Assignment 4

Computational Intelligence, SS2020

Team Members		
Last name	First name	Matriculation Number
Blöcher	Christian	01573246
Bürgener	Max	01531577

1 Linear SVM

1.1 Plots

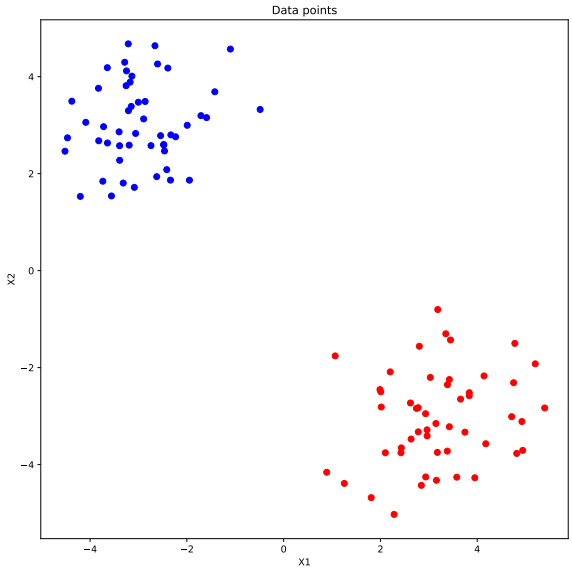
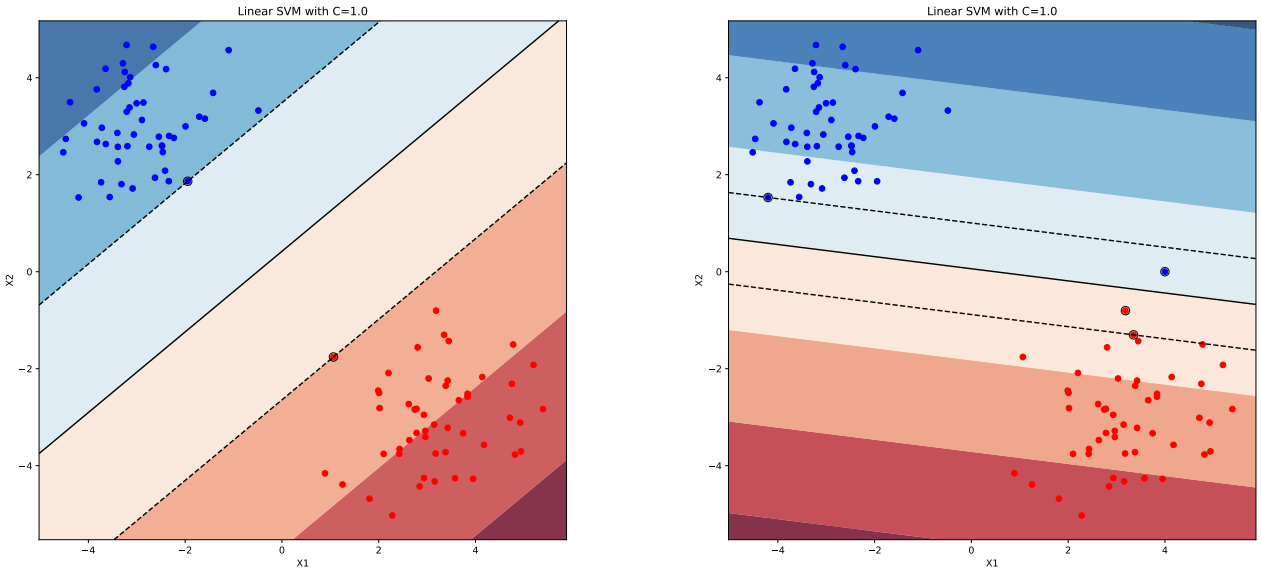


