DD2421, MACHINE LEARNING

Lab 2: Support Vector Machines

Arthur CHANSEL

February 17, 2023

Learning objectives

- The indicator function (Equation 1) describes to which set a point belongs. It takes a new data point as input and returns a value indicating the predicted class label.
- A Kernel function computes the outcome of the scalar product of a higher dimensional pairs. Different kernel functions result in different type of data separation. Linear kernel (Equation 2) results in a linear separation. Polynomial kernel (Equation 3) allows for curved decision boundaries. Radial Basis Function (RBF) kernel (Equation 4) often results in very good boundaries, thanks to the control of the smoothness of the boundary.
- When using slack variables (Equation 5), the C value controls the the relative importance of mistakes in the optimization problem. A high C stands for few acceptable mistakes. A low C stands for many acceptable mistakes.

$$\text{ind}(\vec{s}) = \vec{w}^T \cdot \phi(\vec{s}) - b = \sum_i \alpha_i t_i \mathcal{K}(\vec{s}, \vec{x_i}) - b \tag{1}$$

$$\mathcal{K}(\vec{x}, \vec{y}) = \vec{x}^T \cdot \vec{y} \tag{2}$$

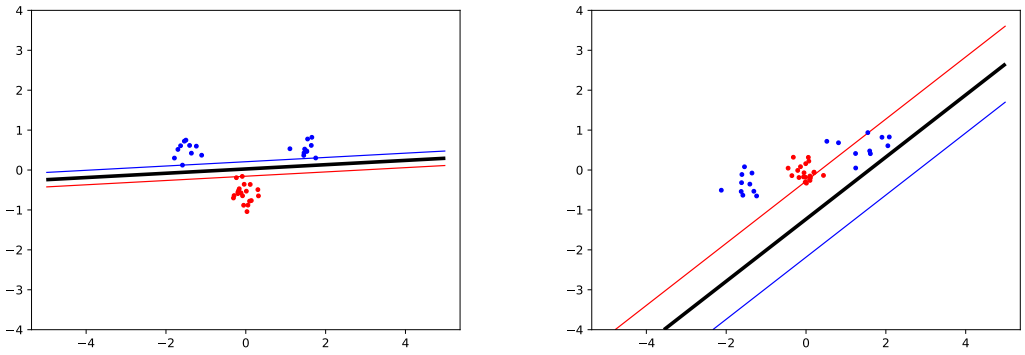
$$\mathcal{K}(\vec{x}, \vec{y}) = (\vec{x}^T \cdot \vec{y} + 1)^p \tag{3}$$

$$\mathcal{K}(\vec{x}, \vec{y}) = e^{-\frac{||\vec{x}-\vec{y}||^2}{2\sigma^2}} \tag{4}$$

$$\min_{\vec{w},b,\vec{\xi}} ||\vec{w}|| + C \sum_i \vec{\xi} \tag{5}$$

1 Exploring and Reporting

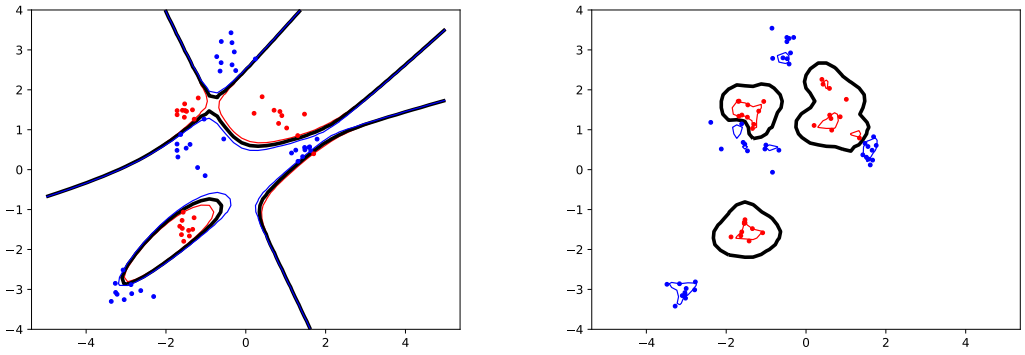
1. With linearly separable clusters, we always get a 'success' from the minimize function. But it is not always the case for a complex data set. Here we created an aligned data set, such that it is not separable with one line. When we only set a lower bound, and we use 'None' for the upper bound, we get a 'failure'. Indeed, there is no solution the minimize function could find. However, if we allow some mistakes, with an upper bound C, the minimize function finally succeeds to find a linear separation, even though it's not a good separation. (see Figure 1)



