Support Vector Machines: Methods and Applications

Sumit Kumar Dey (R0607761)
Master of Artificial Intelligence

August, 2016

Contents

Support Vector Machines: Methods and Applications
Exercise Session 1: Classification

1.1 A Simple Example: Two Gaussians

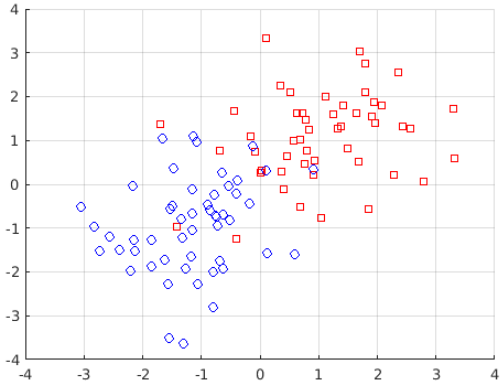


Figure 1.1: Data - toy example

1. *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? In general it's a good idea to show insight in the following material by using such a visualization.*

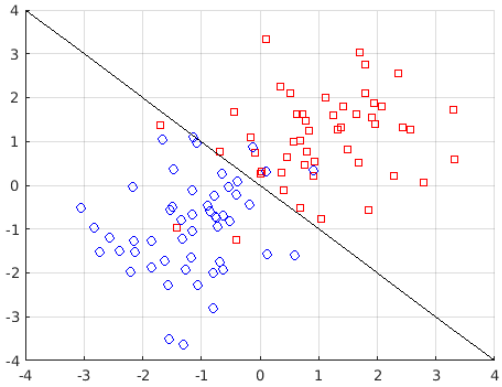


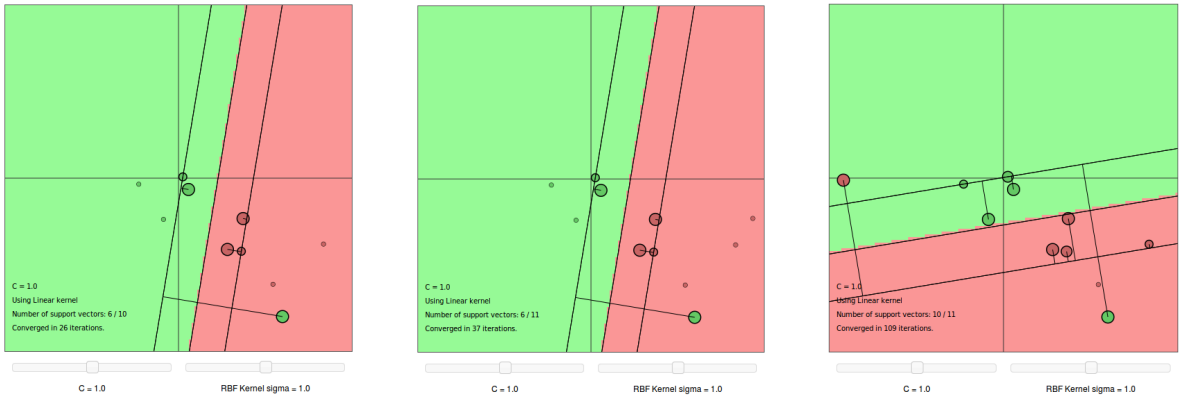
Figure 1.2: Classification decision boundary

1.2 The Support Vector Machine

1. *Switch to using the linear kernel by toggling the "k" button. 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?*

We observe that the hyperplanes and margin change so as to take care of the new points included in the dataset (so as to find a proper decision boundary and hyperplanes).

The extent of change that occurs in decision boundary depends on where the new data points are placed. If the data points are placed on the wrong side of the hyperplane then we can anticipate a drastic change in the decision boundary while the change may not be equally drastic if the data points are placed on the correct side of the hyperplanes and far from the hyperplane. Figure1.3b and figure1.3c show two different examples of effect of adding a new data point to figure1.3a.



(a) Before adding new data point

(b) New data point added on correct side of decision boundary

(c) New data point added to wrong side of decision boundary

Figure 1.3: Effect on classification boundary on adding a new data point

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

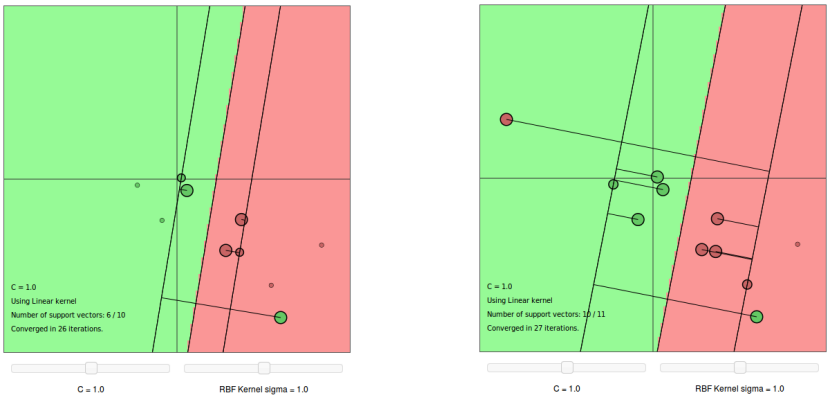
The margin or distance between the hyperplanes increase as value of $\sum_{i=1}^{n} \xi_i$ increases because of this wrongly placed point. Due to increase in margin, more data points may violate the hyperplane which are included as support vectors. Figure1.4a and figure1.4b demonstrate this effect.

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

C is the regularization parameter and the value of C determines how much miss-classification are we willing to tolerate. Allowing some violation also means that we may have a larger margin.

- C is small (soft-margin): We get a larger margin at the cost of higher values of slack variable. As margin large, we have more number of support vectors. Small C leads to high bias and low variance of the hypothesis.

- C is large (hard-margin): We try to avoid miss-classification and thus the margin may get smaller



(a) Before adding new datapoint

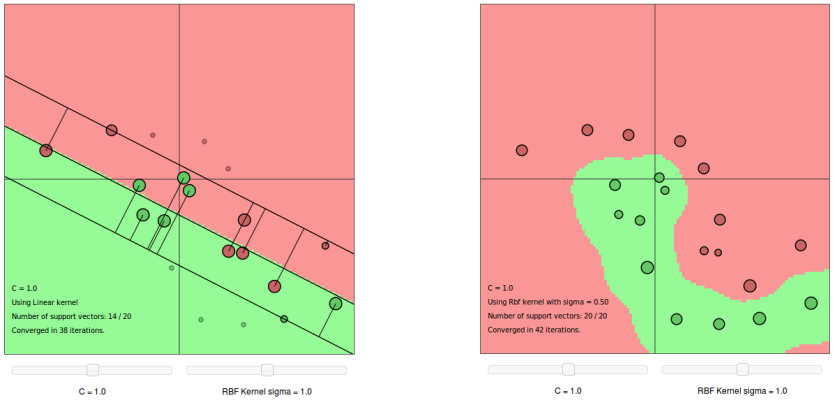
(b) New datapoint added on wrong side of decision boundary

Figure 1.4: Effect on hyperplanes by adding a new datapoint on wrong side

as C increases (to a certain value after which margin cannot be any smaller). As the margin reduces, the number of support vectors also tend of reduce. Increase in the value of C leads to low bias and high variance of the hypothesis.

4. *Follow the instructions and switch back to RBF kernel by toggling the "k" button. Compare to the classification outcome of the linear case.*

The datasets where the datapoints are, the linear classification cannot properly classify such cases. But using Non-linear classification of RBF kernel we are able to classify such data sets. But we also observe that the number of support vectors tend to increase when we use RBF kernel as compared to linear SVM, possibly to 'support' the more complicated non-linear boundary.



(a) Linear decision boundary

(b) Non-linear decision boundary (RBF kernel)

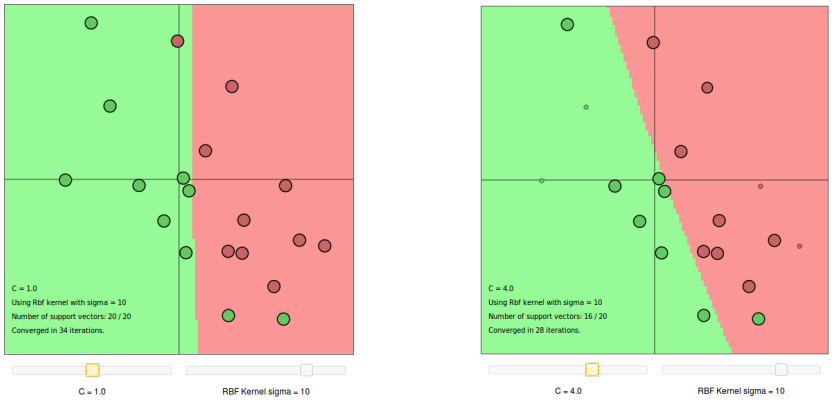
Figure 1.5: Comparison of linear kernal and RBF kernel on same dataset

5. *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 sigma parameter defines how far the influence of a single training example reaches. If sigma is too small, the radius of influence of the support vectors is limited to itself. Thus, it leads to overfitting. When sigma is very large, the radius of influence of support vectors is too large and may lead to underfitting. At very large values, region of influence of any selected support vector would include the whole training set.

The C parameter trades off missclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly. Low value of C leads to high bias of the hypothesis and high value of C leads to high variance of the hypothesis.

For linearly separable choosing sigma on higher side and C bit more than 1 gave good results.



(a) C=1.0, sigma=10 (b) C=4.0, sigma=10

Figure 1.6: RBF kernel on linearly separable data

6. *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 non-linearly separable data, RBF kernel outperforms the linear kernel. The regularization parameter - C determines amount of missclasification that we wish to tolerate. For RBF kernel the value of C determines the complexity of the decision boundary where as for linear kernel it determines the width of margin.

