Machine Learning Report

Daniel Zügner (6832915) Konstantin Kobs (6414943)
Sazan Hoti (6832974)

Exercise 2

Model

For this exercise we used a Batch Gradient Descent approach in contrast to the Stochastic Gradient Descent from last time. The resulting parameters were:

$$\begin{aligned} \alpha &= 0.05 \\ \theta_{\text{bias}} &= 7.7618 \\ \theta_1 &= -1.0793 \\ \theta_2 &= 3.1896 \end{aligned}$$

The model is therefore defined by the function

$$f(x) = g(7.7618 - 1.0793x_1 + 3.1896x_2)$$

where $g(z)$ is the sigmoid function.

Graph

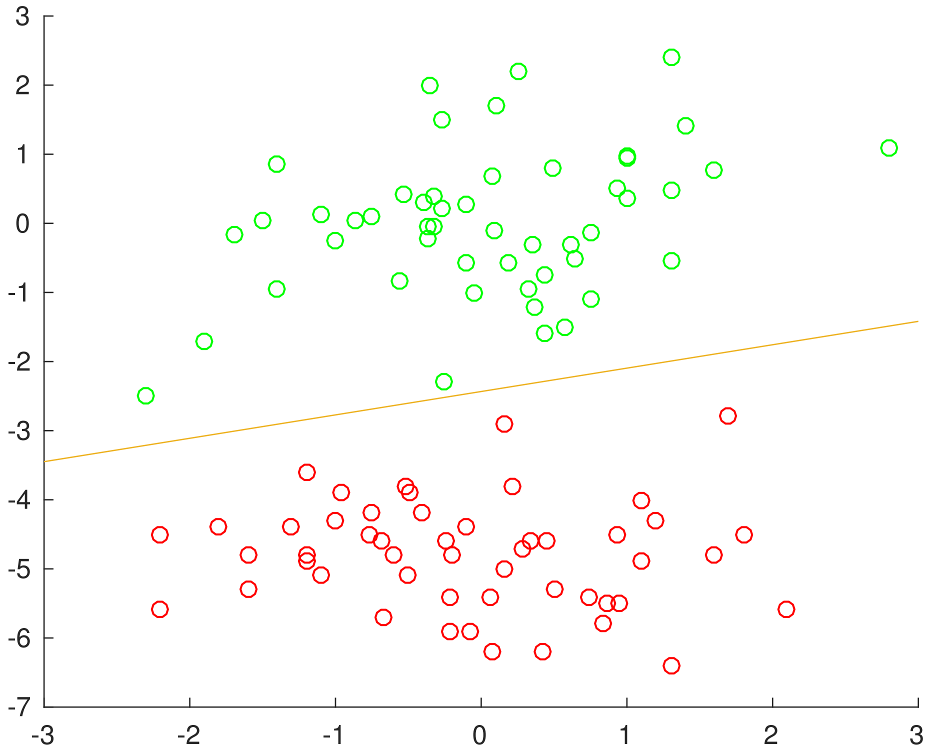

Figure 1: The plotted decision boundary with separately colored data points.

Code

readData.m

```
data = dlmread('data.txt',' ');

alpha = 0.05;

theta = logisticRegression(data, alpha);

positive = data(find(data(:,3) == 1),:);
negative = data(find(data(:,3) == 0),:);
figure
scatter(positive(:,1), positive(:,2), [], 'green'); hold on;
scatter(negative(:,1), negative(:,2), [], 'red'); hold on;
display(theta);

fplot(@(x) -1*(theta(1)+ theta(2)*x)/theta(3), [-3 3]); hold off;
```

logisticRegression.m

```
function theta = logisticRegression(data, alpha)

    theta = rand([3,1]) * 0.02 - 0.01;
    [rows, columns] = size(data);
    vec = ones([rows, 1]);

    x_mat = [vec data(:,1:2)];
    y = data(:,3);

    threshold = 0.1;
    errorDecrease = 1;
    iteration = 0;
    errors = [];

    while(errorDecrease / alpha > threshold)

        % compute matrix product of X and Theta
        % After that we can compute the result of the logistic function
        intermediate = x_mat * theta;
        h = logsig(intermediate); % logsig is the sigmoid function
        diff = (y - h);
        error = diff' * diff;
        update = alpha * diff' * x_mat;
        theta = theta + update';

        if iteration > 0
            errorDecrease = ((errors(end) - error)/errors(end));
        end
        iteration = iteration + 1;
        errors = [errors error];
    end

    display(['Converged after ' num2str(iteration) ' iterations.'])
```

end