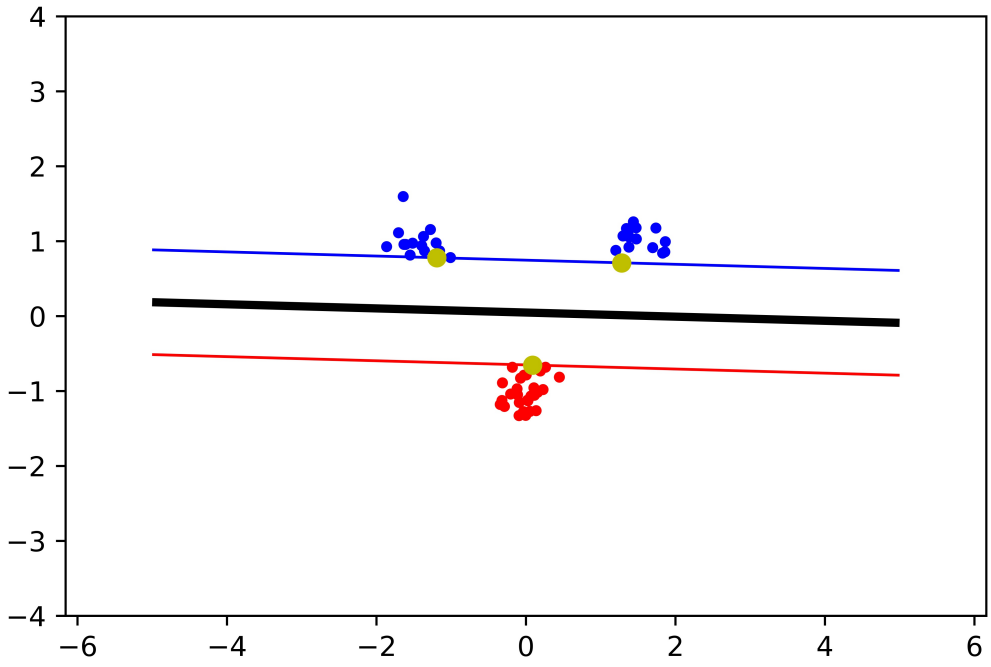
DD2421 Lab 2

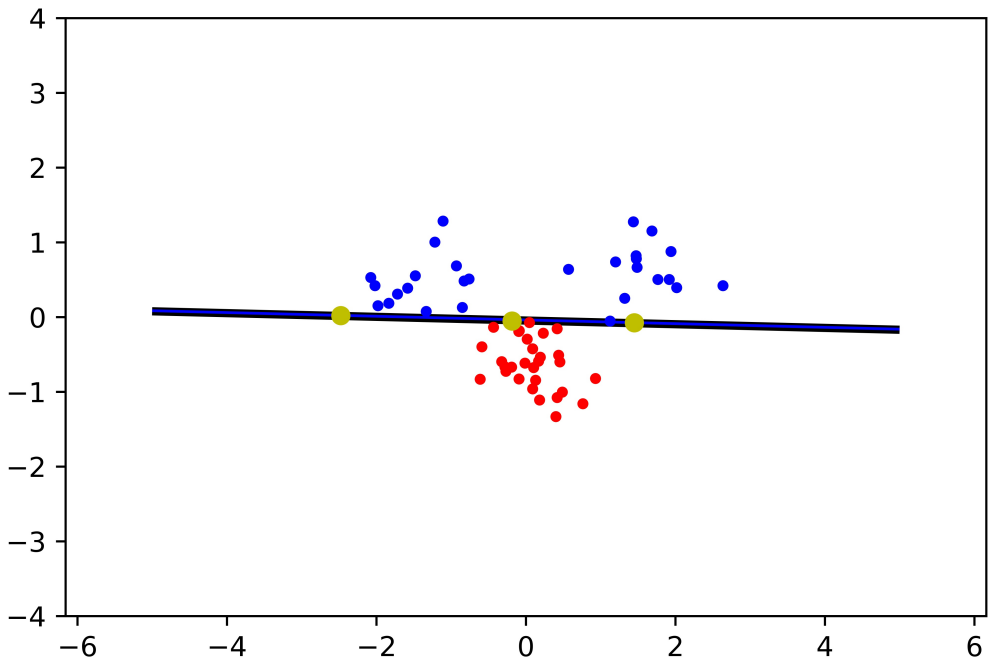
Chen Yuan, Yuxuan Zhang

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.

