FACULTY OF BIOSCIENCE ENGINEERING
Masters of Bioinformatics
Support Vector Machines

Support Vector Machine Assignment

Mohamedhakim Elakhrass

15/07/2016

Contents

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)

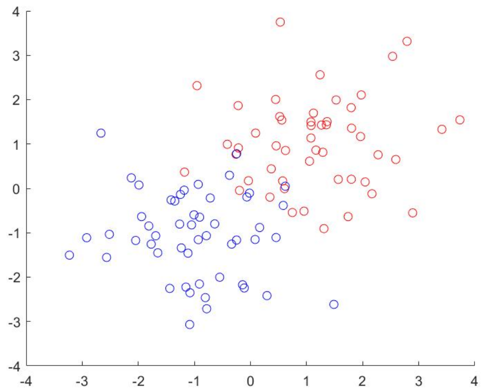


Figure 1: Two Simulated Datasets.

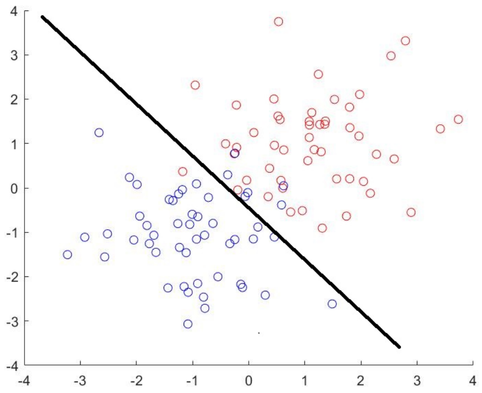


Figure 2: Two Simulated Datasets with Optimal Classifier.

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.

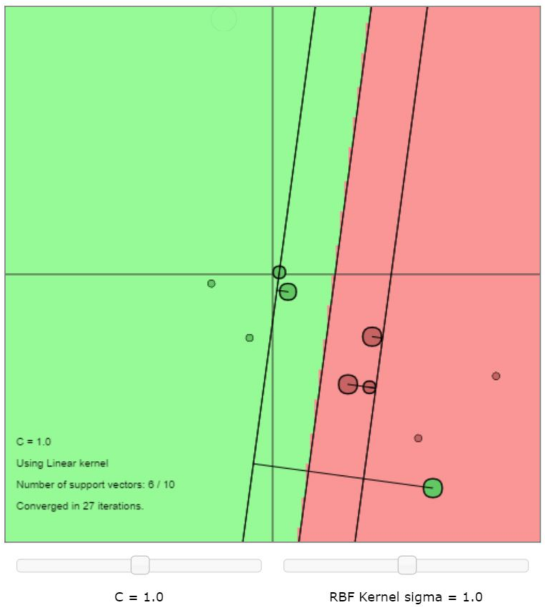

C = 1.0
Using Linear kernel
Number of support vectors: 6 / 10
Converged in 27 iterations.

Figure 3: Default Linear Kernal.

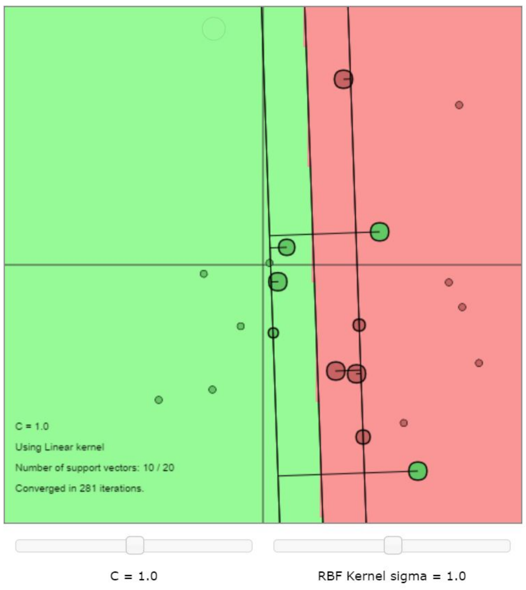

C = 1.0
Using Linear kernel
Number of support vectors: 10 / 20
Converged in 281 iterations.

Figure 4: 10 Data Point Linear Kernal.

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.

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 boundary? How does it affect the classification hyperplane?