The sigma parameter defines how far the influence of a single training example reaches. If sigma is too small, the radius of influence of the support vectors is limited to itself. Thus, it leads to overfitting. When sigma is very large, the radius of influence of support vectors is too large and may lead to underfitting. At very large values, region of influence of any selected support vector would include the whole training set.

7. *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?*

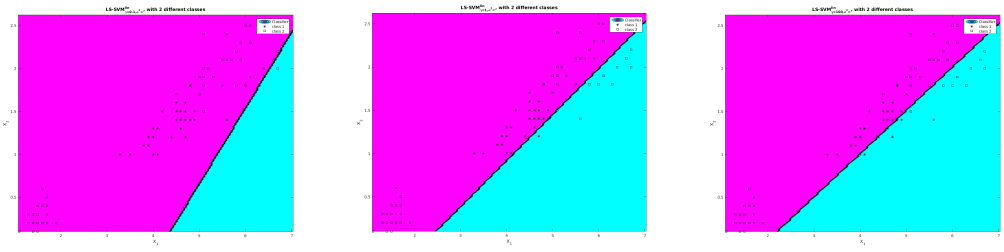
Support vectors are a subset of the training dataset which contribute to the location of decision boundary. They are usually located closest to decision boundary. The data-points which are most difficult to classify are also considered as support vectors. A data point which is miss-classified or is near to the decision boundary is usually taken as support vector.

8. *When does the corresponding importance of a Support Vector change?*

Importance of a support vector is according to the associated lagarangian constant α and slack variable ξ. Positive value of alpha makes a datapoint a support vector. ξ defines the slack and it takes care of the data points that violates the margin. Mostly, importance of a support vector depends on its alpha value. Higher is the value of α, more important the support vector is and higher is its impact on the formation of hyperplane.

1.3 Using LS-SVMlab

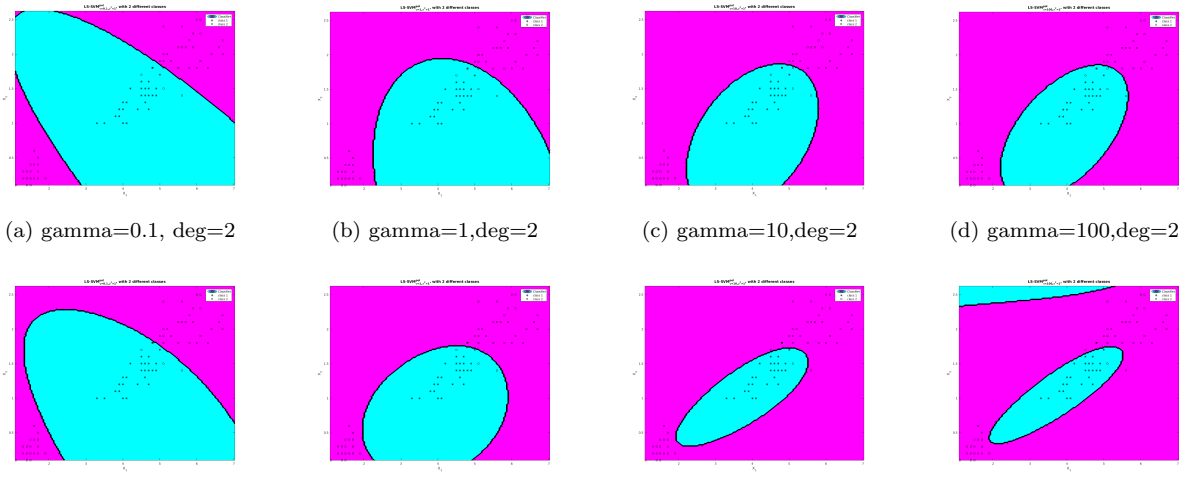
On Iris Dataset



(a) gamma=0.1 (b) gamma=1 (c) gamma=100

Figure 1.7: Linear kernel on Iris dataset

$gam = 0.100 => ontest : \#misclass = 10/20, error = 50.00\% MAE : 1.000, MSE : 2.000$
$gam = 1.000 => ontest : \#misclass = 11/20, error = 55.00\% MAE : 1.100, MSE : 2.200$
$gam = 100.000 => ontest : \#misclass = 11/20, error = 55.00\% MAE : 1.100, MSE : 2.200$



(a) gamma=0.1, deg=2 (b) gamma=1,deg=2 (c) gamma=10,deg=2 (d) gamma=100,deg=2

(e) gamma=0.1, deg=3 (f) gamma=1,deg=3 (g) gamma=10,deg=3 (h) gamma=100,deg=3

Figure 1.8: Polynomial kernel on Iris dataset

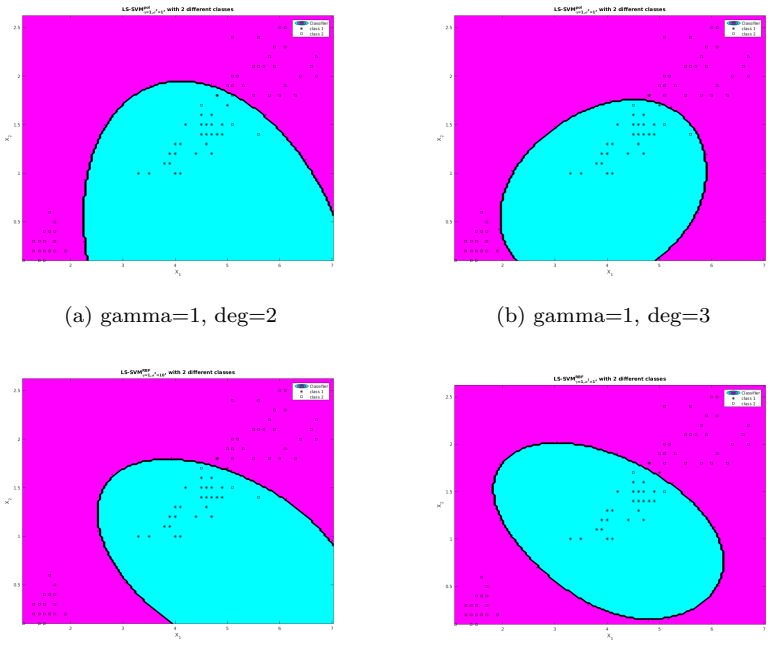
Polynomial Deg=2

$gam = 0.100 => ontest : \#misclass = 1/20, error = 5.00\% MAE : 0.100, MSE : 0.200$
$gam = 1.000 => ontest : \#misclass = 1/20, error = 5.00\%, MAE : 0.100, MSE : 0.200$
$gam = 10.000 => ontest : \#misclass = 1/20, error = 5.00\% MAE : 0.100, MSE : 0.200$
$gam = 100.000 => ontest : \#misclass = 1/20, error = 5.00\%, MAE : 0.100, MSE : 0.200$

Polynomial Deg=3
$gam = 0.100 => ontest : \#misclass = 0/20, error = 0.00\%, MAE : 0.000, MSE : 0.000$
$gam = 1.000 => ontest : \#misclass = 0/20, error = 0.00\%, MAE : 0.000, MSE : 0.000$
$gam = 10.000 => ontest : \#misclass = 0/20, error = 0.00\%, MAE : 0.000, MSE : 0.000$
$gam = 100.000 => ontest : \#misclass = 0/20, error = 0.00\%, MAE : 0.000, MSE : 0.000$

1. *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?*

For degree=1, the decision boundary is linear. As we increase the degree of the polynomial, the non-linearity of the decision boundary increases and is able to fit the training data in better way. The change in degree of polynomial corresponds to the changes in sigma hyperparameter of RBF kernel. As the value of sigma decreases, the decision boundary becomes more non-linear and is able to fit the training data in better way. But as the value of sigma increases, the decision boundary tends to be linear.



(a) gamma=1, deg=2

(b) gamma=1, deg=3

(c) gamma=1,sigma=10

(d) gamma=1,sigma=1

Figure 1.9: Polynomial kernel on Iris dataset

2. *Let us focus on the RBF kernel with kernel parameter the bandwidth* σ^2 *. Try out a good range of different sig2's as kernel parameters. For each individual value of sig2, the corresponding LS-SVM is evaluated on the test set. Make a figure of the sig2's with their corresponding test set performance.*

3. *Now, take a look at the regularization constant gam. Fix a reasonable choice for the sig2 of the RBF kernel and again compare a range of gam's by plotting the corresponding test set performances. What is a good range for gam?*

Response to both questions 2 & 3 Gamma=1 and Sigma2=1, appears to be a good choice.

	sig2=0.01	sig2=0.1	sig2=1	sig2=10	sig2=100	sig2=1000
gam=0.01	50.0%	50.0%	50.0%	50.0%	50.0%	50.0%
gam=0.10	35.0%	10.0%	0.0%	50.0%	50.0%	50.0%
gam=1.00	10.0%	0.0%	0.0%	0.0%	50.0%	50.0%
gam=10.00	10.0%	0.0%	0.0%	5.0%	50.0%	50.0%
gam=100.00	10.0%	5.0%	0.0%	0.0%	5.0%	50.0%
gam=1000.00	10.0%	5.0%	0.0%	0.0%	10.0%	55.0%

Table 1.1: Percentage error with variation in gamma and sigma of RBF kernel on Iris dataset test set

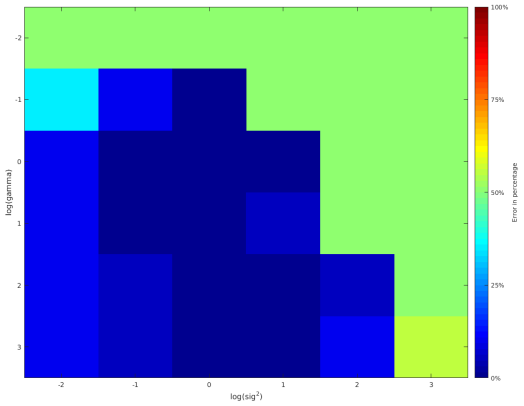


Figure 1.10: Heat map of percentage error with variations in gamma and sigma on Iris test set

