Contents

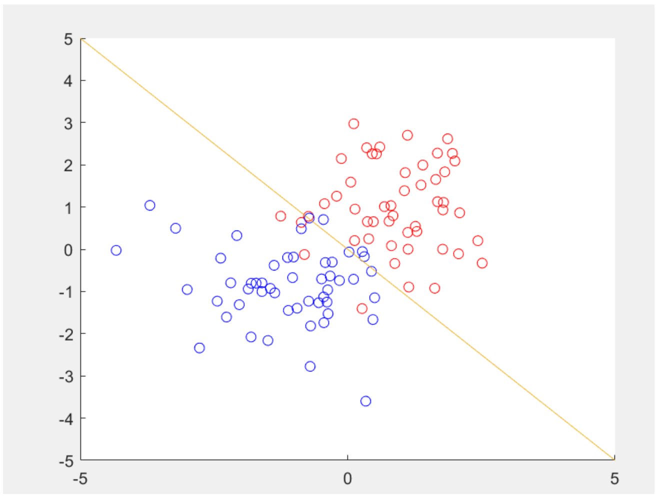
EXERCISE 1

This part is about classification techniques for given data distributions. The exercise begins by generating two Gaussian distributions with the same covariance matrices that are identified by positive/ negative class labels.

1.1 Obtain a line to classify the data by using what you know about the distributions of the data. In which sense is it optimal?

In order to classify this data, I will use the following rule to decide the decision boundary such that:

If the distributions are considered as N(µ1,∑1) and N(µ2,∑2) where µ and ∑ are the mean and variance of the data points, a new data point x will be positive if $|x- \mu 1|_2 < |x- \mu 2|_2$. This means that a data point that is the closest to a particular class will be assigned its label. This is a linearly separable case. I have used y=-x line as a classifier below, consider all labels to the right belong to the positive class. As evident, it is not optimal since it does have a few misclassifications so it can be handled by taking an svm with slack variables parameter or non-linear classifier.



The linear classification can be considered optimal in case of Bayes decision rule. This is because, if the prior class probabilities are equal, the decision rule can be transformed such that:

$$\log \mathrm{P}(C_i|x) \propto -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \frac{1}{2}\log|\Sigma_i| + \log \mathrm{P}(C_i), \quad i = 1, 2.$$

Since $\Sigma_1 = \Sigma_2 = \mathrm{diag}(1,1)$, we can further simplify (1) to

$$\log \mathrm{P}(C_i|x) \propto -\frac{1}{2}\|x-\mu_i\|_2^2 + \log \mathrm{P}(C_i), \quad i = 1, 2.$$

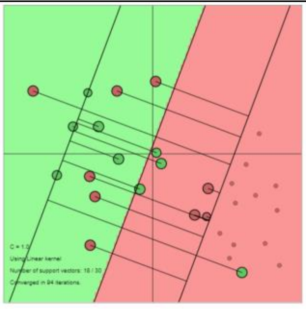
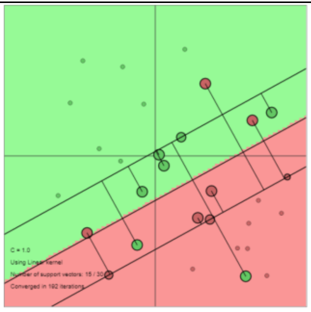
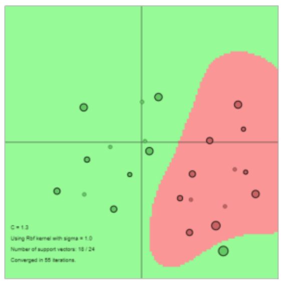
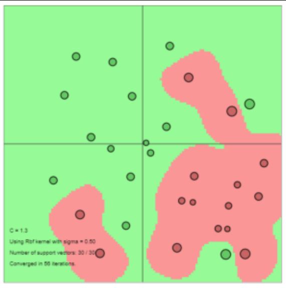
And log P(Ci) is interchangeable with Euclidean distance decision rule. Hence, it is still linear and optimal.

1.2 For each kernel, answer the following questions:

a. What do you observe when you add more data points to the dataset – both on the right and on the wrong side of the hyperplane. How does it affect the classification hyperplane?

b. Try out different values of the regularization hyperparameter C and the kernel parameter sigma. What is the role of the parameters? How do these parameters affect the classification outcome?

Compare classification using the linear kernel with classification using the RBF kernel. Which performs better? Why?

Sr. No.	Linear kernel	RBF kernel
1.2 a	  Adding data on the correct side (left) does not significantly change the margin separating both the classes. There is some misclassification from the starting points that affect the accuracy of this kernel. For the wrong side (right), the model is able to cope well at c=1 despite suffering misclassifications like before.	  Adding data on the wrong side (right) did not affect the model at sigma=1 severely, it quickly adapted to the new data points. However, if I added to a dense cluster of opposite class, it could not rectify the decision boundary unless I changed the sigma. For the correct side (left), there was no significant change apart from the fact that the decision boundaries became closer to the clusters they were classifying.
1.2 b	Changing the ‘c’ parameter tunes the psi variable, which represents misclassification; increasing it did not affect the (left) model above while decreasing it shifted the margin towards the left, touching more points. For the right side, decreasing the c to 0.01 (minimum) provided somewhat similar classification but it used more points to decide the decision boundary.	Increasing the sigma variable reduced the efficiency of the model while decreasing it made the decision boundary come closer to each point individually, hence decreasing the misclassification rate.

From these observations, it is safe to say that RBF kernel is a better classifier for these data points because reducing the sigma variable optimizes the decision boundary for these clusters. However, it should be taken with a pinch of salt since it overfits the data and the kernel may become worse for new data points.

What is a support vector? When does a particular datapoint become a support vector? When does the importance of the support vector change? Illustrate visually.
• What is the role of parameters C and sigma? What happens to the classification boundary if you change these parameters. Illustrate visually.
• What happens to the classification boundary when sigma is taken very large? Why?

Support vectors are those data points that determine the margin for a given classification model.

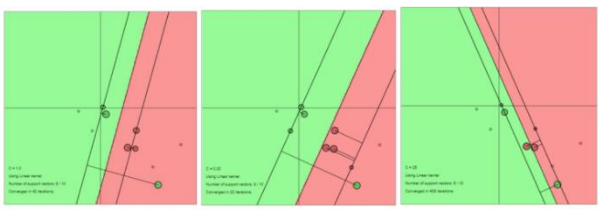


Figure 1 Linear classifier with c=1,20,25

They contribute to the sparsity of the dataset and carry an alpha >=0, located close to the margin (as can be seen by the bigger rounded circles in above images). They are identified by solving the dual representation of the model where the Lagrange multipliers Changing the tuning parameters c and sigma can alter the decision boundary (the number of support vectors being considered in order to tolerate misclassification); if c is large, the penalty for misclassification is similar to regularization in regression analysis and it will lead to better classification (lesser misclassified

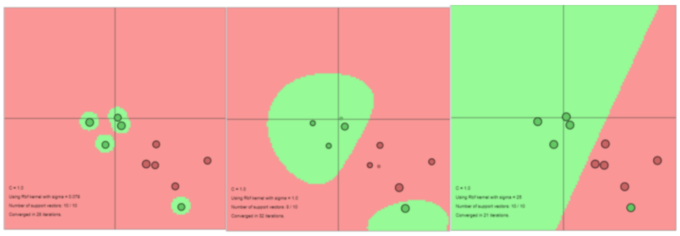


Figure 2 RBF kernel with sigma=0.079, 1, 25

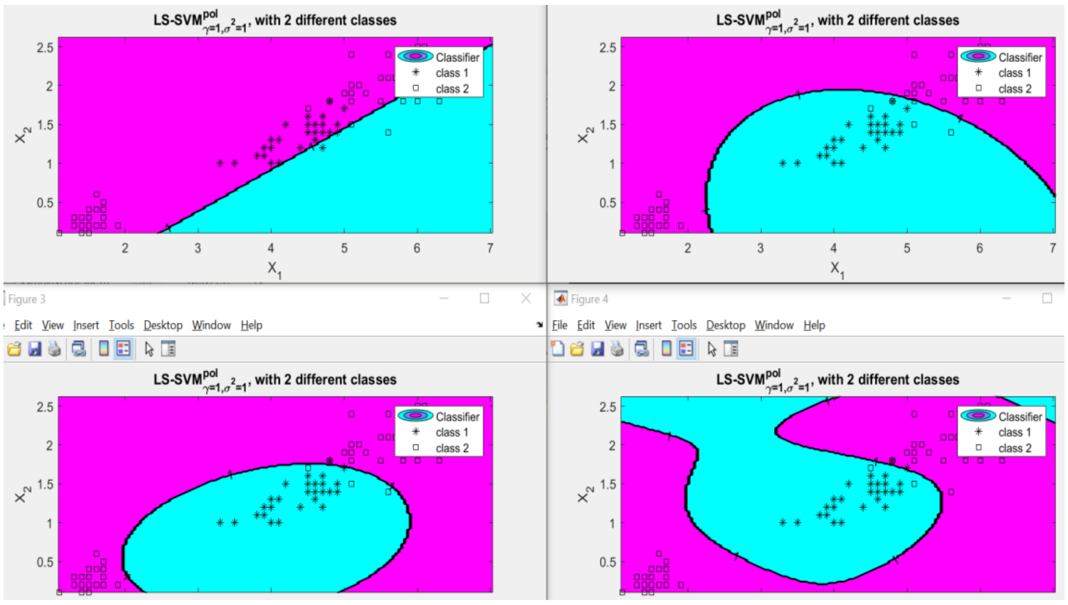
points). Sigma (radius of support vector's influence) changes the sharpness of the boundaries and when it is large, the model will accommodate more data points (more flexible boundary) which may begin to resemble a linear classifier.

1.3 Try out a polynomial kernel with degree = 1*;* 2*;* 3*; : : :* and t = 1 (fix gam = 1). Assess the performance on the test set. What happens when you change the degree of the polynomial kernel?

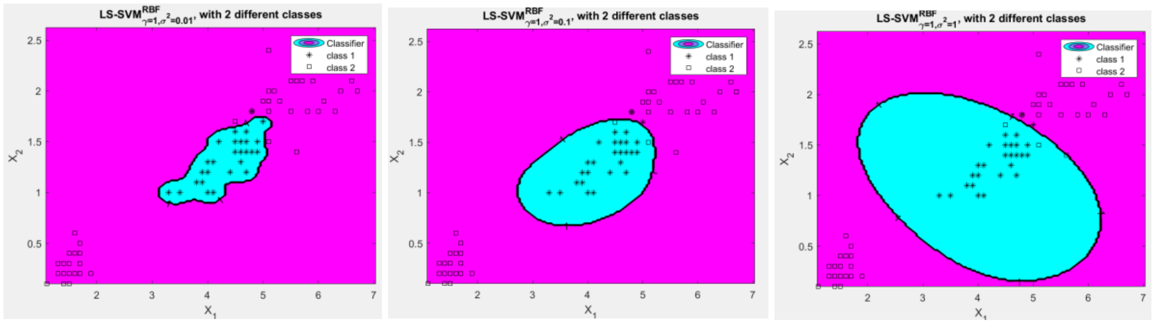
• Let's now focus on the RBF kernel with squared kernel bandwidth *σ*2. { Try out a good range of different sig2 values as kernel parameters (fix gam = 1).

