Lab 2

Quick Summary

Outline

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.

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

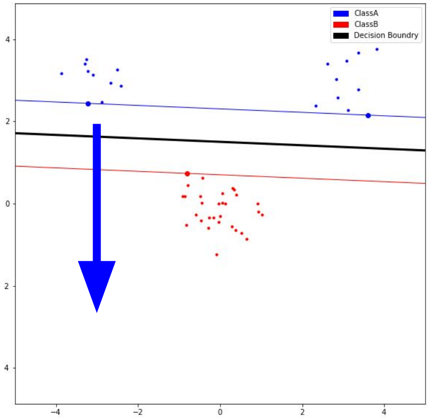
3. The non-linear kernels have parameters; explore how they influence the decision boundary. Reason about this in terms of the bias-variance trade-off.

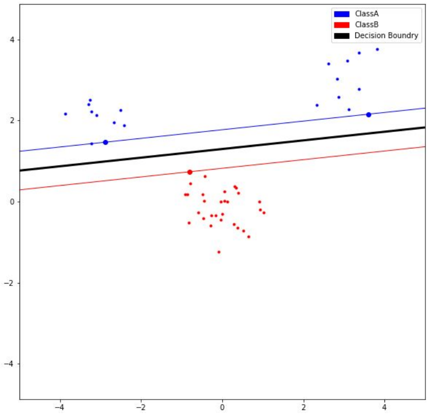
4. Explore the role of the slack parameter C. What happens for very large/small values?

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?

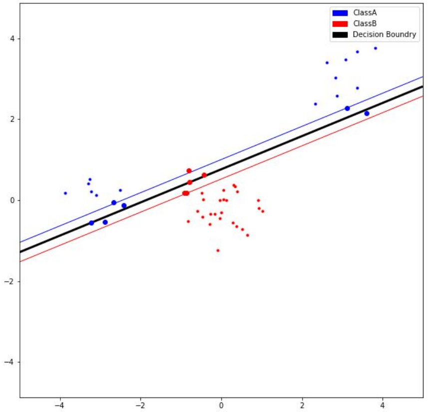
Linear Kernel

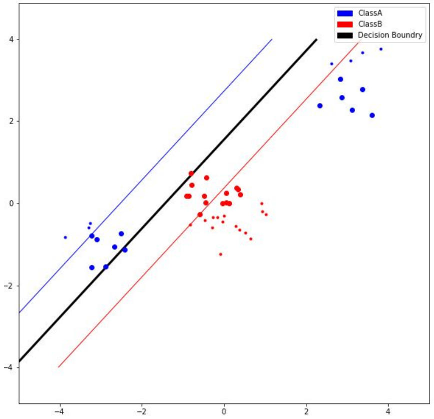
Linearly separable.





Not linearly separable.





