Introduction to Intelligent Systems
Lab Session 3

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

October 2, 2017

Assignment 1

The complete code for this exercise is in **Appendix A** .
We are given in the file lab3_1.mat the class conditional probabilities of salmons and sea basses. We also know that a sea bass is caught 3 times as often as a salmon, meaning that, if the probability to catch a salmon is x, then the probablity to catch a sea bass is $3 * x$. Thus the two values of the prior probabilities are 0.75 for the sea bass and 0.25 for the salmon.
To compute the posterior probabilities we need the class conditional probabilities (likelihood), the prior probabilities and the evidence, which we still have to compute. The evidence can be calculated as follows:

$$Evidence(x) = p(x|\omega_1) * p(\omega_1) + p(x|\omega_2) * p(\omega_2)$$

and it is done with the code in **Listing 1** :

Listing 1: *Code to compute the evidence*

```
evidence = prior_sb*class_conditional_sb + prior_sl*
class_conditional_sl;
```

Having all the necessary data, we can now compute the posterior probabilities with the Bayes Rule:

$$p(\omega_i|x) = \frac{p(x|\omega_i) * p(\omega_i)}{p(x)}$$

and it is done with the code in **Listing 2** :

Listing 2: *Code to compute the posterior probabilities*

```
for i = 1:241
    posterior_sb(i) = (class_conditional_sb(i) * prior_sb)/evidence
(i);
    posterior_sl(i) = (class_conditional_sl(i) * prior_sl)/evidence
(i);
end
```

Figure 1 shows the plot of the two posterior probabilities: the blue plot is the posterior probability of the sea bass, while the red one is the posterior probability of the salmon. It is clear from the plot that the sum of the two probabilities, for each length x, has to be 1, since it is a conditional probability based on the knowledge that x has a specified value.

Plot of the two posterior probabilities

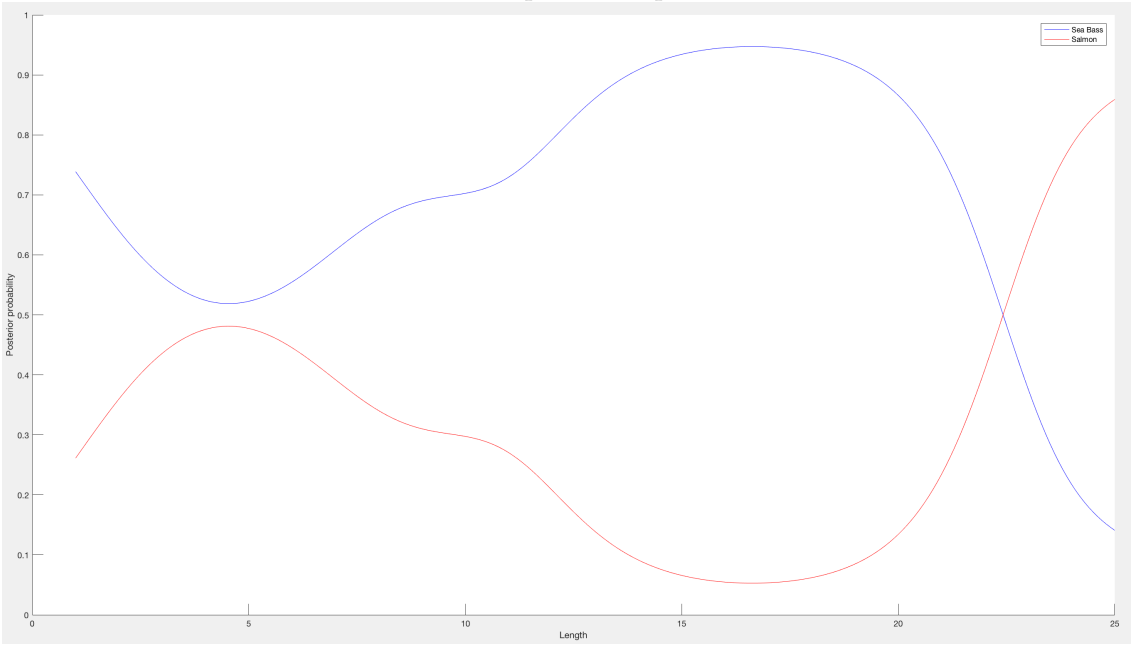

Figure 1: *shows the plot of the posterior probabilities for sea bass and salmon*

