Abgabedokument Exercise 3

Einfuehrung in die Mustererkennung 186.840 WS 2013

David Pfahler, Matthias Gusenbauer, Matthias Vigele

1126287, 1125577, 1126171

January 19, 2014

Part I. The first Report

1. Perceptron

The following section describes how we implemented the online perceptron training algorithm to get the linear discriminant function and evaluates it.

1.1. Application

The `matlab`-function `[ w ] = perco( X,t,maxEpoches )` calculates the perceptron weight vectors with the online perceptron training algorithm. The Listing 1.1 shows this algorithm.

```
Initialise $w, \gamma$
**repeat**
    **for** $i = 1 : N$ **do**
        **if** $w^T(x_i y_i) \leq 0$(misclassified $i$th pattern) **then**
            $w \leftarrow w + \gamma x_i y_i$
        **end if**
    **end for**
**until** all patterns correctly classified
```

The `matlab`-function `plot_perco_results(w,X,classLabel,titleName)` plots the linear discriminant function and the data from the training set labeled by the consigned classes.

Instead of letting the algorithm run until all patterns are correctly classified (see Listing 1.1) it stops after a given number of iterations. Because this number is not fixed by the exercise

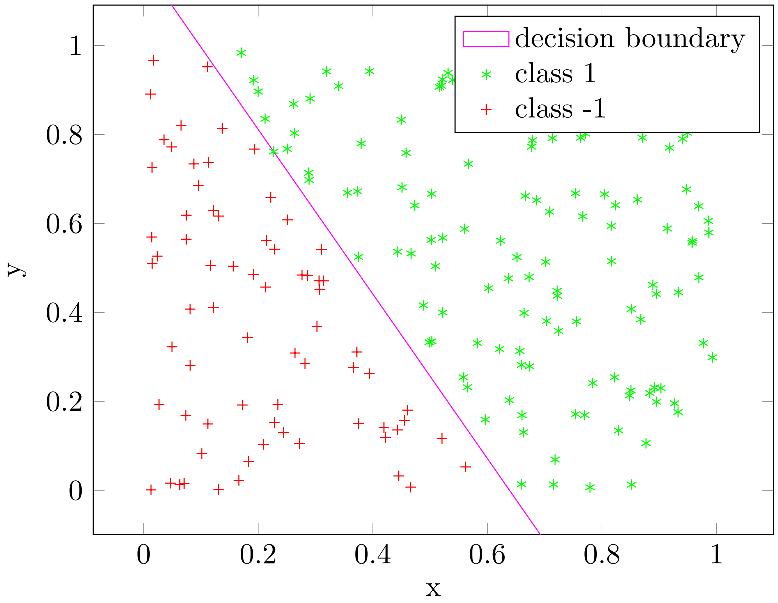

Figure 1: Decision boundary for target data set 1 with 10 epochs

description, we used different values and compared the results in the next section.

First we calculate the lineal discriminant function for the data provided by the Exercise (`perceptrondata.dat`) with the two different target data sets and plot the results (See Figure 1 and Figure 2 for 10 epochs and Figure 3 and Figure 4 for 100 epochs)

The calculation of the perceptron weight vector (see Equation 1) for the first target data set terminated after 9 epoches, because the whole training data set was classified correctly.

$$w_1 = \begin{bmatrix} -4.0000 \\ 6.2628 \\ 3.3863 \end{bmatrix} \tag{1}$$

The calculation of the perceptron weight vector (see Equation 2) for the second target data set did not terminate after 100 epoches.

$$w_2 = \begin{bmatrix} -2.0000 \\ 2.9205 \\ 1.5864 \end{bmatrix} \tag{2}$$

The linear function for the separation of the two classes for the first target data is calculated in Equation 3or for a more general case see Equation 4

