12.10.2021

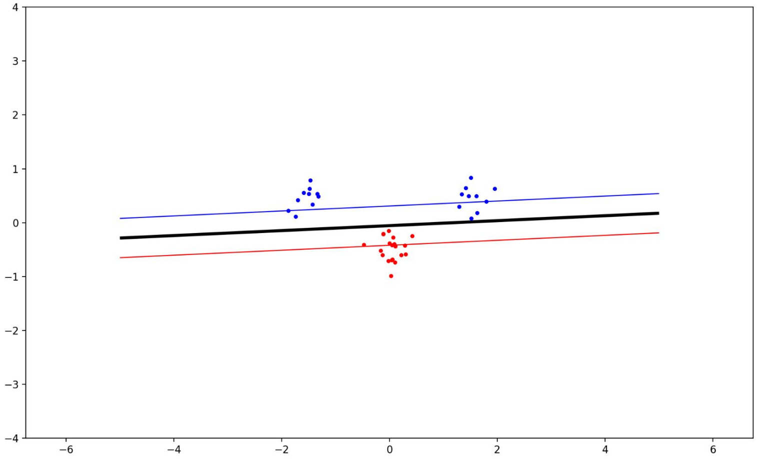
DD2421 - Machine Learning

Lab 2: Support Vector Machines

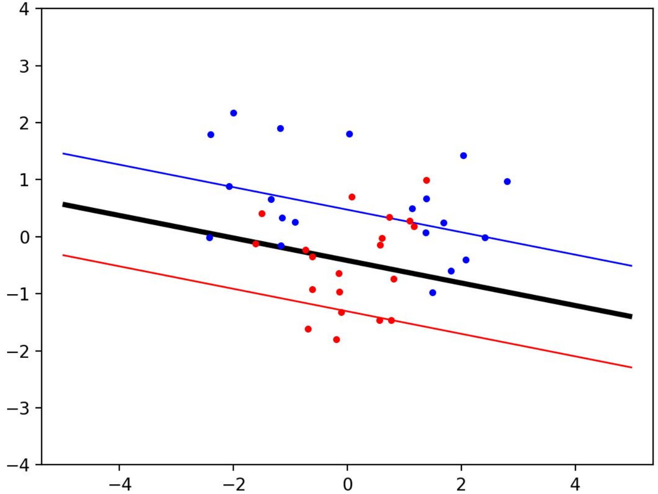
Pablo Laso, Antonia Schroff

*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.*

Standard deviation = 0.2



Standard deviation = 0.8



At some point, it's not possible to have linear decision boundaries anymore. At this point, we might need **non-linear Kernels**.

As soon as the dataset is not separable anymore with a linear boundary, the minimizer can't solve the problem anymore.

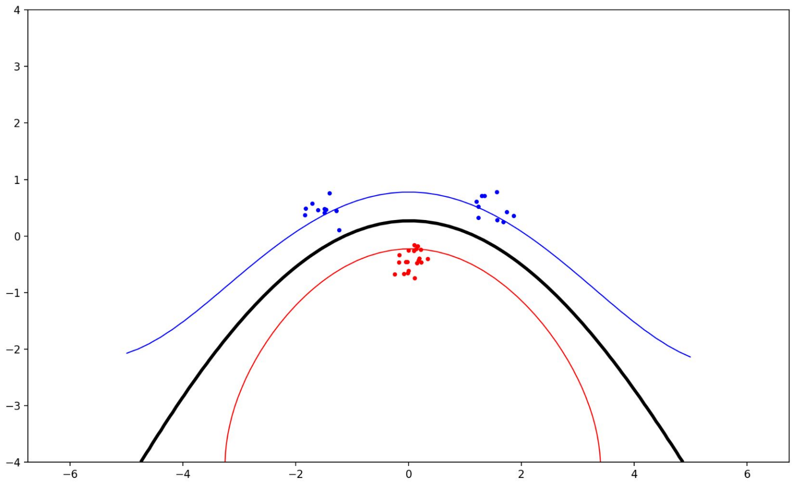
Implement the two ***non-linear kernels****. You should be able to classify very hard data sets with these.*

