Introduction to Intelligent Systems - Lab 3

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

September 28, 2018

Contents

1 Assignment 1

1.1

Given that sea bass is caught 3 times as often as salmon, calculate the normalised posterior probabilities and plot them.

Since sea bass is caught 3 times more often than salmon, the a priori probabilities are 0.75 and 0.25 respectively. The computed posterior probabilities are plotted in Figure 1 below. This corresponds with lines 8-23 of the code at Appendix 3.1.

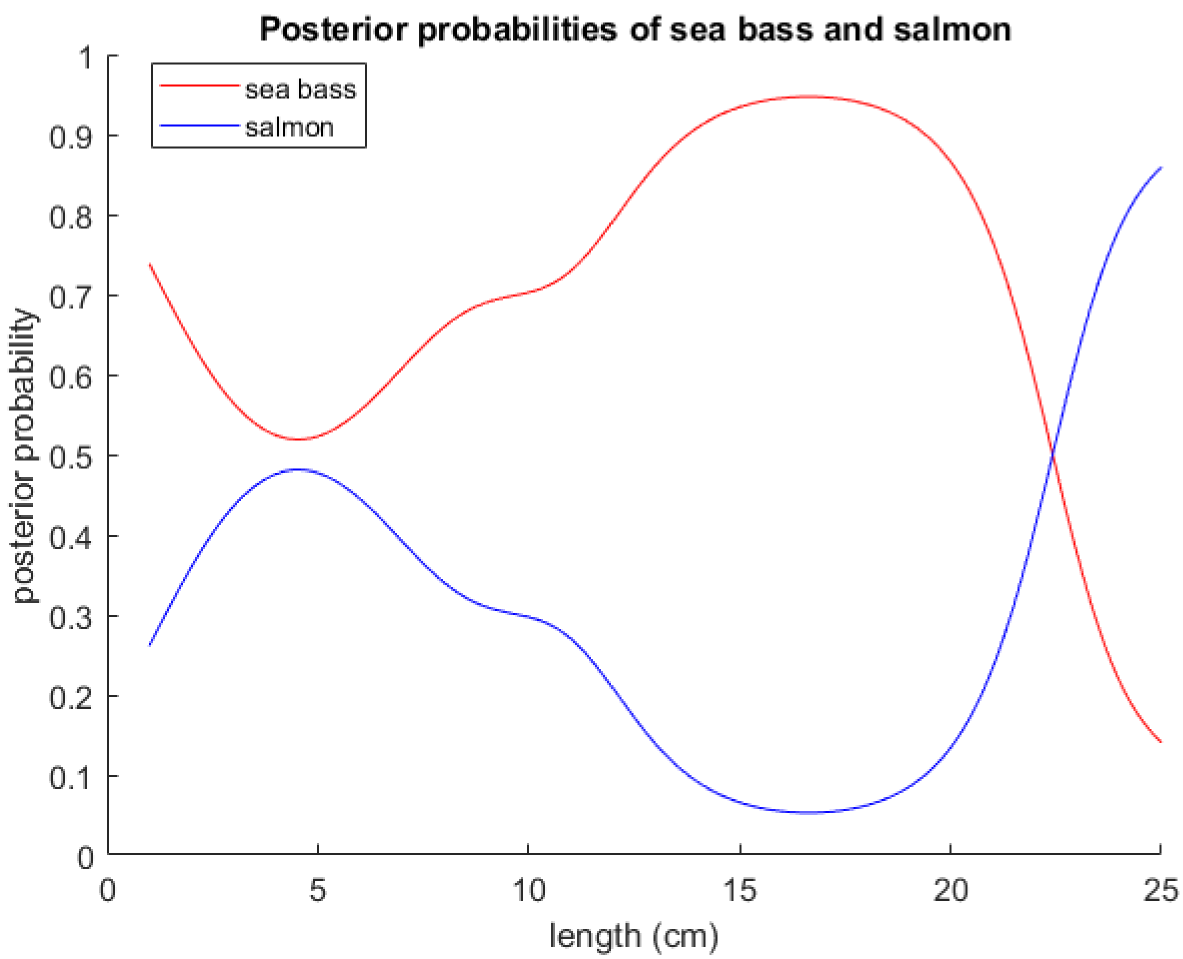

Figure 1: Normalised posterior probabilities of sea bass and salmon.

1.2

According to your posterior probabilities, classify fishes with the lengths of 8 and 20 cm respectively.

For a fish with a length of 8 cm, the posterior probabilities of it being a sea bass or salmon are 0.66 and 0.34 respectively, hence we classify it as a sea bass.

For a fish with a length of 20 cm, the posterior probabilities of it being a sea bass or a salmon are 0.87 and 0.13 respectively, hence we classify it as a sea bass as well.

See Figure 2 for a more visual representation. This corresponds with lines 27-49 of the code at Appendix 3.1.