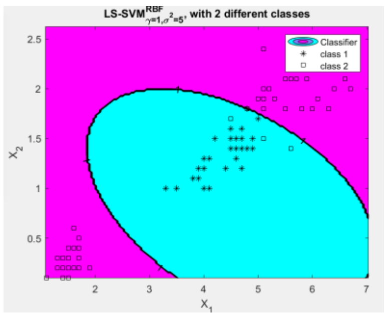
Assess the performance on the test set. What is a good range for sig2? { Fix a reasonable choice for the sig2 parameter and compare the performance using a range of gam. What is a good range for gam?

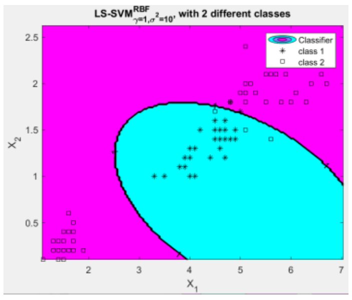
• Compare your results with SampleScript iris.m, which is available on Toledo

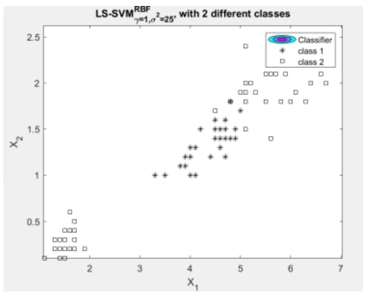


As can be seen above for the iris dataset, changing the degree of LS SVM polynomial function (left to right, ascending degree = 1,2,3,4) changed the classifier system significantly; with a degree of 1, it imitates a linear classifier which is the worst among the other 3 degrees. As the degree increases, the number of misclassified datapoints reduces. The most optimal here seems to be degree = 3.

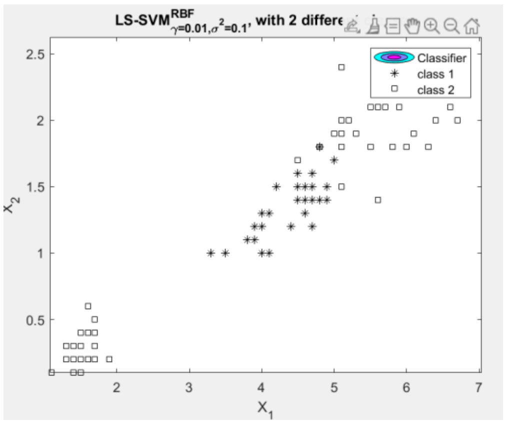


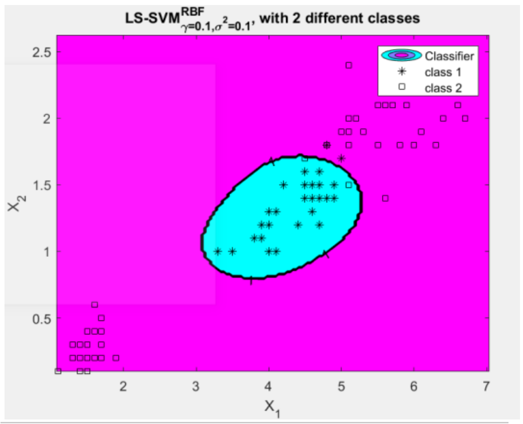


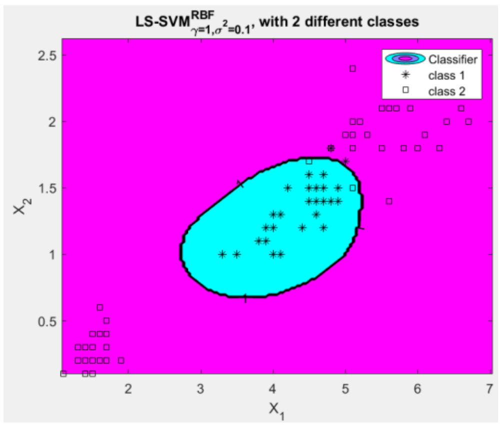


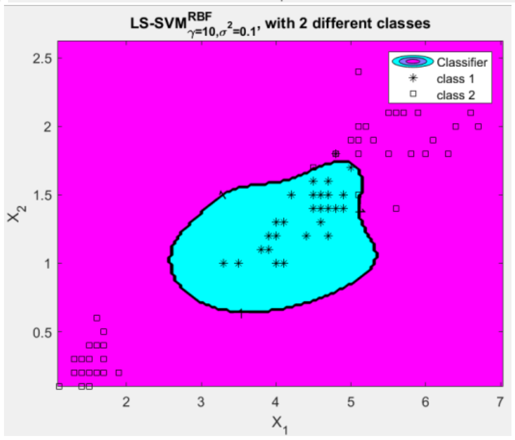


If I increase the value of sigma square in the range [0.01, 0.1, 1, 5, 10, 25] as shown above while gam=1, the model with best performance is at 0.1 as anything less is going to overfit the data while increasing it also increases the error rate. However, the model does not change significantly after sigma square = 5 so the preferable range ofsig2 should be 0.01 to 5.





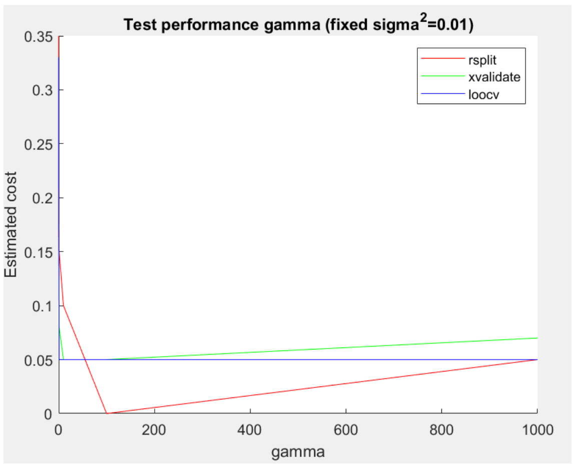


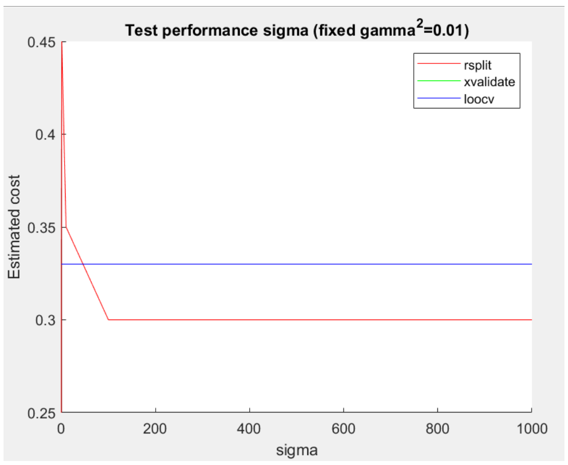


If I fix sig2=0.1 and change value of gam in the range gam = [0.01, 0.1, 1, 10], the model seems to decrease its missclassification after gam-0.1 till gam=100 (tested, not shown above). After that the error rate increases exponentially.

1.3.2 Compute the performance for a range of gam and sig2 values (e.g., *γ; σ*2 = 10-3,. . . , 103). Use the random split method, 10-fold crossvalidation and leave-one-out validation. Visualize the results of each method: do you observe differences? Interpretthe results: which values of gam and sig2 would you choose?

• Why should one prefer crossvalidation over simple validation (random split)? How to choose the value of *k* in *k*-fold crossvalidation?





For changing gamma in the range gam = [0.001, 0.01, 0.1, 1, 10, 100, 1000], the image in left shows how the cost of model is least at 100 for sig=0.01 at rsplit while for fixed gamma = 0.01 and varying sigma in the same range (right), the least cost is at same sigma for rsplit. The cost can be considered akin to accuracy score so in either cases, rsplit leads to a high miscalssification across all ranges of the hyperparameters. On the other hand, xvalidate and loocv have particualr values ay which they suffer high misclassification and then remain comparable to each other. An optimal range to consider thes evalues should be 0.001 to 1000.

The cross validation strategies are better than random splitting because the latter has a drawback of a scenario where all the data points that are the easiest to classify have been provided in the training set which could impact the overall performance of the model. The cross validation methods distribute the data more judiciously and give better performance estimation. The k value can be chosen on the basis of bias variance trade-off where we know that having a large amount of datapoints in training data can lead to high variance and less bias (and vice-versa).

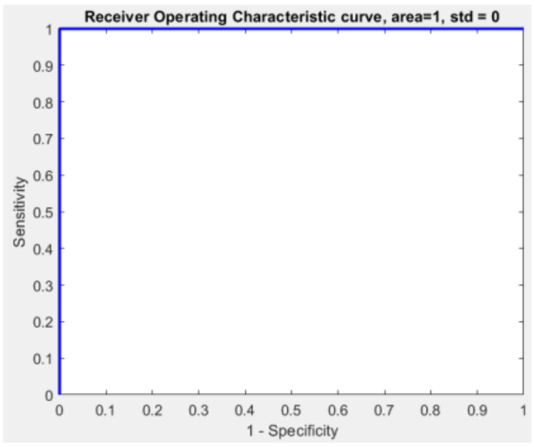
1.3.3 Try out the different 'algorithm'. What differences do you observe? Why do the obtained hyperparameters differ a lot in different runs? What about the cost? Computational speed? Explain the results.

		Run 1			Run 2			Run 3		
		γ	σ^2	Cost	γ	σ^2	Cost	γ	σ^2	Cost
tunels	simplex	4.297	0.062	0.04	0.055	0.165	0.03	0.61	0.06	0.04
svm	gridsearch	6.081	0.408	0.04	0.037	0.782	0.03	16.43	0.07	0.03

As can be seen in the results above, the values of all three parameters changed with every run. The cost remained relatively same for all cases while there is a high variation for other fields. This could be because of the internal randomness of the coupled simulated annealing algorithm that generates randomness to tune the parameters in every run. The computational speed is consistent across these runs.