$$f(x, y) = w_1 + w_2 x + w_3 y \tag{3}$$

$$f(\vec{w}, \vec{y}) = \vec{w} \cdot \vec{y} \tag{4}$$

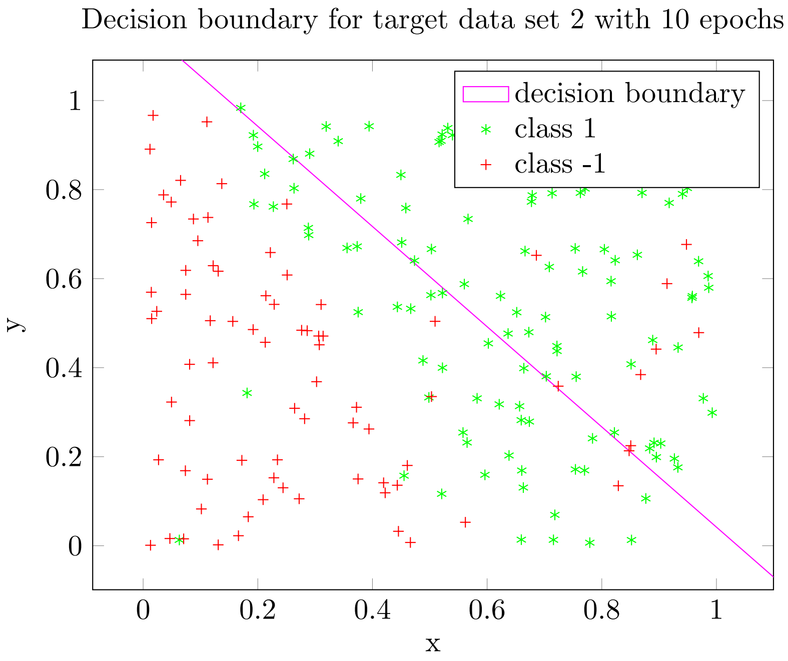

Figure 2

1.2. Discussion

This section discusses the results and the convergence of the algorithm that was introduced above. Furthermore it discusses the OR-, AND- and XOR-problems.

1.2.1. OR-, AND- and XOR-Problem

We used the same algorithm and evaluation techniques like above to solve this problems. But instead of the `perceptrondata.dat` we used our own data to demonstrate and visualize this problem. The data X is shown in the Equation 5. Each column represents an feature vector in homogeneous coordinates. Each problem is represented with a class label vector. Equation 6 shows the vectors for the different problems.

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{5}$$

$$AND = \begin{bmatrix} -1 & -1 & -1 & 1 \end{bmatrix} \qquad OR = \begin{bmatrix} -1 & 1 & 1 & 1 \end{bmatrix} \qquad XOR = \begin{bmatrix} -1 & 1 & 1 & -1 \end{bmatrix} \tag{6}$$

Figure 5 shows the results for the AND-Problem and like the OR-Problem Figure 6 the classification result is 100%. The only problem that cannot get solved with the perceptron is the XOR-Problem (see Figure 7)

There is no linear function that could seperate the 2 classes at the XOR-Problem with a classification rate of 100%. So the calculated weight from the algorithm are all zero. (see Equation 7)

$$wXOR = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

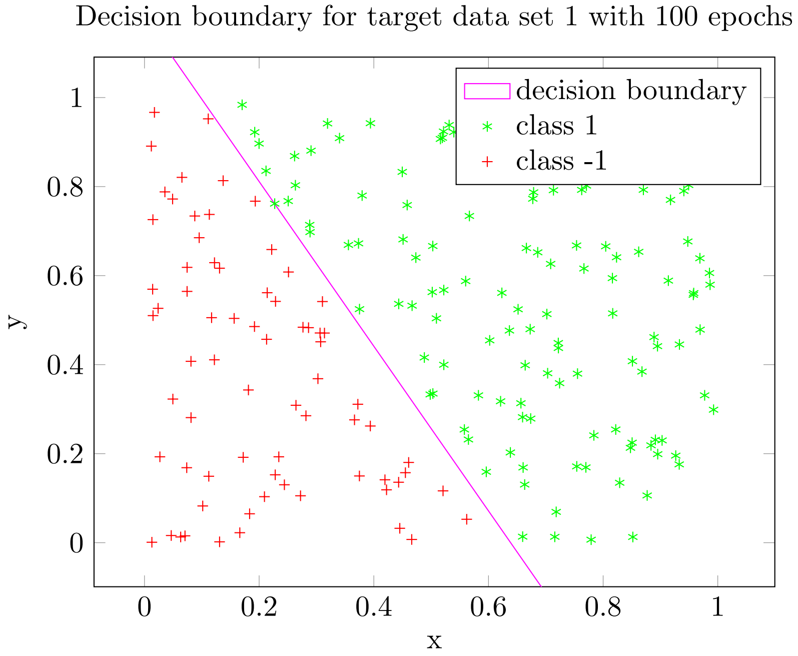

Figure 3

1.2.2. Results & Convergence

The algorithm with the input from the first data set needs only 9 epochs to find a linear discriminant function that has a classification rate of 100% (see Figure 1). But the second data set has some values that could get compared with the XOR-Problem, because they are in the dynamic range of the other class. (see Figure 2)

