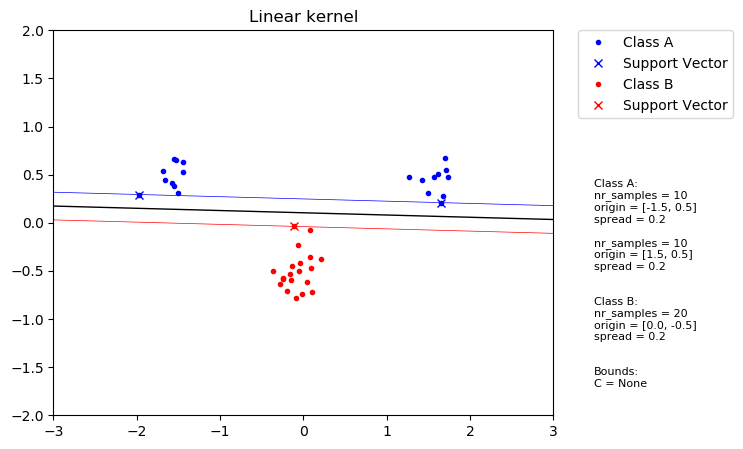
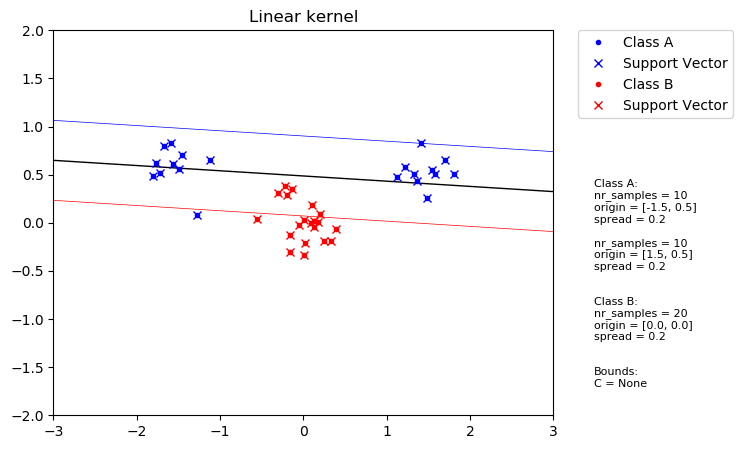
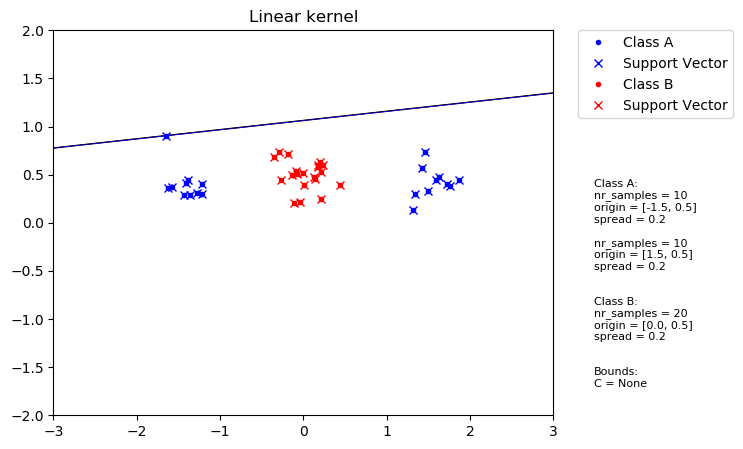
Lab 2: Support Vector Machines