(a) (b)

Figure 1: (a) Separable data set, linear kernel (b) Not separable data set, linear kernel

2. With more complex kernel, we see that we can classify very well complex data sets. (see Figure 2)

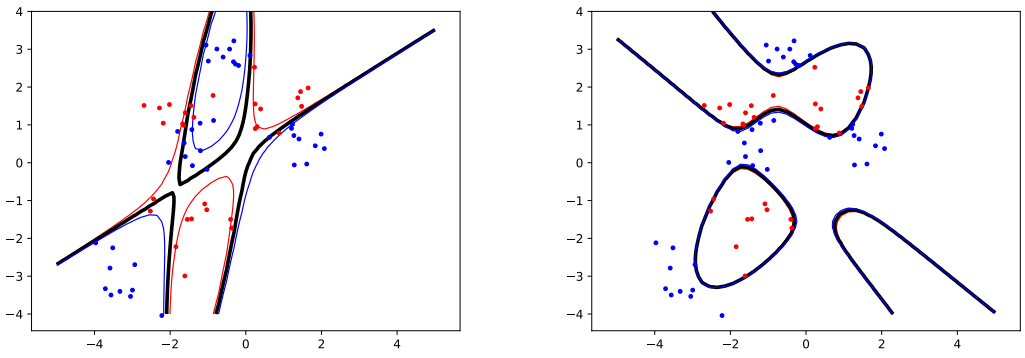


(a) (b)

Figure 2: (a) Polynomial kernel, $p = 4$ (b) RBF kernel, $\sigma = 0.2$

3. For the polynomial kernel, by increasing the parameter p we allow more complex curves. We can now classify more complex data points. Here we used close clusters with high variance, which make them difficult to classify. We see that when we increase the parameter p, we get a more accurate classification. With $p = 3$ we can see some data misclassification, that disappear when $p = 4$. We

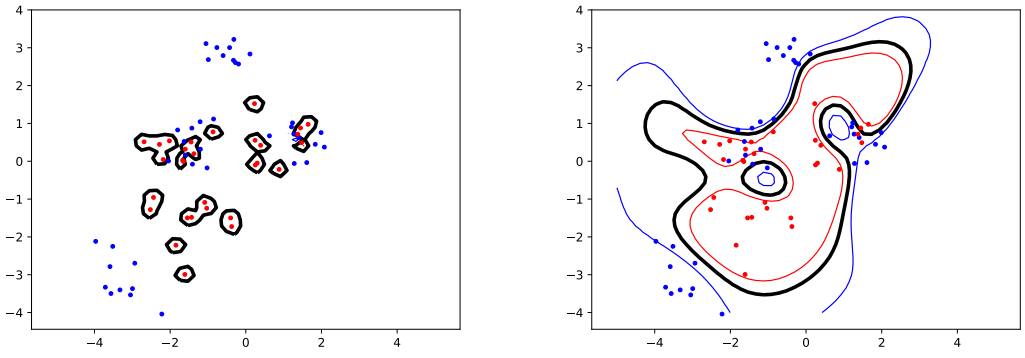
also observe that with $p = 4$, there is a new red zone that contains no data. Thus we can say that increasing p lowers the variance, but may increase the bias. (see Figure 3)



(a) (b)

Figure 3: (a) Polynomial kernel, $p = 3$ (b) $p = 4$

For the RBF kernel, low σ gives a very local classification, almost for each data point, which is in fact overfitting (high biased). High σ gives a more global classification, but then we find out some mistakes. Thus increasing σ increases the variance and lowers the bias. (Figure 4)

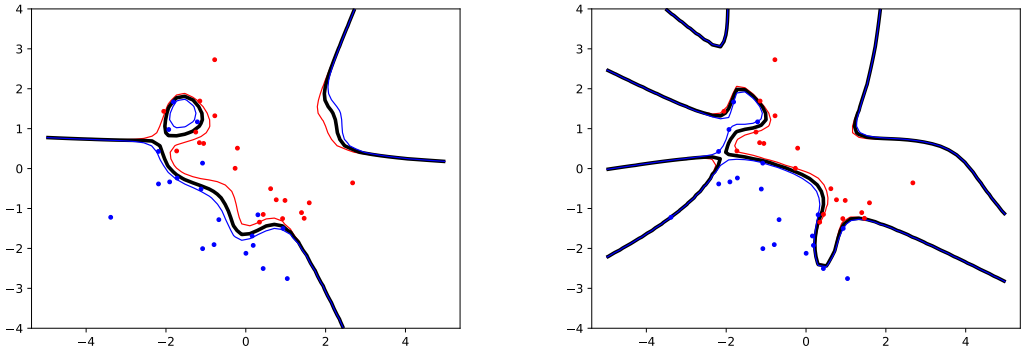


(a) (b)