Non-linear Kernels

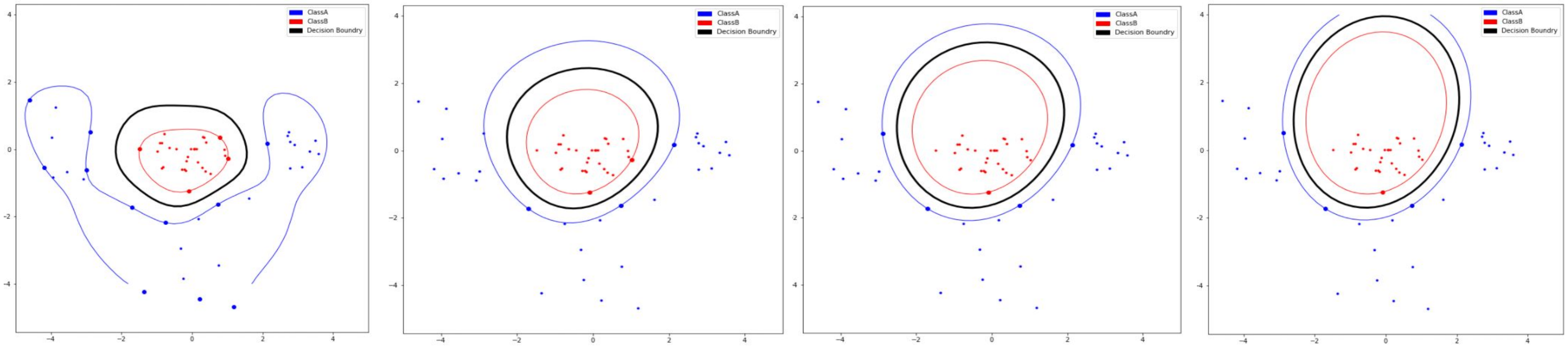
RBF Kernel

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

As sigma is increases, the function smoothness increases

Variance decreases as well, but the bias will be higher

σ = 1 σ = 2 σ = 3 σ = 5



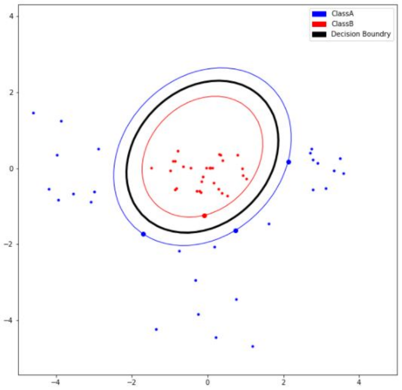
Poly Kernel

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

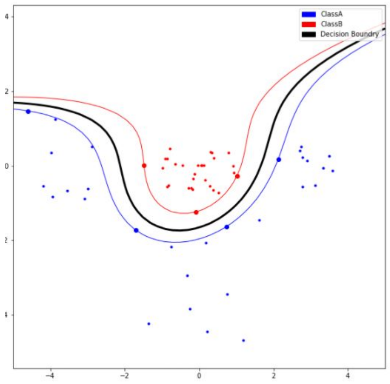
Complexity will increase with a higher order regression polynomial (p)

Variance increases and bias decreases when p is higher

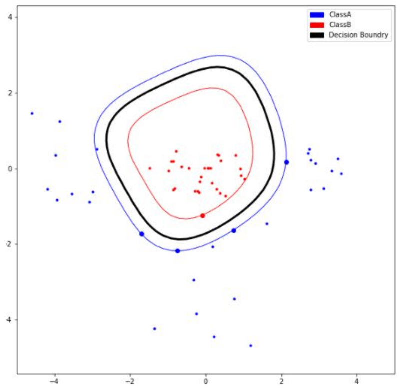
p = 2



p = 3



p = 4



Compare C

Linear Kernel

Compare on “C”

C = 100

$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C \sum_i \xi_i$$

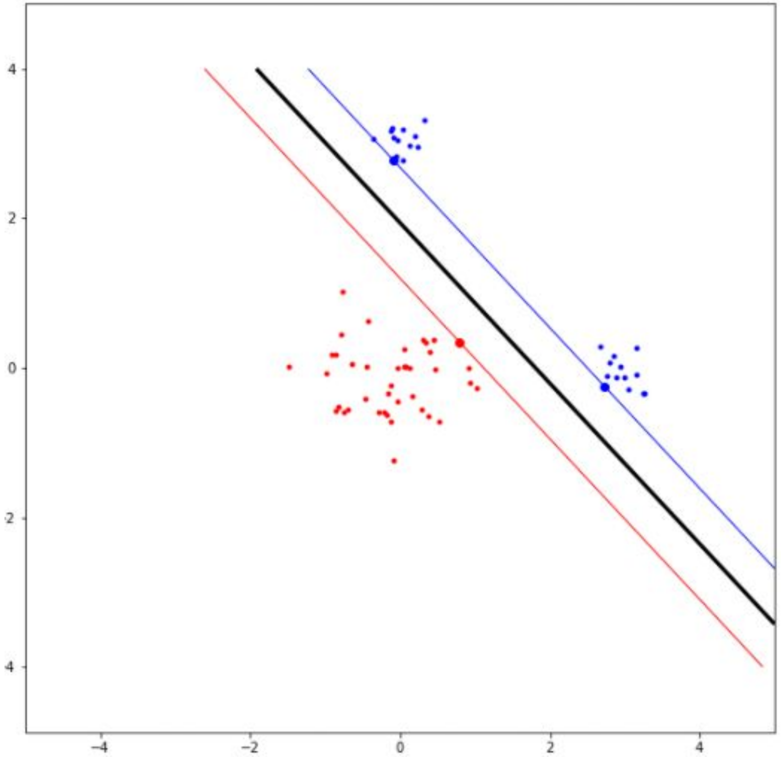
```
#Generate Data
num_points = 40

#Class A1
std_a1 = 0.2
x_a1 = 0.0
y_a1 = 3.0

#Class A2
std_a2 = 0.2
x_a2 = 3.0
y_a2 = 0.0

# #Class A3
# std_a3 = 0.5
# x_a3 = 0.0
# y_a3 = -3.0

#Class B
std_b = 0.8
x_b = 0.0
y_b = 0.0
```



Linear Kernel

Compare on “C”