When classA points are generated around $(-1.5, 1.0)$ and $(1.5, 1.0)$, classB points are generated around $(0.0, -1.0)$, both with $std = 0.2$ and $C = 100000$ (no soft margins), the classifier can easily find the decision boundary.



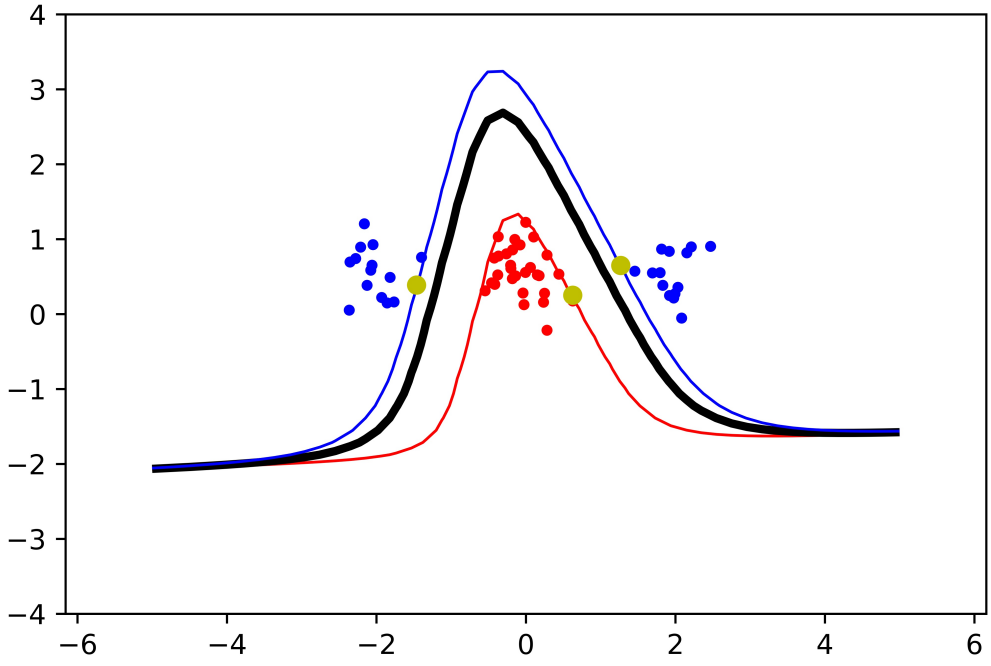
When classA points are generated around $(-1.5, 0.5)$ and $(1.5, 0.5)$, classB points are generated around $(0.0, -0.5)$ (moving the positions of clusters), both with $std = 0.4$ (enlarging the spreads/sizes of clusters) and $C = 100000$ (no soft margins), the classifier can hardly find the decision boundary.