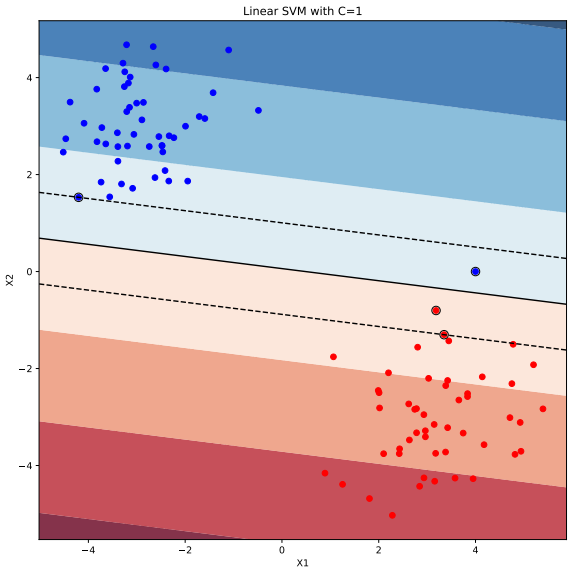
Figure 1: Dataset



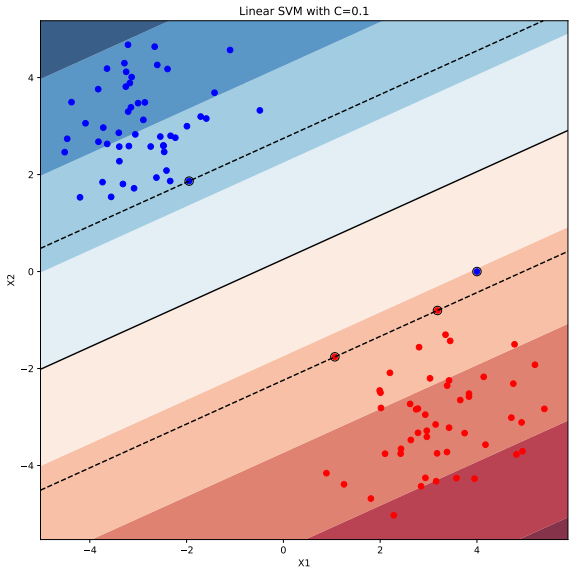
(a) Dataset

(b) Dataset with additional data point

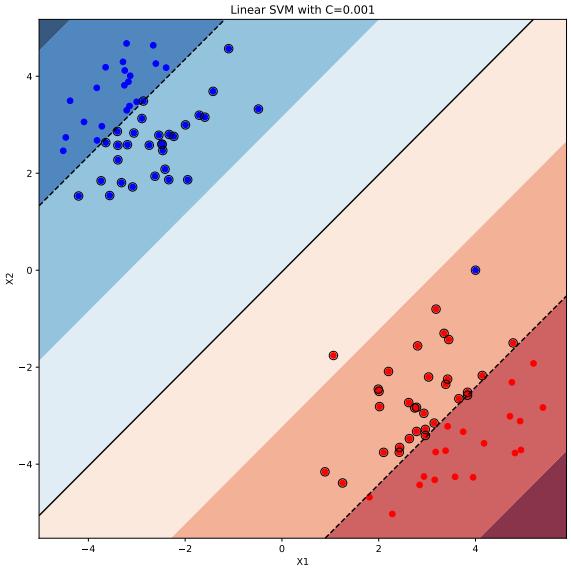
Figure 2: Classification of the dataset using SVM



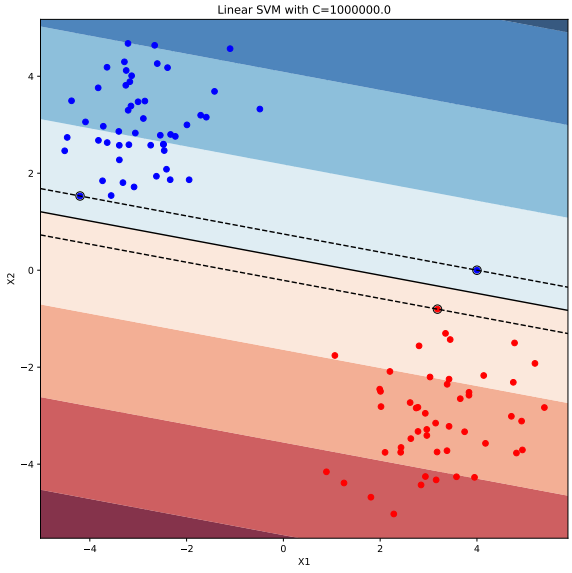
(a) $C = 1$



(b) $C = 0.1$



(c) $C = 0.001$



(d) $C = 1000000$

Figure 3: SVM Classification using different C parameters

1.2 How and why did the decision boundary change when the new point was added?

The margin is defined as the perpendicular distance between the decision boundary and the closest data point. Since the newly added data point is located in the margin the algorithm adjusts its solution by choosing the option with the smallest generalization error. Apparently this solution includes more support vectors (data points within and on the margin boundary).

