ML Labb 2

Magnus Axén
Oscar de Brito Lingman

March 7, 2022

Intro

In this lab we are asked to design a Support Vector Machine (SVM) in python to later on use 3 different kernels to cluster certain randomized points.

We shall first present the problem formulation and what we actually want to program and why.

Question 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.

$$K(\vec{x}, \vec{y}) = \vec{x}^T * \vec{y}$$

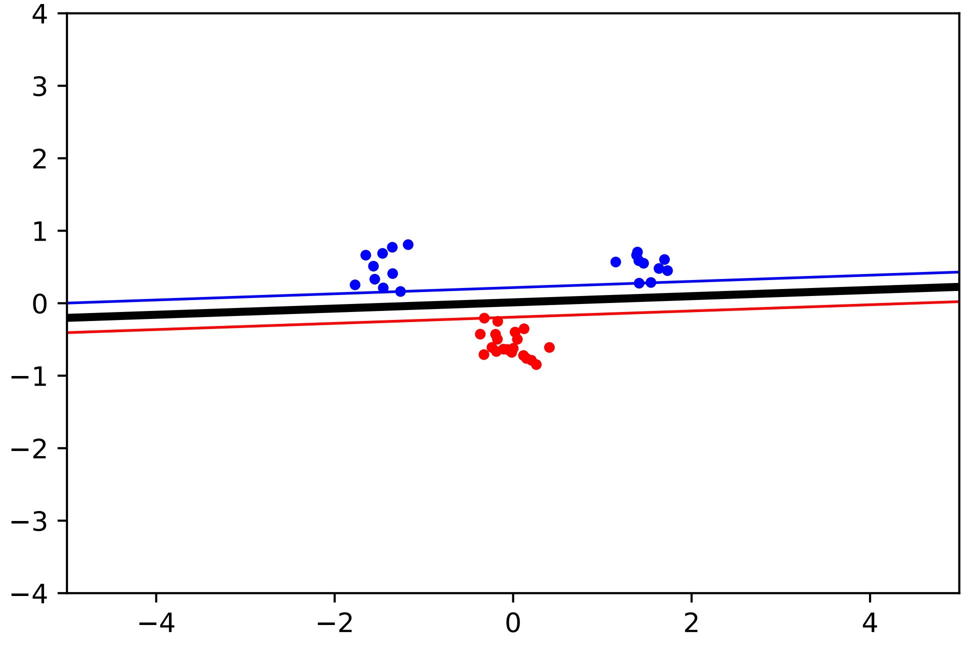

Figure 1: Linear Kernel, Generic Dataset

By increasing the spread-factor of the three clusters, it becomes harder for the classifier to find the boundary. A factor of 0.4 result in the minimize function not finding a solution. The figure below has a factor of 0.3.

