Lab 1 Machine Learning

Anton Stråhle & Jan Alexandersson

February 16, 2020

Dual formulation

The minimization of the objective can be written as

$$\min_{\alpha_1,\dots,\alpha_n} \frac{1}{2} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}^T \begin{bmatrix} t_1 t_1 \kappa(\vec{x}_1, \vec{x}_2) & \dots & t_1 t_n \kappa(\vec{x}_1, \vec{x}_n) \\ \vdots & \ddots & \\ t_n t_1 \kappa(\vec{x}_n, \vec{x}_1) & & t_n t_n \kappa(\vec{x}_n, \vec{x}_n) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} + \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}^T \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

with the corresponding constraints

$$\begin{bmatrix} -1 & \dots & 0 \\ \vdots & \ddots & \\ 0 & & -1 \\ 1 & \dots & 0 \\ \vdots & \ddots & \\ 0 & & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \preccurlyeq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ C \\ \vdots \\ C \end{bmatrix}.$$

We solve this using the QP function in the CVXOPT package.

Assignment 1

- An optimal solution can not be found using a linear kernel if the classes are not lineary separable.
- An optimal solution can not be found if the data is not separable with the polynomial degree used. For example, degree 3 may work but degree 2 may not, then data is separable with the degree 3.
- When using a radial kernel, an optimal solution can not be found if the data is not separable at all.

For the provided dataset we could not find an optimal solution using linear kernel or polynomial kernel with degree 2, for some seeds. In the figures below our data is separable with polynomial degree 2.

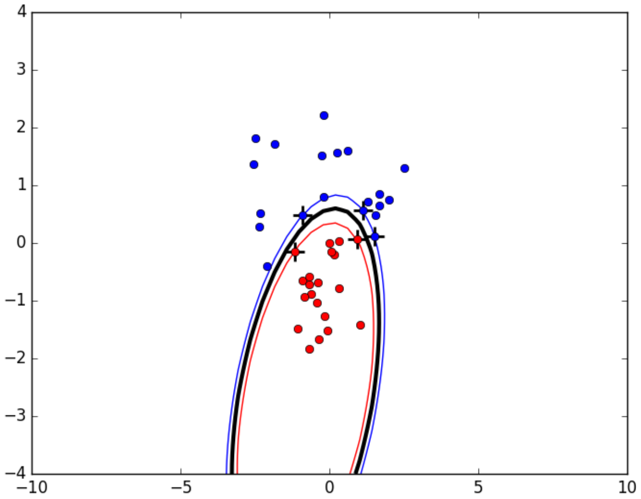


Figure 1: $C = 0$, polynomial degree 2

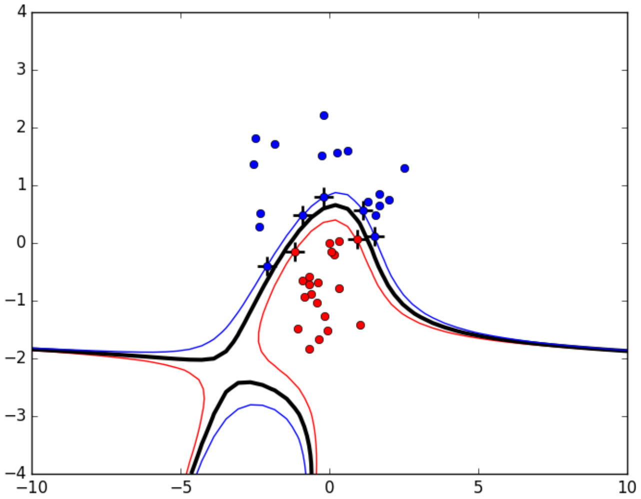


Figure 2: $C = 0$, polynomial degree 3

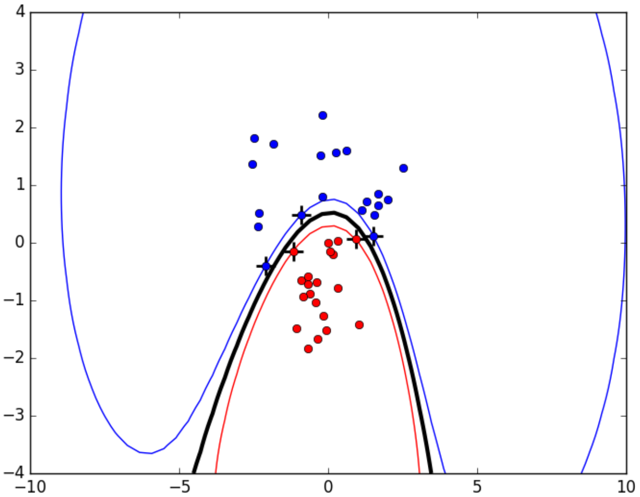


Figure 3: $C = 0$, sigma $= 2$

Assignment 2

```
def radial_kernel(x, y):
    diff = np.subtract(x, y)
    return math.exp((-np.dot(diff, diff)) / (2 * SIGMA * SIGMA))

def polynomial_kernel(x, y):
    return np.power((np.dot(x, y) + 1), POLYNOMIAL_GRADE)
```

Assignment 3

When using a polynomial kernel a higher degree or will classify more precise accordiong to the training set which will increase the variance but decreasing the bias. Similarily, when using a Radial Basis Function kernel a lower sigma will increase variance and decrease bias. This may lead to overfitting.