1.3 The inverse regularization parameter C

- C controls the penalization of data points within the margin. It is scaling data samples so that samples can be misclassified or lie within the margin boundaries. This is helpful since not all problems can be clearly separated.
- Figure 3 shows the classification using different values for C. The margin increases with decreasing values for C. As a result the classification is showing more support vectors and also misclassifies the additional data point since these points are less penalized. A very large value for C leads to the strict constraint that the margin borders are denoting the closest data point to the decision boundary. Therefore the margin is very small.

2 Nonlinear (kernel) SVM

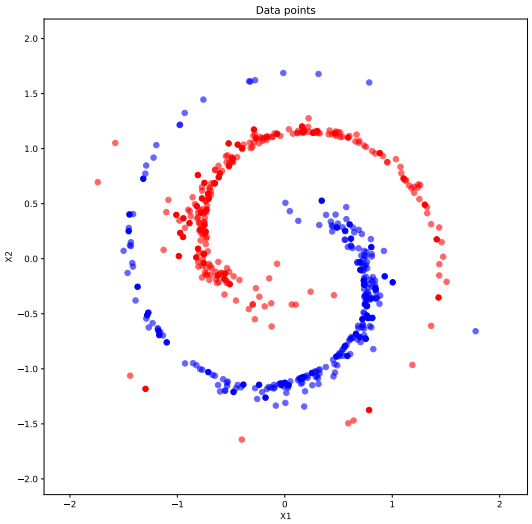


Figure 4: Nonlinear binary classification problem.

2.1 Linear kernel

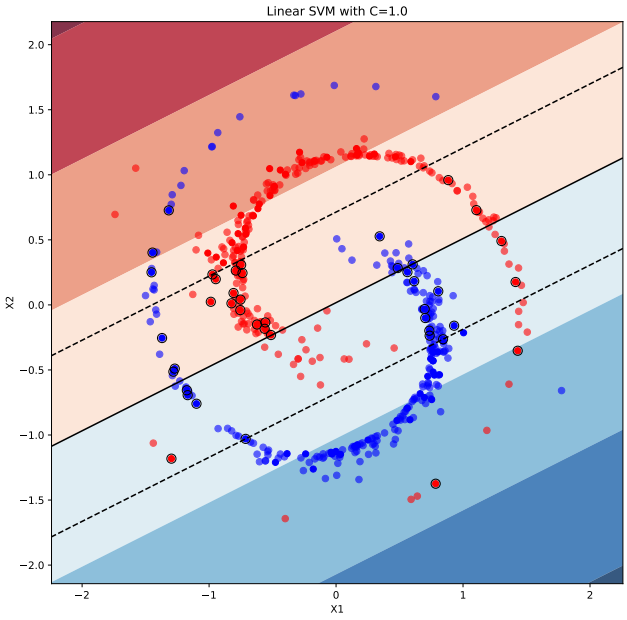


Figure 5: Decision boundary obtained by linear kernel.

Test score
81.25%

Table 1: Test score obtained by linear kernel.

2.2 Polynomial kernel

Best degree	Test score
8	95.75%

Table 2: Highest test score obtained by polynomial kernel.