According to the Bayesian decision rule, to classify a new fish as sea bass or salmon, given its length, we should choose the kind of fish with the highest posterior probability for that length. Thus we have now all the data to classify new fishes.
Given a fish with a length of 8cm, the probability (calculated using the posterior probabilities) of it being a sea bass is 0.6597, and the probability of it being a salmon is 0.3403: thus we will classify the new fish as a sea bass. The same procedure applies to a new fish with length 20cm: the probability of it being a sea bass is 0.8658 and the probability of it being a salmon is 0.1342: thus we will classify this new fish as a sea bass as well.
Figure 2 shows a plot of the two posterior probabilites, with the addition of the two new fishes, represented by the green lines. We can observe from the plot that the probability of the second fish being a sea bass is greater than the probability of the first fish being a sea bass: this means that the error probability for the classification of the first fish is higher than the one for the second fish.

Plot of the posterior probabilities with the two new fishes

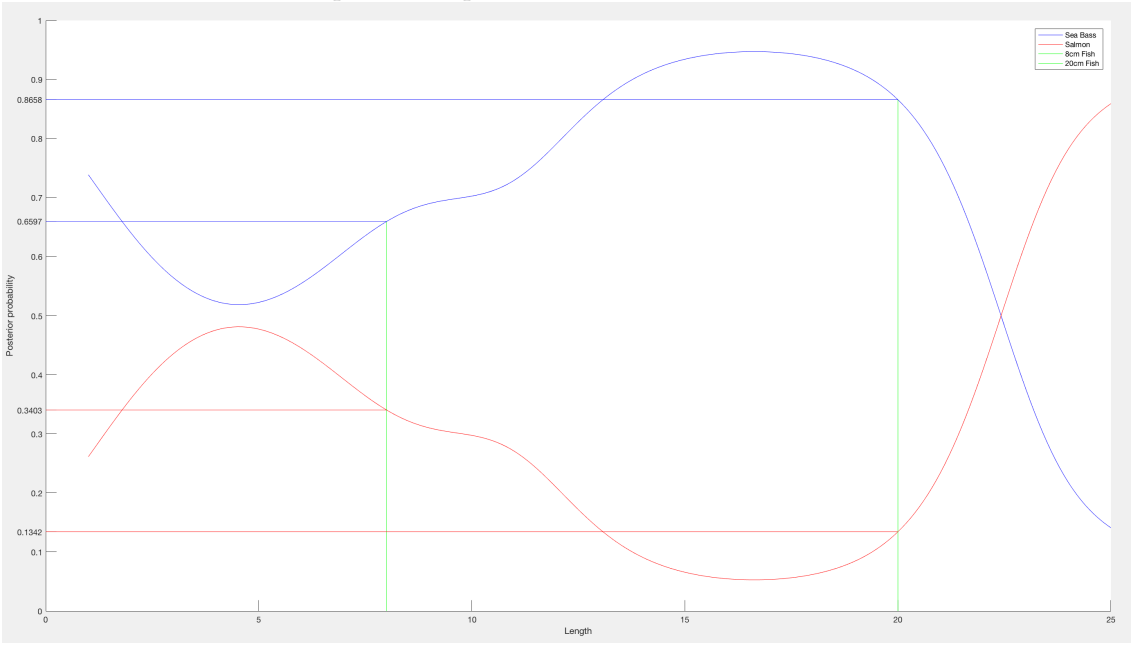

Figure 2: *shows the plot of the posterior probabilities with the green lines indicating the new fishes*

We can also calculate the probability of doing an error during the classification as folllows:

$$p(error|x) = min[p(\omega_1|x), p(\omega_2|x)]$$

and plot the results to see how likely our decision rule is to compute a wrong result. **Figure 3** shows the plot of the error probability in the classification process.