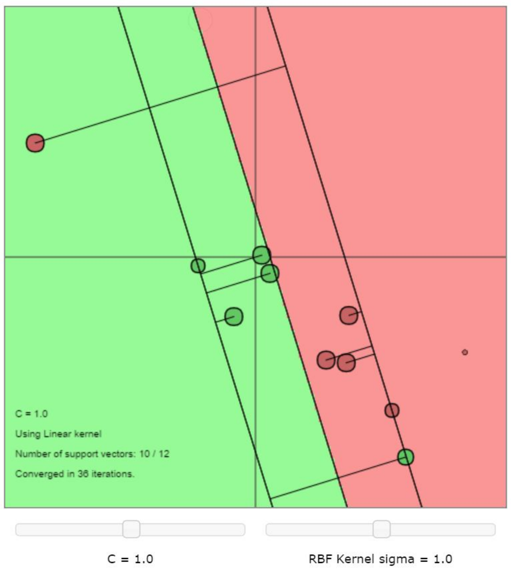
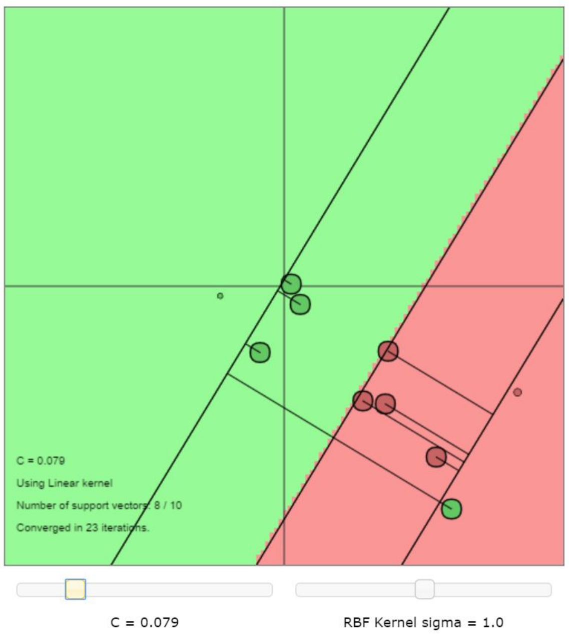

C = 1.0
Using Linear kernel
Number of support vectors: 10 / 12
Converged in 36 iterations.

Figure 5: Outlier data point

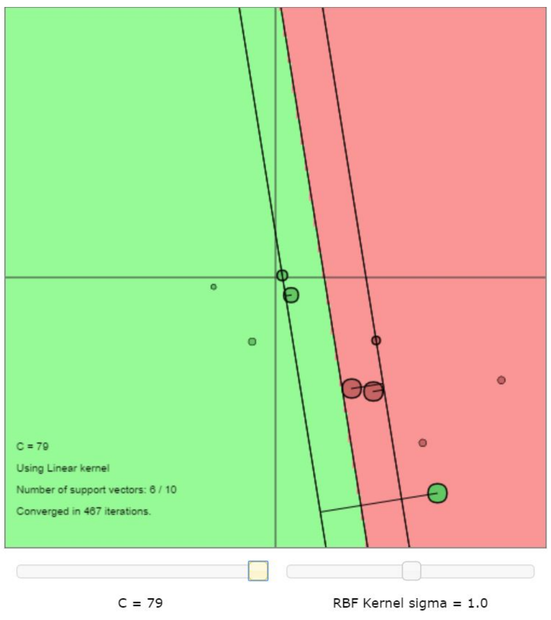
A data point added to the wrong side of the boundary changes the direction of the hyperplane towards that point. If too many points are added the classifier misclassified all points of the opposing class.

Try different values of C regularization hyperparameter. How does it affect the classification outcome? What is the role of it?
Using the intial dataset the effects of the regularization hyperparemter 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.


C = 0.079
Using Linear kernel
Number of support vectors: 8 / 10
Converged in 23 iterations.

C = 0.079
RBF Kernel sigma = 1.0

Figure 6: C = .079


C = 79
Using Linear kernel
Number of support vectors: 6 / 10
Converged in 467 iterations.