2.3 RBF kernel

Best γ	Test score
1.85	94.25%

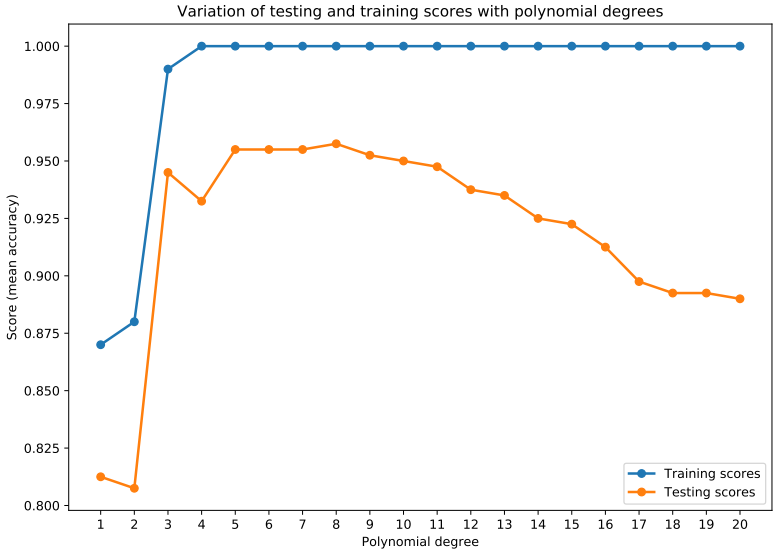
Table 3: Highest test score obtained by RBF kernel.

2.4 Comparison of results obtained by each of the three kernels

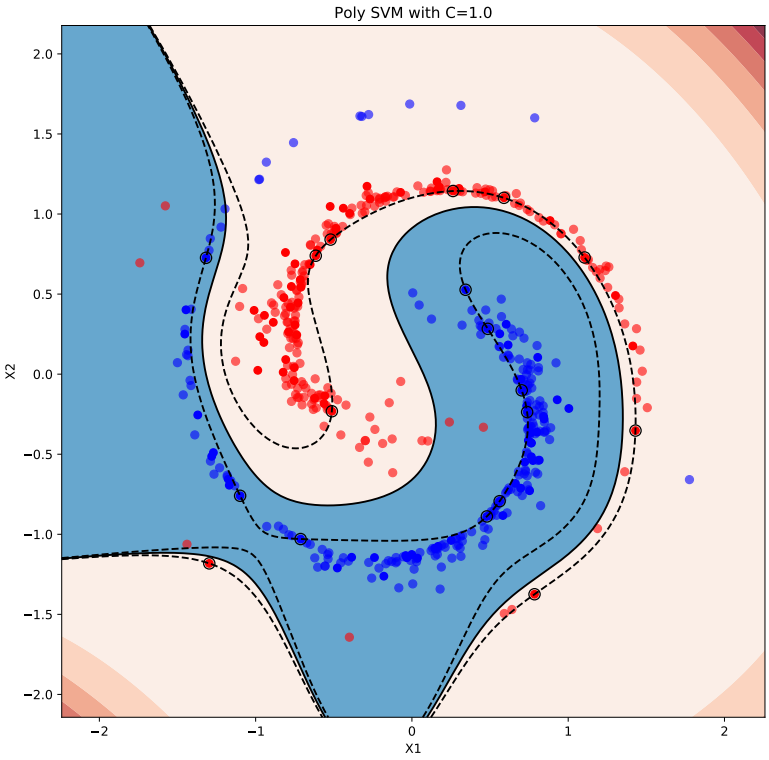
- While the linear kernel performs expectedly badly on nonlinearly separable data (s. figure 5 and table 1), testing scores obtained by polynomial kernels with degrees $3 \dots 13$ and RBF kernels with $\gamma \geq 1$ are much higher (s. figures 6a and 7a). A polynomial kernel with degree 8 gives the best results (s. table 2), in figure 6b one can see that the support vectors that are used to produce the decision boundary exclusively lie on the margin boundaries (as opposed to the best RBF result in figure 7b and table 3).

- Although the decision boundary obtained by the linear kernel is the most basic of the three, it takes the most support vectors, which are either misclassified or lie within the margin (s. figure 5). The polynomial kernel's decision boundary is more complex but relies on the fewest support vectors (exclusively on the margin borders, s. figure 6b), because this approach fits the given data best. The most complex decision boundary (with insular patterns, s. figure 7b) is generated by the RBF kernel with a medium number of support vectors. Rather than only increasing with the complexity of the decision boundary, the number of needed support vectors also decreases the better the kernel fits the data.