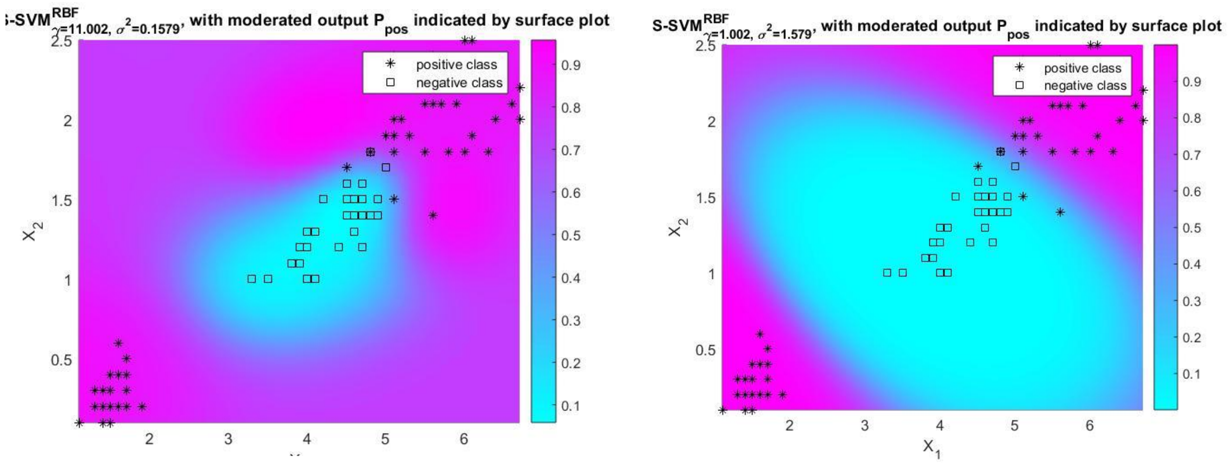
1.3.4 In practice, we compute the ROC curve on the test set, rather than on the training set. Why?
• Generate the ROC curve for the iris.mat dataset (use tuned gam and sig2 values). Interpret the result.

It is optimal to evaluate the classifier model on the test set because if we use the training data here, it will lead to great bias since the model has already been exposed to those data points. If unknown data points are introduced during testing, it will lead to a better generalization and non-biased accuracy scoring.



The ROC curve shows an area of 1, which is a sign that missclassification rate is 0.

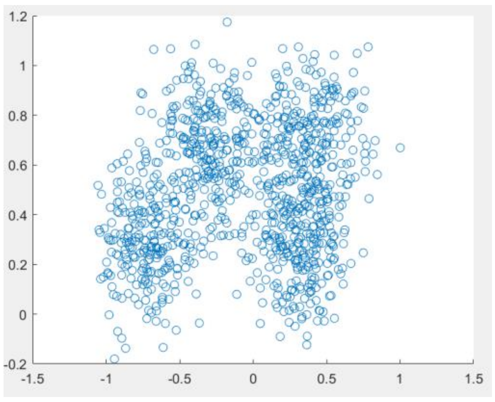
1.3.5 How do you interpret the colors of the plot? Hint: activate the colorbar of the figure.5
• Change the values of gam and sig2. Visualize and discuss the influence of the parameters on the figure.



The positive class is * and the box is negative class, each signified by blue and magenta colors respectively. By changing the parameters (increased sig2 and decreased gam) the probability surface changes such that increasing sig2 makes the boundaries more flexible (blue region expands) while increasing the gam makes the existing boundaries sharper.

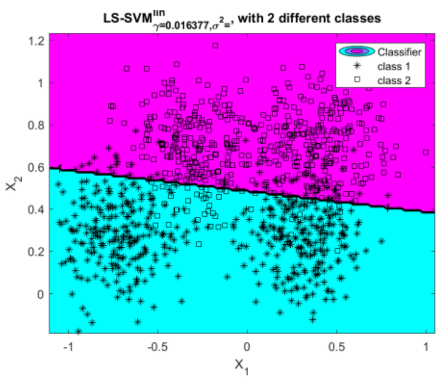
HOMEWORK EXERCISES

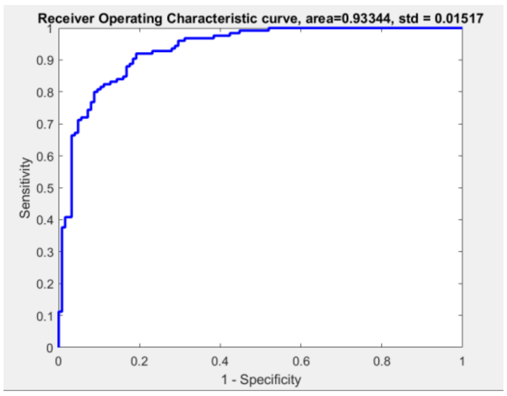
1. RIPLEY



The data is distributed in a characteristic shape which is further classified into 2 categories. There are 1250 datapoints with class labels -1 and 1. A non linear SVM should be best for this kind of distribution.
The 3 models that I tried are: linear, polynomial and RBF kernel. The parameters were tuned using simplex and the final model which performed best was Polynomial.

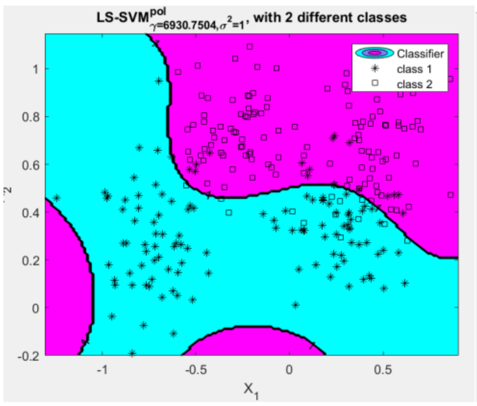
LINEAR:

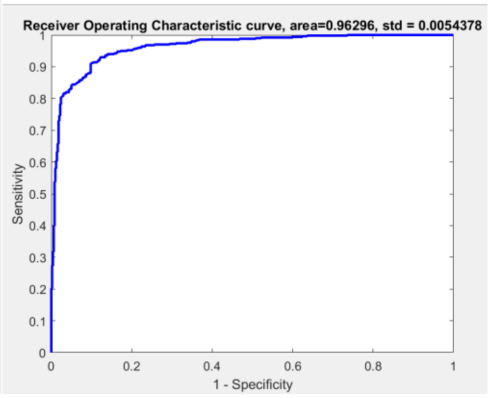




[gam] : 0.016

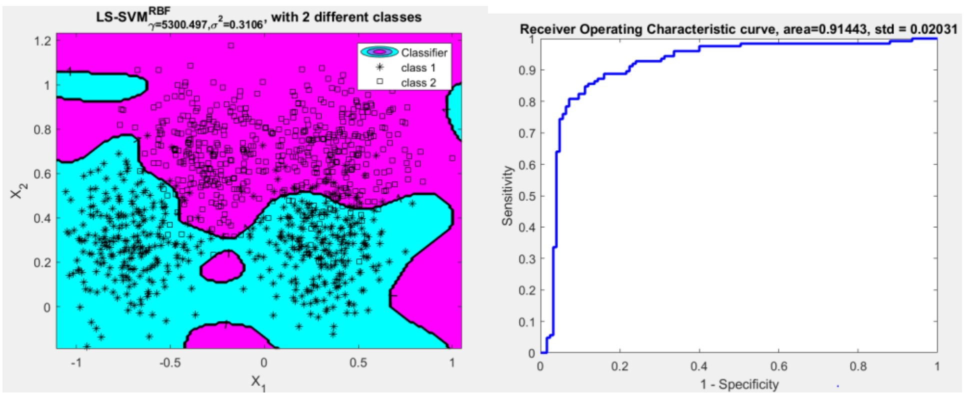
POLYNOMIAL:





[gamma t degree]: 6930.7504 1.636346 5

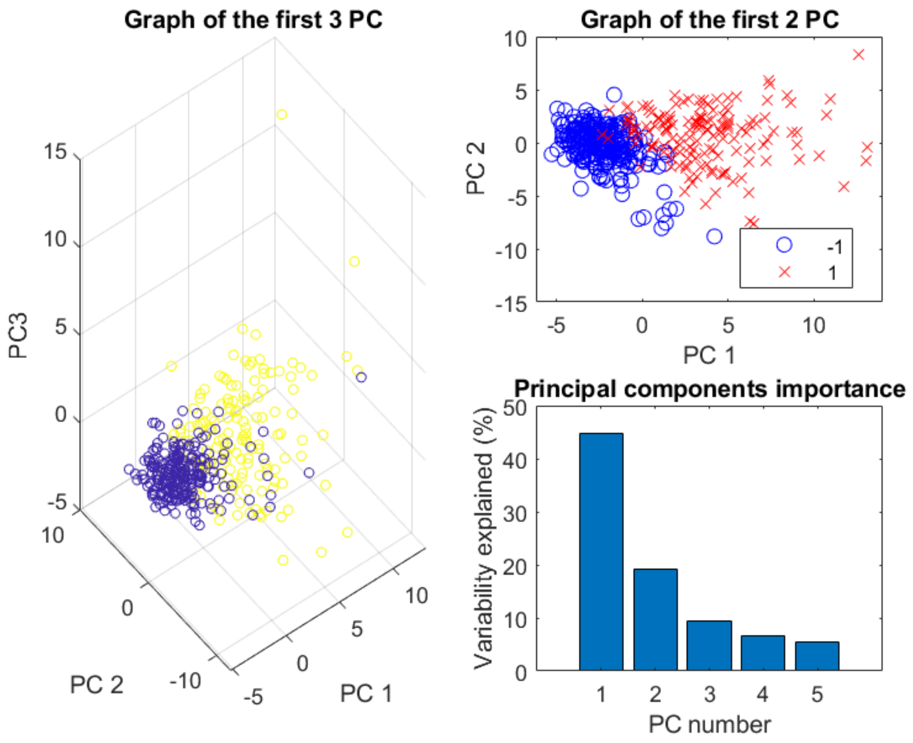
RBF:



[gam]	5300.497
[sig2]	0.3106

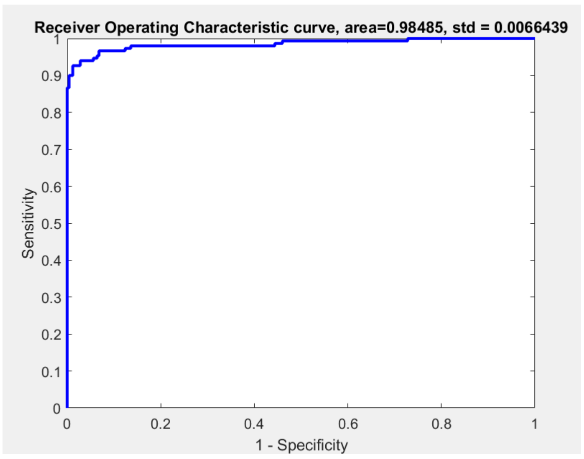
In this case, polynomial function performed the best with highest area under the curve. RBF follows pretty close as well and its classification seems more intuitive too but the polynomial kernel gets more accurate cases.

