongly be assumed to come from two different persons. (Note that here the terms acceptance (of an impostor) and rejection (of an authentic person) are related to the alternative hypothesis stating that two iris codes which are compared come form the same person, the zero hypothesis being that they come from two different persons. False acceptance and false rejection thus correspond to an error type I and II, respectively, in terms of statistical decision theory and hypothesis testing.)

```
% Compute both cumulative distribution functions of the S and D
    sets
DNormCDF = normcdf(DRange, DMean, sqrt(DVar));
SNormCDF = normcdf(SRange, SMean, sqrt(SVar));
% To get the false acceptance error below 0.0005, we iterate the
    cdf
```

```
        % function of the D set until we get as close as possible to the
            value
        Ddc = 0;
        while Ddc < numel(DNormCDF)-1 && DNormCDF(Ddc+1) < 0.0005
            Ddc = Ddc + 1;
        end
        % To find the false rejection rate, we must firstly find the index
            where
        % the decision criterion is, so we iterate until we find it
        Sdc = 0;
        while Sdc < numel(SRange)-1 && SRange(Sdc+1) < DRange(Ddc)
            Sdc = Sdc + 1;
        end

        fprintf('The decision criterion is set at %.4f, considering HD
            normalized.', DRange(Ddc));
        fprintf('In absolute numbers that is %.4f.', DRange(Ddc)*30)
        fprintf(['Furthermore, the false acceptance rate for this value of
            the decision ' ...
            'criterion is %.6f, and the false rejection rate is %.6f.'],
               DNormCDF(Ddc), 1-SNormCDF(Sdc));

        plot(SRange, SNormCDF, 'b')
        hold on
        plot(DRange, DNormCDF, 'r')
        line([DRange(Ddc) DRange(Ddc)], ylim, 'LineWidth', 2, 'Color', 'y'
            )
        title('CDF of S and D sets')
        xlabel('Normalized HD')
        ylabel('Probability(Z < X)')
        legend('CDF of S', 'CDF of D', 'Decision Criteria')
        xlim([0 1])
        hold off
```

Code from exercise 5

Output obtained:

```
The decision criterion is set at 0.1973, considering HD normalized.
In absolute numbers that is 5.9192.
Furthermore, the false acceptance rate for this value of the decision
criterion is 0.000426, and the false rejection rate is 0.002614.
```

As explained in the code, we basically iterate through the CDF function until we get the highest value lower than the false acceptance error chosen (aproximately 0.0005), the false acceptance rate we found is 0.000426.
Once we get that value, we see that our decision criterion is set at 0.1973 considering the HD normalized. However, we know that our misclassifications can go up to 30, therefore the actual decision criterion is at 5.9192. This means that two iris codes are interpreted as being from the same person when the HD is lower than this value, and interpreted as an impostor otherwise.

The rate to have a false acceptance, when two iris codes of two different people are misinterpreted as being from the same person, is 0.000426. Whereas the rate to have a false rejection, when two iriscodes from a person are misinterpreted as being from two different people, is 0.002614.
Moreover, we can see the decision criteria chosen in the following plot (obtained from the code), marked as a yellow vertical line between the two Cumulative Density Functions (CDF) of both S and D sets.

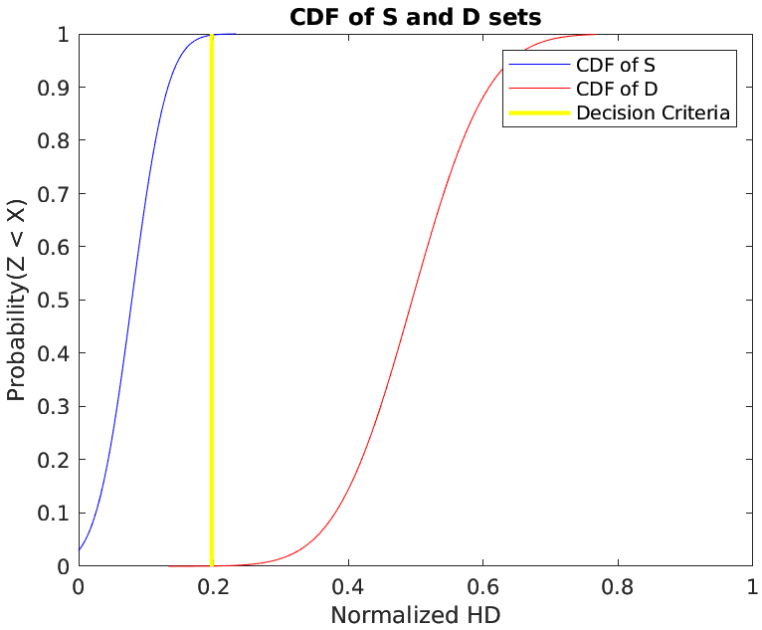

Figure 9: Sets S and D CDF

4 Work division

During this practical assignment we've worked together on all assignments, thinking them through and agreeing on the answers. While writing up the report one of us did the part corresponding to assignment 1, while the other did assignment 2. The part of assignment 3 was equally divided among us.