FACULTY OF BIOSCIENCE ENGINEERING
Masters of Bioinformatics
Support Vector Machines

KU LEUVEN

Support Vector Machine Assignment

Mohamedhakim Elakhrass

15/07/2016

Contents

1 Assignment 1

1.1 Objectives

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

1.2 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)

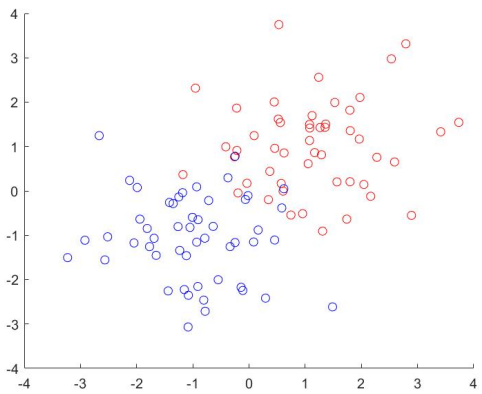


Figure 1: Two Simulated Datasets.

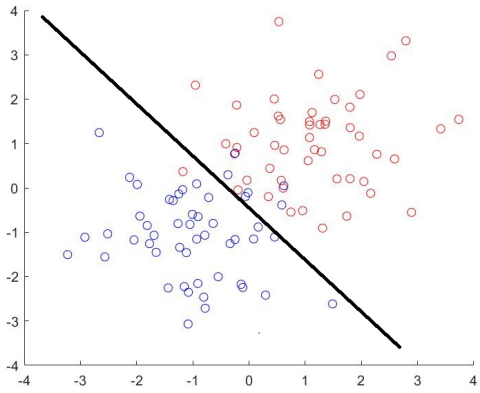


Figure 2: Two Simulated Datasets with Optimal Classifier.

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.

1.3 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.

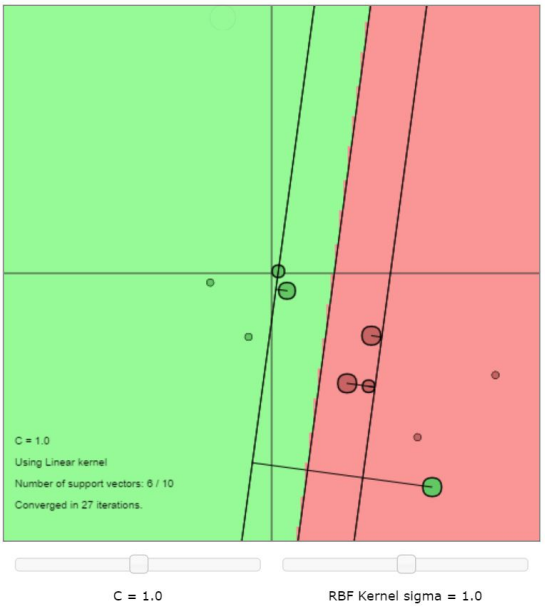


Figure 3: Default Linear Kernal.

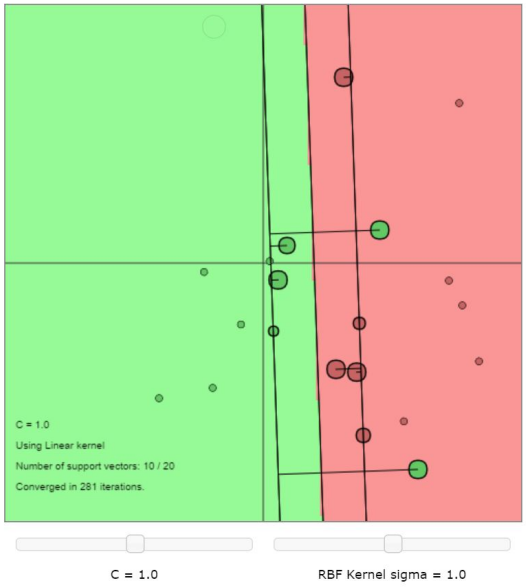


Figure 4: 10 Data Point Linear Kernal.

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?

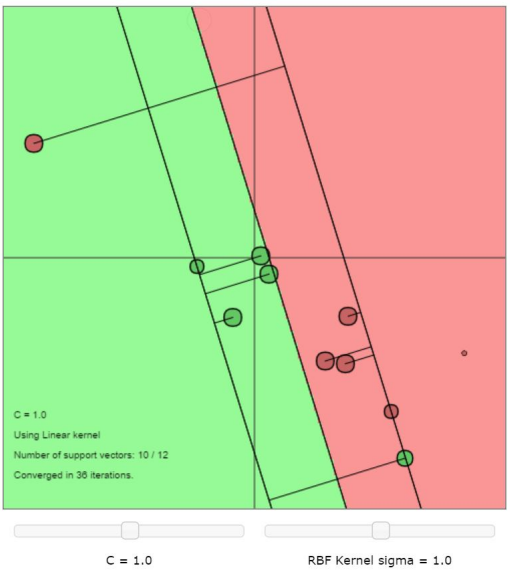


Figure 5: Outlier data point

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.

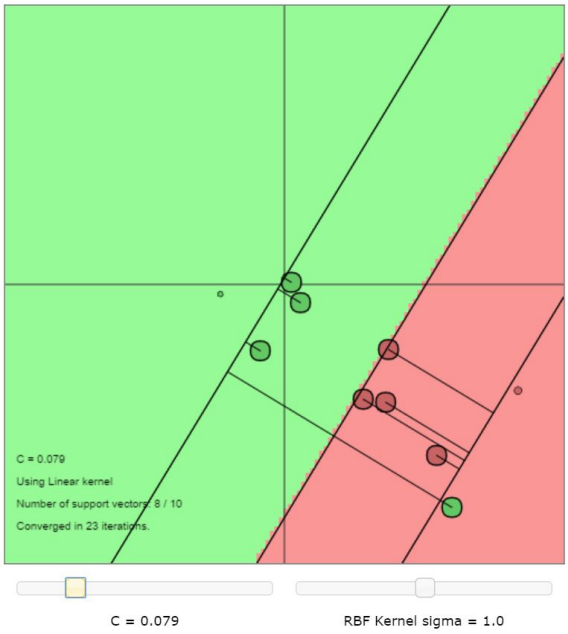


Figure 6: C = .079

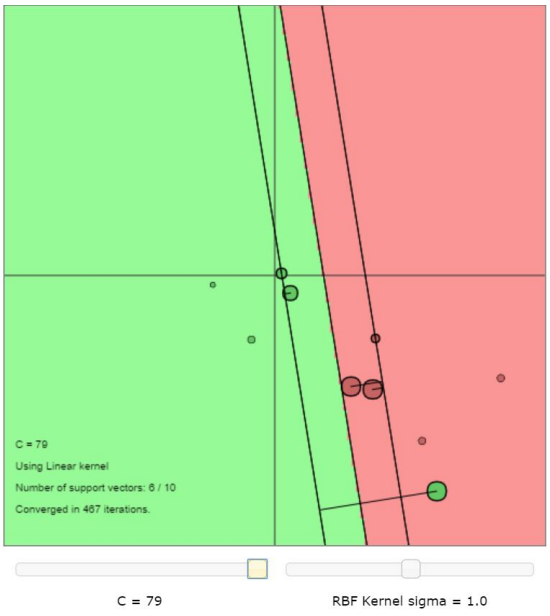
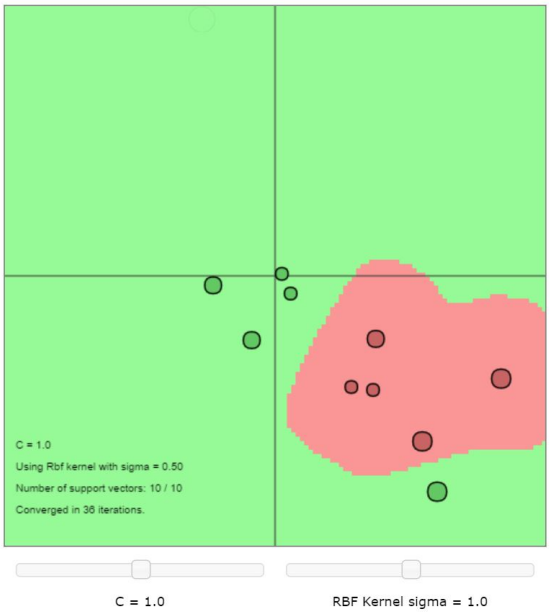


Figure 7: C = 79

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 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 $\sigma = 10$