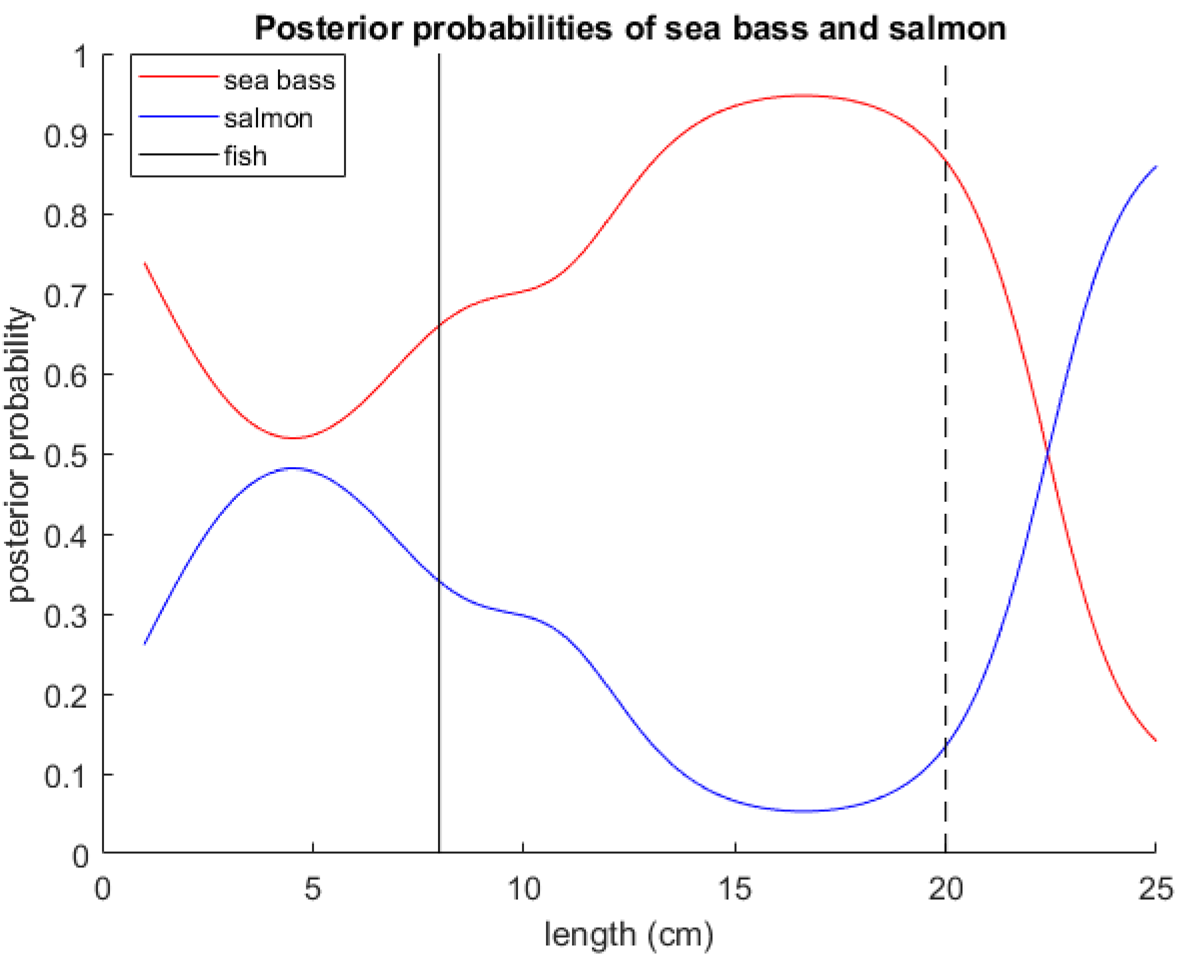

Figure 2: Normalised posterior probabilities of sea bass and salmon, together with lines indicating fishes with a length of 8 or 20 cm.

2 Assignment 2

2.1

Plot the elements of the two sets S1 and S2 on the x-axis of a Fig. 1. Plot in the same Fig. 1 the points of the test set T.
See Figure 3. Note that the x-axes of all figures in this assignment do not contain a label, since no unit was specified. The corresponding code is at lines 7-23 at Appendix 3.2.

