Introduction to Intelligent Systems - Lab 2

Tom Apol (s2701650)
Nielis Brouwer (s3850706)
Group: 23

September 24, 2018

Contents

1 Assignment 1

1.1

Plot histograms of both sets in one figure.
Please refer to figure 1 and lines 7-23 of the code at Appendix 4.1.

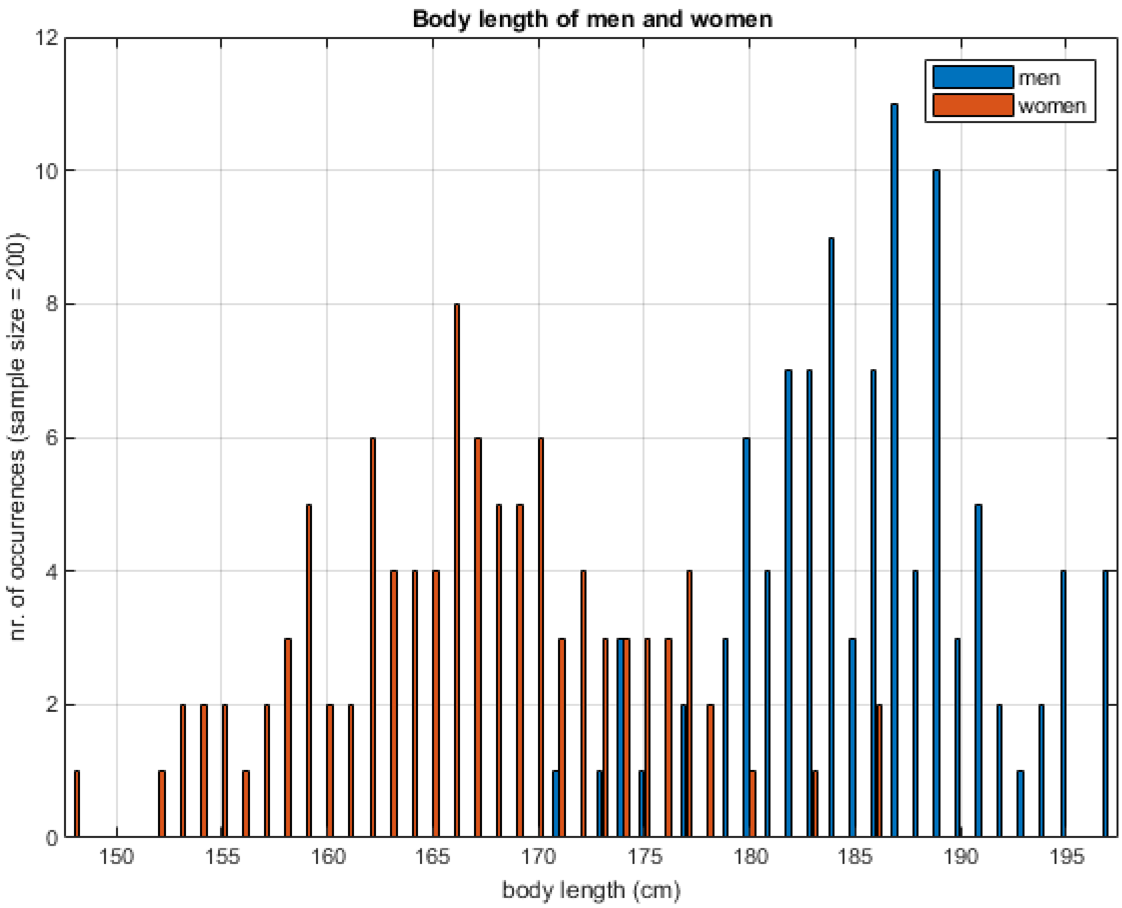

Figure 1: Histogram of the body length of 100 men and 100 women.

1.2

How many men are classified incorrect? And how many women?
0 men were classified incorrectly, while 35 women were classified incorrectly. Note that we define this measure as $women < dc \leq men$, as our histograms show women to be generally shorter than men.
See lines 27-36 of the code at Appendix 4.1.