To demonstrate the convergence of the perceptron it is only useful to look at the data set with the XOR-Problem. Figure 8 shows the development of the classification rate to the epochs. This does not converge (even with the maximum of epoches set to 1000 and more).

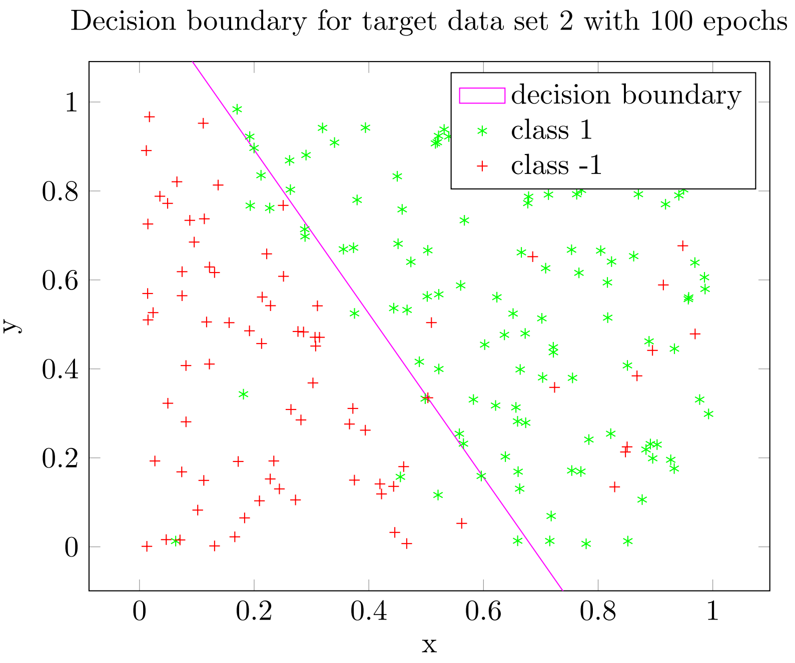