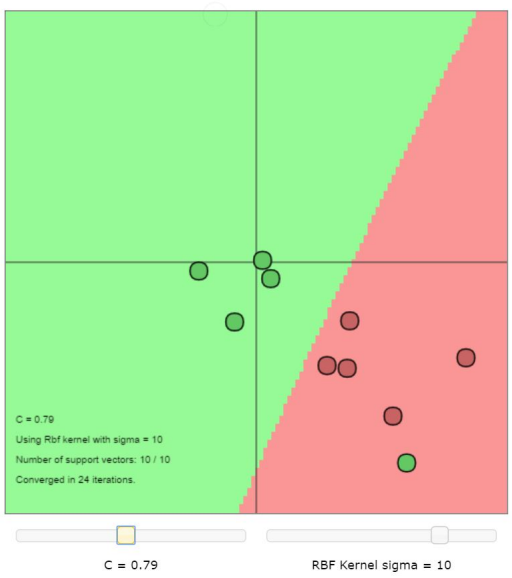
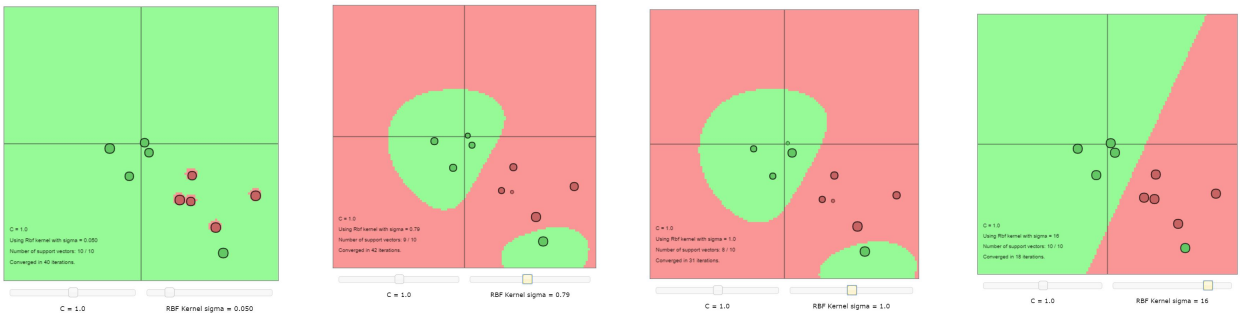


Figure 9: Value of $\sigma = 0.79$

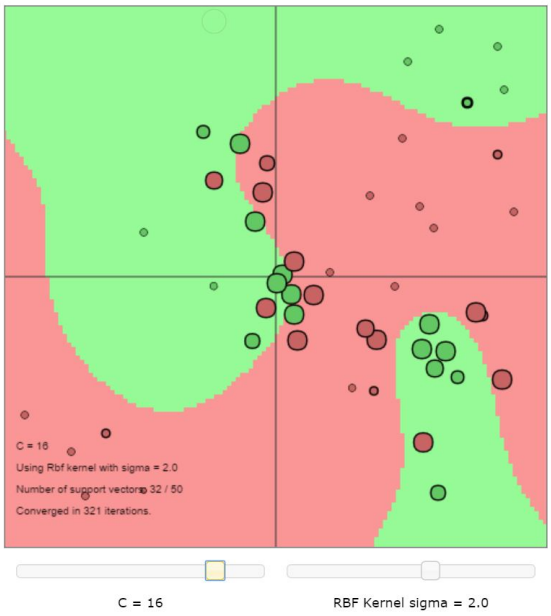


(a) Value of $\sigma = 0.050$ (b) Value of $\sigma = 0.79$ (c) Value of $\sigma = 1$ (d) Value of $\sigma = 16$

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.



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.

1.4 Using LSSVM

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

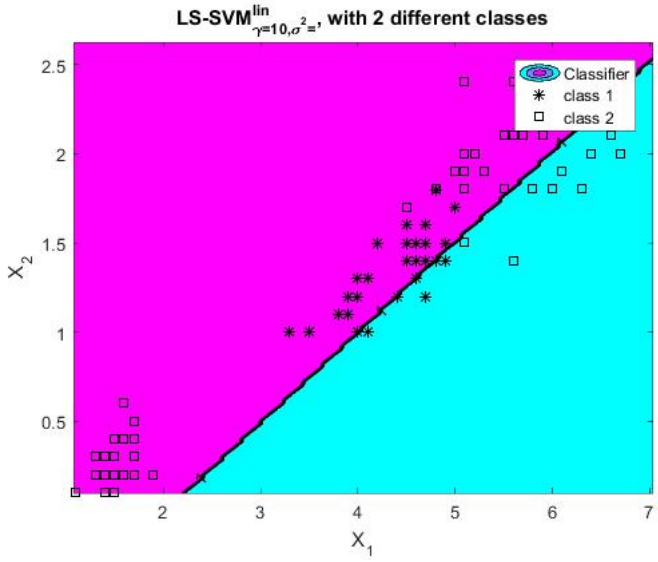
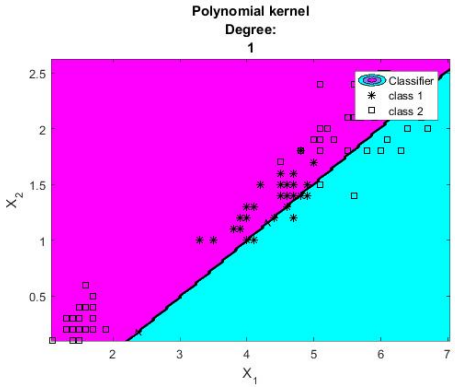


Figure 12: Value of $\sigma = 0.79$

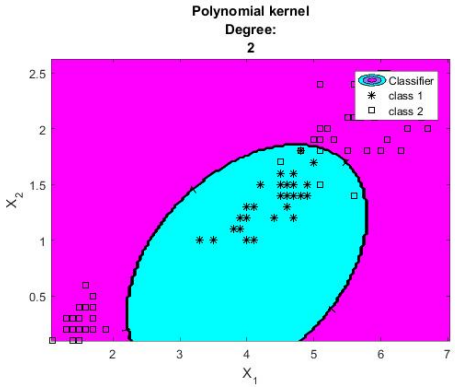
What is the performance on the test set Xt and Yt?
The performance shows a 55% error rate.

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?

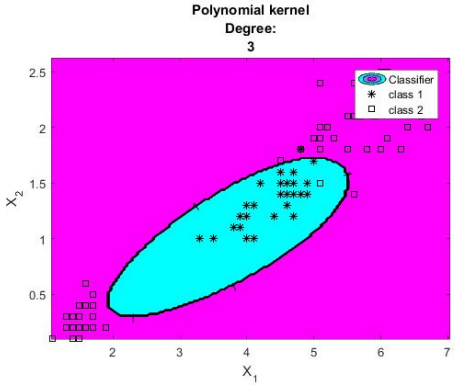
The first degree polynomial is just a linear kernel, therefore it has the same results. The 2nd degree has 4 misclassification. The 3rd degree has 3 misclassification. The 4th degree has 3 and the 5th has 3. Increasing the degree of polynomial is a little like decreasing sigma of the RBF. It creates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



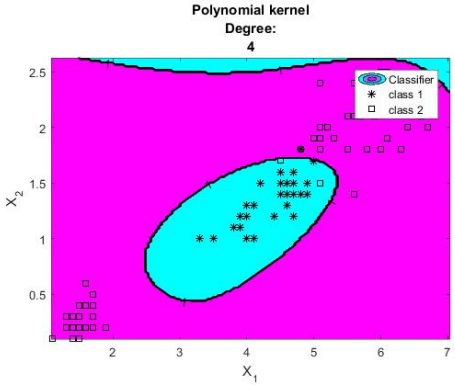
(a) Degree 1



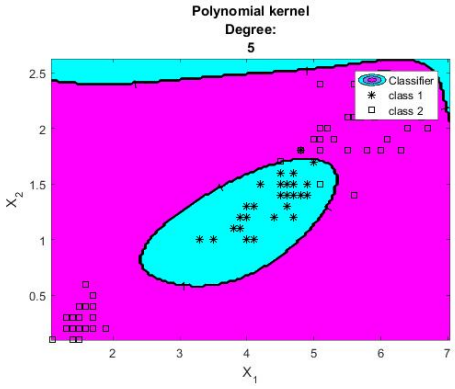
(b) Degree 2



(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?

