DD2421 - Lab 2

Pierre Rudin

October 6, 2017

1 Introduction

In this lab we take a look at the Support Vector Machine, coupled with different kernel functions.

2 Running and Reporting

2.1 Question 1

Move the clusters around to make it easier or harder for the classifier to find a decent boundary. Pay attention to when the **qt** function prints an error message that it can not find a solution.

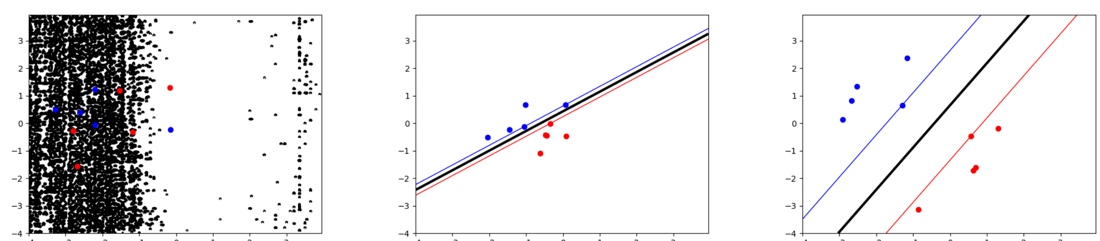

Figure 1: SVM used with a linear kernel, where the different classes are colliding, slightly separable, and distinctively separable

If the linear kernel SVM can classify our data-set or not depends only on if it is possible to separate the classes with a straight line. To increase the probability that the SVM will make correct classifications we should move the clusters further away from each other. Error messages occur when the SVM cannot make correct classifications e.g. when there ain't no straight line that separate the different classes.

2.2 Question 2

Implement some of the non-linear kernels. you should be able to classify very hard data-sets.

This doesn't really look like a question, but hey, Radial Basis Function and Polynomial kernels are implemented and working. More on them in next section.

2.3 Question 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.

2.3.1 Polynomial

As p is increased variance decreases and bias increases. When p becomes to large it will no longer produce plausible classifications, so in this case I would say that the upper boundary for what would be considered a good p lays somewhere in between 10 and 20.

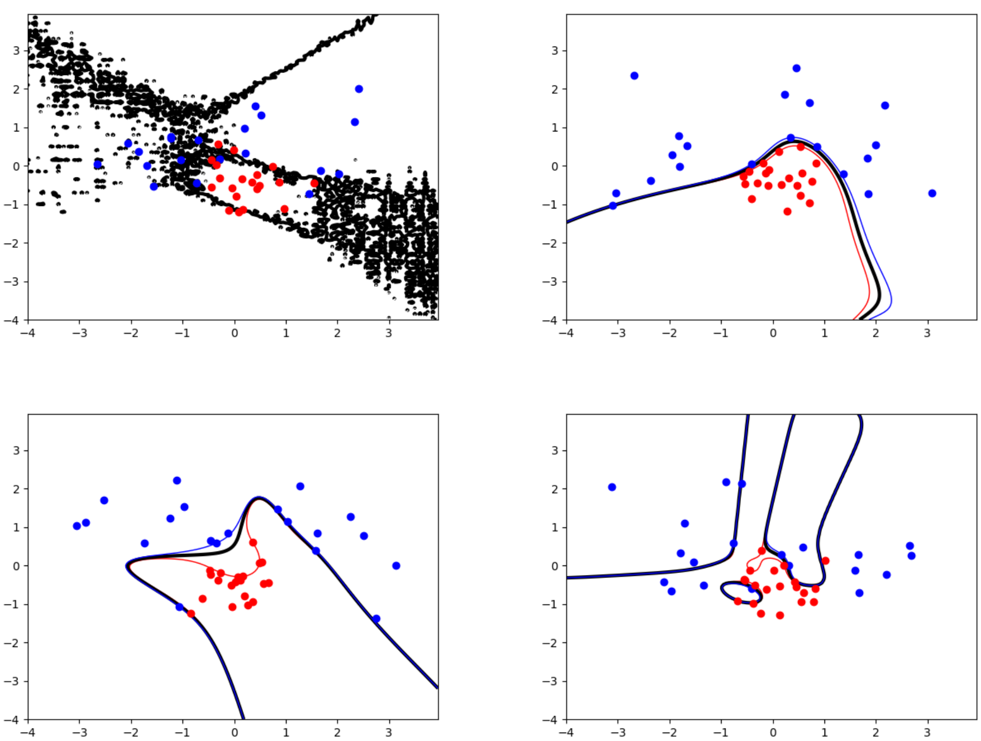

Figure 2: Plotted classifications where p = [3,4,10,20]

2.3.2 Radial basis function

When playing around with σ it is noted that a higher value yields lower bias and higher variance, and in contrast that a low value on σ gives high bias and low variance. Also noted is that the number of support vectors increases when σ is lowered. Testing on a data-set with 40 samples yielded the following number of support vectors given different σ, $\sigma = .1$ $\rightarrow$ 40 support vectors, $\sigma = .5 \rightarrow$ 20 support vectors, $\sigma = 1 \rightarrow$ 10 support vectors and σ $= 5 \rightarrow$ 5 support vectors. This is a pretty small data-set however we don't want to many support vectors, so in this case $\sigma = 5$ is preferable, to consider though is that $\sigma = 5$ didn't always find a solution. So if we want a persistent solution finder, $\sigma = 1$ would be to prefer for this data-set.

