

# Introduction to Intelligent Systems: Assignment 2

## Introduction to PR and Iris recognition

Damiano Melotti (S3838706)  
Amadeus Beckmann (S3839508)

September 24, 2018

### 1 Assignment 1

#### 1.1

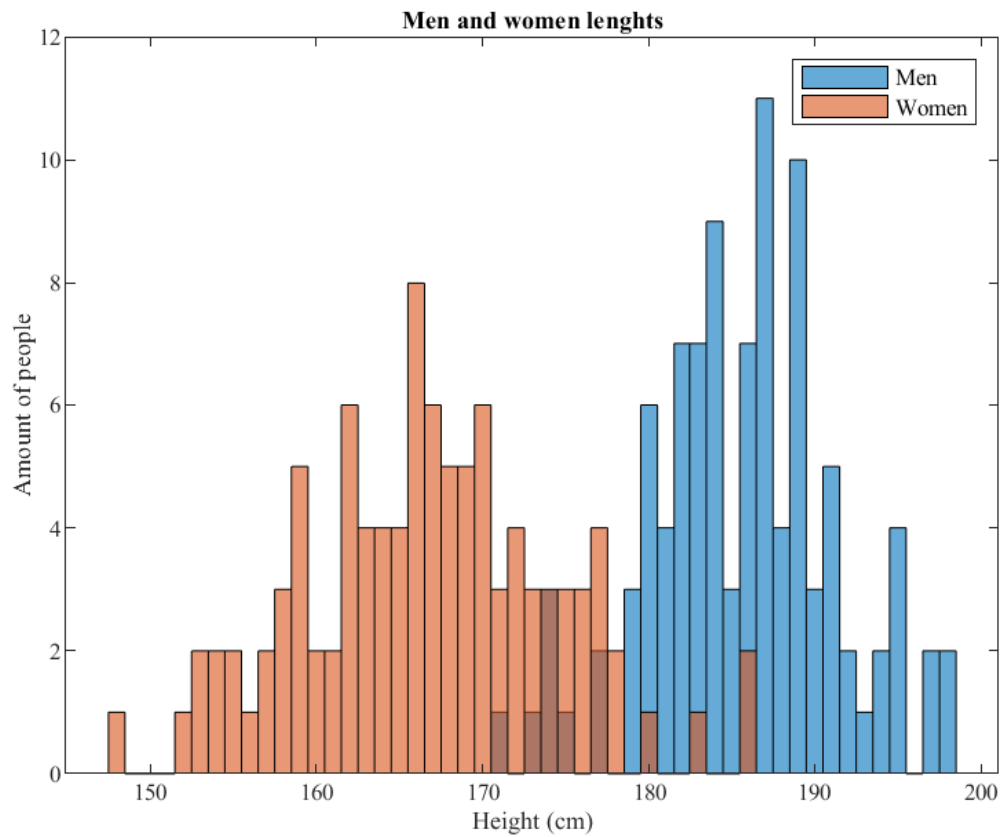


Figure 1: Histograms of both sets of data

#### 1.2

Choosing the decision criterion at 170 cm and considering that persons who are exactly on the threshold are women, 0 men and 29 women are classified incorrect.

### 1.3

For this data, the best decision criterion (to minimize the sum of incorrect classifications) is at 178 cm. Persons who are exactly on the threshold are considered women.

### 1.4 Code

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

% Plot histograms
histogram(length_men);
hold on;
histogram(length_women);

% Add title , axis labels and legend
title('Men_and_women_lengths');
xlabel('Height_(cm)');
ylabel('Amount_of_people');
set(gca, 'fontsize', 8, 'fontname', 'Times_New_Roman');
legend('Men', 'Women');

% Find the best decision criterion with the least amount of misjudgements
best = Inf;
for i = 170:180
    sum = numel(find(length_women > i)) + numel(find(length_men <= i));