Plot of the error probability

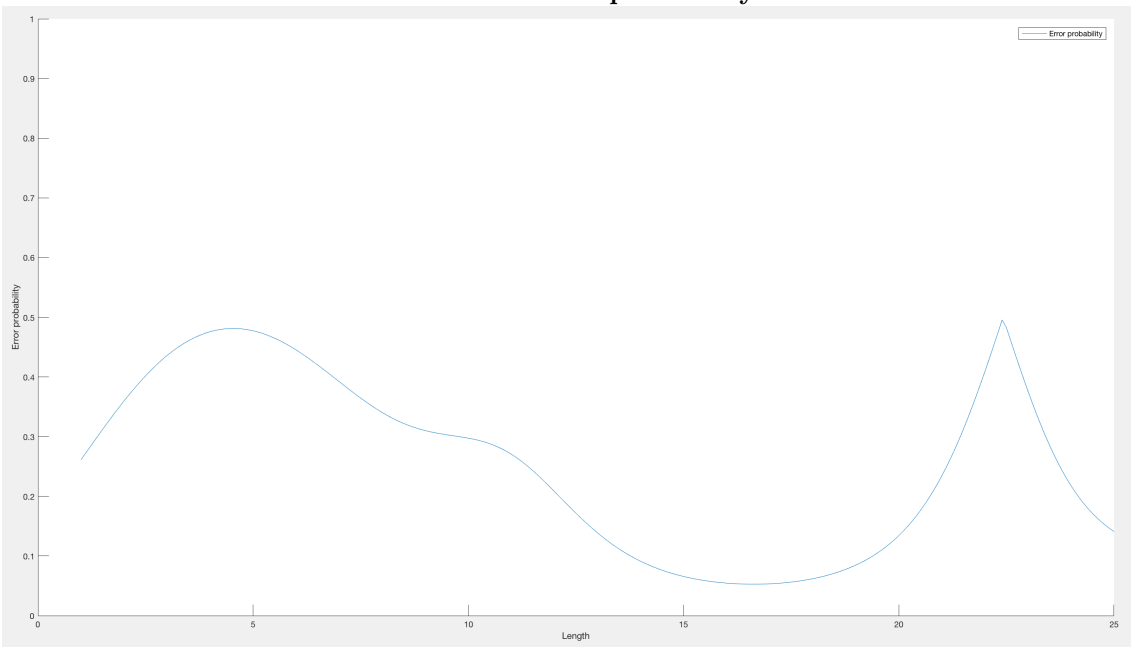

Figure 3: *shows the plot of the error probability for the Bayes classification*

Assignment 2

The full code of the exercise is in **Appendix B** . Throughout the exercise we use ω_1 referring to class 1, and ω_2 referring to class 2.

1. The data contained in S1 and S2 is a collection of 1D data, which can be plotted on the x-axis, as done in **Figure 4** : the elements of S1 are blue circles, while the elements of S2 are red circles. The elements of the test set T are also in **Figure 4** as black boxes. Observing the plot it is clear that there is not a simple separation between the two classes, as there is not a point on the x-axis that separates all the elements of S1 from all the elements of S2.

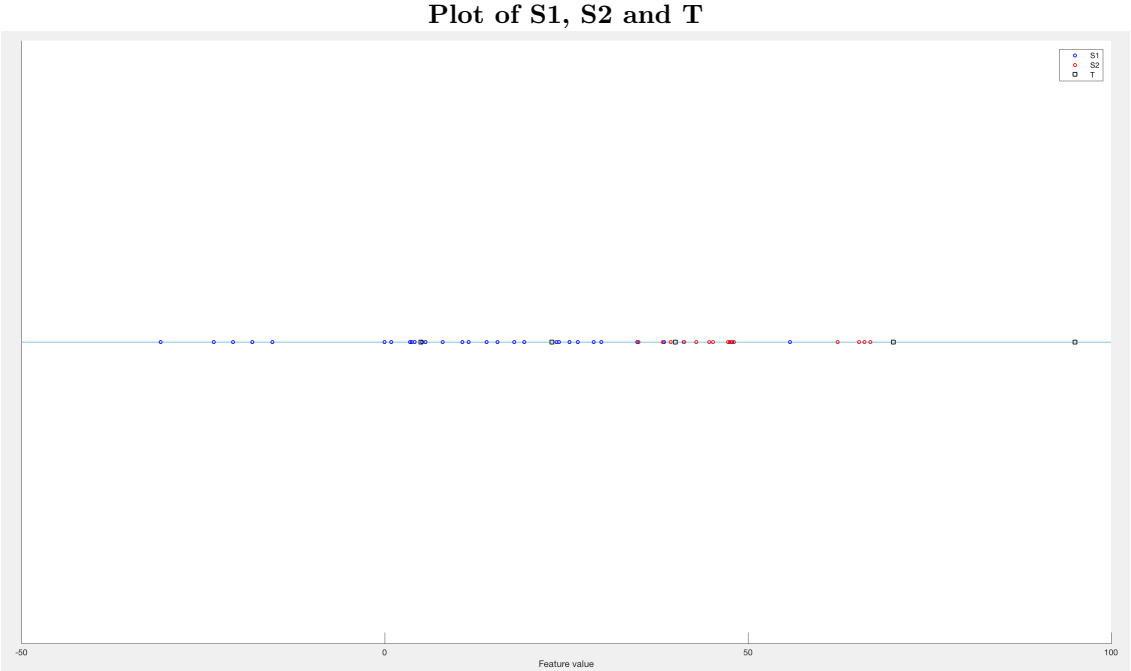

Figure 4: *shows the plot of the elements of S1, S2 and T: the first ones are blue circles, the second ones are red circles and the last ones are black boxes*