2. WISCONSIN



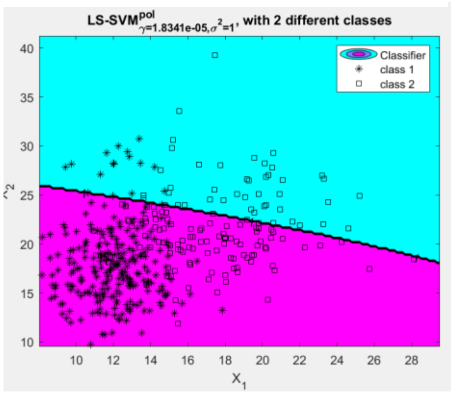
Since this a high dimensional data (569x30) I used PCA to visualize the distribution of data. There are 3 PCs that can explain most of the variability which allows some scope of linear separability.

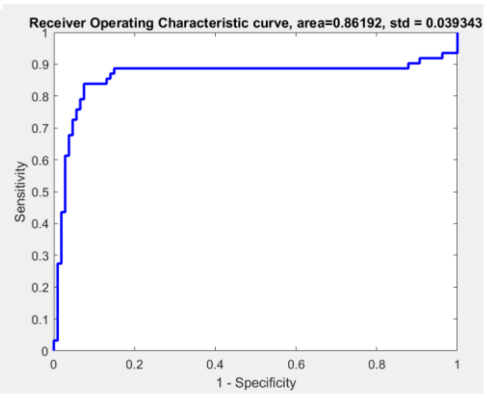
LINEAR:



Gam=0.0638

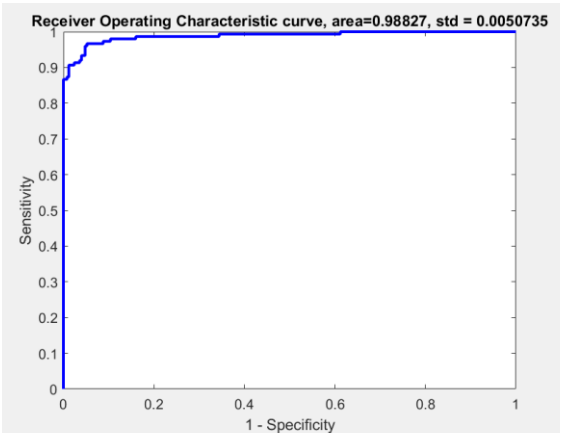
POLYNOMIAL:





[gamma t degree]: 1.834053e-05 113.1728 3

RBF:

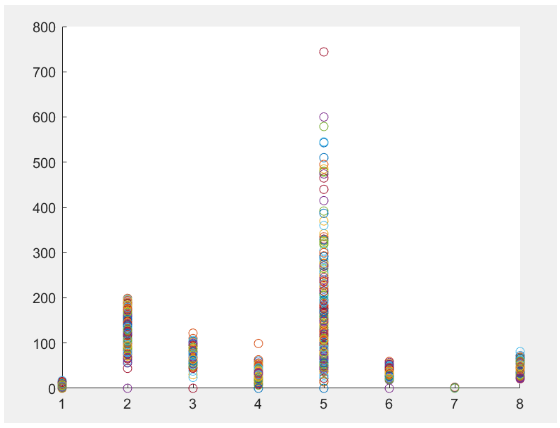


[gam] 78.8037
[sig2] 463.4064

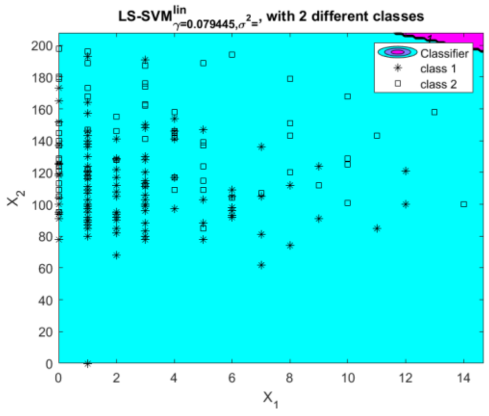
The RBF kernel has greatest area under the curve with its detected hyperparameters, hence it is the best choice for this dataset.

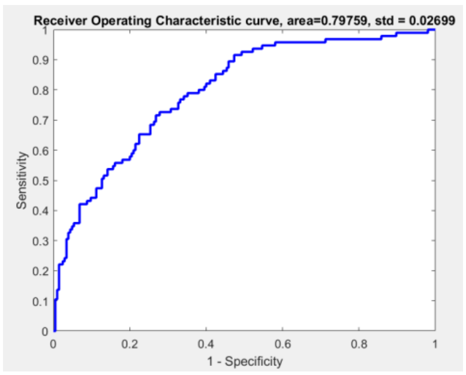
3. DIABETES

This is a large dataset as well (600x8) which can be seen in the scatterplot below. It is a binary classification task that can be analysed against the 3 models:



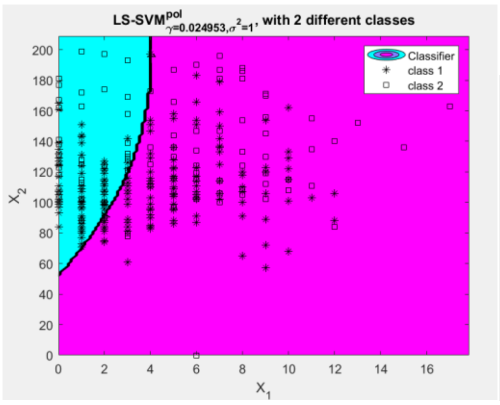
LINEAR:

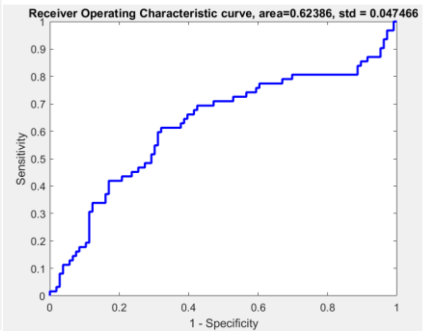




Gam: 0.079

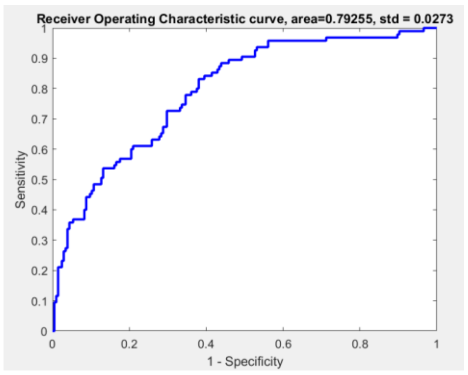
POLYNOMIAL:





[gamma t degree]: 0.024953 6.9572 3

RBF:



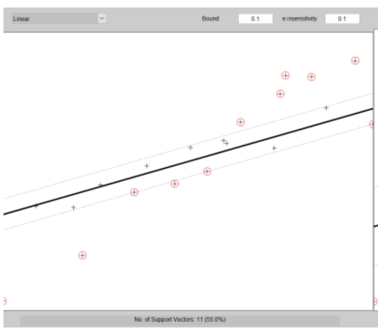
[gamma sig2]: 229.263247 17580.1711

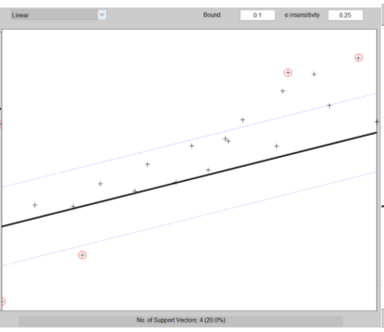
This dataset performs best with RBF classifier but it still has a relatively bad accuracy as compared to the other datasets. I am not satisfied by the performance of this model, it likely requires a more customized non linear model.

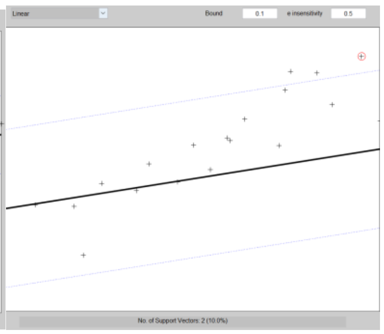
EXERCISE 2

This section is dedicated to function estimation and time series prediction.

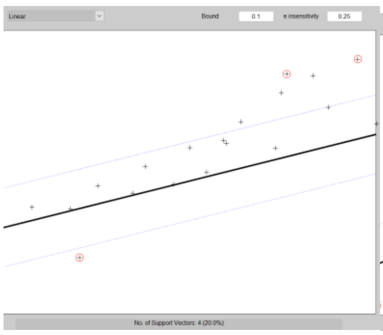
2.1 Construct a dataset where a linear kernel is better than any other kernel (around 20 data points). What is the influence of e (try small values such as 0:10; 0:25; 0:50; : : :) and of Bound (try larger increments such as 0:01; 0:10; 1; 10; 100). Where does the sparsity property come in?
• Construct a more challenging dataset (around 20 data points). Which kernel is best suited for your dataset? Motivate why.
• In what respect is SVM regression different from a classical least squares fit?

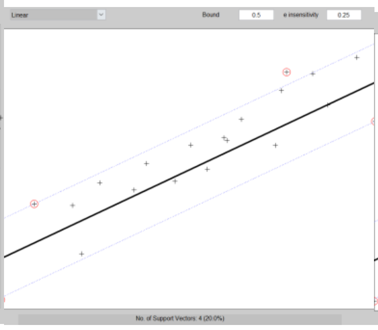

Linear Bound 0.1 e insensitivity 0.1
No. of Support Vectors: 11 (55.0%)

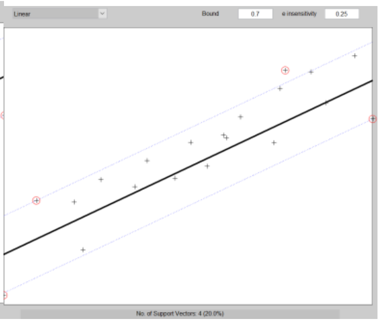

Linear Bound 0.1 e insensitivity 0.25
No. of Support Vectors: 4 (20.0%)


Linear Bound 0.1 e insensitivity 0.5
No. of Support Vectors: 2 (10.0%)

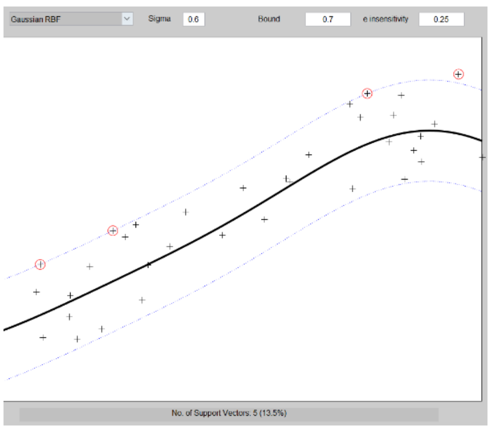
Changing the e sensitivity in the range [0.1, 0.25, 0.5] with lower bound = 0.1 decreases the number of support vectors being considered for the linear model. This is the sparsity property as the epsilon tube broadens to accommodate lesser support vectors.