    if (sum < best)
        best = sum;
        thr = i;
    end
end

fprintf('The_best_decision_criterion_is_at%d.\n',thr);
```

Listing 1: Code that shows the histograms and calculates the best decision criterion

## 2 Assignment 2

### 2.1



Figure 2: Scatter of people's heights and hair lengths

### 2.2

We chose this decision boundary considering that it's more likely that a tall person is a man, combined with the fact that women usually have longer hair. These assumptions come from common sense. With our decision criterion we divided the population in two exactly parts: however, it is very unlikely (as seen with the previous exercise) that this division is perfect, since there surely are short men with long hair and tall women with short hair.

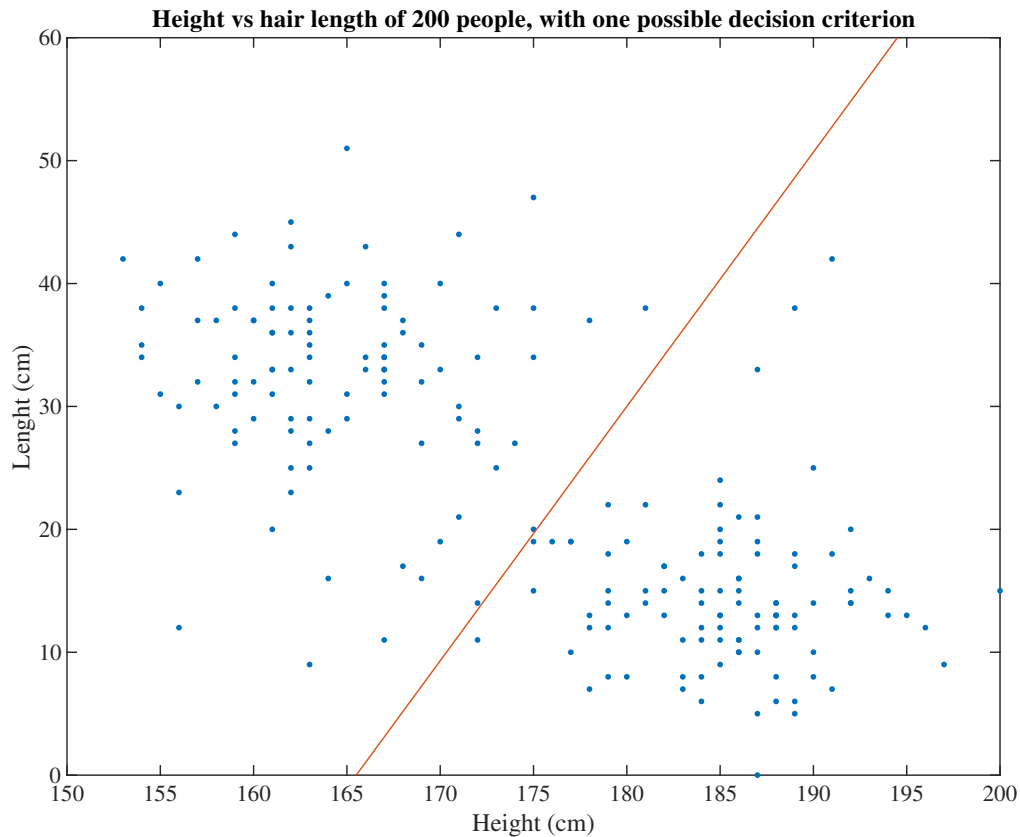


Figure 3: Same scatter with our threshold

## 2.3 Code

```
load( './lab_week2_data/lab1_2.mat' );
x = measurements(:,1);
y = measurements(:,2);

% Plot data
figure(1);
plot(x,y, '.');
title( 'Height vs hair length of 200 people' );
xlabel( 'Height (cm)' );
ylabel( 'Lenght (cm)' );
set(gca, 'fontsize', 8, 'fontname', 'Times New Roman');

% Code for the second question
figure(2);
plot(x,y, '.');
title( 'Height vs hair length of 200 people, with one possible decision criterion' );
xlabel( 'Height (cm)' );
ylabel( 'Lenght (cm)' );
set(gca, 'fontsize', 8, 'fontname', 'Times New Roman');
hold on;
plot([165.5 194.5], [0 60]);
```

```

p1 = [165.5 0];
p2 = [194.5 60];

% a and b just keep the count of points on the two sides of the line
a = 0;
b = 0;
for i=1:200
    d = (measurements(i,1)-p1(1))*(p2(2)-p1(2)) -
        (measurements(i,2)-p1(2))*(p2(1)-p1(1));
    if d > 0
        a = a+1;
    else
        b = b+1;
    end
end

fprintf('Persons considered women: %d\nPersons considered men: %d\n', a, b);

```

Listing 2: Code that loads the needed data, creates the two plots and checks the division of the points

## 3 Assignment 3

### 3.1

Analysing the rows of the same person it is clear that the iris codes are almost equal: the differences are only in a few bits. On the other hand, iris codes from different people of course present many differences.

### 3.2

```

% array of struct containing the iriscodes of the 20 persons
person = repmat(struct('iriscodes', []), 20, 1);

% loading files
for i=1:20
    filename = sprintf('lab_week2_data/person%02d.mat', i);
    person(i) = load(filename);
end

S = zeros(1, 1000);
for i=1:1000
    % random person
    random_iriscodes = person(randi(20)).iriscodes;