1.3 Choice of Hyperparameters

	sig2=0.1	sig2=1	sig2=10	sig2=100
gam=0.10	10.0%	0.0%	50.0%	50.0%
gam=1.00	0.0%	0.0%	10.0%	50.0%
gam=10.00	0.0%	0.0%	0.0%	50.0%
gam=100.00	0.0%	0.0%	0.0%	5.0%
gam=1000.00	0.0%	0.0%	0.0%	0.0%

Table 1.2: Percentage error on validation set with variation in gamma and sigma of RBF kernel on Iris dataset

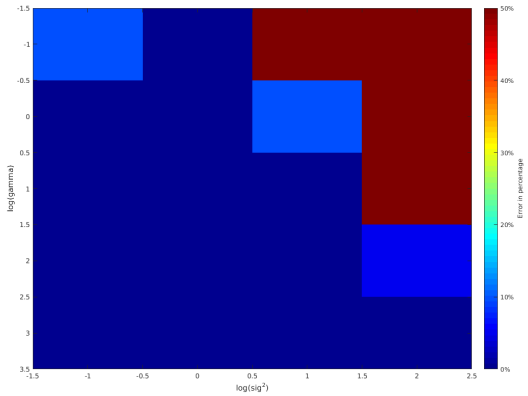


Figure 1.11: Heat map of percentage error with variations in gamma and sigma on Iris test set

	sig2=0.1	sig2=1	sig2=10	sig2=100
gam=0.10	0.040	0.050	0.330	0.330
gam=1.00	0.040	0.050	0.050	0.330
gam=10.00	0.040	0.050	0.060	0.340
gam=100.00	0.040	0.040	0.040	0.050
gam=1000.00	0.050	0.040	0.050	0.060

Table 1.3: Cross-validation error with variations in gamma and sigma of RBF kernel on Iris dataset

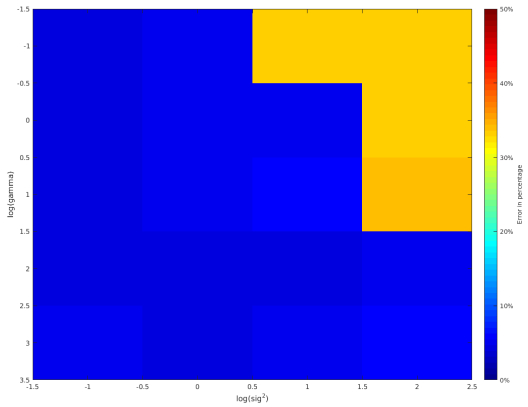


Figure 1.12: Heat map of cross-validation error with variations in gamma and sigma on Iris test set

	sig2=0.1	sig2=1	sig2=10	sig2=100
gam=0.10	0.040	0.050	0.330	0.330
gam=1.00	0.040	0.050	0.050	0.330
gam=10.00	0.040	0.050	0.060	0.340
gam=100.00	0.040	0.040	0.040	0.050
gam=1000.00	0.050	0.040	0.050	0.050

Table 1.4: Leave-one-out error with variations in gamma and sigma of RBF kernel on Iris dataset

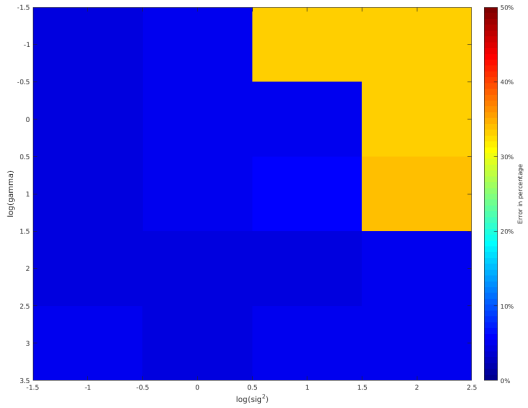


Figure 1.13: Heat map of Leave-one-out error with variations in gamma and sigma on Iris test set

	γ	σ^2
ds, gridsearch	0.083149	0.9653
csa, gridsearch	369.0242	0.02865769
ds, simplex	7.1996	0.47482
csa, simplex	1.7016	2.2753

Table 1.5: Gamma and sigma values obtained by various tuning approaches

1.4 Homework Problems

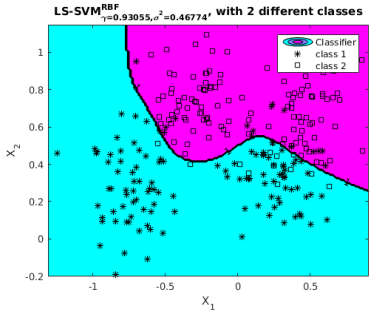
1.4 Ripley Dataset

1.4 Breast Cancer Dataset

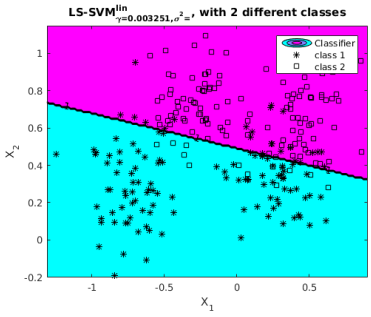
Percentage Error on validation set(Linear Kernel): 3.750%
Percentage Error on test set(Linear Kernel): 4.734%
Percentage Error on validation set(RBF Kernel): 2.500%
Percentage Error on test set(RBF Kernel): 2.959%

1.4 Diabetes Dataset

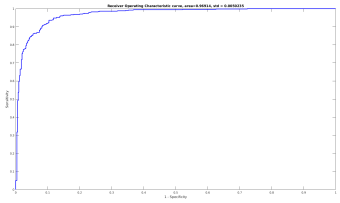
Percentage Error on validation set(Linear Kernel): 23.333%
Percentage Error on test set(Linear Kernel): 36.905%
Percentage Error on validation set(RBF Kernel): 23.333%
Percentage Error on test set(RBF Kernel): 25.000%



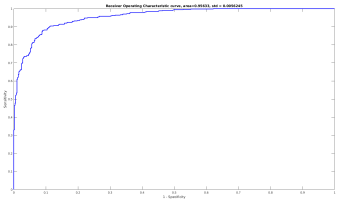
(a) RBF Kernel



(b) Linear Kernel

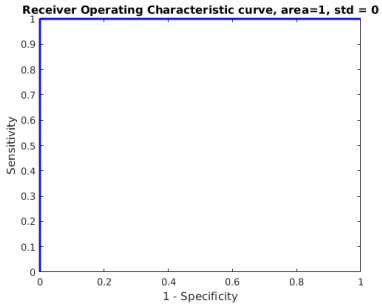


(c) RBF Kernel ROC

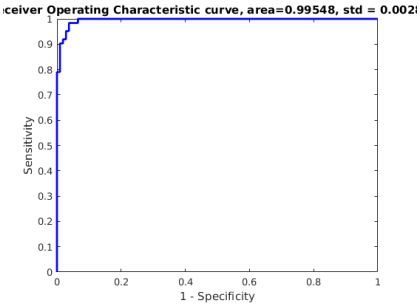


(d) Linear Kernel ROC

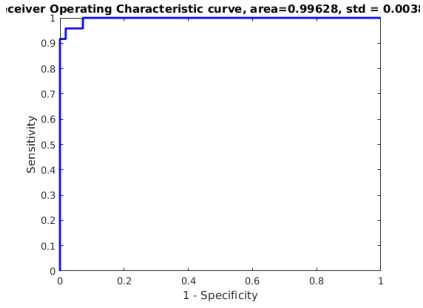
Figure 1.14: Ripley Dataset



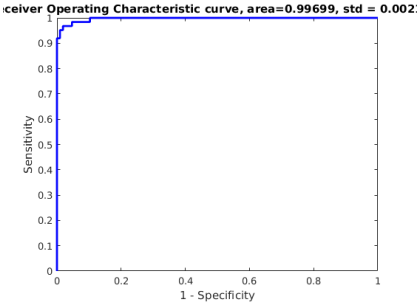
(a) Linear Kernel - Validation Set



(b) Linear Kernel - Test set

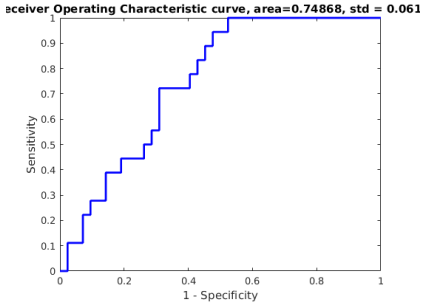


(c) RBF Kernel - Validation Set

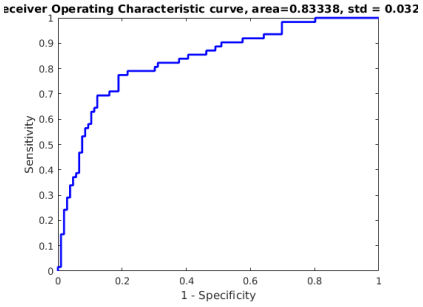


(d) RBF Kernel - Test Set

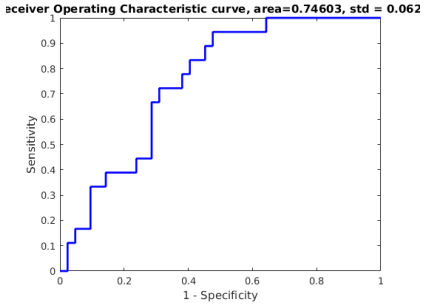
Figure 1.15: Breast Cancer Dataset



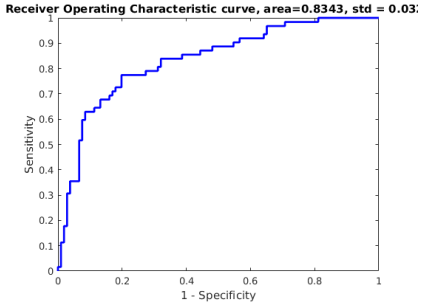
(a) Linear Kernel - Validation Set



(b) Linear Kernel - Test set



(c) RBF Kernel - Validation Set



(d) RBF Kernel - Test Set

Figure 1.16: Diabetes Dataset

Support Vector Machines: Methods and Applications
Exercise Session 2:
Function Estimation and Time-series Prediction