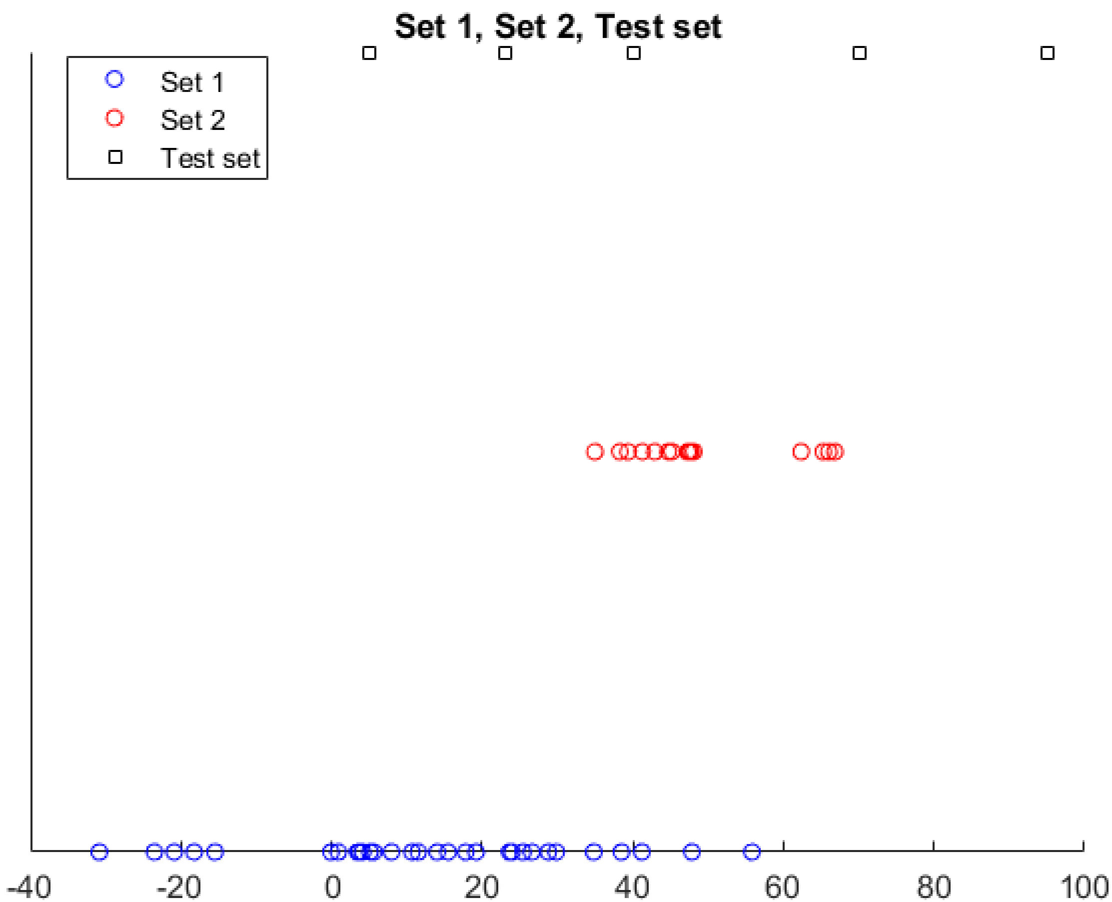

Figure 3: The elements of sets S1, S2 and the test set T.

2.2

Let's assume S1 and S2 are drawn from normal distributions. Compute the parameters (mean and standard deviation) of the two distributions, using maximum estimation. Create a Fig.2 in which you plot the two Gaussian functions, one in blue the other in red, together with the points of the two training data sets S1 and S2. The two functions are the class conditional probability densities $p(x|\omega_1)$ and $p(x|\omega_2)$ that the two classes produce a value x of the considered feature. Comment on the use of the functions, in comparison to using the rugged data.

See Figure 4.