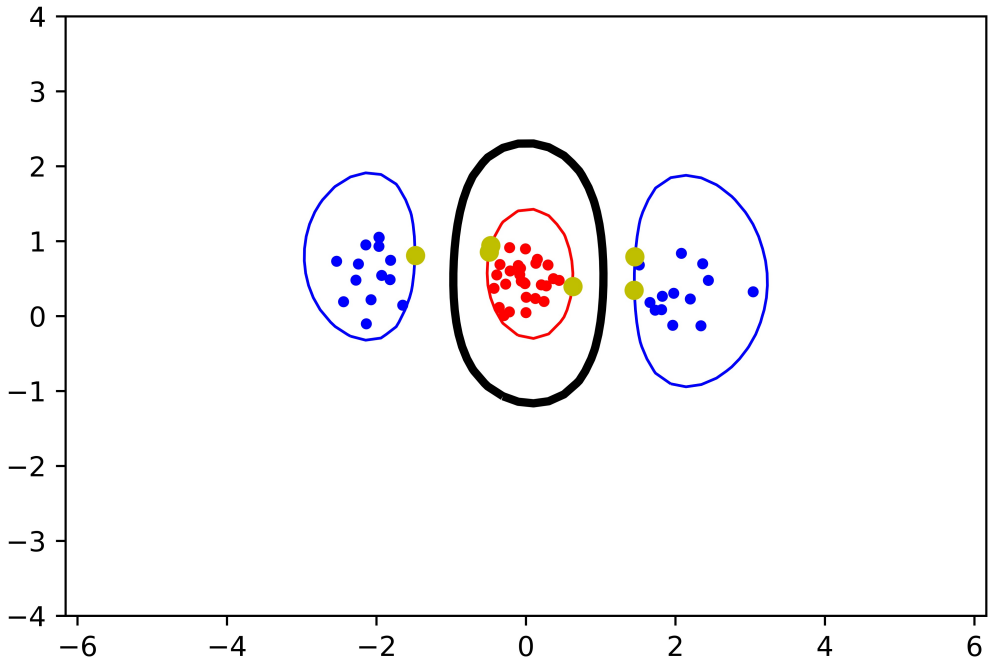
It can be seen that when the two clusters are not serperable by a straight line, the optimizer with the linear kernel won't be able to find a solution at all.

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

For a polynomial kernel with $p = 5$, when classA points are generated around $(-2.0, 0.5)$ and $(2.0, 0.5)$, classB points are generated around $(0.0, 0.5)$, both with $std = 0.3$ and $C = 100000$ (no soft margins):



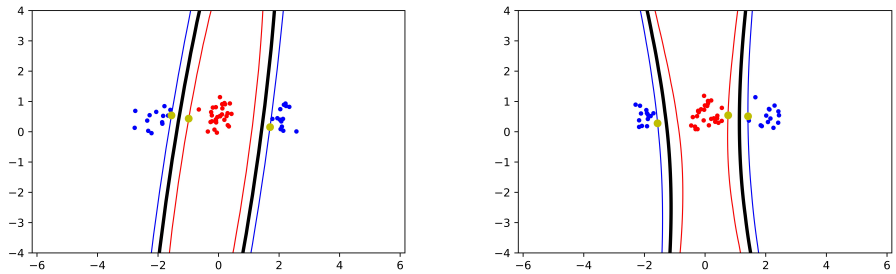
For an RBF kernel with $\sigma = 1$, when classA points are generated around $(-2.0, 0.5)$ and $(2.0, 0.5)$, classB points are generated around $(0.0, 0.5)$, both with $std = 0.3$ and $C = 100000$ (no soft margins):



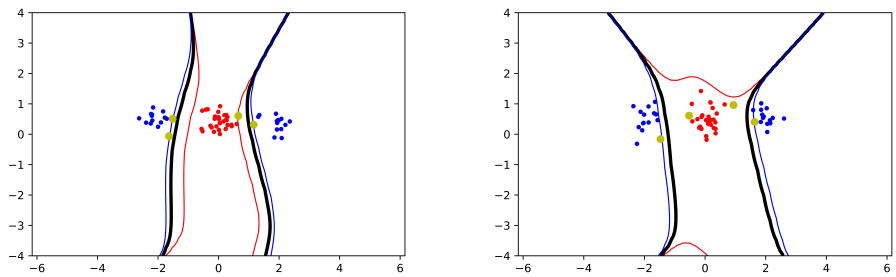
3. The non-linear kernels have parameters; explore how they influence the decision boundary. Reason about this in terms of the bias-variance trade-off.

Test data is generated with the same parameters as Q2.

