DD2421: MACHINE LEARNING LAB2

Author
Adel Abdelsamed

Contents

1 Solutions

Assignment 1: Move the clusters around and change their sizes to make it easier or harder for the classifier to find a decent boundary. Pay attention to when the optimizer (minimize function) is not able to find a solution at all.

Solution Assignment 1:
In this assignment the linear kernel

$$\kappa(x, y) = x^T \cdot y \tag{1}$$

is utilized. This results in linear separating hyperplanes. The optimizer is guaranteed to find a solution under the assumption of linear separability of the data.

In the following, two examples are presented. In Figure 1, the dataset is clearly linearly separable and hence the decision boundary and the optimal classifier margin are computed and shown. In Figure 2, the data is not linearly separable and hence the optimizer fails to find a solution.

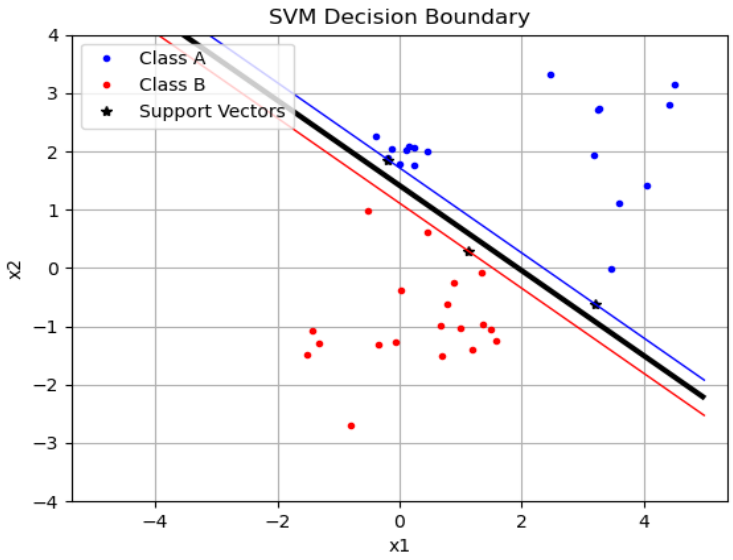


Figure 1: Linearly separable dataset with optimal classifier margin.

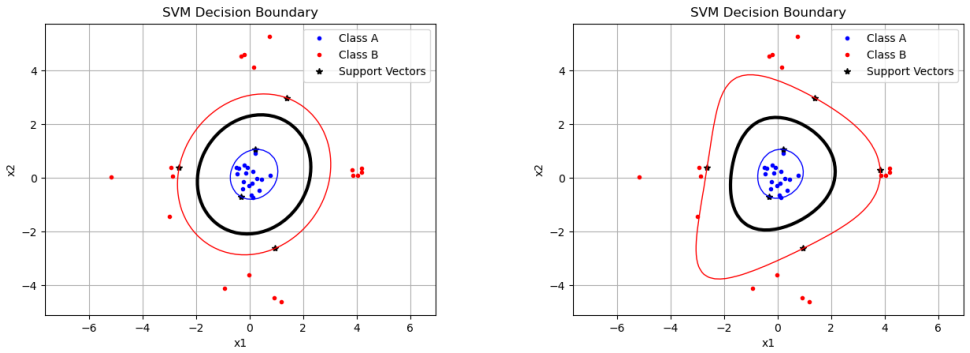
Assignment 2: Implement the two non-linear kernels. You should be able to classify very hard data sets with these.

Using the kernel trick, the data set can be efficiently transformed into high-dimensional spaces, while limiting the computations to the scalar product between two input data points.

Hence, the polynomial and radial basis functions were implemented

$$\begin{aligned}\kappa(x,y) &= (x^T \cdot y + 1)^p \\ \kappa(x,y) &= e^{-\frac{||x-y||^2}{2\sigma^2}},\end{aligned} \tag{2}$$

where p and σ are hyperparamters of the chosen kernel. Utilizing these kernels, nonlinear decision boundaries can be trained as shown in Figure 2.



(a) Linearly non-separable dataset with the polynomial kernel with $p = 2$.

(b) Linearly non-separable dataset with the RBF kernel with $\sigma = 2$.

Figure 2: Datasets with nonlinear decision boundaries.

Assignment 3: The nonlinear kernels have parameters; explore how they influence the decision boundary. Reason about this in terms of the bias variance trade-off.