**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.**

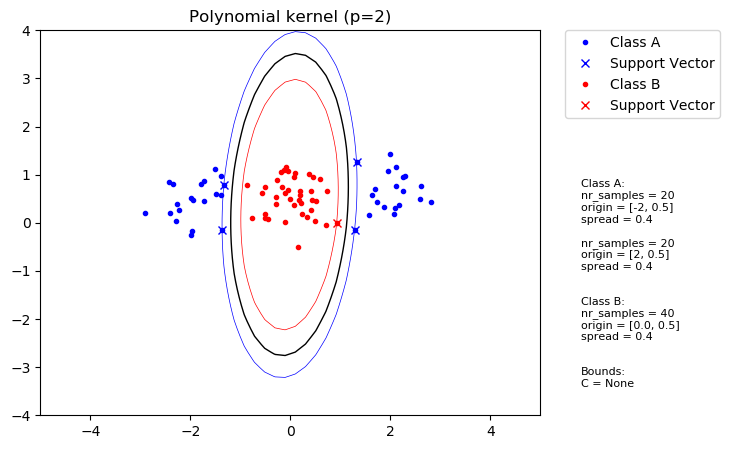
The following images showcases what happens when the origin of cluster B is moved upward along the y-axis. The closer the clusters are moved together the harder it is for the minimize function to find a solution. When the origins have the same y-value for all clusters, the minimizer completely failed.

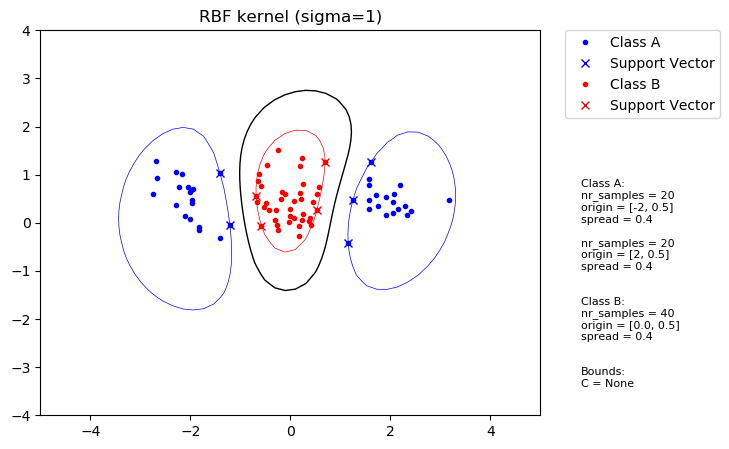






**2. Implement the two non-linear kernels. You should be able to classify very hard data sets with these.**



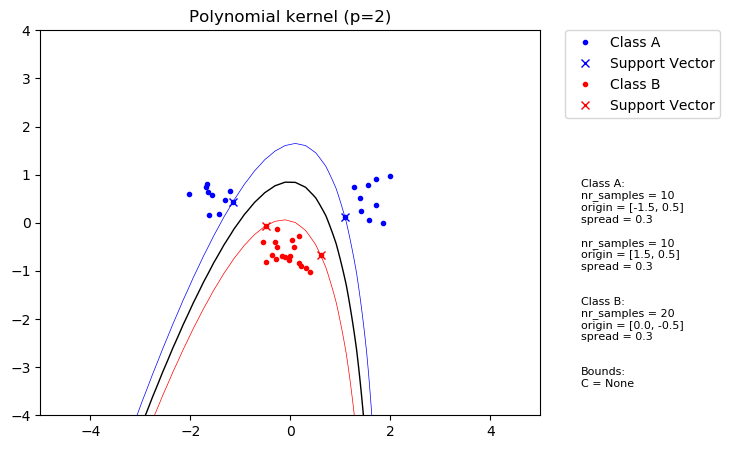


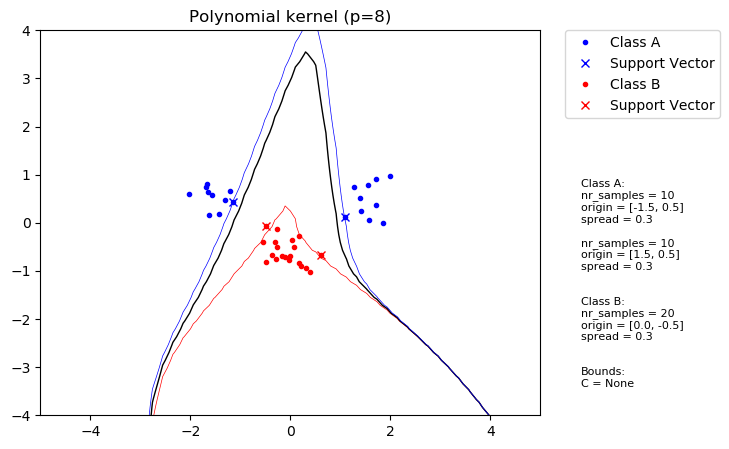
**3. The non-linear kernels have parameters; explore how they influence the decision boundary. Reason about this in terms of the bias variance trade-off.**