Linear Bound 0.1 e insensitivity 0.25
No. of Support Vectors: 4 (20.0%)


Linear Bound 0.5 e insensitivity 0.25
No. of Support Vectors: 4 (20.0%)


Linear Bound 0.7 e insensitivity 0.25
No. of Support Vectors: 4 (20.0%)

On setting e sensitivity to 0.25 and shifting bound in the range [0.1, 0.5, 0.7], the boundary comes closer to the data points until it touches one, otherwise it tries to maintain a distance between the 2 classes.


Gaussian RBF Sigma 0.6 Bound 0.7 e insensitivity 0.25
No. of Support Vectors: 5 (13.5%)

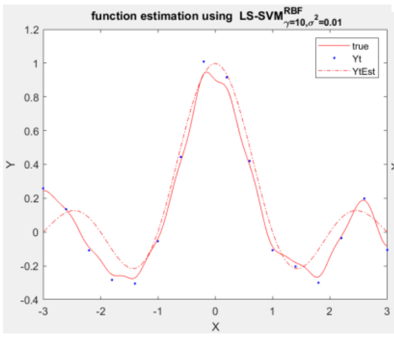
For more data points, an RBF kernel was able to model the distribution best with the parameters sig = 0.6, bound = 0.7 and e sensitivity = 0.25.

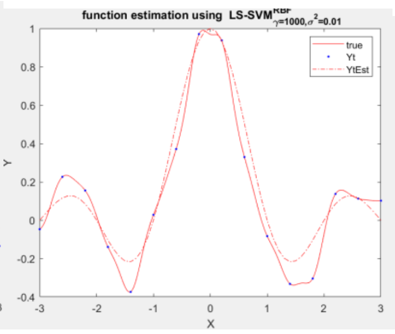
Classical regression with least squares uses all data points to regress the model while SVM regression uses select few support vectors. The latter uses e sensitive value as loss function while least squares uses a Euclidean distance between the line and each data point.

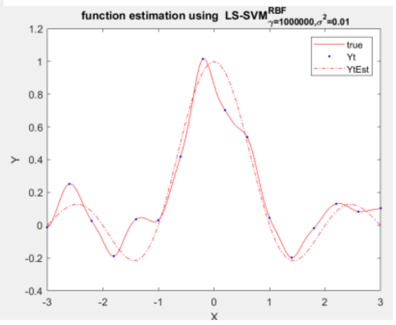
2.2.1 Try out a range of different gam and sig2 parameter values (e.g., gam = 10; 103; 106 and sig2 = 0:01; 1; 100) and visualize the resulting function estimation on the test set data points. Discuss the resulting function estimation. Report the mean squared error for every combination (gam, sig2).

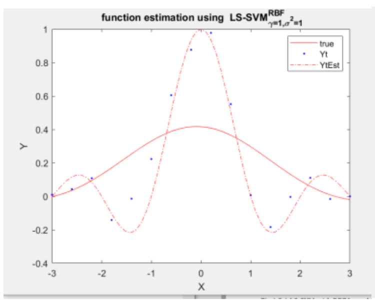
• Do you think there is one optimal pair of hyperparameters? Argument why (not).

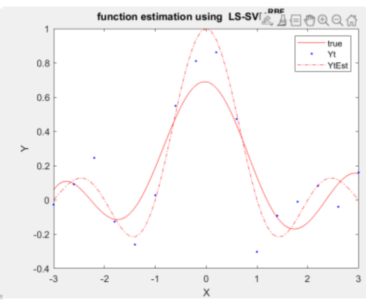
• Tune the gam and sig2 parameters using the tunelssvm procedure. Use multiple runs: what can you say about the hyperparameters and the results? Use both the simplex and gridsearch algorithms and report differences.

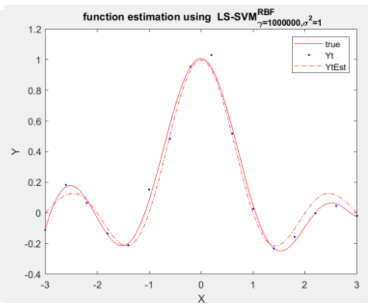


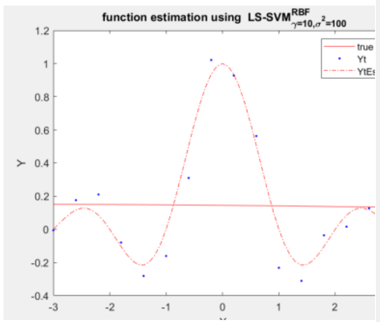


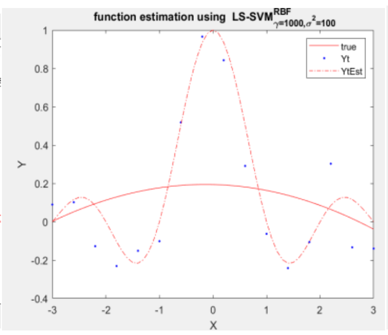


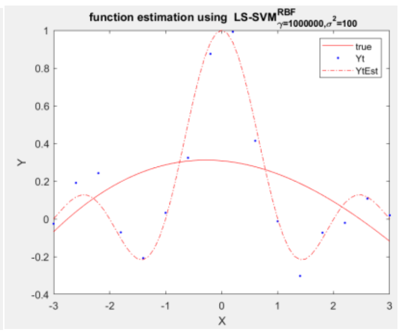












	Sig2 = 0.01	Sig2 = 1	Sig2 = 100
Gam = 10	0.027	0.0886	0.118
Gam = 1000	0.0128	0.0267	0.1365
Gam = 1000000	0.0105	0.0079	0.1052

Large values of sig2 affect the function estimation negatively as the MSE tends to increase, as visible in the graphs above. An optima is achieved at gam=1000000, sig2 = 1. This means that choosing these hyperparameters will lead to a good trade-off between the MSE and smoothness of the curve. There could very well be other optimal hyperparameters as the tuning may be done by tunnelsvm, in which case the CSA algorithm selects random starting points and may affect the consequent results. For instance:

		Run 1			Run 2			Run 3		
		γ	σ^2	Cost	γ	σ^2	Cost	γ	σ^2	Cost
tunel ssvm	simplex	53.5012	0.3966	0.0097	40.1719	0.3811	0.00997	22.1992	0.3648	0.00988
	Grid search	1063.1458	0.3291	0.0099	67.7362	0.4112	0.0097	33.2883	0.3606	0.0099

The cost function here remains constant while sig2 is also consistent [03-0.4]. Gam on the other hand, varies a lot even though one could say that the gridsearch is more significantly affected than simplex.

2.2.2 Discuss in a schematic way how parameter tuning works using the Bayesian framework. Illustrate this scheme by interpreting the function calls denoted above.

A summary of the bayesian framework is given as:

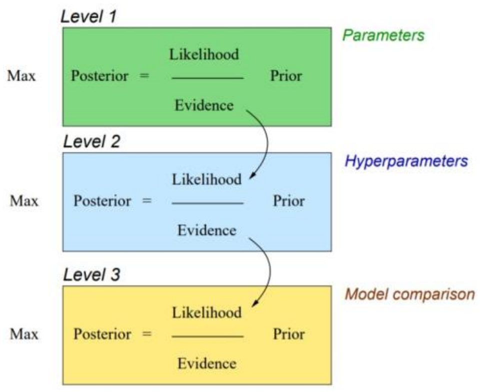
$$Posterior\ Probabilioty = \frac{\text{Likelihood}}{\text{Evidence}} * Prior$$

The Bayesian LS-SVM is formulated using three levels of posterior distribution estimation

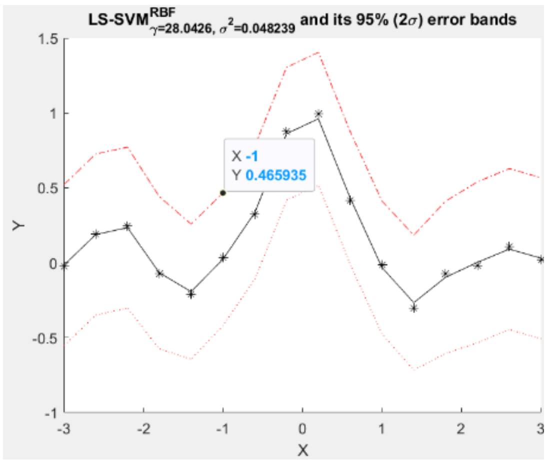
Level 1: the parameter associated with the model, alpha and b are estimated using the posterior probability.

Level 2: The hyperparameters gam and sig^2 are estimated by the posterior probability.

Level 3: The models are compared by the cost function using the posterior probability.

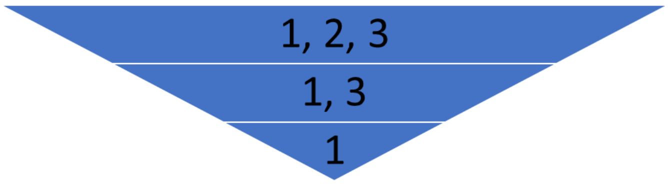

Level 1
Max
Posterior = Likelihood / Evidence Prior
Parameters
Level 2
Max
Posterior = Likelihood / Evidence Prior
Hyperparameters
Level 3
Max
Posterior = Likelihood / Evidence Prior
Model comparison

It is represented by 3 layer schema where an application of bayes rule implies using a prior distribution from the layer above. This involves likelihood and evidence of data, the prior for the first layer being the regularization in primal form of SVM (least squares term will be the likelihood). Then, a log of the bayes rule is taken, whose parameters are b and w, can be estimated. So the posterior is represented with these terms and the second layer can be estimated in terms of zeta and mu (apply bayes rule). Finally, the third level of inference estimates kernel parameters.



