

Support Vector Machine Assignment

Mohamedhakim Elakhrass

15/07/2016

Contents

1	Objectives	1
2	Simple 2 Gaussian	1
3	The Support Vector Machine	1

1 Objectives

The objectives of this assignment are to learn and work with real life applications of SVM.

2 Simple 2 Gaussian

This sections looks at two simulated datasets. The datasets are created using the MatLab *randn()* function. One dataset is centered around (1,1) and the other (-1,1)

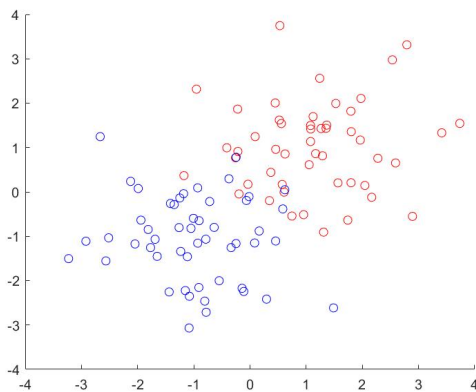


Figure 1: Two Simulated Datasets.

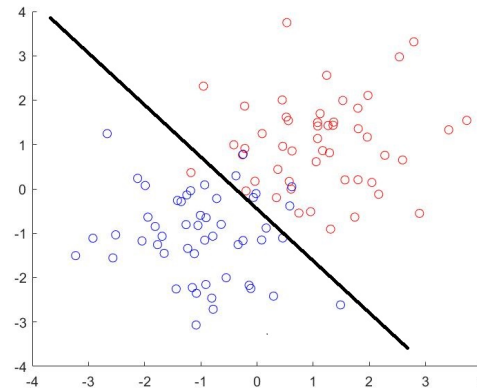


Figure 2: Two Simulated Datasets with Optimal Classifier.

Given this figure, can you make a geometric construction using lines to estimate the optimal classifier? Under which conditions do you think this construction is optimal/-valid?

Figure 1:a is the output of the simulated datasets. As shown in figure 1:b it is possible to show an optimal classifier. This classifier is known as the Bayes Classifier. A test observation is assigned with predictor vector x_0 to the class j for which

$$Pr(Y = j|X = x_0)$$

is largest. The classifier is optimal because it produces the lowest possible error rate and allows for some overlap. The classifier is valid because the underlying distribution of the dataset is known. This falls into the special case $\Sigma_{xx1} = \Sigma_{xx2} = \Sigma_{xx}$, the covariance matrices are equal and the decision boundary is linear.

3 The Support Vector Machine

This section will deal with an online demo of a linear and none linear SVM. It will be used to learn the intuition behind changing SVM parameters and changes in the dataset.

Adjust the existing datasets to have at least 10 data points for each class. What do you observe when you are adding data points to the classes? How drastically can classification boundaries change.

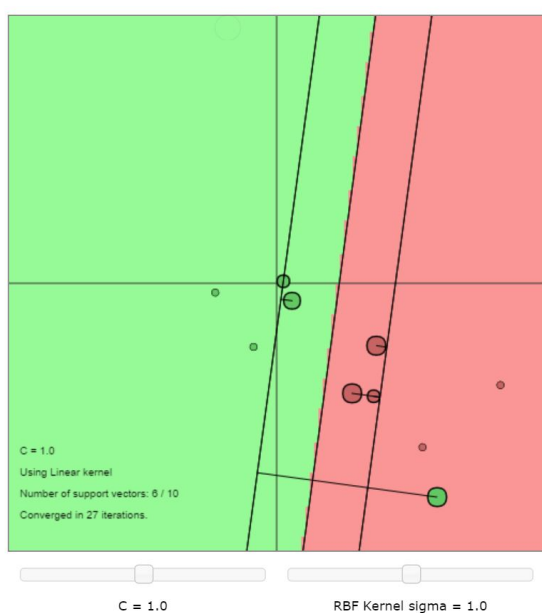


Figure 3: Default Linear Kernel.

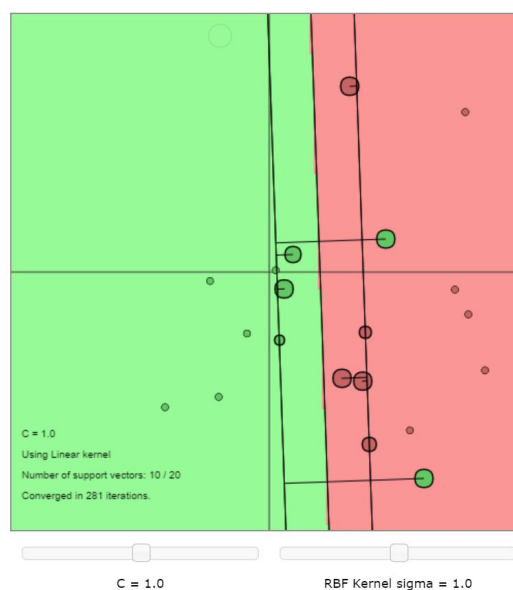


Figure 4: 10 Data Point Linear Kernel.