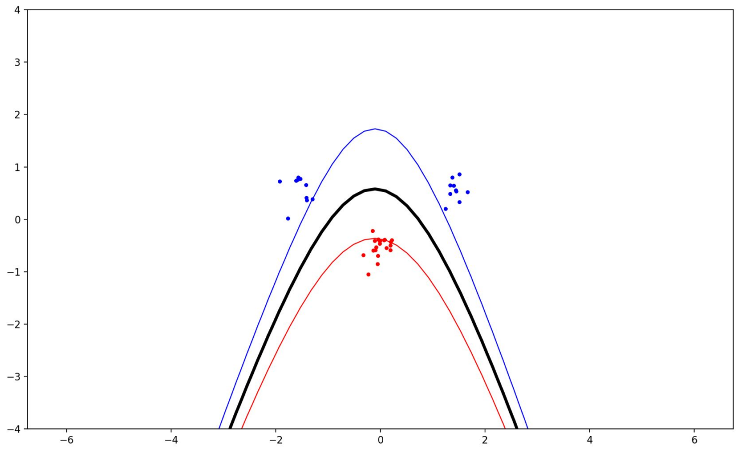
RBL Kernel

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

Poly Kernel

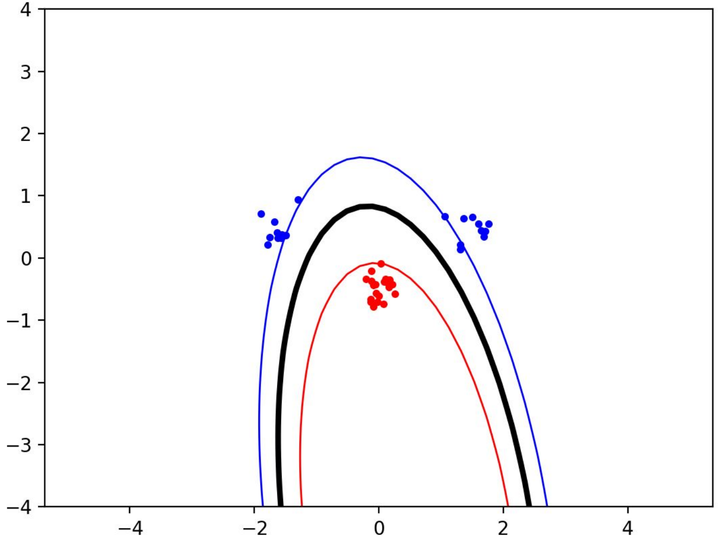
$$\mathcal{K}(\vec{x}, \vec{y}) = (\vec{x}^T \cdot \vec{y} + 1)^p$$



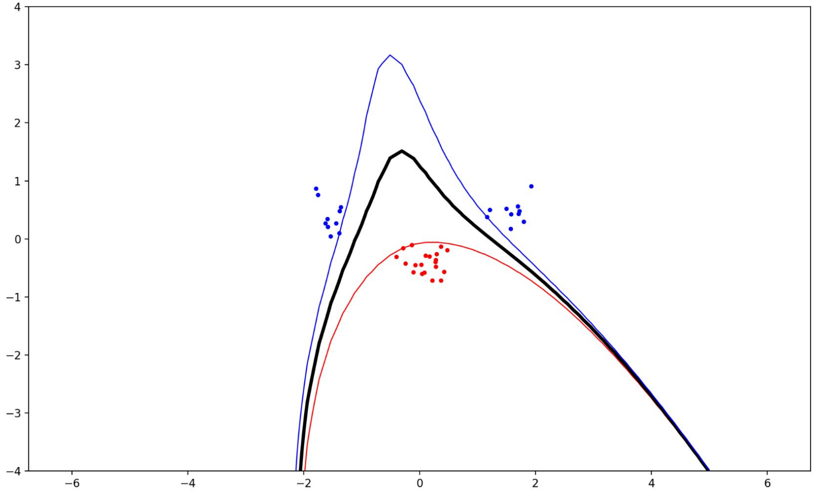


Polynomial *using different degrees*

Polynomial Kernel (**Degree 2**)



Polynomial Kernel (**Degree 4**)