2.1 The Support Vector Machine for Regression

1. *Construct a data-set of around 20 data-points. Which kernel is best suited for your data-set? Construct a data-set where a linear kernel is better than any other kernel. What is the influence of different values of e? (try small values e.g., 0.1, 0.25, 0.5...) and of Bound (i.e. hyperparameter C in the previous exercise session)? Try values with larger increments e.g. 0.01, 0.1, 1, 10, 100. Where does the sparsity property come in? In what respect is SVM regression different from a classical Least Squares fit?*

In the constructed dataset of 20 points, the Gaussian RBF kernel (σ: 5) appears to be a good approximation.
Difference between Least Squares and SVM regression

- In least squares fit, we try to minimize the projection of the datapoints on the function or hyperplane. Where as in SVM regression, we consider the normal distance from the hyperplane to the datapoint. SVM is more geometrically motivated as compared to LS.

- In SVM regression, only the support vectors are considered and rest of the datapoints have no role to play. In the least sqaures fit all data points are considered for function approximation.

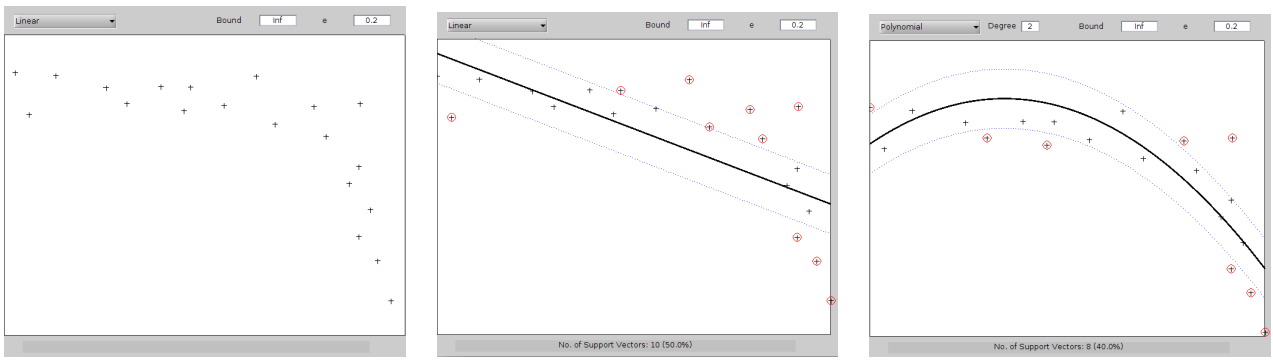
2.2 A Simple Example: Sum of Cosines

Q. *Make plots showing the results on training and test sets with different values of gam and sig2. Do you think there is one pair of optimal hyper-parameters?*

We plot the results for different values of gam [1, 10, 100, 1000] and sig2 [0.01, 0.1, 1]. We observe that there are multiple sets of hyperparameters that produce comparably good results. We observed good results with higher values of gam and lower values of sig2.

2.3 Hyper-parameter Tuning

Following tables represent the tuned hyperparameters based on different approaches over 20 iterations in each case. It is evident that there can be various sets of hyperparameters resulting in similar performance. Randomised directional search produces greater variability in the tuned hyper parameters



(a) Dataset - 20 points

(b) Linear Kernel

(c) Polynomial, Degree: 2

(d) Polynomial, Degree: 3

(e) Polynomial, Degree: 4

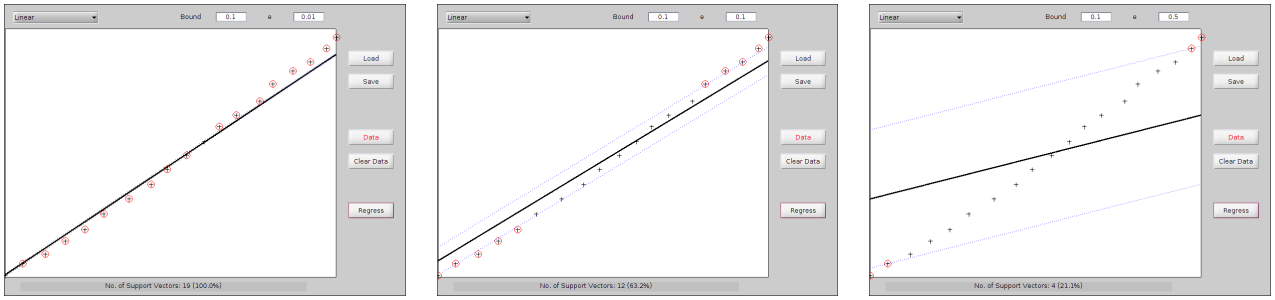
(f) Gaussian RBF, σ: 1

(g) Gaussian RBF, σ: 5

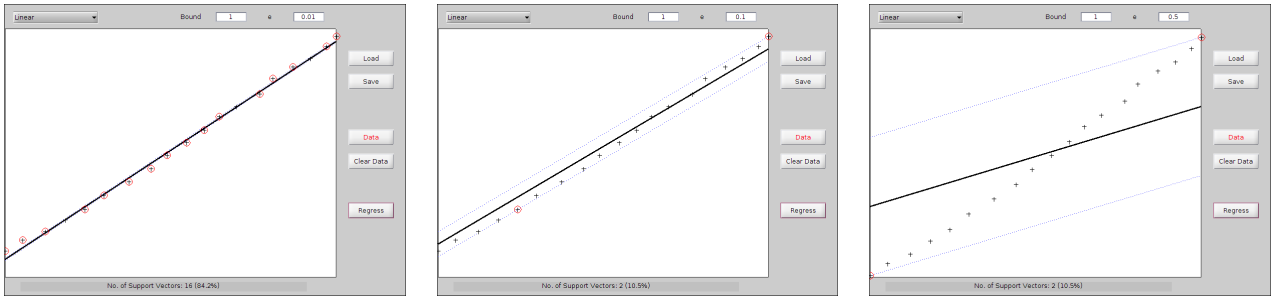
(h) Gaussian RBF, σ: 10

(i) Gaussian RBF, σ: 50

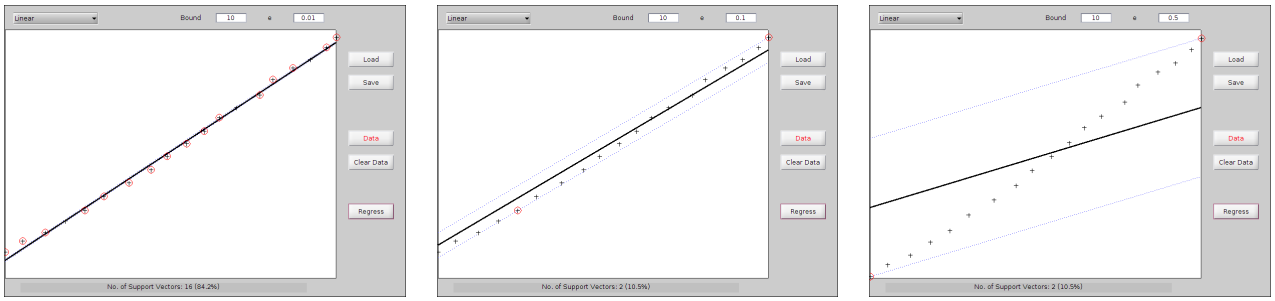
Figure 2.1: Different kernels on a dataset of 20 points