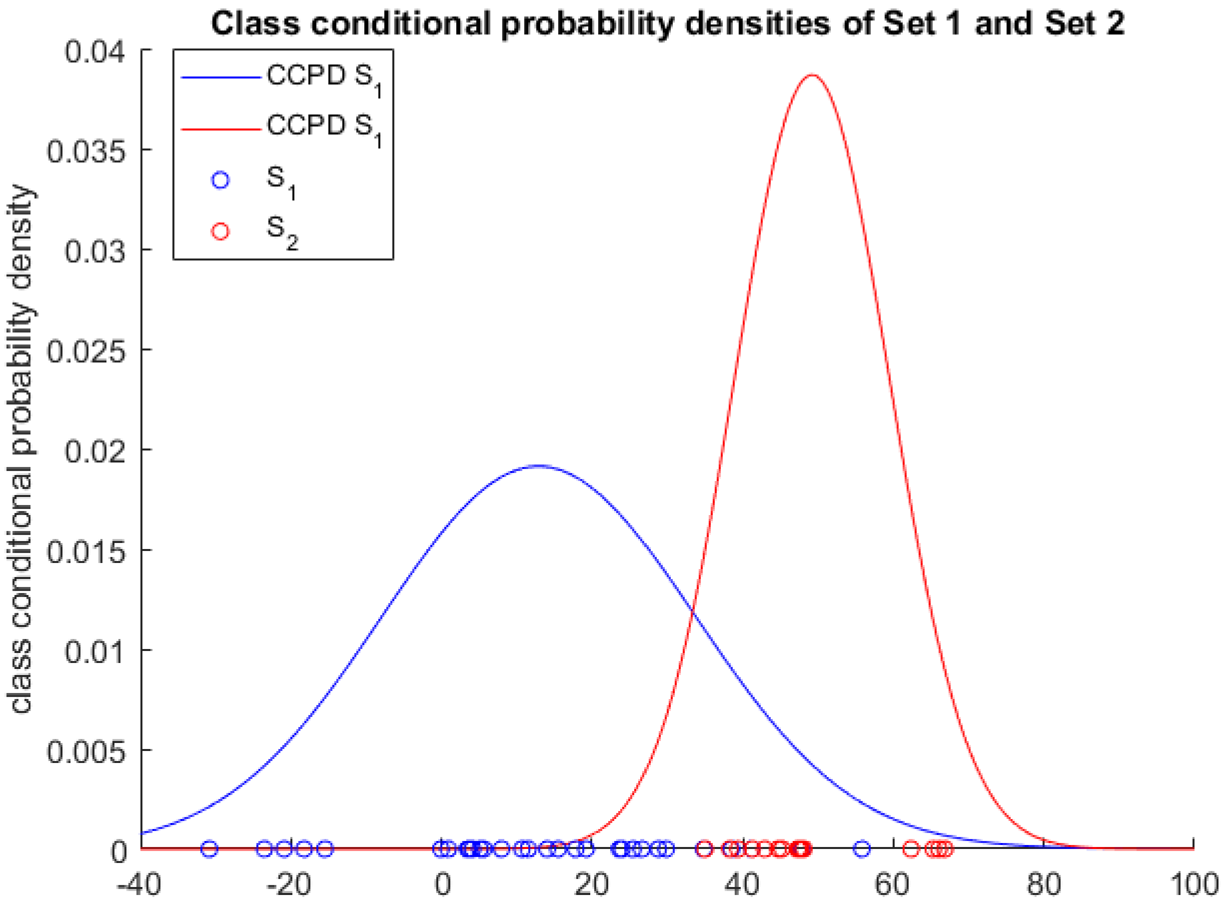

Figure 4: Class conditional probability densities of Set 1 and Set 2, together with their corresponding original data points.

These class conditional probability functions are, when used in a situation where it can be assumed the underlying distribution is in fact (quite similar to) a normal distribution, a lot more versatile than simply using the rugged data, since they are continuous and thus we can interpolate and, more importantly, extrapolate using these functions.
The main downside is that, when using them, we are assuming the underlying distribution has a particular shape (e.g. a bell curve), while this may not be the case. Outside of having a theoretical model to which the underlying distribution 'should' adhere to, the uncertainty of how the underlying distribution actually looks like decreases with a larger sample size.
The corresponding code is at lines 27-55 at Appendix 3.2.

2.3

Estimate the prior probabilities $P(\omega_1)$ and $P(\omega_2)$.

$$P(\omega_1) = \frac{length(S_1)}{length(S_1) + length(S_2)} = \frac{30}{30+15} = \frac{2}{3}$$
$$P(\omega_2) = \frac{length(S_2)}{length(S_1) + length(S_2)} = \frac{15}{30+15} = \frac{1}{3}$$

The corresponding code is at lines 59-60 at Appendix 3.2.

2.4

Create a Fig. 3 in which you plot the two products $P(\omega_1)p(x|\omega_1)$ and $P(\omega_2)p(x|\omega_2)$, together with the points of the two training data sets S1 and S2 and the test set T.

See Figure 5.