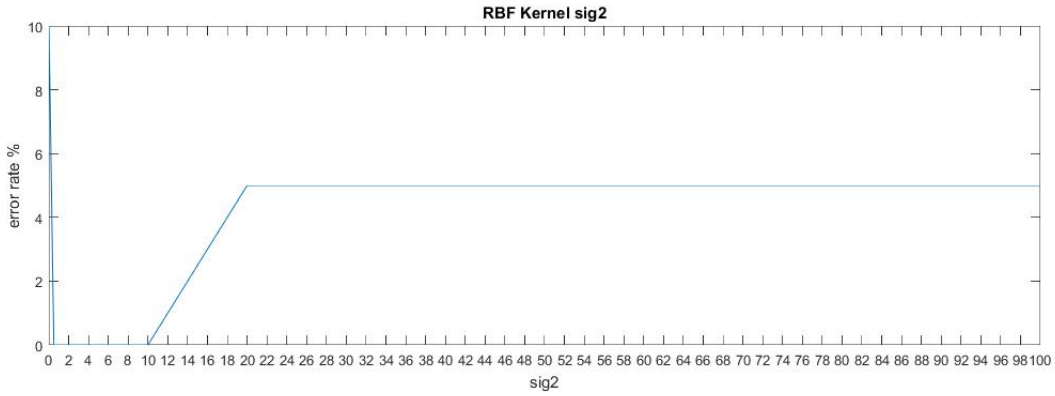


Figure 14: Error rate VS. Sig2

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.

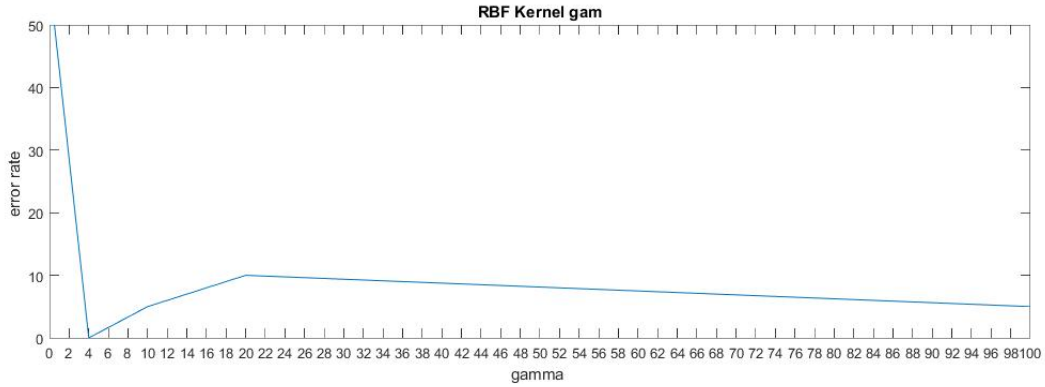


Figure 15: Error rate VS. gam

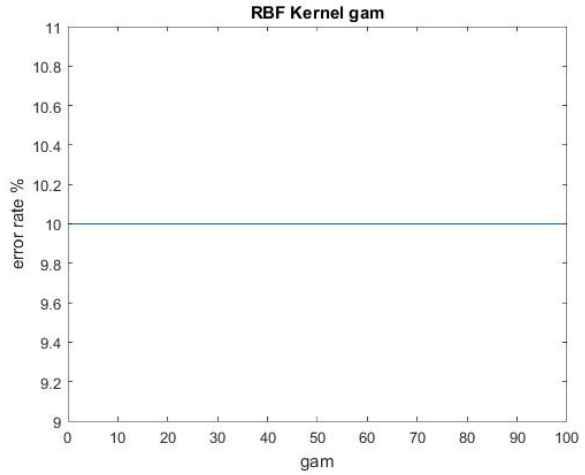
Comparing to the results from the sample script you get a very similar outlook. You need a good balance between gam and sig to balance between over fitting and misclassification.

1.5 Choice of Hyper-parameters

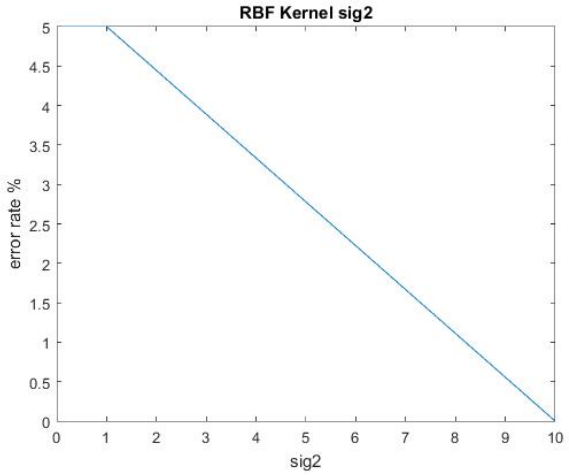
In this section the intuition developed from the previous sections will be used to start using autotuning algorithms.

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.

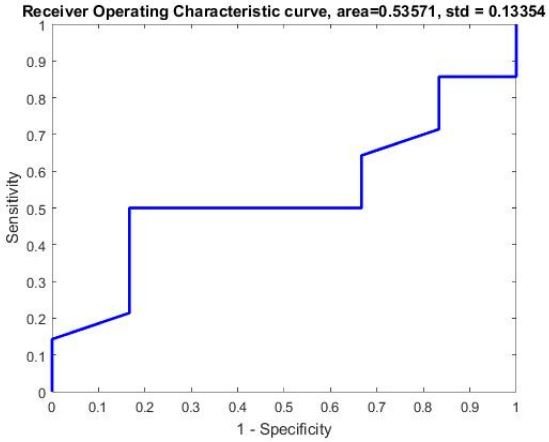


Figure 17: Error rate VS. gam

1.6 Playing around with data

This section will be about apply techniques learned in a more tutorial based setting to real date sets. A set of 5 questions will be answered about each data set.

1.6.1 Ripley Dataset

1. Look at the plotted data. What seems to be important properties of the data?

The data is split into four distinct groups with slight over lap.

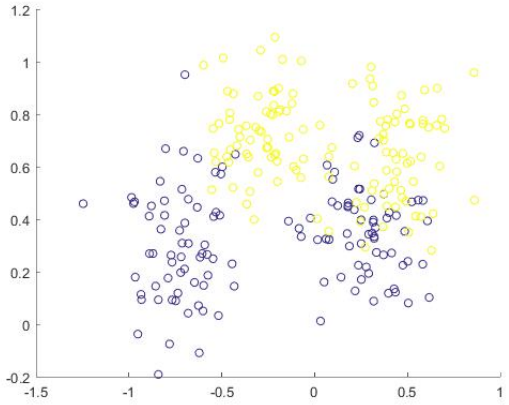


Figure 18: data visualized

2. Try a linear model. Do you think its sufficient?
The linear model cannot model this data at all. This is because the data is not linearly separable.

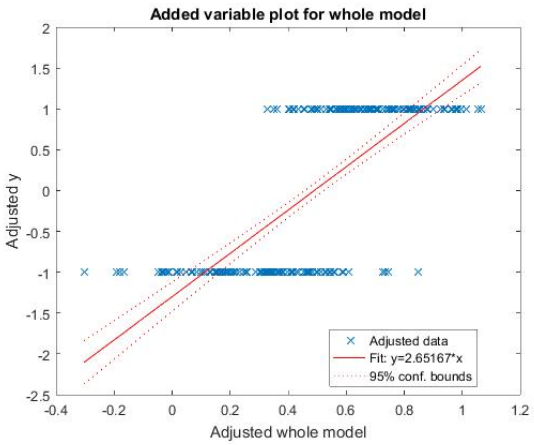


Figure 19: Linear Model for ripley

3. Try the RBF kernel and tune its parameter. Is the tuning acceptable? Run several times the routine tunelssvm. Do the tuned parameters gam and sig2 change? What about the performance?

```
  Determine initial tuning parameters for simplex...: # cooling cycle(s) 1
 |-                                                  -|
 ************************************************** done

1. Coupled Simulated Annealing results:  [gam]         368.7525
                                         [sig2]        1.0644
                                         F(X)=         0.076

TUNELSSVM: chosen specifications:
2. optimization routine:           gridsearch
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   18.3591      0.08737
   cost of starting values:           0.092
   time needed for 1 evaluation (sec):0.17188
   limits of the grid:   [gam]         18.3591077      54727.7288
                         [sig2]        0.08737      12.9669

OPTIMIZATION IN LOG SCALE...

grain =

     7

FIRST ITERATION:
   X=5.5768    0.062397 ,F(X)=0.077;
ITERATION: 2
  dF=0.001, dX=0.49927, X=5.0848   -0.022728 ,F(X)=0.076;
Obtained hyper-parameters: [gamma sig2]: 161.5526     0.9775281
```

(a) CSA and gridsearch

```
1. Coupled Simulated Annealing results:  [gam]         0.22111
                                         [sig2]        0.28298
                                         F(X)=         0.079

TUNELSSVM: chosen specifications:
2. optimization routine:           simplex
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   0.22111     0.28298

Iteration   Func-count     min f(x)     log(gamma)     log(sig2)     Procedure

     1           3     7.900000e-02      -0.3091        -1.2624      initial
     2           5     7.900000e-02      -0.3091        -1.2624      contract inside
     3           7     7.900000e-02      -0.3091        -1.2624      contract inside
Optimisation terminated sucessfully (TolFun criterion)

Simplex results:
X=0.734122   0.282975, F(X)=7.900000e-02

Obtained hyper-parameters: [gamma sig2]: 0.73412     0.28298
```

