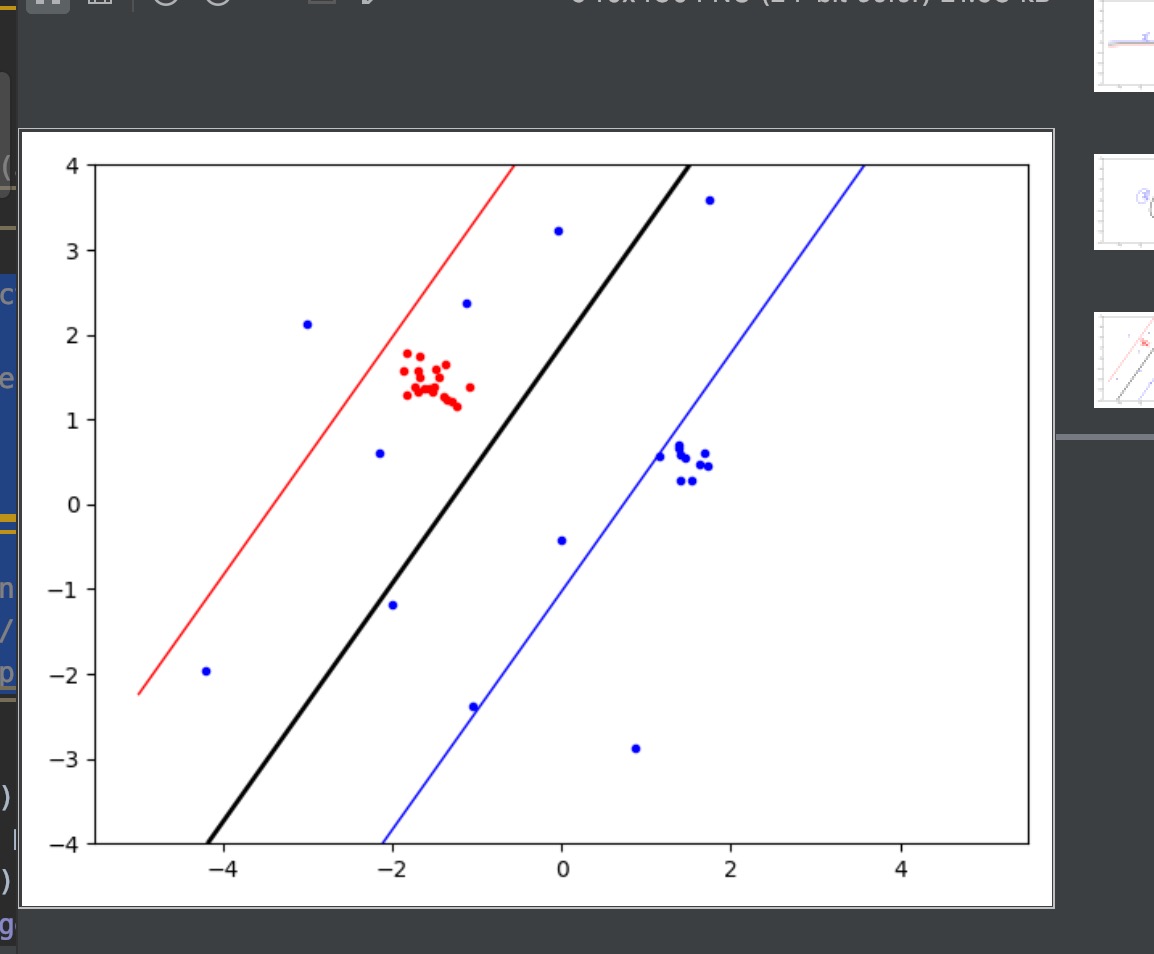
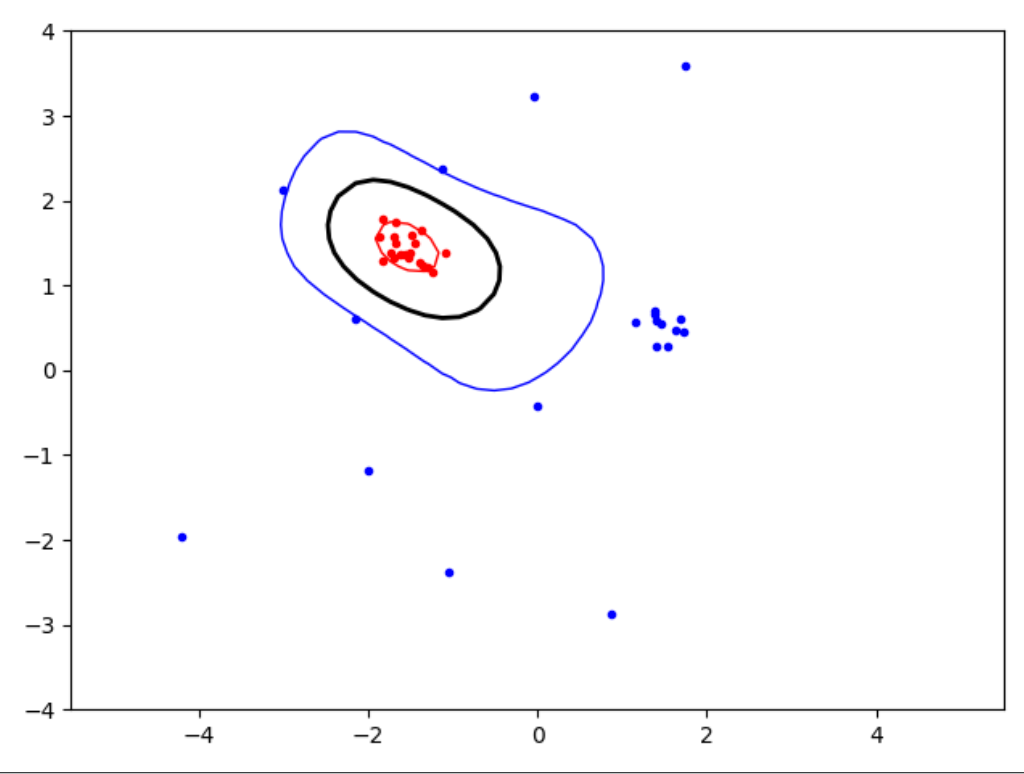
1. Move the clusters around and change their sizes to make it easier or harder for the classifier to find a decent boundary. Pay attention to when the optimizer (minimize function) is not able to find a solution at all.