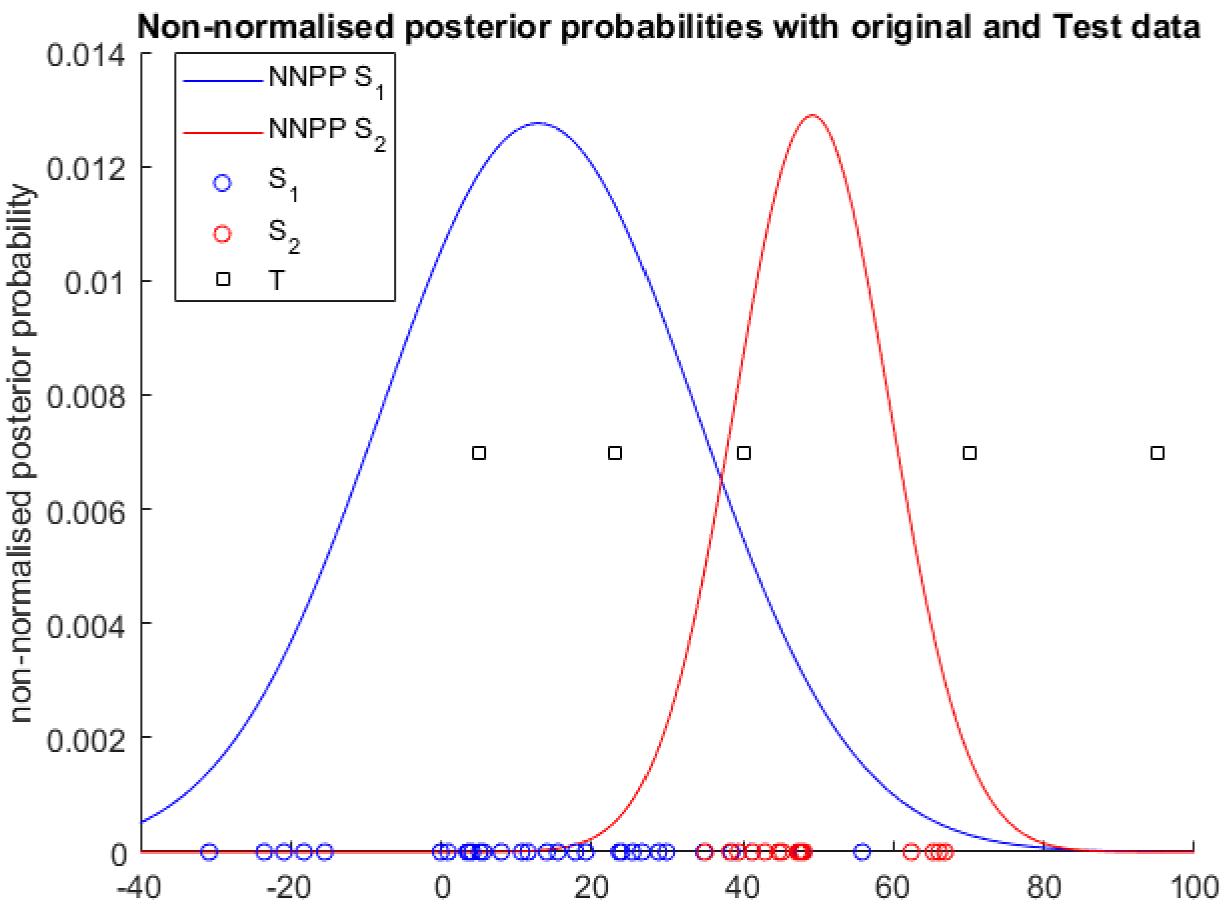

Figure 5: Non-normalised posterior probabilities $(P(\omega_1)p(x|\omega_1)$ and $P(\omega_2)p(x|\omega_2))$ of S_1 and S_2 with original and Test data.

The corresponding code is at lines 64-83 at Appendix 3.2.

2.5

Substitute the values of the prior probabilities and the class conditional probability densities determined above in the equation $P(\omega_1)p(x|\omega_1) = P(\omega_2)p(x|\omega_2)$ and solve the resulting equation for x in order to determine the value(s) of the decision criterion that we should use for classification.

The following substitution and simplification was used to determine the x values to use as decision criterions. To check our answers, we decided to let Matlab solve and evaluate the initial equation on its own. To our delight, both methods resulted in the same answers.

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

$\equiv$

$$P(\omega_1) \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\hat{\sigma}_1} \cdot e^{-\frac{(x-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2}} = P(\omega_2) \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\hat{\sigma}_2} \cdot e^{-\frac{(x-\hat{\mu}_2)^2}{2\hat{\sigma}_2^2}}$$

$\equiv$

$$\hat{\sigma}_2 \cdot P(\omega_1) \cdot e^{-\frac{(x-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2}} = \hat{\sigma}_1 \cdot P(\omega_2) \cdot e^{-\frac{(x-\hat{\mu}_2)^2}{2\hat{\sigma}_2^2}}$$

$\equiv$

$$\ln(\hat{\sigma}_2 \cdot P(\omega_1)) - \frac{(x-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2} = \ln(\hat{\sigma}_1 \cdot P(\omega_2)) - \frac{(x-\hat{\mu}_2)^2}{2\hat{\sigma}_2^2}$$

$\equiv$

$$\frac{(x-\hat{\mu}_2)^2}{2\hat{\sigma}_2^2} - \frac{(x-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2} + \ln(\frac{\hat{\sigma}_2 \cdot P(\omega_1)}{\hat{\sigma}_1 \cdot P(\omega_2)}) = 0$$

$\equiv$

$$\frac{\hat{\sigma}_1^2(x-\hat{\mu}_2)^2 - \hat{\sigma}_2^2(x-\hat{\mu}_1)^2}{2\hat{\sigma}_1^2\hat{\sigma}_2^2} + \ln(\frac{\hat{\sigma}_2 \cdot P(\omega_1)}{\hat{\sigma}_1 \cdot P(\omega_2)}) = 0$$

$\equiv$

$$\hat{\sigma}_1^2(x-\hat{\mu}_2)^2 - \hat{\sigma}_2^2(x-\hat{\mu}_1)^2 + 2\hat{\sigma}_1^2\hat{\sigma}_2^2 \cdot \ln(\frac{\hat{\sigma}_2 \cdot P(\omega_1)}{\hat{\sigma}_1 \cdot P(\omega_2)}) = 0$$

$\equiv$

$$\hat{\sigma}_1^2(x^2 - 2\hat{\mu}_2 x + \hat{\mu}_2^2) - \hat{\sigma}_2^2(x^2 - 2\hat{\mu}_1 x + \hat{\mu}_1^2) + 2\hat{\sigma}_1^2\hat{\sigma}_2^2 \cdot \ln(\frac{\hat{\sigma}_2 \cdot P(\omega_1)}{\hat{\sigma}_1 \cdot P(\omega_2)}) = 0$$

$\equiv$

$$(\hat{\sigma}_1^2 - \hat{\sigma}_2^2)x^2 + 2(\hat{\sigma}_2^2\hat{\mu}_1 - \hat{\sigma}_1^2\hat{\mu}_2)x + \hat{\sigma}_1^2\hat{\mu}_2^2 - \hat{\sigma}_2^2\hat{\mu}_1^2 + 2\hat{\sigma}_1^2\hat{\sigma}_2^2 \cdot \ln(\frac{\hat{\sigma}_2 \cdot P(\omega_1)}{\hat{\sigma}_1 \cdot P(\omega_2)}) = 0$$