1.3

What decision criterion should be used to minimize total number of misclassifications?
The decision criterion with the minimal amount of misclassification is 179 with a total number of 12 misclassifications. We compute this by iterating over the range of possible decision criterion values and remembering the set of best decision criteria.
See lines 40-62 of the code at Appendix 4.1.

2 Assignment 2

2.1

Plot the length versus the hair length.
See figure 2 and the code at lines 7-11 at Appendix 4.2.

Figure 2: Scatter plot of body length versus hair length of 100 men and 100 women. Note that the red dot is the median of the data set, such that we can make a more educated guess about the decision boundary in part 2.

2.2

Where would you draw the decision boundary and why?
Seeing as there seem to be two clusters of points (one to the upper left and one to the lower right of the plot), we assume the decision boundary would divide these two clusters. To this end, we computed the median and drew a straight line.
See figure 3 and the code at lines 15-22 at Appendix 4.2.

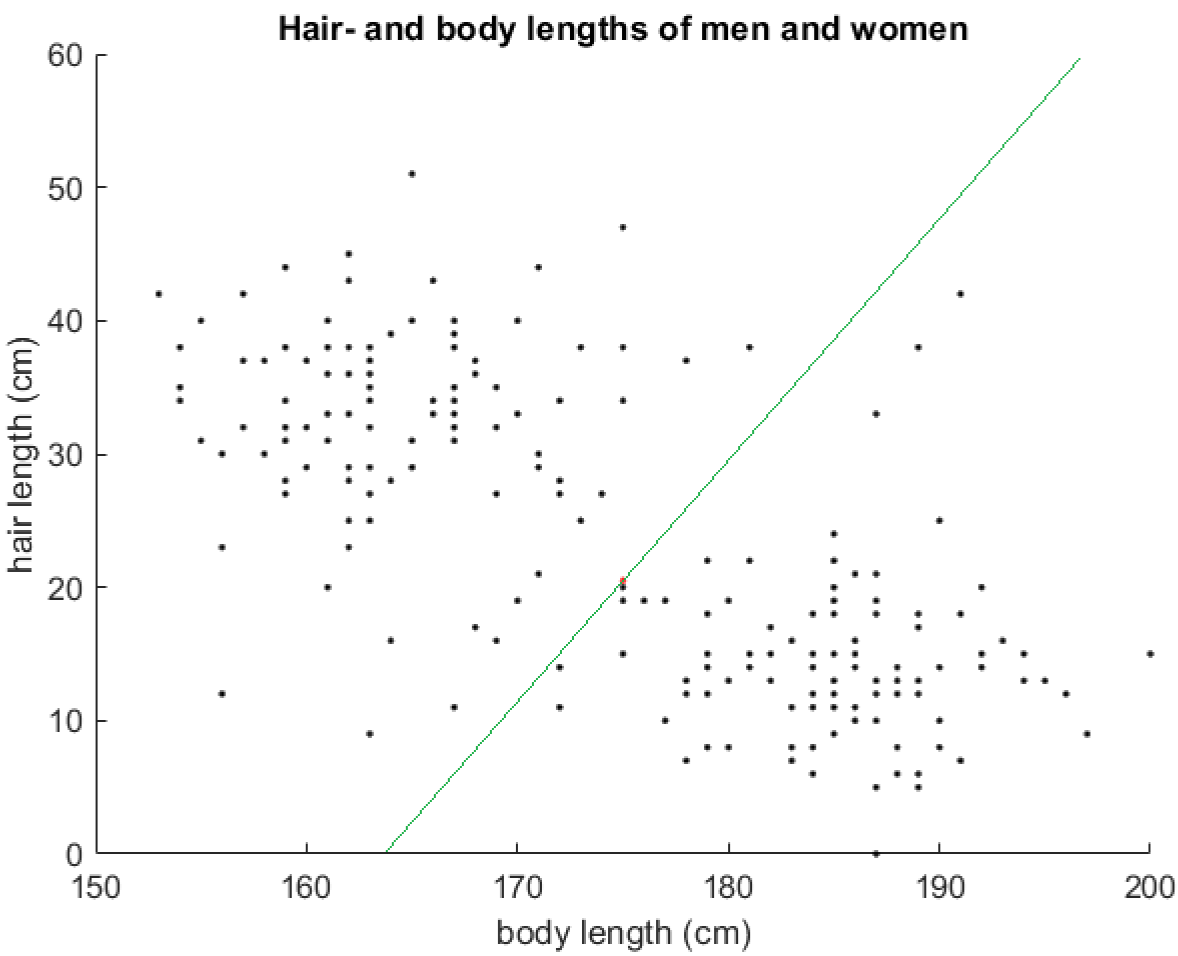

Figure 3: The same plot as in figure 2, but now with a green decision boundary.

3 Assignment 3

3.1

Compare the rows of one file, notice that they can only differ in a few positions. Compare two rows that come from two different files, notice that two such iris codes differ in about 15 positions
When computing the Hamming distance between two random rows of a random person, the resulting value was always around 2 or 3, which is indeed quite low. This is to be expected, as both rows originate from the same person after all.
When computing the Hamming between the first row of person 01 and 02, the resulting value was 16, which is indeed quite close to 15. Values in this range are to be expected, as the patterns are not the same, nor are they completely each other's inverse. See the code at lines 2-17 at Appendix 4.3.1.

3.2

Compute set S and set D
See the code at lines 21-60 at Appendix 4.3.1.

3.3

Plot the histograms of S and D in one figure with different colours. How much do the two histograms overlap?
Due to the inherent randomness of the program, and the relatively small sample size of 1000 for both sets, the overlap is rather small. Both histograms overlap barely, sometimes not at all, or in the case below only two bars, at the outskirts of the bell curve. For succinctness' sake, only the plot of part 4 is shown, since it shows both the histograms and their respective normal distribution plots. The figure below shows the overlap in more detail.
See the code at lines 70-89 at Appendix 4.3.1.