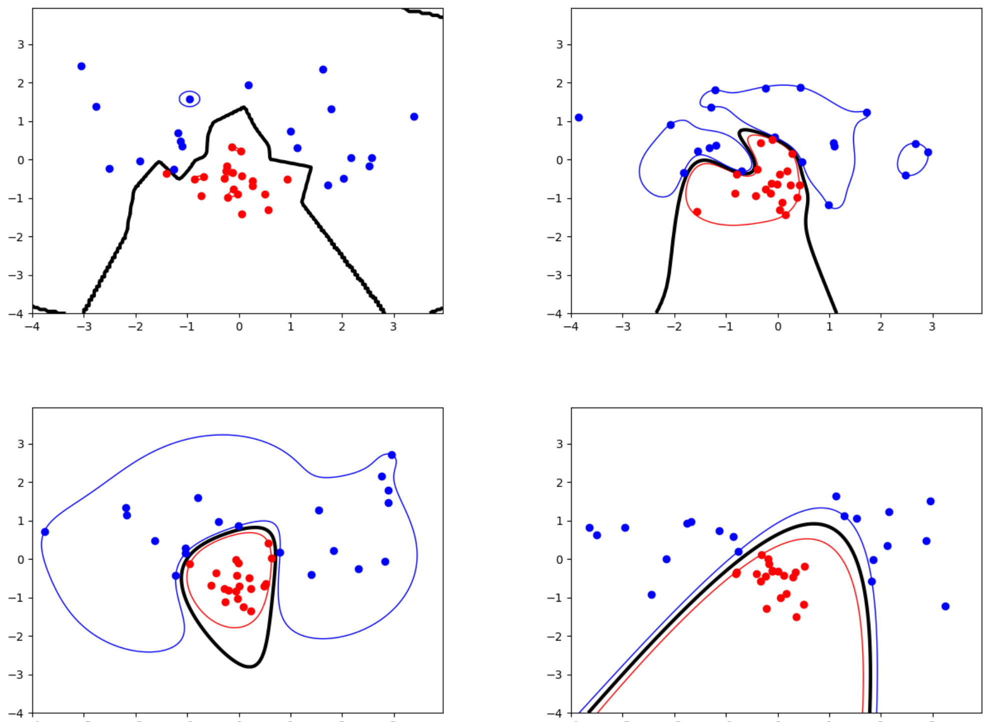

Figure 3: Plotted classifications where σ = [.1,.5,1,5]

3 Slack Implementation

3.1 Question 1

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

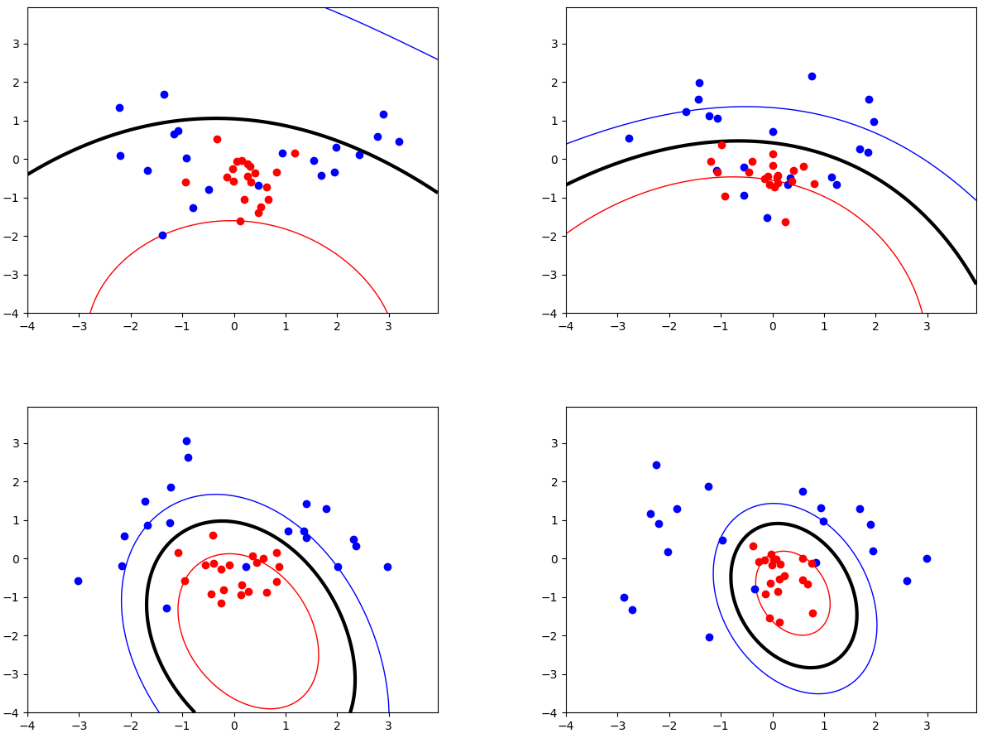

Figure 4: Classifications made using rbf kernel where $\sigma = 5$ and $C = [1,10,100,100]$

To view the effects of C, a Radial Basis Function where used, and since $\sigma = 5$ was the best performer in the previous task it is used again. Another reason to select $\sigma = 5$ was the fact that it sometimes struggled to classify the data-set.
The first thing that is noticed when C is decreasing is that the red and blue lines gets closer to each other. This is expected since a smaller C should allow fewer errors and errors can only occur between these lines. Another observation is that the number of support vectors also are decreasing, and this is also expected since the support vectors are those data-samples along the lines and those in between.

3.2 Question 2

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 and vice versa?

In general it is not desirable to have to many support vectors, a rule of thumb is that the number of supports vectors should not be more than 10-20 % of the number of data samples. So when the model classifying is so complex that the number of support vectors gets to high it is generally better to use a simpler model with some slack added.