From this we can determine the constants a, b and c from the quadratic formula and use it to find our x values.
$a = \hat{\sigma}_1^2 - \hat{\sigma}_2^2$

$b = 2(\hat{\sigma}_2^2\hat{\mu}_1 - \hat{\sigma}_1^2\hat{\mu}_2)$

$c = \hat{\sigma}_1^2\hat{\mu}_2^2 - \hat{\sigma}_2^2\hat{\mu}_1^2 + 2\hat{\sigma}_1^2\hat{\sigma}_2^2 \cdot \ln(\frac{\hat{\sigma}_2 \cdot P(\omega_1)}{\hat{\sigma}_1 \cdot P(\omega_2)})$

$x_1 = \frac{-b-\sqrt{b^2-4ac}}{2a} \approx 37.1$

$x_2 = \frac{-b+\sqrt{b^2-4ac}}{2a} \approx 84.7$

These values seem to visually coincide with the points where the two functions are equal, as seen in part 6.

See the code at lines 88-101 for the manual calculation, and 104-119 for the automatic solving and evaluation of the initial equation at Appendix 3.2.

2.6

Using the thus obtained value(s) of the decision criterion, determine the classes of the points in the test set [5 23 40 70 95]. Create a Fig. 4 which copies Fig. 3 and adds to it a specification of which domain on the x-axis belongs to class 1 (blue) and which complementary domain corresponds to class 2 (red).
According to our previously obtained decision criterion, the test set would be classified in the following manner:

$\omega_1 \ni \ [5, 23, 95]$
$\omega_2 \ni \ [40, 70]$

See Figure 6 for the visual representation of this classification.

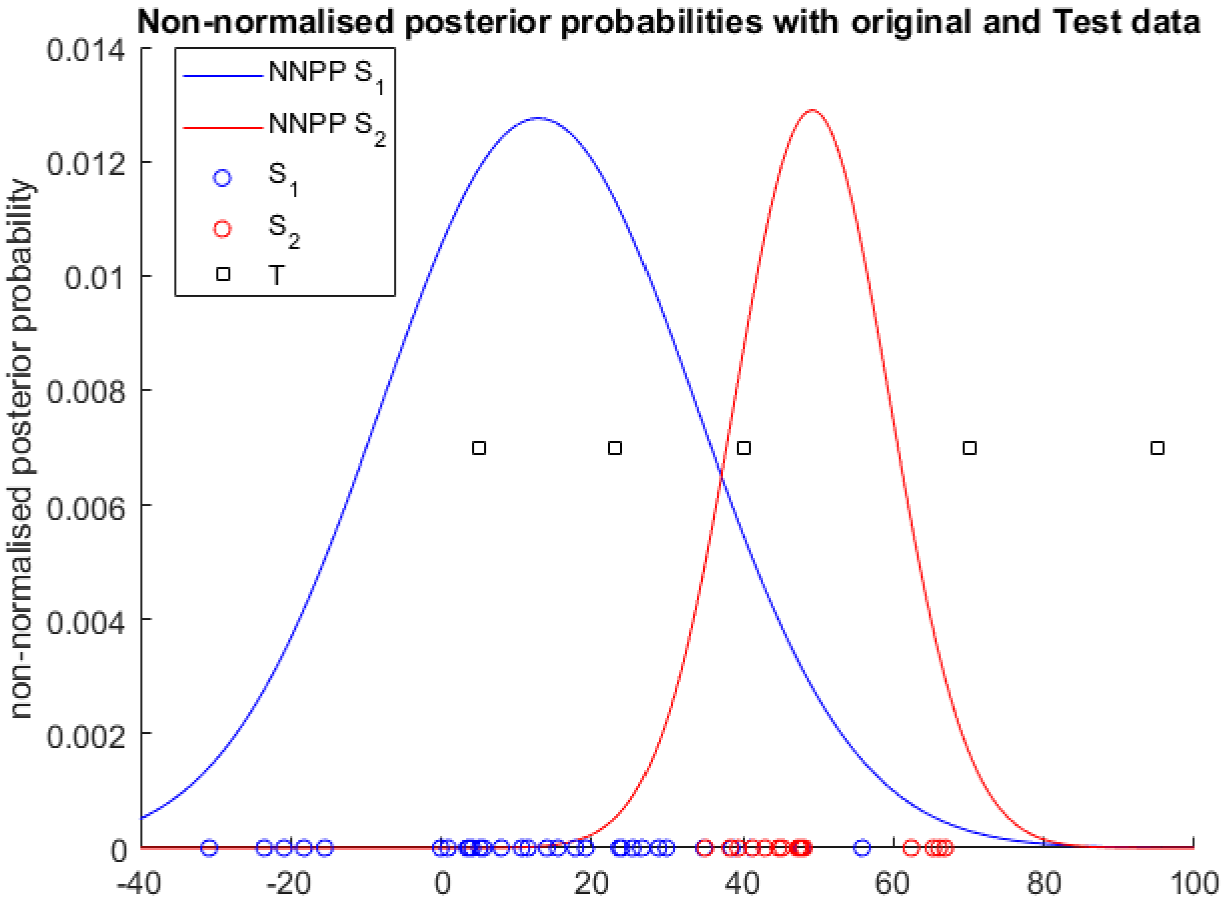

Figure 6: Non-normalised posterior probabilities ($P(\omega_1)p(x|\omega_1)$ and $P(\omega_2)p(x|\omega_2)$) of S_1 and S_2 with original data, Test data, and decision criterions.