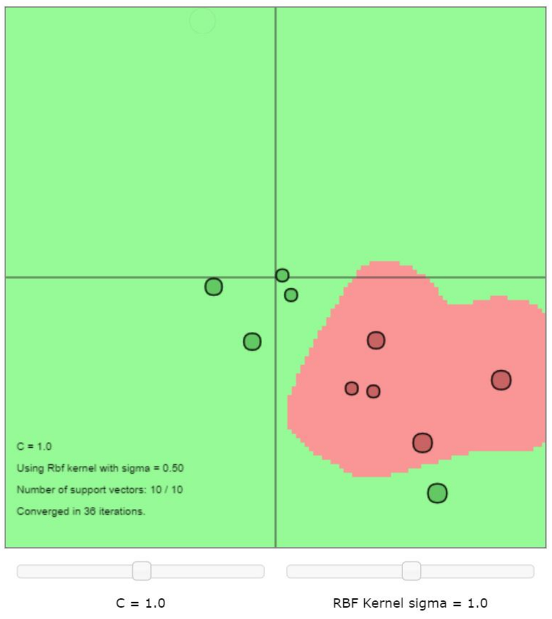
C = 79
RBF Kernel sigma = 1.0

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 hyper parameter. What is your intuition? How does it affect the classification boundaries? Now try to change both hyper parameters. What is the right choice of those if your data is almost linearly separable?

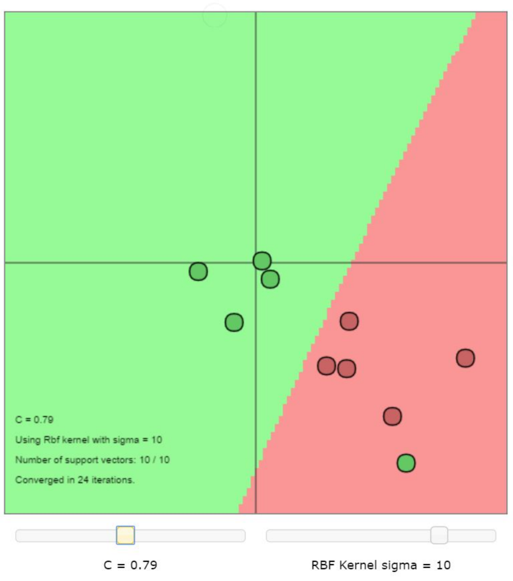
The default RBF kernal has no misclassification as opposed to the linear classifier. This is because the data set is not linearly separable, therefore the linear classifier performs poorly. The RBF kernal has the ability to wrap around the data. Small values of σ however will lead to a risk of over fitting 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 validation data but the model will not generalize to the test data. Using the RBF kernal you can get a near linear decision boundary with $c = 79$ and $\sigma = 10$


C = 1.0
Using Rbf kernel with sigma = 0.50
Number of support vectors: 10 / 10
Converged in 36 iterations.

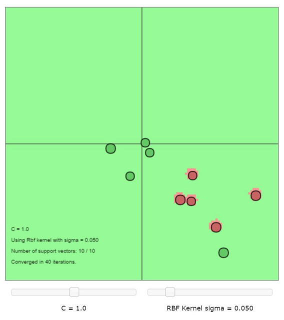
C = 1.0

RBF Kernel sigma = 1.0



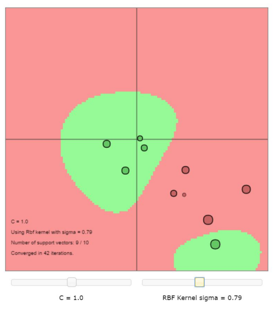
```
C = 0.79
Using Rbf kernel with sigma = 10
Number of support vectors: 10 / 10
Converged in 24 iterations.
```

Figure 9: Value of $\sigma = 0.79$



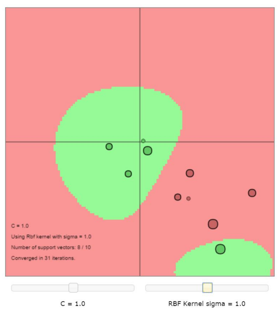
```
C = 1.0
Using Rbf kernel with sigma = 0.050
Number of support vectors: 10 / 10
Converged in 40 iterations.
```