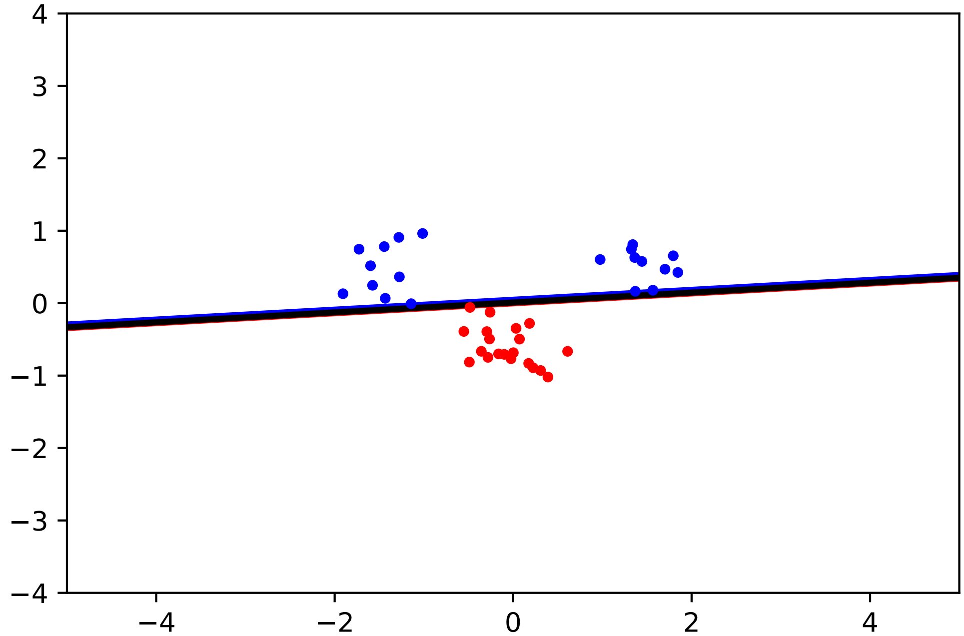

Figure 2: Linear Kernel, 0.3 spread-factor

Finally we tried moving the clusters around (the blue ones) to increase difficulty for the classifier.

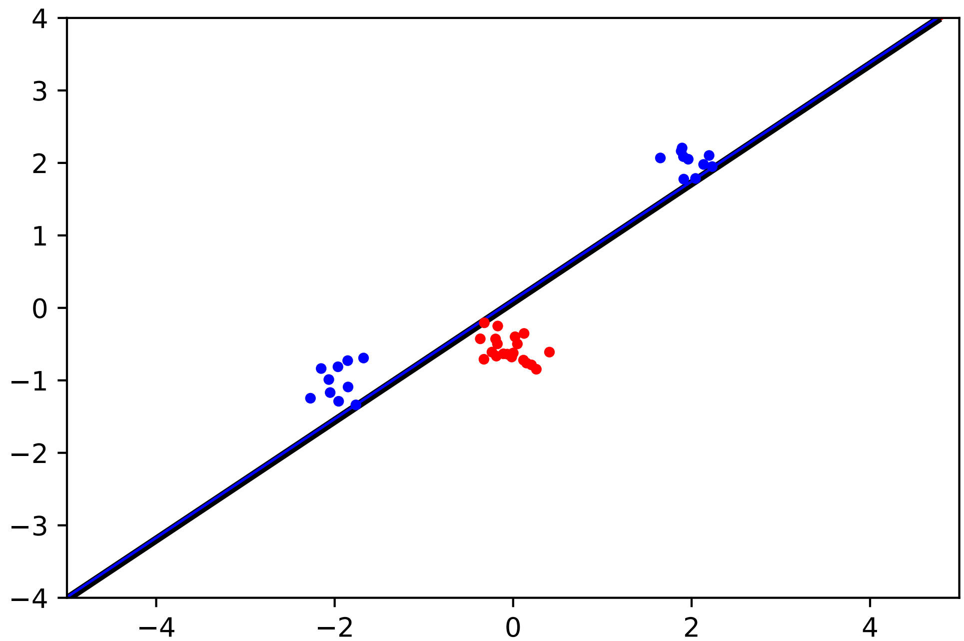

Figure 3: Linear Kernel, clusters at new coordinates.