(b) CSA and simplex

```
1. Directional Search (DFO)    results:  [gam]         6.9568
                                         [sig2]        0.50812
                                         F(X)=         0.078

TUNELSSVM: chosen specifications:
2. optimization routine:           gridsearch
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   0.34636    0.041709
   cost of starting values:           0.093
   time needed for 1 evaluation (sec):0.125
   limits of the grid:   [gam]         0.34635641      1032.4739
                         [sig2]        0.041709      6.1901

OPTIMIZATION IN LOG SCALE...

grain =

     7

FIRST ITERATION:
   X=6.9397     0.15629 ,F(X)=0.076;
ITERATION: 2
  dF=0, dX=2.0323, X=8.9632     0.34506 ,F(X)=0.076;
Obtained hyper-parameters: [gamma sig2]: 7810.5382      1.4120692
```

(c) DS and grid search

```
1. Directional Search (DFO)    results:  [gam]         12.941
                                         [sig2]        0.66404
                                         F(X)=         0.081

TUNELSSVM: chosen specifications:
2. optimization routine:           simplex
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   12.941     0.664043

Iteration   Func-count     min f(x)     log(gamma)     log(sig2)     Procedure

     1           3     8.100000e-02      3.7604        -0.4094      initial
     2           5     8.100000e-02      3.7604        -0.4094      contract inside
     3           7     8.100000e-02      3.7604        -0.4094      contract inside
     4           9     8.100000e-02      3.7604        -0.4094      contract inside
     5          11     8.100000e-02      3.7604        -0.4094      contract outside
optimisation terminated sucessfully (TolFun criterion)

Simplex results:
X=42.965768   0.664043, F(X)=8.100000e-02

Obtained hyper-parameters: [gamma sig2]: 42.9658     0.664043
```

(d) DS and simplex

Figure 20: Different tuning methods

5. Judge your final model. Is the methodology perfectly suited for this data-set? It is well suited but with such a high AUC it may be prone to overfitting.

1.6.2 Diabetes Dataset

1. Look at the plotted data. What seems to be important properties of the data?

This data is almost completely indistinguishable from each other. There is almost complete overlap between the two data sets.

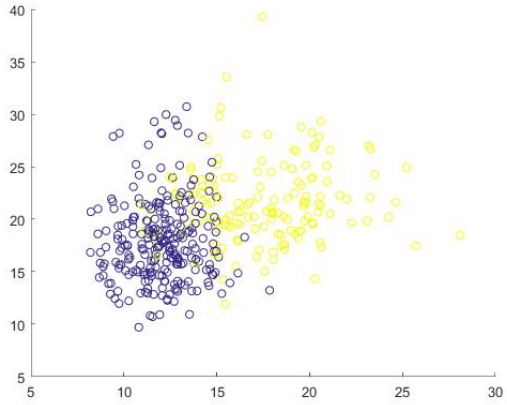


Figure 21: Data viz for breast

2. Try a linear model. Do you think its sufficient?

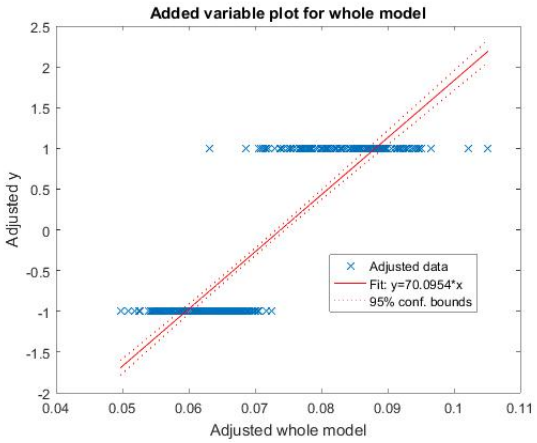


Figure 22: Linear Model for breast

The linear model is not sufficient.

3. Try the RBF kernel and tune its parameter. Is the tuning acceptable? Run several times the routine tunelssvm. Do the tuned parameters gam and sig2 change? What about the performance?

```
1. Coupled Simulated Annealing results:  [gam]         13.411
                                         [sig2]        0.86553
                                         F(X)=         0.077

TUNELSSVM: chosen specifications:
2. optimization routine:           gridsearch
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   0.66769    0.071047
   cost of starting values:           0.086
   time needed for 1 evaluation (sec):0.15625
   limits of the grid:   [gam]         0.66769398      1990.3677
                         [sig2]        0.0710469       10.5443

OPTIMIZATION IN LOG SCALE...

grain =

     7

FIRST ITERATION:
   X=2.2627    -0.14441 ,F(X)=0.076;
ITERATION: 2
  dF=0.001, dX=1.0403, X=1.2784    -0.48095 ,F(X)=0.075;
Obtained hyper-parameters: [gamma sig2]: 3.5909      0.6182
```

(a) CSA and gridsearch

```
1. Coupled Simulated Annealing results:  [gam]         1.2873
                                         [sig2]        0.54541
                                         F(X)=         0.079

TUNELSSVM: chosen specifications:
2. optimization routine:           simplex
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   1.2873     0.54541

Iteration   Func-count    min f(x)    log(gamma)    log(sig2)    Procedure

     1          3     7.900000e-02      0.2525       -0.6062     initial
     2          5     7.900000e-02      0.2525       -0.6062     contract inside
     3          7     7.900000e-02      0.2525       -0.6062     contract inside
     4          8     7.900000e-02      0.2525       -0.6062     reflect
     5         12     7.900000e-02      0.2525       -0.6062     shrink
     6         14     7.900000e-02      0.2525       -0.6062     contract inside
     7         16     7.900000e-02      0.2525       -0.6062     contract inside
     8         17     7.900000e-02      0.2525       -0.6062     reflect
     9         19     7.900000e-02      0.2525       -0.6062     contract inside
    10         23     7.900000e-02      0.2525       -0.6062     shrink
    11         25     7.900000e-02      0.2525       -0.6062     contract outside
optimisation terminated sucessfully (TolFun criterion)

Simplex results:
X=1.287277   0.545411, F(X)=7.900000e-02

Obtained hyper-parameters: [gamma sig2]: 1.2873      0.54541
```

(b) CSA and simplex

```
1. Directional Search (DFO)     results:  [gam]         1.5062
                                          [sig2]        0.38047
                                          F(X)=         0.077

TUNELSSVM: chosen specifications:
2. optimization routine:           gridsearch
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   0.074988    0.031231
   cost of starting values:           0.083
   time needed for 1 evaluation (sec):0.10938
   limits of the grid:   [gam]         0.07498804      223.5362
                         [sig2]        0.031231        4.6351

OPTIMIZATION IN LOG SCALE...

grain =

     7

FIRST ITERATION:
   X=-2.5904     -1.7997 ,F(X)=0.077;
ITERATION: 2
  dF=0, dX=1.7056, X=-0.90502     -1.5377 ,F(X)=0.077;
Obtained hyper-parameters: [gamma sig2]: 0.40453      0.21488
```

(c) DS and grid search

```
1. Directional Search (DFO)     results:  [gam]         0.51039
                                          [sig2]        0.58979
                                          F(X)=         0.08

TUNELSSVM: chosen specifications:
2. optimization routine:           simplex
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   0.51039     0.58979

Iteration   Func-count    min f(x)    log(gamma)    log(sig2)    Procedure

     1          3     8.000000e-02      -0.6726       -0.5280     initial
     2          5     8.000000e-02      -0.6726       -0.5280     contract outside
     3          7     8.000000e-02      -0.6726       -0.5280     contract outside
     4          9     8.000000e-02      -0.6726       -0.5280     contract inside
     5         10     8.000000e-02      -0.6726       -0.5280     reflect
optimisation terminated sucessfully (TolFun criterion)

Simplex results:
X=0.510395   0.589793, F(X)=8.000000e-02

Obtained hyper-parameters: [gamma sig2]: 0.51039      0.58979
```

(d) DS and simplex

Figure 23: Different tuning methods

5. Judge your final model. Is the methodology perfectly suited for this data-set?

1.6.3 Breast Cancer Dataset

1. Look at the plotted data. What seems to be important properties of the data?

The datasets are completely intertwined.