(a) Bound(C): 0.1, e: 0.01 (b) Bound(C): 0.1, e: 0.1 (c) Bound(C): 0.1, e: 0.5



(d) Bound(C): 1, e: 0.01 (e) Bound(C): 1, e: 0.1 (f) Bound(C): 1, e: 0.5



(g) Bound(C): 10, e: 0.01 (h) Bound(C): 10, e: 0.1 (i) Bound(C): 10, e: 0.5

Figure 2.2: Different kernels on a dataset of 20 points

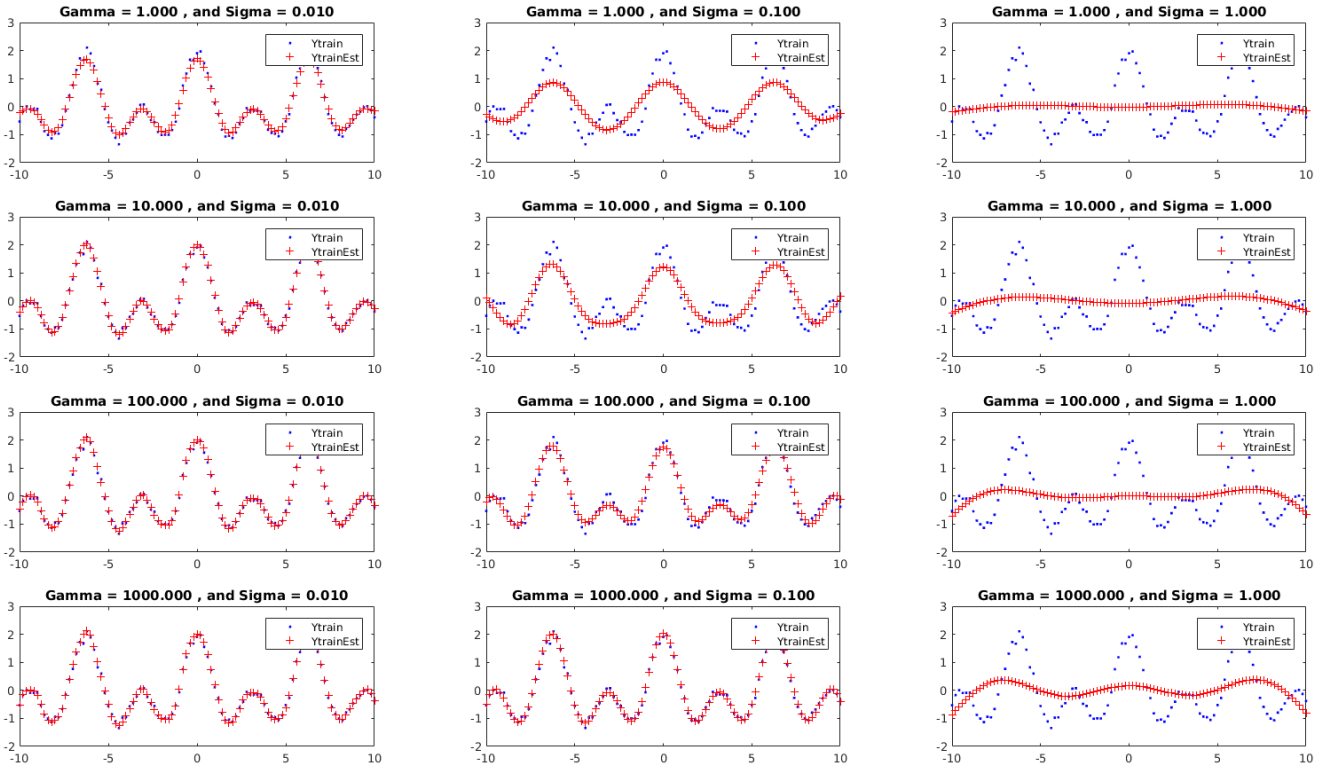


Figure 2.3: Sum of Cosines: Different values of gamma and sig2 on test set

as compared to CSA (Coupled Simulated Annealing). Execution time for simplex is less than that of grid search.

	γ	σ^2	MSE_{train}	MSE_{CV}	MSE_{LOO}	Time
mean	38.0009e+003	103.0765e-003	11.3364e-003	11.3027e-003	11.0410e-003	544.3835e-003
minimum	2.1437e+003	79.7470e-003	10.4007e-003	10.4451e-003	10.9726e-003	529.0210e-003
maximum	106.1297e+003	123.3822e-003	12.2867e-003	13.6332e-003	11.2320e-003	584.1050e-003

Table 2.1: CSA and Grid Search

	γ	σ^2	MSE_{train}	MSE_{CV}	MSE_{LOO}	Time
mean	47.7354e+003	364.7881e+000	109.4618e-003	113.2988e-003	109.7737e-003	538.9370e-003
minimum	205.7515e-009	66.1252e-003	10.3216e-003	10.7720e-003	10.9738e-003	506.8450e-003
maximum	284.2463e+003	7.1331e+003	993.6103e-003	1.0268e+000	998.0057e-003	589.1060e-003

Table 2.2: DS and Grid Search

	γ	σ^2	MSE_{train}	MSE_{CV}	MSE_{LOO}	Time
mean	36.7957e+003	93.4261e-003	11.2570e-003	12.0464e-003	11.1421e-003	354.1961e-003
minimum	223.7982e+000	34.3692e-003	10.3300e-003	10.6226e-003	10.9821e-003	334.8530e-003
maximum	331.9740e+003	129.3528e-003	13.5759e-003	23.8497e-003	12.6661e-003	392.1240e-003

Table 2.3: CSA and Simplex

	γ	σ^2	MSE_{train}	MSE_{CV}	MSE_{LOO}	Time
mean	7.4381e+003	2.5477e+006	308.1157e-003	307.2455e-003	307.3300e-003	320.5366e-003
minimum	667.7450e-009	24.8595e-003	10.6380e-003	10.8032e-003	10.9822e-003	236.6830e-003
maximum	39.7409e+003	22.2526e+006	1.0081e+000	1.0037e+000	998.0057e-003	373.2830e-003

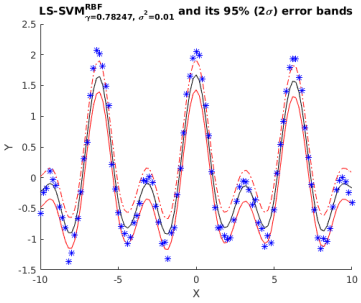
Table 2.4: DS and Simplex

2.4 Application of the Bayesian Framework

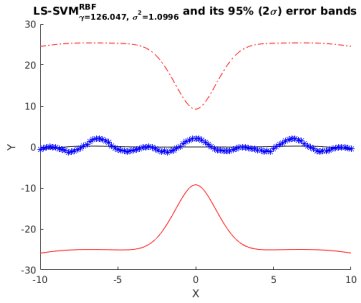
In bayesian framework, error bars are computed in the case of regression and class probabilities are computed in the case of classification. The bayesian framework allows us to tune the parameters for the sum of cosines dataset without use of cross validation or any validation sets. The sum of cosines dataset is tuned for different starting values of gam [0.1, 1, 10] and sig2 [0.01, 0.1, 1]

Here, we perform classification on Iris dataset. The bayesian framework is used to compute the class probabilities and is represented by the colours. We train using different values of gam [0.5, 5, 50] and sig2 [0.075, 0.75, 7.5]. We observe that the classification is strict for lower values of gam and sig2 and is liberal as we increase the values of gam and sig2.

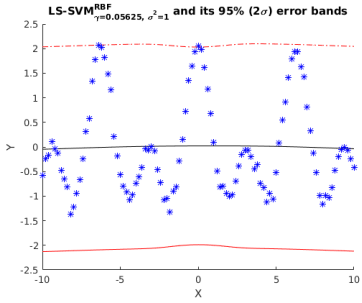
Feature selection can be done using cross-validation on each feature independently and combination of features, then choosing features based on minimisation of error metric like MSE or MAE.