Figure 4: (a) RBF kernel, $\sigma = 0.1$ (b) $\sigma = 1$

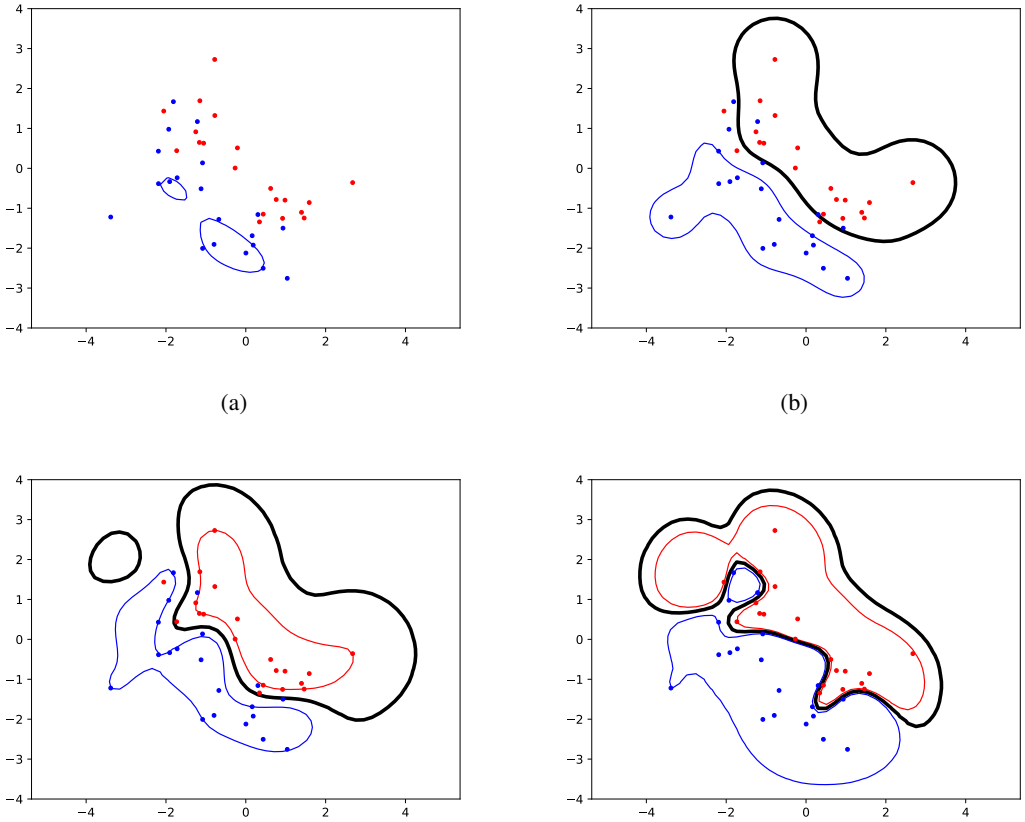
4. The slack parameter C modulates the allowed mistakes. A high C means few mistakes allowed, and a low C means high mistakes allowed. We see on Figures 5 and 6 that we have misclassifications with low C models, and no errors with high C models.

5. For complex data sets, the choice between slack variables and complex kernel is very specific to each case. The basic linear kernel is very easy to implement, but we see that in some cases its result may be very bad, even with a lot of slack. But we also may have a data set well described by a very complex kernel, which can be approximated with a simple kernel and slack variables. It wouldn't be perfect, but much faster and easier to compute. It's up to us to make the trade-off between the accuracy and the complexity.



(a) (b)

Figure 5: (a) Polynomial kernel, $C = 0.1$ (b) $C = 100$



(a) (b)

(c) (d)

Figure 6: (a) RBF kernel, $C = 0.1$ (b) $C = 1$ (c) $C = 10$ (d) $C = 100$