2. To analyze the underlying distribution that produced the data, we observe how symmetric the given data is. The mean of S1 is 12.9 and its median is 12.8, which are really similar values. Thus we can hypothesize that the data distribution is symmetric. To further verify this hypothesis, we also compute the first quartile, which is 3.47, and the third quartile, which is 26.58 . The median is slightly nearer to the first quartile than to the third quartile (if it was exactly in the middle, it would be 15). Considering that the distance from the middle is not significantly big, we can conclude that a symmetric distribution should fit the data of S1, and, since the occurrances of values far from the mean are rare, a normal distribution would be a good choice.
   The same analysis can be done with S2, which has a mean of 49.17, a median of 47.26, a first quartile of 41.65 and a third quartile of 58.78. Although this time the difference between the mean and the median is bigger, it is still not big enough to consider a non-symmetric distribution. Furthermore, the median is almost in the middle of the computed quartiles, and thus we also choose a normal distribution for S2. The code used to compute these values is in **Listing 3** (in the complete script this part has been commented, as it provides only preliminar analysis and is not really useful to answer the next questions).

Listing 3: *Code to compute the values which determined the distribution to use*

```
med_S1 = median(S1);
mean_S1 = mean(S1);
Q1_S1 = quantile(S1, 0.25);
Q3_S1 = quantile(S1, 0.75);

med_S2 = median(S2);
mean_S2 = mean(S2);
Q1_S2 = quantile(S2, 0.25);
Q3_S2 = quantile(S2, 0.75);

fprintf("Mean_S1: %f\tMedian_S1: %f\nQ1_S1: %f\tQ3_S1: %f\
tExpectedMedian_S1: %f\n", mean_S1, med_S1, Q1_S1, Q3_S1, (
Q3_S1 + Q1_S1)/2);
fprintf("Mean_S2: %f\tMedian_S2: %f\nQ1_S2: %f\tQ3_S2: %f\
tExpectedMedian_S2: %f\n", mean_S2, med_S2, Q1_S2, Q3_S2, (
Q3_S2 + Q1_S2)/2);
```

Figure 5 shows the plots of the two gaussian functions, which are the class conditional probability densities for S1(in blue) and for S2(in red).

Plot of S1, S2 and the Gaussian distributions that generated them

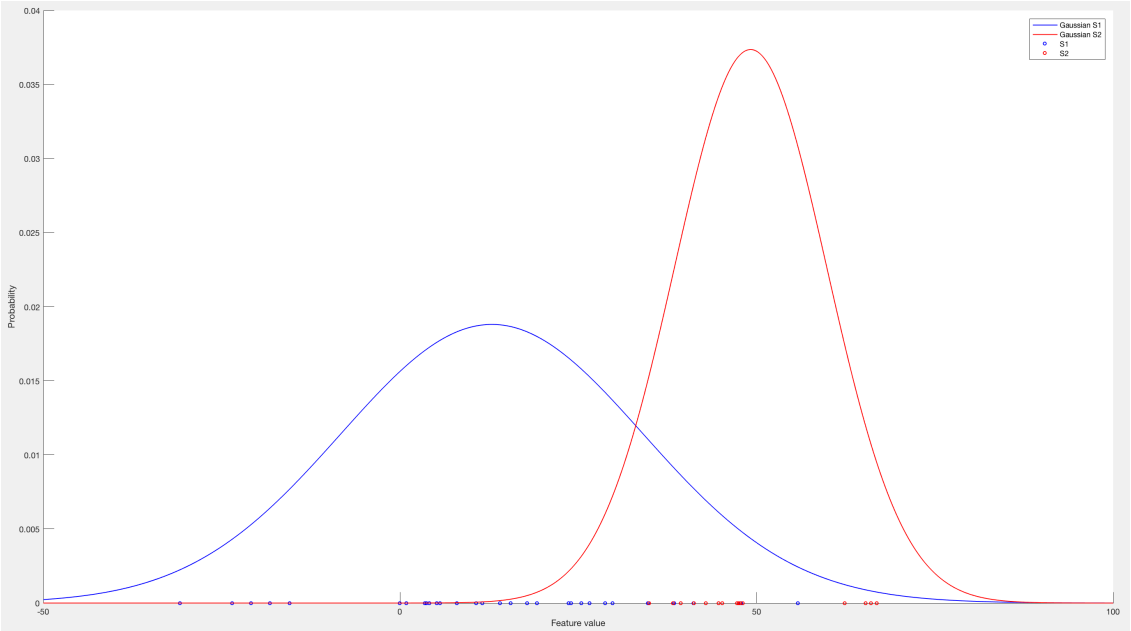

Figure 5: *shows the plot of the elements of S1 and S2 and of the gaussian distributions that generated them*