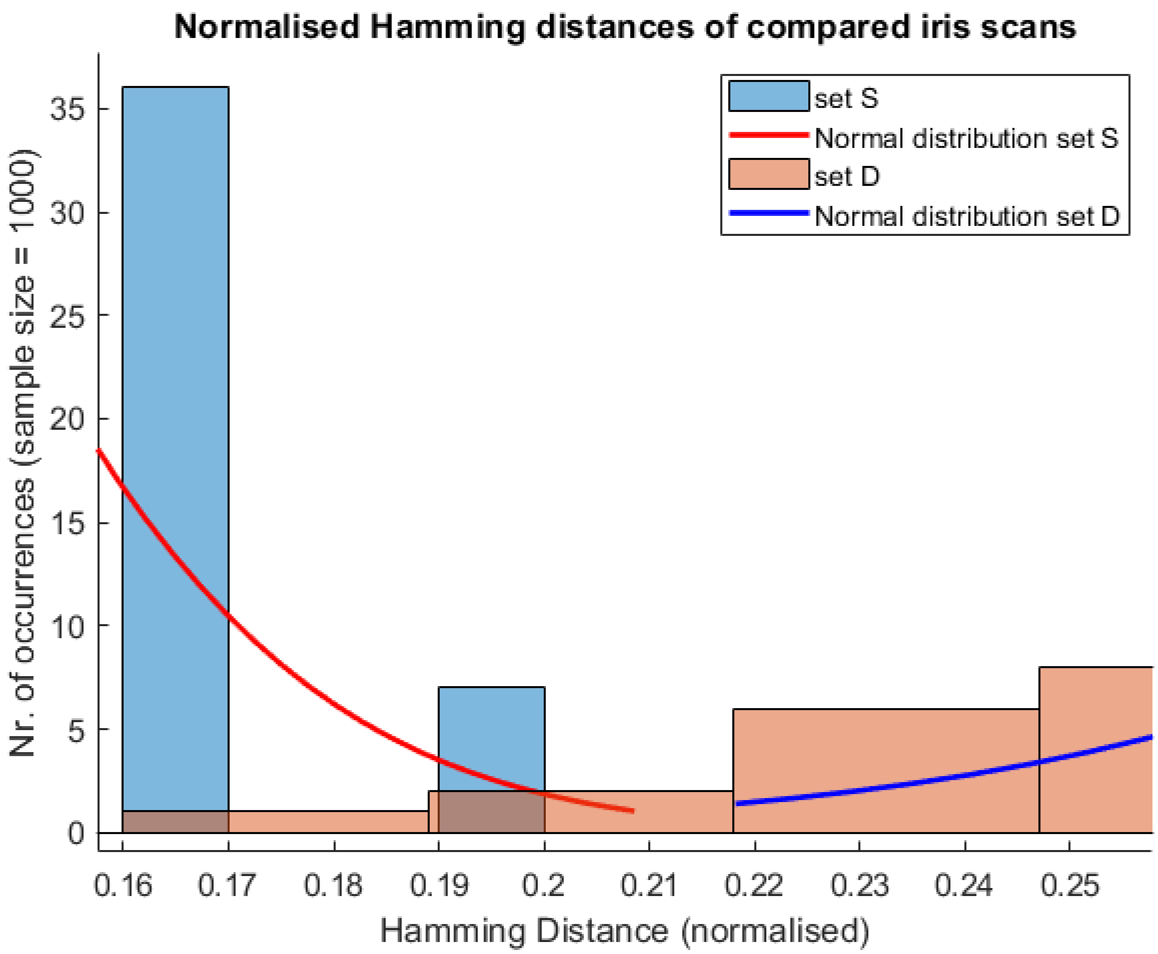

Figure 4: A close-up of the overlap between the two histograms. For the full plot, see Figure 5.

3.4

Compute the means and the variances of the sets S and D, how well do the normal distributions fit the histograms?
We used the mean and variances of sets S and D implicitly by using the Matlab function: *histfit()*, in order to conveniently plot both the histograms and their respective normal distributions. While set D's normal distribution fits well, the same can not be said of the normal distribution of set S. This is because of the sparseness of set S's histogram, where the gaps are taken into account as well, thus drastically lowering the amplitude of the inferred distribution.
See the code at lines 65-89 at Appendix 4.3.1.