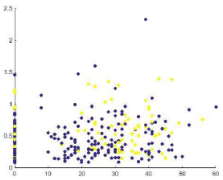


Figure 24: viz of diabetes

2. Try a linear model. Do you think its sufficient?

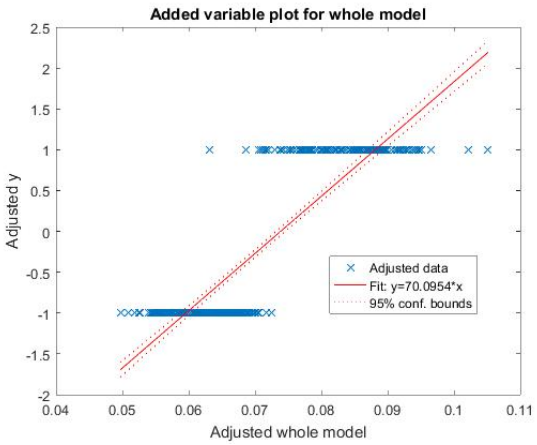


Figure 25: viz of diabetes

3. Try the RBF kernel and tune its parameter. Is the tuning acceptable? Run several times the routine tunelssvm. Do the tuned parameters gam and sig2 change? What about the performance?

```
1. Coupled Simulated Annealing results:  [gam]         17.9642
                                         [sig2]        98.2897
                                         F(X)=         0.0175

 TUNELSSVM: chosen specifications:
 2. optimization routine:           gridsearch
    cost function:                  crossvalidatelssvm
    kernel function                 RBF_kernel


 3. starting values:                   0.89439      8.0681
    cost of starting values:           0.025
    time needed for 1 evaluation (sec):0
    limits of the grid:   [gam]         0.8943871      2666.1304
                          [sig2]        8.0681084      1197.4135

OPTIMIZATION IN LOG SCALE...

grain =

     7

FIRST ITERATION:
   X=1.2217      3.7546 ,F(X)=0.0175;
ITERATION: 2
  dF=0, dX=1.6499, X=2.8682      3.6487 ,F(X)=0.0175;
Obtained hyper-parameters: [gamma sig2]: 17.6061      38.4262
```

(a) CSA and gridsearch

```
1. Coupled Simulated Annealing results:  [gam]         117.4299
                                         [sig2]        146.5861
                                         F(X)=         0.015

TUNELSSVM: chosen specifications:
2. optimization routine:           simplex
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   117.4299      146.5861

Iteration   Func-count    min f(x)    log(gamma)    log(sig2)    Procedure

     1           3     1.500000e-02      4.7658        4.9876      initial
     2           5     1.500000e-02      4.7658        4.9876      contract inside
     3           6     1.500000e-02      4.7658        4.9876      reflect
     4          10     1.500000e-02      4.7658        4.9876      shrink
     5          12     1.500000e-02      4.7658        4.9876      contract inside
optimisation terminated sucessfully (TolFun criterion)

Simplex results:
X=117.429890   146.586113, F(X)=1.500000e-02

Obtained hyper-parameters: [gamma sig2]: 117.4299      146.5861
```

(b) CSA and simplex

```
1. Coupled Simulated Annealing results:  [gam]         73.3297
                                         [sig2]        98.6477
                                         F(X)=         0.015

 TUNELSSVM: chosen specifications:
 2. optimization routine:           gridsearch
    cost function:                  crossvalidatelssvm
    kernel function                 RBF_kernel


 3. starting values:                   3.6509      8.0975
    cost of starting values:           0.025
    time needed for 1 evaluation (sec):0.0625
    limits of the grid:   [gam]         3.65087158      10883.0948
                          [sig2]        8.0974991      1201.7754

OPTIMIZATION IN LOG SCALE...

grain =

     7

FIRST ITERATION:
   X=2.6283      3.7582 ,F(X)=0.015;
ITERATION: 2
  dF=0, dX=1.0903, X=3.7118      3.8795 ,F(X)=0.015;
Obtained hyper-parameters: [gamma sig2]: 40.9281      48.3992
```

(c) DS and grid search

```
1. Coupled Simulated Annealing results:  [gam]         94.0476
                                         [sig2]        41.9444
                                         F(X)=         0.015

TUNELSSVM: chosen specifications:
2. optimization routine:           simplex
   cost function:                  crossvalidatelssvm
   kernel function                 RBF_kernel


3. starting values:                   94.0476      41.9444

Iteration   Func-count    min f(x)    log(gamma)    log(sig2)    Procedure

     1           3     1.500000e-02      4.5438        3.7363      initial
     2           5     1.500000e-02      4.5438        3.7363      contract outside
     3           9     1.500000e-02      4.5438        3.7363      shrink
     4          11     1.500000e-02      4.5438        3.7363      contract inside
     5          15     1.500000e-02      4.5438        3.7363      shrink
     6          16     1.500000e-02      4.5438        3.7363      reflect
     7          18     1.500000e-02      4.5438        3.7363      contract outside
optimisation terminated sucessfully (TolFun criterion)

Simplex results:
X=94.047602   41.944447, F(X)=1.500000e-02

Obtained hyper-parameters: [gamma sig2]: 94.0476      41.9444
```

(d) DS and simplex

Figure 26: Different tuning methods

2 Assignment 2

2.1 The support vector machine for regression

SVMs can also be used to solve regression problems. In the following exercises that means modeling 20 data points. The higher the e values the less support vectors are used. Sparsity comes into play when you show the model new data. A model where the bound is inf and the e is equal to 0 will generalize very poorly even if it fits well the validation set. In the default setup you do not have sparseness. All data points are support vectors, which takes away the benefit of SVMs, non-zero support vectors describing the model instead of all data points.

Construct a data-set of around 20 data-points. Which kernel is best suited for your data-set?

The initial 20 point dataset was created using the uiregress interface. The best kernel was the Exponential RBF.

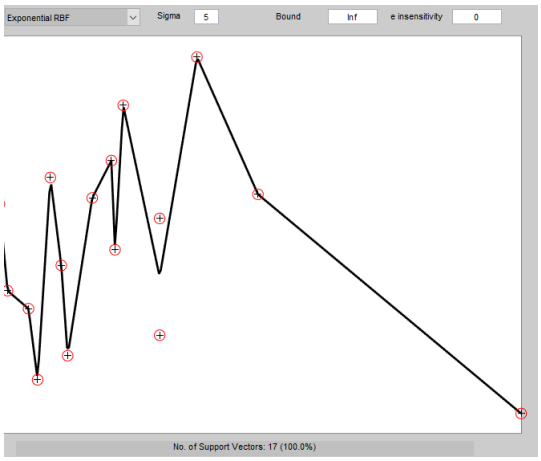
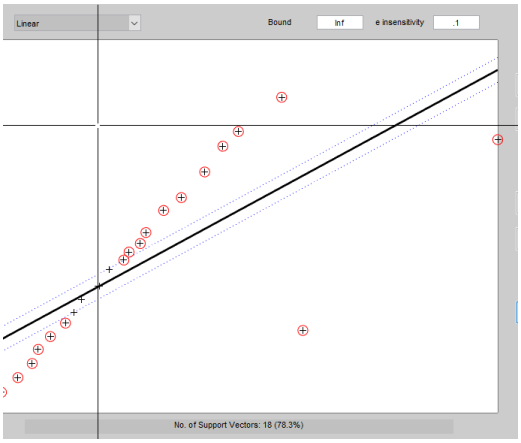
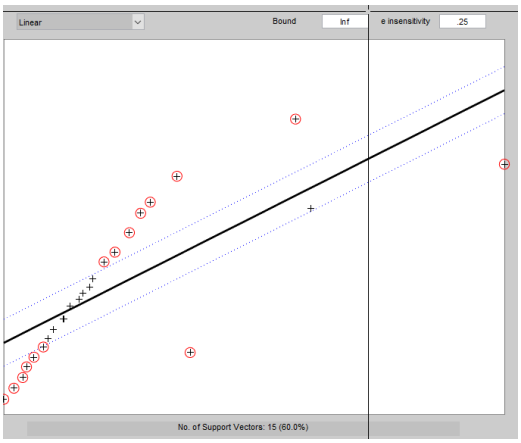


Figure 27: 20 Data points

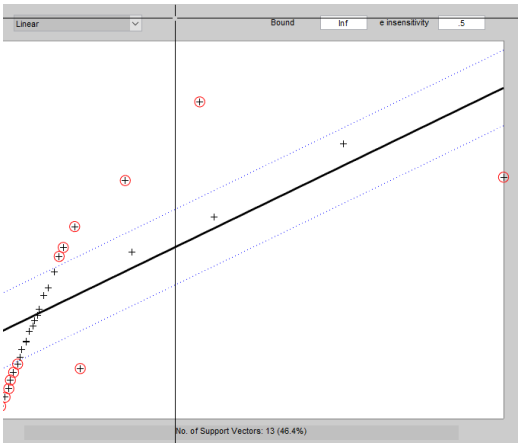
The the higher the e value the further it is from being a good model.