3. To estimate the prior probabilities we consider the relative frequencies of occurrence: the set S1 provides us with 30 elements, while S2 has only 15 elements. Thus the total amount of elements is 45. To compute the relative frequency of occurrence for S1 we just have to divide the number of elements of S1 by 45, obtaining 0.6667. Applying the same procedure to S2 leads to a value of 0.3333. Thus the prior probabilities are 0.6667 for ω_1 and 0.3333 for ω_2.

4. When we multiply the prior probability by the class conditional probability, we obtain a function which is not a probability density function anymore(if we integrate it in its domain we don't obtain 1, but only 0.6667 for ω_1 and 0.3333 for ω_2), but just a relative probability function, meaning that it still gives information about the given event, but it is not scaled to be a probability density. To scale it properly we should divide it by the evidence, but we do not need it for the purpose of this exercise, since it only asks us to establish a decision criterion and not to compute the posterior probabilities.
Considering the discussed features, we can expect this function to have the same shape as the class conditional probability, just differently scaled. **Figure 6** shows the plots of the two relative probability functions.

Plot of S1, S2 and the relative probability functions

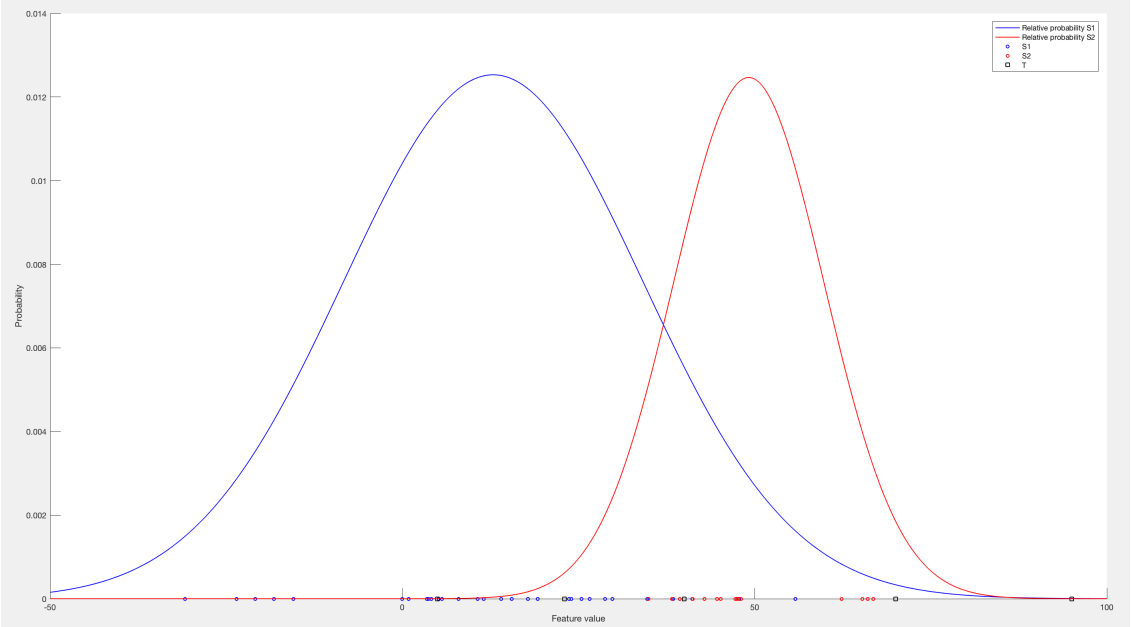

Figure 6: *shows the plot of the elements of S1 and S2 and of their relative probability functions*

5. To find the decision criterion we solve the equation

$$p(\omega_1)p(x|\omega_1) = p(\omega_2)p(x|\omega_2)$$

and find the values for which the left side of the equation is bigger than the right side. The code used to compute this is in **Listing 4** .