Set part of classA ‘s variance from 0.2 to 2, making it more scattered. It’ s the situation which classA and classB totally non-linearly separable.

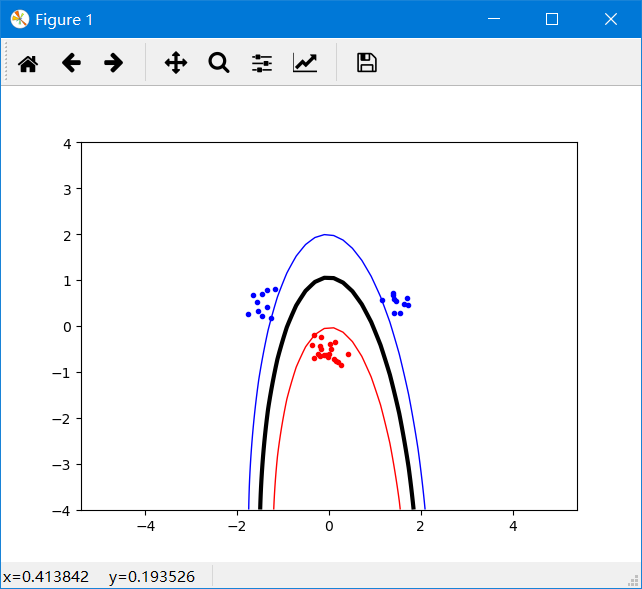
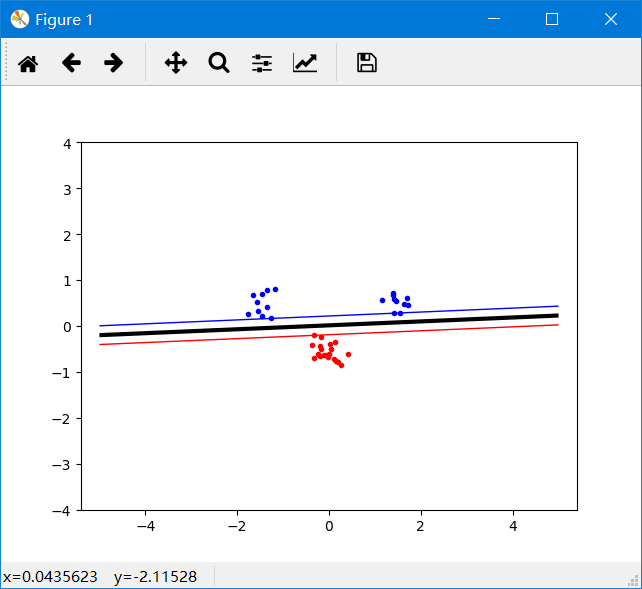
1. Implement the two non-linear kernels. You should be able to classify very hard data sets with these.

