Prof. Dr. Hauke Schramm
Department of Computer Science
Kiel University

Pattern Recognition

Exercises

Practice Sheet 5

Exercise L-5.1 (Decision boundaries)

In this exercise, we will model data sets of *two classes* using two Gaussian distributions as likelihoods and illustrate the decision boundary. This boundary marks the line (in case of 2D variables) where the posterior probability of class $\boldsymbol{\omega_0}$ equals the posterior probability of class $\boldsymbol{\omega_1}$, i.e. $P(\omega_0 \mid \boldsymbol{x}) = P(\omega_1 \mid \boldsymbol{x})$.

Load the Octave file twoClasses.mat using the load function. This stores two variables into your workspace: `patterns` and `targets`. The `patterns` variable contains the 2D observation vectors (one 2D vector per column) and the `targets` variable the corresponding class labels. Note that the observations are sorted with respect to their class, i.e. the first 2000 observations are from class ω_0 and the remaining 2000 observations are from class ω_1.

Examples:

Show the first 5 observations of class ω_0:

```
>> patterns(:,1:5)

ans =

    3.6642    4.9162    3.9818    5.3453    2.5478
   20.0939   15.3323   16.6113   15.4816   18.4773
```

Show the first 5 class labels of class ω_1

```
>> targets(2001:2005)

ans =

     1     1     1     1     1
```

Remark: In this special case, where the data is sorted with respect to the class label, a label variable is not really required. However, note that usually this is not the case.

Prof. Dr. Hauke Schramm
Department of Computer Science
Kiel University

Part a.

Organize the data in a *cell array* with appropriate labeling of the

- classes as 'Class0' and 'Class1'
- variables as 'Sensor1' and 'Sensor2'
- observations as 'Obs_1' to 'Obs_2000'

Example:

Show the cell array content of the first two observations of the first sensor for class one.

```
>> cloud1(1:3, 2, 1)

ans = Sensor1
ans =  3.6642
ans =  4.9162
```

For the next exercise it is helpful to generate four help variables storing the two sensor signals of class one and two.

Class[1,2]_sensor[1,2] : Sensor signal of class [1,2] and sensor [1,2]

Part b.

We next aim at modeling the data of each class with an individual normal distribution (see the lecture for details). These two models can be later used to compute the likelihoods $p(\boldsymbol{x} \mid \omega_0)$ and $p(\boldsymbol{x} \mid \omega_1)$ for each class and each observation vector $\boldsymbol{x}$.

First, we must decide whether to use a Gaussian model with independent or dependent components. How would you decide on that question? After selecting an appropriate modeling, we must estimate the model parameters (mean vector, variance vector or covariance matrix) using the same techniques as in Practice Sheet 4.

Plot the two point distributions of the classes in different colors (use the command `legend` to label your classes) and superimpose the contours of the two Gaussian distributions using the function `contour`. Assure that you achieve an appropriate modeling of each class before continuing.

Part c.

Generate 3D plots (function `surf`) of (Figure 1) the two likelihoods and (Figure 2) the posterior probability $P(\omega_0 \mid \boldsymbol{x})$ and illustrate the decision regions with different colors. See the lecture for further details about plotting 3D graphs.

Prof. Dr. Hauke Schramm
Department of Computer Science
Kiel University

Hints

Decision boundary:

In classification theory, the ***Bayes Decision Rule*** (see the lecture) instructs us to decide for the class ω with the highest posterior probability $P(\omega \mid \boldsymbol{x})$. Considering only two classes, the ***Bayes Decision Boundary*** is a region (in 2D case: line) where $P(\omega_0 \mid \boldsymbol{x}) = P(\omega_1 \mid \boldsymbol{x}) = 0.5$. This boundary separates our 2D feature space into one region where we decide for class ω_0 and another region where we decide for class ω_1. Figure 1 illustrates this concept for a simple example.