(a) Value of $\sigma = 0.050$



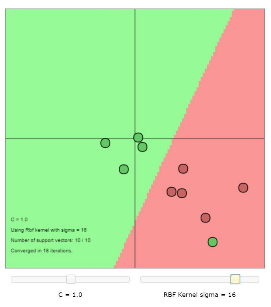
```
C = 1.0
Using Rbf kernel with sigma = 0.79
Number of support vectors: 9 / 10
Converged in 42 iterations.
```

(b) Value of $\sigma = 0.79$



```
C = 1.0
Using Rbf kernel with sigma = 1.0
Number of support vectors: 8 / 10
Converged in 31 iterations.
```

(c) Value of $\sigma = 1$



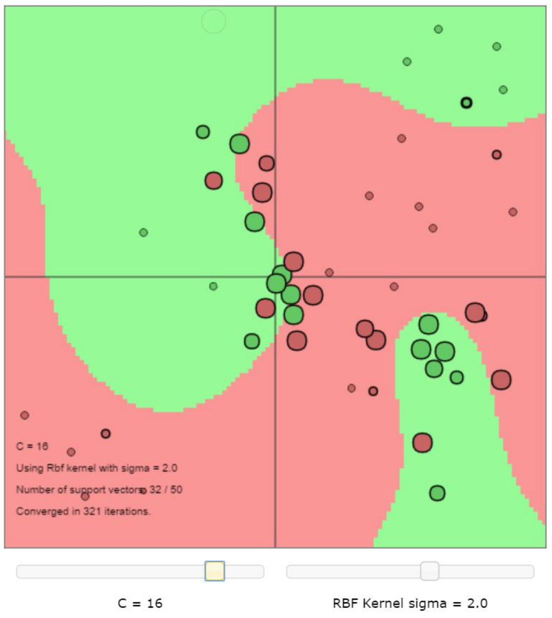
```
C = 1.0
Using Rbf kernel with sigma = 16
Number of support vectors: 10 / 10
Converged in 18 iterations.
```

(d) Value of $\sigma = 16$

Figure 10: How σ effects RBF classifier with C = 1.0

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)

For this set 25 red and 25 green points were created. These points were made to create a dataset as described above. An RBF kernel was chosen because the dataset is not linearly separable so using a linear kernel would be impossible. The parameters were chosen to appropriately classify the data points but also to leave some slack for misclassification. Due to the overlapping regions, if everything was perfectly classified on this validation set you would most likely see an over fitting when you introduce your test set. The C and sigma values were adjusted accordingly. The C value was chosen relatively high to allow for the misclassification. The Sigma is relatively small because a large sigma would cause a near linear decision boundary which would be useless in this case.



```
C = 16
Using Rbf kernel with sigma = 2.0
Number of support vectors 32 / 50
Converged in 321 iterations.
```

C = 16

RBF Kernel sigma = 2.0

What is the role of Support Vectors? Change the data-sets to make the number of support vectors increase/decrease. When does a particular datapoint become a Support Vector?
As a property of this being a quadratic programming problem that is a dual problem many resulting values of α_k are equal to zero in the classifier $y(x) = sign[\sum_{k=1}^{\#SV} \alpha_k y_k x_k^T x + b]$ This is a property called sparseness. The sum is of the none zero α_k which are now support vectors. Geometrically they are located close to the decision boundary. In a linear kernel all vectors inside or at the edge of the margin are support vectors as well as misclassified data points. For the RBF kernel x is replaced by $\varphi(x)$. However this can be infinite dimensional. Therefore the problem can only be solved in the dual.

When does the corresponding importance of a Support Vector change?
The importance of support vector changes when the influence it has on the decision boundary is altered. This can happen when changing the c parameter.

4 Using LSSVM

In this section the matlab package LSSVM will be used to explore the Iris Dataset.