- The RBF kernel generalizes best, as both its training and testing scores continue to rise with increasing γ, without any overfitting (s. figure 7a). Overfitting occurs when using the polynomial kernel with high degrees ($>> 8$, s. figure 6a)).

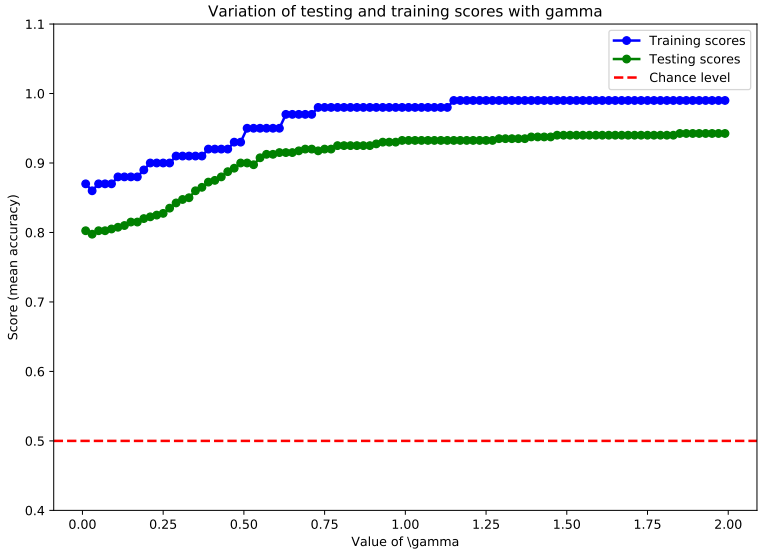


(a) Score for varying polynomial degree.

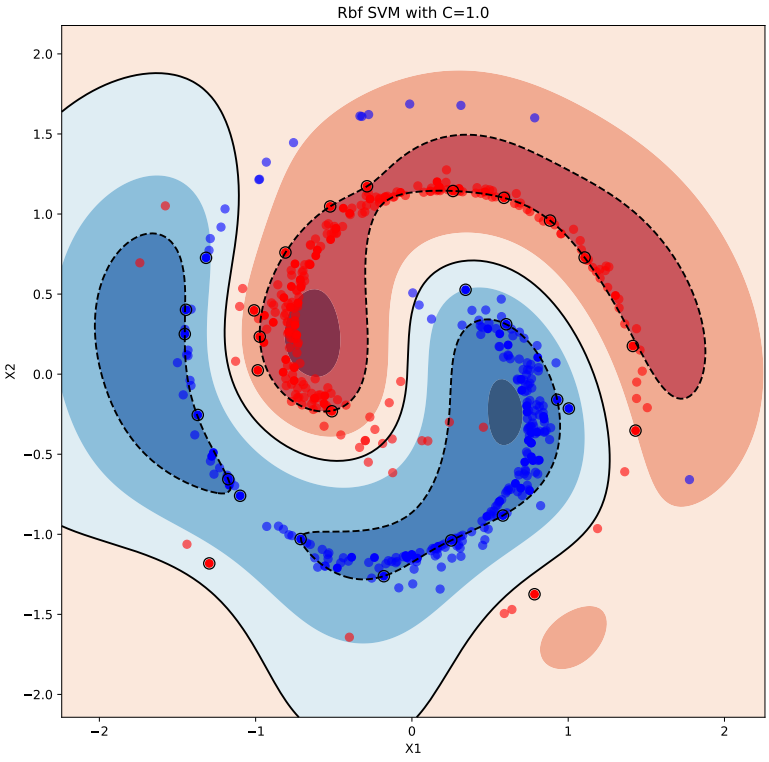


(b) Decision boundary for polynomial degree 8.

Figure 6: Results obtained by polynomial kernel.



(a) Score for varying γ.



(b) Decision boundary for $\gamma = 1.85$.

Figure 7: Results obtained by RBF kernel.