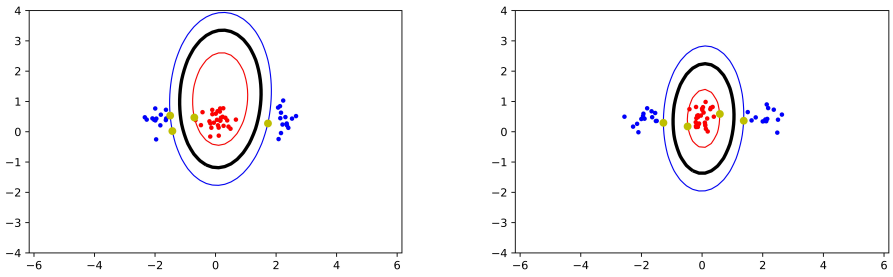
For polynomial kernels, $p = 2$:



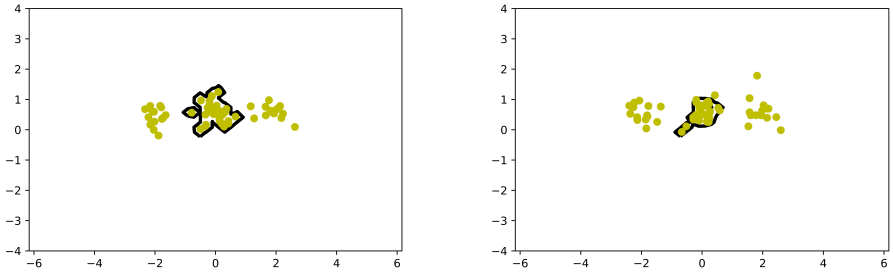
$p = 8$:



For RBF kernels, $\sigma = 5$:



$\sigma = 0.5$:

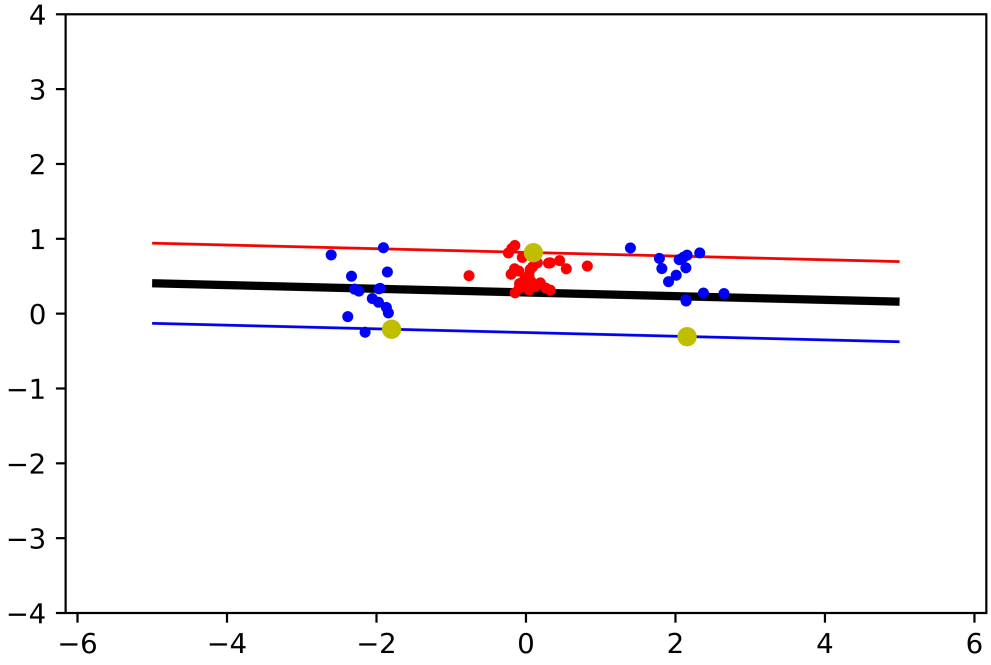


less smooth (larger p, smaller σ) => lower bias but larger variance

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

Test data is generated with the same parameters as Q2.

With a linear kernel, when $C = 1$: (however no solution when $C = 100000$)



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?

It depends on whether the given data is linearly seperable (whether it's necessary to increase dimension).