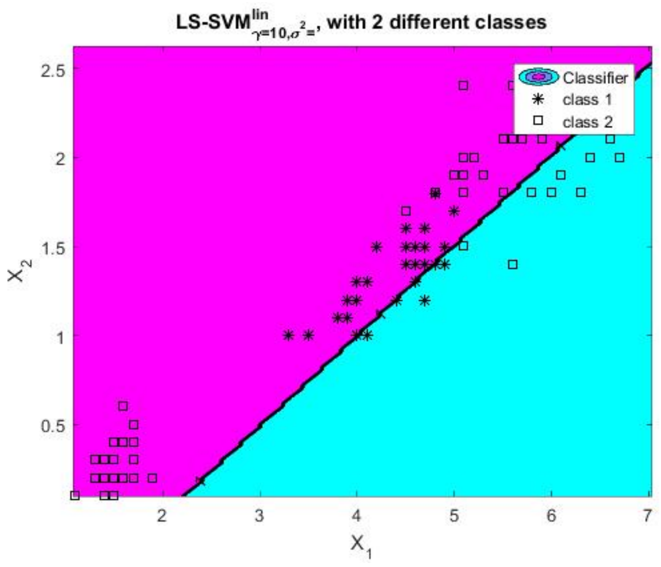
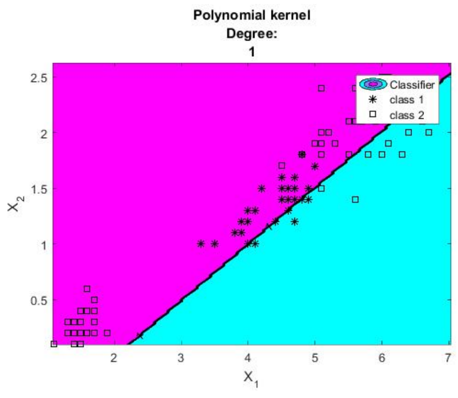


Figure 12: Value of $\sigma = 0.79$

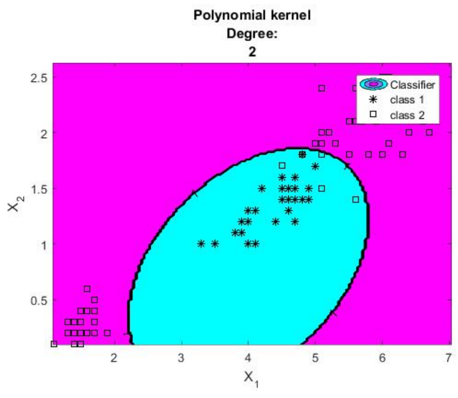
What is the performance on the test set Xt and Yt?
The performance shows a 55% error rate.

What happens when you are changing the degree of a polynomial kernel? Explain the obtained results. Does it correspond to the changes in sigma hyperparameter of the RBF kernel in the previous example?

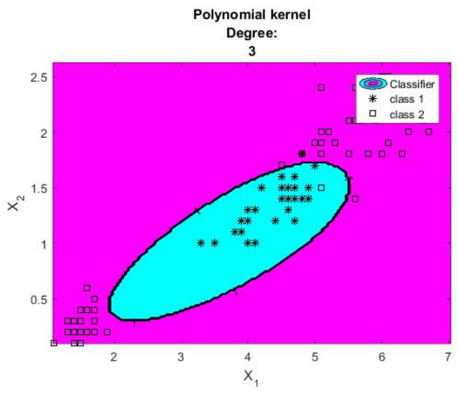
The first degree polynomial is just a linear kernel, therefore it has the same results. The 2nd degree has 4 misclassification. The 3rd degree has 3 misclassification. The 4th degree has 3 and the 5th has 3. Increasing the degree of polynomial is a little like decreasing sigma of the RBF. It creates a less linear boundary that is more prone to over fitting.