C = 1

$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C\sum_i \xi_i$$

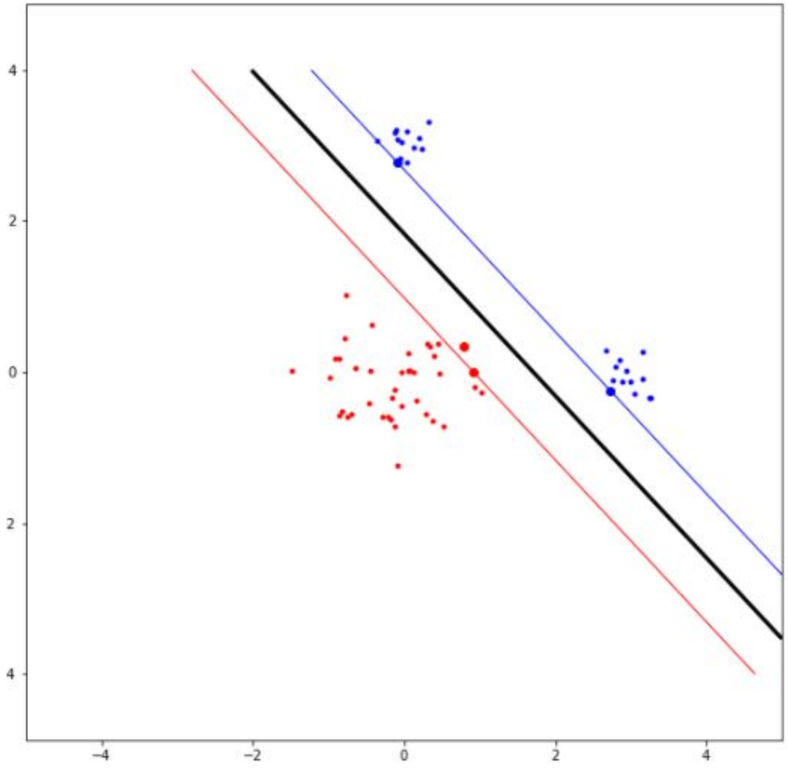
```
#Generate Data
num_points = 40

#Class A1
std_a1 = 0.2
x_a1 = 0.0
y_a1 = 3.0

#Class A2
std_a2 = 0.2
x_a2 = 3.0
y_a2 = 0.0

# #Class A3
# std_a3 = 0.5
# x_a3 = 0.0
# y_a3 = -3.0

#Class B
std_b = 0.8
x_b = 0.0
y_b = 0.0
```



Linear Kernel

Compare on "C"

C = 0.1

$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C\sum_i \xi_i$$

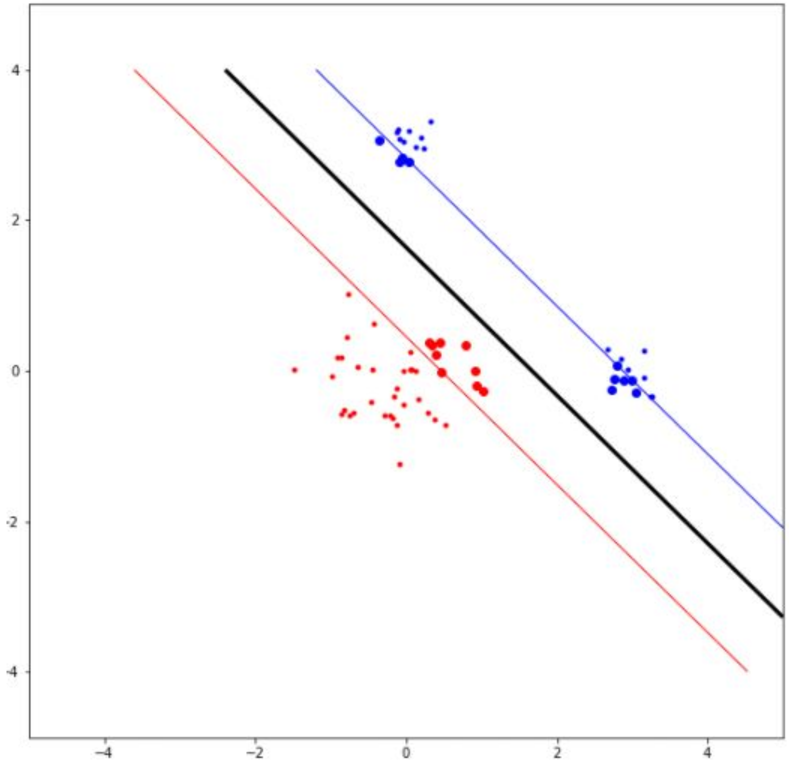
```
#Generate Data
num_points = 40

#Class A1
std_a1 = 0.2
x_a1 = 0.0
y_a1 = 3.0

#Class A2
std_a2 = 0.2
x_a2 = 3.0
y_a2 = 0.0

# #Class A3
# std_a3 = 0.5
# x_a3 = 0.0
# y_a3 = -3.0

#Class B
std_b = 0.8
x_b = 0.0
y_b = 0.0
```



RBF Kernel

Compare on “C”