We can claerly observe in the figures below that we have clearly overfitted when using a too low value of sigma.

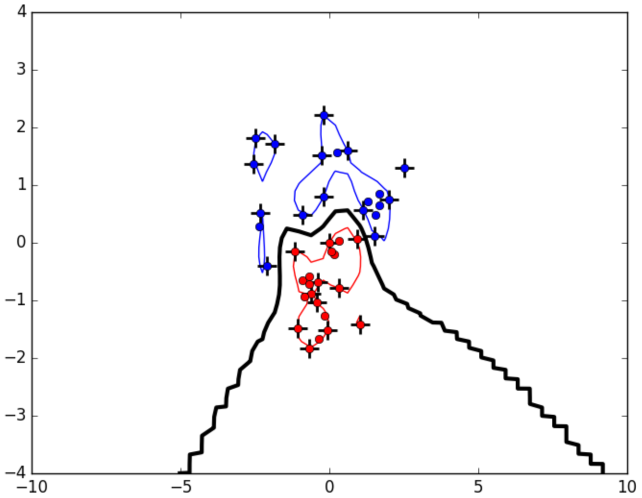


Figure 4: $C = 0$, sigma $= 0.5$

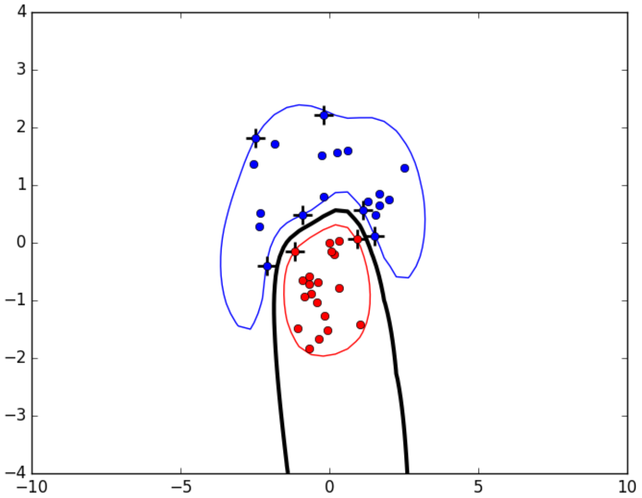


Figure 5: $C = 0$, sigma $= 1$

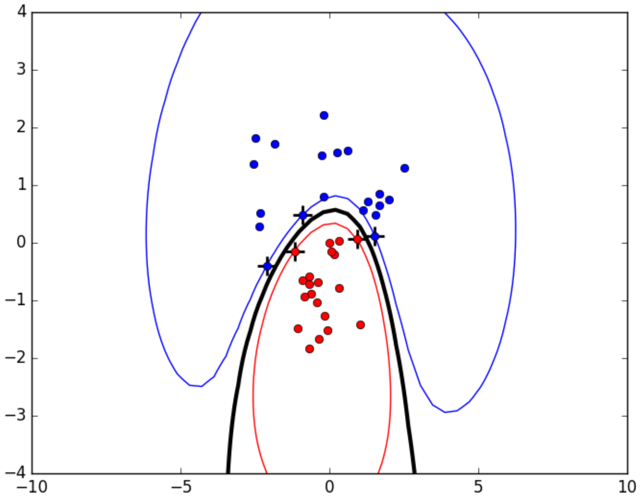


Figure 6: $C = 0$, sigma $= 2$

Assignment 4

If C is small, then there will be fewer support vectors and hence the resulting classifier will have low bias but high variance. This is because a low value of C does not allow for the margins to be as violated as a high value. The main advantage of using slack is that, when for example, if the underlying true calssification is lineary separated but the data has some noise we will still be able to use a linear kernel and allow for some violation of the margins. If the underlying model is simple but some datapoints prevents a simple classification, slack variables are useful. A large/high value of the slack variable will however reduce the accuracy of the model.

Below is the same data as before which was not lineary separable, however when allowing for some slack we can use linear classification.