The corresponding code is at lines 124-143 at Appendix 3.2.

2.7

Evaluate the misclassification rate of the created classifier for each of the two classes.
The misclassification rate of ω_1 is:

$$\sum_{i=1}^{n}(\int_{lb_i}^{ub_i} P(\omega_2)p(x|\omega_2)\ dx)$$

where lb_i and ub_i are the respective lower- and upper bounds of each i-th area that is classified as ω_1, and vice versa for the misclassification rate of ω_2.
Going from our plot, this gives us the following misclassification rates:

$$Misclassification\, rate\, \omega_1 \approx \int_{-\infty}^{37.1} P(\omega_2)p(x|\omega_2)\ dx + \int_{84.7}^{\infty} P(\omega_2)p(x|\omega_2)\ dx \approx 0.1213$$

$$Misclassification\ rate\ \omega_2 \approx \int_{37.1}^{84.7} P(\omega_1)p(x|\omega_1)\ dx\ \ \approx\ \ 0.1227$$

In our code we used the normal cumulative distribution function (`normcdf`) to compute these integrals. See lines 147-153 of the code at Appendix 3.2.

3 Appendix

3.1

```
close all;

load('../lab_week3_data/lab3_1.mat')

%========================== 1.1 ================================%
% compute posterior probabilities
% occurences of sea bass vs salmon is 3:1
a_priori_seabass = .75;
a_priori_salmon = .25;

evidence = p_seabass * a_priori_seabass + p_salmon * a_priori_salmon;

posterior_seabass = p_seabass * a_priori_seabass ./ evidence;
posterior_salmon = p_salmon * a_priori_salmon ./ evidence;

% plot figure
figure; hold on;
plot(l, posterior_seabass, 'r-');
plot(l, posterior_salmon, 'b-');
xlabel('length (cm)');
ylabel('posterior probability');
title('Posterior probabilities of sea bass and salmon');
legend('sea bass', 'salmon');

%========================== 1.2 ================================%
% get posterior probabilities for lengths 8 and 20
l_8_index = 71;
l_20_index = 191;
pos_sb_8 = posterior_seabass(l_8_index)
pos_sal_8 = posterior_salmon(l_8_index)
pos_sb_20 = posterior_seabass(l_20_index)
pos_sal_20 = posterior_salmon(l_20_index)


% plot figure
figure; hold on;
x_vals_8 = zeros(21)+8;
x_vals_20 = zeros(21)+20;
y_vals = 0:.05:1;

plot(l, posterior_seabass, 'r-');
plot(l, posterior_salmon, 'b-');
```

```
plot(x_vals_8, y_vals, 'k-');
plot(x_vals_20, y_vals, 'k--');

xlabel('length (cm)');
ylabel('posterior probability');
title('Posterior probabilities of sea bass and salmon');
legend('sea bass', 'salmon', 'fish');
```