3 Multiclass classification

3.1 'One vs. Rest' vs. 'One vs. All' algorithms

SVMs are fundamentally binary classifiers. One can use different approaches to solve problems with K classes.
The 'One vs. Rest' algorithm compares the to be determined class (positive) with the rest of the dataset (negative). During the scoring phase the algorithm determines the probability for each class of a sample using K binary classifiers.
'One vs. All' creates $K(K-1)/2$ classifiers, which means that we use all possible pairs of classes. Each of the binary classifiers predicts one class. For classification the class with the highest number of predictions will be chosen. In comparison to the 'One vs. Rest approach' this algorithm requires much more time to get trained and to classify.

3.2 Classification using linear and RBF kernels

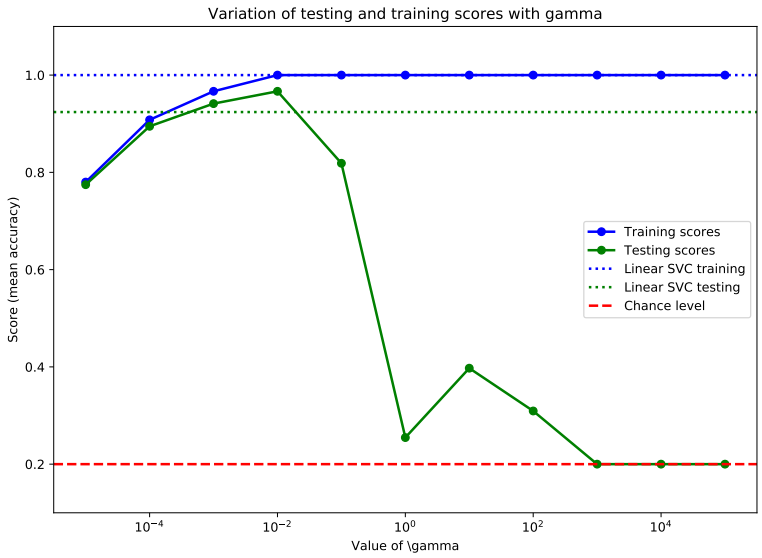


Figure 8: Scores using linear and RBF kernels on MNIST Dataset

- Figure 8 shows that the linear kernel performs very well on MNIST image classification. Using $\gamma = 0.1$ the RBF classification performs slightly better than the linear kernel. For higher γ values the classification overfits and the testing scores descend quickly.
- The linear kernel performs well, because the sample features which are generated from the MNIST pictures must be very well separable with linear kernels in higher dimensions.

3.3 Confusion matrix

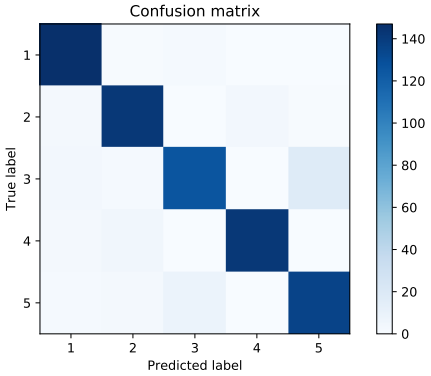


Figure 9: Confusion matrix

Labels	1	2	3	4	5
1	147	1	2	0	0
2	3	142	0	4	1
3	4	2	126	0	18
4	3	5	0	142	0
5	2	3	9	0	136

Table 4: Confusion matrix

The rows of the confusion matrix correspond to the true classes and the columns to the predicted classes. Class three is the most misclassified class with a score of 84%.

3.4 Misclassified images

Figure 10: Misclassified images

Figure 10 shows the column of predicted class three. The images are showing the real numbers, which have been misclassified as class three. Apparently the algorithms has problems to distinguish threes and fives. Probably because the lower part of these numbers are very similar. The reason for both falsely determined ones could be that they are untypical ones. The first one is very bold and the other is a bit rotated.

4 SVM with Gradient Descent

4.1 Gradient Descent

$\mathbf{w}_{opt}$	b_{opt}	Final cost
$\begin{bmatrix} -0.2195 \\ 0.2084 \end{bmatrix}$	0.0025	0.0569

Figure 11: Results of GD.