Question 2

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

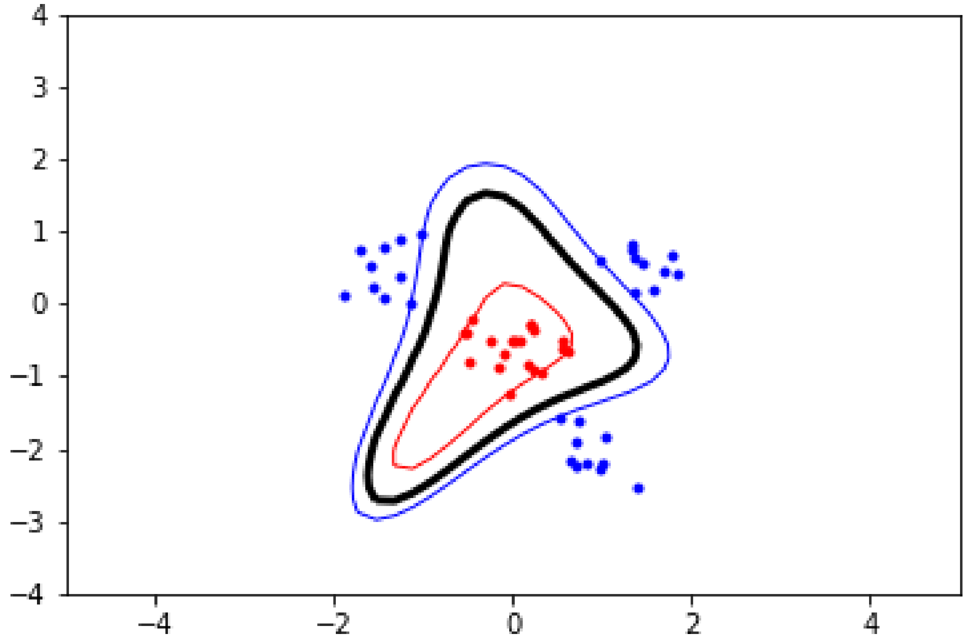

Figure 4: Polynomial Kernel, hard dataset, p = 4

$$K(\vec{x}, \vec{y}) = (\vec{x}^T * \vec{y} + 1)^P$$

p = 1 (Linear)
p = 2 (Quadratic shapes)
p = 3, 4, 5 ("Complex shapes")

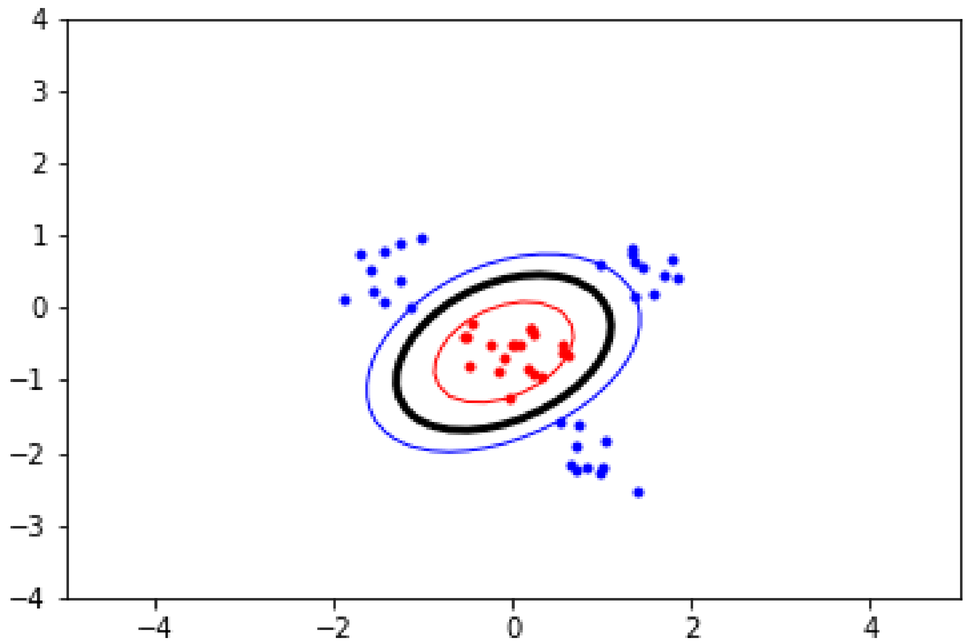

Figure 5: RBF Kernel, hard dataset, $\sigma = 3$

$$K(\vec{x}, \vec{y}) = e^{-\frac{||\vec{x}-\vec{y}||^2}{2*\sigma^2}}$$

Question 3

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

Polynomial

As we increase the polynomial degree there is the risk of over fitting the data, say we have a lot of points then perhaps a 15 degree polynomial will be fitted as it increases the margin the best, but its quite improbable that it will actually be a 15 degree polynomial. Thus as we increase the p we increase the bias and the variance will be lowered for the points.