Data points added inside the margin drastically change the decision boundary and become support vectors. Data points added to the side of the opposing color are also automatically support vectors but remain misclassified. Data points added to the same side as its own color have very little effect on the decision boundary, although the closer to the boundary the larger the effect.

What if you add an outlying datapoint which lies on the wrong side of the classification hyperplane? How does it affect the classification hyperplane?

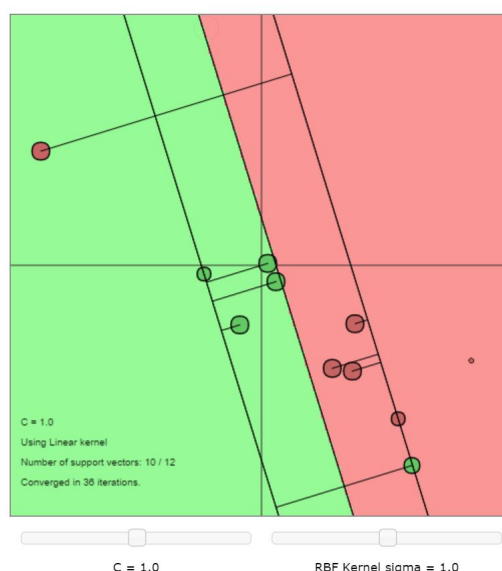


Figure 5: Outlier data point

A data point added to the wrong side of the boundary changes the direction of the hyperplane towards that point. If too many points are added the classifier misclassified all points of the opposing class.

Try different values of C regularization hyperparameter. How does it affect the classification outcome? What is the role of it?

Using the initial dataset the effects of the regularization hyperparameter can be seen. The parameter control the slack of the SVM model. When C is high there is less tolerance for misclassification and therefore a smaller margin. When C is large there is higher tolerance for misclassification and a larger margin.

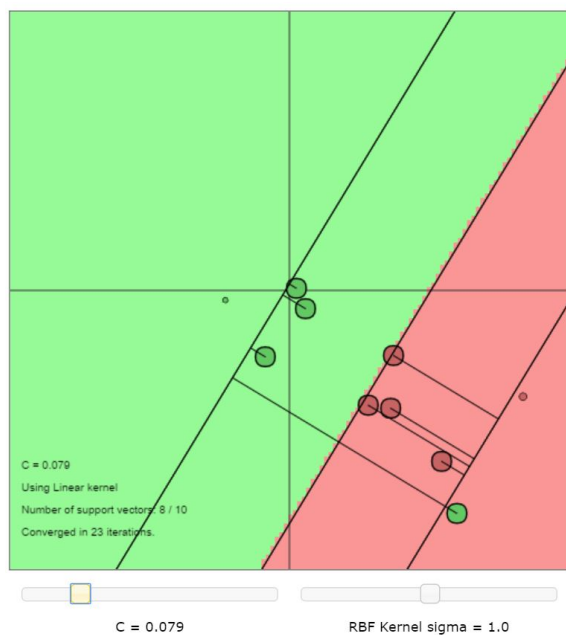


Figure 6: $C = .079$

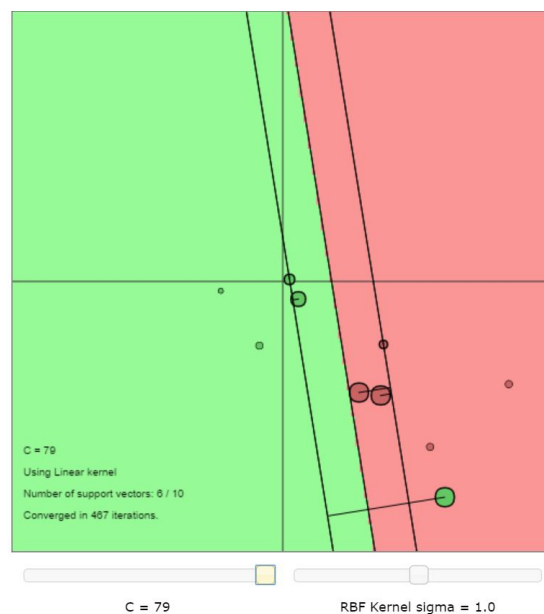
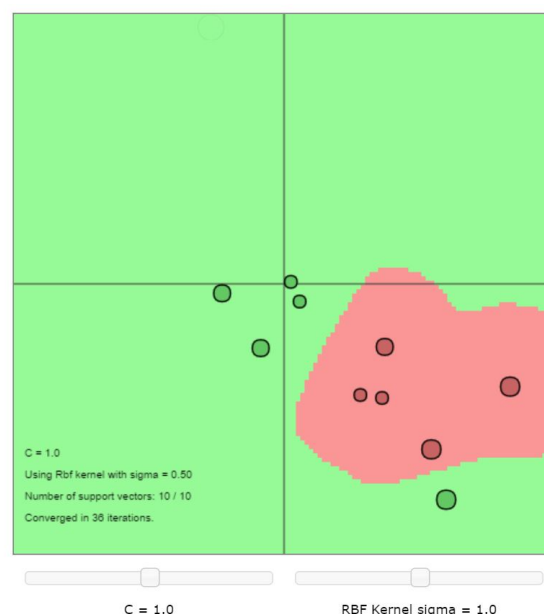