In the polynomial kernel *p* dictates the degree of the polynomial. A low *p* value generates rounder simpler shapes which doesn’t follow the individual samples as closely and therefore introduces more bias. A larger *p* value allows for more complex and sharper boundaries as seen in the second image. This in turn makes the boundaries follow individual samples more closely but is prone to overfitting.

low *p* value => increased bias

high *p* value => increased variance

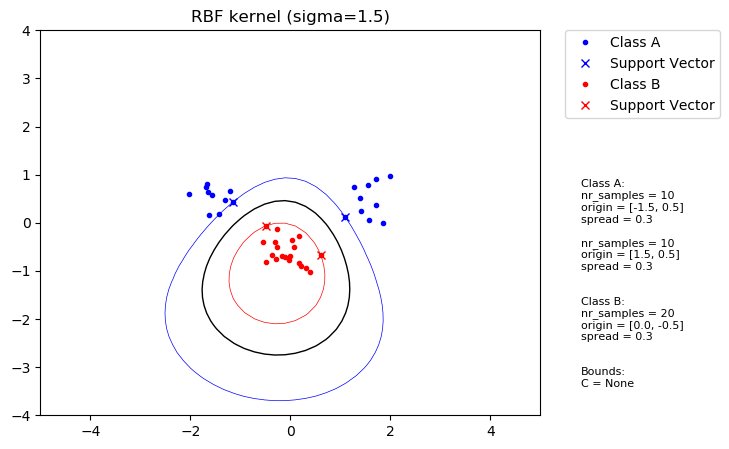


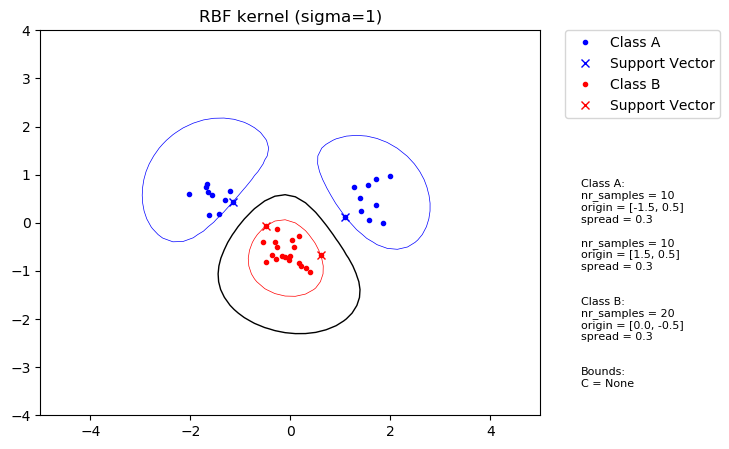


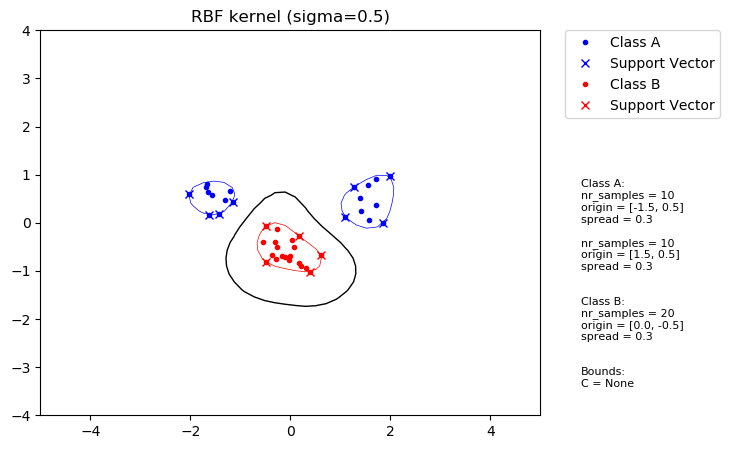
In the RBF kernel sigma is used to determine how smooth the boundaries are. Higher sigma’s produce smoother curves at the cost of an increase in bias. When sigma is reduced the curves follow the samples more closely and the variance increase, i.e. more overfitting occurs.