Show GIF

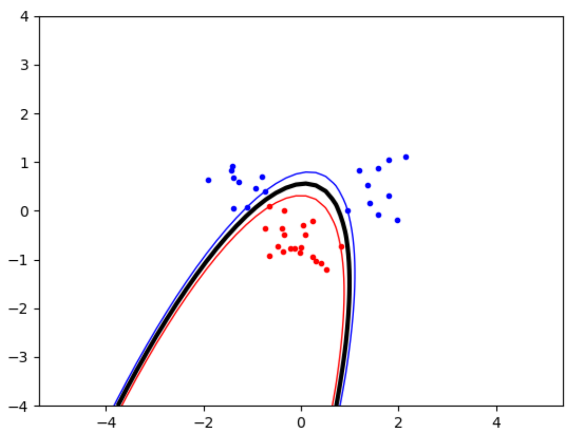

Figure 6: Second degree

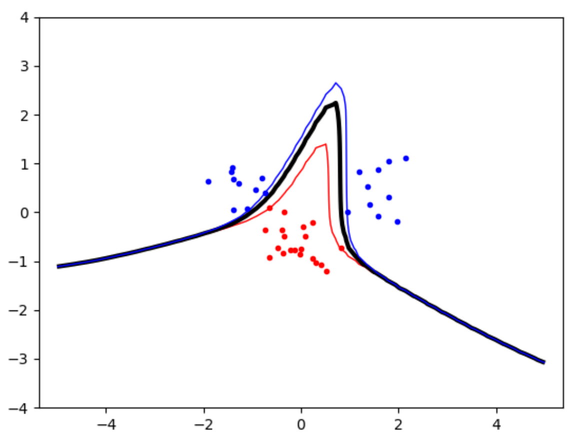

Figure 7: Ninth degree

Radial Basis Function (RBF)

First of all we note if sigma is low the bias will be high and variance low, and if the sigma is high the bias will be low and variance high. Since, given a small sigma the "reach" of the Kernel becomes larger whilst if we have a high sigma the reach is smaller and the values will vary less.

Show GIF

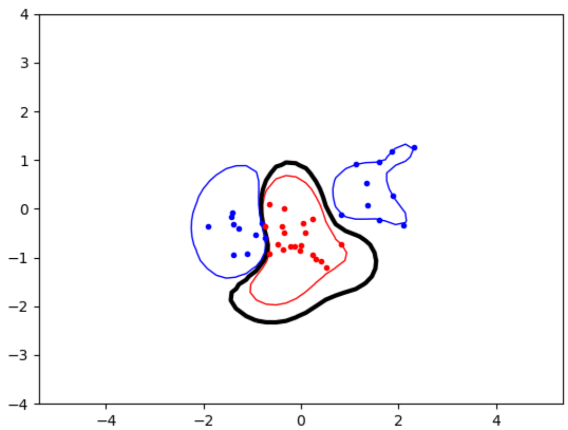

Figure 8: Sigma=0.5

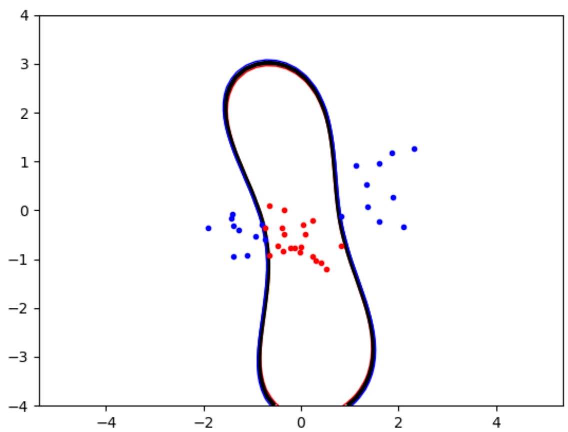

Figure 9: Sigma=2

Question 4

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

The slack variable allows for misclassification in the algorithm to more encourage ”generality” in the model. For a difficult problem where there is a lot of noise in the data requires a slack variable for a boundary to be found. Can be see in the last figure that when $C = 100$ its approcahing ideal if there was no noice, $y = 0$.