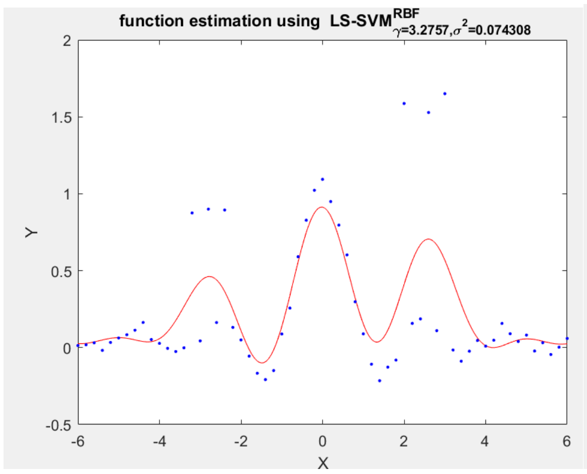
2.3 Visualize the results in a simple figure. How can you do input selection in a similar way using the crossvalidate function instead of the Bayesian framework?

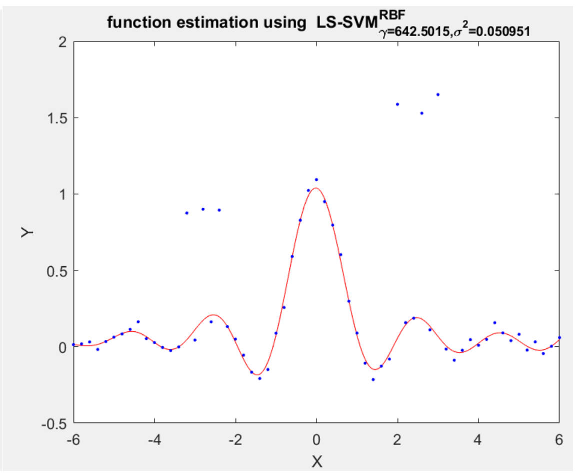
ARD is done using backward selection so all the parameters are considered in the beginning, then they are ranked and selected according to their significance to the model. This can be visualized as:



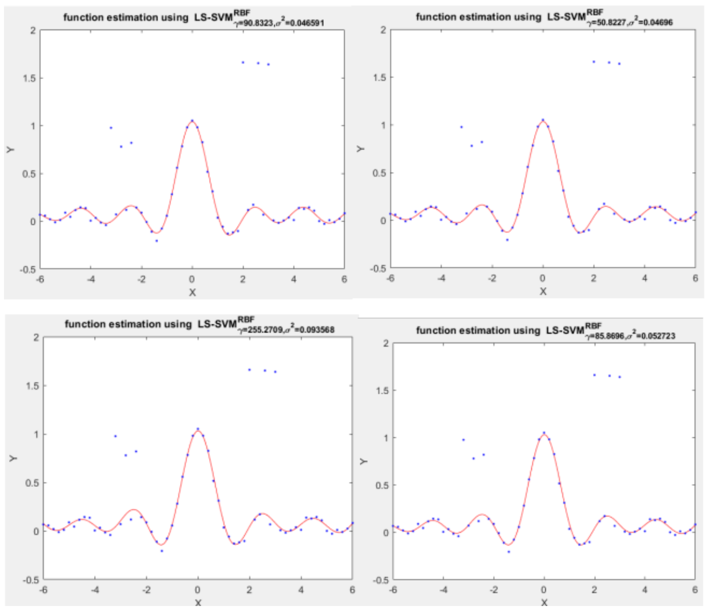
Cross validate can be used to find the significant features in a similar fashion using recursive feature elimination. Here, a feature is ignored and the cost function is calculated; if it decreases sharply then the feature is reconsidered and its impact on the cost can be determined. But unlike bayesican framework ARD, it cannot be ranked.

2.4 Visualize and discuss the results. Compare the non-robust version with the robust version. Do you spot any differences? 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' and 'wmyriad'. Report on differences. Check the user's guide of LS-SVMlab for more information





Robust regression considers the noise (non-Gaussian) and outliers in data by using LS-SVM. It finds a ronust estimation of the interquantile range of error variables and assigns a weight to each point, hence detecting the outstanding ones (outliers). If we compare a robust (right, above) to the model using LS-SVM without outlier detection (left), the shape of the model is enough to describe the adversity of a non-robust version. The mean absolute error is preferred over MSE because the squared error term results in a lesser value (less computational requirement). The mean square error takes the square of the distance so the outlier influence on the function is more, model can be swayed away from the true fit.



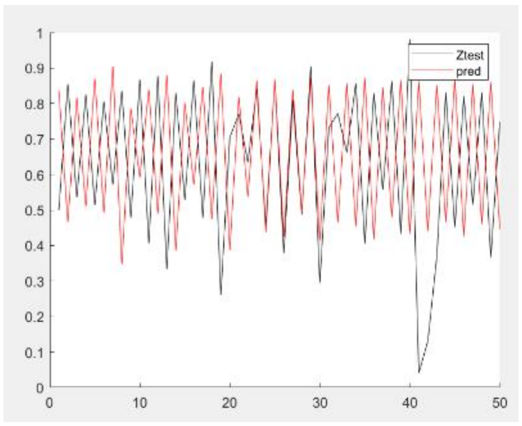
Changing the regularization function (myriad,logistics,hampel,huber) does not drastically differ in performance (detecting outliers).

HOMEWORK EXERCISES

Time series prediction

1. **LogMap dataset**
    a. **As indicated numerous times before, the parameters gam and sig2 can be optimized using crossvalidation. In the same way, one can optimize order as a parameter. Define a strategy to tune these 3 parameters**

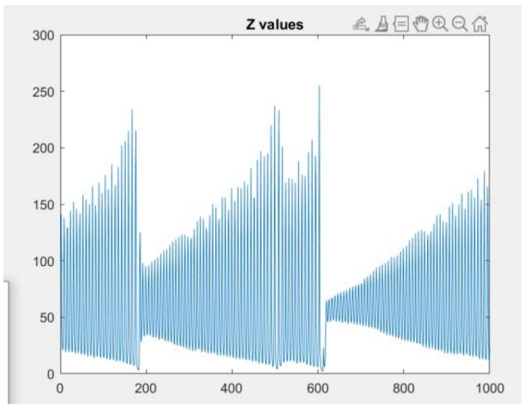
    b. **Do time series prediction using the optimized parameter settings. Visualize your results. Discuss.**



In order to tune sig, gam and order parameters, I will first fix a certain order and then apply tunnelsvm to determine the appropriate gam and sig2 for it. Then I will check the MAE/MSE of prediction against the known test data to find the optimum choice of hyperparameters. This can be set on a loop or even done manually, I found an optima with hit-and-trial.

An initial run with order = 10, gam=sig2=10. Here, MAE=0.647 and MSE = 0.1303. Then, I applied the setting for multiple gam and sig values as I did before in section 2.2.1 and got an optima at gam=1000000, sig2= 1000 and order = 20.

2. **Santa Fe dataset**
    a. **Does order = 50 for the utilized auto-regressive model sounds like a good choice?**
    b. **Would it be sensible to use the performance of this recurrent prediction on the validation set to optimize hyperparameters and the model order?**
    c. **Tune the parameters (order, gam and sig2) and do time series prediction. Visualizeyour results. Discuss.**

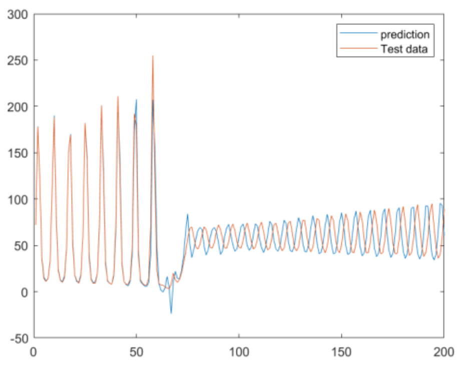


This dataset (left) seems to have a periodic distribution; I optimized the RBF kernel sig2 and gam using tunnelsvm + crossvalidation (10 fold) on the training data. To get the order, I used hit-and-trial among different values and found the least MSE/MAE.

I achieved the following results (image below):

[gamma sig2 order]: 7.59222 31.2 50

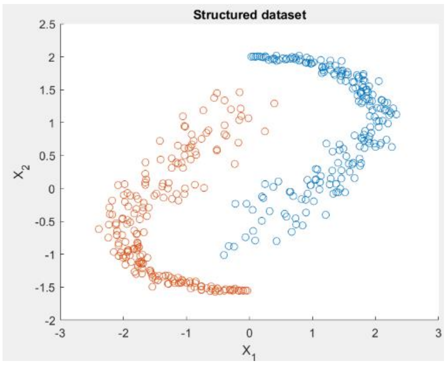
The order 50 found for the regressive model may be the optimal value or at the very least, very close to the optimal value. It has MAE = 57.7 and MSE = 476.4. In hindsight, using cross validation may not be a good choice because the training set may be a small part (remote region) in a time series data while the test comes from another highly variant kind of distribution but in continuation. This may adversely impact the predictions. As evident in this image, the future 'oscillations' of the data change drastically in the amplitude after a certain point.



EXERCISE 3

This exercise is about kernel PCAs and denoising data using them.

3.1 Describe how you can do denoising using PCA. Describe what happens with the denoising if you increase the number of principal components.
• Compare linear PCA with kernel PCA. What are the main differences? How many principal components can you obtain?
• For the dataset at hand, propose a technique to tune the number of components, the hyperparameter and the kernel parameters.



PCA is a data ordination technique that transforms a high dimensional matrix into a reduced, more interpretable form of data. It uses orthogonal linear transformations to find data projection of k dimensions after looking at the variance contained in different principal vectors. The one with least explained variance is the one that contains noise. In kernel PCA, the data is mapped through a non-linear transformation into a higher dimensional space using a kernel where the PCA can be applied to non-linearly reconstruct the data (by discarding the noisy components). Increasing PCs will overfit the data for a classification/prediction model as more of the components explaining the variance in the data will become part of the training set, adversely affecting noise detection. Given below is a comparison of how linear PCA differs with kernel PCA depending on the number of PCs selected.

