Practical Exercise 5
Abgabe: Lena C. Wolos, Sven Niendorf, Isabelle Maye

Der Code ist zusätzlich als Datei einsehbar.

Part a:

Code:

```
%Task 1
%Part A

load ("twoClasses.mat")
c_0= patterns(:,1:2000);
c_0= transpose(c_0);
c_1= patterns(:,2001:4000);
c_1= transpose(c_1);


L_col=cell(1,3);          % cell array containing the labels for the columns
L_row=cell(2000,1);         % cell array containing the labels for the rows

L_col(1,2)="Sensor1";
L_col(1,3)="Sensor2";

for i=1:2000
  L_row(i,1)=strcat("Obs", int2str(i));
end

% Adding labels to c_0
L_col(1,1)="Class0";
c_0_L = num2cell(c_0);
c_0_L = cat (2,L_row, c_0_L);
c_0_L = cat (1,L_col, c_0_L);

% Adding labels to c_1
L_col(1,1)="Class1";
c_1_L = num2cell(c_1);
c_1_L = cat (2,L_row, c_1_L);
c_1_L = cat (1,L_col, c_1_L);

%assembeling cloud1
cloud1 =  cat(3, c_0_L, c_1_L);   %assambeling iris1


%defining help variables
Class0_sensor1= c_0(:,1);
Class0_sensor2= c_0(:,2);
Class1_sensor1= c_1(:,1);
Class1_sensor2= c_1(:,2);

```

Beispielaufrufe:

```
>> cloud1(1:3, 2, 1)
ans =
{
  [1,1] = Sensor1
  [2,1] = 3.6642
  [3,1] = 4.9162
}
```

```
>> cloud1(1:5,3,2)
ans =
{
  [1,1] = Sensor2
  [2,1] = 16.917
  [3,1] = 19.842
  [4,1] = 18.312
  [5,1] = 18.422
}
```

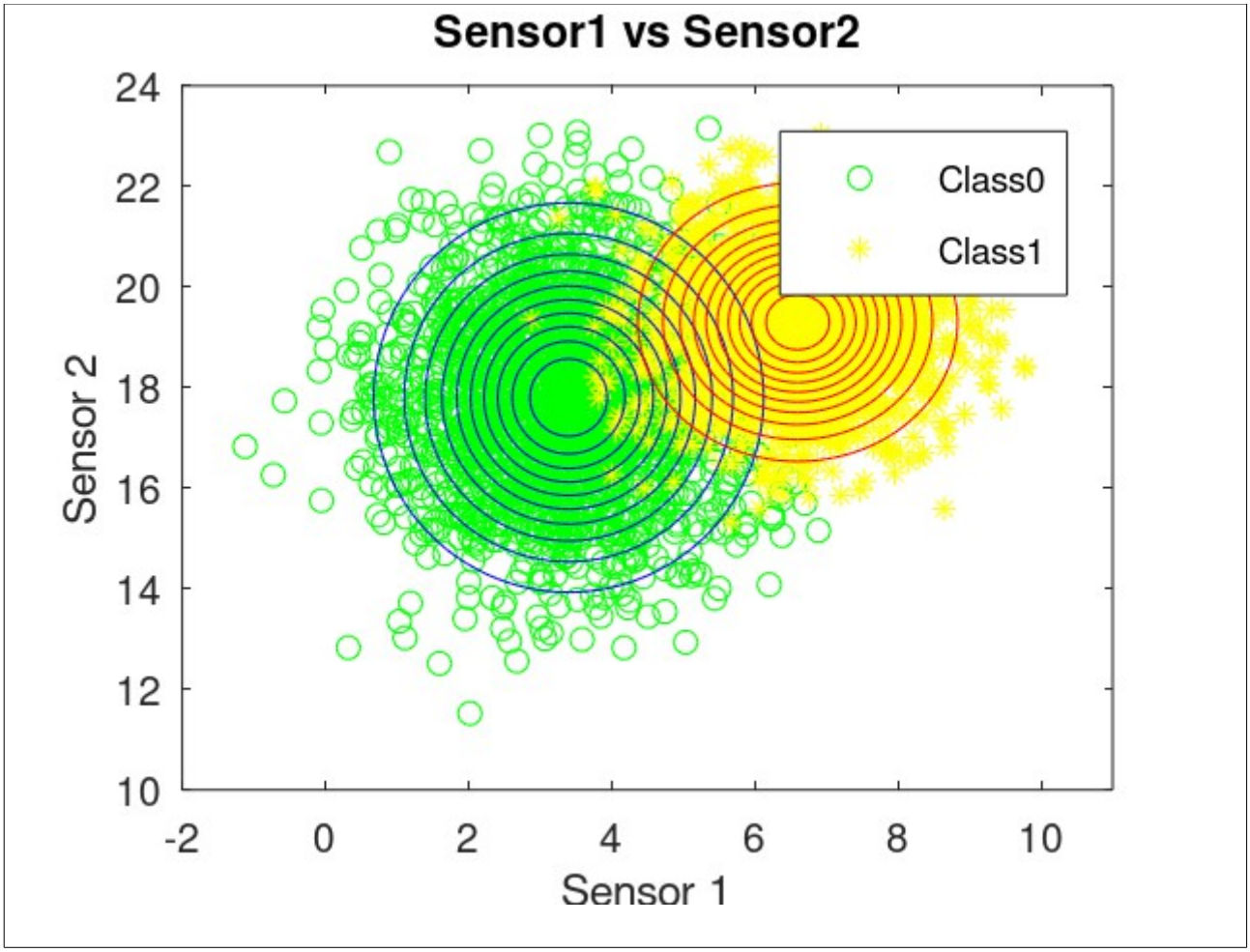
Part b:

Code:

```
%Part B
% Gaussian model with independent components

%Calculating mean for class 0 and 1
m01 = mean(Class0_sensor1);  %mean for class 0 and sensor 1
m02 = mean(Class0_sensor2);
m11 = mean(Class1_sensor1);
m12 = mean(Class1_sensor2);

%Calculating standard deviation for class 0 and 1
d01=sqrt(var(Class0_sensor1));
d02=sqrt(var(Class0_sensor2));
d11=sqrt(var(Class1_sensor1));
d12=sqrt(var(Class1_sensor2));

%Declaring interval of values for each sensor
i1= -2:0.1:11;
i2= 10:0.1:24;

%Calculating the maginals for class 0 and 1
p0_s1 = exp(-0.5*((i1-m01)./d01).^2)./(sqrt(2*pi)*d01);
p0_s2 = exp(-0.5*((i2-m02)./d02).^2)./(sqrt(2*pi)*d02);
p0_joint= p0_s2'*p0_s1;
p1_s1 = exp(-0.5*((i1-m11)./d11).^2)./(sqrt(2*pi)*d11);
p1_s2 = exp(-0.5*((i2-m12)./d12).^2)./(sqrt(2*pi)*d12);
p1_joint= p1_s2'*p1_s1;

```

```
70  %Plotting the two point distributions of the classes
71  %with contours of the two Gaussian distributions
72  figure (3);
73  plot(Class0_sensor1,Class0_sensor2,"go",Class1_sensor1,Class1_sensor2,"y*")
74    title('Sensor1 vs Sensor2');
75    xlabel ('Sensor 1');
76    ylabel ('Sensor 2');
77    legend("Class0","Class1");
78    hold on;
79    contour(i1,i2,p0_joint,"blue");
80    contour(i1,i2,p1_joint,"red");
81    hold off;
82
```

Plot:

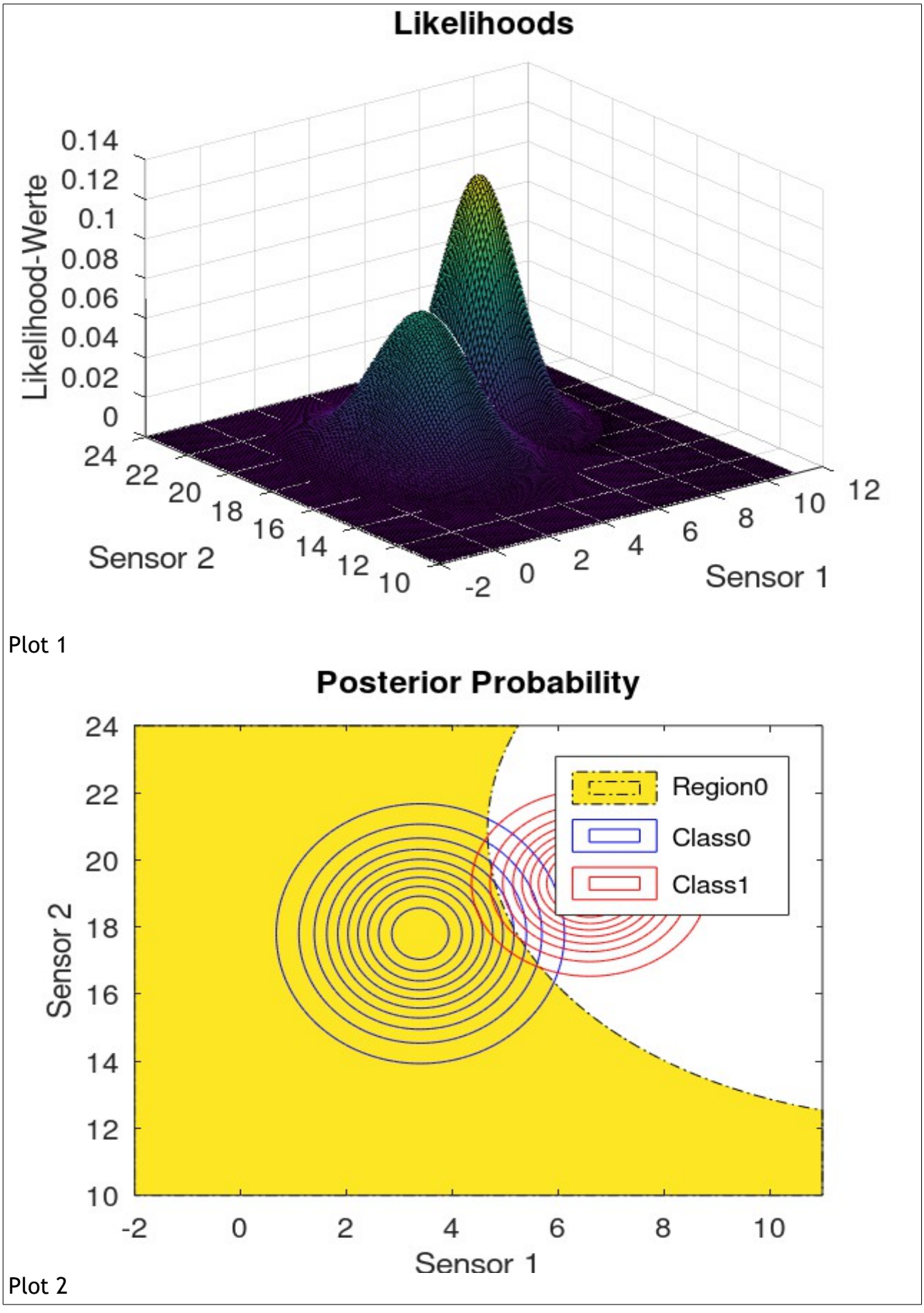


Part c:

Code:

```
%Part C
%Calculating the posterior probabilities of class 0
p_x_0 = p0_joint;
p_x_1 = p1_joint;

p_0_x = p_x_0 ./(p_x_0 + p_x_1);


%Plotting the graphs
figure(1);

surf(i1,i2,p0_joint);
hold on;
surf(i1,i2,p1_joint);
title('Likelihoods');
xlabel ('Sensor 1');
ylabel ('Sensor 2');
zlabel ('Likelihood-Werte');
hold off;

figure (2);
contour(i1, i2, p_0_x, [0.5 0.5], 'k-.');
contourf(i1, i2, p_0_x, [0.5 0.5], 'k-.');
hold on
contour(i1,i2,p0_joint,"blue");
contour(i1,i2,p1_joint,"red");
title ('Posterior Probability');
xlabel ('Sensor 1');
ylabel ('Sensor 2');
legend("Region0","Class0","Class1");
hold off;

```

Plots:



Plot 1

Plot 2