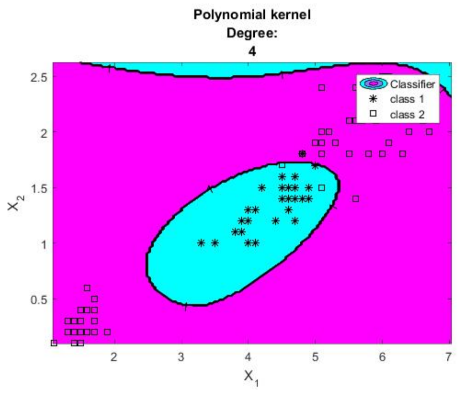
(a) Degree 1



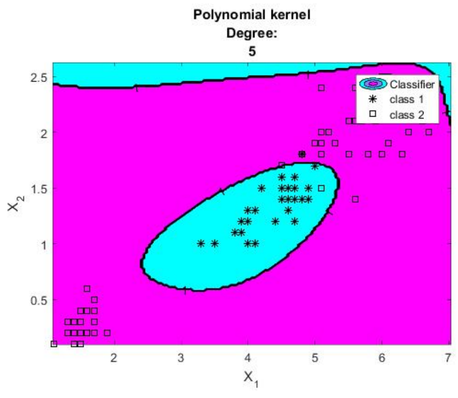
(b) Degree 2



(c) Degree 3



(d) Degree 4



(e) Degree 5

Figure 13: Varying Degrees of the Polynomial Kernel

Try out a good range of different sig2s as kernel parameters. For each individual value of sig2, the corresponding LS-SVM is evaluated on the test set. Make a figure of the sig2s with their corresponding test set performance. Fix a reasonable choice for the sig2 of the RBF kernel and again compare a range of gams by plotting the corresponding test set performances. What is a good range for gam?

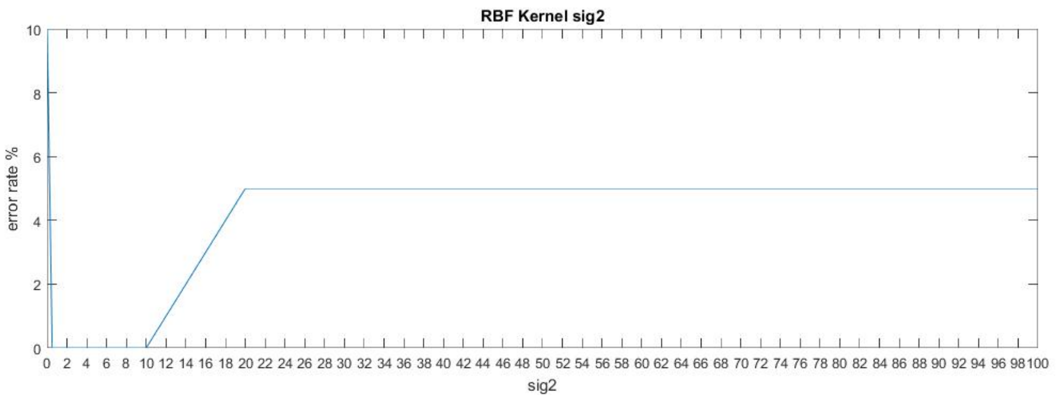


Figure 14: Error rate VS. Sig2

A range used for both sig2 and gamma was 0.01, 0.5, 4, 10, 20, 100.Between point 5 and 14 seem to have the best range for sig2. With a sig2 of 12 the best gam is between 2 and 8.