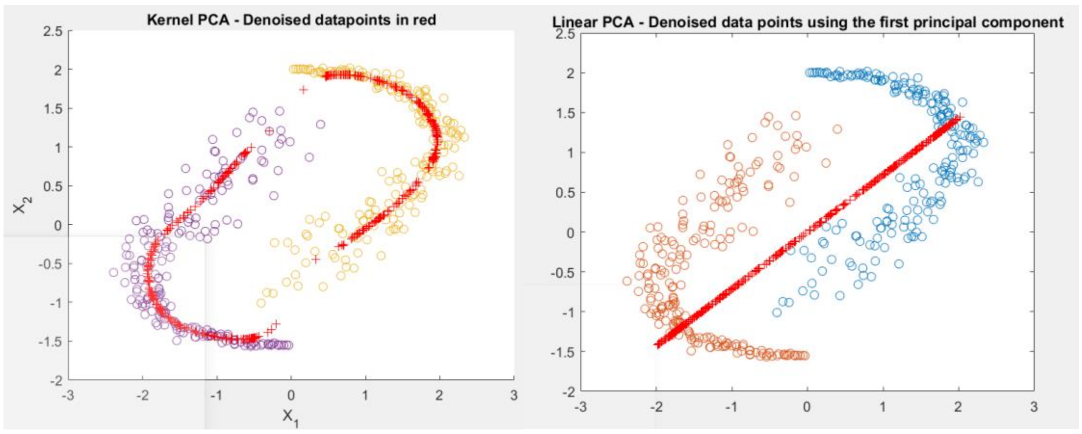


Figure 3 Kernel PCA with 6 PCs (L) Linear PCA with bad noise detection (R)

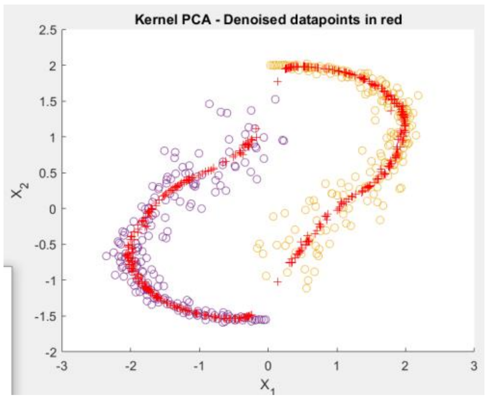


Figure 4 Kernel PCA with 10 PCs

In kernel PCA, there is dimensional reduction of feature space while linear PCA works directly on the data. The maximum number of components in linear PCA is limited to the variables present in the data while for kernel PCA, it is the size of training set. In the latter, it is possible to result in more PCs than the original dataset's variables.

To tune the parameters, we can perform cross validation and look for optimal solutions based on minimizing error (criteria).

3.2 Explain briefly how spectral clustering works.

- **What are the differences between spectral clustering and classification?**
- **Edit the script and try different values of sig2 (e.g., 0.001, 0.005, 0.01, 0.02). What is the influence of the sig2 parameter on the clustering results?**

Spectral clustering is like weighted PCA where a Laplacian matrix of the kernel function is decomposed to get the sign of eigenvectors representing the identity/class of each datapoint. It is unsupervised unlike classification, such that the class labels are not known beforehand. Assessing accuracy is difficult here (compared to classification techniques) and solely depends on the similarity criterion between clusters. Lower values of sigma give good accuracy (different rings classified into separate classes) which otherwise becomes inaccurate with high sig2.

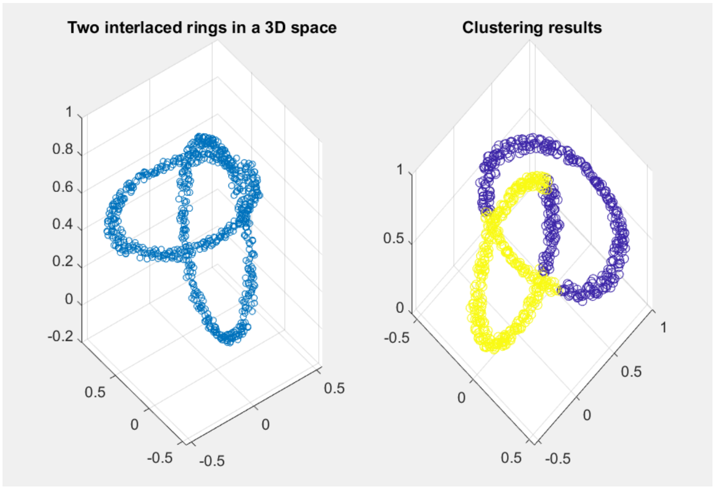


Figure 5 Clustering at sig2 = 0.05

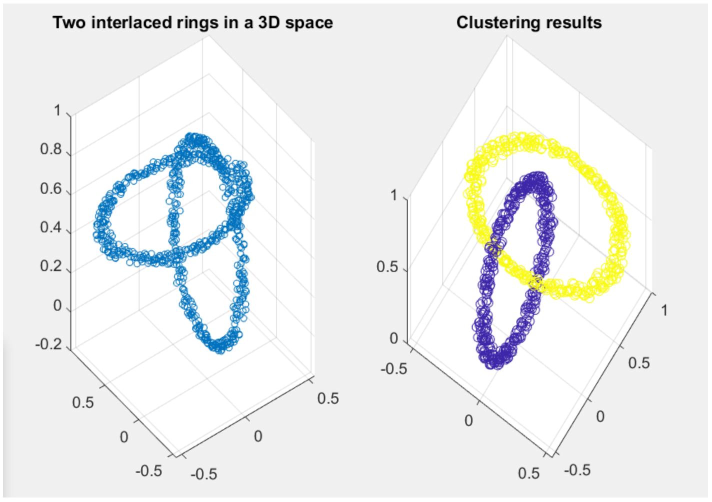


Figure 6 Clustering at sig2 = 0.005 (better)

3.3 In which setting would one be interested in solving a model in the primal? In which case is a solution in the dual more advantageous?

• What is the effect of the chosen kernel parameter sig2 on the resulting fixed-size subset of data points (see fixedsize script1.m)? Can you intuitively describe to what subset the algorithm converges?

• Run fslssvm script.m. Compare the results of fixed-size LS-SVM to L0-approximation in terms of test errors, number of support vectors and computational time

The advantage of SVM is that the same data can be represented as primal and dual depending on the requirement of the question. If we are working with lower dimensional data, computing the scalar product $w^T x$ is easy and does not require the need for support vectors. A dual is required when alpha values can give a better generalization of the solution rather than including all the dimensions of the original dataset (low training set data, more dimensions).

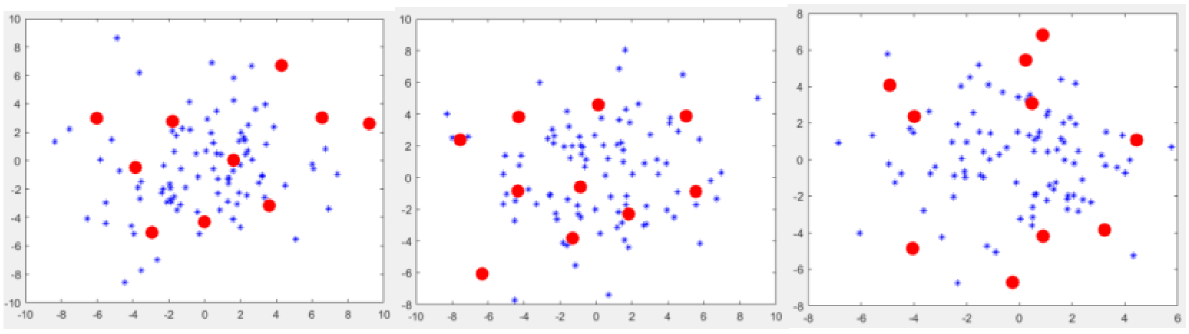
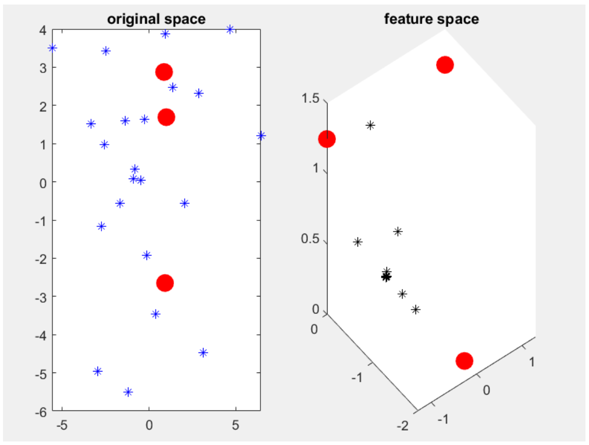
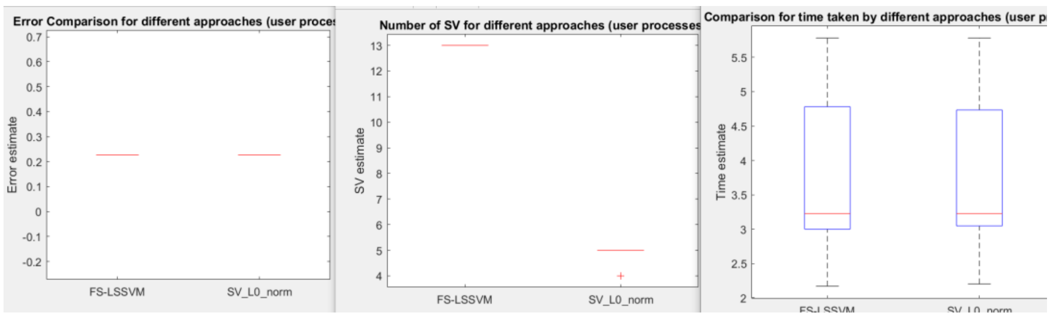


Figure 7 Spectral clustering for sig2 = 0.01, 1, 100

Increasing sig2 also increases the distance between groups of datapoints. For a smaller value, the subsets are not as dispersed, coming close to each other. The convergence seems to happen towards the periphery/observational cloud of the feature space (see image below, fixedsize_script2.m)



Running the fslssvm script gives following results:



There is some difference in the number of support vectors chosen between the two techniques but the time and error constraints are not too far apart. L0 approximation puts sparsity in LS-SVM by optimizing non zero elements of a vector.

HOMEWORK PROBLEMS

1. **Kernel PCA**
    a. **Illustrate the difference between linear and kernel PCA by giving an example of digit denoising for noisefactor = 1:0. Give your comments on the results (based on visual inspection).**
    b. **What happens when the sig2 parameter is much bigger than the suggested estimate?What if the parameter value is much smaller? In order to investigate this, change the sigmafactor parameter for equispaced values in logarithmic scale.**
    c. **Investigate the reconstruction error on training (Xtest) and validation sets (Xtest1 and Xtest2), as a function of the kernel PCA denoising parameters. Select parameter values such that the error on the validation sets is minimal. Can you observe any improvements in denoising using these optimized parameter settings?**

This dataset is based on handwritten digits and the main difference between linear and kernel PCA at noisefactor=1.