high sigma value => increased bias

low sigma value => increased variance

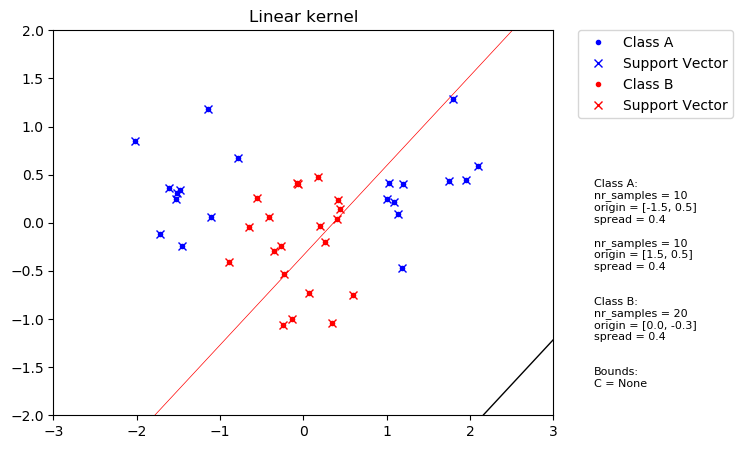


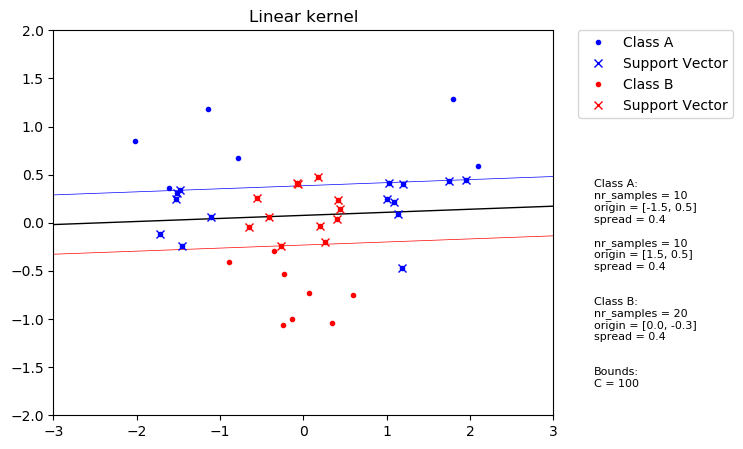


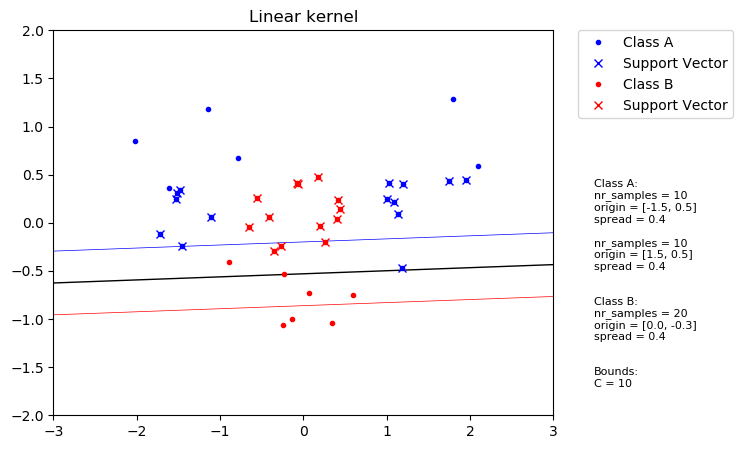


**4. Explore the role of the slack parameter C. What happens for very large/small values?**

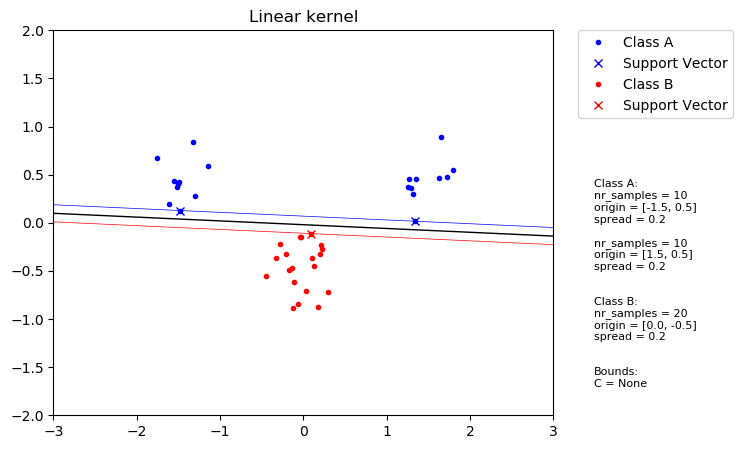
For a difficult problem where there is a lot of noise in the data requires a slack variable for a boundary to be found. By decreasing the *C* value which is the upper bound for *ɑ* limits the amount