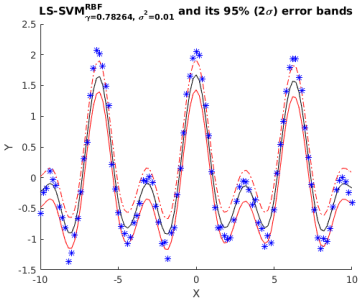
(a) γ:0.1, σ^2:0.01



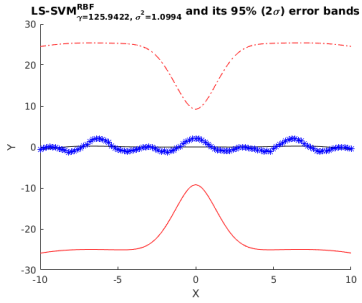
(b) γ:0.1, σ^2:0.1



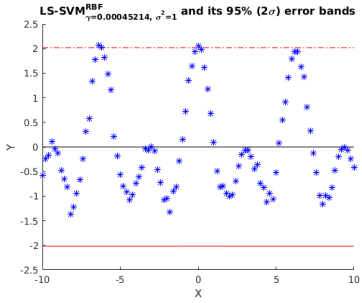
(c) γ:0.1, σ^2:1



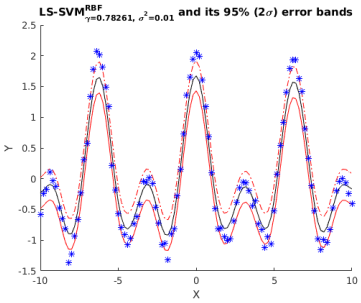
(d) γ:1, σ^2:0.01



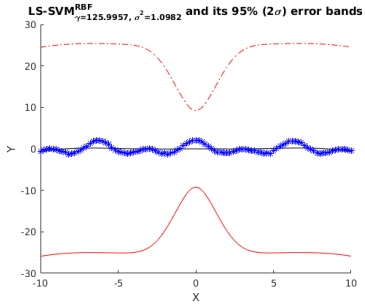
(e) γ:1, σ^2:0.1



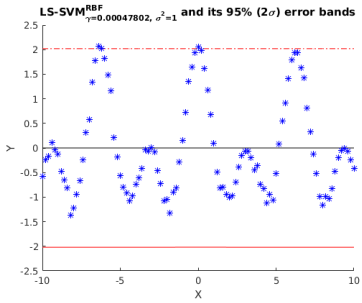
(f) γ:1, σ^2:1



(g) γ:10, σ^2:0.01

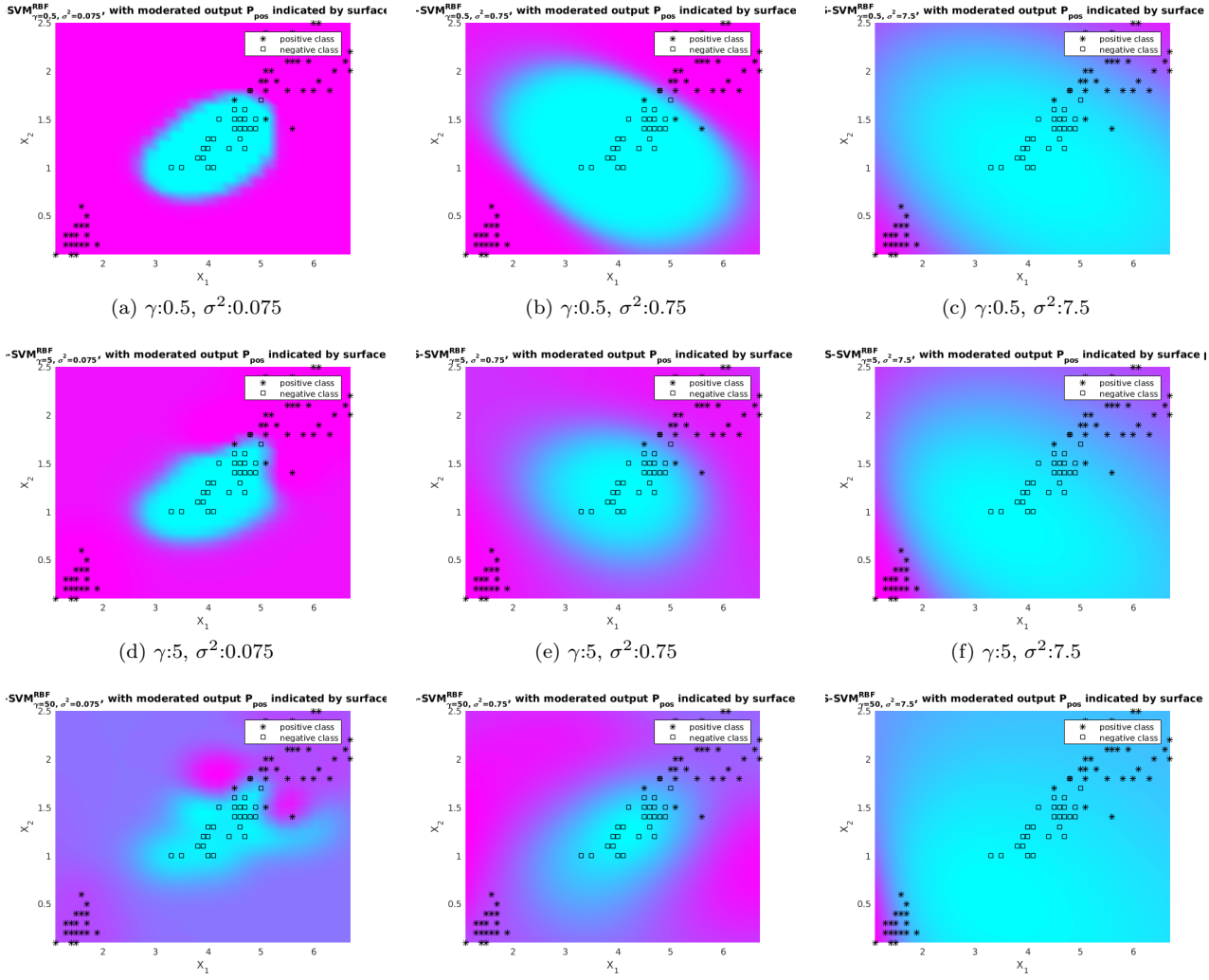


(h) γ:10, σ^2:0.1



(i) γ:10, σ^2:1

Figure 2.4: Bayesian Hyperparameter Tuning for Function Estimation with RBF for different initial values values of γ and σ^2



(a) γ:0.5, σ^2:0.075 (b) γ:0.5, σ^2:0.75 (c) γ:0.5, σ^2:7.5

(d) γ:5, σ^2:0.075 (e) γ:5, σ^2:0.75 (f) γ:5, σ^2:7.5

(g) γ:50, σ^2:0.075 (h) γ:50, σ^2:0.75 (i) γ:50, σ^2:7.5

Figure 2.5: Probability of positive and negative class computed using the Bayesian Framework using RBF-kernel on Iris dataset for different values of γ and σ^2

2.5 Robust Regression

Q. *Discuss the results. Why in this case is the mean absolute error (mae) preferred over the classical mean squared error (mse)? Try alternatives to the weighting function wFun (e.g. 'whampel', 'wlogistic','wmyriad').*

In the presence of outliers the MAE should be preferred as it is comparatively more resilient to the presence of outliers. MSE uses squared errors which makes it sensitive to outliers.

We observe visibly better results with robust cross-validation and the results of different weighting functions seem comparable.

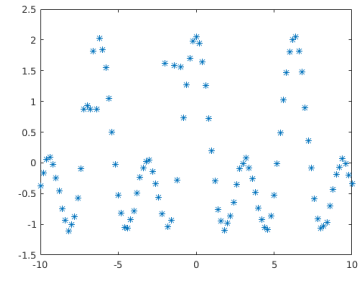
2.6 Homework Problem

Q. *Does order=50 for the utilized auto-regressive model sound like a good choice?*

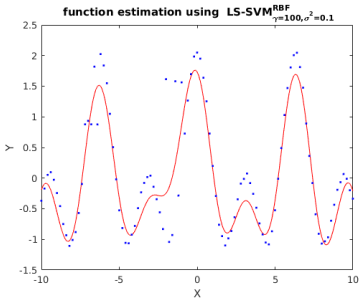
The orders lower than 50 seem to perform significantly worse than that the order 50. In the higher orders as the number of datapoints on which mse is calculated is lower, it doesn't give us a true metric of quality. Under this case, the order 50 and 60 seems to make a good trade off and hence can be considered as good choice.

Q. *Would it be sensible to use the performance of this recurrent prediction on the validation set to optimize the hyper-parameters and the model order?*

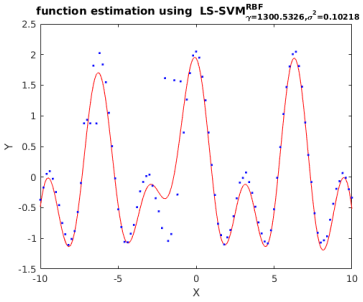
I assume that splitting a later part of the training set as validation set and use it for tuning hyper-parameters and model order would be a sensible choice.



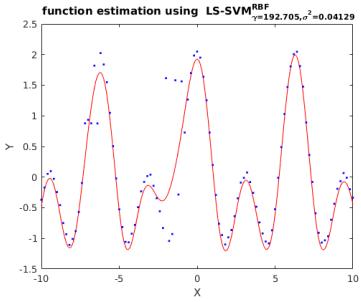
(a) Datapoints



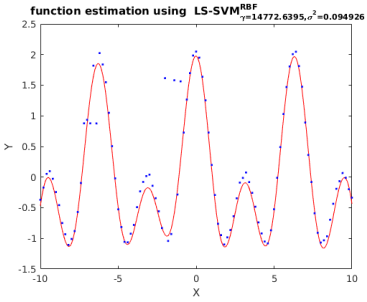
(b) γ:100, σ^2:0.1



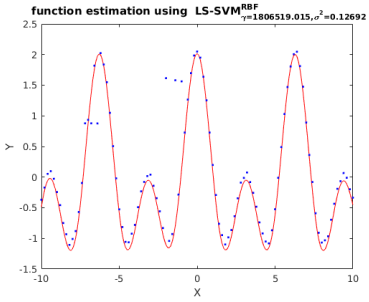
(c) MSE, CV



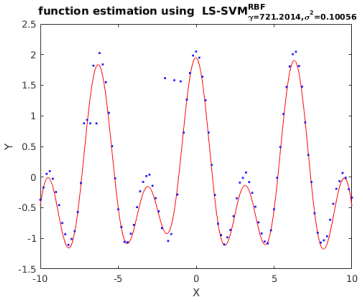
(d) MAE, CV



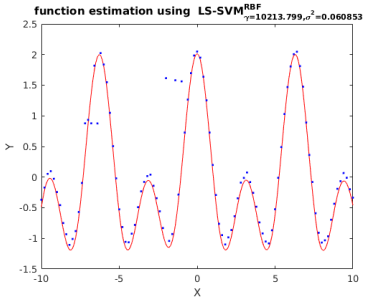
(e) MAE, robust CV, Huber



(f) MAE, robust CV, Hampel

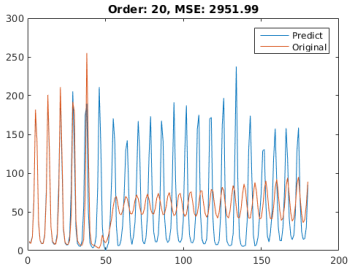


(g) MAE, robust CV, Logistic

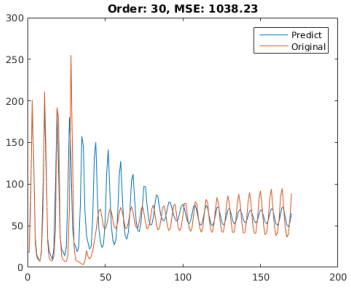


(h) MAE, robust CV, Myriad

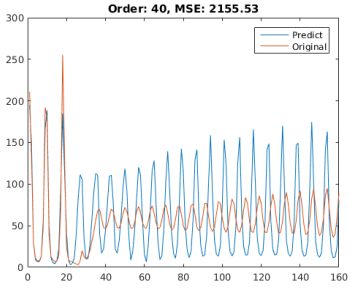
Figure 2.6: Hyper-parameter tuning based on robust cross-validation on a dataset with outlierst