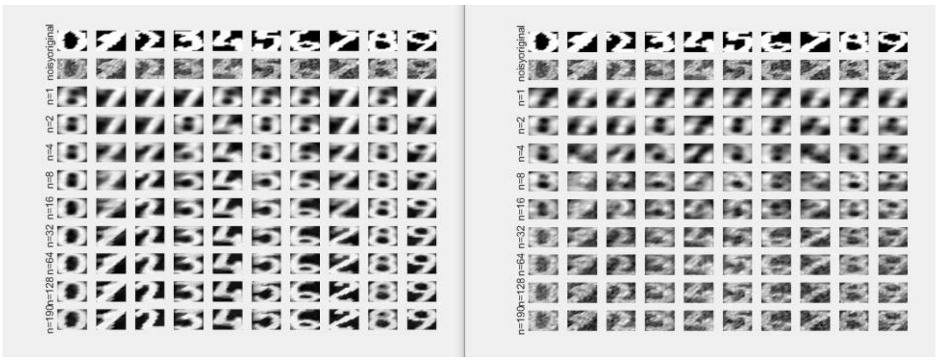


Figure 8 Kernel & Linear PCA at default sig2 [sigfactor=0.7]

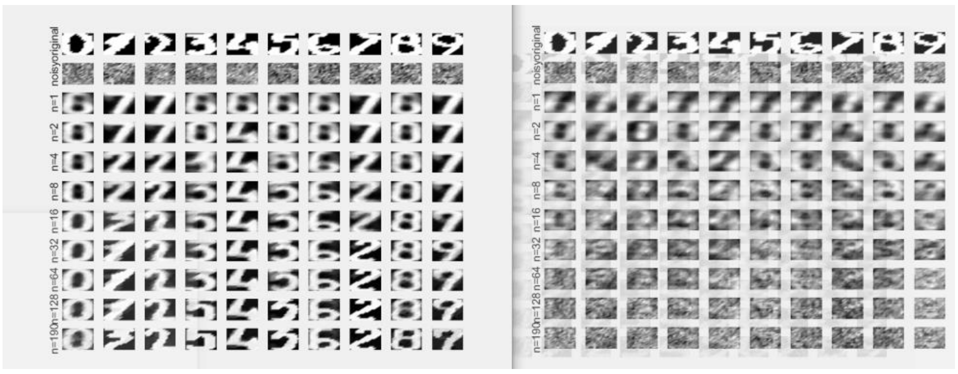


Figure 9 Kernel & Linear PCA at high sig2 = 20

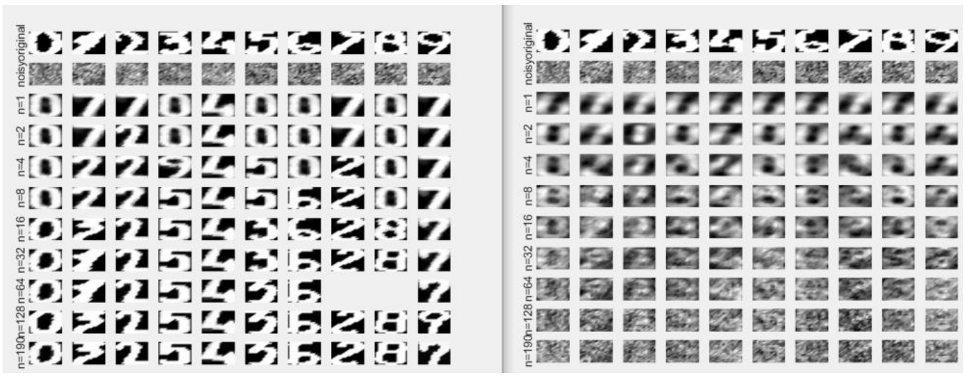
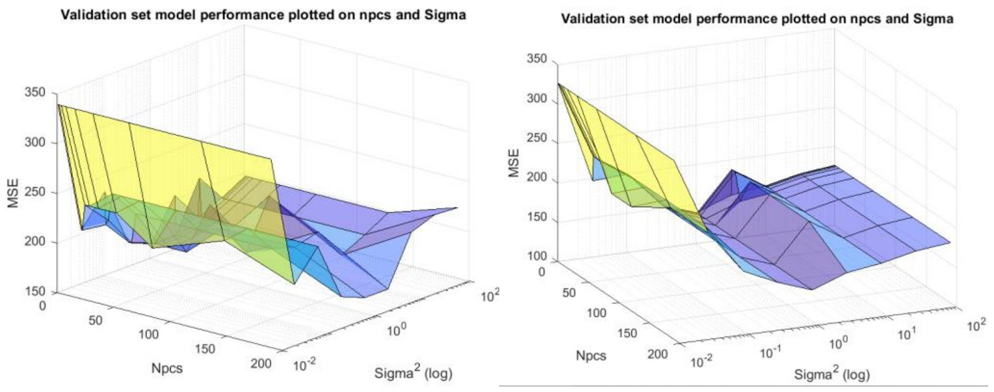


Figure 10 Kernel & Linear PCA at low sig2 [sigfactor = 0.2]

The increasing n (PCs) definitely improves the denoising of the dataset but there is evident clarity in kernel PCA from the very beginning. Increasing sig2 makes the kernel PCA blurrier. Decreasing the parameter gives a clearer image at the end; a sweet spot for the sig2 needs to be determined for the dataset.

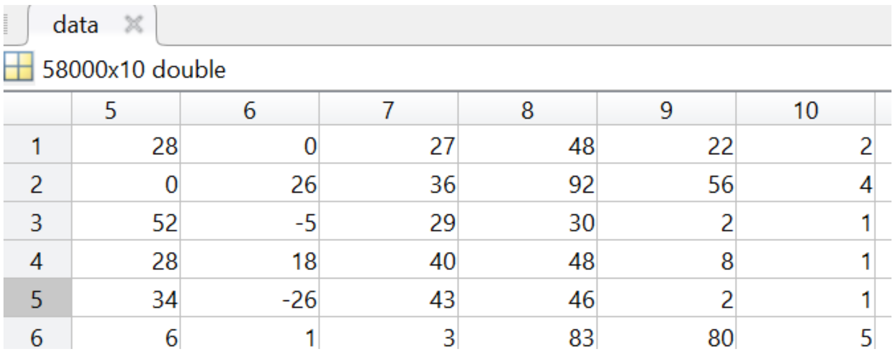
The reconstruction of the denoised images still shows some level of error, indicating that a minimum number of PCs is required for a good noise detection. Choosing a higher n (number of PCs) is a good way to counter this issue (it acts as a parameter for the kernel PCA). The choice of sigma is a significant factor as well, as can be seen from the images below for both the validation sets Xtest1 and Xtest2. The most optimal sig2 = [1-10], n = 200 for Xtest1 and sig2=1, n=200 as they give the minimum MSE. These values give an improved performance of the model.



2. Fixed size LS-SVM

a. Shuttle: Explore and visualize (part of) the dataset. How many datapoints? How many and meaning of attributes? How many classes? What is to be expected about classification performance? Visualize and explain the obtained results

The dataset is 58000x10 in dimension and carries different ranges of values in each column. It is not standardized and most of the columns are sparsely populated (multiple zeros). Although there are a total of 9 numerical classes, 80% data belongs to one class (https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)) and this makes it an unbalanced dataset.


data

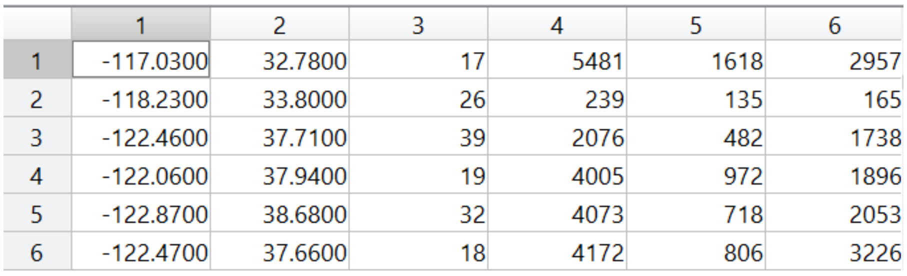
58000x10 double

	5	6	7	8	9	10
1	28	0	27	48	22	2
2	0	26	36	92	56	4
3	52	-5	29	30	2	1
4	28	18	40	48	8	1
5	34	-26	43	46	2	1
6	6	1	3	83	80	5

This dataset would best benefit from cross validation strategies that do not end uo training on just one class data, otherwise this bias will affect its performance adversely. The LS-SVM model for this dataset is already described in section 3.3

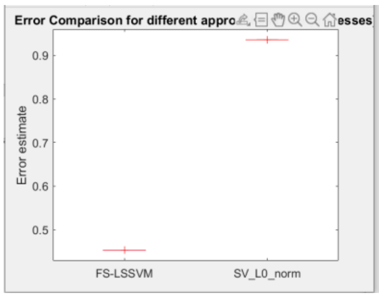
b. California: Explore and visualize (part of) the dataset. How many datapoints? How many and meaning of attributes? Visualize and explain the obtained result

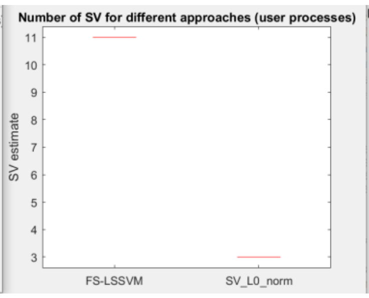
This dataset is 20640x9 dimensionally and carries information regarding the geographical location of densely populated areas in California. It has values like age of house & number of rooms that have not been normalized.

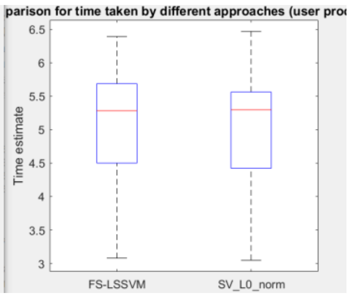


	1	2	3	4	5	6
1	-117.0300	32.7800	17	5481	1618	2957
2	-118.2300	33.8000	26	239	135	165
3	-122.4600	37.7100	39	2076	482	1738
4	-122.0600	37.9400	19	4005	972	1896
5	-122.8700	38.6800	32	4073	718	2053
6	-122.4700	37.6600	18	4172	806	3226

There are more errors in L0 approximation for this dataset as compared to FS-LSSVM and this may be because it uses less support vectors; the time taken by both approaches is very similar.


Error Comparison for different appro esses
Error estimate
0.9
0.8
0.7
0.6
0.5
FS-LSSVM
SV_L0_norm


Number of SV for different approaches (user processes)
SV estimate
11
10
9
8
7
6
5
4
3
FS-LSSVM
SV_L0_norm


parison for time taken by different approaches (user proc
Time estimate
6.5
6
5.5
5
4.5
4
3.5
3
FS-LSSVM
SV_L0_norm