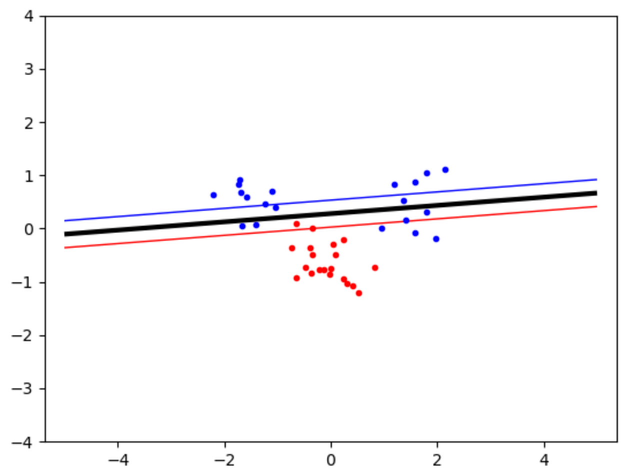

Figure 10: C=none

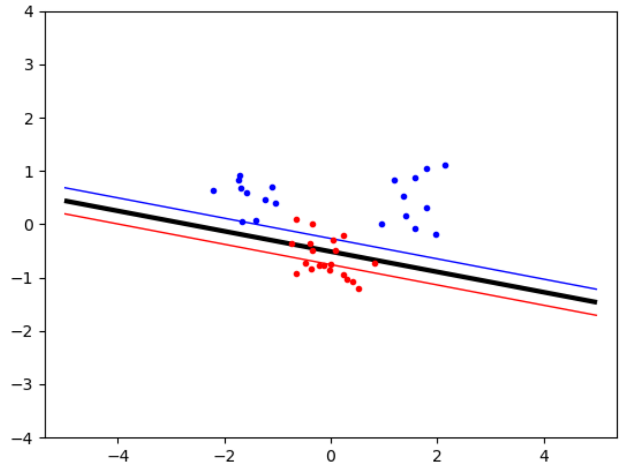

Figure 11: C=10

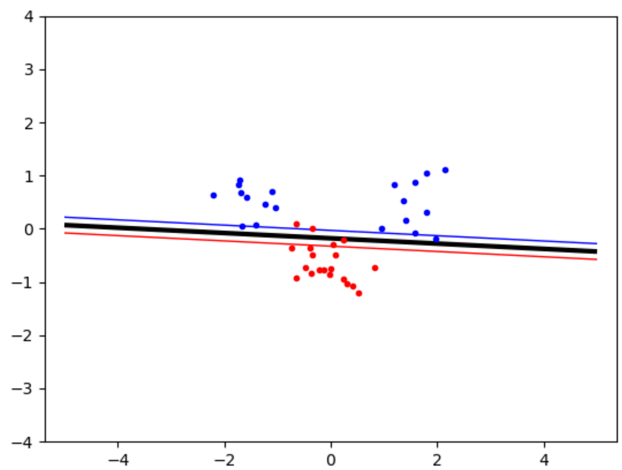

Figure 12: C=100

Question 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?

When the model is clearly separable but noisy more slack should be used as we can then separate them but need to have slack to allow for misclassification. However when the data is not clearly separable but noisy we prefer a more complex model.