Listing 4: *Code to compute the solution of the equation*

```
    equation = p_w1 * (1/(sqrt(2*pi) * std_S1)) * exp(- ((y -
mean_S1).^2)/(2*(std_S1.^2))) == p_w2 * (1/(sqrt(2*pi) * std_S2
)) * exp(- ((y - mean_S2).^2)/(2*(std_S2.^2)));
    decision_criterion = double(solve(equation, y));
```

The solution to the previous equation is thus given by $x_1 = 37.06$ and $x_2 = 85.9$. This means that every value less than 37.06 or greater than 85.9 will be classified as ω_1, while the values in between will be classified as ω_2.
The actual procedure to compute the decision criterion usually uses the posterior probabilities instead of the relative probabilities that we used. But, when put in an equation like the one

we're solving, the evidence at the denominator can be simplified, leaving thus only the two relative probabilities.

6. Following the rule stated above, the values 5, 23 and 95 (from the test set) are classified as ω_1, while 40 and 70 are classified as ω_2. **Figure 7** shows the previous plot modified to immediately identify the domain of class ω_1 and the domain of class ω_2: the former is defined by the blue segments under the x-axis, while the latter is defined by the red segment under the x-axis.

Plot of S1, S2, T and the relative probabilities, with highlighted domains

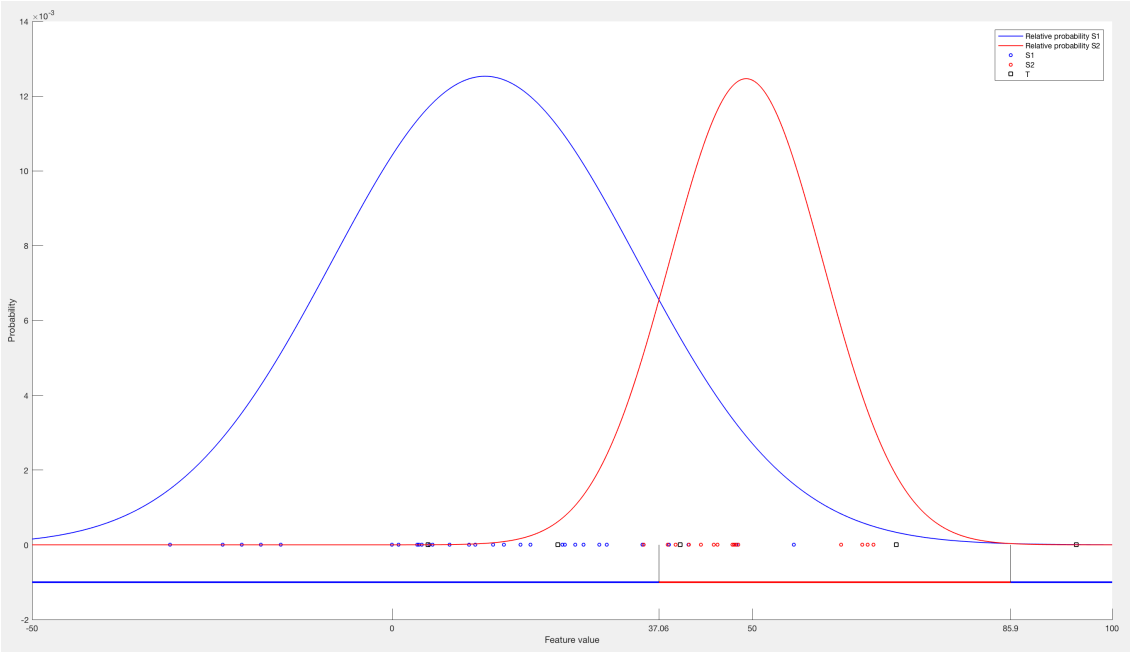

Figure 7: *shows the plot of the elements of S1, S2, T and the relative probabilities and shows how the domain is divided to classify to class ω_1 and ω_2*

7. The misclassification rate for class ω_1 is: $\int_{37.06}^{85.9} P(\omega_1)p(x|\omega_1)$, while the misclassification rate for class ω_1 is: $\int_{-\infty}^{37.06} P(\omega_2)p(x|\omega_2) + \int_{85.9}^{\infty} P(\omega_2)p(x|\omega_2)$. Basically they correspond to the area covered by their relative probability function for x outside their classification domain. In this case, the misclassification rate for ω_1 is 0.1268, while for ω_2 it is 0.1292. The method we used to compute the decision criterion should lead us to the minimum possible value of the misclassification rates (if we had chosen a different decision criterion, the sum of the misclassification rates would have certainly been higher).