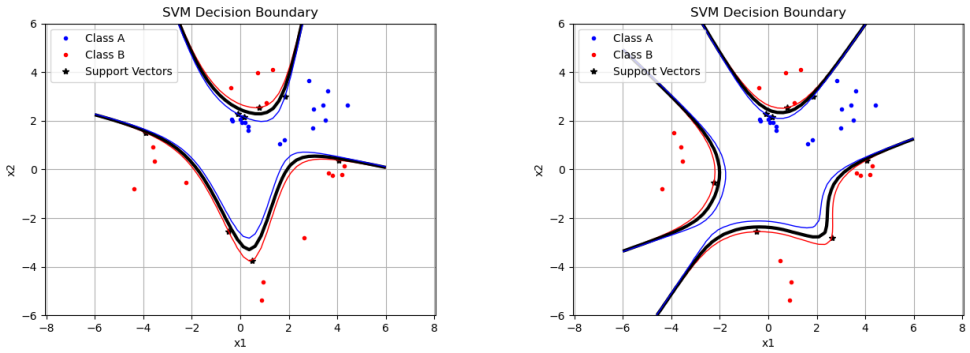
In Figure 3, the influence of the polynomial order is investigated. Increasing the polynomial order results in a more complex decision boundary. The more complex the decision boundary, the more overly specialized the decision boundary w.r.t the training data. Hence, a higher order polynomial increases variance as the learned model will not generalize well onto unseen data. Lower order polynomials will result in simpler models, hence introducing bias.

For the RBF Kernel, the influence of varying the hyperparameter σ is shown in Figure 4. It is clear that as σ increases the influence of data points further apart is significantly higher. Hence, a bigger σ results in a decision boundary which is less sensitive to noise and is less likely to overfit. Decreasing the hyperparameter will result in a more complex decision boundary as only nearby points will be considered.

Assignment 4: Explore the role of the slack parameter C. What happens for very large/small values?

The problem formulation after introducing slack variable is

$$\begin{aligned}&\min_{\boldsymbol{w},\xi,b} \quad ||\boldsymbol{w}|| + C\sum_i \xi_i \\ &s.t. \;\; t_i(\boldsymbol{w}\cdot\phi(\boldsymbol{x}) - b) \geq 1 - \xi_i \quad ,\forall i.\end{aligned} \tag{3}$$



(a) Linearly non-separable dataset with the polynomial kernel with $p = 3$.

(b) Linearly non-separable dataset with the polynomial kernel with $p = 5$.

Figure 3: Influence of polynomial order p.

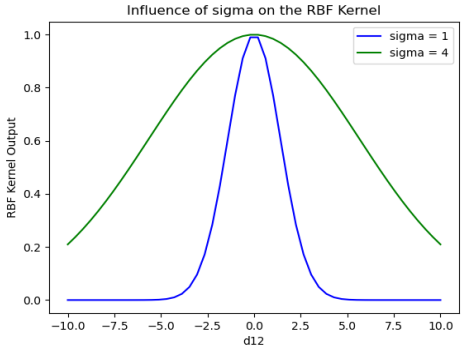
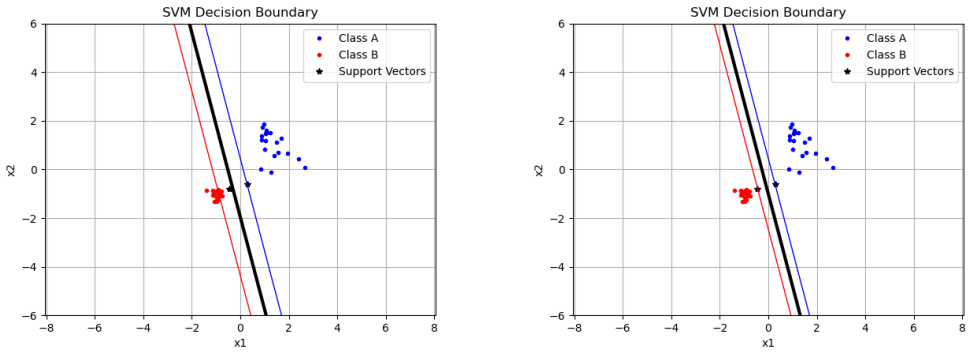


Figure 4: Influence of σ on the RBF Kernel.

The parameter C controls the size of the slack variables and can be seen as a regularizer to our problem. A higher penalty on the slack variables will result in small slack variables, which means the data points will try to remain outside the margin classifier. A lower penalty will result in bigger slack variables ξ_i allowing more data points to lie within the classifier margin. This effect can be seen in Figure 5. Using very big C values will hence tend to the same result as in the case of no slack, while using very small C values will result in a larger number of support vectors and hence increasing computation and training time.

Assignment 5: Imagine that you are given data that is not easily separable. When should you opt for more slack rather than going for a more complex model (kernel) and vice versa?

This involves a trade-off between model complexity and regularization. If inherent pattern of the underlying data set is clearly nonlinear, then a more complex model should be used. However, as the model complexity is increased, the model is more likely to overfit. Hence proper tuning of the hyperparameters is important to optimize the model complexity. Furthermore, if the data is noisy or has overlapping data points, then introducing a slack variable is essential. This allows for more classification error on the training set but makes



(a) Linearly separable dataset with the linear kernel and $C = 2$.

(b) Linearly separable dataset with the linear kernel and $C = 4$.

Figure 5: Influence of the slack variable.

the model more robust to noise and improves the generalization to noisy data. Hence, we introduce some bias by simplyfing our model.

It is important to strike the right balance between model complexity and introducing slack. As introducing slack for a model assumption that is inherently not accurate will not improve classification accuracy.