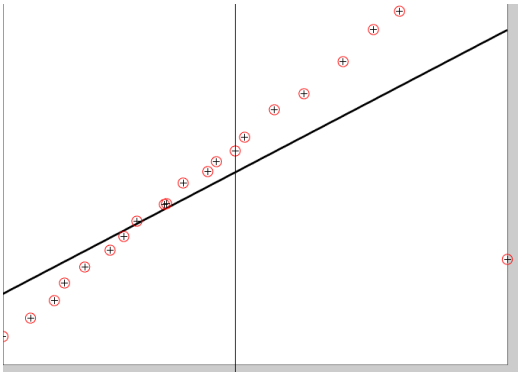
(a) e = .1



(b) e =.25

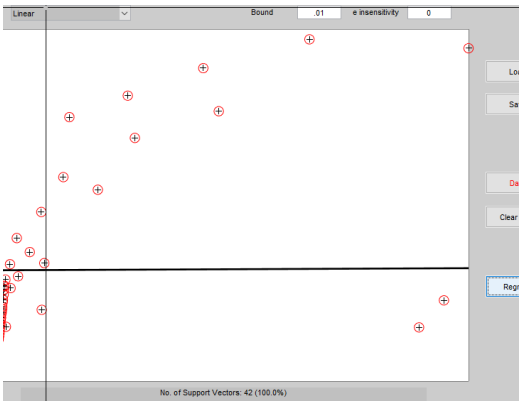


(c) e = .5

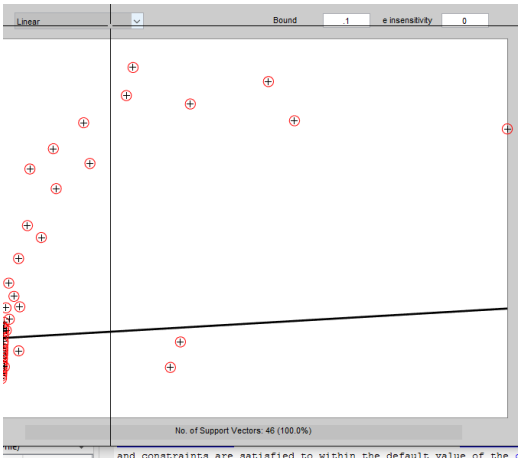


(d) default values

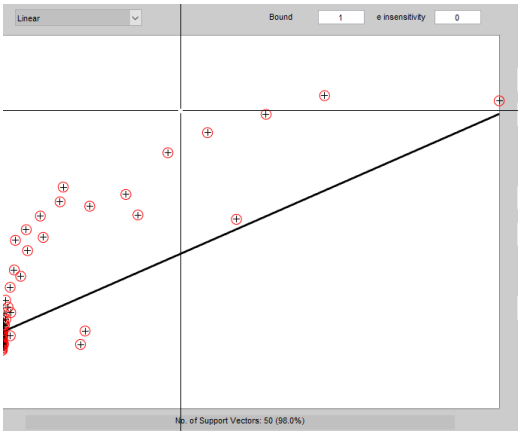
Figure 28: Different values of e



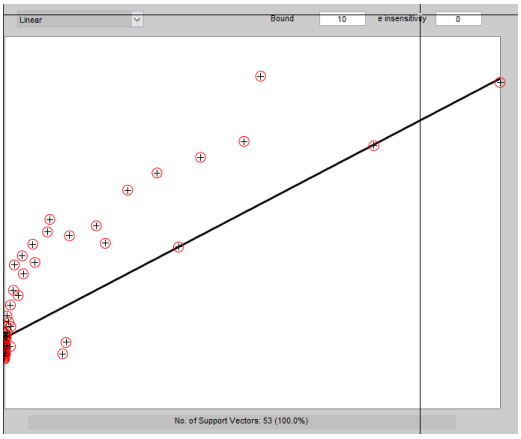
(a) bound = 0.01



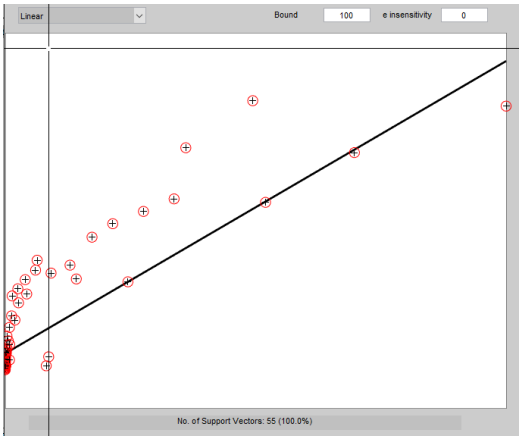
(b) bound = 0.1



(c) bound = 1



(d) bound = 10

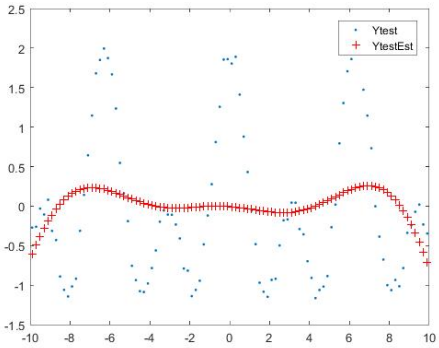


(e) bound = 100

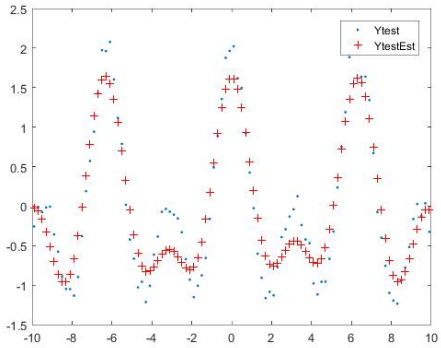
Figure 29: Different values for the bound

2.2 A Simple Example: Sum of Cosines

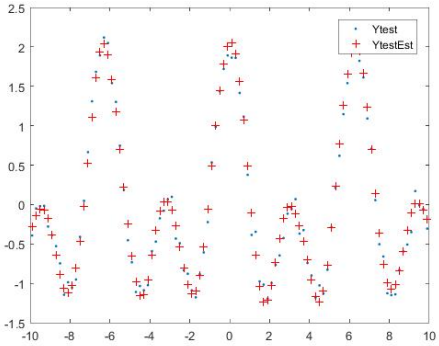
Do you think there is one pair of optimal hyper-parameters? The model was very constrained with the original parameters. Due to its linear nature sig2 was too high. Reducing both parameters gave a better fit.



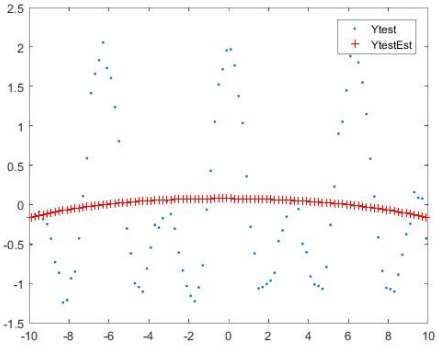
(a) arbitrary sigma and gamma, 100 and 1.0



(b) gamma = 50, sigma = 0.1



(c) gamma = 25, sigma = 0.01



(d) gamma = 200 and sigma = 10

Figure 30: Different values for sigma and gamma

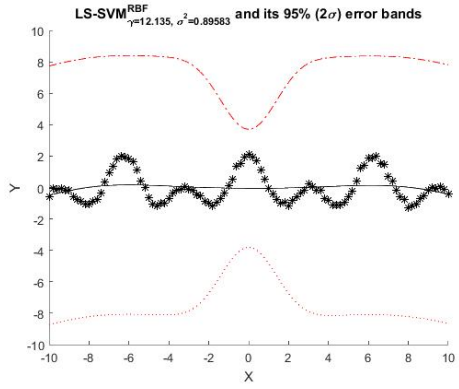
2.3 Hyper-parameter Tuning

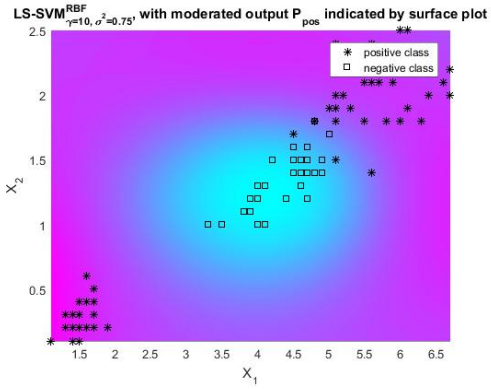
What can you say about the values of the hyperparameters and the results? What is the difference between 'simplex' and 'gridsearch'? Is there any significant deviation in results? Which method is faster and why?