Figure 4

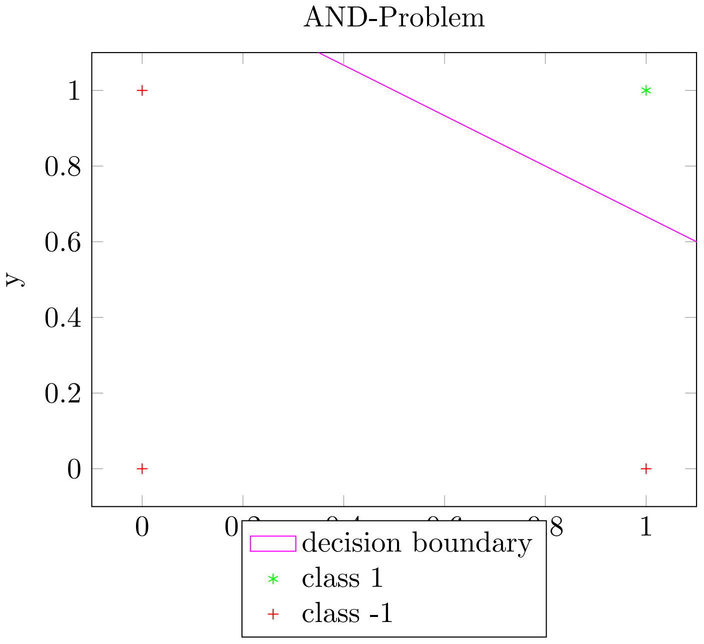

Figure 5

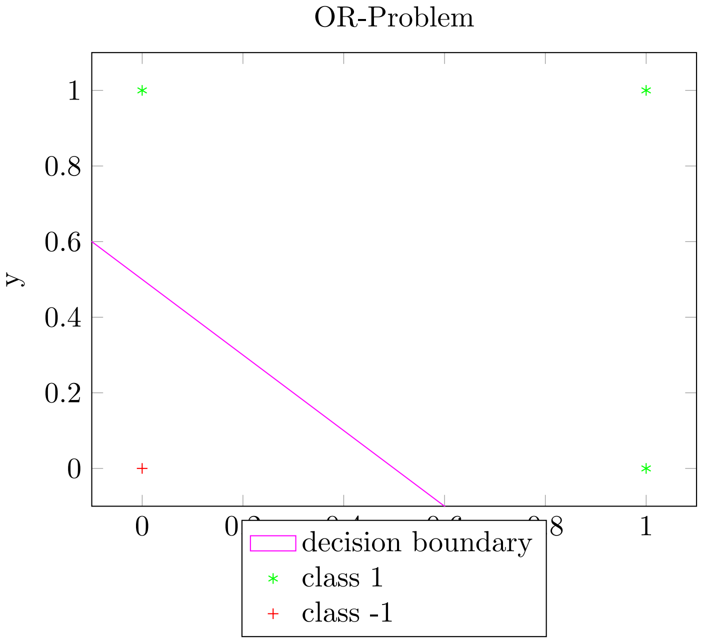

Figure 6

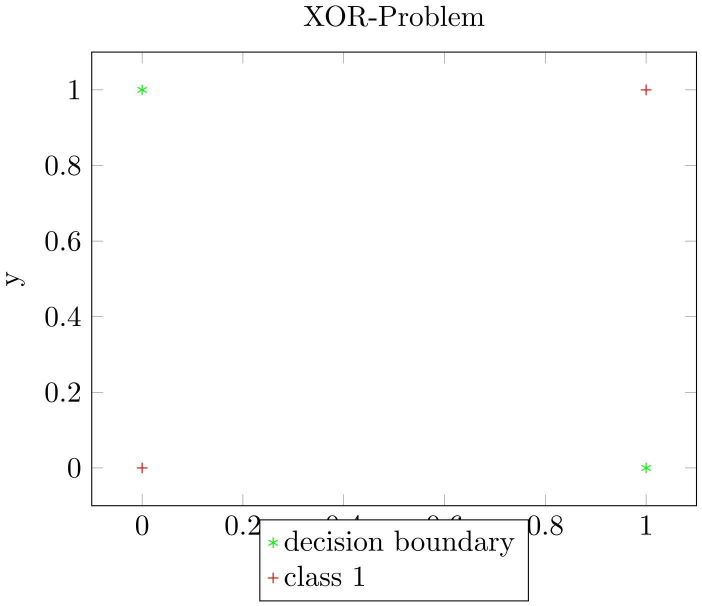

Figure 7

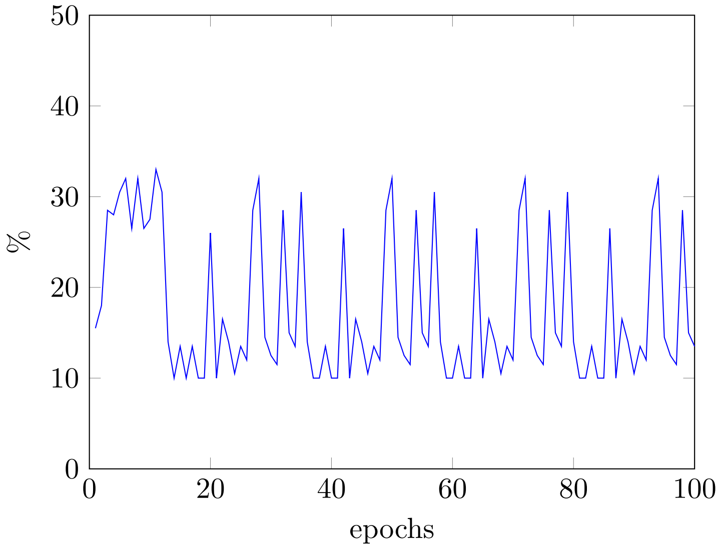

Figure 8: the development of the classification rate to the epochs of the data set 2