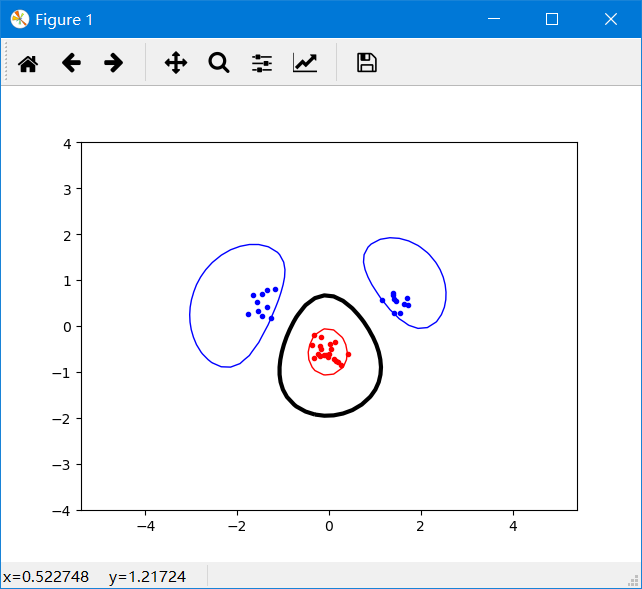
After inplement of the RBF kernel, the above data set is separable.



Other examples:

Linear kernel Polynomial kernels P=2



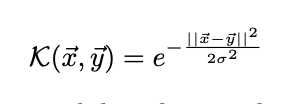


Radial Basis Function(RBF) kernel

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

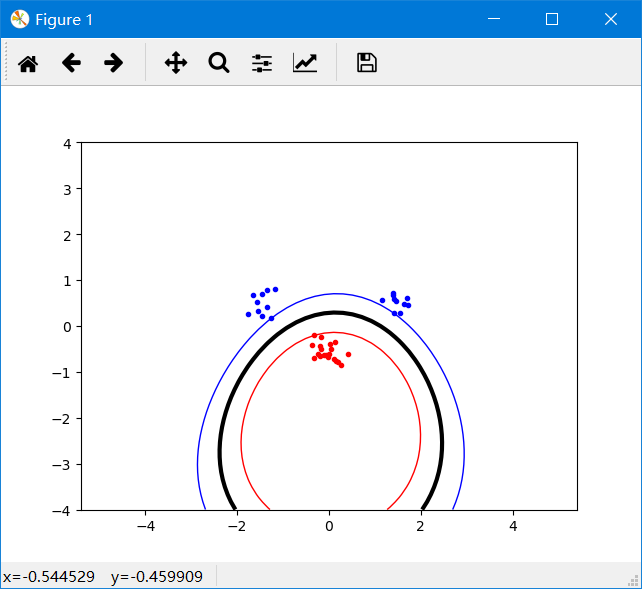
In polynomial kernels, P determines the power of the function. The bigger the P is, the more complex the decision boundary will be. And it’s likely to overfit when P become too big. Low bias but High variance.