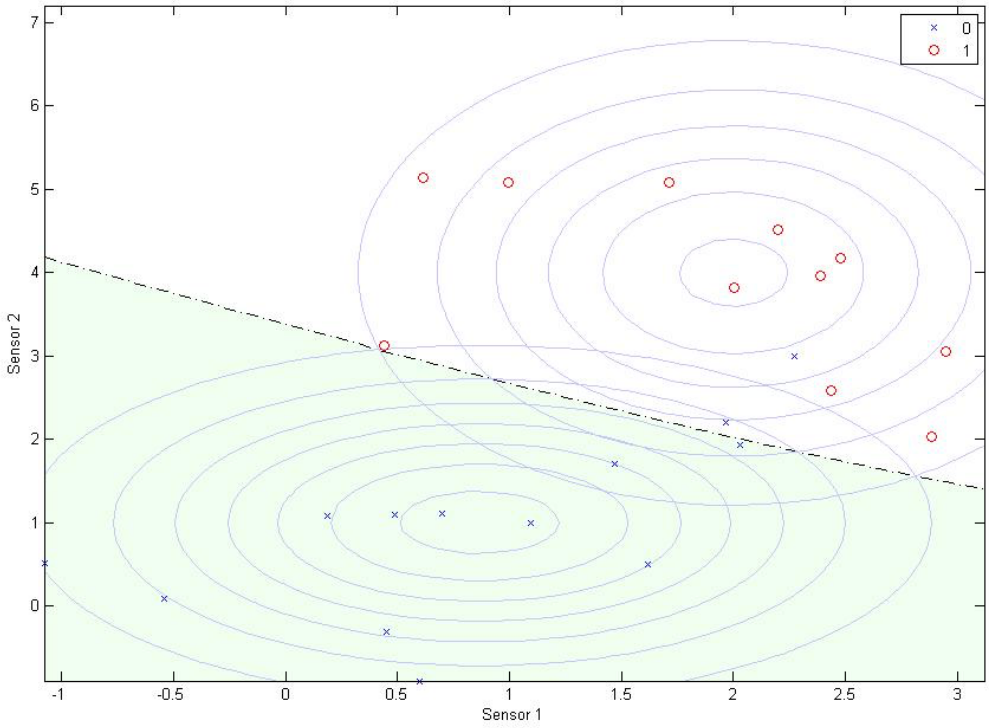

Figure 1. Example point distribution of two classes with Gaussian model contours and decision boundary.

In order to determine the decision boundary we must compute the posterior probabilities $P(\omega \mid \boldsymbol{x})$ for both classes using ***Bayes rule*** (see lecture):

$$P(y \mid x) = \frac{P(x \mid y) \cdot P(y)}{\sum_{y \in Y} P(x \mid y) \cdot P(y)}$$

Following this rule, the posterior probabilities of class 0 and 1 become:

$$P(\omega_0 \mid x) = \frac{p(x|\omega_0) \cdot P(\omega_0)}{p(x|\omega_0) \cdot P(\omega_0) + p(x|\omega_1) \cdot P(\omega_1)}$$

$$P(\omega_1 \mid x) = \frac{p(x|\omega_1) \cdot P(\omega_1)}{p(x|\omega_0) \cdot P(\omega_0) + p(x|\omega_1) \cdot P(\omega_1)}$$

With the two Gaussians computed above, we already have models for the likelihoods $p(\boldsymbol{x} \mid \omega_0)$ and $p(\boldsymbol{x} \mid \omega_1)$. Therefore, in order to compute the required posteriors, we only need the prior probabilities $P(\omega_0)$ and $P(\omega_1)$. These two (scalar) values can be estimated from the number of observations of class ω_0 and class ω_1 in the input data.

Use the given equation to compute the posterior $P(\omega_1 \mid \boldsymbol{x})$ (denoted as p_1_x in your code) and illustrate the decision boundary by using the function contour as follows:

```
>> contour(pts_x, pts_y, p_1_x, [0.5 0.5], 'k-.');
```

This plots a line at points where $P(\omega_1 \mid \boldsymbol{x}) = 0.5$. (Note that at these points $P(\omega_0 \mid \boldsymbol{x})$ must be 0.5 as well.)

You may as well mark the decision *regions* using the function `contourf`:

```
>> contourf(pts_x, pts_y, p_1_x, [0.5 0.5], 'k-.');
```

Remark: The function `contourf` may cover existing points in the figure. To avoid this, you must first use `contourf` and then add the points and Gaussian contours (use the command '`hold on`' to join multiple figures).