A. Assignment 1

Listing 5: *This is the file* `script1.m`

```
%Load the data
A = load('lab3_1.mat');
length = A.l;
class_conditional_sb = A.p_seabass;
class_conditional_sl = A.p_salmon;

%Prior probabilities
prior_sb = 0.75; prior_sl = 0.25;

%Compute evidence
evidence = prior_sb*class_conditional_sb + prior_sl*
class_conditional_sl;

%Compute posterior probability
posterior_sb = zeros(1,241); posterior_sl = zeros(1,241);
for i = 1:241
    posterior_sb(i) = (class_conditional_sb(i) * prior_sb)/
evidence(i);
    posterior_sl(i) = (class_conditional_sl(i) * prior_sl)/
evidence(i);
end

%Plot the posterior probabilities
hold on;
figure(1);
plot(length, posterior_sb, 'Color', 'blue');
plot(length, posterior_sl, 'Color', 'red');
hold off;
legend('Sea Bass', 'Salmon');
xlabel ('Length');
ylabel ('Posterior probability');

%-----------------------------------------
%Plot the posterior probabilities with the new fishes highlighted

%New fishes length
fish1_length = 8;
fish2_length = 20;

%Compute the probability of fish1 being a sea bass or a salmon
pr_sb_fish1 = posterior_sb((fish1_length-0.9)*10);
pr_sl_fish1 = posterior_sl((fish1_length-0.9)*10);

%Compute the probability of fish2 being a sea bass or a salmon
pr_sb_fish2 = posterior_sb((fish2_length-0.9)*10);
pr_sl_fish2 = posterior_sl((fish2_length-0.9)*10);
```

```
%Plot all the data
figure(2);
hold on;
hAx = gca;
plot(length, posterior_sb, 'Color', 'blue');
plot(length, posterior_sl, 'Color', 'red');
line([fish1_length fish1_length],[0 pr_sb_fish1], 'Color', 'green'
);
line([fish2_length fish2_length],[0 pr_sb_fish2], 'Color', 'green'
);

line([0 fish1_length],[pr_sb_fish1 pr_sb_fish1], 'Color', 'blue');
line([0 fish1_length],[pr_sl_fish1 pr_sl_fish1], 'Color', 'red');

line([0 fish2_length],[pr_sb_fish2 pr_sb_fish2], 'Color', 'blue');
line([0 fish2_length],[pr_sl_fish2 pr_sl_fish2], 'Color', 'red');
set(hAx, 'YTick', [0 0.1 pr_sl_fish2 0.2 0.3 pr_sl_fish1 0.4 0.5
0.6 pr_sb_fish1 0.7 0.8 pr_sb_fish2 0.9 1]);

hold off;
legend('Sea Bass', 'Salmon', '8cm Fish', '20cm Fish');
xlabel ('Length');
ylabel ('Posterior probability');

%Compute and plot the error function
figure(3);
err_fun = zeros(1,size(posterior_sb,2));
hold on;
for i = 1:241
    err_fun(i) = min(posterior_sb(i), posterior_sl(i));
end
plot(length, err_fun);
ylim([0 1]);
hold off;
legend('Error probability');
xlabel ('Length');
ylabel ('Error probability');
```