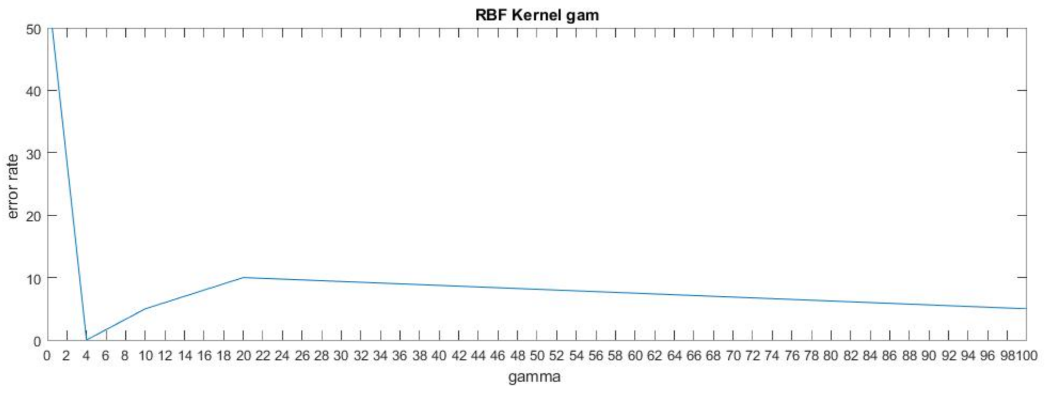


Figure 15: Error rate VS. gam

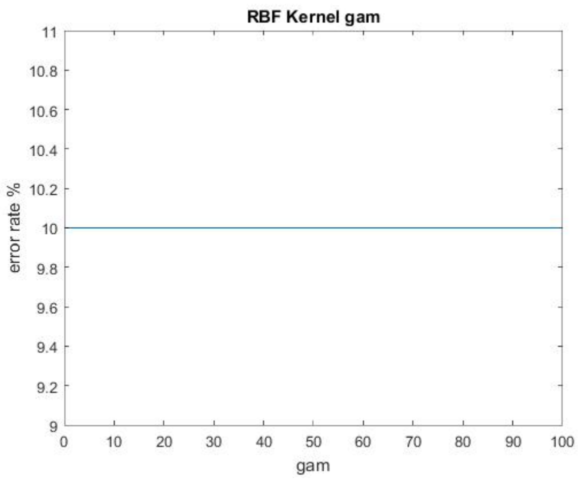
Comparing to the results from the sample script you get a very similar outlook. You need a good balance between gam and sig to balance between over fitting and misclassification.

5 Choice of Hyper-parameters

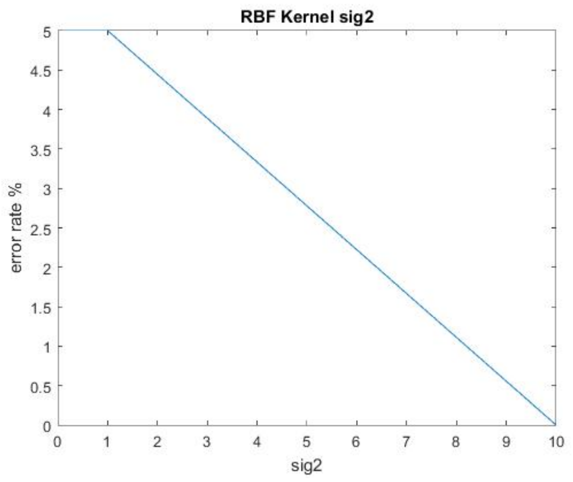
In this section the intuition developed from the previous sections will be used to start using autotuning algorithms.

Motivate why this validation set used to optimize a number of(tuning-) parameters cannot used again to measure the final model.

Validation set is used to optimize parameters to make sure that the model can generalize well. If someone was to just train the model on one data set and validate it on another without using a test set the model will likely "memorize the data" (difference between 'dumb' AI and smart 'AI'). Therefore a set of data that was not touched during the tuning process must be the final test to see if the model can generalize.



(a) Error for Gam values 1, 10, 100 with sig2 = 1



(b) Error for Sig values 0.1,1,10 with gam = 10

Figure 16: Tuning

Across the different values the error does not really change. It stays steady across all values of gam and sig2 except for when cig2 is 10, then you see a 50% decrease in error.

Think about a clear and intuitive way to represent this technique. Why should one prefer this method over a simple validation? Change crossvalidate procedure for leaveoneout (removing the 10). Is it giving better results? In which cases one would prefer each?

They both perform exactly the same. Leave one out would be prefered when the data set is very large. Leave one out is computationally expensive. LOOCV also gives you a hire chance of over fitting since your model has seen all of your data.

Try to change different parameters like 'csa' (Coupled Simulated Annealing) vs. 'ds' (Randomized Directional Search) and 'simplex' (Nelder-Mead method) vs. 'gridsearch' (brute force gridsearch). What differences do you observe? Why in some cases the obtained hyperparameters differ a lot?

Since there is an element of randomness each run will yield different results. Since this is finding hyper parameters the problem in non-convex and local minima can occur.

The ROC is often used to asses a model. It is important not to use this on the validation set. It is for the same reason you do not judge a model based on the validation set. Models that memorize their data are not good models. Models must generalize well.

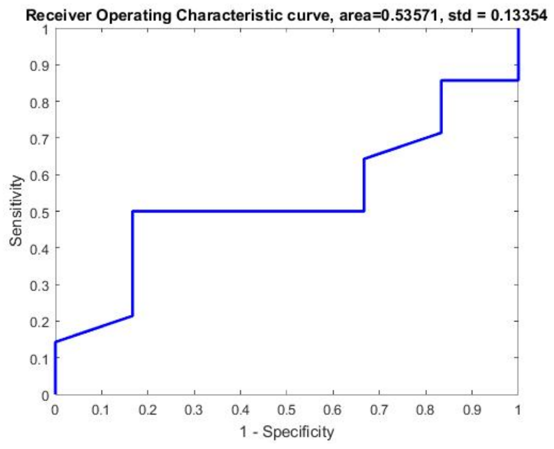


Figure 17: Error rate VS. gam