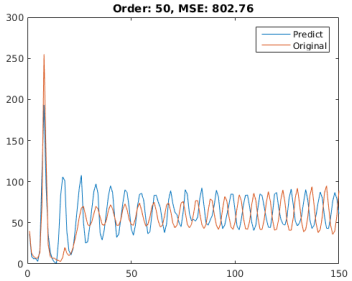
(a) Order: 20



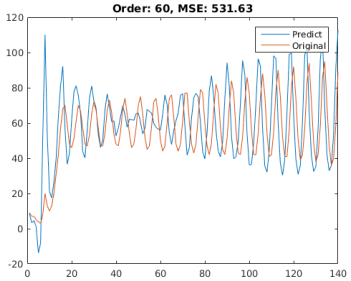
(b) Order: 30



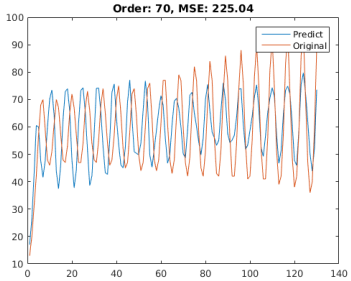
(c) Order: 40



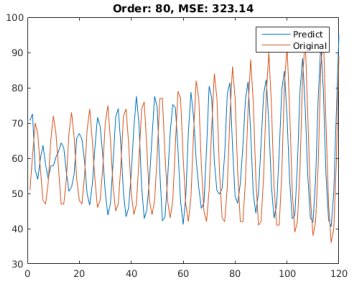
(d) Order: 50



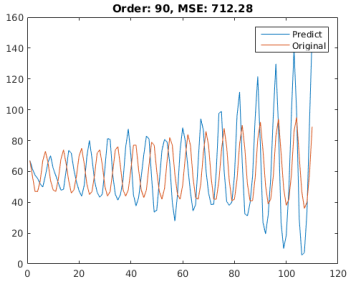
(e) Order: 60



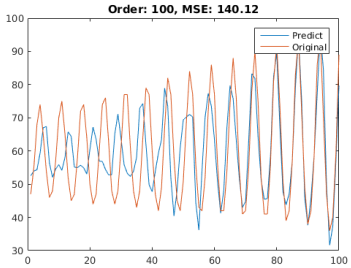
(f) Order: 70



(g) Order: 80



(h) Order: 90



(i) Order: 100

Figure 2.7: Predictions on the Santa Fe test set for different orders

Support Vector Machines: Methods and Applications
Exercise Session 3: Unsupervised Learning

3.1 Kernel Principal Component Analysis

Q. *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?*

Using kernel PCA, taking 1 component only two points are constructed, indicating that we require more components for better reconstruction. We see that with 6 components it is a good approximation of denoised reconstruction. With 10 components we see that noise is slighlty affecting the reconstructed data. With 20 components there is almost no denoising. An optimum value of components can be chosen by observing the cumulative sum of eigen values.

Linear PCA finds the maximum variance in data along orthogonal straight lines. Hence, linear PCA is not suitable for denoising yin-yang dataset. In linear PCA the number of components is equal the dimension of the input space. In K-PCA the number of dimensions is equal to the number of datapoints.

The hyper-parameters can be tuned using cross-validation minimising the reconstruction error. The number of components can be chosen by cumulative sum of eigen values.

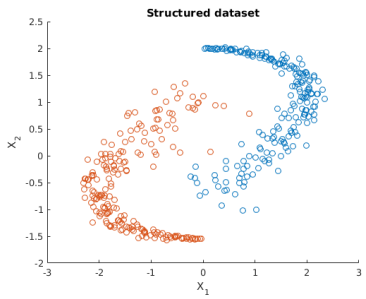
3.2 Handwritten Digit Denoising

In this exercise we add noise to the images of handwritten digits and then try to recover (reconstruct) the digits with K-PCA and PCA.

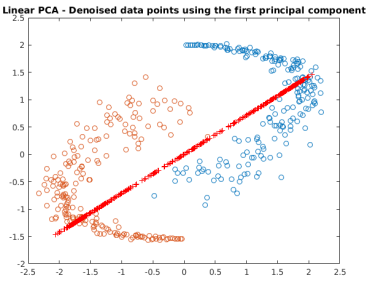
For both Kernal PCA and Linear PCA, the reconstruction is better with increase in number of components. But for given number of components the result of kernel PCA is better than that of linear PCA. But in kernel PCA, strangely the reconstruction of digit 7 looks like digit 2.

3.3 Spectral Clustering

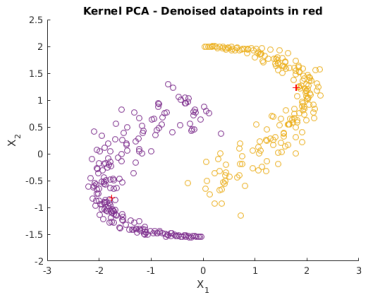
The clustering is performed using RBF kernel and σ^2 plays a role in how the cluster are formed. σ^2 defines how far is the influence of a data point. We find that for lower values of σ^2 the two rings are identified as belonging to different clusters. But with higher value we observe that the clusters are



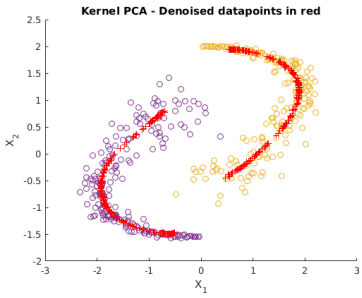
(a) Datapoints



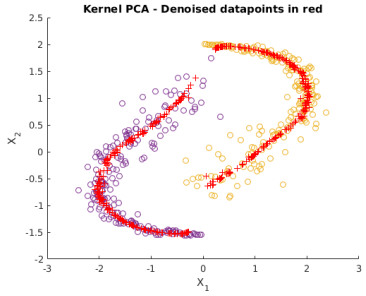
(b) PCA with one principal component



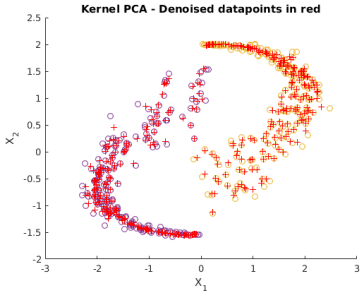
(c) Kpca - 1 component



(d) Kpca - 6 components

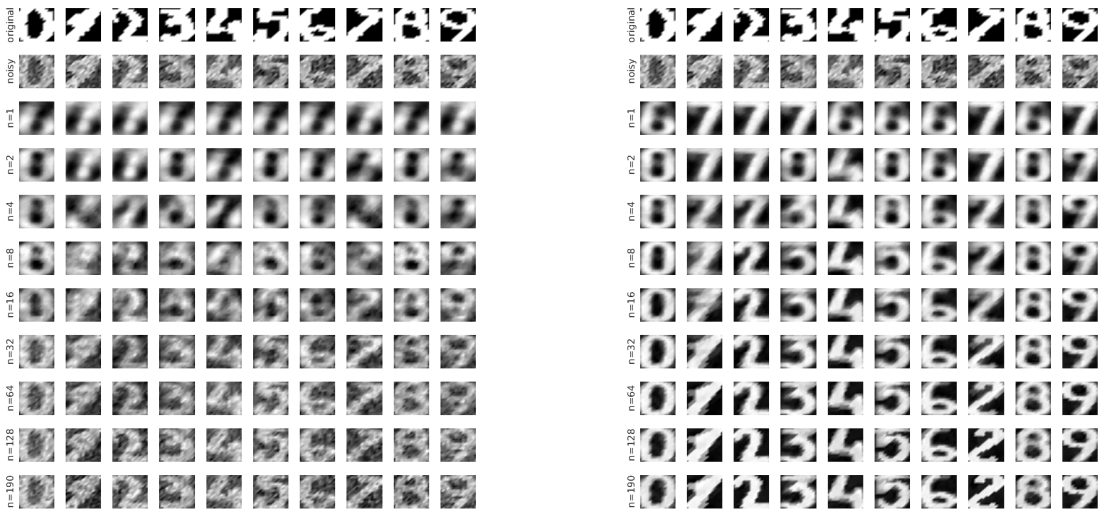


(e) Kpca - 10 components



(f) Kpca - 20 components

Figure 3.1: Denoising Yin-Yang dataset using Kernel PCA by choosing different number of components



(a) Denoising handwritten digits using PCA

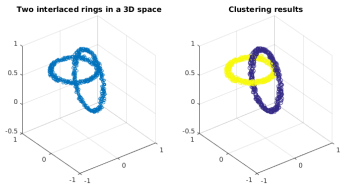
(b) Denoising handwritten digits using Kernel PCA

Figure 3.2: Denoising handwritten digit dataset using Kernel PCA and linear PCA by choosing different number of components

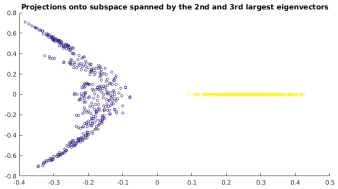
now divided through the middle of the interlaced rings.

3.4 Fixed-size LS-SVM

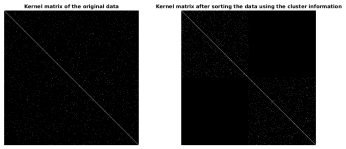
Using the Entropy criterion a subset of 10 points are extracted from 100 datapoints for two different (extreme) values of σ^2 - 0.01 and 100. We observe that for higher values of $sigma^2$ the points of the subet are chosen at the periphery of the dataset, away from the mean. For lower value of σ^2, the datapoints in the subsets are located comparatively nearer to the mean of the dataset.