3.2

```
close all;

load('../lab_week3_data/normdist.mat');

%=========================== 2.1 ===================================%
% create plot
figure; hold on;

% create plot handles for legend items
h = zeros(3,1);
h(1) = plot(nan, nan, 'bo');
h(2) = plot(nan, nan, 'ro');
h(3) = plot(nan, nan, 'ks');

plot(S1, zeros(30), 'bo');
plot(S2, zeros(15)+1, 'ro');
plot(T, zeros(5)+2, 'ks');

% Remove unnecessary y-axis ticks.
set(gca,'YTickLabel',[]);
set(gca,'YTick',[]);
title('Set 1, Set 2, Test set');
legend(h, 'Set 1', 'Set 2', 'Test set');

%=========================== 2.2 ===================================%
% compute mean and standard deviation of S1 and S2
mu_hat_S1 = sum(S1) / length(S1);
sigma_hat_S1 = sqrt(sum((S1-mu_hat_S1).*(S1-mu_hat_S1)) / length(S1));

mu_hat_S2 = sum(S2) / length(S2);
sigma_hat_S2 = sqrt(sum((S2-mu_hat_S2).*(S2-mu_hat_S2)) / length(S2));

% create and plot the normal distributions
x_values = -40:.1:100;

pdf_S1 = normpdf(x_values, mu_hat_S1, sigma_hat_S1);
pdf_S2 = normpdf(x_values, mu_hat_S2, sigma_hat_S2);

figure; hold on;

% create plot handles for legend items
h = zeros(4,1);
h(1) = plot(nan, nan, 'b-');
h(2) = plot(nan, nan, 'r-');
```

```matlab
h(3) = plot(nan, nan, 'bo');
h(4) = plot(nan, nan, 'ro');


plot(x_values, pdf_S1, 'b-');
plot(x_values, pdf_S2, 'r-');
plot(S1, zeros(30), 'bo');
plot(S2, zeros(15), 'ro');
ylabel('class conditional probability density');
title('Class conditional probability densities of Set 1 and Set 2');
legend(h, 'CCPD S_1', 'CCPD S_1', 'S_1', 'S_2');

%============================== 2.3 ==============================%
% compute a priori probabilities
prior_S1 = length(S1)/(length(S1)+length(S2))
prior_S2 = length(S2)/(length(S1)+length(S2))

%============================== 2.4 ==============================%

figure; hold on;

% create plot handles for legend items
h = zeros(5,1);
h(1) = plot(nan, nan, 'b-');
h(2) = plot(nan, nan, 'r-');
h(3) = plot(nan, nan, 'bo');
h(4) = plot(nan, nan, 'ro');
h(5) = plot(nan, nan, 'ks');

% create plot
plot(x_values, prior_S1*pdf_S1, 'b-');
plot(x_values, prior_S2*pdf_S2, 'r-');
plot(S1, zeros(30), 'bo');
plot(S2, zeros(15), 'ro');
plot(T, zeros(5)+0.007, 'ks');
ylabel('non-normalised posterior probability');
title(['Non-normalised posterior probabilities with' ...
  ' original and Test data']);
legend(h, 'NNPP S_1', 'NNPP S_2', 'S_1', 'S_2', 'T');

%============================== 2.5 ==============================%

% manual calculation
a = sigma_hat_S1 * sigma_hat_S1 - sigma_hat_S2 * sigma_hat_S2;
b = 2*(sigma_hat_S2 * sigma_hat_S2 * mu_hat_S1 - ...
    sigma_hat_S1 * sigma_hat_S1 * mu_hat_S2);
```

```
c = sigma_hat_S1 * sigma_hat_S1 * mu_hat_S2 * mu_hat_S2 - ...
    sigma_hat_S2 * sigma_hat_S2 * mu_hat_S1 * mu_hat_S1 + ...
    2 * sigma_hat_S1 * sigma_hat_S1 * sigma_hat_S2 * sigma_hat_S2 * ...
    log(prior_S1 * sigma_hat_S2 / (prior_S2 * sigma_hat_S1));

D = b*b - 4*a*c;

% Assuming D is non-negative and we won't have complex answers,
% sqrt(D) is always non-negative, so x1 <= x2.
x1 = (-b - sqrt(D)) / (2*a);
x2 = (-b + sqrt(D)) / (2*a);

% matlab calculation
syms 'x';
expression = solve( ...
    prior_S1 * sigma_hat_S2* ...
    exp(-(x-mu_hat_S1)*(x-mu_hat_S1)/(2*sigma_hat_S1*sigma_hat_S1)) ...
    == ...
    prior_S2 * sigma_hat_S1* ...
    exp(-(x-mu_hat_S2)*(x-mu_hat_S2)/(2*sigma_hat_S2*sigma_hat_S2)) ...
    );
x_eval = eval(expression);
if x_eval(1) < x_eval(2)  % ensure x_1 <= x_2
  x_1 = x_eval(1);
  x_2 = x_eval(2);
else
  x_1 = x_eval(2);
  x_2 = x_eval(1);
end

%=========================== 2.6 ===============================%
% create plot
% create plot handles for legend items
figure; hold on;
h = zeros(6,1);
h(1) = plot(nan, nan, 'b-');
h(2) = plot(nan, nan, 'r-');
h(3) = plot(nan, nan, 'bo');
h(4) = plot(nan, nan, 'ro');
h(5) = plot(nan, nan, 'ks');
h(6) = plot(nan, nan, 'k-');

plot(x_values, prior_S1*pdf_S1, 'b-');
plot(x_values, prior_S2*pdf_S2, 'r-');
plot(S1, zeros(30), 'bo');
plot(S2, zeros(15), 'ro');
```

```
plot(T, zeros(5)+0.007, 'ks');
plot([x_1 x_1], [0 0.014], 'k-');
plot([x_2 x_2], [0 0.014], 'k-');
ylabel('non-normalised posterior probability');
title(['Non-normalised posterior probabilities with' 10 ...
'original data, Test data, and decision criterions']);
legend(h, 'NNPP_S_1', 'NNPP_S_2', 'S_1', 'S_2', 'T', 'DC');

%========================= 2.7 ==============================%
%error rate S1
omega_1_error_a = normcdf(x_1, mu_hat_S2, sigma_hat_S2);
omega_1_error_b = 1 - normcdf(x_2, mu_hat_S2, sigma_hat_S2);
omega_1_error = omega_1_error_a + omega_1_error_b

%error rate S2
omega_2_error = normcdf(x_2, mu_hat_S1, sigma_hat_S1) - ...
    normcdf(x_1, mu_hat_S1, sigma_hat_S1)
```