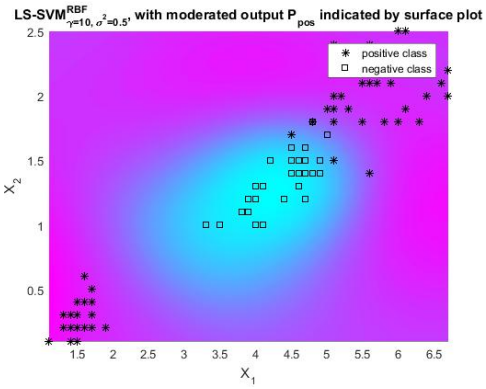
The hyperparameters vary in a pretty tight range between sigma = .05-.11 to gamma = 500-1000. The simplex is an algorithm while the gridsearch is brute force. Although they happened to fall within similar ranges. The fastest is csa and simplex together because these are algorithms that go about parameter tunning methodically. The results of all methods are similar but the DS and gridsearch take a lot of time.

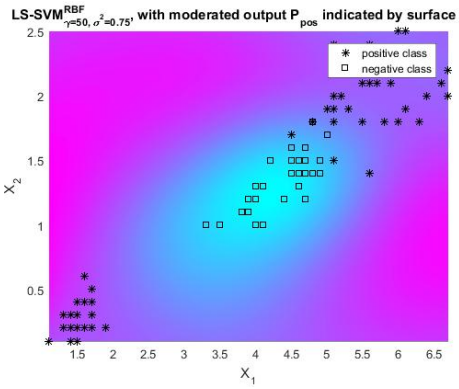
2.4 Application of the Bayesian Framework

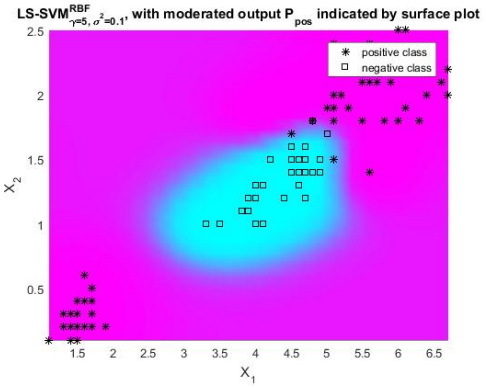
A decreasing of the sig2 value increases the probability of the data to belong to the positive class. Increasing of the gamma parameter allowed for more misclassification. If a point falls into a blue area it is more likely to be a negative class and if it falls in into a purple area it is more likely to be positive. Lowering sigma gives you a more rigid boundary and lowers the chance of misclassification. Raising gamma gives you a fuzzier decision boundary.

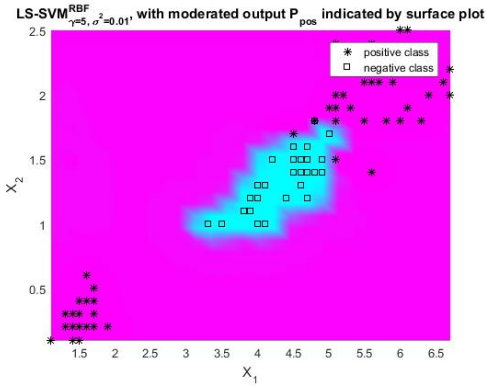


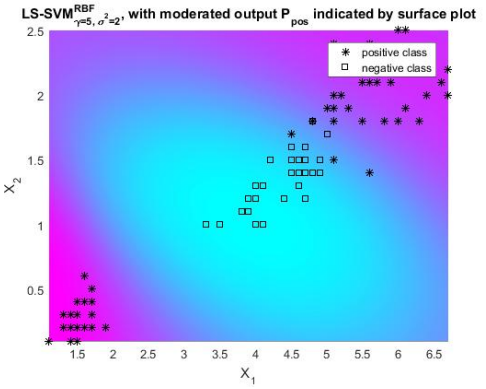










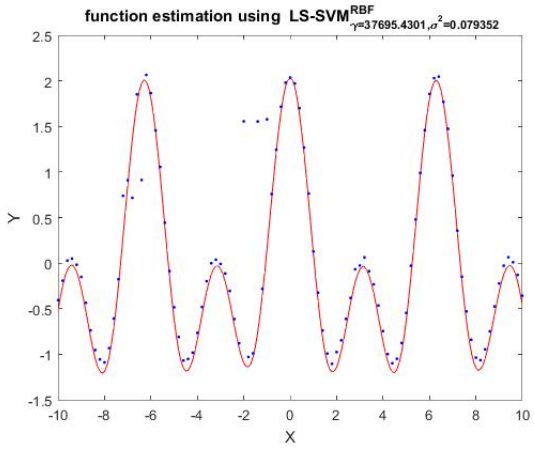


Can you think of a similar way to do input selection by using the crossvalidate function instead of the Bayesian framework?

It possible by creating all possible variable combinations and crossvalidating all models to score the MSE. It would take a long time be prohibitively time consuming.

2.5 Robust Regression

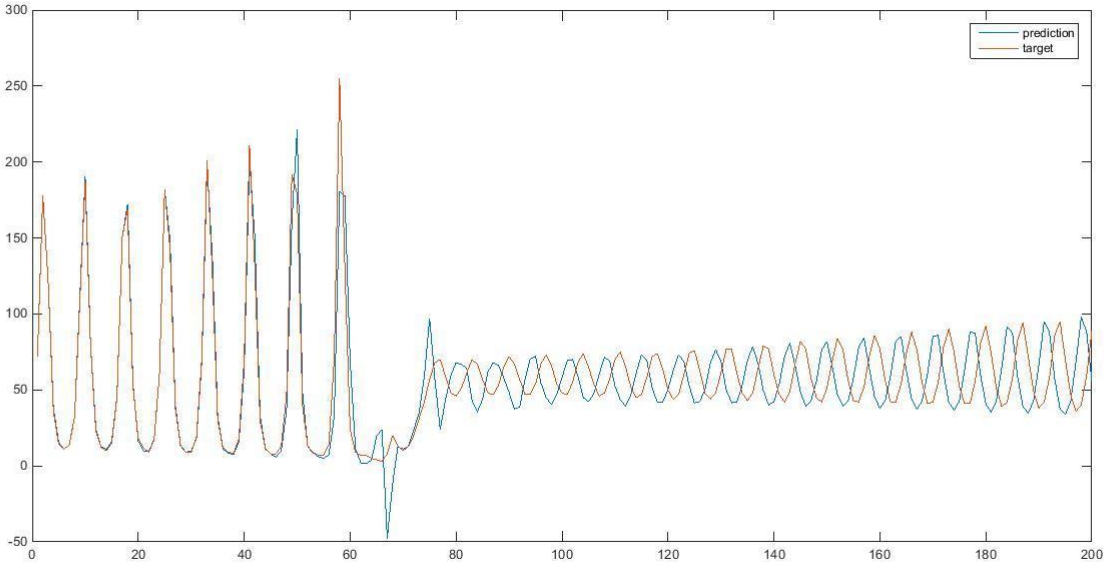
A model of with outlier in the data will cause noise in the model. It will cause the model to be bais by their values. MAE is used because it is less sensitive to outliers.



2.6 Time-series Prediction

SVM can also work on predicted time-series sequences.

2.7 Sante Fe laser Data



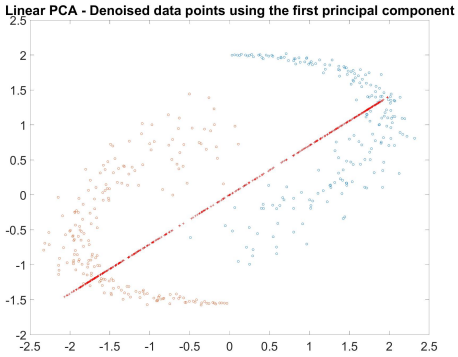
Yes this seems like a good choice. The model lags but can still predict pretty well.

If you used a validation set to optimize you would get autocorrelation.

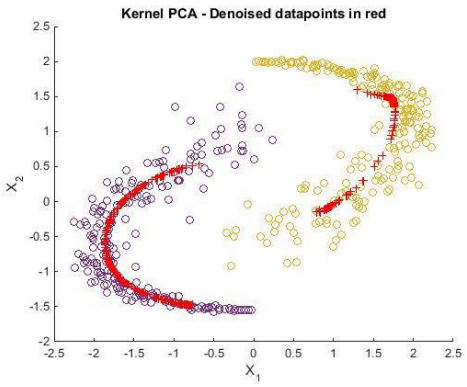
3 Assignment 3

This section will discuss unsupervised learning.