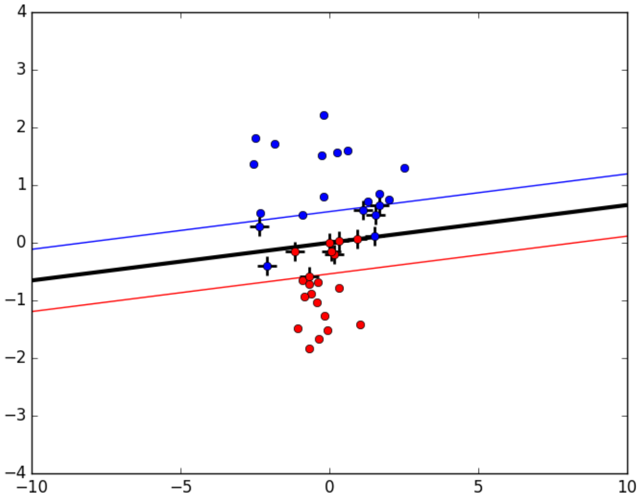


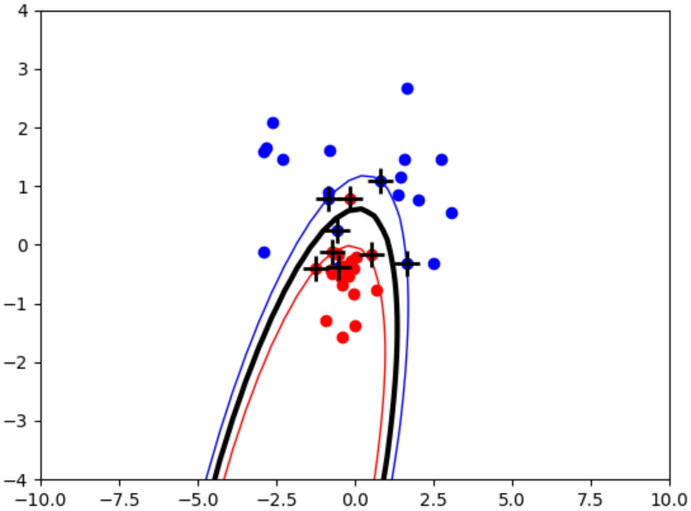
Figure 7: $C = 1$, linear kernel

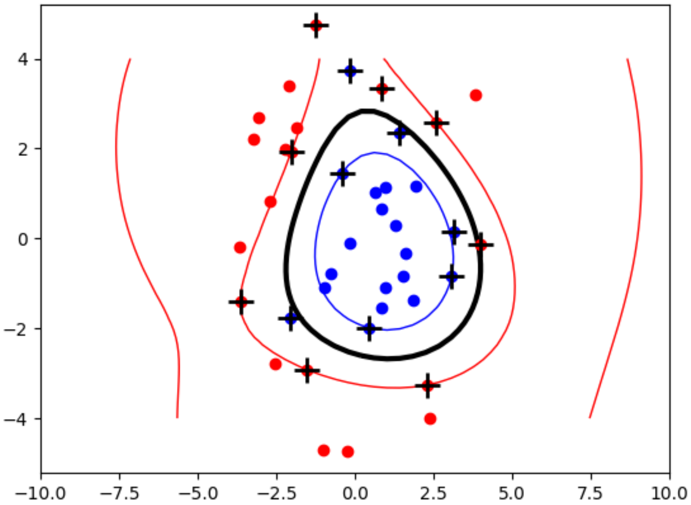
Assignment 5

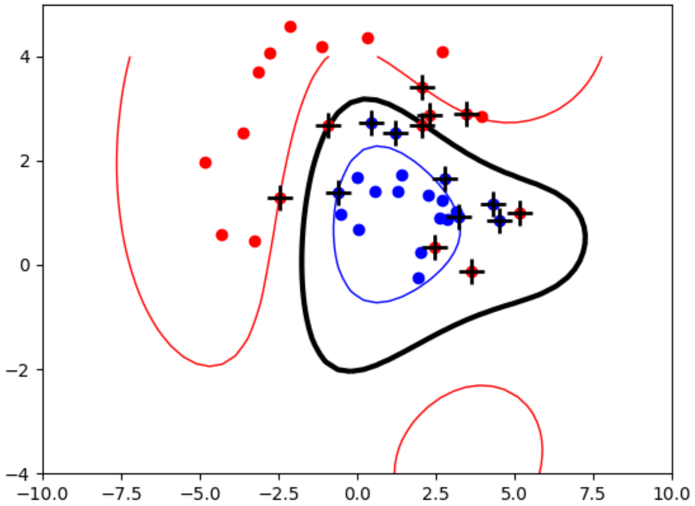
If we have a noisy dataset, i.e. multiple points deviating from the main clusters we should not opt for a more complex kernel since this would in some sense overfit the boundary to include these specific outliers which of course would impact future predictions negativeley. Instead we should allow for extra slack by increasing C, thus sacrificing some predictive power by obtaining a more general, and hopefully better predicting, boundary.

If we however have have a very sparse and spread out data with no clear major clusters we should consider using a more complex model to be able to incorporate the spread of the different classes. By allowing for extra slack in this case we might obtain a boundry that missclassifies certain minor classes which of course is not to be desired. In these cases with a lot of spread and no clear clusters we might also question the use of SVM entierly.

Appendix


4
3
2
1
0
−1
−2
−3
−4
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


4
2
0
−2
−4
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


4
2
0
−2
−4
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0