In RBF kernel, the parameter sigma is used to control the smoothness of the boundary. When sigma becomes smaller, the model may overfit easily. The influence of support vectors to other points is very small. They focus on their own points.

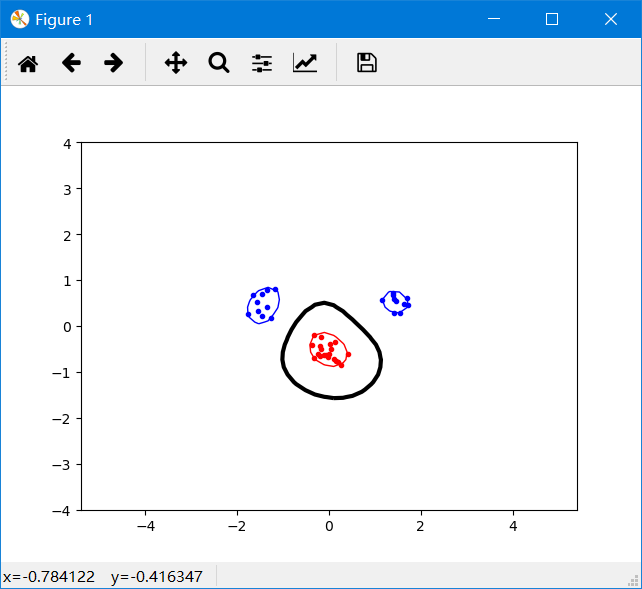


From the equation of RBF kernel, if sigma -> 0, all of alpha is larger than 0(the limitation of the equation -> 0, alpha can be any value)(only when xi = xj, K(xi, xj)!=0), so all of the vectors are support vectors. It has no classification abilty.

Sigma=3:



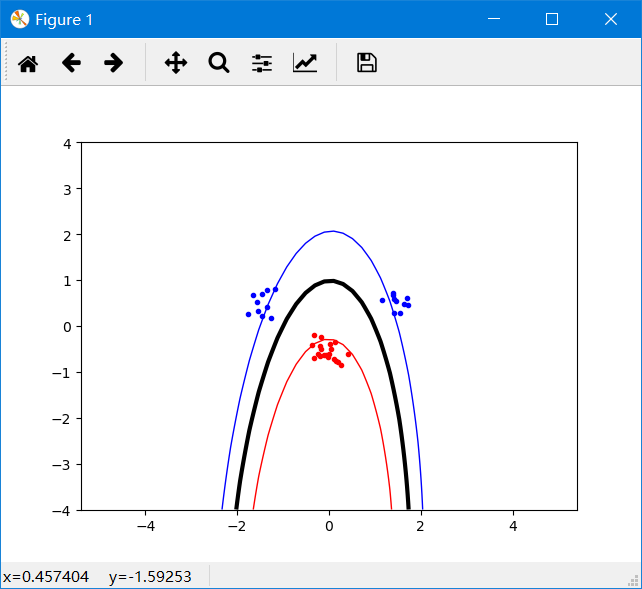
Sigma=0.3:



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

Noisy data typically deserve a low C value, allowing for more slack, since individual datapoints in strange locations should not be taken too seriously. When C become very large, the model will take every datapoint into consideration.

C=0.2 :



1. 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. If in the original low-dimension space, the samples are not so much nonlinearly separable, with only a little points that affect the outcomes. In this case, we can take these points as noise and allow a little bit slack to our model, to make the samples linearly separable. However, if the samples are obviously nonlinearly separable, we should first use a more complex kernel to map them into high dimension space, to make them more separable, and then use the slack parameter to discard the noise.

A smoother model under certain kernel can increase the value of C to fit more samples accurately. And when you change to a more complex model, there are increasing computation demands and need more time.