    % random permutation of rows to be sure that we don't pick the same row
    iriscodes_perm = randperm(20);

    % the first two rows with the iriscodes
    row1 = random_iriscodes(iriscodes_perm(1), :);
    row2 = random_iriscodes(iriscodes_perm(2), :);

    % normalized hamming distance
    S(i) = sum(xor(row1, row2) == 1) / 30;
end

```

```

D = zeros(1, 1000);
for i=1:1000
    person_perm = randperm(20);
    iriscode1 = person(person_perm(1)).iriscode(randi(20), :);
    iriscode2 = person(person_perm(2)).iriscode(randi(20), :);

    % normalized hamming distance
    D(i) = sum(xor(iriscode1, iriscode2) == 1) / 30;
end

histogram(S);
hold on;
histogram(D);
legend('S', 'D');

```

Listing 3: Code that calculates the HD and creates the plot

### 3.3

The two histograms overlap very little around 0.2. This probably happens because two people have very similar iris codes, but have still at least 5 different features. When choosing the decision criterion this possibility has to be taken into account: considering that iris identification is usually used under highly sensitive conditions, it is surely better to have a false negative rather than a false positive.

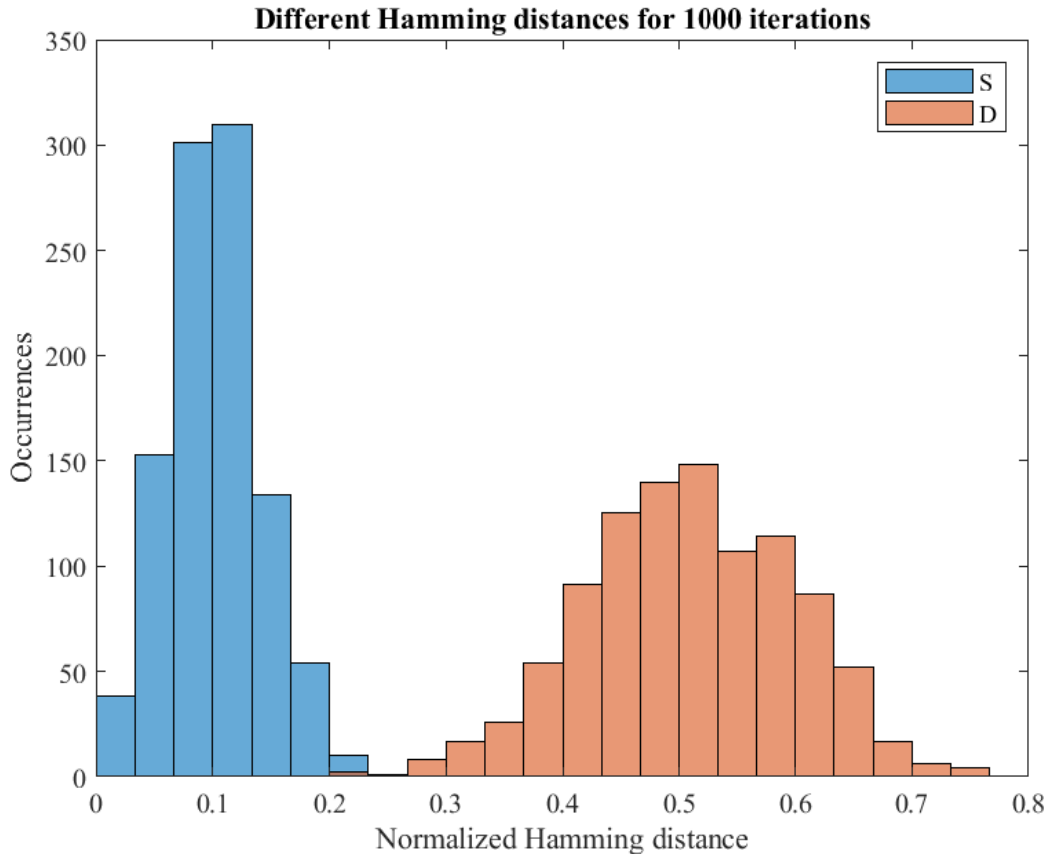


Figure 4: Normalized HD running 1000 iterations