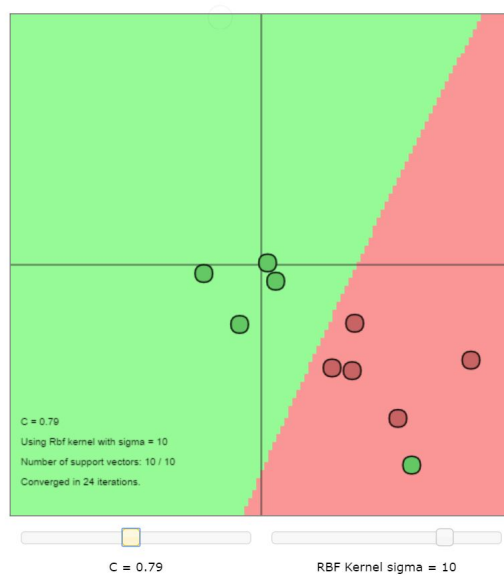
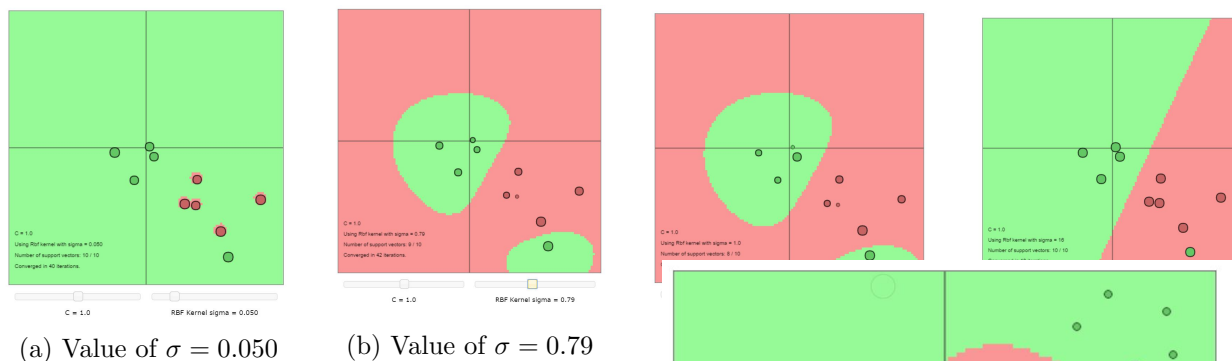
Figure 7: $C = 79$

Follow the instructions and switch back to RBF kernel by toggling the k button. Compare to the classification outcome of the linear case. Try to change the RBF kernel sigma hyperparameter. What is your intuition? How does it affect the classification boundaries? Now try to change both hyperparameters. What is the right choice of those if your data is almost linearly separable?

The default RBF kernel has no misclassification as opposed to the linear classifier. This is because the data set is not linearly separable, therefore the linear classifier performs poorly. The RBF kernel has the ability to wrap around the data. Small values of σ however will lead to a risk of over fitting and generalize well. Misclassification is only seen at the higher end of σ . The higher σ is the more linear the decision boundary.

The right choice to make when the data is almost linear separable is the linear kernel. As stated above, data that is almost linearly separable but has some overlap can be best classified using Bayes Optimal classifier. This will lead to a linear decision boundary with some misclassification. If an RBF kernel is chosen you may have good predictions on your training data but the model will not generalize to the test data. Using the RBF kernel you can get a near linear decision boundary with $c = 79$ and $\sigma = 10$



Figure 9: Value of $\sigma = 0.79$ Figure 10: How σ effects RBF

Create a linearly non-separable dataset with an overlapping region between classes (e.g. similar to the previous Gaussian clouds). Give comments on the role of the chosen kernel, the regularization parameter (C) and the kernel parameter (sigma)