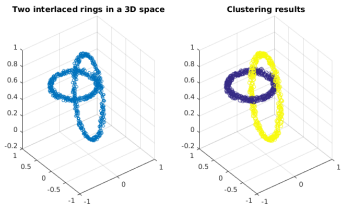
(a) σ^2: 0.001



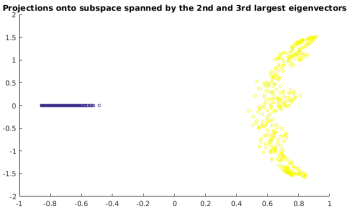
(b) σ^2: 0.001



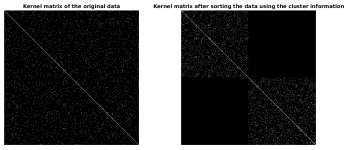
(c) σ^2: 0.001



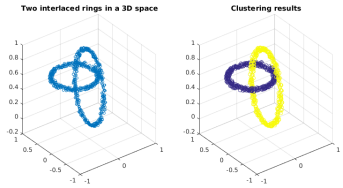
(d) σ^2: 0.005



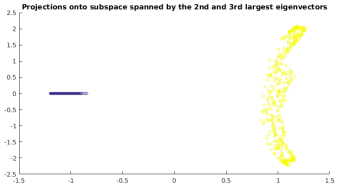
(e) σ^2: 0.005



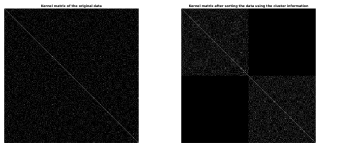
(f) σ^2: 0.005



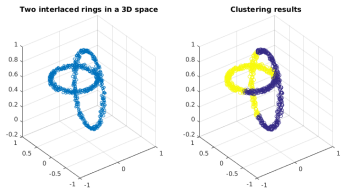
(g) σ^2: 0.01



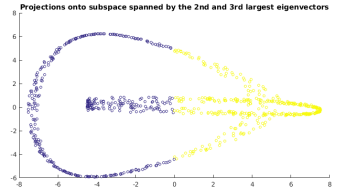
(h) σ^2: 0.01



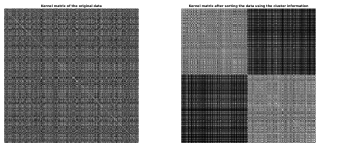
(i) σ^2: 0.01



(j) σ^2: 0.2

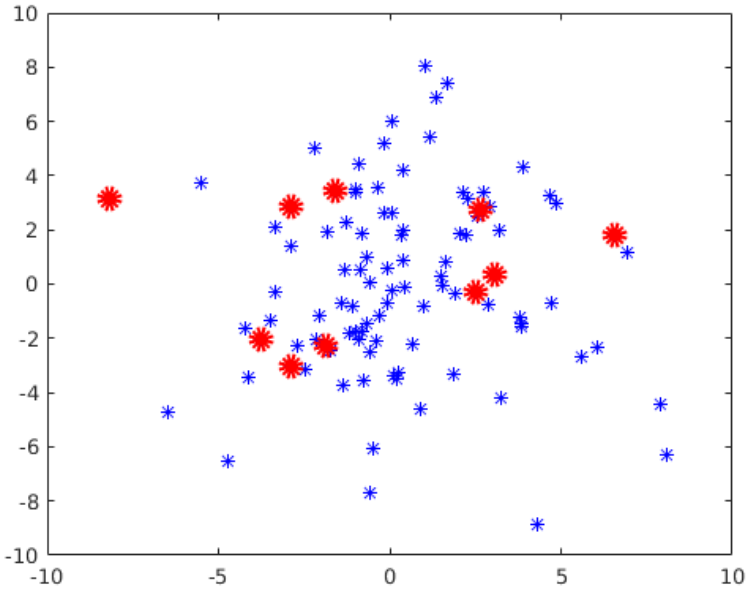


(k) σ^2: 0.2

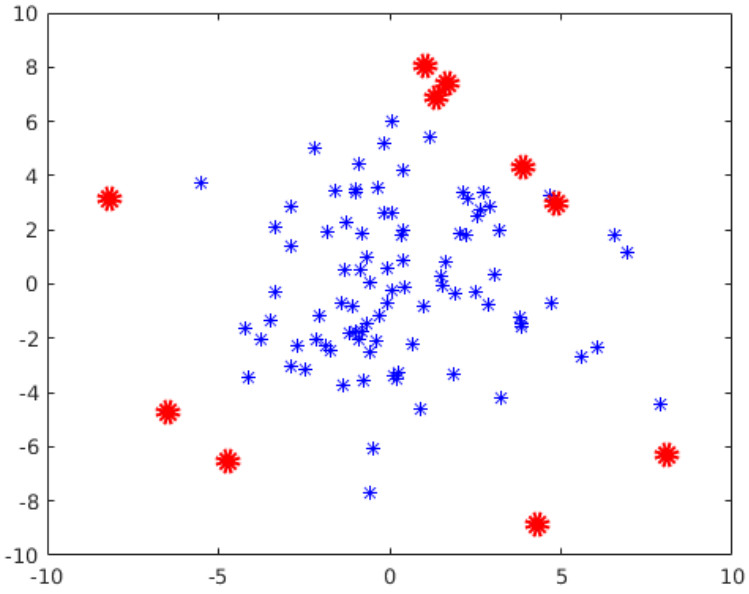


(l) σ^2: 0.2

Figure 3.3: Kernel spectral clustering with different values of σ^2

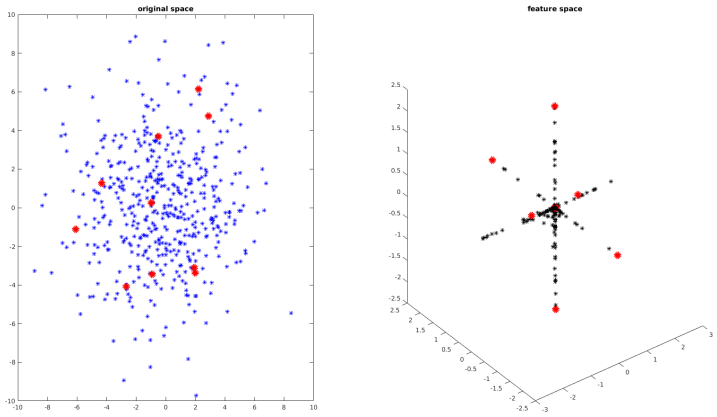


(a) σ^2: 0.01

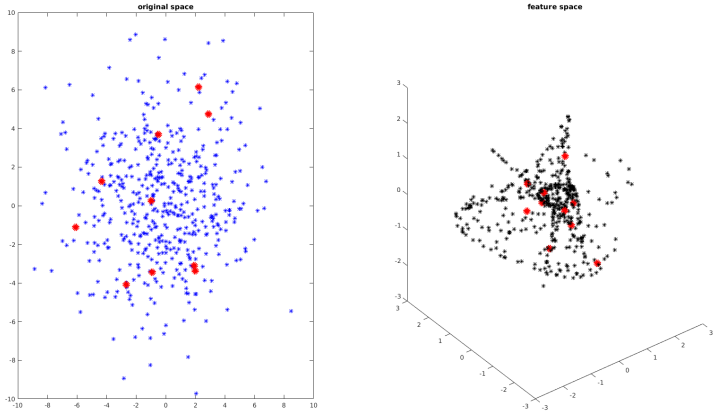


(b) σ^2: 100

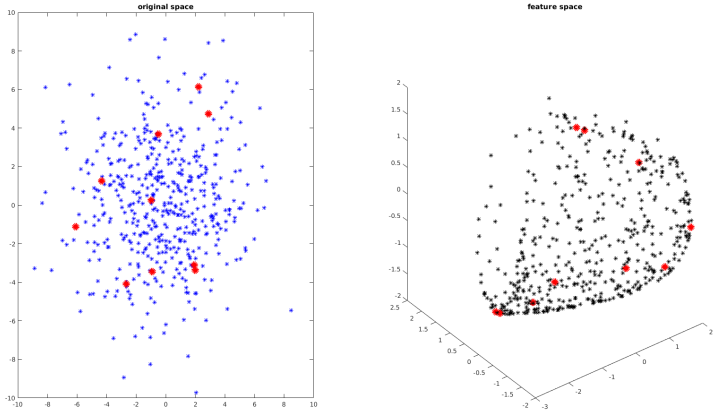
Figure 3.4: Influence of σ^2 on selection of datapoints



(a) σ^2: 1

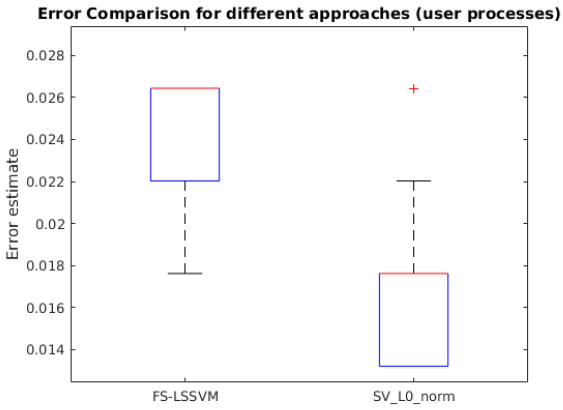


(b) σ^2: 10

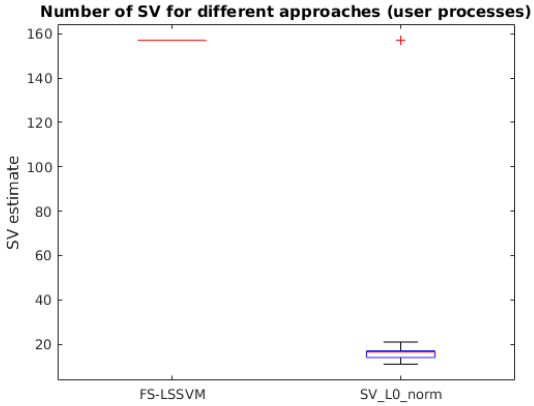


(c) σ^2: 100

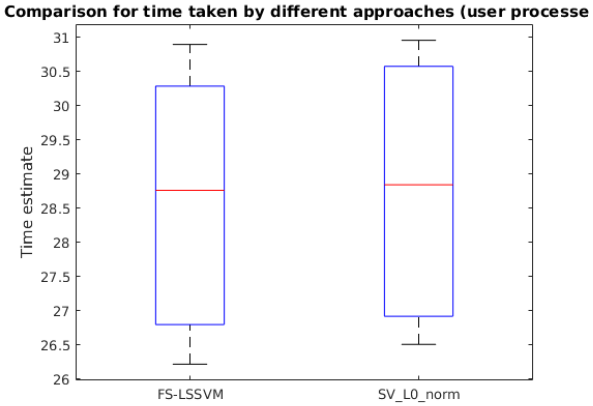
Figure 3.5: Feature space corresponding to different values of σ^2



(a) Error



(b) Number of support vectors



(c) Execution time

Figure 3.6: Comparison of Fixed-size LS-SVM and l_0 -approximation on breast cancer wisconsin data