B. Assignment 2

Listing 6: *This is the file* `script2.m`

```
%Load data
S = load('normdist.mat');
S1 = S.S1;
S2 = S.S2;
T = S.T;

%{
%Data analisys for symmetry
%Computations for S1
med_S1 = median(S1);
mean_S1 = mean(S1);
Q1_S1 = quantile(S1, 0.25);
Q3_S1 = quantile(S1, 0.75);
%Computations for S2
med_S2 = median(S2);
mean_S2 = mean(S2);
Q1_S2 = quantile(S2, 0.25);
Q3_S2 = quantile(S2, 0.75);
%Print the result
fprintf("Mean_S1: %f\tMedian_S1: %f\nQ1_S1: %f\tQ3_S1: %f\
    tExpectedMedian_S1: %f\n", mean_S1, med_S1, Q1_S1, Q3_S1, (Q3_S1 +
    Q1_S1)/2);
fprintf("Mean_S2: %f\tMedian_S2: %f\nQ1_S2: %f\tQ3_S2: %f\
    tExpectedMedian_S2: %f\n", mean_S2, med_S2, Q1_S2, Q3_S2, (Q3_S2 +
    Q1_S2)/2);
%}

% Display S1, S2 and T
figure(1);
hold on;
p_S1 = plot(S1, 0, 'bo', 'MarkerSize',4);
p_S2 = plot(S2, 0, 'ro', 'MarkerSize',4);
p_T = plot(T, 0, 'ks', 'MarkerSize',6);
line([-50 100],[0 0]);
hold off;
xlabel('Feature value');
set(gca, 'YTick', []);
legend([p_S1(1) p_S2(1) p_T(1)], 'S1', 'S2', 'T');



%Compute and display gaussians
mean_S1 = mean(S1); std_S1 = std(S1);
mean_S2 = mean(S2); std_S2 = std(S2);
x = linspace(-50, 100,1000);
norm_S1 = normpdf(x, mean_S1, std_S1);
norm_S2 = normpdf(x, mean_S2, std_S2);
```

```
figure(2);
hold on;
p_S1 = plot(S1, 0, 'bo', 'MarkerSize',4);
p_S2 = plot(S2, 0, 'ro', 'MarkerSize',4);
p_norm_S1 = plot(x, norm_S1, 'LineWidth', 1, 'Color', 'blue');
p_norm_S2 = plot(x, norm_S2, 'LineWidth', 1, 'Color', 'red');
hold off;
legend ([p_norm_S1 p_norm_S2 p_S1(1) p_S2(1)], 'Gaussian S1', '
    Gaussian S2', 'S1', 'S2');
xlabel('Feature value');
ylabel('Probability');

%Estimate P(w1) and P(w2)
data_set_size = size(S1) + size(S2);
p_w1 = size(S1)/data_set_size;
p_w2 = size(S2)/data_set_size;

%Plot points with relative functions
figure(3);
hold on;
p_S1 = plot(S1, 0, 'bo', 'MarkerSize',4);
p_S2 = plot(S2, 0, 'ro', 'MarkerSize',4);
p_T = plot(T, 0, 'ks', 'MarkerSize',6);
p_rel_S1 = plot(x, norm_S1*p_w1, 'LineWidth', 1, 'Color', 'blue');
p_rel_S2 = plot(x, norm_S2*p_w2, 'LineWidth', 1, 'Color', 'red');
hold off;
legend ([p_rel_S1 p_rel_S2 p_S1(1) p_S2(1) p_T(1)], 'Relative
    probability S1', 'Relative probability S2', 'S1', 'S2', 'T');
xlabel('Feature value');
ylabel('Probability');

%Find decision criterion
syms y;
rtp = sqrt(2*pi);
equation = p_w1 * (1/(rtp * std_S1)) * exp(- ((y - mean_S1).^2)/(2*(
    std_S1.^2))) == p_w2 * (1/(rtp * std_S2)) * exp(- ((y - mean_S2)
    .^2)/(2*(std_S2.^2)));
decision_criterion = double(solve(equation, y));

%Plot with clear distinction between domains
figure(4);
hold on;
p_S1 = plot(S1, 0, 'bo', 'MarkerSize',4);
p_S2 = plot(S2, 0, 'ro', 'MarkerSize',4);
p_T = plot(T, 0, 'ks', 'MarkerSize',6);
p_rel_S1 = plot(x, norm_S1*p_w1, 'LineWidth', 1, 'Color', 'blue');
p_rel_S2 = plot(x, norm_S2*p_w2, 'LineWidth', 1, 'Color', 'red');

line([-50 decision_criterion(1)],[-0.001 -0.001], 'Color', 'blue', '
    LineWidth', 2);
```

```
line([decision_criterion(1) decision_criterion(2)],[-.001 -.001], '
   Color', 'red', 'LineWidth', 2);
line([decision_criterion(2) 100],[-0.001 -0.001], 'Color', 'blue', '
   LineWidth', 2);

line([decision_criterion(1) decision_criterion(1)],[-0.001 0], 'Color',
    'black');
line([decision_criterion(2) decision_criterion(2)],[-0.001 0], 'Color',
    'black');

hold off;
legend ([p_rel_S1 p_rel_S2 p_S1(1) p_S2(1) p_T(1)], 'Relative
   probability S1', 'Relative probability S2', 'S1', 'S2', 'T');
xlabel('Feature value');
ylabel('Probability');
set(gca, 'XTick', [-50 0 37.06 50 85.9 100]);

%Misclassification rates
cdf_S1 = normcdf(x, mean_S1, std_S1);
cdf_S2 = normcdf(x, mean_S2, std_S2);

error_S1 = cdf_S1(ceil((decision_criterion(2)+50)/.15)) - cdf_S1(ceil
   ((decision_criterion(1)+50)/.15));
error_S2 = 1 + cdf_S2(ceil((decision_criterion(1)+50)/.15)) - cdf_S2(
   ceil((decision_criterion(2) + 50)/.15));
fprintf("Errors_ %f\t%f\n", error_S1, error_S2);
```