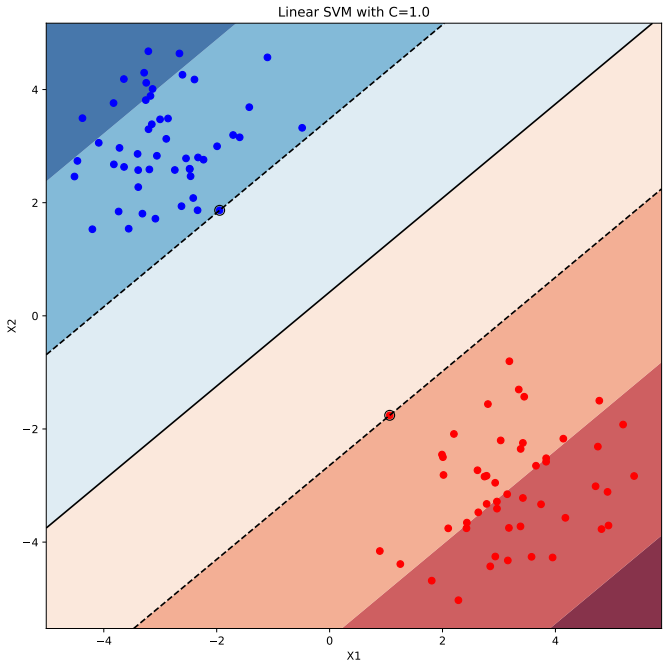
With the chosen number of iterations 10 and step size $\eta = 0.1$ the GD algorithm approximates the global minimum reasonably well and finds the optimal parameters $\mathbf{w}_{opt}$ and b_{opt} (s. figure 11).

4.2 Results of classification

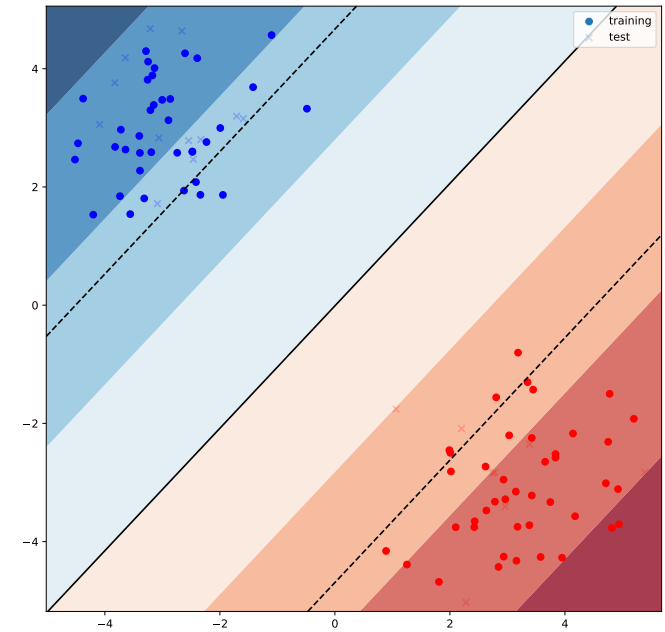
Both SVM implementations from task 1 and 4 classify the dataset with an accuracy of 100% (s. figure 12). The first implementation's decision boundary relies on only two support vectors on the margin. The margin of the SVM implemented in task 4 is larger with samples lying within the margin as well. One reason for this could be the random separation of the data in training and testing sets. In figure 12b the closest red point to the decision boundary is part of the testing set and can therefore not be used to generate the decision boundary.

4.3 Drawbacks of Primal Formulation

In Dual Formulation the dual coefficients α_i for all samples i are known, i.e. one only needs to sum over the used support vectors to classify new samples. In Primal Formulation one needs to use the whole training set, leading to larger computational cost. Because the max-term of the Primal Formulation is non-smooth and therefore not differentiable at $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$, subgradients with a conditional function $\mathbb{I}_i$ are needed. Then efficient GD solvers can be used.
Additionally - as opposed to Dual Formulation - the Kernel Trick can not be used, making the use of kernels more difficult.



(a) Task 1.



(b) Task 4.

Figure 12: Decision functions obtained by SVMs in tasks 1 and 4.