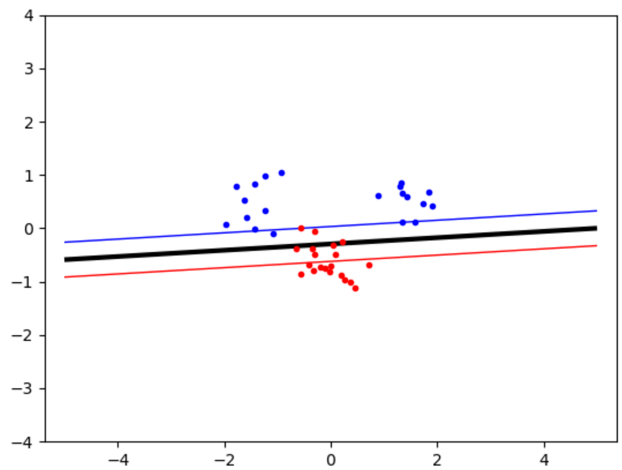

Figure 13: Linearly separable, noisy, but with slack

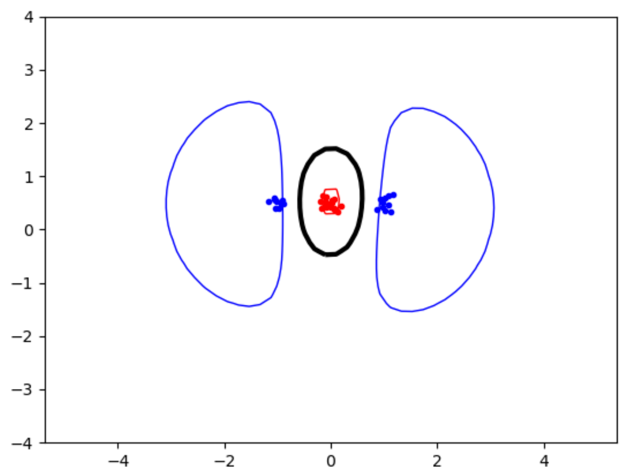

Figure 14: Not linearly seperable, but not too noisy, prefer complex

However if we are in a situation in which it is not linearly separable then, we can in a noisy situation use more slack to constitute for that. And if we are in a situation in which its low noise we can simply just use a more complex model.

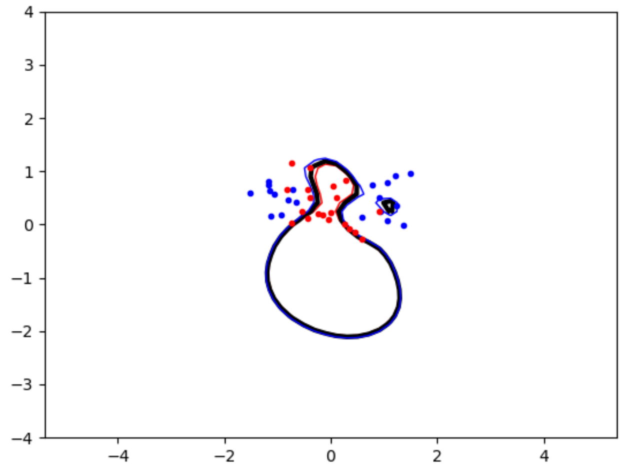

Figure 15: Without Slack

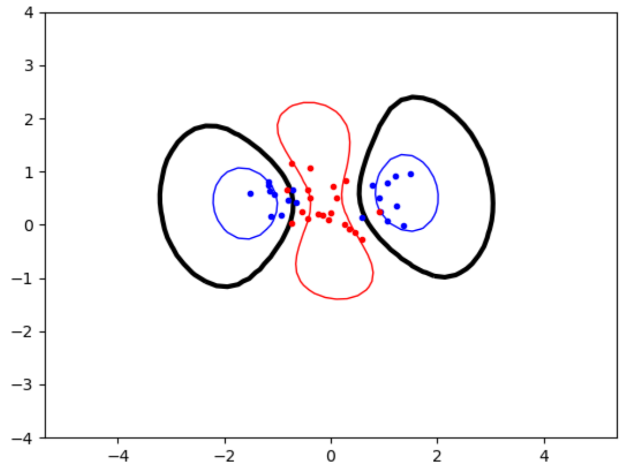

Figure 16: With slack