It is important to note that we had to manually specify bin widths to create the histograms and move the data points half a bin width along the X axis, because Matlab seems to have problems with floating point numbers and does not display our normalized records correctly. So our workaround creates  $N$  containers, where  $N$  is the difference between the smallest and largest measured value in  $S$  multiplied by 30. For a vector  $S = [0, 0.03333, 0.03333, 0.06666]$  3 containers would be created with the intervals  $[0, 0.03333)$  and  $[0.03333, 0.06666)$ . Shifting the data does not invalidate it, because we are dealing with histograms and not an exact plot, and it is needed only for an unambiguous assignability into one of the containers.

Figure 4 shows our tweaked version, while figure 5 shows the histogram using the default MATLAB functions.

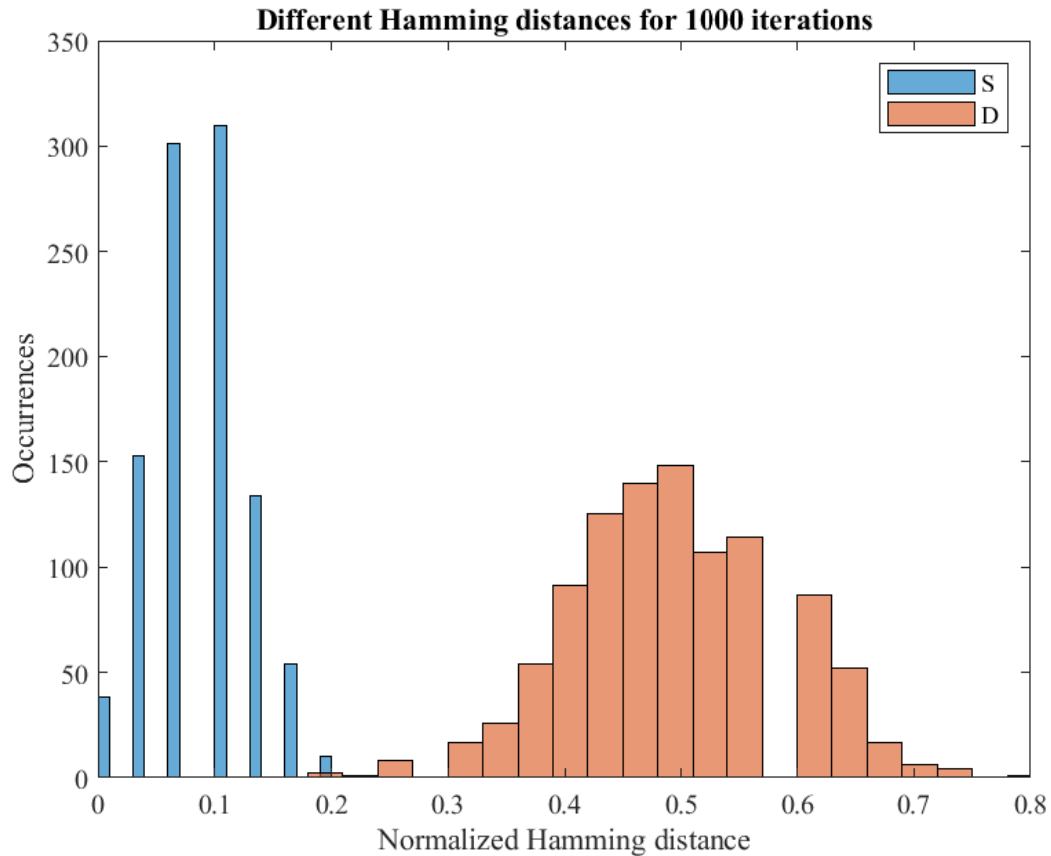


Figure 5: Normalized HD running 1000 iterations without adjusting the bins

### 3.4

The normal distributions over the histograms fit well, they are not perfectly describing them but they offer a good approximation.

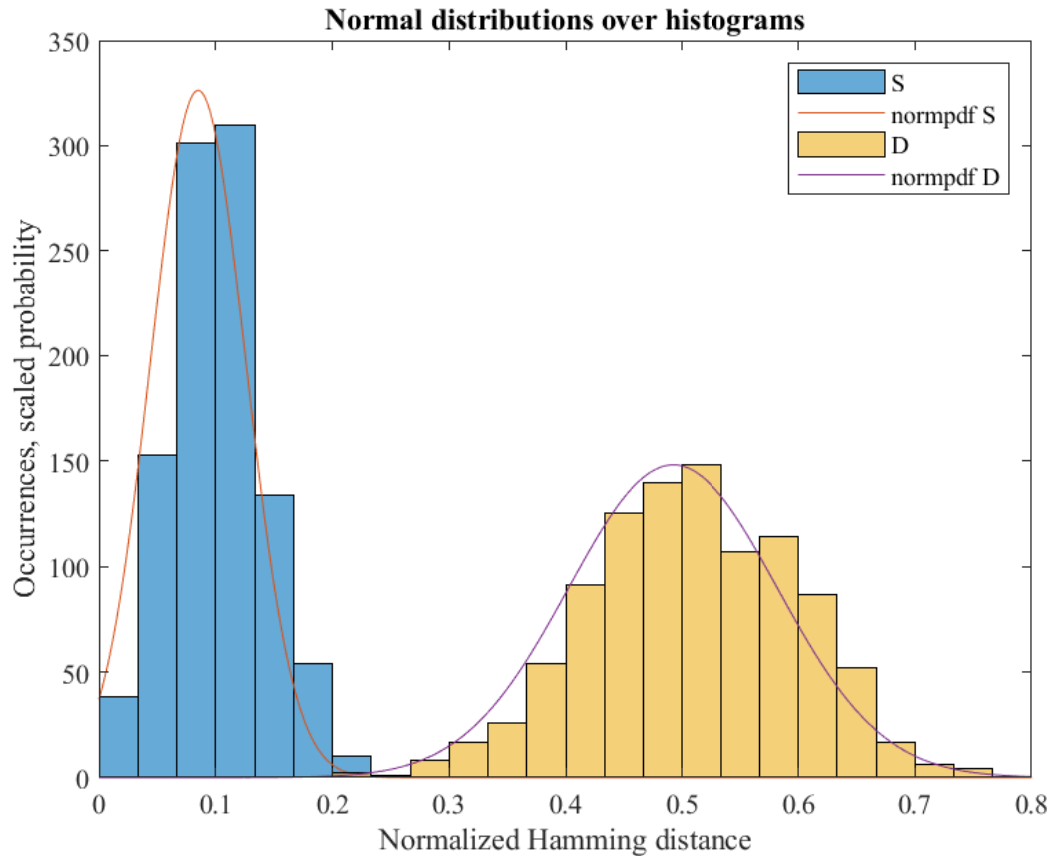


Figure 6: Histograms with normal distributions

### 3.5

### 3.6 Code