of “pull” a misclassified sample can have. This can be seen in the second figure where the boundary is a lot closer to y = 0 which would be the ideal boundary (if there was no noise). Though, it’s important to not limit alpha too much and therefore not give enough “pull” for the correct samples as can be seen in the last figure.

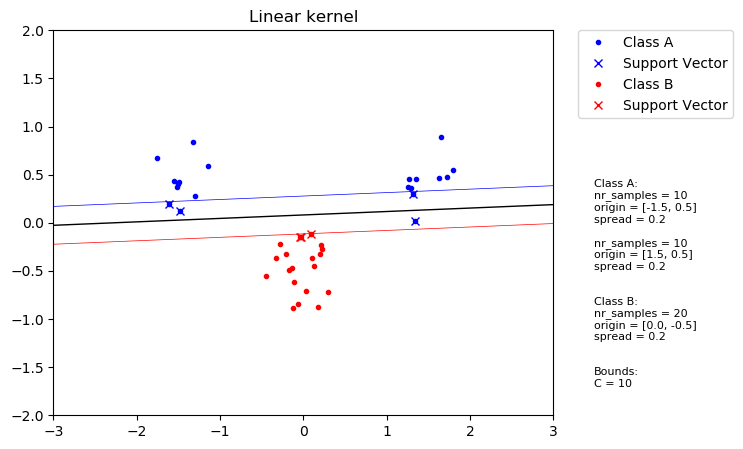






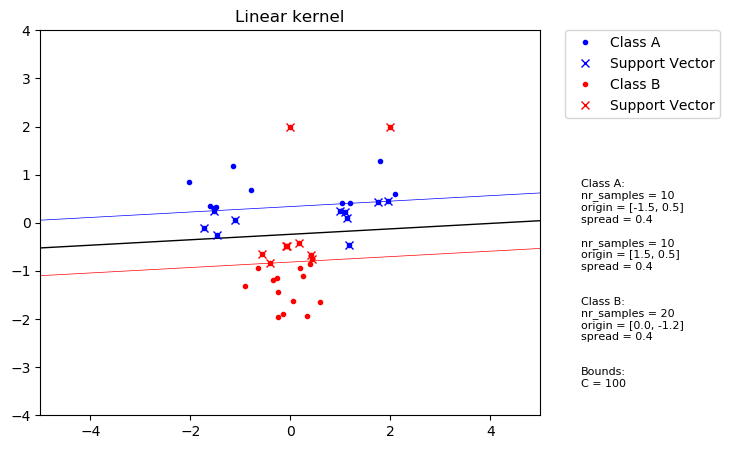
Another aspect of C is that lower values will allow for larger margins as can be seen in the following figures.