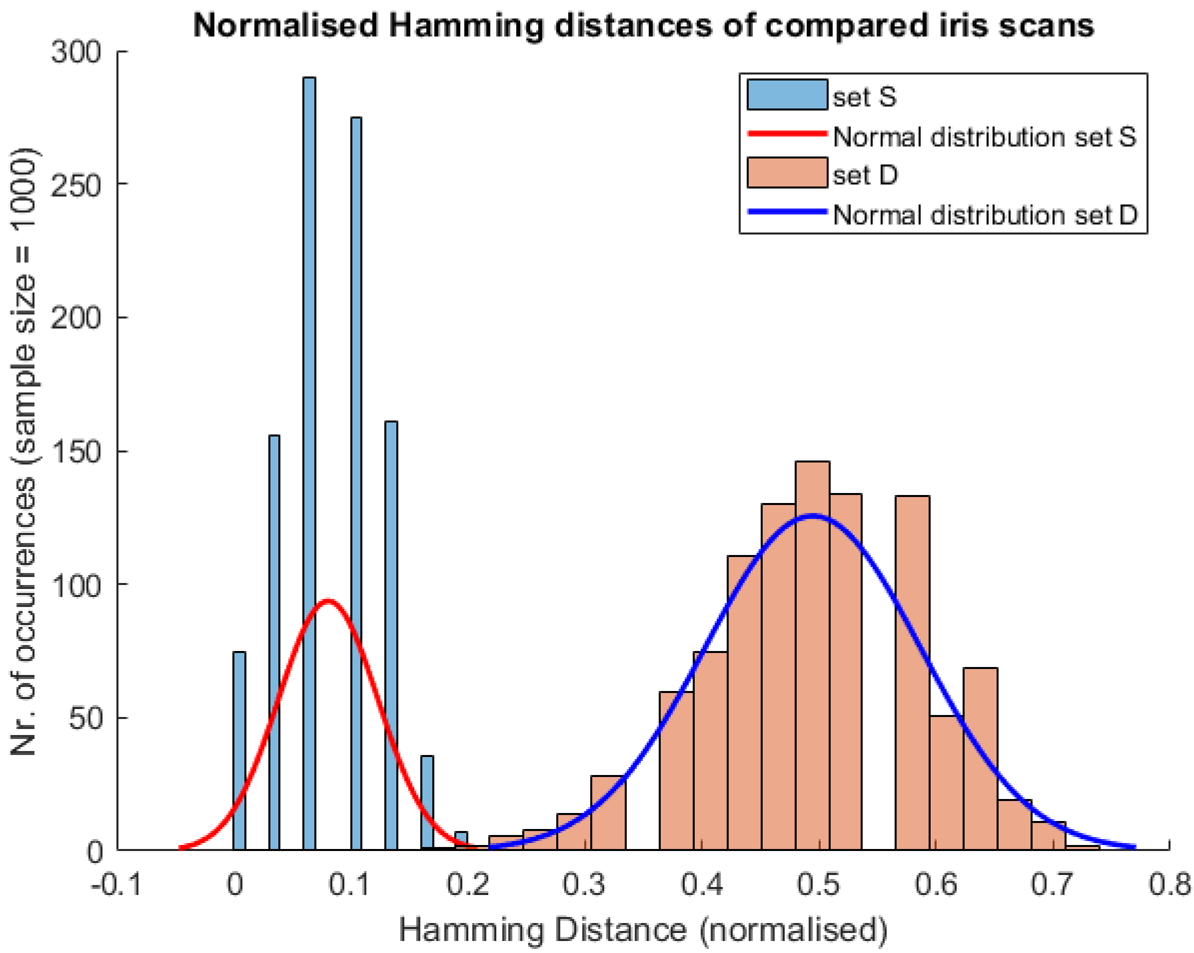

Figure 5: The histograms of set S and set D, together with their respective normal distributions.

4 Appendix - Code

The code expects to be residing in a folder on the same level as the provided "*lab_week2_data*" folder, and loads the corresponding data from said folder.

4.1 Assignment 1

```
close all;

load('../lab_week2_data/lab1_1.mat');

%==========================================================================%

% set edges for histogram
edges = min(min(length_men),min(length_women)):1:max(max(length_men), ...
            max(length_women));

% set data for histogram
h1 = histcounts(length_men,edges);
h2 = histcounts(length_women,edges);

% plot histograms
figure;
bar(edges(1:end-1),[h1; h2]');
grid on;

xlabel("body length (cm)");
ylabel("nr. of occurrences (sample size = 200)");
title("Body length of men and women");
legend("men", "women");

%==========================================================================%

% find misclassifications for men and women for dc=170,
%  where the difference between women and men is defined as follows:
%  women < dc <= men
%  dc : decision criterion
%  mc : misclassifications

dc = 170;
% output values
mc_men_170 = sum(h1(1:dc - edges(1)))
mc_women_170 = sum(h2(dc - edges(1)+1:end))