```
% array of struct containing the iriscode of the 20 persons
person = repmat(struct('iriscode', []), 20, 1);
```

```
% loading files
```

```
for i=1:20
    filename = sprintf('lab-week2_data/person%02d.mat', i);
    person(i) = load(filename);
end
```

```
S = zeros(1, 1000);
```

```
for i=1:1000
    % random person
    random_iriscode = person(randi(20)).iriscode;
```

```
% random permutation of rows to be sure that we don't pick the same row
iriscode_perm = randperm(20);
```

```
% the first two rows with the iriscode
row1 = random_iriscode(iriscode_perm(1), :);
```



```

    row2 = random_iriscode(iriscode_perm(2), :);

    % normalized hamming distance
    S(i) = sum(xor(row1, row2) == 1) / 30;
end

D = zeros(1, 1000);
for i=1:1000
    person_perm = randperm(20);
    iriscode1 = person(person_perm(1)).iriscode(randi(20), :);
    iriscode2 = person(person_perm(2)).iriscode(randi(20), :);

    % normalized hamming distance
    D(i) = sum(xor(iriscode1, iriscode2) == 1) / 30;
end

% 3.3 adjusted version
figure(1);
histogram(S + 1/60, min(S):(1/30):max(S)+1/30);
axis([0 0.8 0 350]);
hold on;
histogram(D + 1/60, min(D):(1/30):max(D)+1/30);
legend('S', 'D');
title('Different_Hamming_distances_for_1000_iterations');
xlabel('Normalized_Hamming_distance');
ylabel('Occurrences');
set(gca, 'fontsize', 9.5, 'fontname', 'Times_New_Roman');

% default version
figure(2);
histogram(S);
axis([0 0.8 0 350]);
hold on;
histogram(D);
legend('S', 'D');
title('Different_Hamming_distances_for_1000_iterations');
xlabel('Normalized_Hamming_distance');
ylabel('Occurrences');
set(gca, 'fontsize', 9.5, 'fontname', 'Times_New_Roman');

figure(3);
histogram(S + 1/60, min(S):(1/30):max(S)+1/30);
hold on;
x = [0:0.001:1];
f = normpdf(x, mean(S), std(S))*1000/30;
g = normpdf(x, mean(D), std(D))*1000/30;
axis([0 0.8 0 350]);
plot(x, f);
histogram(D + 1/60, min(D):(1/30):max(D)+1/30);
plot(x, g);
legend('S', 'normpdf_S', 'D', 'normpdf_D');
title('Normal_distributions_over_histograms');
xlabel('Normalized_Hamming_distance');
ylabel('Occurrences_scaled_probability');
set(gca, 'fontsize', 9.5, 'fontname', 'Times_New_Roman');

```