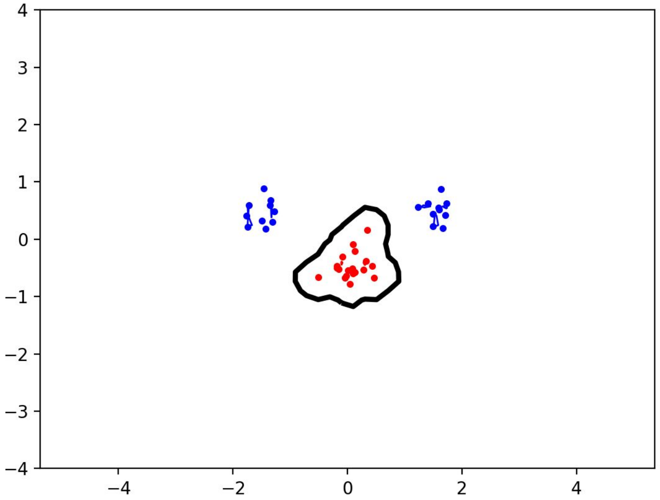


low degree -> under fitting -> low variance and high bias (complex decision boundary)

high degree -> over fitting -> high variance and low bias (simple decision boundary)

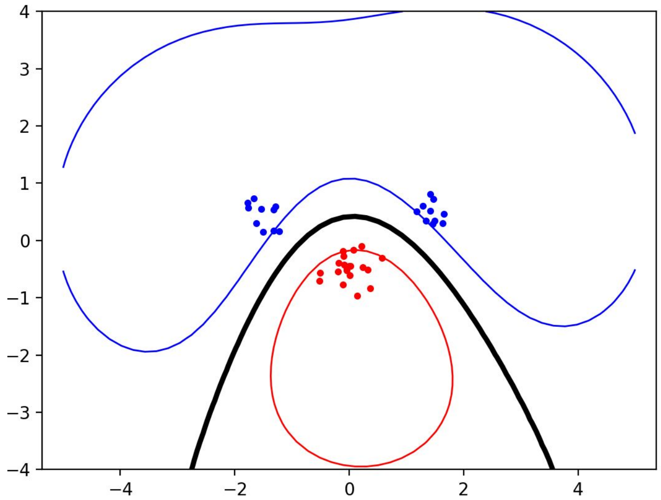
__RBL__ using different sigma

Sigma = 0.1



Low Sigma -> complex decision boundary -> over fitting

Sigma = 2.0



High Sigma -> Smooth decision boundary -> under fitting

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?

- We use a non-linear Kernel in complex problems to find a separation line.

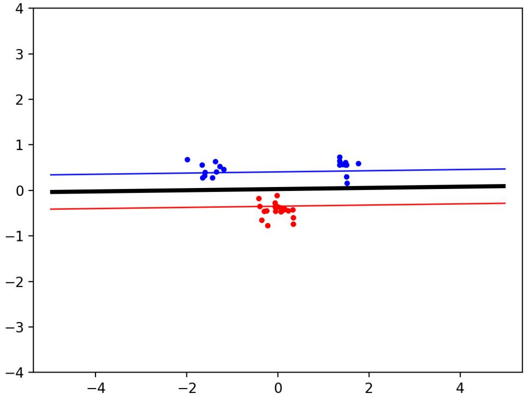
- If the dataset is noisy though and we try to seperate it, it doesn’t generalize well.

- In that case, more slack is good for noisy environments

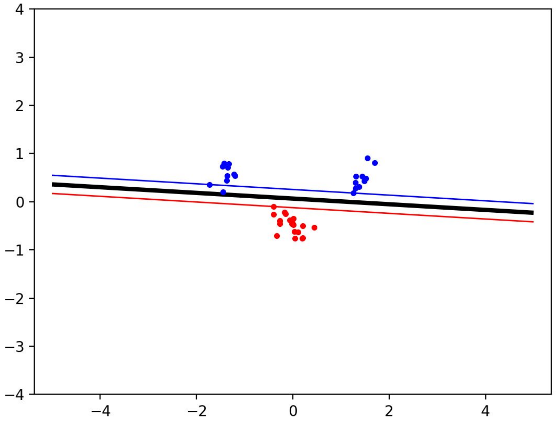
Explore the role of the slack parameter C. What happens for very large/small values?

- C controls how many points of the dataset can be misclassified by the constructed indicator function
- High C --> less misclassification allowed
- good in noisy environments

C = 1



C = 100



The end :)