%==========================================================================%

```

```
% find optimal decision criterion
optimal_dc = edges(1); % initialise with the first value
optimal_mc = intmax; % represents infinity

for i = 1:length(edges)
    dc = edges(i);
    mc_men = sum(h1(1:dc - edges(1)));
    mc_women = sum(h2(dc - edges(1)+1:end));
    mc_sum = mc_men + mc_women;

    if mc_sum < optimal_mc(1)
        optimal_mc = mc_sum;
        optimal_dc = dc;

    % If more optimal dc's exist, add them to our list/vector.
    elseif mc_sum == optimal_mc(1)
        optimal_dc = cat(2, optimal_dc, dc);
    end
end

% print optimal decision criterion
optimal_dc
optimal_mc
```

4.2 Assignment 2

```
close all;

load('../lab_week2_data/lab1_2.mat');

%==========================================================================%

% plot data
figure; hold on;
person_lengths = measurements(:,1);
hair_lengths = measurements(:,2);
plot(person_lengths, hair_lengths, 'k.')

%==========================================================================%

% plot the median point of all data, in order to make an educated guess
%  of the decision boundary
m_l = median(person_lengths);
m_h = median(hair_lengths);
plot(m_l, m_h, 'r.');
xlabel("body length (cm)");
ylabel("hair length (cm)");
title("Hair- and body lengths of men and women");
```