C = 1000

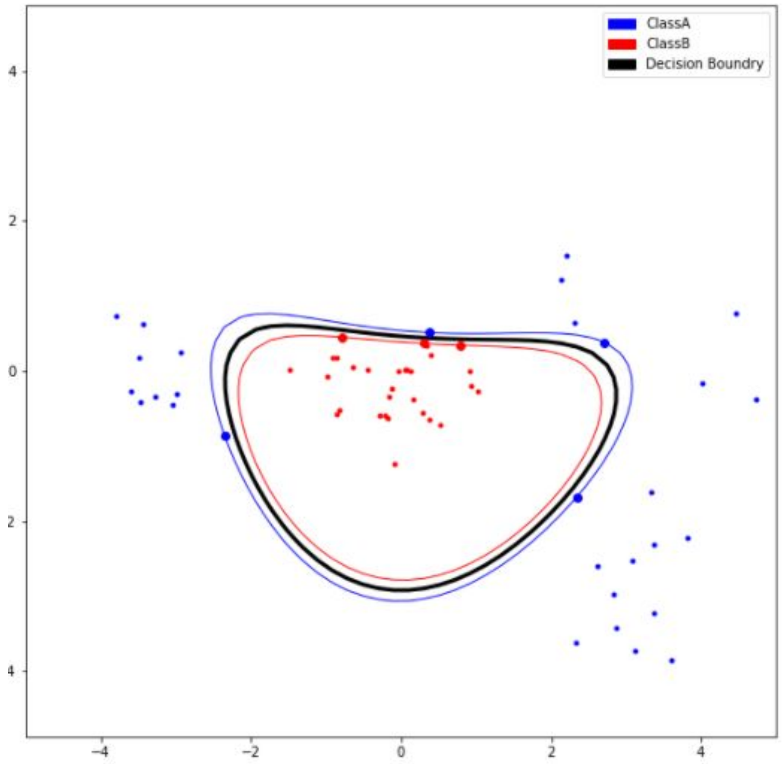
$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C\sum_i \xi_i$$

```
#Class A1
std_a1 = 1.5
x_a1 = 3.0
y_a1 = 0.0

#Class A2
std_a2 = 0.5
x_a2 = 3.0
y_a2 = -3.0

#Class A3
std_a3 = 0.5
x_a3 = -3.0
y_a3 = 0.0

#Class B
std_b = 0.5
x_b = 0.0
y_b = 0.0
```



RBF Kernel

Compare on “C”

C = 1

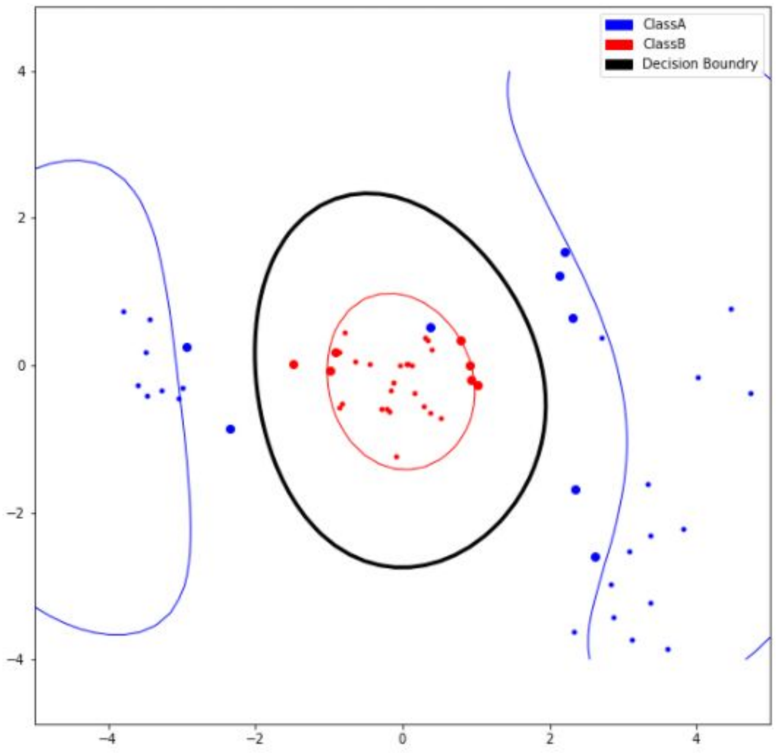
$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C\sum_i \xi_i$$

```
#Class A1
std_a1 = 1.5
x_a1 = 3.0
y_a1 = 0.0

#Class A2
std_a2 = 0.5
x_a2 = 3.0
y_a2 = -3.0

#Class A3
std_a3 = 0.5
x_a3 = -3.0
y_a3 = 0.0

#Class B
std_b = 0.5
x_b = 0.0
y_b = 0.0
```



When to address Slack versus Kernels?

It depends on the data set as well.

Opting for more slack is a good idea when there is noise in the data, but the classes are still **linearly separable**. More slack will lead to higher variance and lower bias.

Otherwise, more complex data sets may require more complex model for better return on bias and variance.