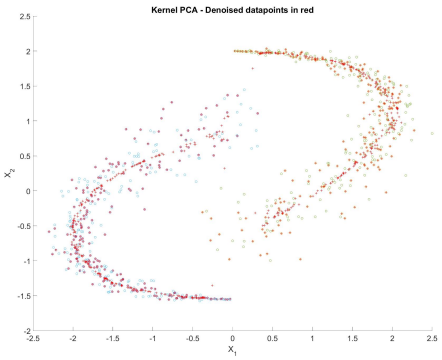
3.1 PCA



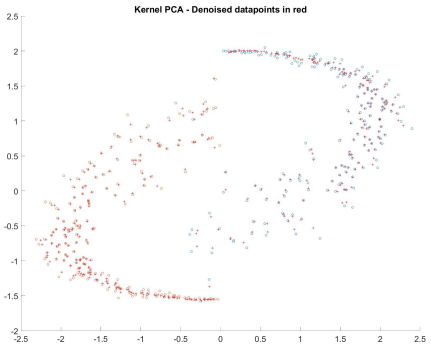
(a) 3 components



(b) 3 components



(c) 9 components



(d) 20 components

Kernel PCA can be used for feature extraction, denoising, dimensionality reduction and density estimation. Can you describe what's happening with the denoising if you increase the number of principal components? What is the difference with linear PCA? How many principal components can you obtain with kernel PCA? and with linear PCA? Can you think of a technique to tune the number of components and the kernel hyper-parameter?

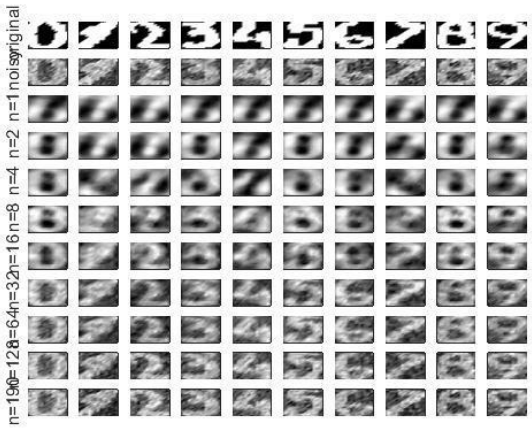
As the number of PC increased the model started to look better. However after a certain amount you start to overfit. Linear PCA does not work for this data at all. Around 3 components we start to get a model that gets really well. The none linear character of that is being captures. The linear PCA is a dimension reduction technique that can have as many components as dimensions. A kernel PCA can have infinite number of dimensions because it takes place in the kernel space.

3.2 Handwritten Digit Denoising

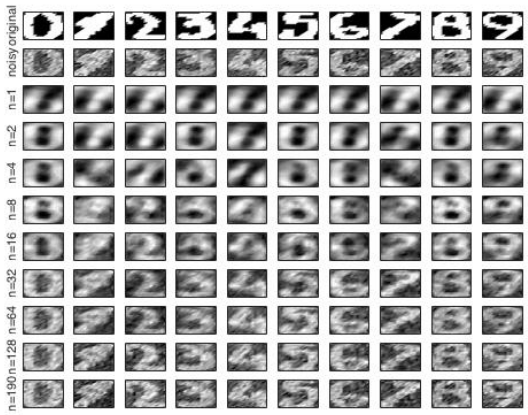
This data-set consists of features of handwritten numerals (0 to 9) extracted from a collection of Dutch utility maps. Try the sample script on Toledo and explain what you

observe.

First line is the original data. The second has noice added to it. The rows that follow show the PCA's trying to remove the noise. As you can see the kernel PCA does much better than the liner one.



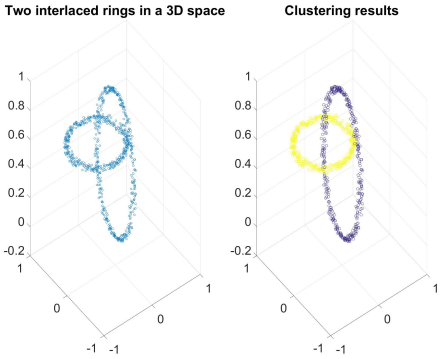
(a) Kernel PCA



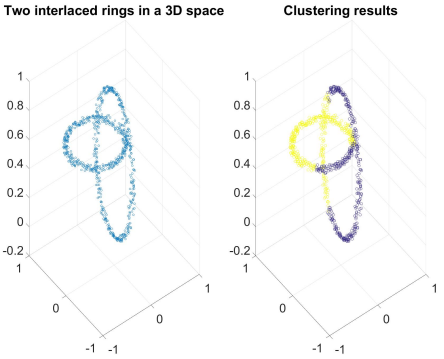
(b) Linear PCA

3.3 Spectral Clustering

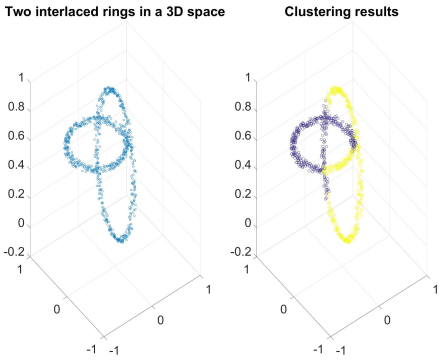
What is the difference with classification? Classification is a supervised method. This means that it is trained on data where the "answer" is known. Data that already has a set of predicitions. Clustering estimates which class a particular points belongs into.



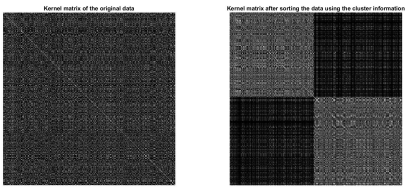
(a) sig = 0.001



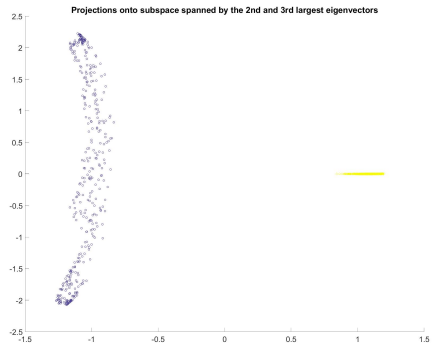
(b) sig = 0.01



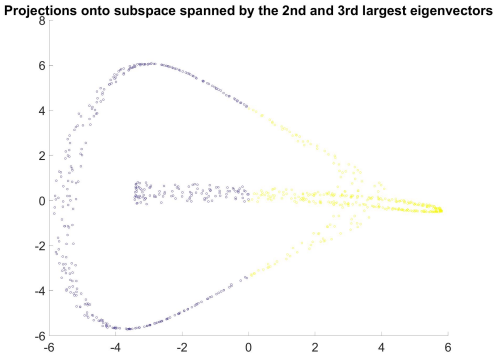
(c) sig = 0.005



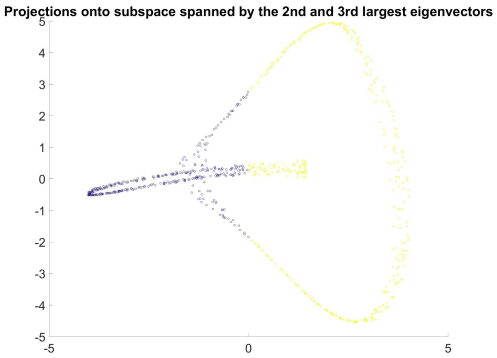
The low sigma classifies perfectly in one ring. The higher sigma goes the more misclassification is allowed. Eventually though when it goes high enough the other ring gets classified perfectly.



(a) sig = 0.001



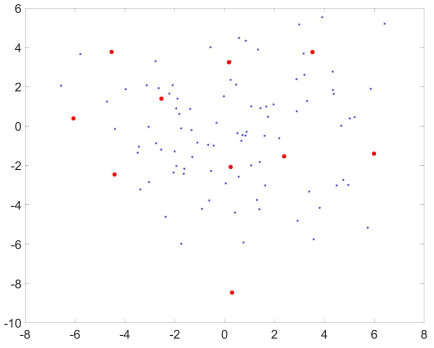
(b) sig = 0.01



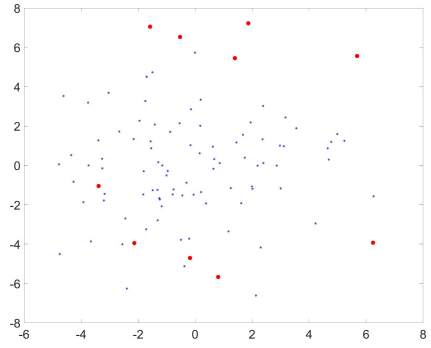
(c) sig = 0.005

3.4 Fixed Size LS-SVM

What is the influence of the chosen sig2?



(a) sig = 0.001



(b) sig = 0.01