4.3 Assignment 3

4.3.1 Main script

```

% Compute Hamming distance between iris scans of the same person.
rand_file_num = randi(20);
random_row_nr1 = randi(20);
random_row_nr2 = randi(20);

data1 = load_iris_scan_data(rand_file_num);

hamming_distance_same = sum(bitxor(data1(random_row_nr1,:), ...
    data1(random_row_nr2,:)))

% Compute Hamming distance between iris scans of differing persons.
person_number = 1;
data1 = load_iris_scan_data(person_number);
person_number = 2;
data2 = load_iris_scan_data(person_number);
hamming_distance_different = sum(bitxor(data1(1,:), data2(1,:)))

%==========================================================================%

set_S = zeros(1,1000);
set_D = zeros(1,1000);

% Compute set S
for i = 1:1000
    rand_file_num = randi(20);
    loaded_person_data = load_iris_scan_data(rand_file_num);
    random_row_nr1 = randi(20);
    random_row_nr2 = randi(20);

    row1 = loaded_person_data(random_row_nr1,:);
    row2 = loaded_person_data(random_row_nr2,:);
    norm_hd = sum(bitxor(row1, row2))/30;
    set_S(i) = norm_hd;
end


% Compute set D
for i = 1:1000
    rand_file_num1 = randi(20);
    rand_file_num2 = randi(20);
    random_row_nr1 = randi(20);
```

```
    random_row_nr2 = randi(20);

    % rand perm
    while rand_file_num1 == rand_file_num2
        rand_file_num2 = randi(20);
    end

    % select person with random number
    loaded_person_data1 = load_iris_scan_data(rand_file_num1);
    loaded_person_data2 = load_iris_scan_data(rand_file_num2);

    % read random row of random person
    row1 = loaded_person_data1(random_row_nr1, :);
    row2 = loaded_person_data2(random_row_nr2, :);
    norm_hd = sum(bitxor(row1, row2))/30;

    set_D(i) = norm_hd;
end

%==================================================================%

%calculate mean and standard deviation
mean_S = mean(set_S);
mean_D = mean(set_D);
std_S = std(set_S);
std_D = std(set_D);

% generate figure and histograms.
% histfit uses and calculates mean and std. deviation of the sets itself.
close all;
figure; hold on;

% obtain bin-counts (= length of edges vector) for smoother histogram
%  and normal distribution plot creation.
[hc_s, edges_s] = histcounts(set_S);
[hc_d, edges_d] = histcounts(set_D);

h_s = histfit(set_S, length(edges_s)-1);
h_d = histfit(set_D, length(edges_d)-1);
h_s(2).Color = [1 0 0]; % set normal dist. colour to red
h_d(2).Color = [0 0 1]; % set normal dist. colour to blue
alpha(.5);
xlabel("Hamming Distance (normalised)");
ylabel("Nr. of occurrences (sample size = 1000)");
title("Normalised Hamming distances of compared iris scans");
legend("set S", "Normal distribution set S", ...
```

```
    "set D", "Normal distribution set D");
```

4.3.2 load_iris_scan_data function

```
function person_data = load_iris_scan_data(person_nr)
%Loads iris scan data from disk. Does not contain a check to stop invalid
%  input types or invalid input numbers.

folder_name = '../lab_week2_data/';
data_name1 = 'person';
data_name2 = '.mat';

% convert person_nr to int
if isnumeric(person_nr)
    person_nr = int64(person_nr);
elseif ischar(person_nr)
    person_nr = int64(str2double(['int32(' person_nr ')']));
end

% convert person_nr back to char, plus add padding
str_person_nr = int2str(person_nr);
if person_nr < 10
    str_person_nr = strcat('0', str_person_nr);
end


person_data = load(strcat(strcat(strcat(folder_name, data_name1),...
    str_person_nr), data_name2));
person_data = person_data.iriscode;

end
```