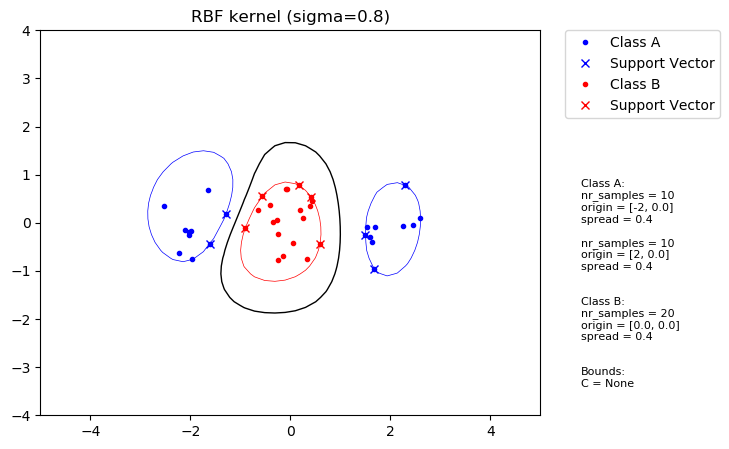


**5. Imagine that you are given data that is not easily separable. When should you opt for more slack rather than going for a more complex model (kernel) and vice versa?**

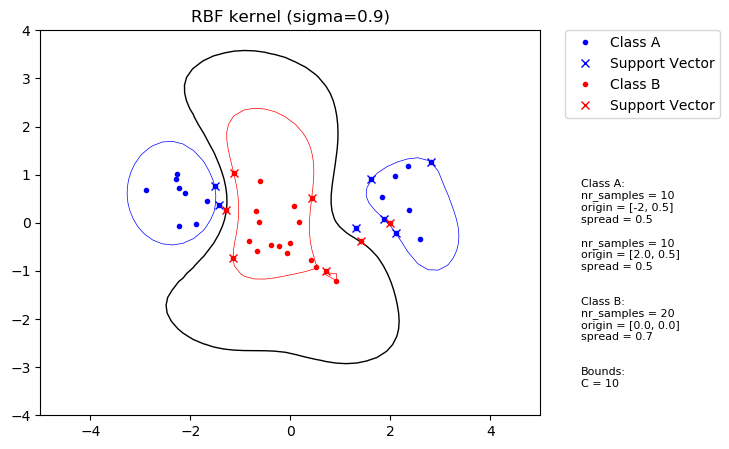
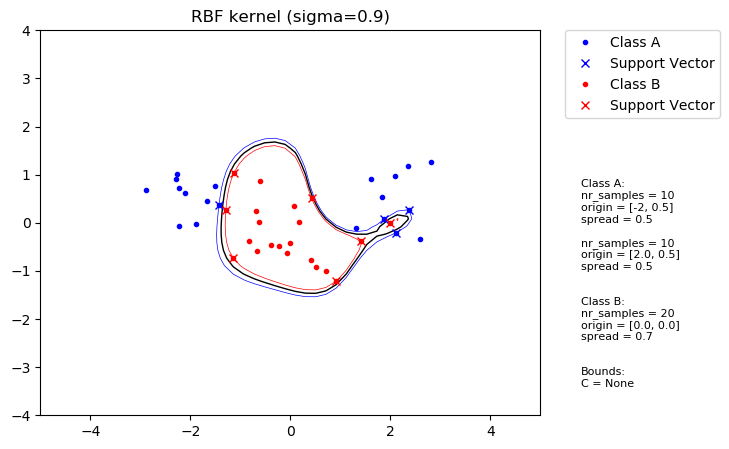
When the data is linearly separable but noisy then slack should be used. In the following figure two samples are way off, by increasing slack these misclassifications don’t skew the boundary too much:



When the data is not linearly separable but and not too noisy a complex model is better:



If the data is not linearly separable and is noisy than a complex model with slack is the best as can be seen in the following figures:



clusters linearly separable, low noise => simple model

clusters linearly separable, high noise => simple model + slack

clusters not linearly separable, low noise => complex model

clusters not linearly separable, high noise => complex model + slack